KECE471 Computer Vision

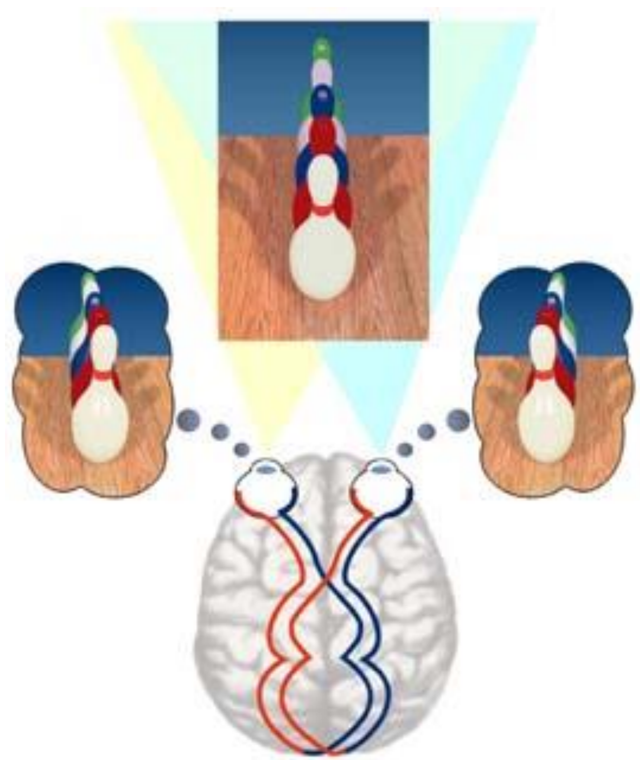
Stereo

Chang-Su Kim

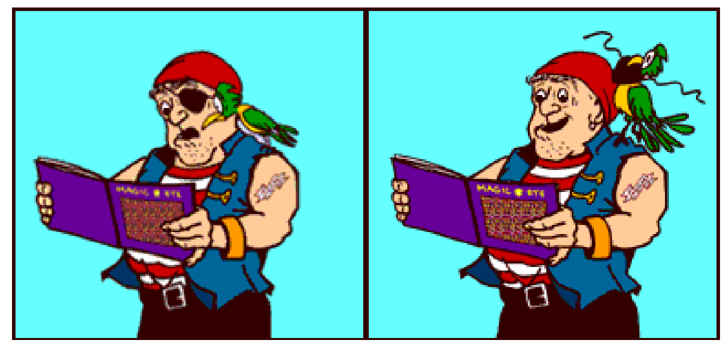
Chapter 11, Computer Vision by Forsyth and Ponce
Note: Most contents were copied from the lecture notes of Prof. Kyeong Mu Lee in SNU

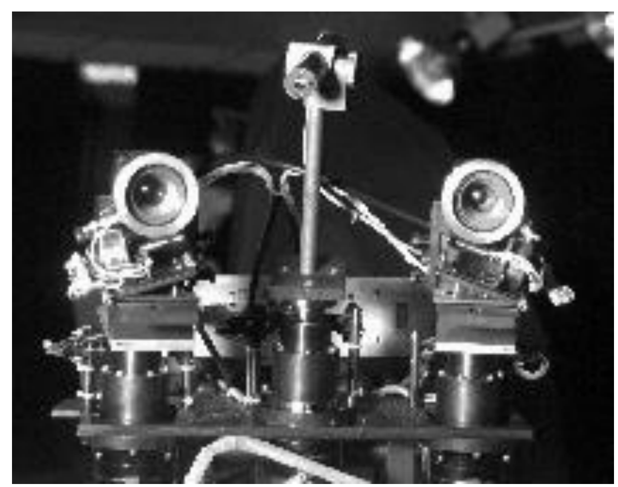
Stereo

- Inferring depth information using two cameras like a human
- Two eyes perceives three-dimension



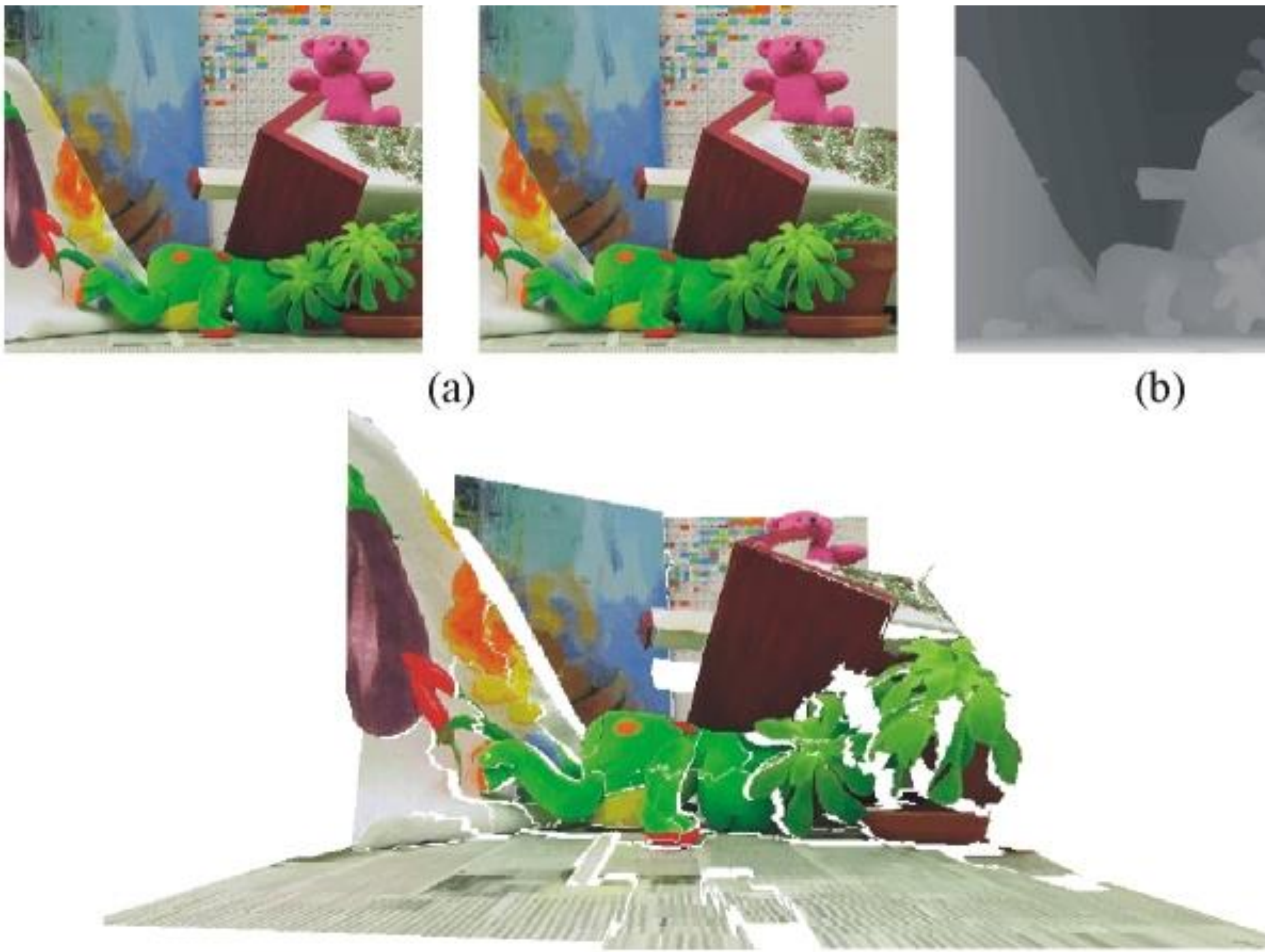
Human eyes





Robot eyes

Stereo



(a) (b) (c)

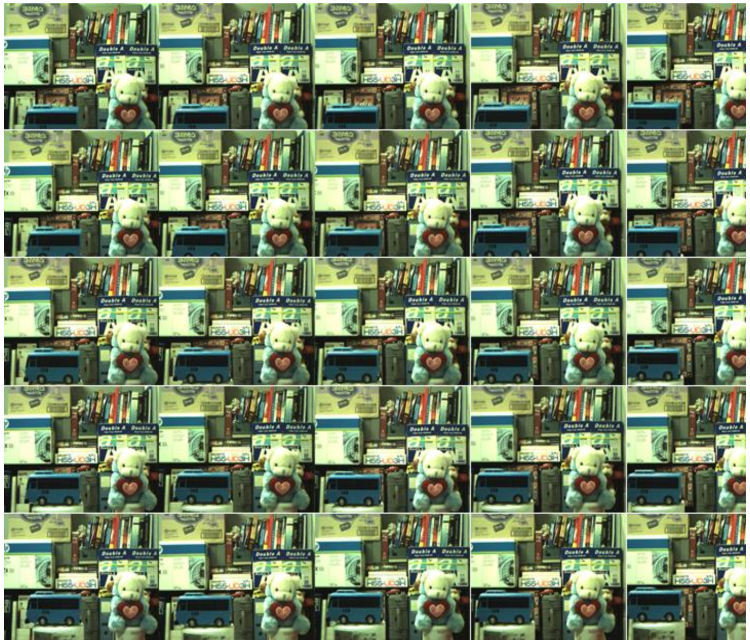
Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

Teesta suspension bridge-Darjeeling, India

Stereo

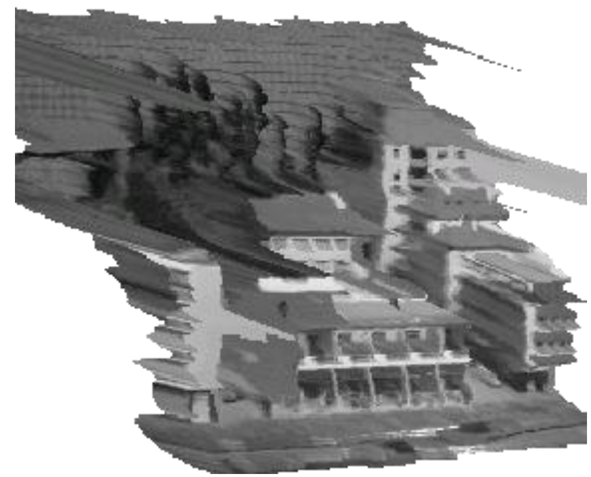
- Inferring depth information using two eyes or cameras
- Two eyes perceive 3rd dimension

(a)



(b)

Applications



[Matthies,Szeliski,Kanade'88]

Applications

input image

317 images (hemisphere)

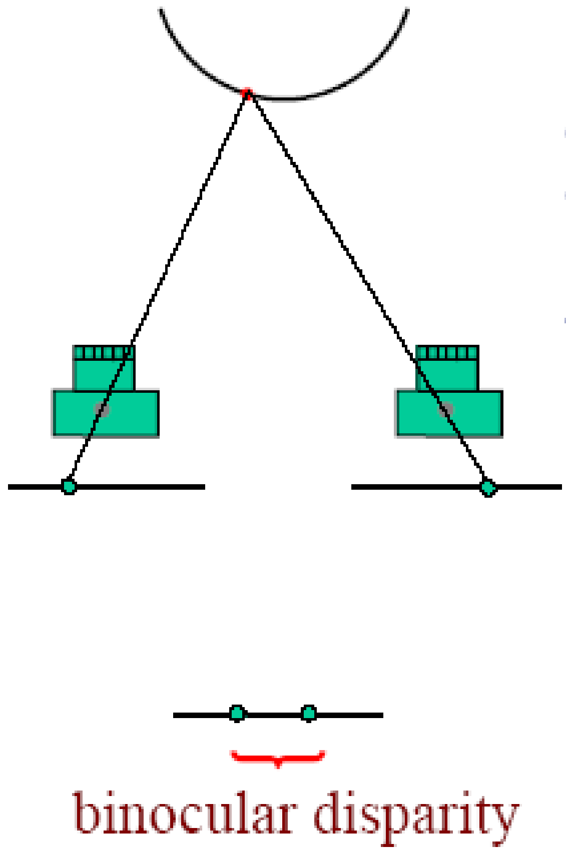
ground truth model

Goesele, Curless, Seitz, 2006

Binocular Stereo

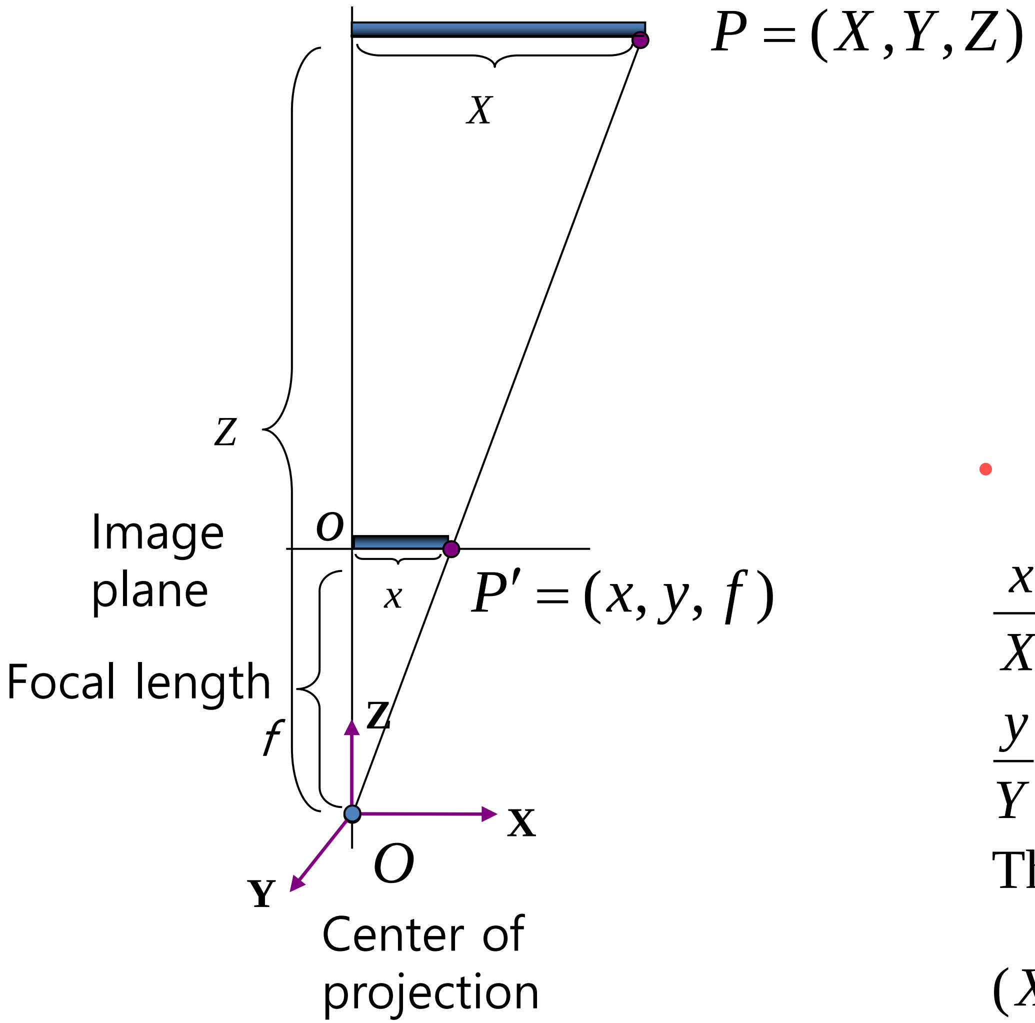
From known geometry of the cameras and estimated disparity, recover depth in the scene

Left



Right

Pinhole Camera Model



- 3D to 2D projection:

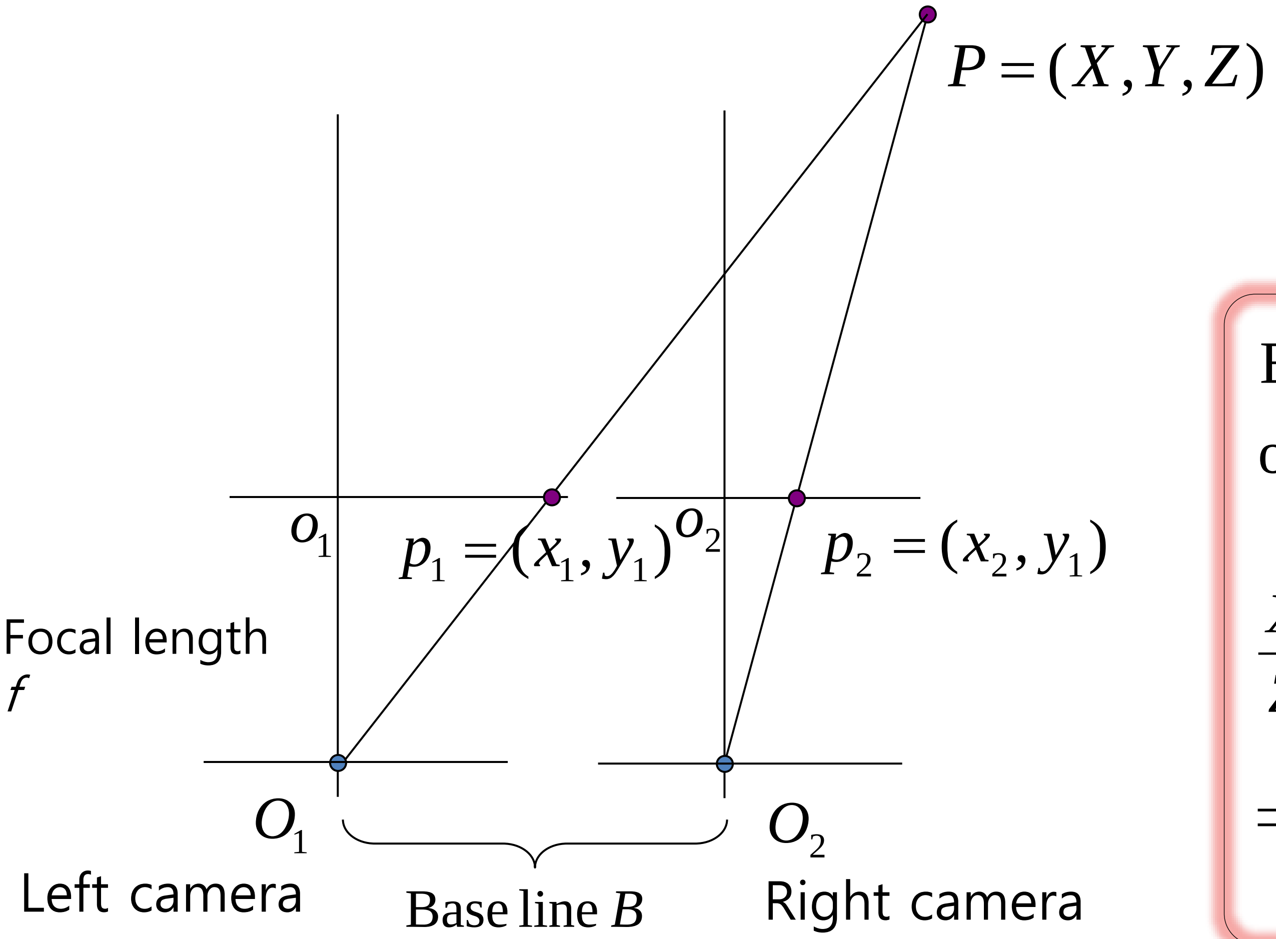
$$\frac{x}{X} = \frac{f}{Z} \Rightarrow x = f\,\frac{X}{Z}$$

