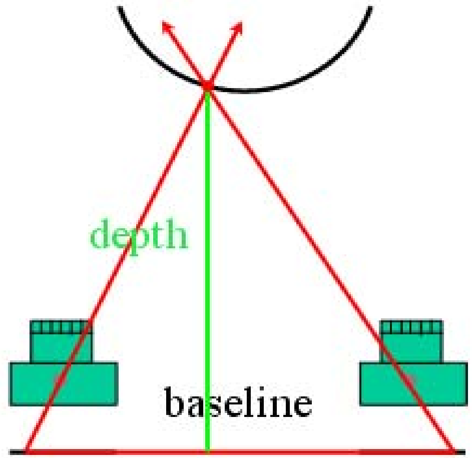
Stereo Matching

한양대학교
HANYANG UNIVERSITY

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Vision [1]



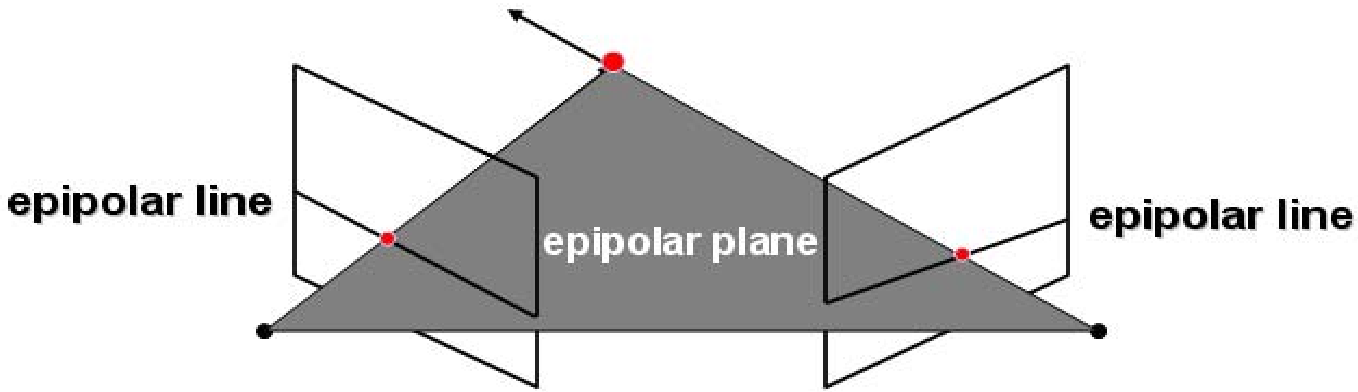
Triangulate on two images of the same point to recover depth.

- Feature matching across views
- Calibrated cameras

Matching correlation windows across scan lines

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Reduction of Searching by Epipolar Constraint [1]



- Epipolar Constraint
  - Matching points lie along corresponding epipolar lines
  - Reduces correspondence problem to 1D search along *conjugate epipolar lines*
  - Greatly reduces cost and ambiguity of matching

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

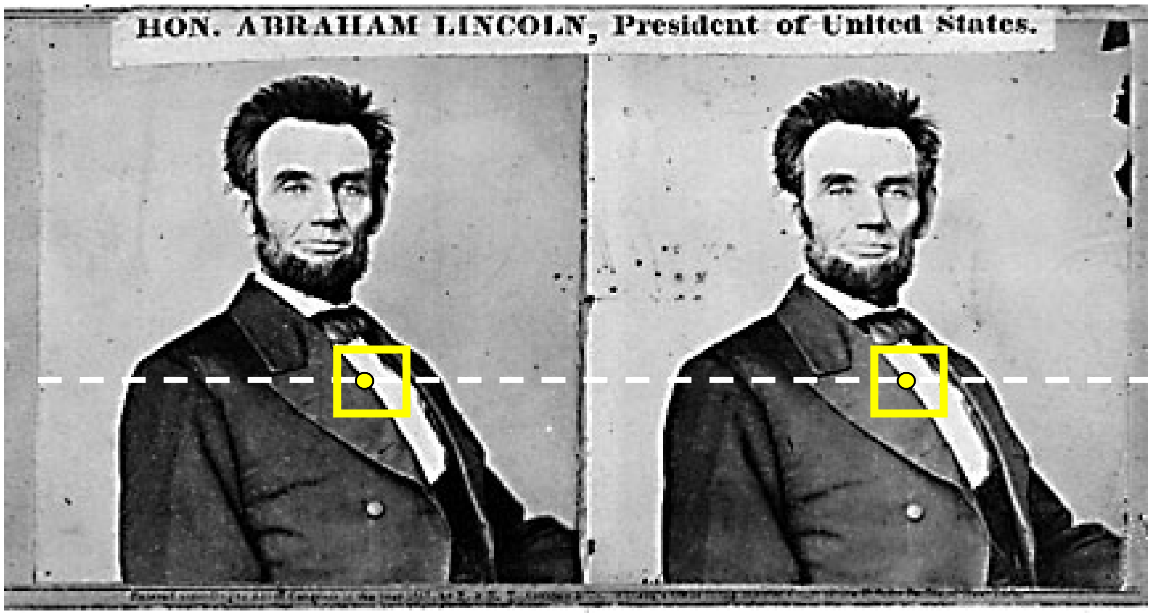
Photometric Constraint [1]

Same world point has same intensity in both images.

- True for Lambertian surfaces
  - A Lambertian surface has a brightness that is independent of viewing angle
- Violations:
  - Noise
  - Specularity
  - Non-Lambertian materials
  - Pixels that contain multiple surfaces

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Photometric Constraint [1]



For each epipolar line

For each pixel in the left image

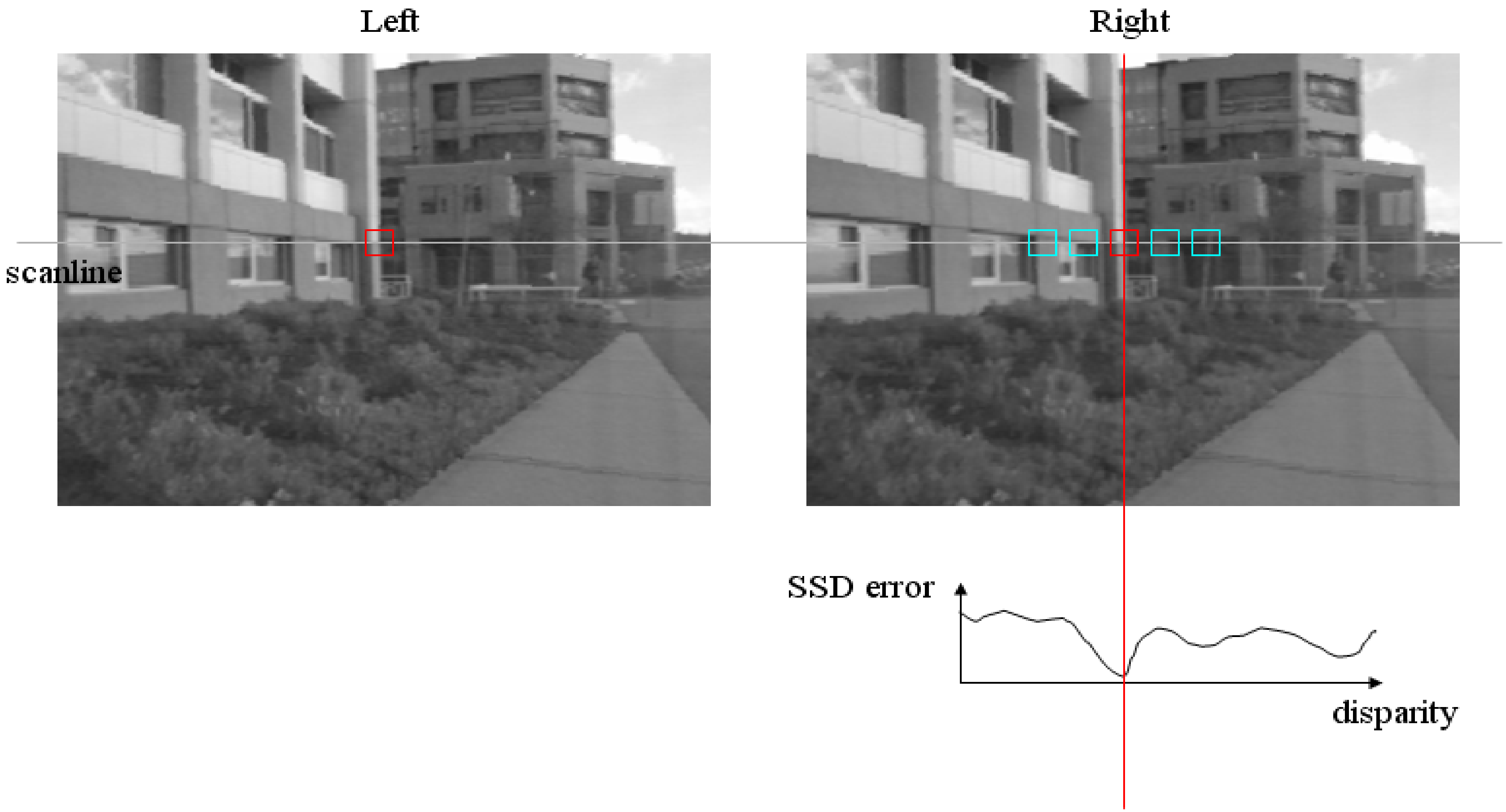
- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

This leaves too much ambiguity, so:

Improvement: match windows

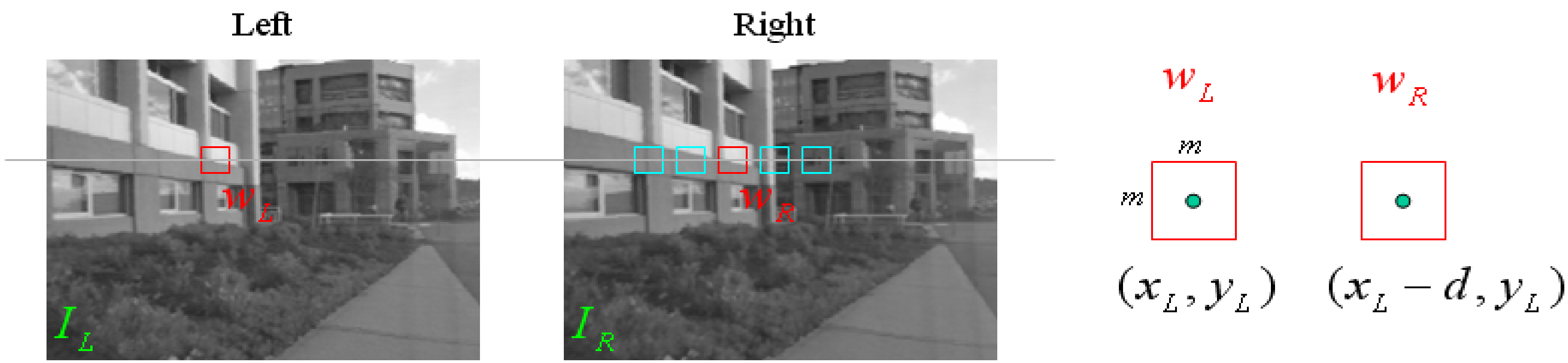
E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Correspondence Using Correlation [1]



E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Sum of Squared Difference (SSD) [1]



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 :

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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Image Normalization [1]

- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- For these reason and more, it is a good idea to normalize the pixels in each window:

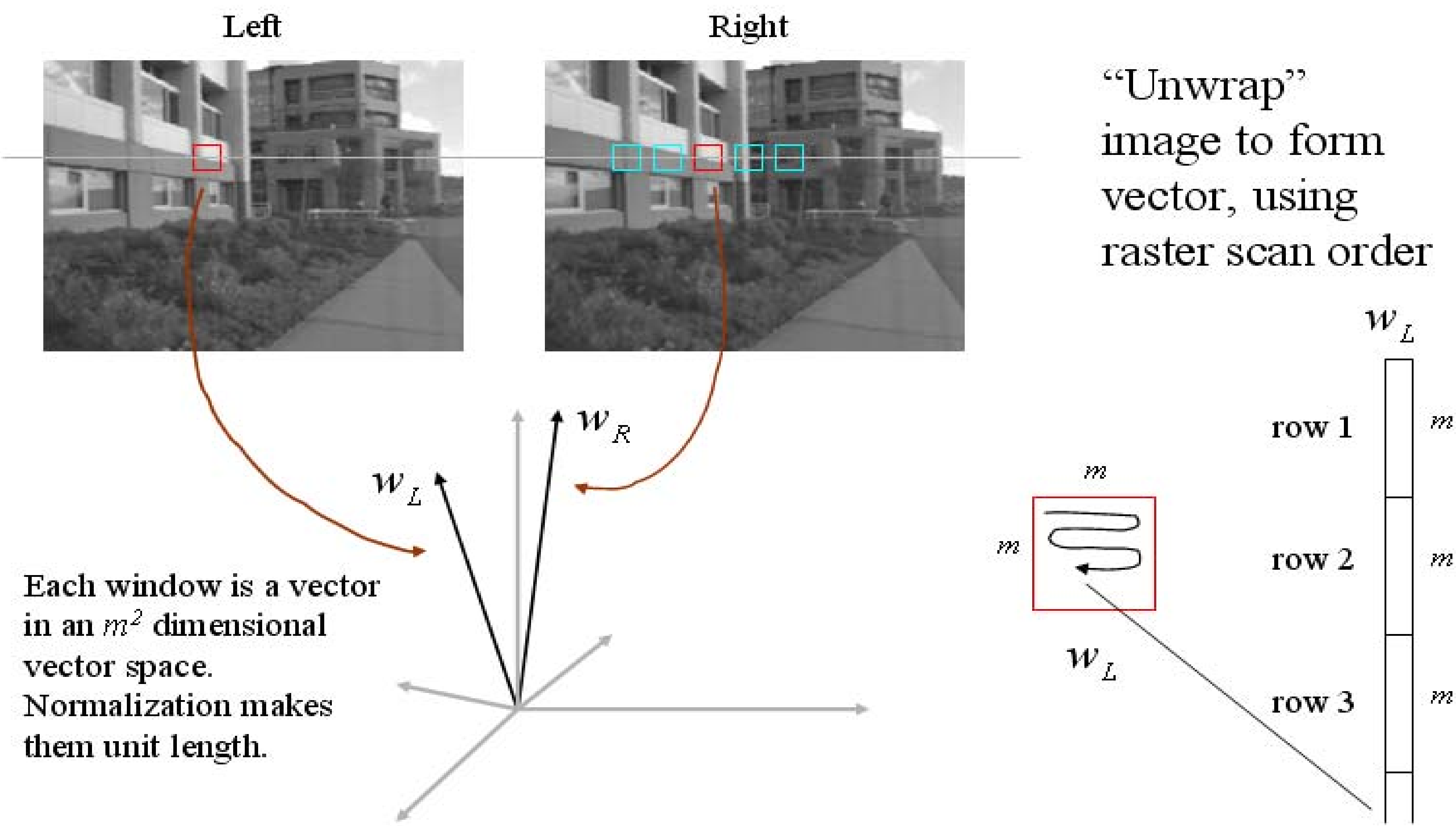
$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v)$$ Average pixel

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

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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Images as Vectors [1]

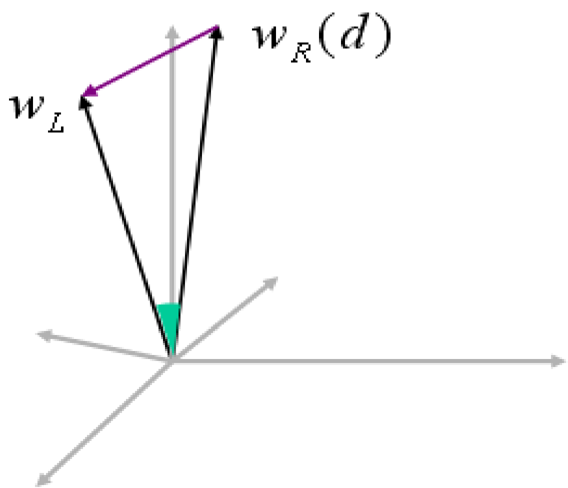


“Unwrap” image to form vector, using raster scan order

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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Image Metrics [1]



(Normalized) Sum of Squared Differences

$$C_{\text{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

$$= \left\| w_L - w_R(d) \right\|^2$$

Normalized Correlation

$$C_{\text{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

$$= w_L \cdot w_R(d) = \cos\theta$$

$$d^* = \arg\min_d \left\| w_L - w_R(d) \right\|^2 = \arg\max_d w_L \cdot w_R(d)$$

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Result [1]

Left

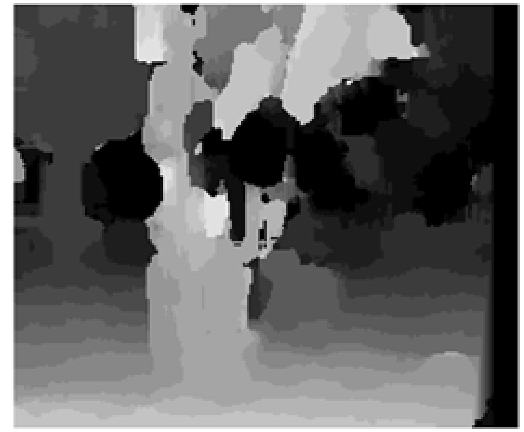
Disparity Map

Images courtesy of Point Grey Research

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Window Size [1]

W = 3



W = 20

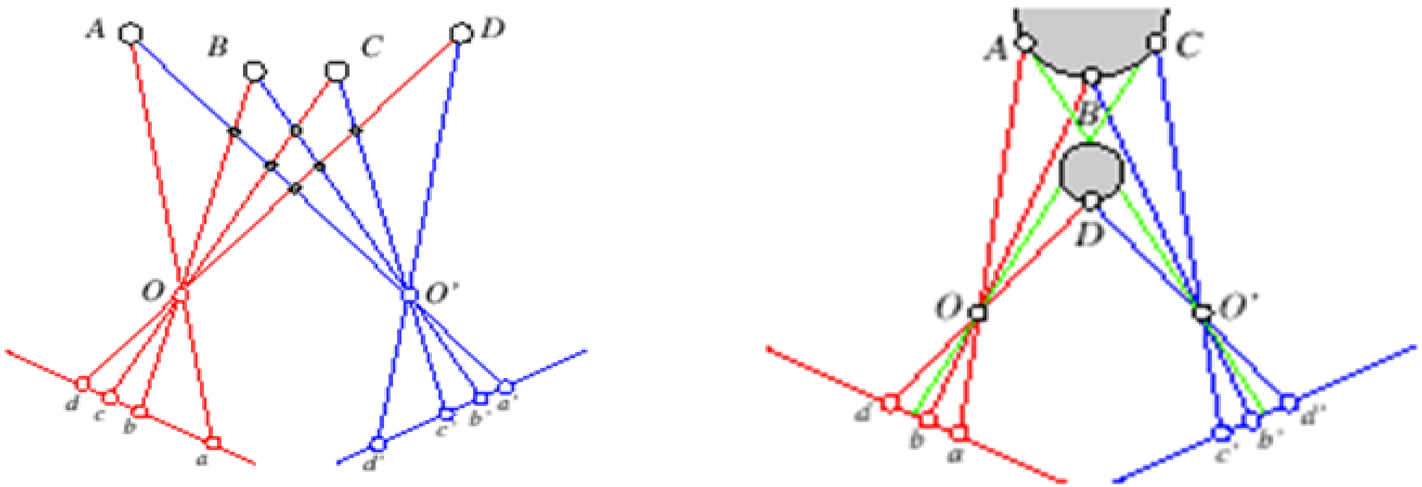
- Effect of window size
- Some approaches have been developed to use an adaptive window size (try multiple sizes and select best match)

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.