$$\frac{y}{Y} = \frac{f}{Z} \Rightarrow y = f\,\frac{Y}{Z}$$

Thus

$$(X, Y, Z) \rightarrow (x, y) = (f\,\frac{X}{Z},\ f\,\frac{Y}{Z})$$

Basic Stereo Model

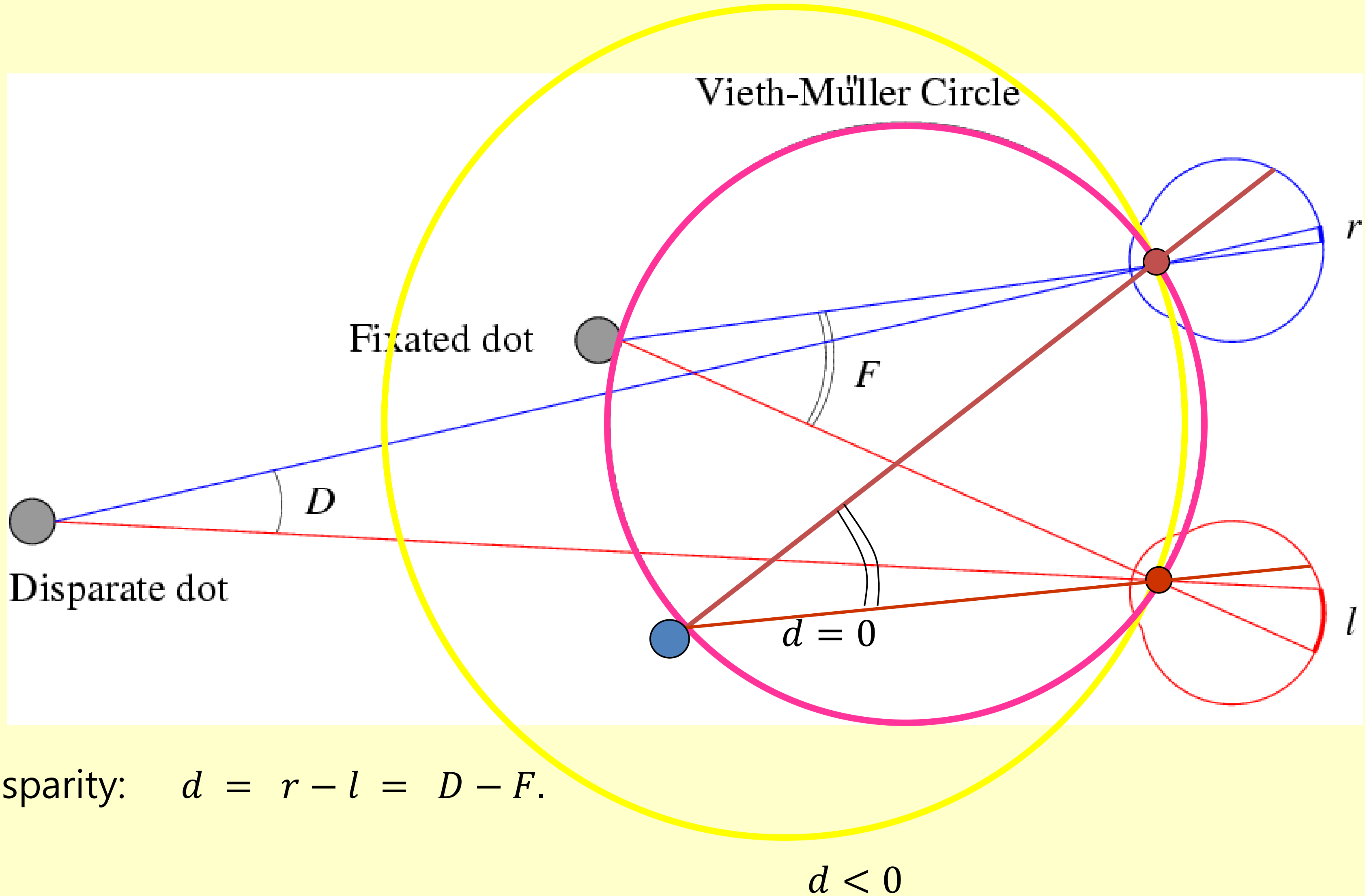


Express Z as a function of x_1, x_2, f, B

$$\frac{X}{Z} = \frac{x_1}{f} \text{ and } \frac{X - B}{Z} = \frac{x_2}{f}$$

$$\Rightarrow \boxed{Z = \frac{fB}{x_1 - x_2} = \frac{fB}{d(p_1)}}$$

Human Stereopsis: Reconstruction



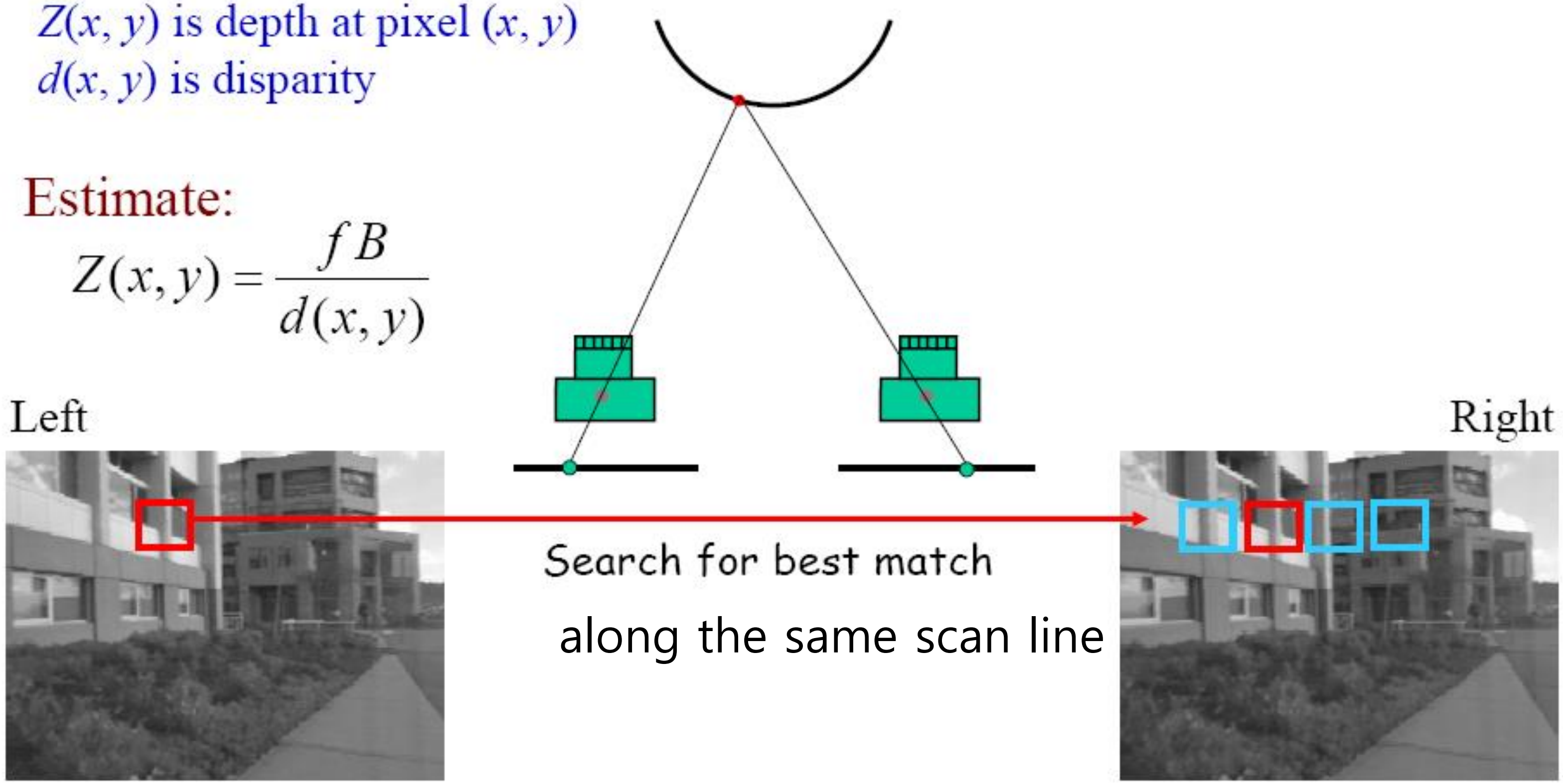
Disparity: $d \;=\; r - l \;=\; D - F.$

Finding Correspondence

$Z(x, y)$ is depth at pixel (x, y)
$d(x, y)$ is disparity

Estimate:

$$Z(x,y) = \frac{f\,B}{d(x,y)}$$

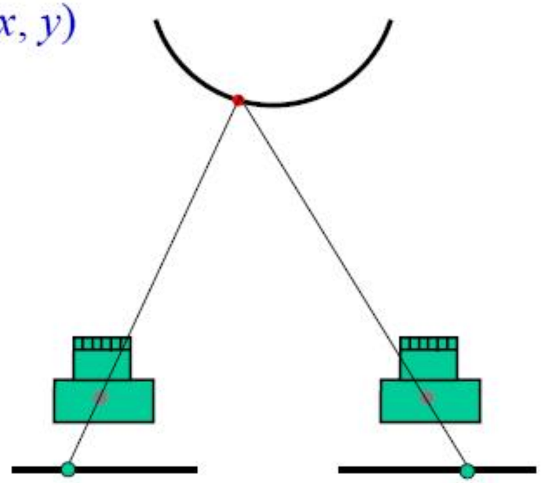


Finding Correspondence

$Z(x, y)$ is depth at pixel (x, y)
$d(x, y)$ is disparity

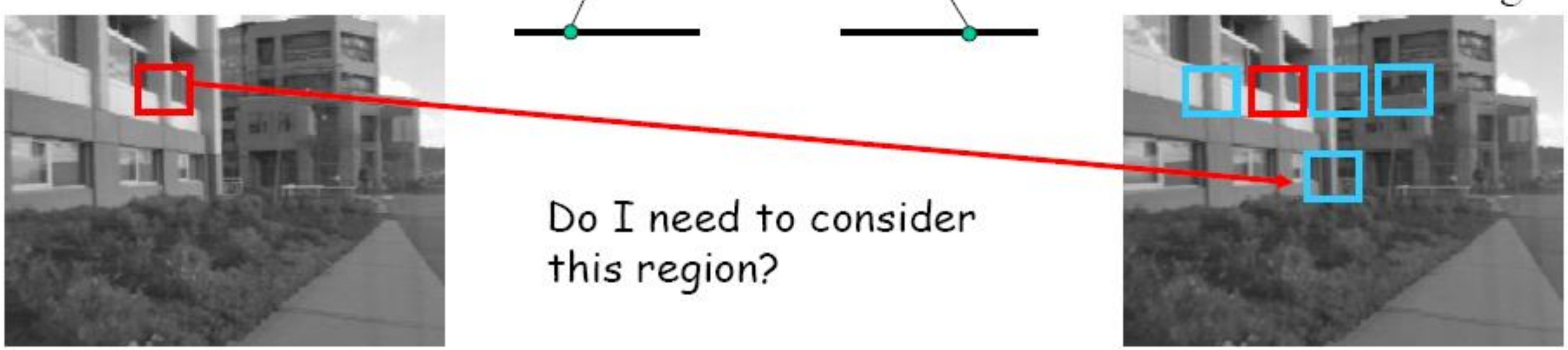
Estimate:

$$Z(x,y) = \frac{fB}{d(x,y)}$$



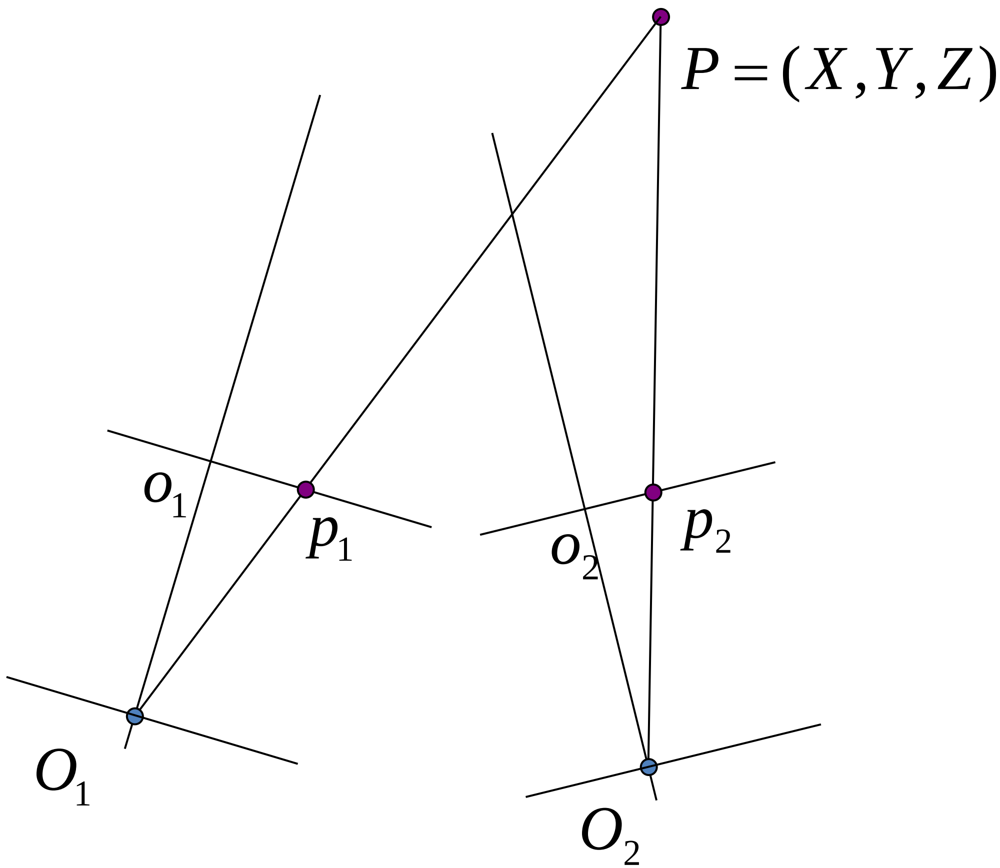
Left

Right

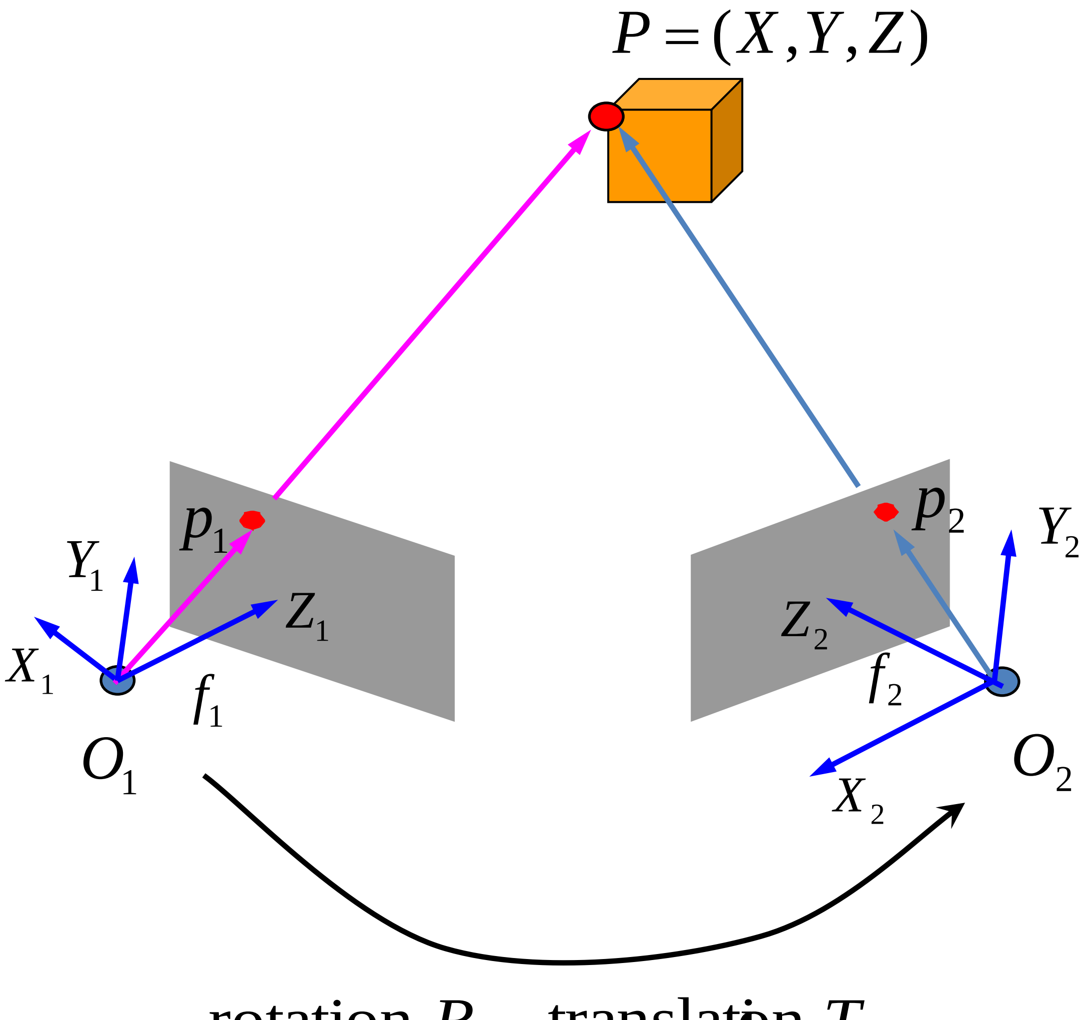


General stereo

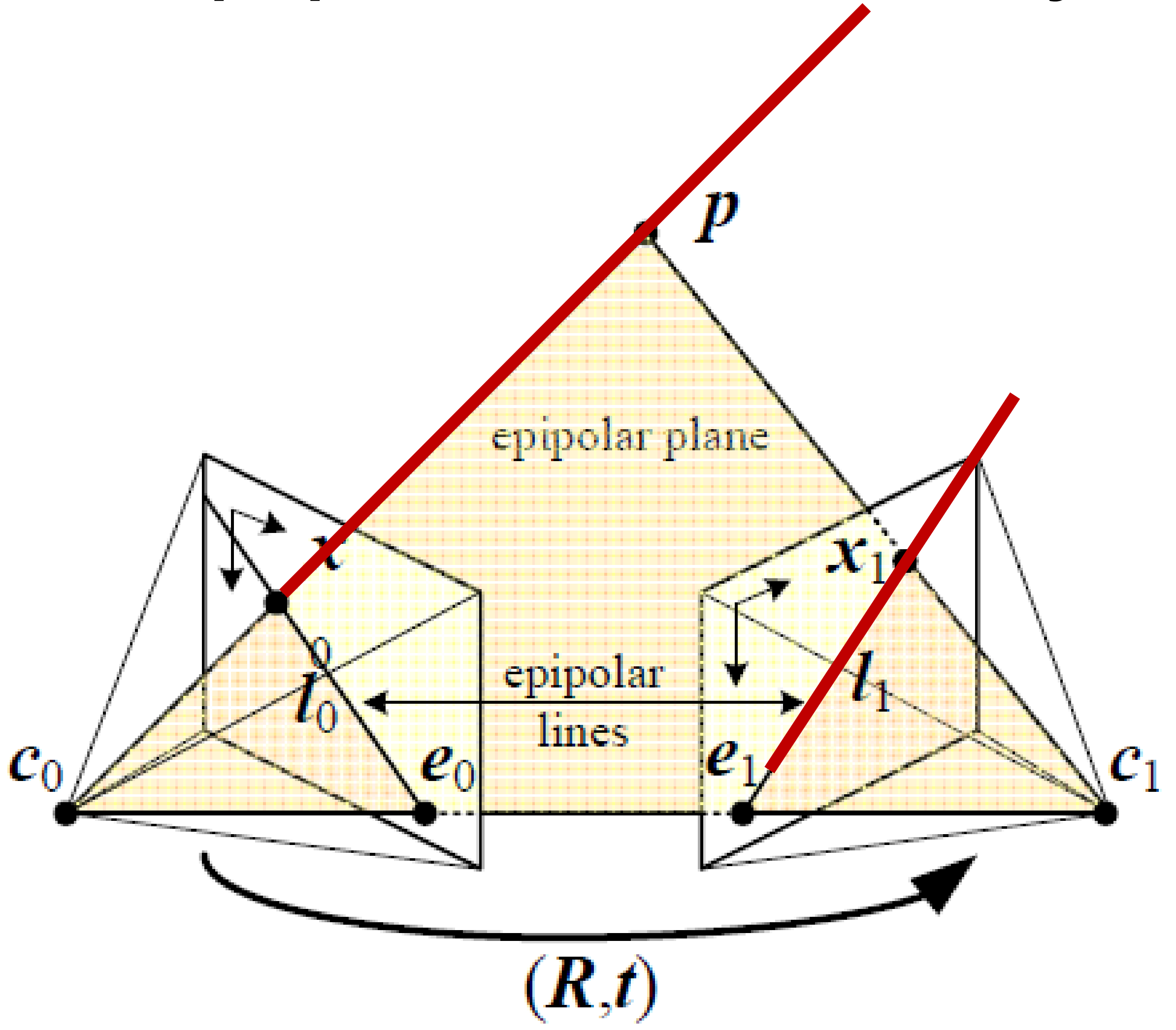
- What if two cameras are not parallel?



Epipolar Geometry

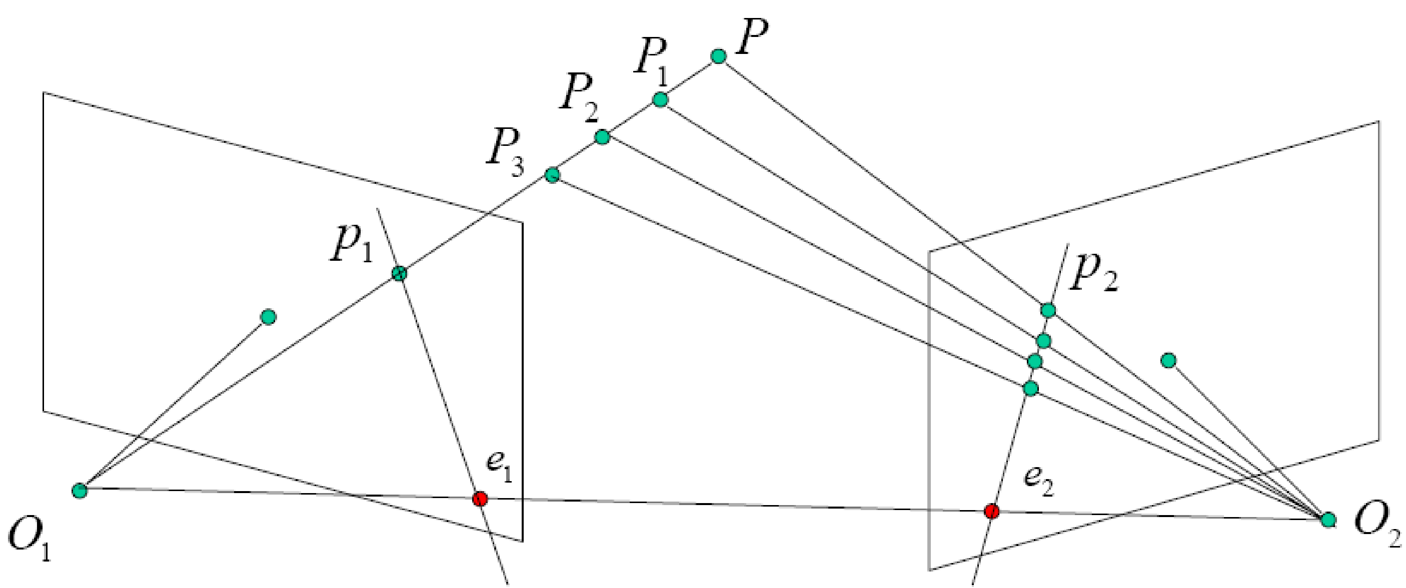


Epipolar Geometry



Epipolar Geometry

- Epipolar Constraint
  - A matching points lies on the associated epipolar line
  - It reduces the correspondence problem to 1D search along the epipolar line
  - It reduces the cost and ambiguity of matching



Rectification

- Simple case
  - Cameras are parallel
  - Focal lengths are the same
  - Two image planes lie on the same plane

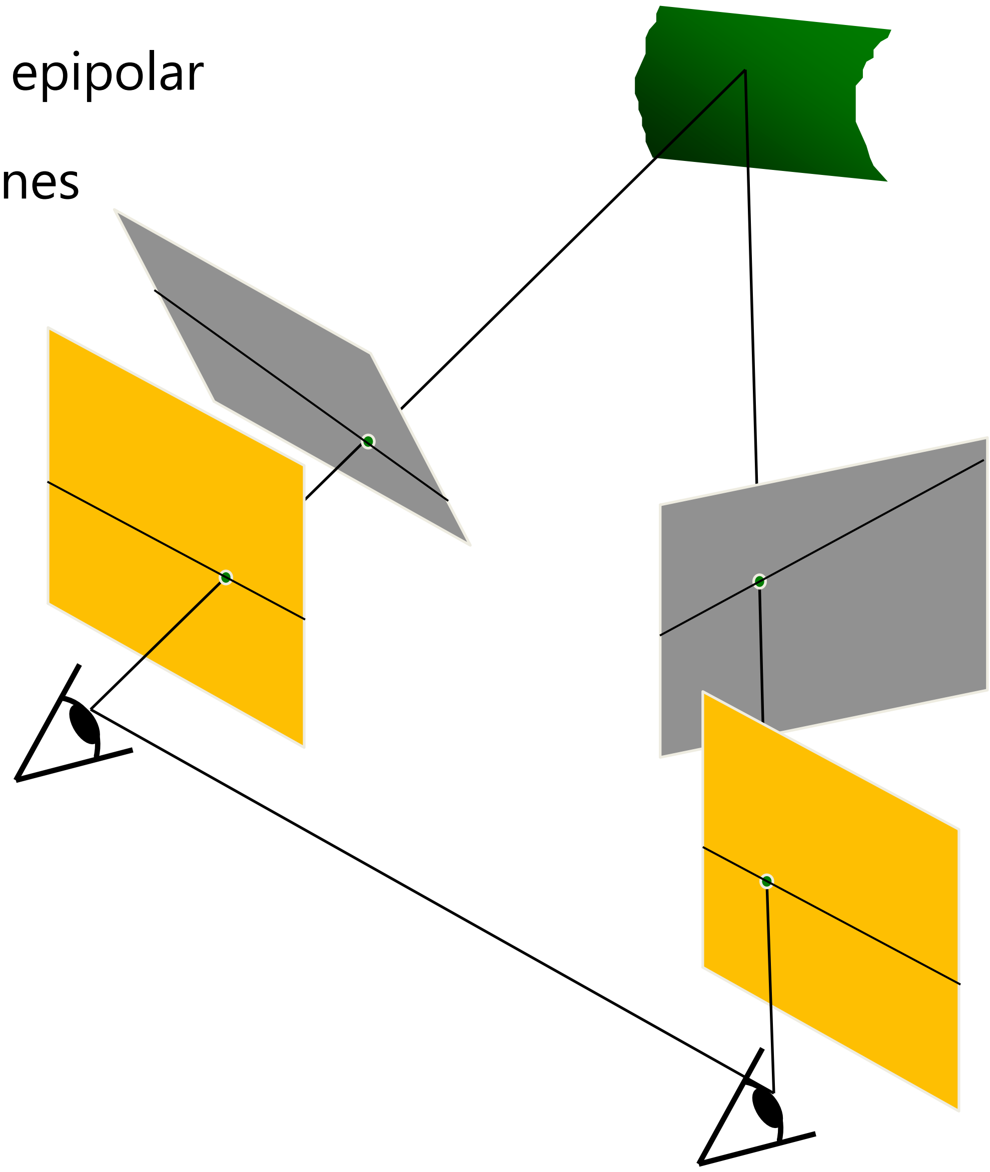
- Then, epipolar lines correspond to scan lines

- Rectification is a procedure to convert images so that the assumptions are satisfied
  - It simplifies algorithms
  - It improves efficiency

[KM Lee, Lecture Notes]

Rectification

- Reproject (warp) images so that epipolar lines are aligned with the scan lines



Rectification

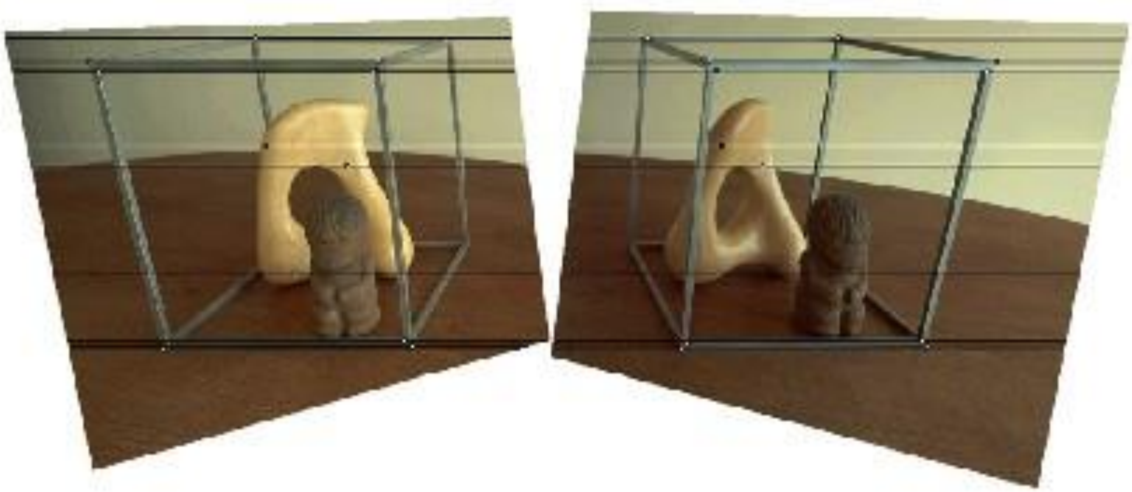
(a) Original image pair overlayed with several epipolar lines.

(b) Image pair transformed by the specialized projective mapping $\mathbf{H}_p$ and $\mathbf{H}'_p$. Note that the epipolar lines are now parallel to each other in each image.

[Loop and Zhang, CVPR'99]

Rectification

(c) Image pair transformed by the similarity $\mathbf{H}_r$ and $\mathbf{H}'_r$. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).



(d) Final image rectification after shearing transform $\mathbf{H}_s$ and $\mathbf{H}'_s$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

[Loop and Zhang, CVPR'99]

Correspondence: What to Match?

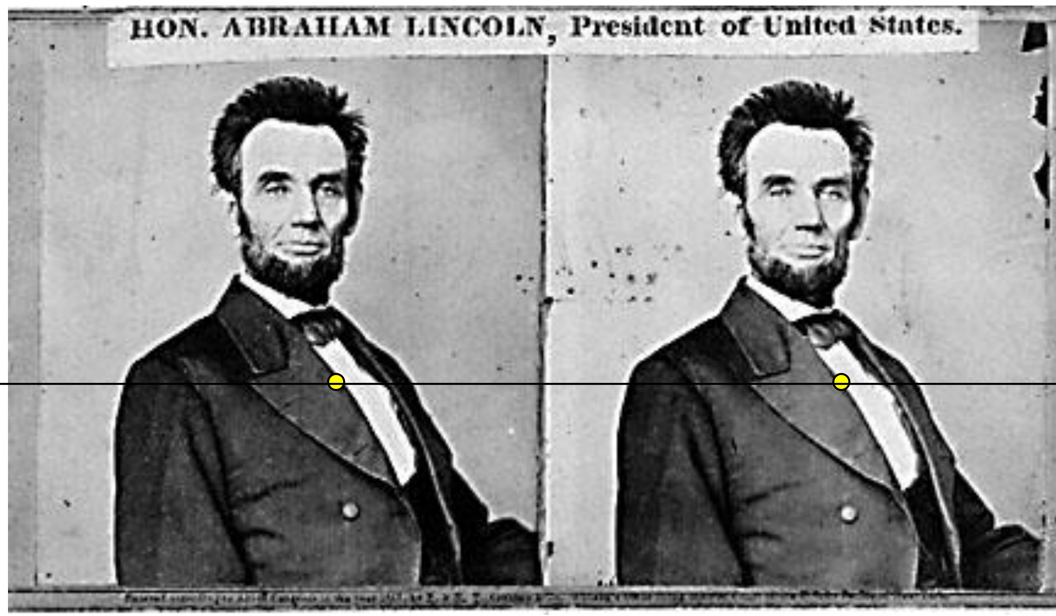
- Objects?
  - More identifiable, but difficult to compute
- Pixels?
  - Easier to handle, but maybe ambiguous
- Edges?
- Collections of pixels (regions)?

Correspondence: Photometric Constraint

- Assume that the same world point has the same intensity in both images.
  - However, it is not true in general
    - Noise
    - Illumination
    - Camera calibration

Pixel Matching

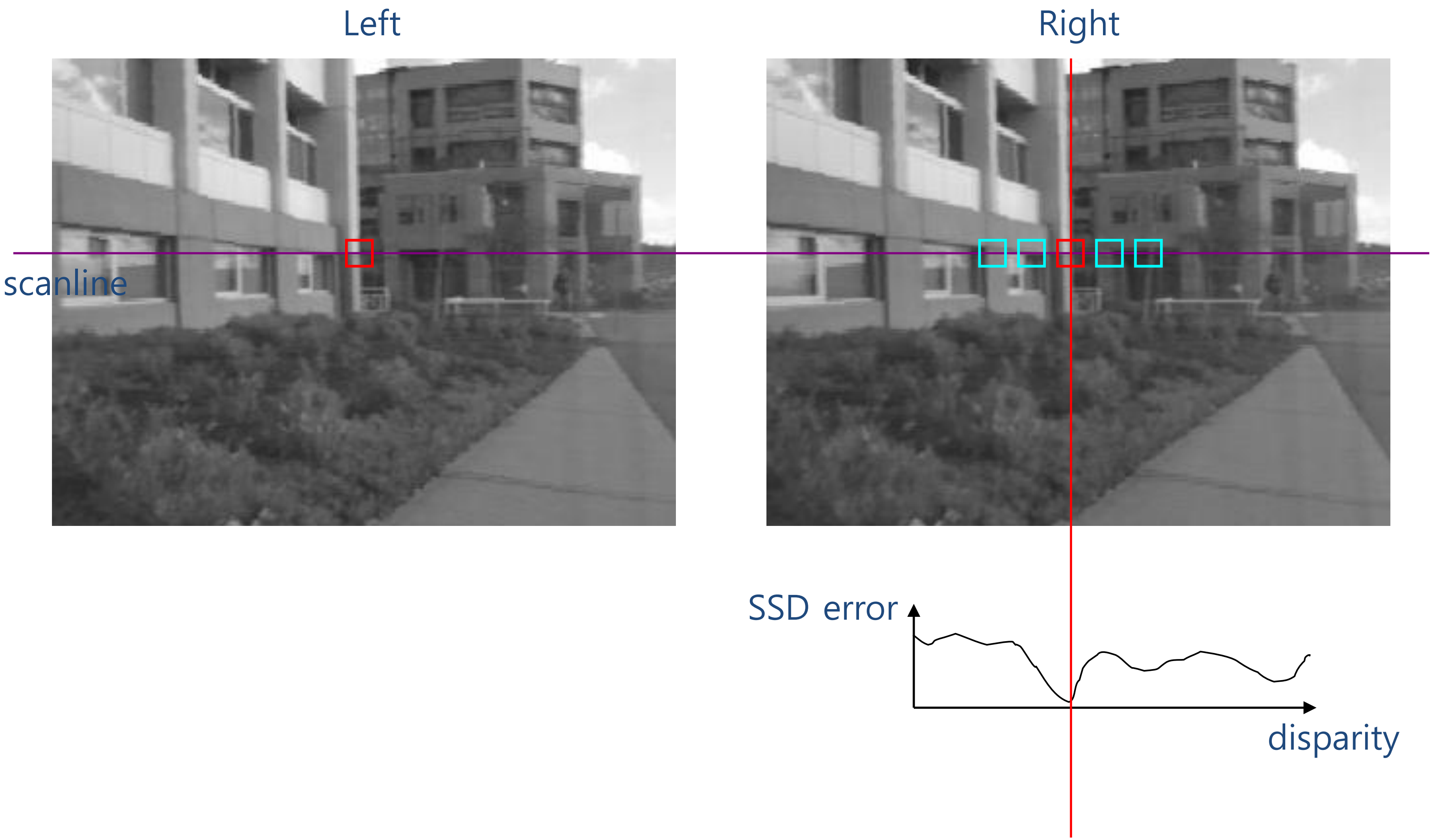
What if ?



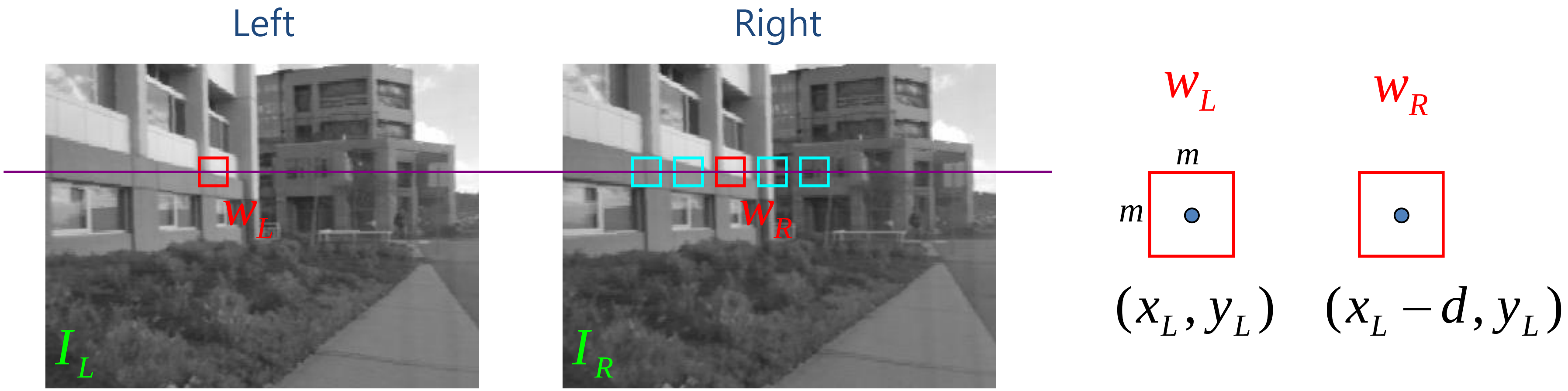
For each scanline , for each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost
- This will never work, so: **match windows**

Correspondence Using Window Matching



SSD



- Two blocks $\boldsymbol{w}_L$ and $\boldsymbol{w}_R$
- $SSD = \|\boldsymbol{w}_L - \boldsymbol{w}_R\|^2$