Ordering Constraint [3]

- If an object a is left on an object b in the left image then object a will also appear to the left of object b in the right image

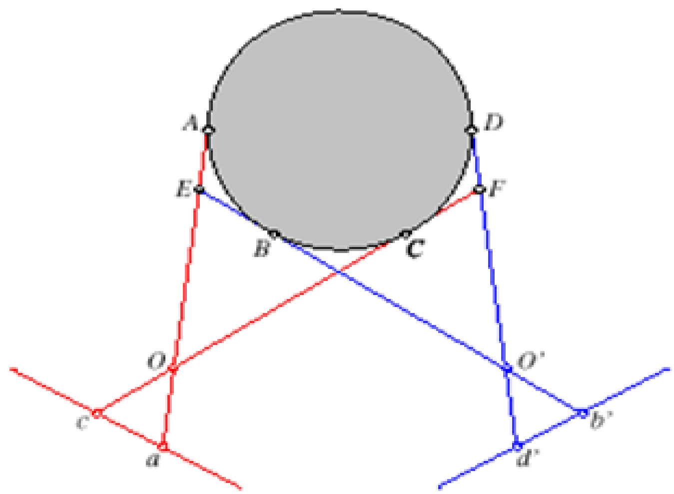


Ordering constraint… …and its failure

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Smooth Surface Problem [3]

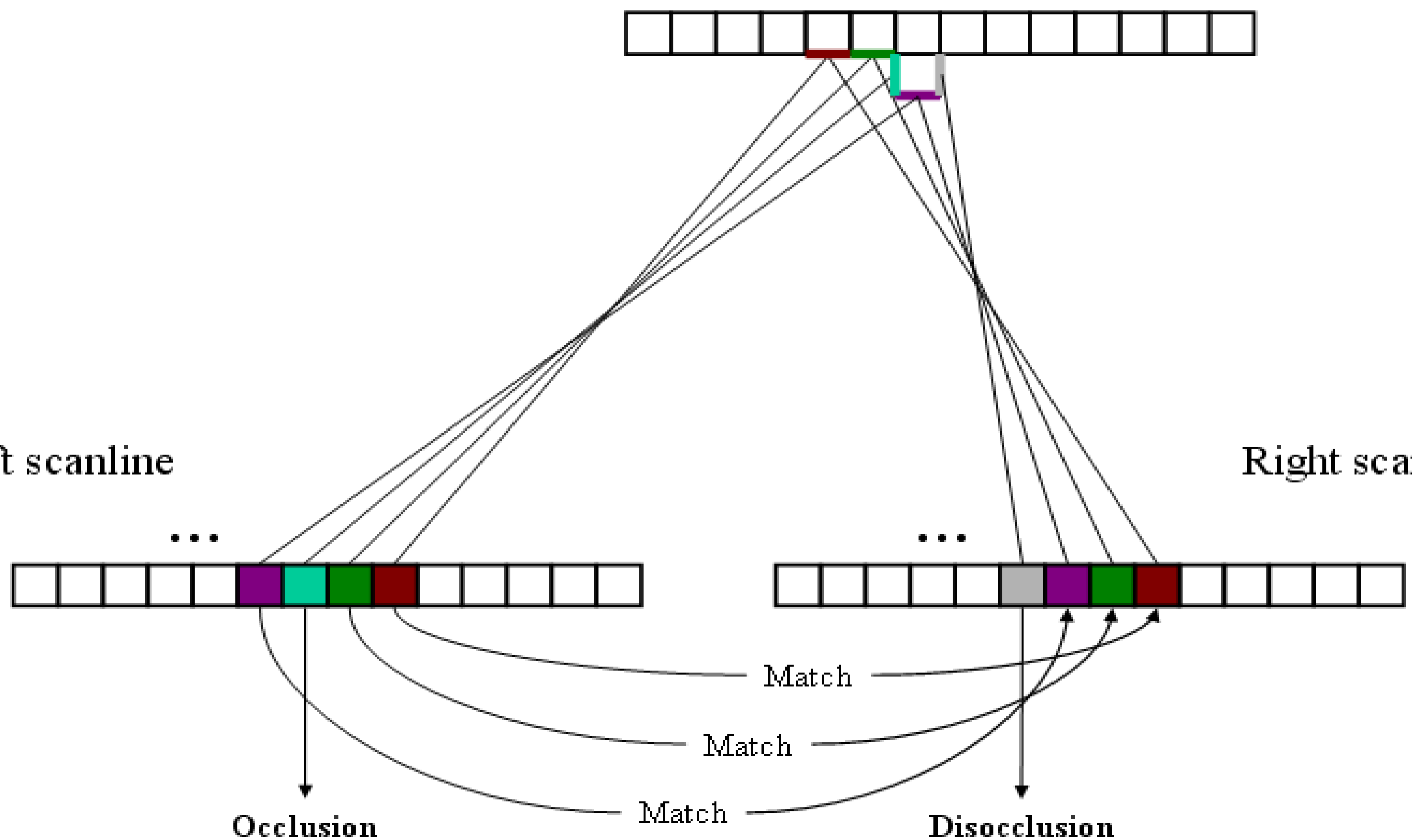
- Correspondence fail for smooth surfaces



- There is currently no good solution to the correspondence problem

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Occlusion [1]

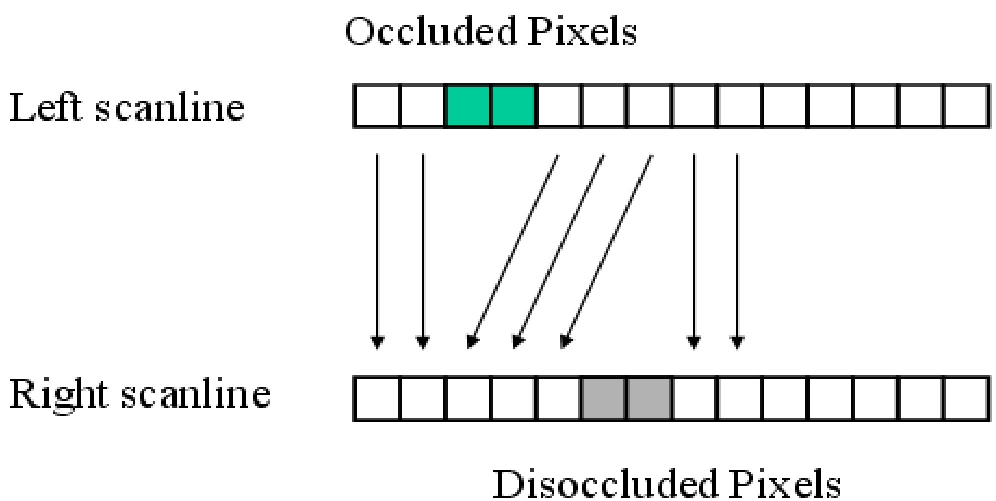


Left Occlusion

Right Occlusion

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Search over Correspondence [1]

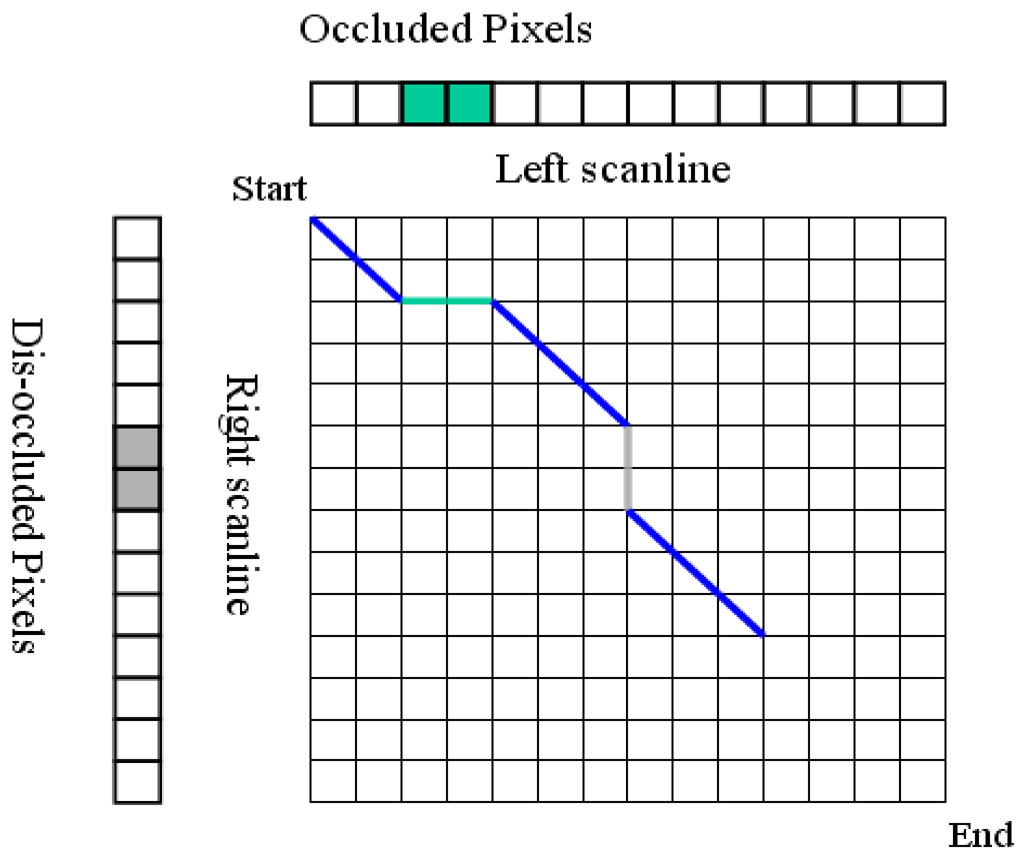


Three cases:

- 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)

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Matching with Dynamic Programming [1]

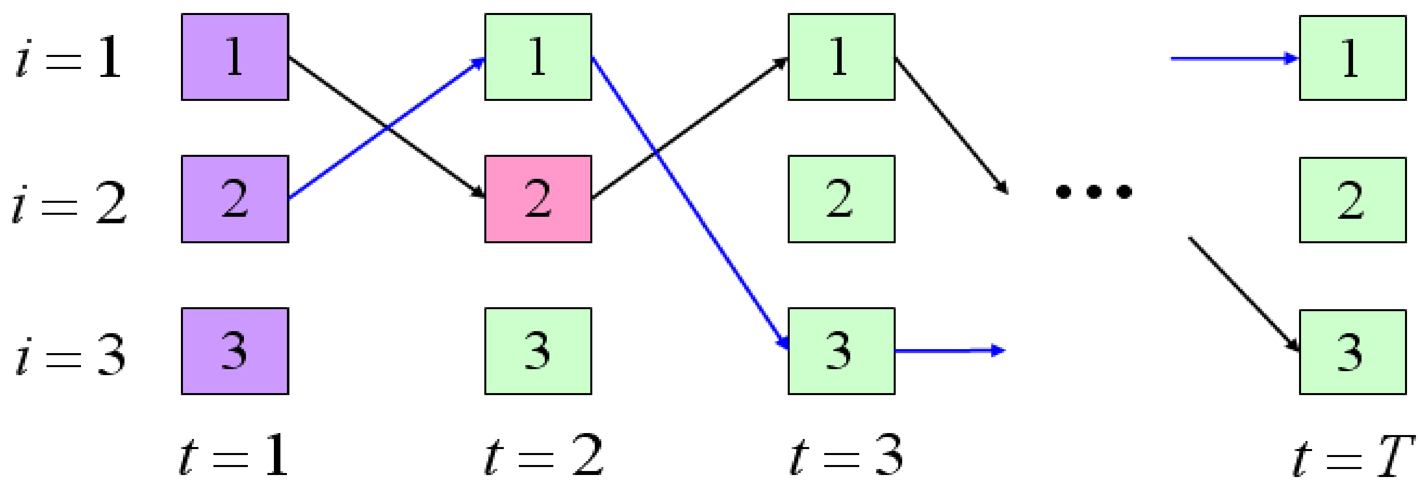


Dynamic programming yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

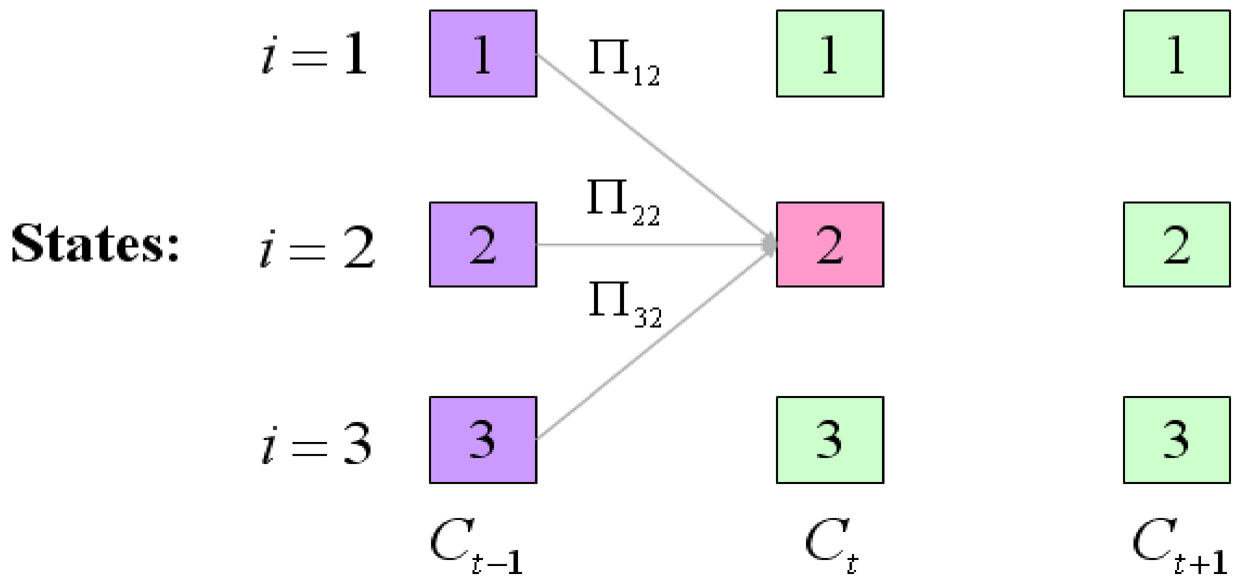
Dynamic Programming [1]

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



How many paths through this trellis? 3^T

Dynamic Programming [1]

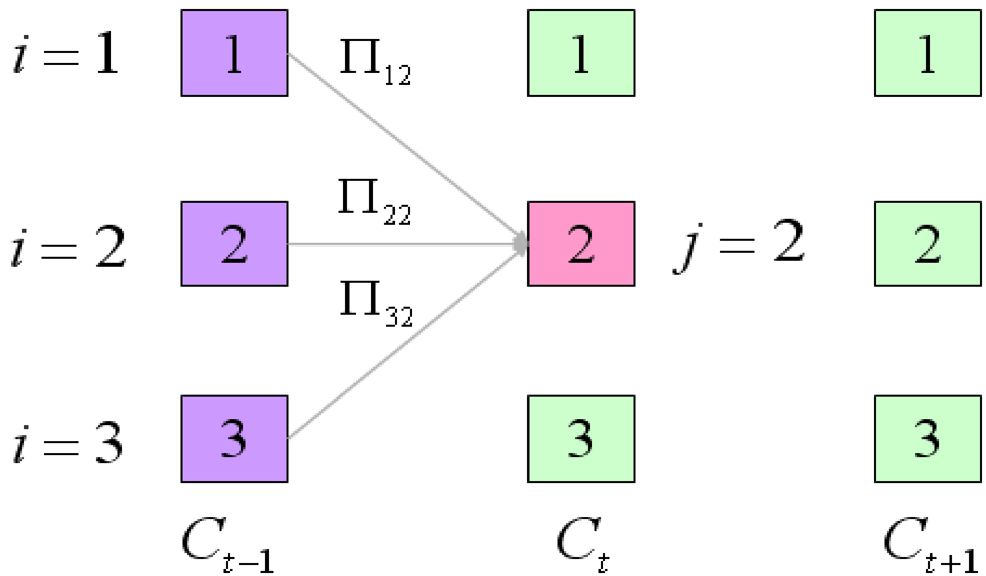


Suppose cost can be decomposed into stages:

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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Dynamic Programming [1]

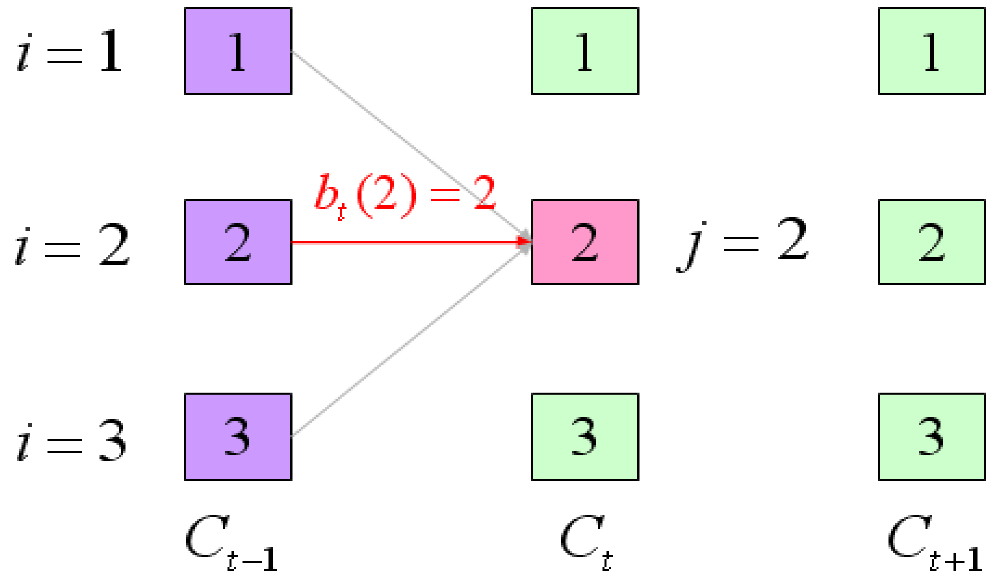


Principle of Optimality for an n-stage assignment problem:

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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Dynamic Programming [1]

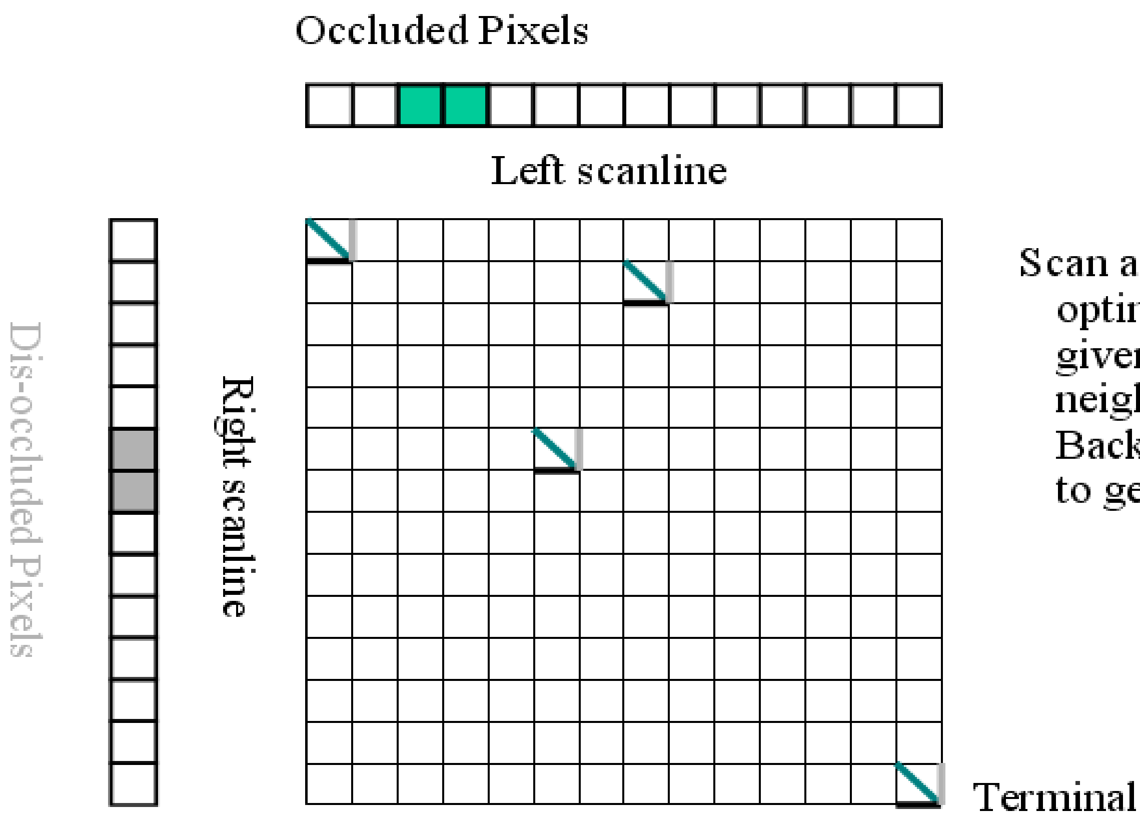


$$C_t(j) = \min_i (\Pi_{ij} + C_{t-1}(i))$$

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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