Normalization

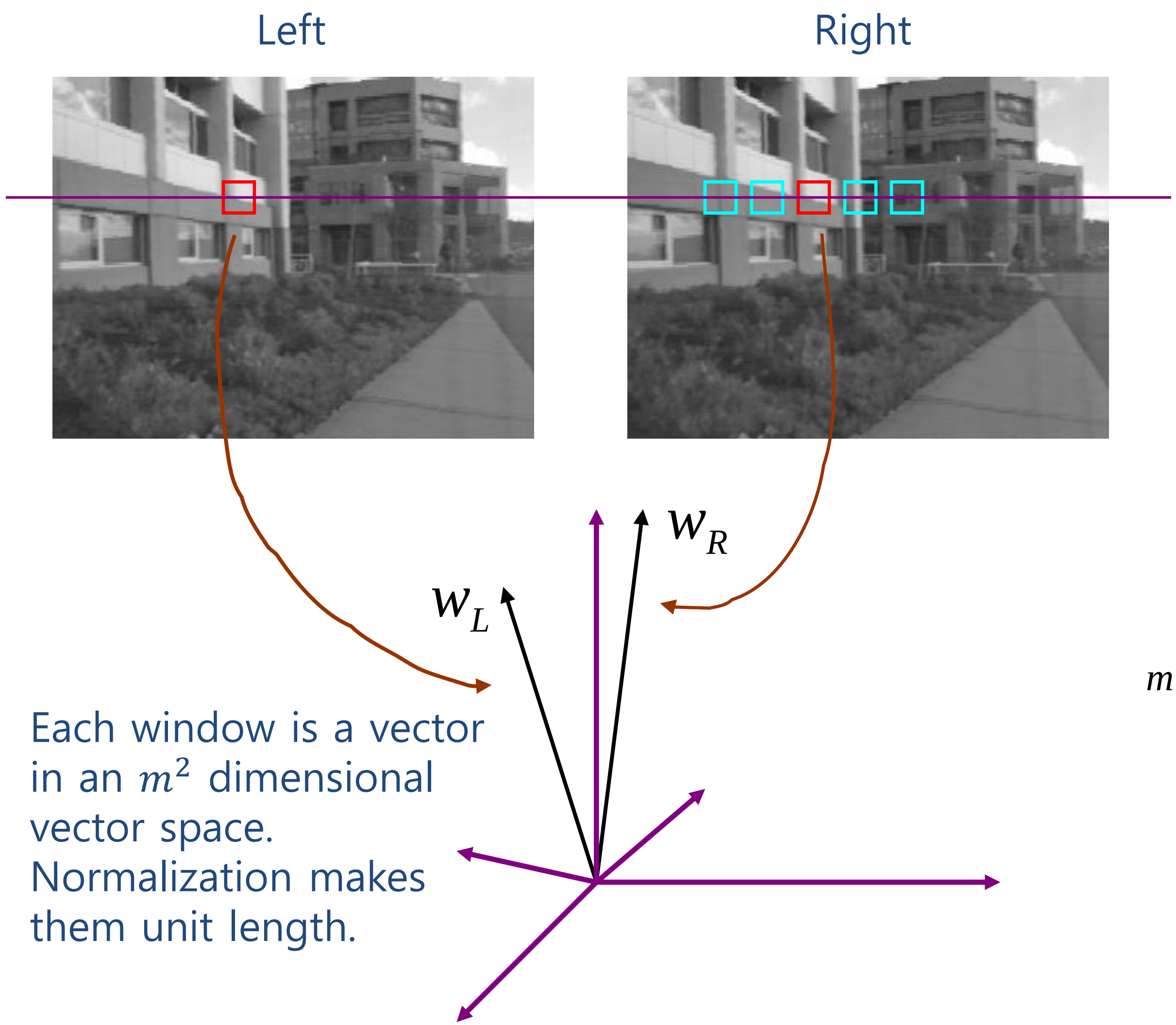
- There can be differences in gain and sensitivity
- Normalize the pixels in each window

$$\tilde{\boldsymbol{w}} = \frac{\boldsymbol{w} - \mu\boldsymbol{1}}{\|\boldsymbol{w} - \mu\boldsymbol{1}\|}$$

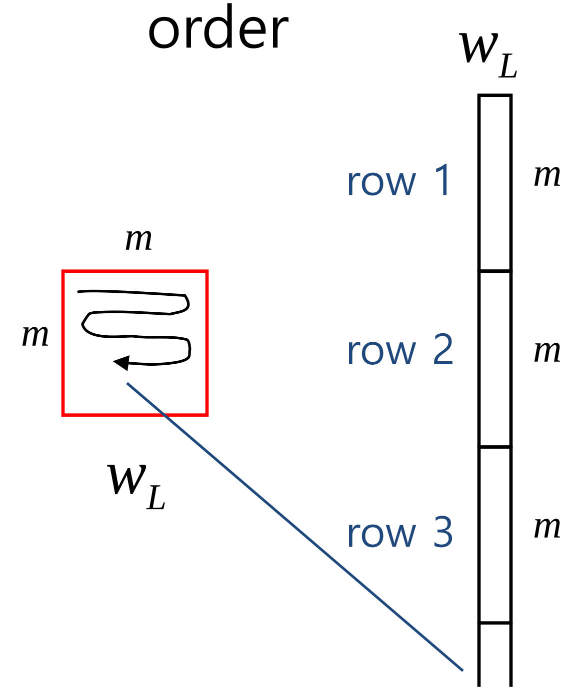
- Minimizing SSD becomes maximizing NCC (normalized cross correlation)

$$\|\tilde{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_R\|^2 = 2 - 2\tilde{\boldsymbol{w}}_L \cdot \tilde{\boldsymbol{w}}_R$$

Normalization



"Unwrap" image to form vector, using raster scan order

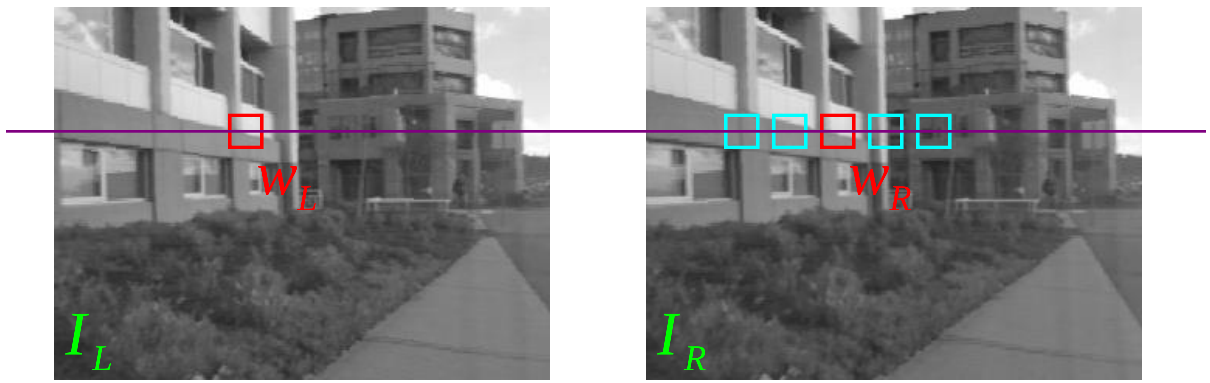


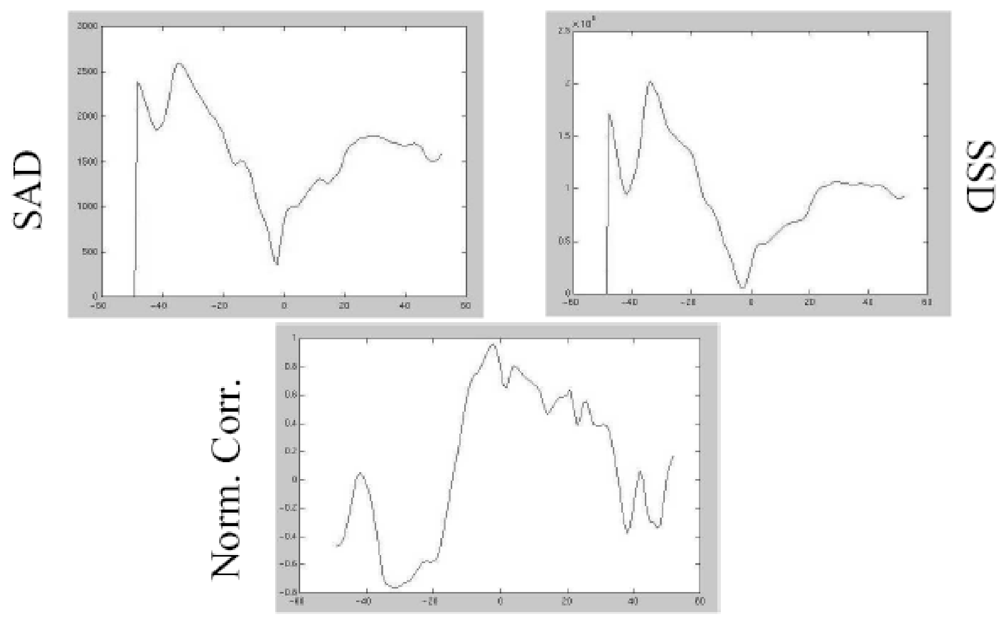
Each window is a vector in an m^2 dimensional vector space. Normalization makes them unit length.

Distance Metrics

Left

Right





Stereo Results

Images courtesy of Point Grey Research

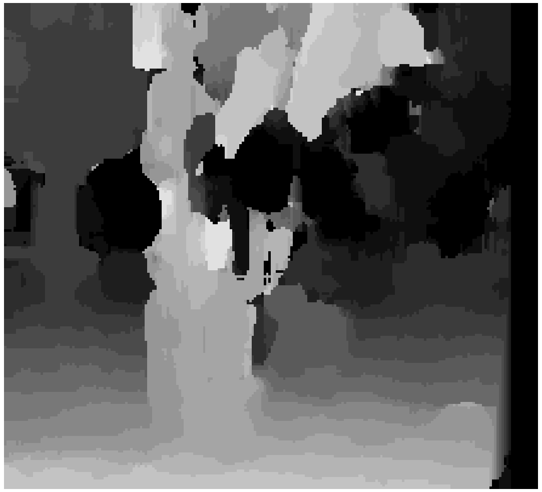
Disparity Map

Problems with Window-Based Matching

- Disparity within the window may not be constant
- Blur across depth discontinuities
- Poor performance in textureless regions
- Erroneous results in occluded regions

Window Size

W = 3



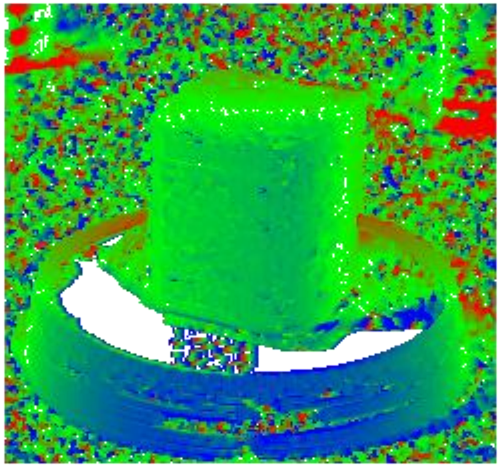
W = 20

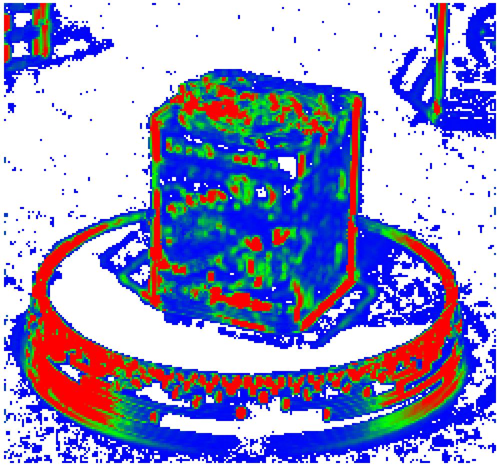
- The results depend on the window size
- Some approaches have been developed to use an adaptive window size (try multiple sizes and select best match)

Certainty Modeling

- Compute certainty map from correlations


input


depth map


certainty map

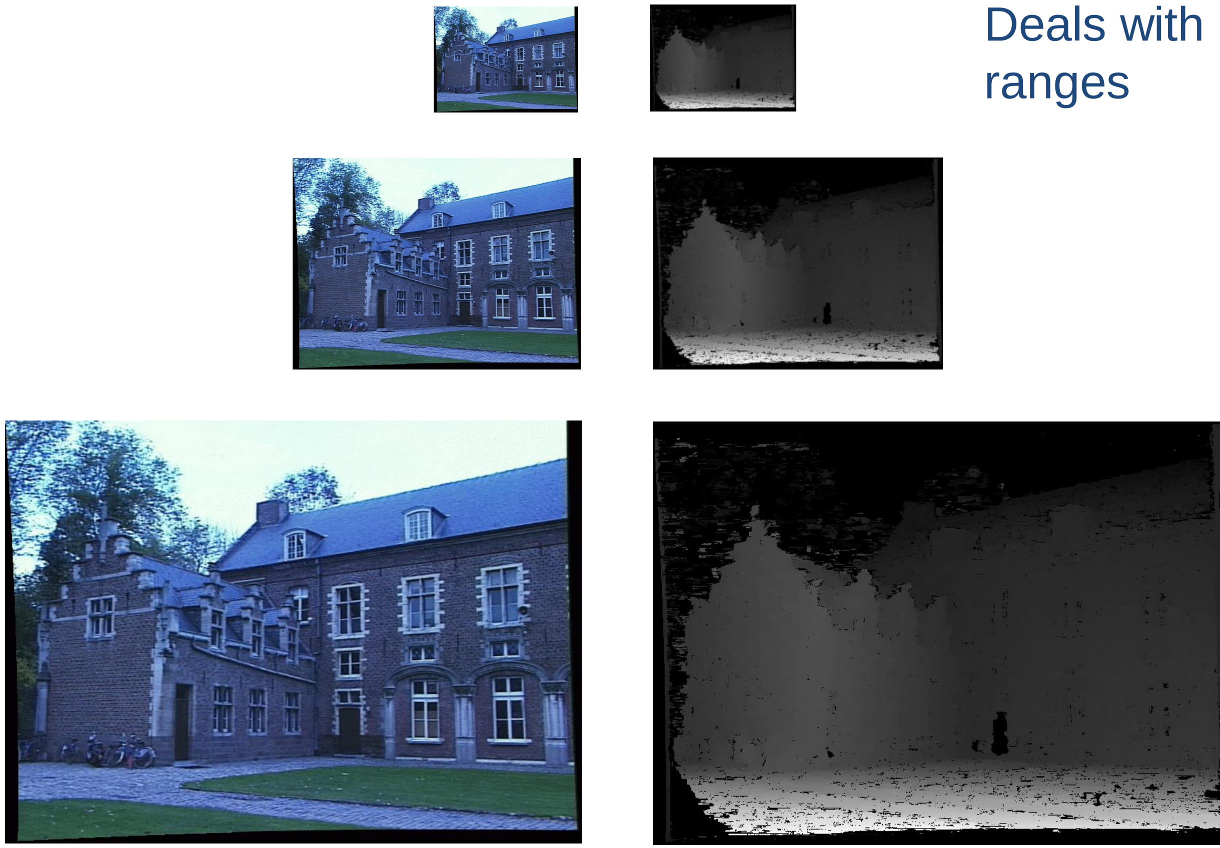
[Szeliski, 1991]

Hierarchical Stereo Matching

Allows faster computation

Deals with large disparity ranges

Downsampling (Gaussian pyramid)



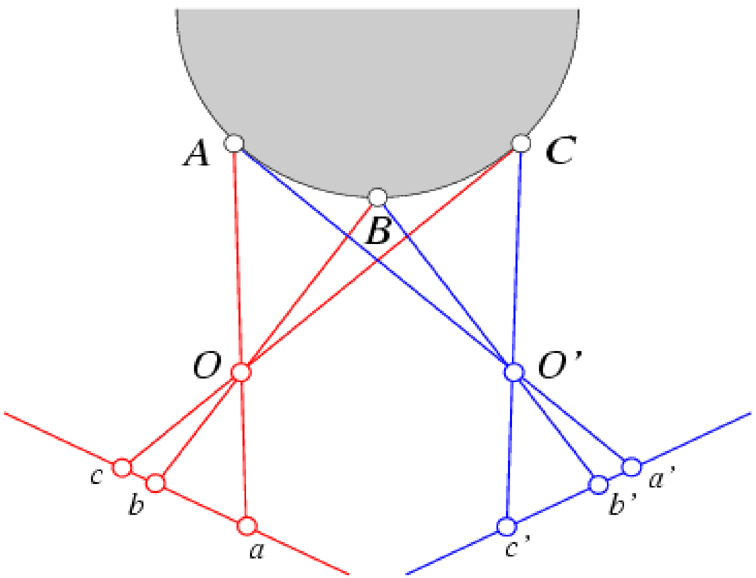
Disparity propagation

(Falkenhagen´97;Van Meerbergen,Vergauwen,Pollefeys,VanGool IJCV‘02)

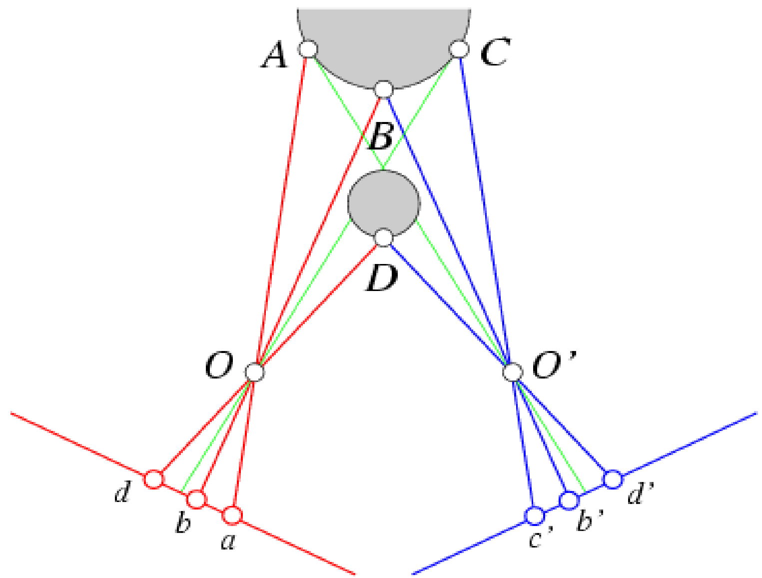
Stereo Matching Using Dynamic Programming

Ordering Constraint

- Points on the epipolar lines appear in the same order
- It may not be true in some cases, but can be assumed for most cases
- This is the basic assumption of the stereo matching using dynamic programming