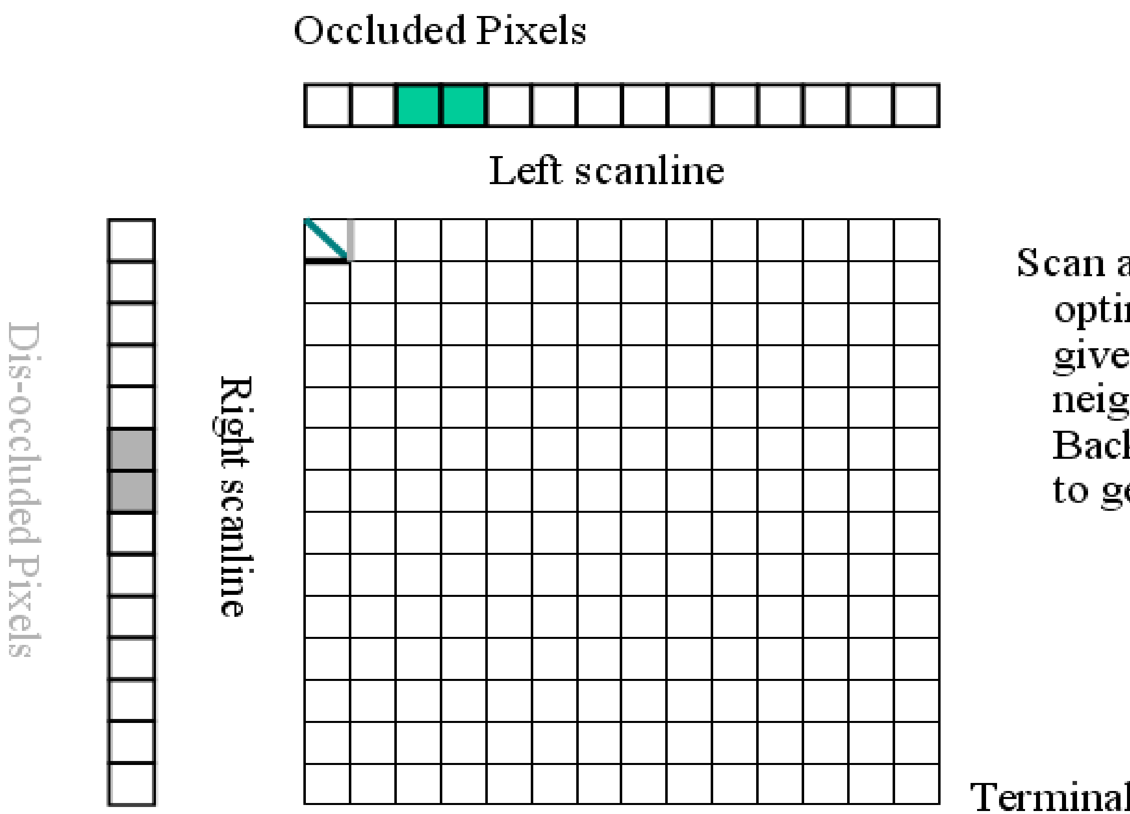
Dynamic Programming [3]



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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

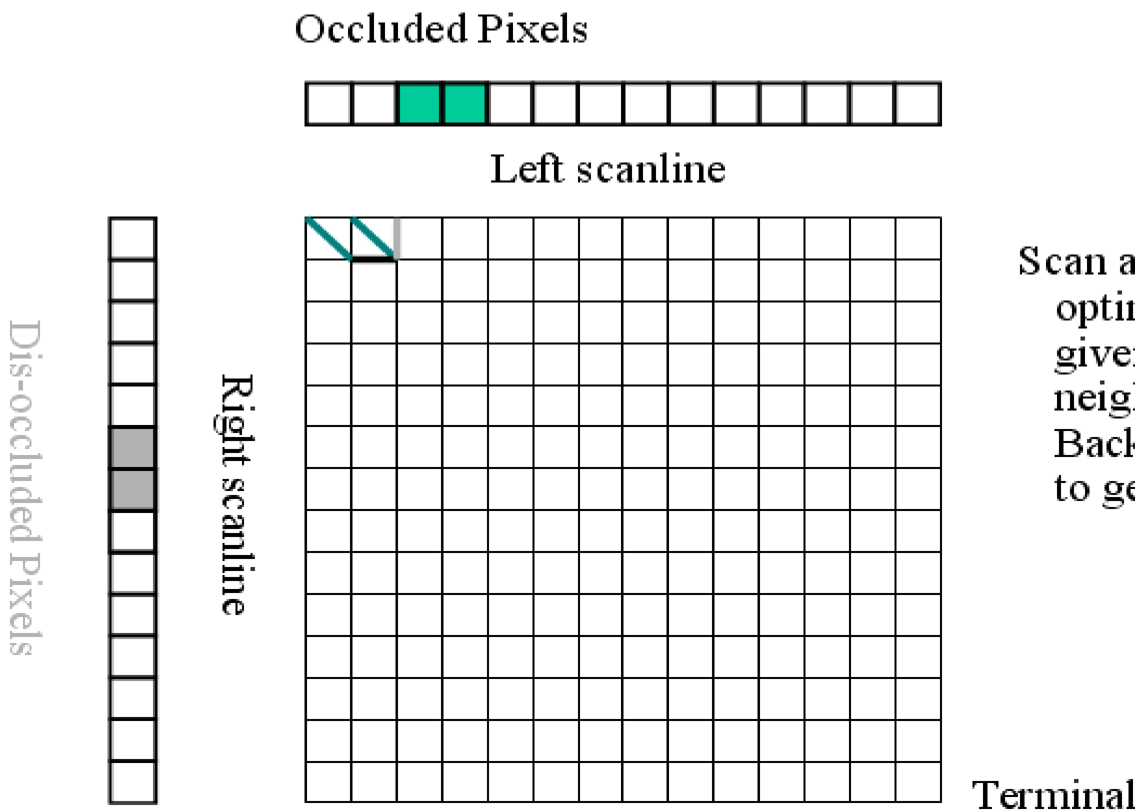
Dynamic Programming [3]



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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

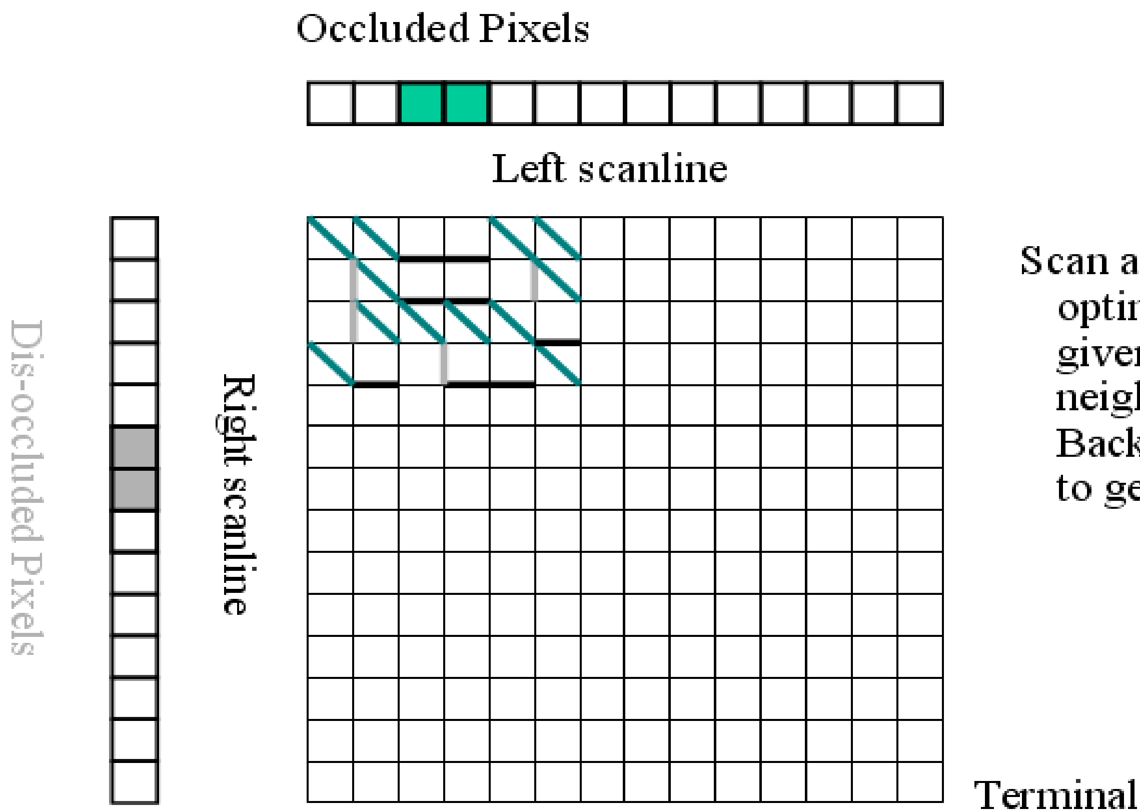
Dynamic Programming [3]



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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

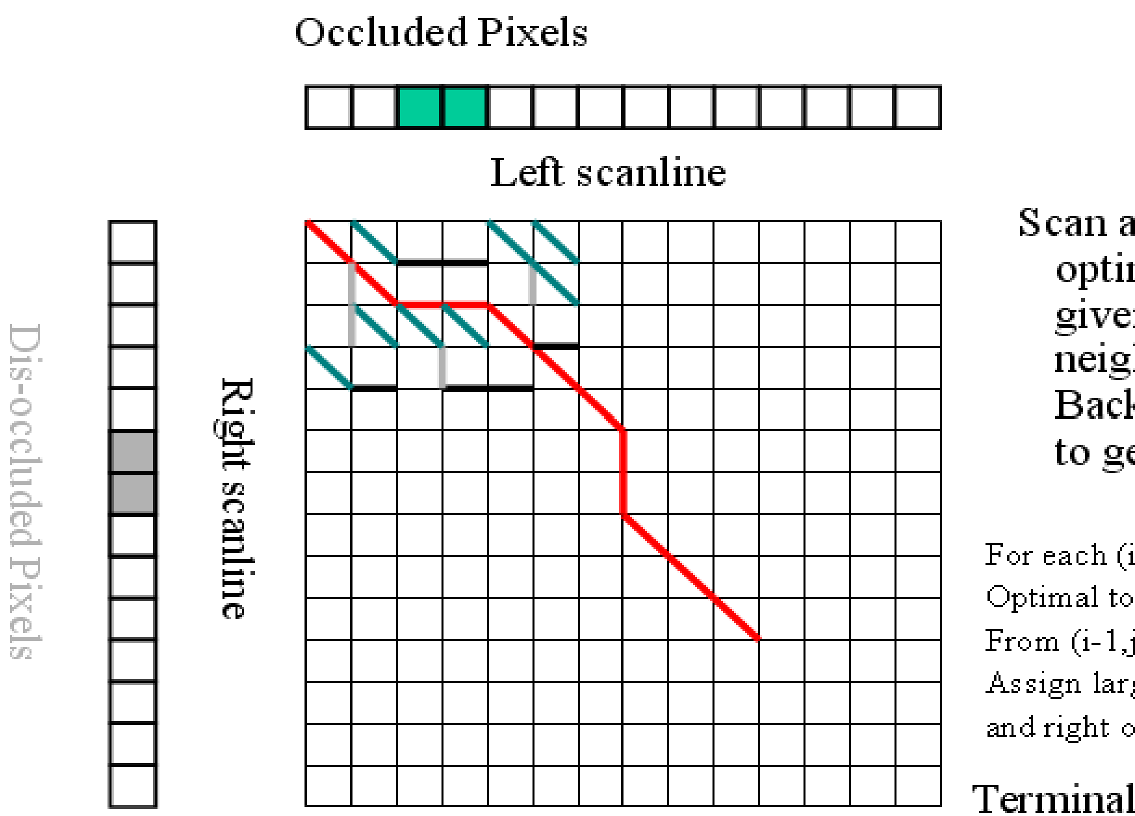
Dynamic Programming [3]



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

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Dynamic Programming [3]



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

For each (i,j), look for what is Optimal to get to: would it be From (i-1,j-1), or (i,j-1), or (i-1,j-1)?
Assign large values for left occlusion and right occlusion.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Dynamic Programming [3]

```
for(i=1;i≤ N;i++){
    for(j=1;j≤ M;j++){
        min1 = C(i-1,j-1)+c(z_{1,i},z_{2,j});
        min2 = C(i-1,j)+Occlusion;
        min3 = C(i,j-1)+Occlusion;
        C(i,j) = cmin = min(min1,min2,min3);
        if(min1==cmin) M(i,j) = 1;
        if(min2==cmin) M(i,j) = 2;
        if(min3==cmin) M(i,j) = 3;
    }}
```

Pseudo-code describing how to calculate the optimal match

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Dynamic Programming [3]

```
p=N;
q=M;
while(p!=0 && q!=0){
    switch(M(p,q)){
        case 1:
            p matches q
            p--;q--;
            break;
        case 2:
            p is unmatched
            p--;
            break;
        case 3:
            q is unmatched
            q--;
            break;
    }}
```

Pseudo-code describing how to reconstruct the optimal path

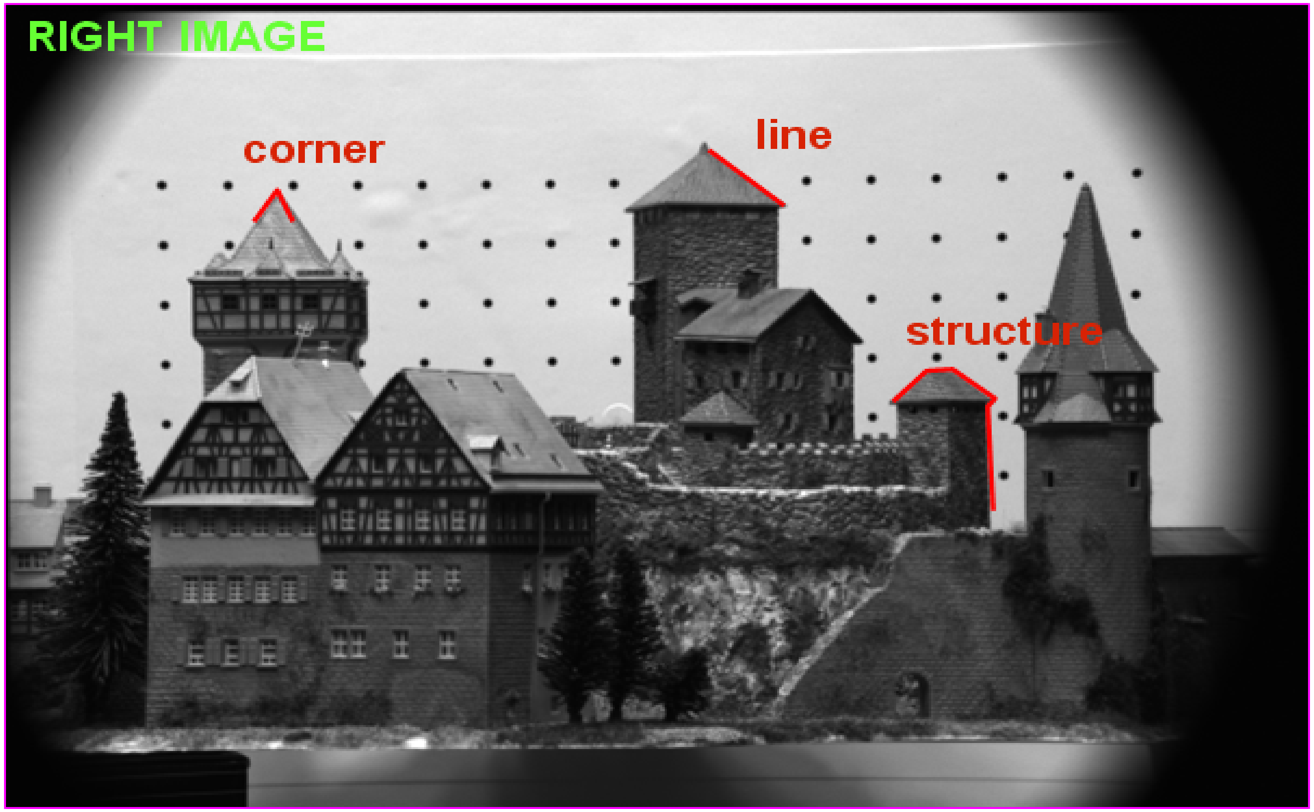
E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Dynamic Programming [3]

Local errors may be propagated along a scan-line and no inter scan-line consistency is enforced.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Correspondence by Feature [3]



- Search in the right image… the disparity (dx, dy) is the displacement when the similarity measure is maximum

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Segment-Based Stereo Matching [3]

Assumption

- Depth discontinuity tend to correlate well with color edges
- Disparity variation within a segment is small
- Approximating the scene with piece-wise planar surfaces

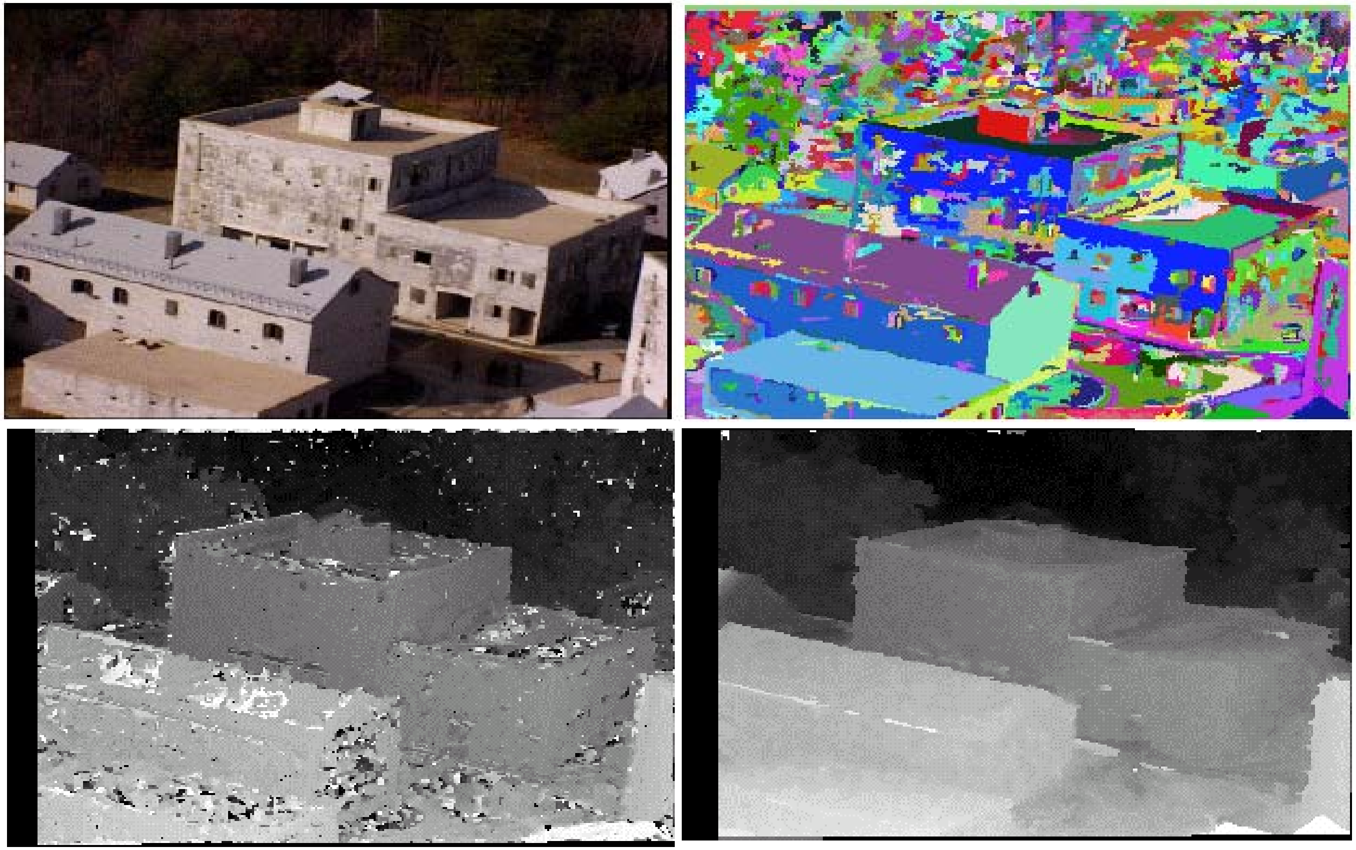
E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Segment-Based Stereo Matching [3]