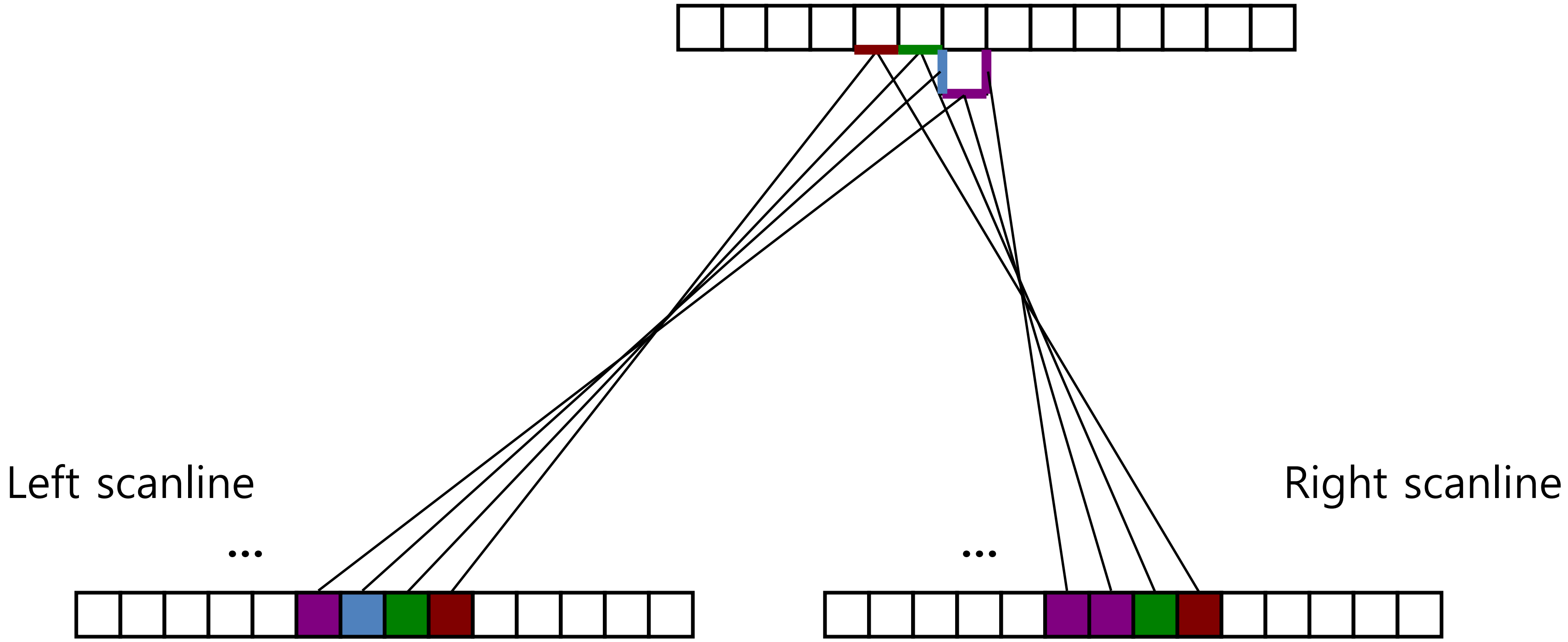


Ordering constraint…

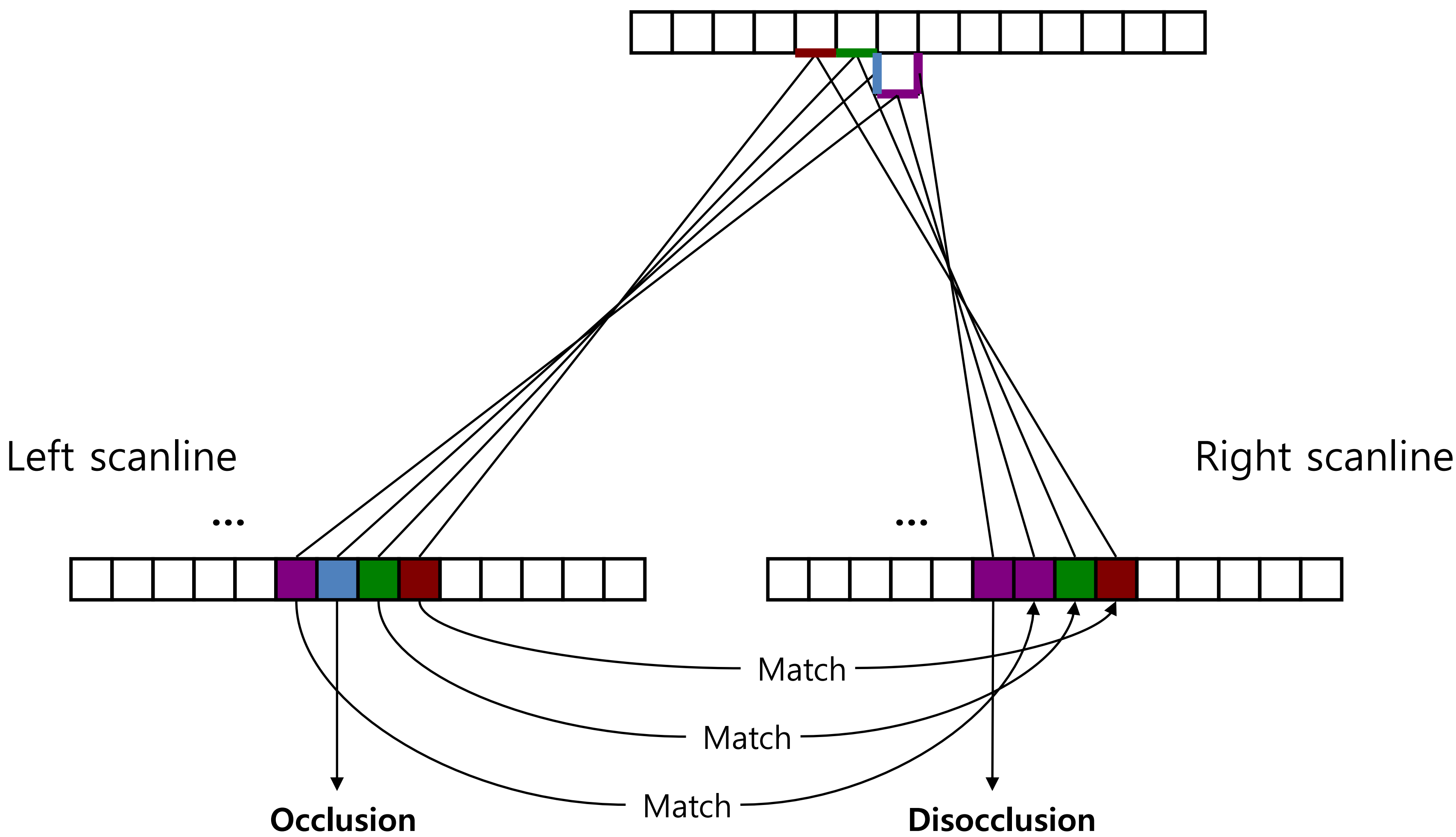


…and its failure

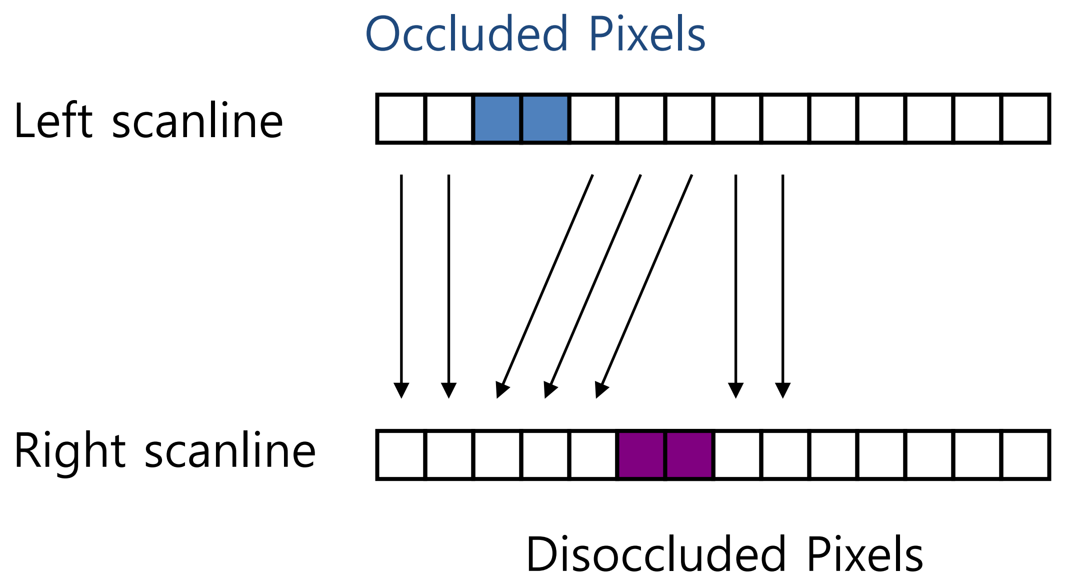
Occlusion and Disocclusion



Occlusion and Disocclusion



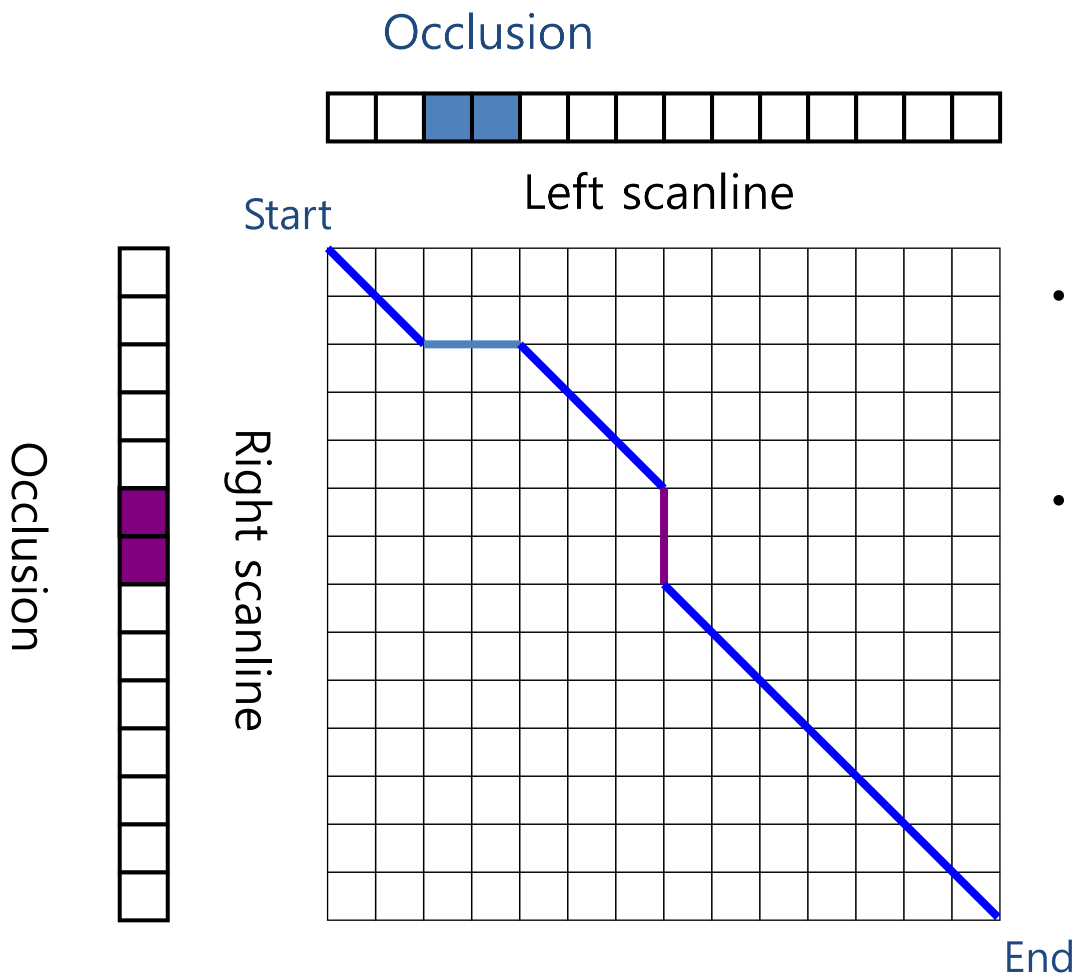
Search over Correspondences



Three cases:

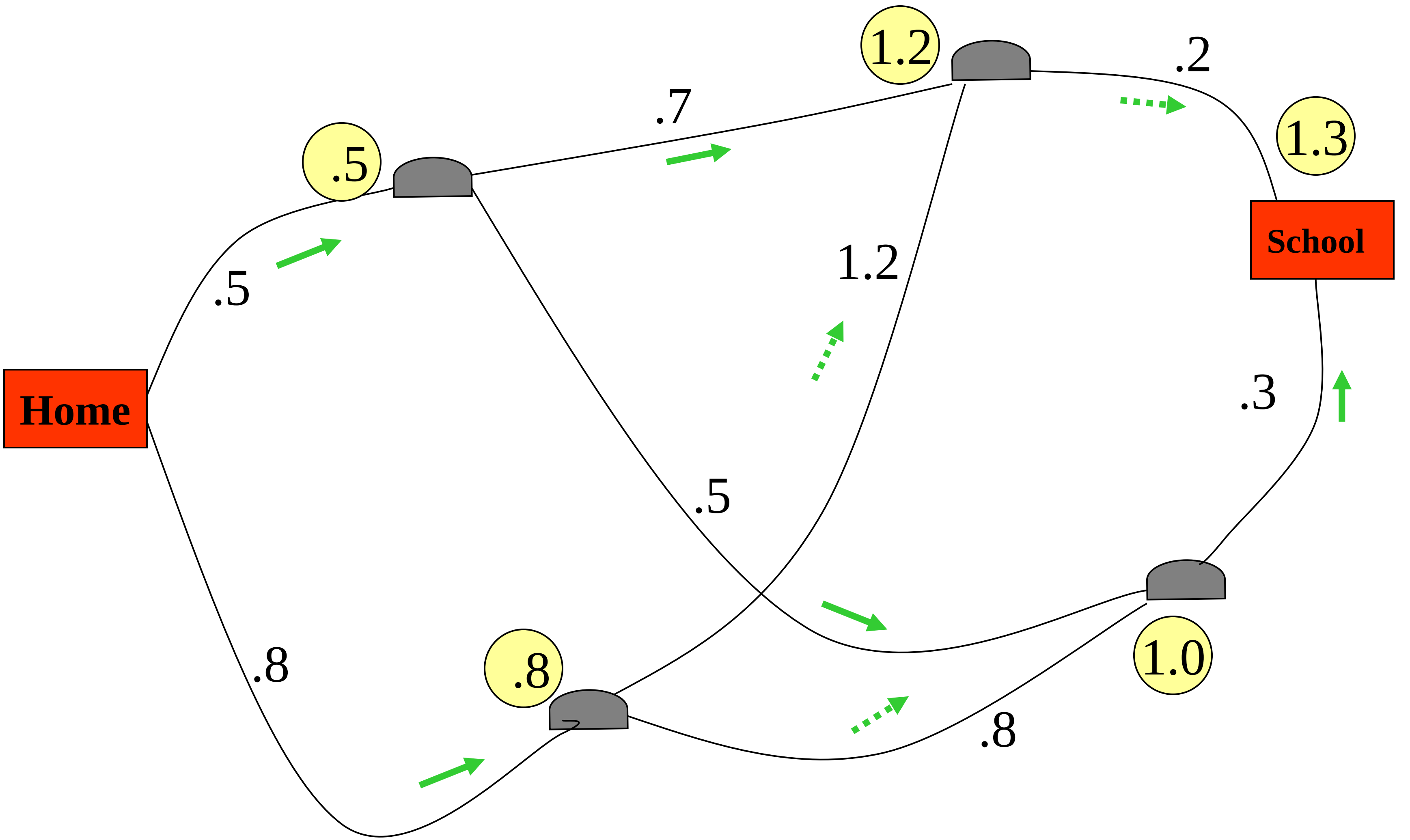
- Sequential – add cost of match (small if intensities agree)
- Occluded – add cost of no match (large cost)
- Disoccluded – add cost of no match (large cost)

Dynamic Programming Approach

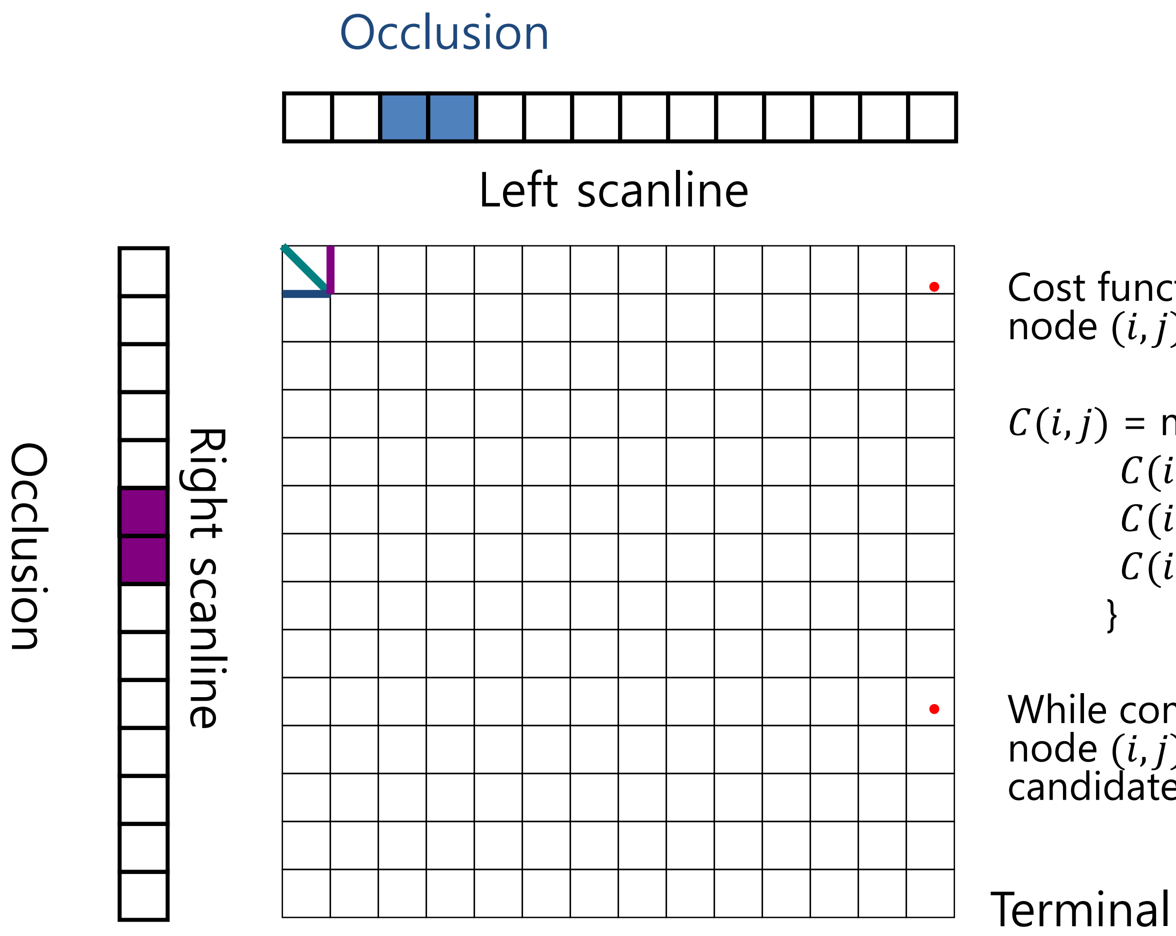


- Dynamic programming yields the optimal path, satisfying the ordering constraint
- Every segment on each scan line will be labeled as either matching or occlusion
  - Diagonal arc: matching
  - Horizontal arc: left occlusion
  - Vertical arc: right occlusion