- Plane equation is fitted in each segment based on initial disparity estimation obtained SSD or Correlation
- Global matching criteria: if a depth map is good, warping the reference image to the other view according to this depth will render an image that matches the real view
- Optimization by iterative neighborhood depth hypothesizing

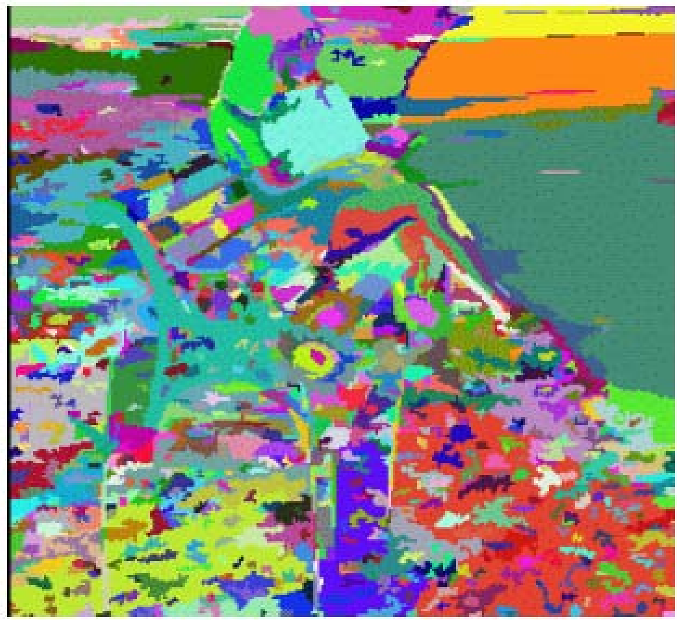
E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Segment-Based Stereo Matching [3]



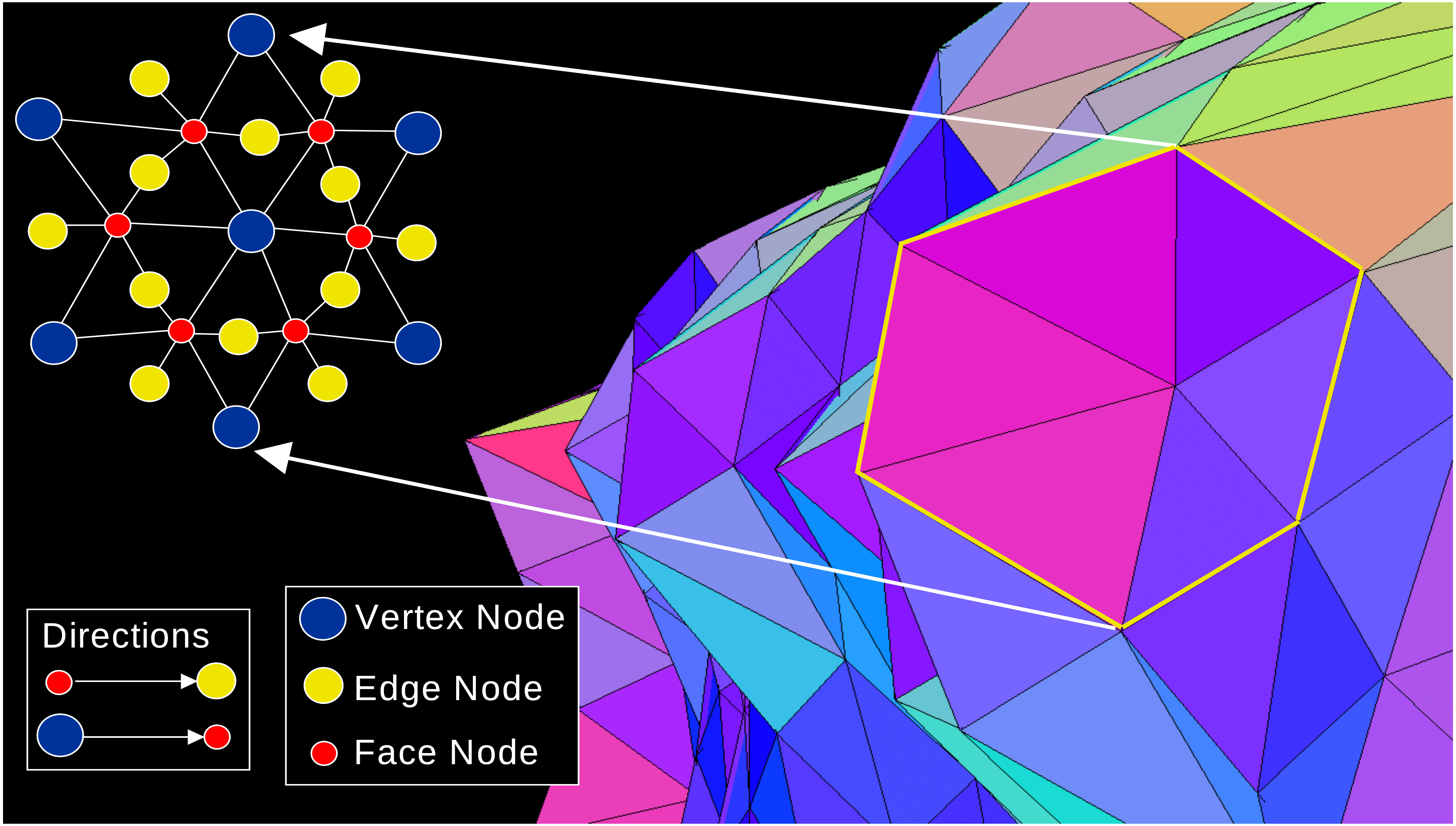
Hai Tao and Harpreet W. Sawhney

Segment-Based Stereo Matching [3]


E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Network of Constraints [2]



E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Network of Constraints [2]

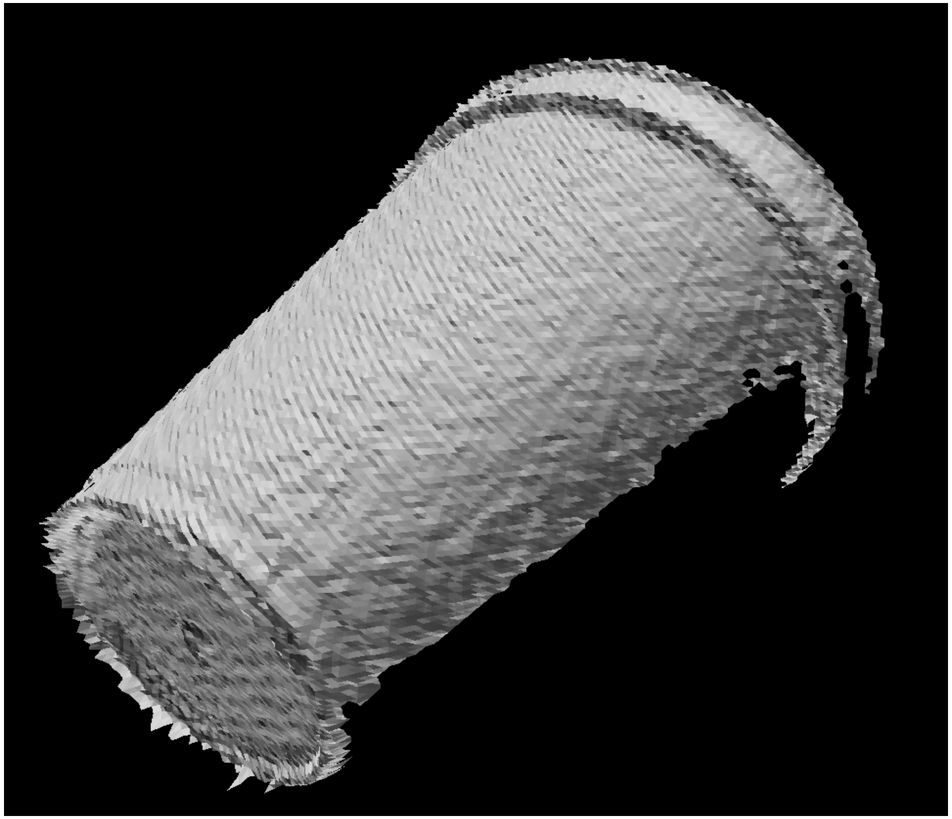
- Potential function: contains a sensor-model term and a surface prior