Bellman’s Optimality Principle



Dynamic Programming Approach

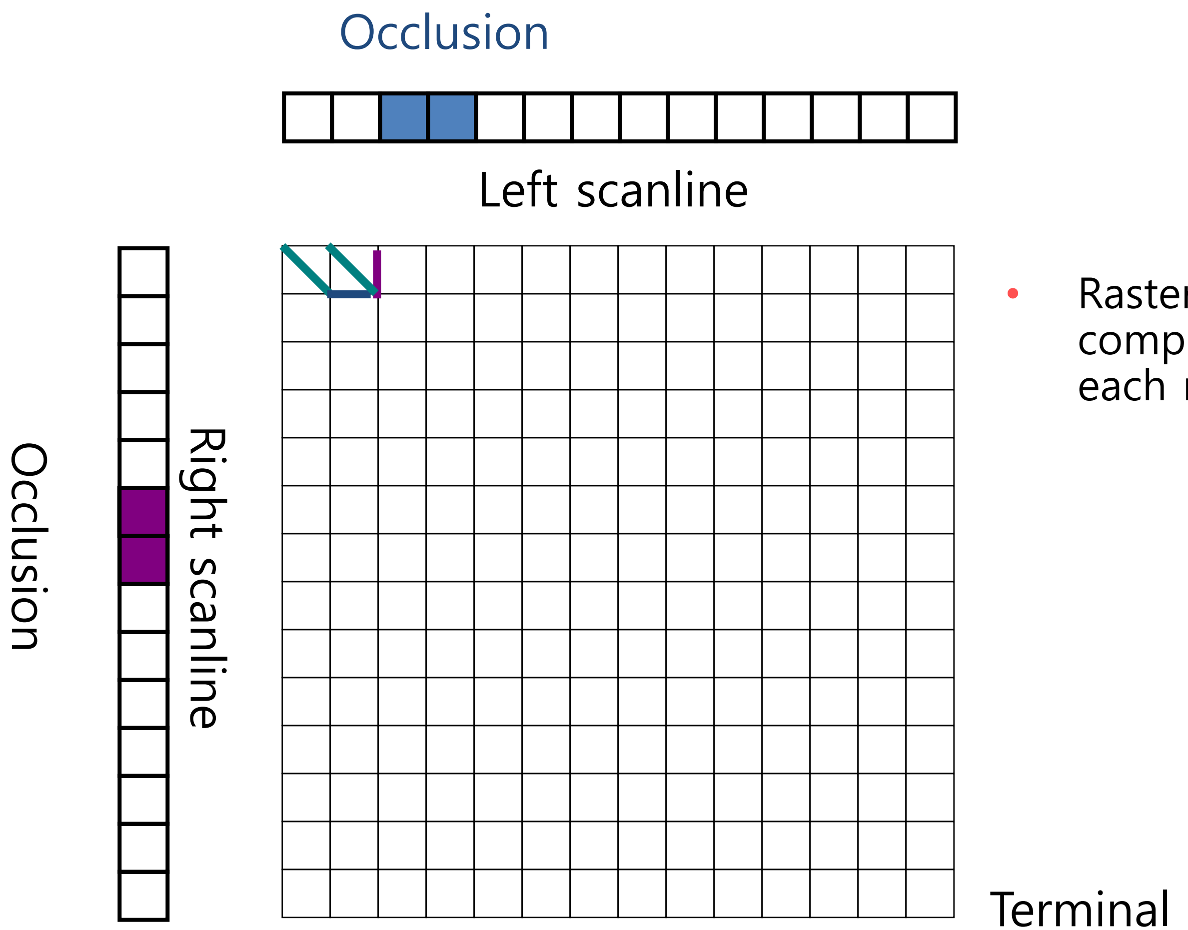


Cost function $C(i,j)$: the optimal cost up to node (i,j).

$$C(i,j) = \min\{\begin{array}{l} C(i-1,j-1) + \text{matching cost}, \\ C(i-1,j) + \text{left occlusion penalty}, \\ C(i,j-1) + \text{right occlusion penalty} \end{array}\}$$

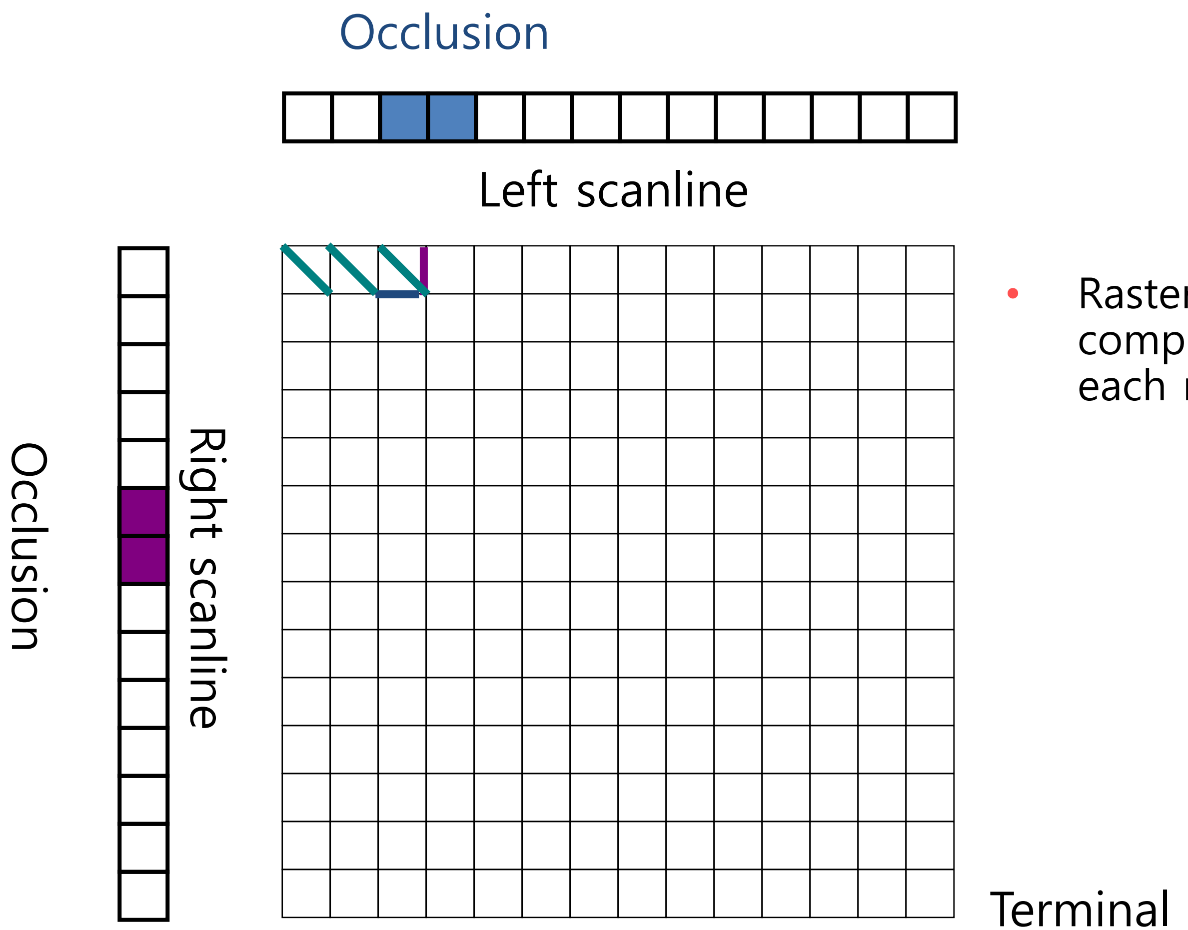
While computing the cost, we record how node (i,j) is connected to one of the three candidates

Dynamic Programming Approach



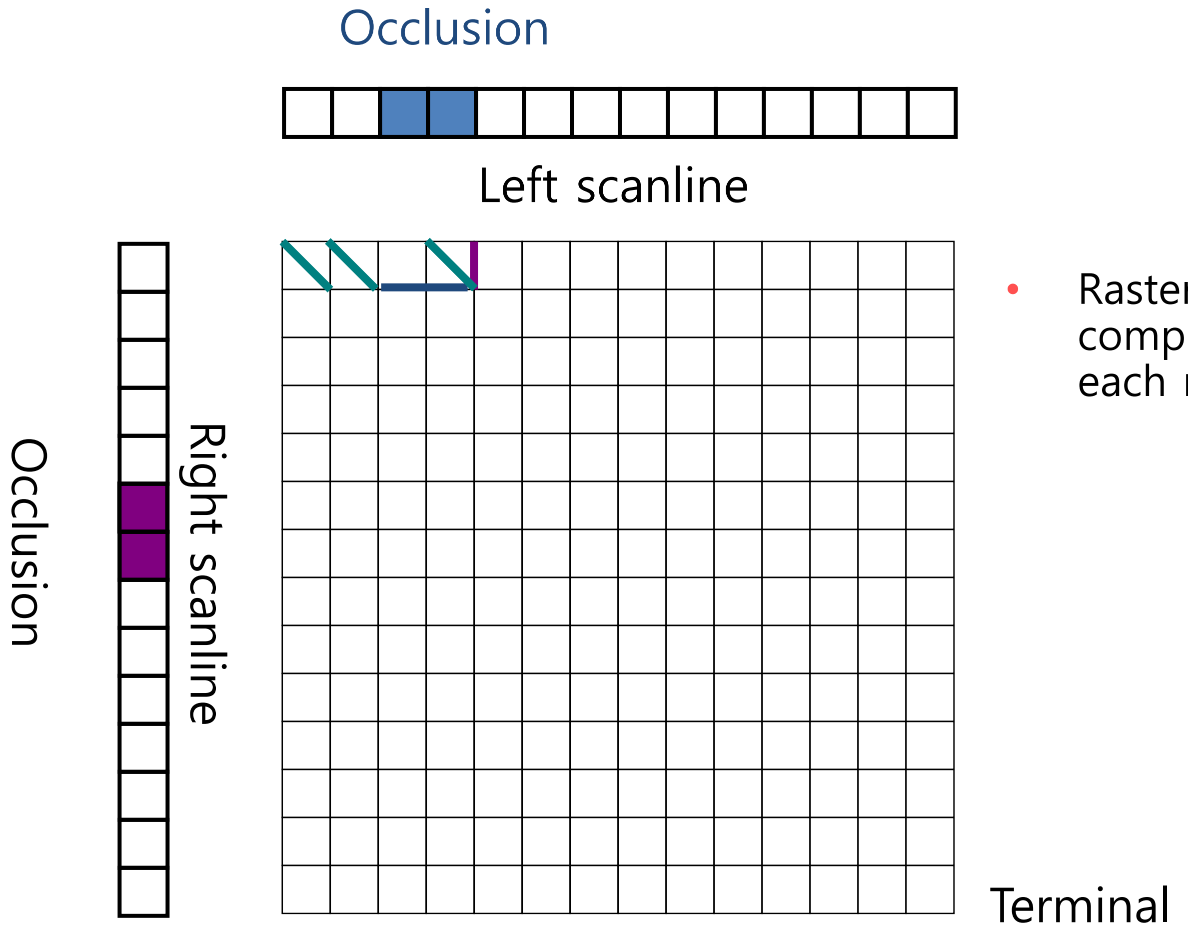
- Raster-scan the nodes, computing optimal cost for each node.

Dynamic Programming Approach



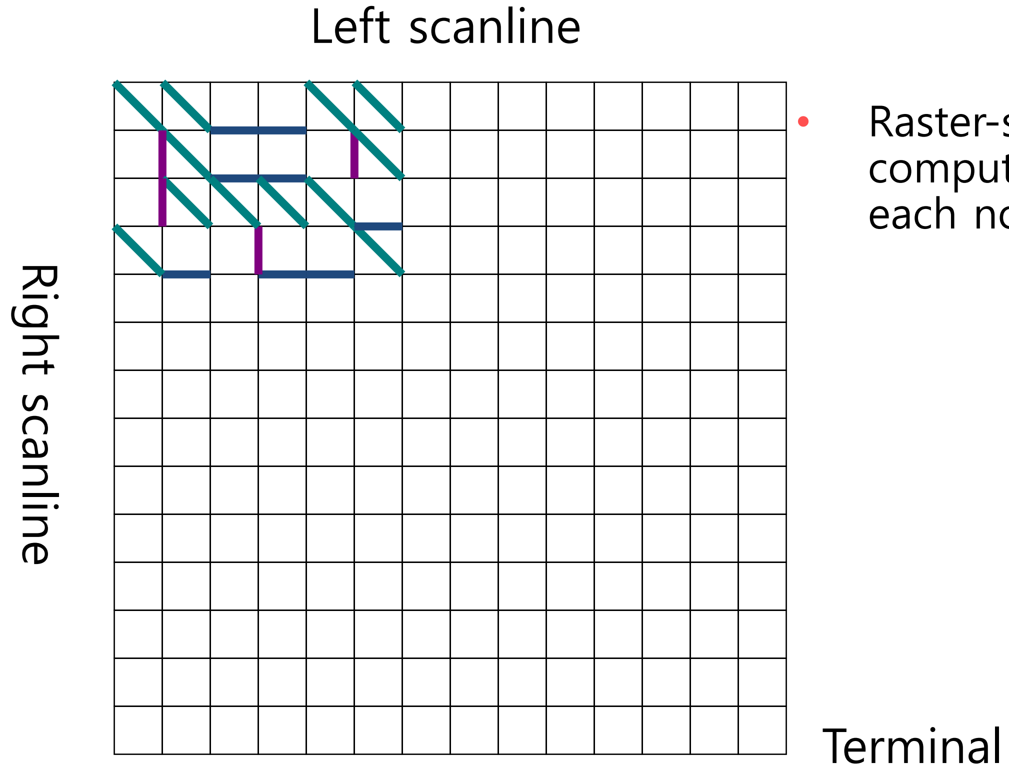
- Raster-scan the nodes, computing optimal cost for each node.

Dynamic Programming Approach



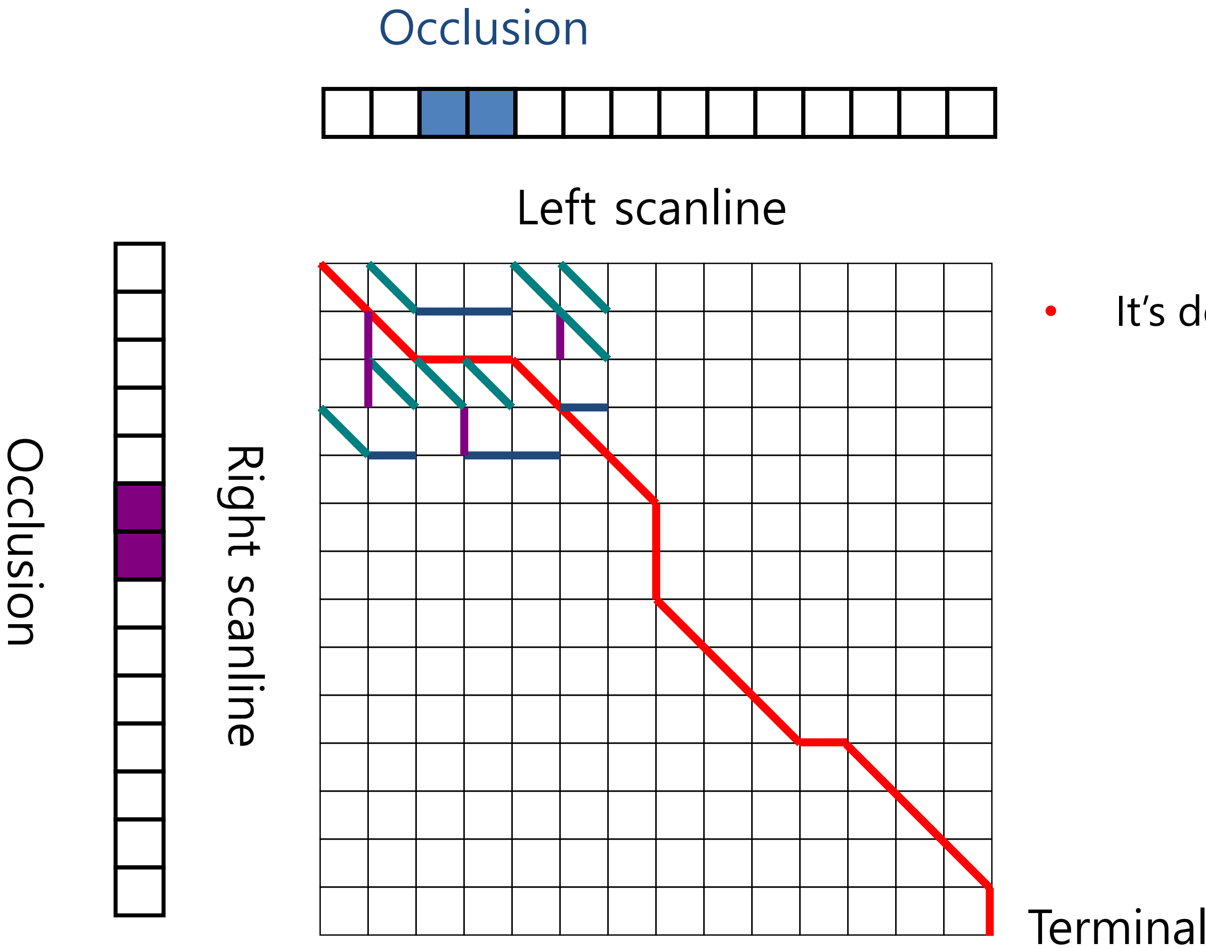
- Raster-scan the nodes, computing optimal cost for each node.

Dynamic Programming Approach



- Raster-scan the nodes, computing optimal cost for each node.

Dynamic Programming Approach



- It's done

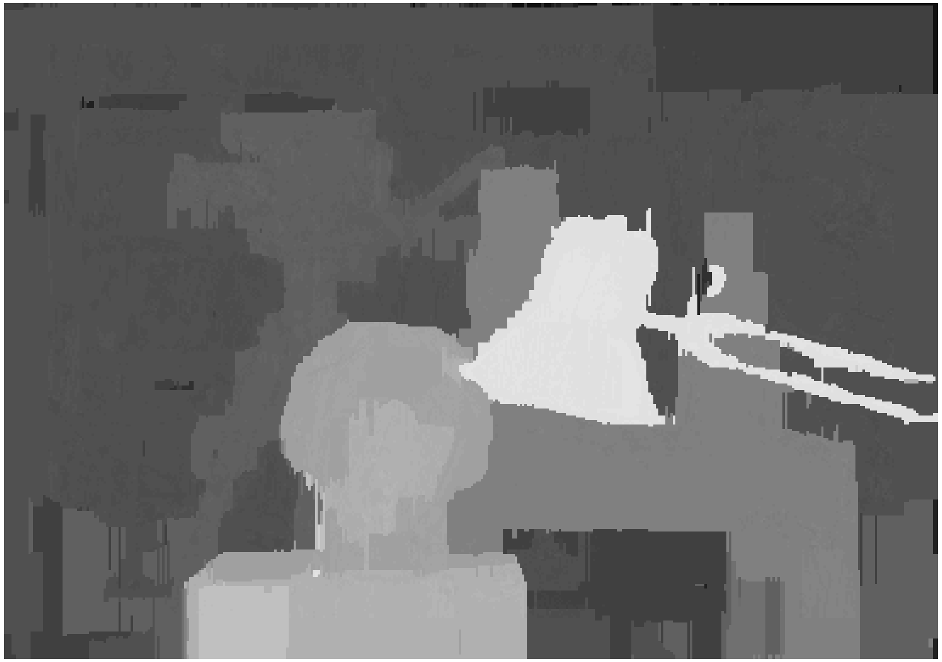
Dynamic Programming Approach

- It treats each scan line independently and thus may generate streaking artifacts
- An error can propagate

Streaking artifacts

Dynamic Programming Approach

- Enforcing inter-scanline continuity constraint
  - J.C. Kim, K.M. Lee, B.T. Choi, and S.U. Lee, “A dense stereo matching using two-pass dynamic programming with generalized ground control points” CVPR 2005
  - Y. Ohta and T. Kanade, “Stereo by Intra- and Inter-Scanline Search,” IEEE Trans. PAMI, 7(2):139-154 (1985).



Taxonomy and Categorization

- Four steps
  1. Matching cost computation
  2. Cost aggregation
  3. Disparity computation and optimization
  4. Disparity refinement

[Scharstein and Szeliski, 2002]

Four Steps: Example

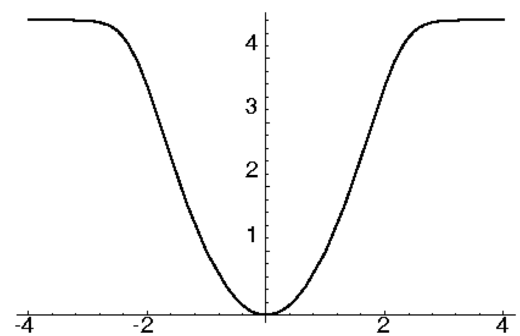
1. For every disparity, compute *raw* matching costs

$$E_0(x, y, d) = \rho(I_L(x + d, y) - I_R(x, y))$$

– $\rho(x) = x^2$

– $\rho(x) = |x|$

– Robust M-estimator $r(\cdot)$ $\Rightarrow$



- Why use a robust function?
- Occlusions, other outliers

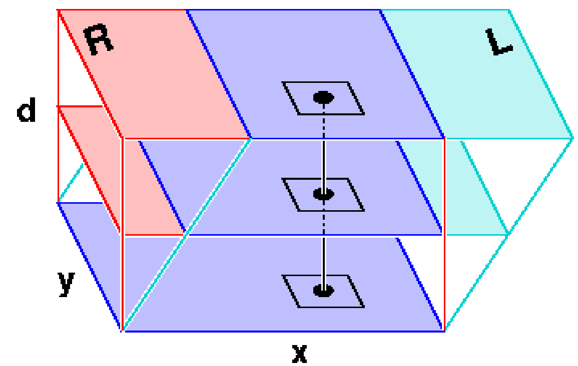
[Szeliski, Lecture Notes]

Four Steps: Example

2. Aggregate costs spatially

$$E(x, y; d) = \sum_{(x',y') \in N(x,y)} E_0(x', y', d)$$

- Here, we are using a *box filter* (efficient moving average implementation)
- Alternatively, weighted average, …



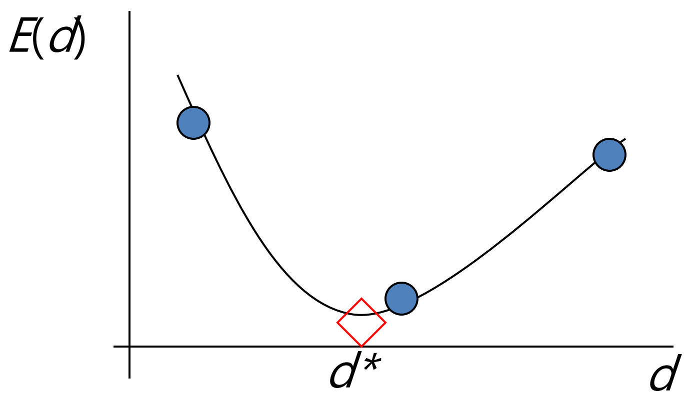
[Szeliski, Lecture Notes]

Four Steps: Example

3. Choose winning disparity at each pixel

$$d(x, y) = \arg\min_{d} E(x, y; d)$$

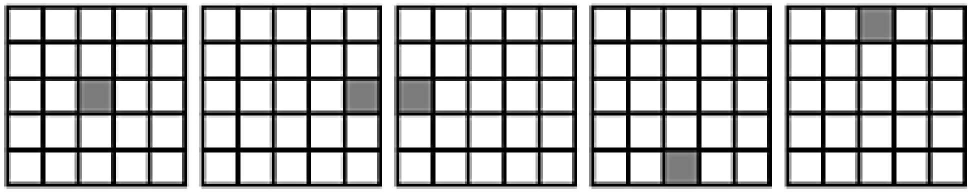
4. Interpolate to *sub-pixel* accuracy



[Szeliski, Lecture Notes]

Cost Aggregation

- Shiftable window



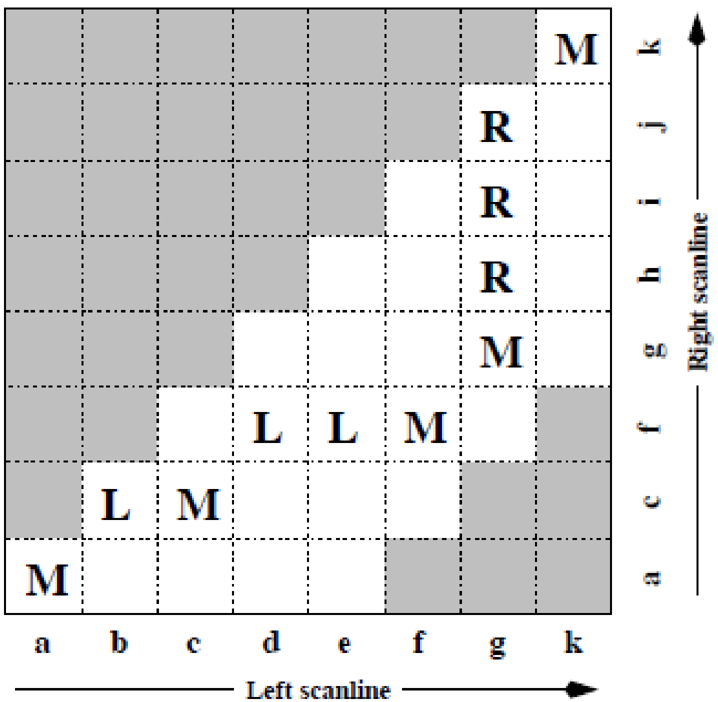
- Variable windows, adaptive weights, and segmentation-based

(a) (b) (c) (d)

[Szeliski, Lecture Notes]

Disparity Optimization

- Dynamic Programming
  - Scanline optimization
  - Evaluate best cumulative cost at each pixel



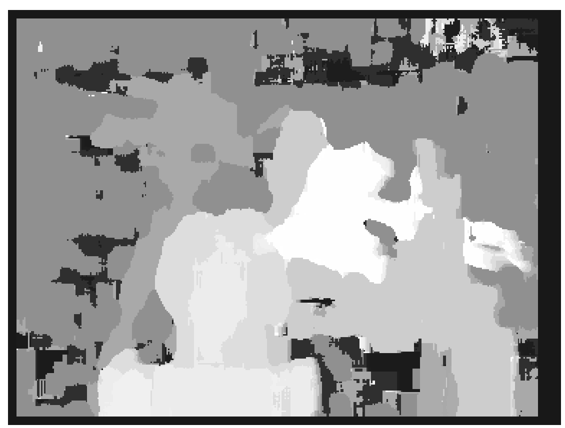
[Szeliski, Lecture Notes]

Disparity Optimization

- Cost function

$$E(\boldsymbol{d}) = E_{\text{data}}(\boldsymbol{d}) + \lambda \cdot E_{\text{smooth}}(\boldsymbol{d})$$

- Recent Trend
  - Belief propagation
  - Graph-cut

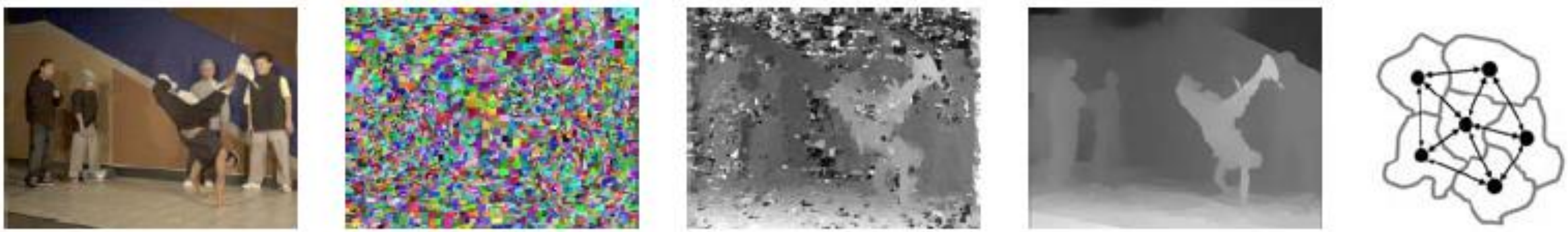


SAD WTA

Graph cut

[Szeliski, Lecture Notes]

Segmentation-Based Stereo Matching



(a) (b) (c) (d) (e)

Figure 11.12 Segmentation-based stereo matching (Zitnick, Kang, Uyttendaele *et al.* 2004) © 2004 ACM: (a) input color image; (b) color-based segmentation; (c) initial disparity estimates; (d) final piecewise-smoothed disparities; (e) MRF neighborhood defined over the segments in the disparity space distribution (Zitnick and Kang 2007) © 2007 Springer.

Middlebury Evaluation

- http://vision.middlebury.edu/

vision.middlebury.edu

stereo • mview • MRF • flow • color

Stereo Evaluation • Datasets • Code • Submit

Daniel Scharstein • Richard Szeliski

Welcome to the Middlebury Stereo Vision Page, formerly located at www.middlebury.edu/stereo. 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.

Middlebury Evaluation

Error Threshold = 1

Error Threshold...

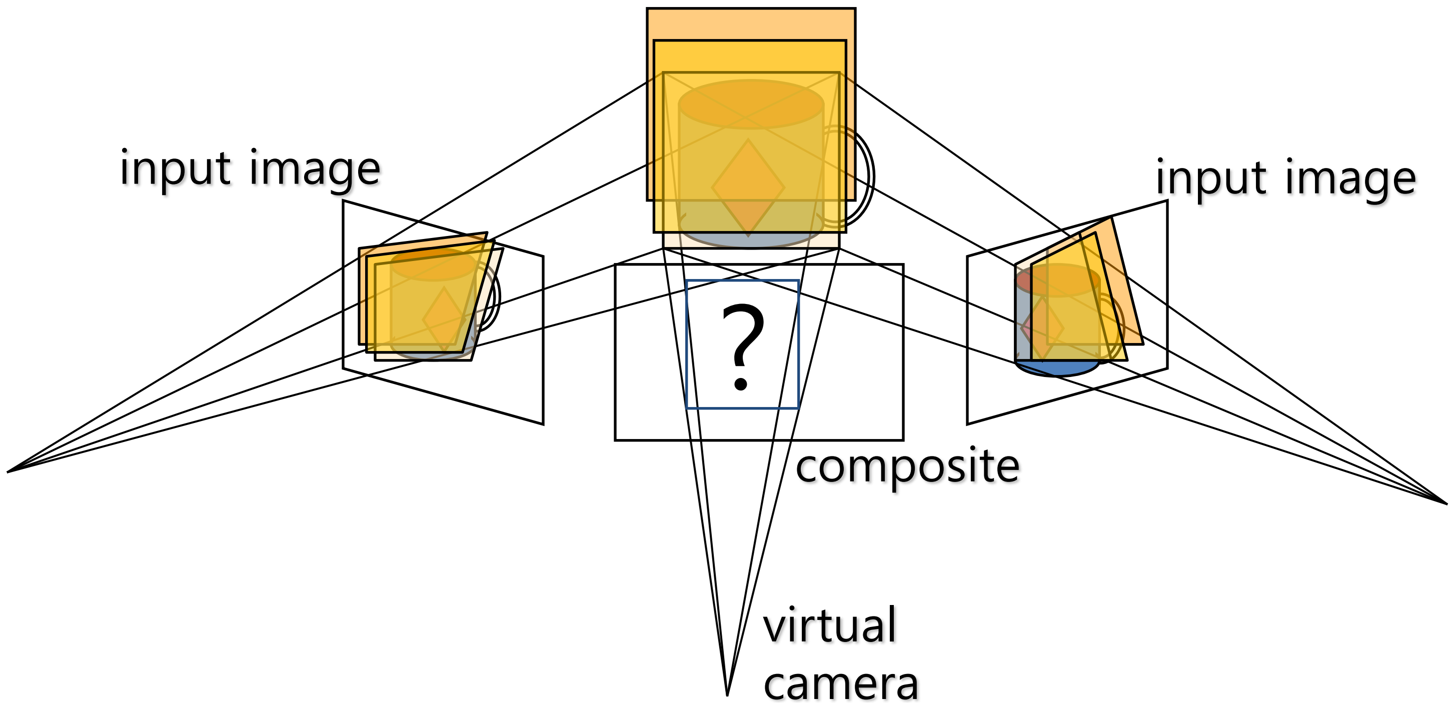
Algorithm	Avg. Rank	Tsukuba (ground truth) nonocc	Tsukuba all	Tsukuba disc	Venus (ground truth) nonocc	Venus all	Venus disc	Teddy (ground truth) nonocc	Teddy all	Teddy disc	Cones (ground truth) nonocc	Cones all	Cones disc	Average percent of bad pixels (explanation)
		Sort by nonocc			Sort by all						Sort by disc			
AdaptGCP [137]	7,5	1,03 13	1,29 6	5,60 16	0,10 3	**0,14** 1	1,30 5	4,63 17	6,47 8	12,5 18	**1,81** 1	**5,70** 1	**5,33** 1	3,83
ADCensus [94]	11,6	1,07 19	1,48 15	5,73 22	0,09 2	0,25 10	1,15 3	4,10 11	6,22 5	10,9 10	2,42 14	7,25 13	6,95 15	3,97
AdaptingBP [17]	14,5	1,11 23	1,37 9	5,79 24	0,10 4	0,21 6	1,44 7	4,22 13	7,06 12	11,8 14	2,48 17	7,92 22	7,32 23	4,23
CoopRegion [41]	14,9	0,87 4	**1,16** 1	4,61 4	0,11 5	0,21 5	1,54 9	5,16 27	8,31 17	13,0 23	2,79 32	7,18 12	8,01 40	4,41
RVbased [116]	18,7	0,95 9	1,42 13	4,98 9	0,11 7	0,29 15	**1,07** 1	5,98 37	11,6 49	15,4 46	2,35 11	7,61 14	6,81 13	4,88
RDP [102]	19,3	0,97 10	1,39 11	5,00 10	0,21 33	0,38 24	1,89 19	4,84 19	9,94 28	12,6 19	2,53 19	7,69 16	7,38 24	4,57
DoubleBP [35]	19,5	0,88 6	1,29 5	4,76 7	0,13 10	0,45 37	1,87 18	3,53 8	8,30 16	9,63 6	2,90 39	8,78 50	7,79 32	4,19
MultiRBF [153]	20,4	1,33 44	1,56 18	6,02 31	0,13 9	0,17 3	1,84 16	5,09 25	6,36 6	13,4 27	2,90 40	6,76 8	7,10 18	4,39
OutlierConf [42]	20,8	0,88 5	1,43 14	4,74 6	0,18 22	0,26 13	2,40 34	5,01 22	9,12 23	12,8 22	2,78 31	8,57 41	6,99 16	4,60
AdaptiveGF [151]	24,1	1,04 15	1,53 16	5,62 17	0,17 21	0,41 29	1,98 22	5,71 34	11,3 41	14,3 32	2,44 15	8,22 30	7,05 17	4,98
SubPixSearch [127]	26,2	2,04 87	2,48 76	6,40 41	0,14 14	0,40 28	1,74 13	4,00 10	6,39 7	11,0 12	2,24 8	6,87 10	6,50 8	4,18
SubPixDoubleBP [30]	26,5	1,24 32	1,76 38	5,98 30	0,12 8	0,46 39	1,74 13	3,45 7	8,38 18	10,0 8	2,93 43	8,73 47	7,91 35	4,39
SurfaceStereo [79]	26,5	1,28 39	1,65 26	6,78 49	0,19 24	0,28 14	2,61 47	3,12 4	**5,10** 1	8,65 2	2,89 38	7,95 25	8,26 49	4,06
LLR [135]	28,0	1,05 16	1,65 25	5,64 18	0,29 55	0,81 71	3,07 56	4,56 15	9,81 27	12,2 15	2,17 5	8,02 27	6,42 6	4,64
WarpMat [55]	30,8	1,16 24	1,35 8	6,04 32	0,18 23	0,24 9	2,44 38	5,02 23	9,30 24	13,0 25	3,49 60	8,47 39	9,01 65	4,98
ObjectStereo [98]	31,9	1,22 31	1,62 21	6,36 38	0,59 85	0,69 63	4,61 86	4,13 12	7,59 13	11,2 13	2,20 6	6,99 11	6,36 4	4,46
PMF [138]	34,4	1,74 70	2,04 59	8,07 79	0,33 61	0,49 44	4,16 79	**2,52** 1	5,87 4	**8,30** 1	2,13 3	6,80 9	6,32 3	4,06
CSM [120]	46,2	**0,82** 1	1,20 2	4,39 2	0,34 62	0,61 53	2,55 43	7,67 85	12,4 71	17,2 75	3,33 57	9,35 67	7,96 37	5,65

ETC

- Plane sweep stereo
- Multi-view stereo

Plane Sweep Stereo

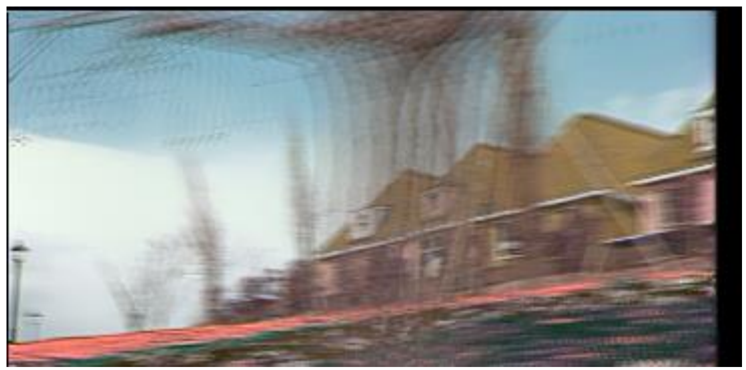
- Sweep family of planes through volume



[Szeliski, Lecture Notes]

Plane Sweep Stereo

- For each depth plane
  - compute composite (mosaic) image — *mean*



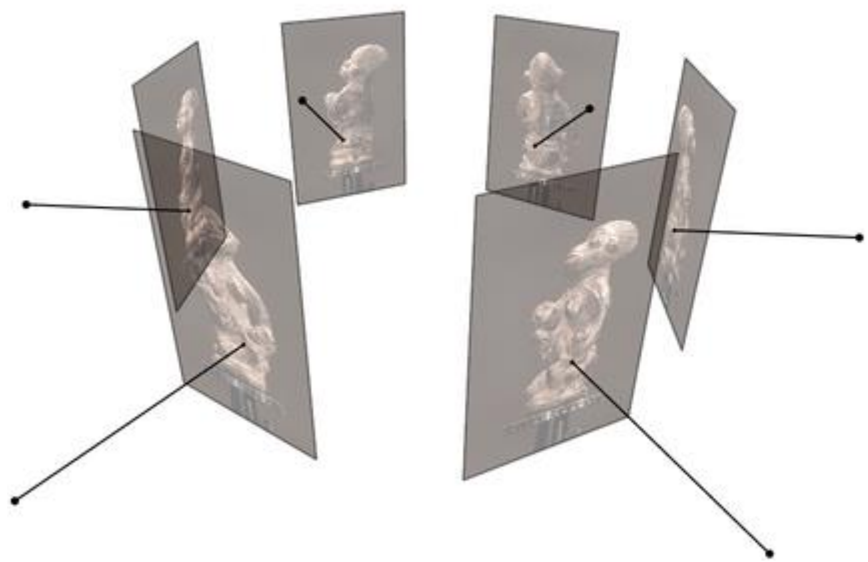
  - compute error image — *variance*
  - convert to confidence and aggregate spatially
- Select winning depth at each pixel

[Szeliski, Lecture Notes]

Multi-view Stereo

Input: calibrated images from several viewpoints

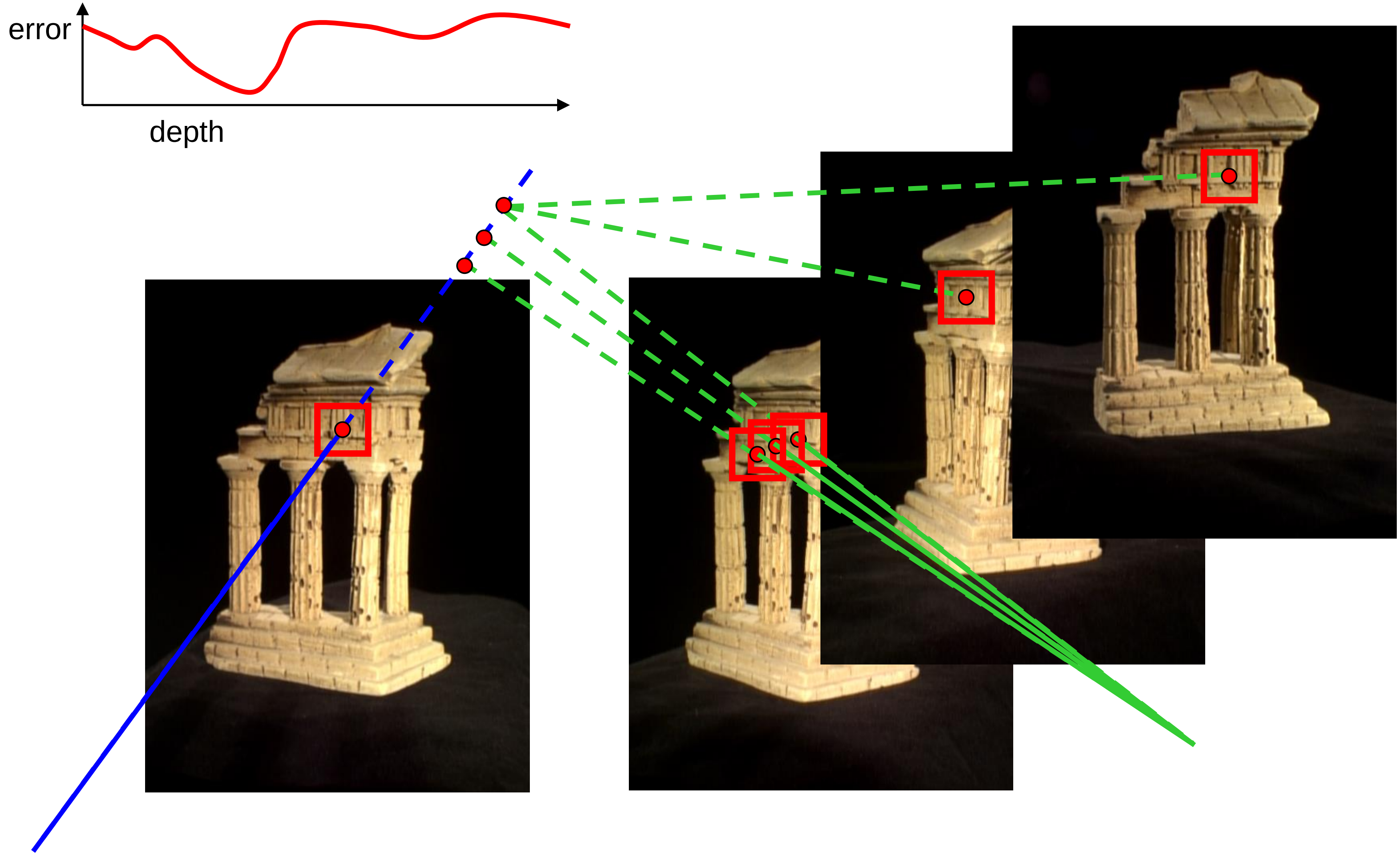
Output: 3D object model



Figures by Carlos Hernandez

[Seitz, Lecture Notes]

Multi-view Stereo

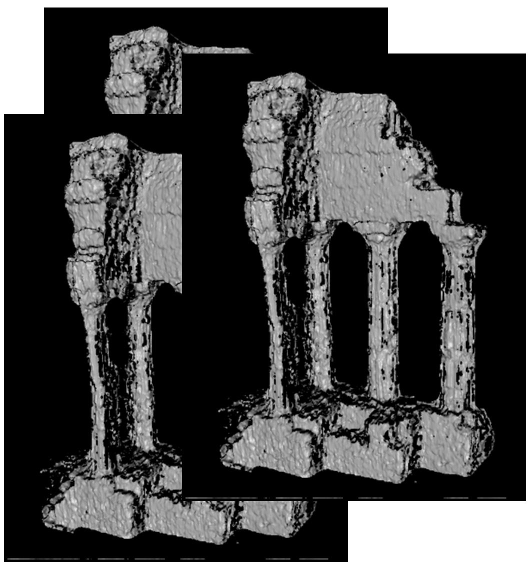


[Seitz, Lecture Notes]

Merging Depth Maps

[Curless and Levoy 1996]

– compute weighted average of depth maps



set of depth maps
(one per view)

merged surface
mesh

[Seitz, Lecture Notes]

Merging Depth Maps

input image

317 images
(hemisphere)

ground truth model

[Goesele, Curless, Seitz, 2006]

[Seitz, Lecture Notes]

Example I

CONSISTENT STEREO MATCHING

I-L. Jung, T.-Y. Chung, J.-Y. Sim, and C.-S. Kim, “Consistent stereo matching under varying radiometric conditions,” IEEE Trans. Multimedia, vol. 15, pp. 56-69, Jan. 2013.

Pseudo-Disparity Estimation

- Failures of color consistency assumption
  - Corresponding pixels may have different colors
  - Colors are affected by various illumination conditions

Different exposure conditions

Pseudo-Disparity Estimation

- Idea
  - Histogram = probability distribution of pixel values in an image
  - Cumulative histogram values = the ranks of pixel brightness
  - Corresponding pixels indicate the same scene point
    - Their colors can be different
    - But their ranks in each image should be almost the same

Pseudo-Disparity Estimation

- Joint CDF maps
  - K_0 : The joint CDF for the left view
  - K_1 : The joint CDF for the right view


Adaptive Color Transform

- Affine Color Mapping

$$\gamma_1(\boldsymbol{p} - \widetilde{\boldsymbol{d}}_{\boldsymbol{p}}) = \psi\gamma_0(\boldsymbol{p}) + \zeta\mathbf{1}$$

- Parameter Estimation
  - Least squares

$$\left\| W \times \left(\begin{bmatrix} \gamma_1(\mathbf{p} - \tilde{\mathbf{d}}_{\mathbf{p}}) \\ \gamma_1(\mathbf{p}_1 - \tilde{\mathbf{d}}_{\mathbf{p}_1}) \\ \vdots \\ \gamma_1(\mathbf{p}_N - \tilde{\mathbf{d}}_{\mathbf{p}_N}) \end{bmatrix} - \begin{bmatrix} \gamma_0(\mathbf{p}) & \mathbf{1} \\ \gamma_0(\mathbf{p}_1) & \mathbf{1} \\ \vdots & \vdots \\ \gamma_0(\mathbf{p}_N) & \mathbf{1} \end{bmatrix} \begin{bmatrix} \psi \\ \zeta \end{bmatrix} \right) \right\|^2$$

Color Transform Results

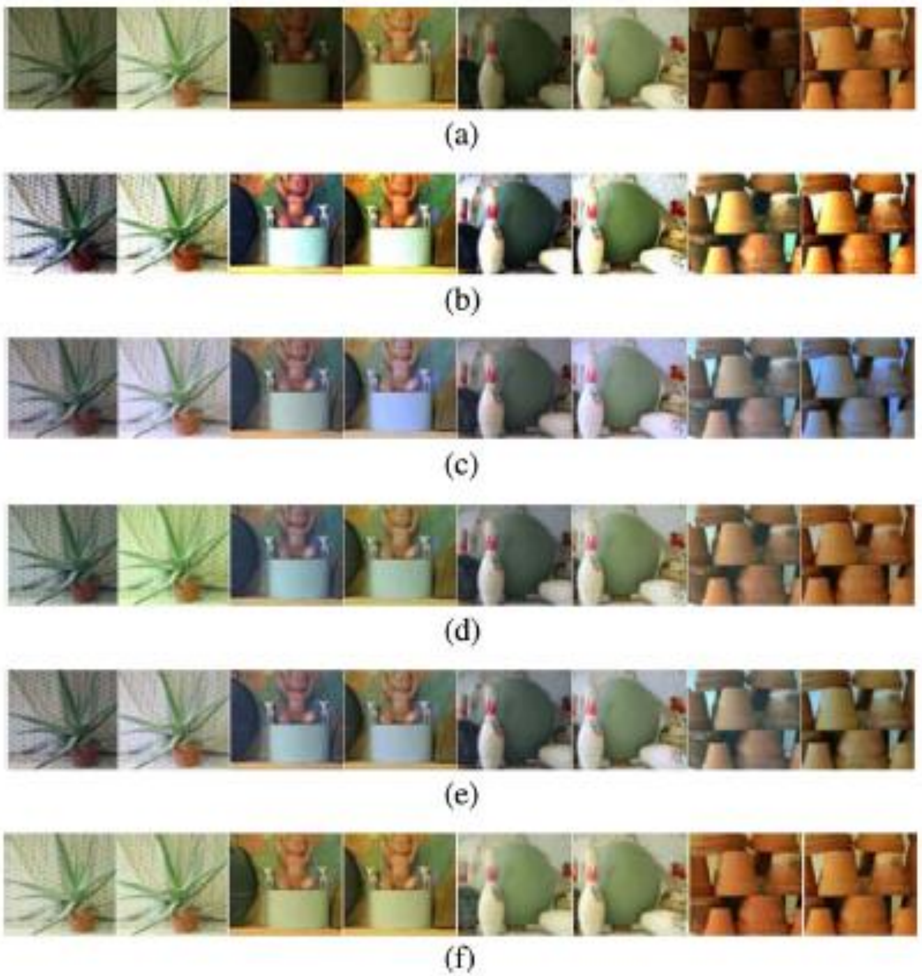


Fig. 8. Comparison of the color transform algorithms on the "Aloe," "Baby3," "Bowling2," and "Flowerpots" datasets with the same lighting condition 1 but with different exposure conditions (1 for left images and 2 for right images). (a) Original images; (b) gray world assumption; (c) comprehensive color normalization; (d) grey-edge; (e) shade of gray; (f) proposed algorithm.

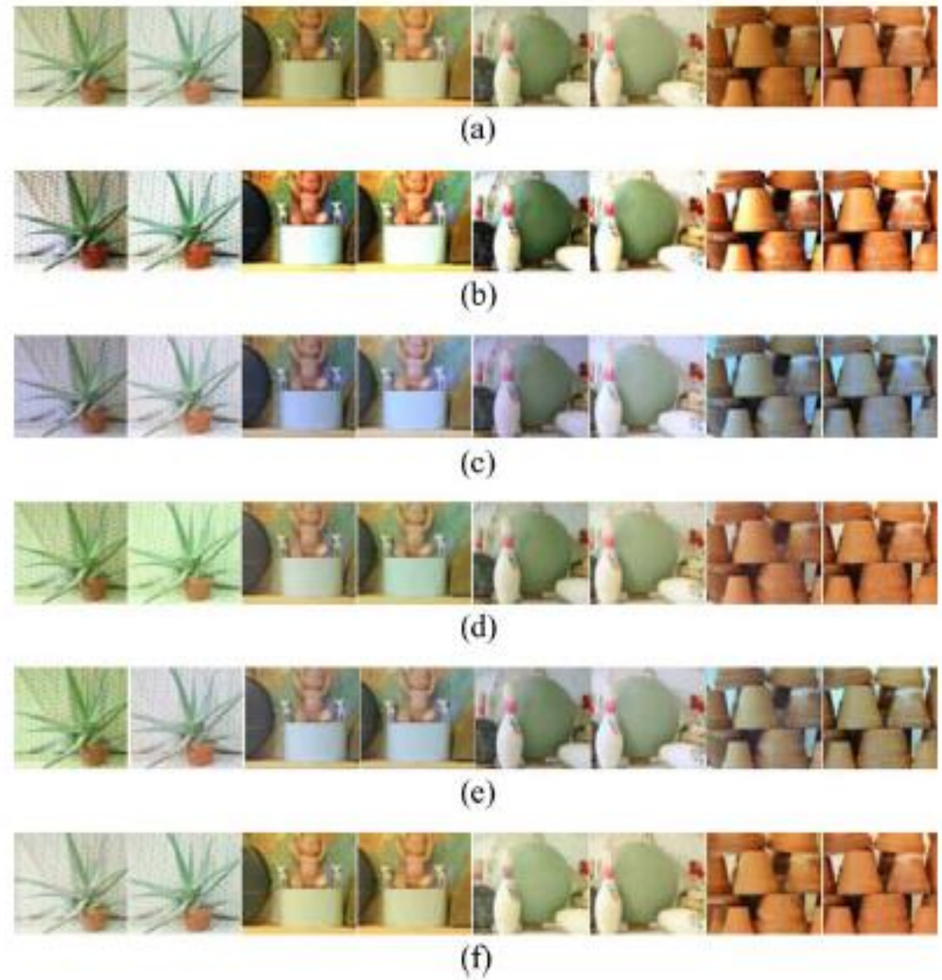
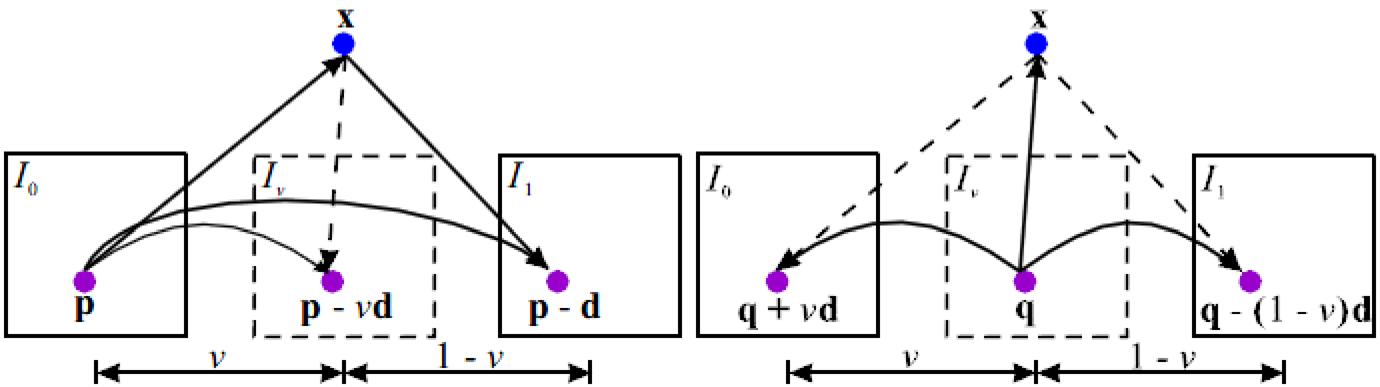


Fig. 9. Comparison of the color transform algorithms on the "Aloe," "Baby3," "Bowling2," and "Flowerpots" datasets with the same exposure condition 2 but with different lighting conditions (1 for left images and 2 for right images). (a) Original images; (b) gray world assumption; (c) comprehensive color normalization; (d) grey-edge; (e) shade of gray; (f) proposed algorithm.

Consistent Stereo Matching

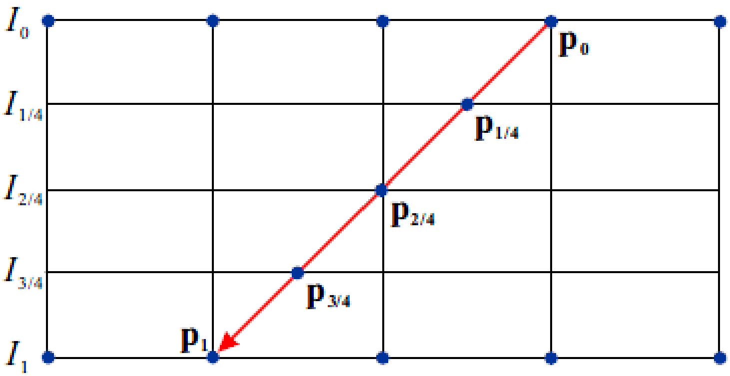
- Forward vs. inverse mappings



(a) (b)

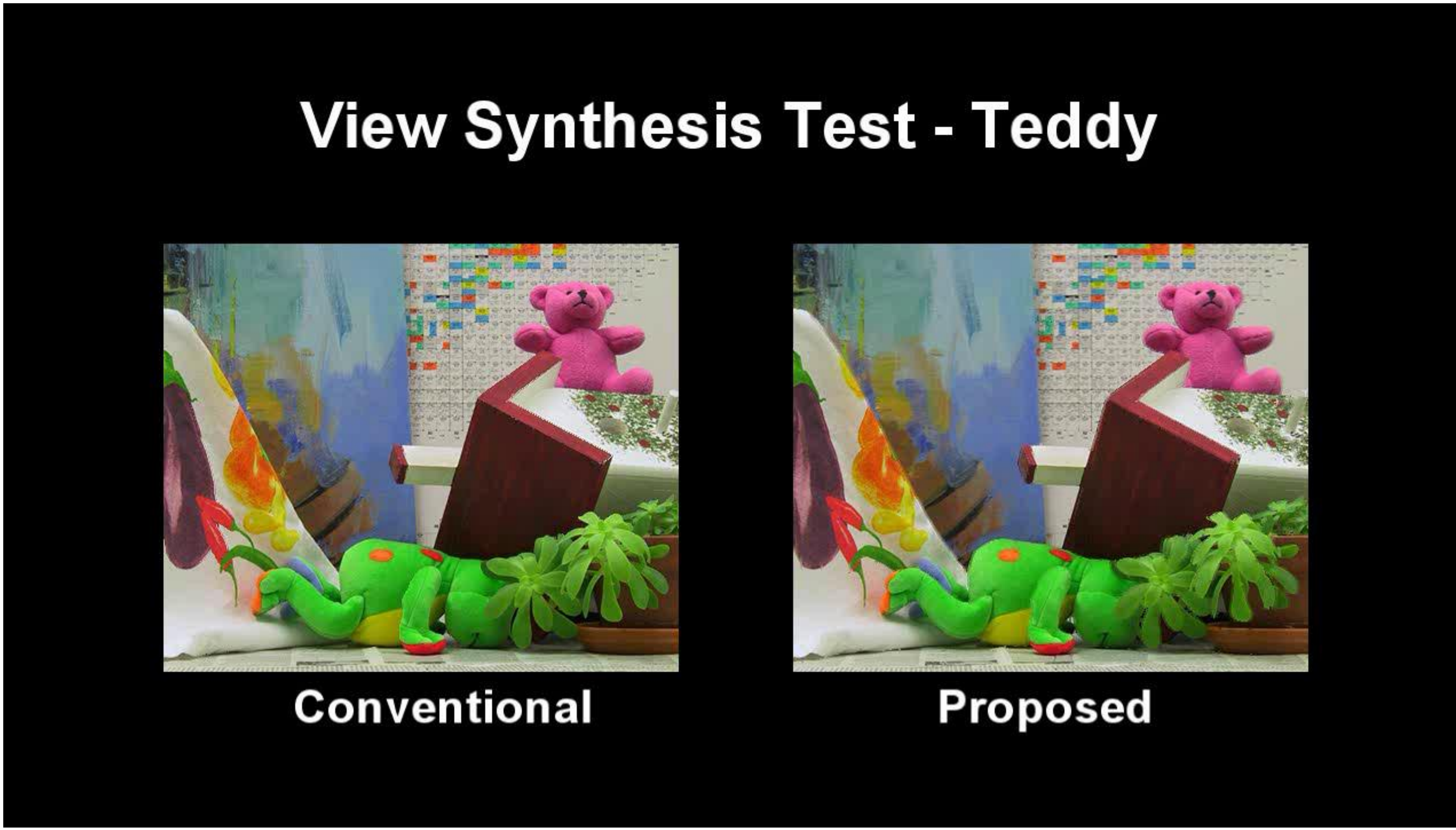
Consistent Stereo Matching

- Reliability term for matching cost computations



$$\begin{aligned} \mathrm{d}^* &= \arg\min_{\mathrm{d}} c_0(\mathrm{p}_0, \mathrm{d}) = \arg\min_{\mathrm{d}} c_{\frac{1}{4}}(\mathrm{p}_{\frac{1}{4}}, \mathrm{d}) \\ &= \arg\min_{\mathrm{d}} c_{\frac{2}{4}}(\mathrm{p}_{\frac{2}{4}}, \mathrm{d}) = \arg\min_{\mathrm{d}} c_{\frac{3}{4}}(\mathrm{p}_{\frac{3}{4}}, \mathrm{d}) \\ &= \arg\min_{\mathrm{d}} c_1(\mathrm{p}_1, \mathrm{d}). \end{aligned}$$

View Synthesis Results



Conclusions

- **Rank-based pseudo-disparity estimation** for color matching
- **Consistency Criterion**
- Especially good for view synthesis applications
- Computationally complicated