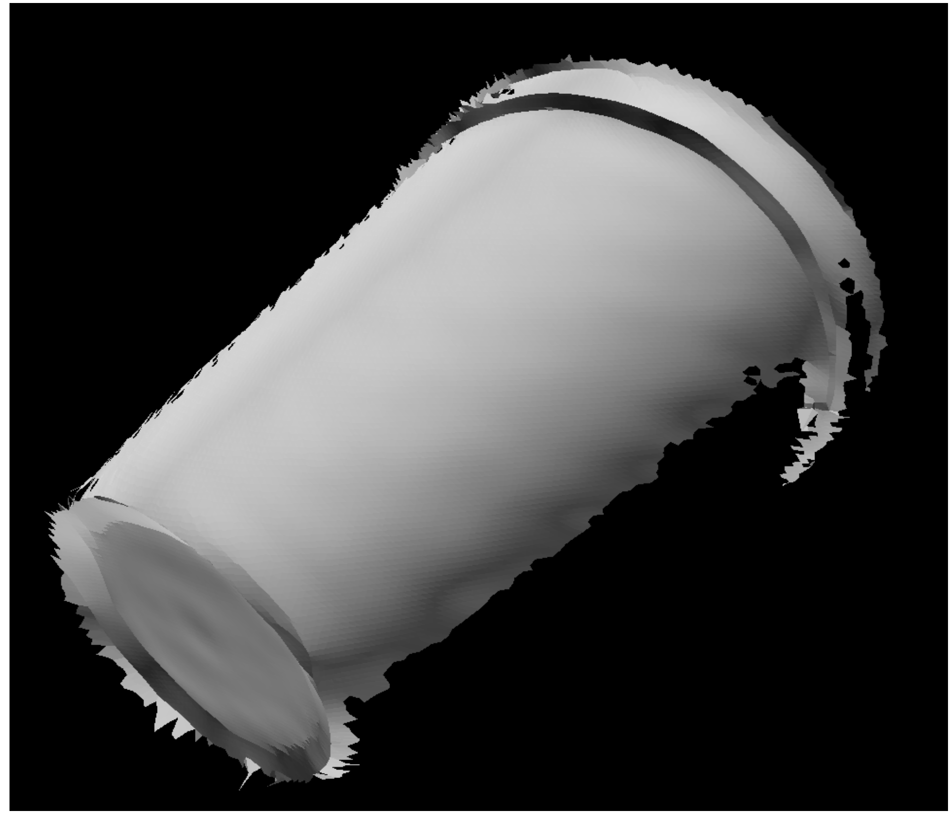
$$\Psi = \sum_i (x_i - x_{0i})^T \Omega_i (x_i - x_{0i}) + \sum_j \psi_j (1 - n_1 \cdot n_2)$$

- The edge potential is important!
- Minimize Ψ by *conjugate gradient*
  - Optimize systems with tens of thousands of parameters in just a couple seconds
  - Time to converge is O(N), between 0.7 sec (25,000 nodes in the MRF) and 25 sec (900,000 nodes)

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Smoothing by MRF [2]

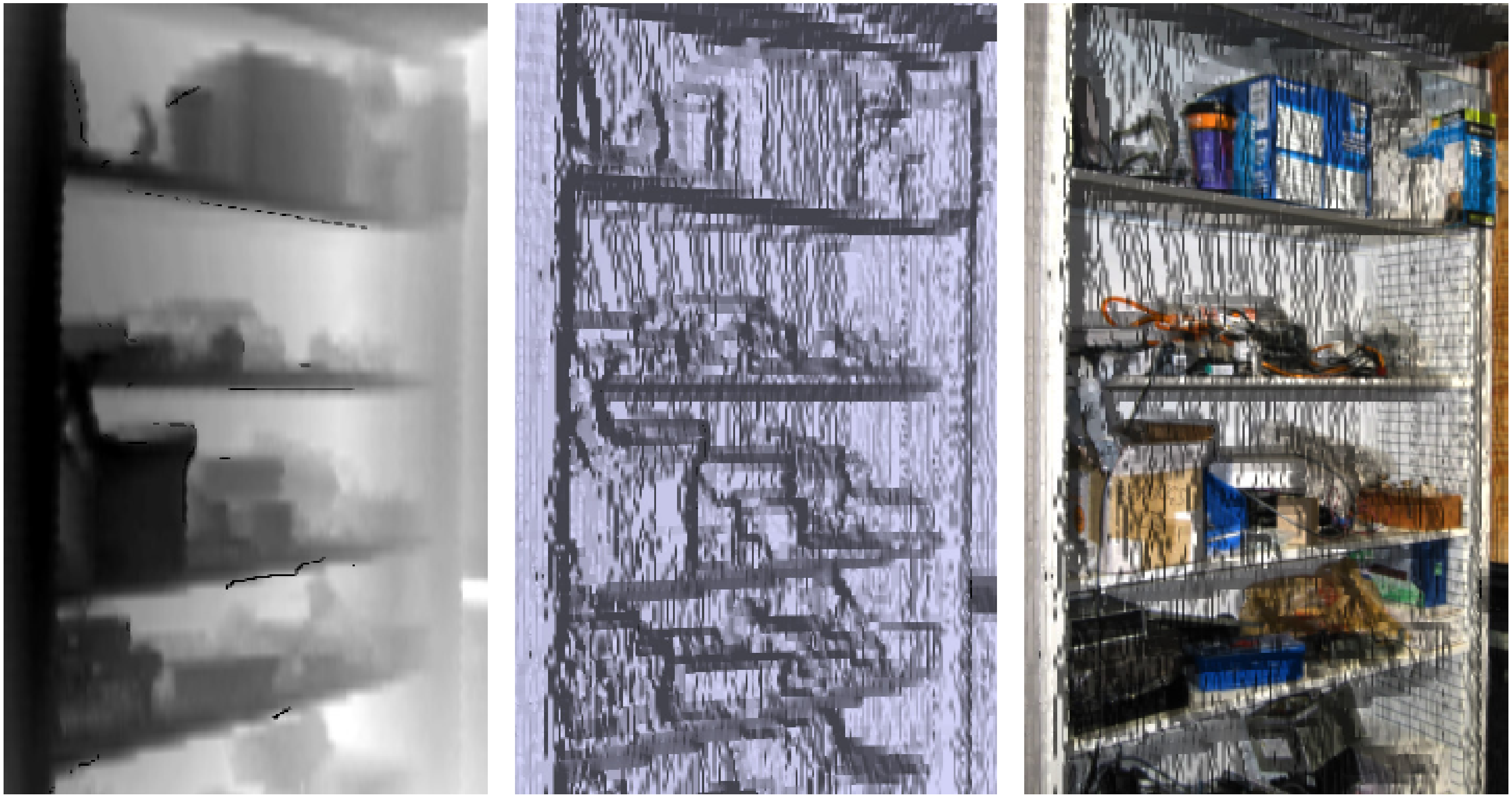




E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Smoothing by MRF [4]

(a) Raw low-res depth map (b) Raw low-res 3D model (c) Image mapped onto 3D model



E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Smoothing by MRF [4]

(d) MRF high-res depth map (e) MRF high-res 3D model (f) Image mapped onto 3D model

Smoothing by MRF [4]

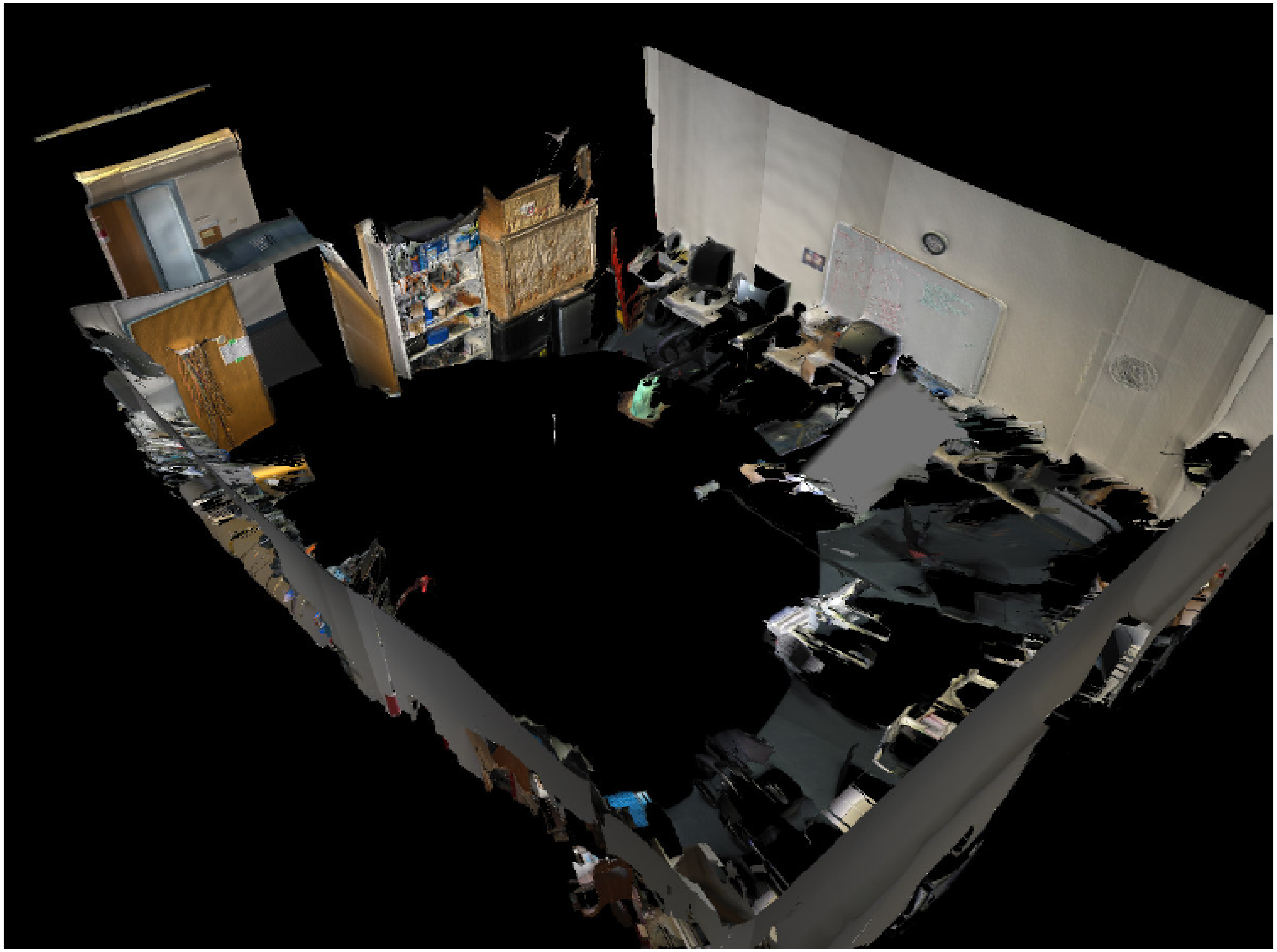


Figure 5: 3D model of a larger indoor environment, after applying our MRF.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Testing and Comparison [1]

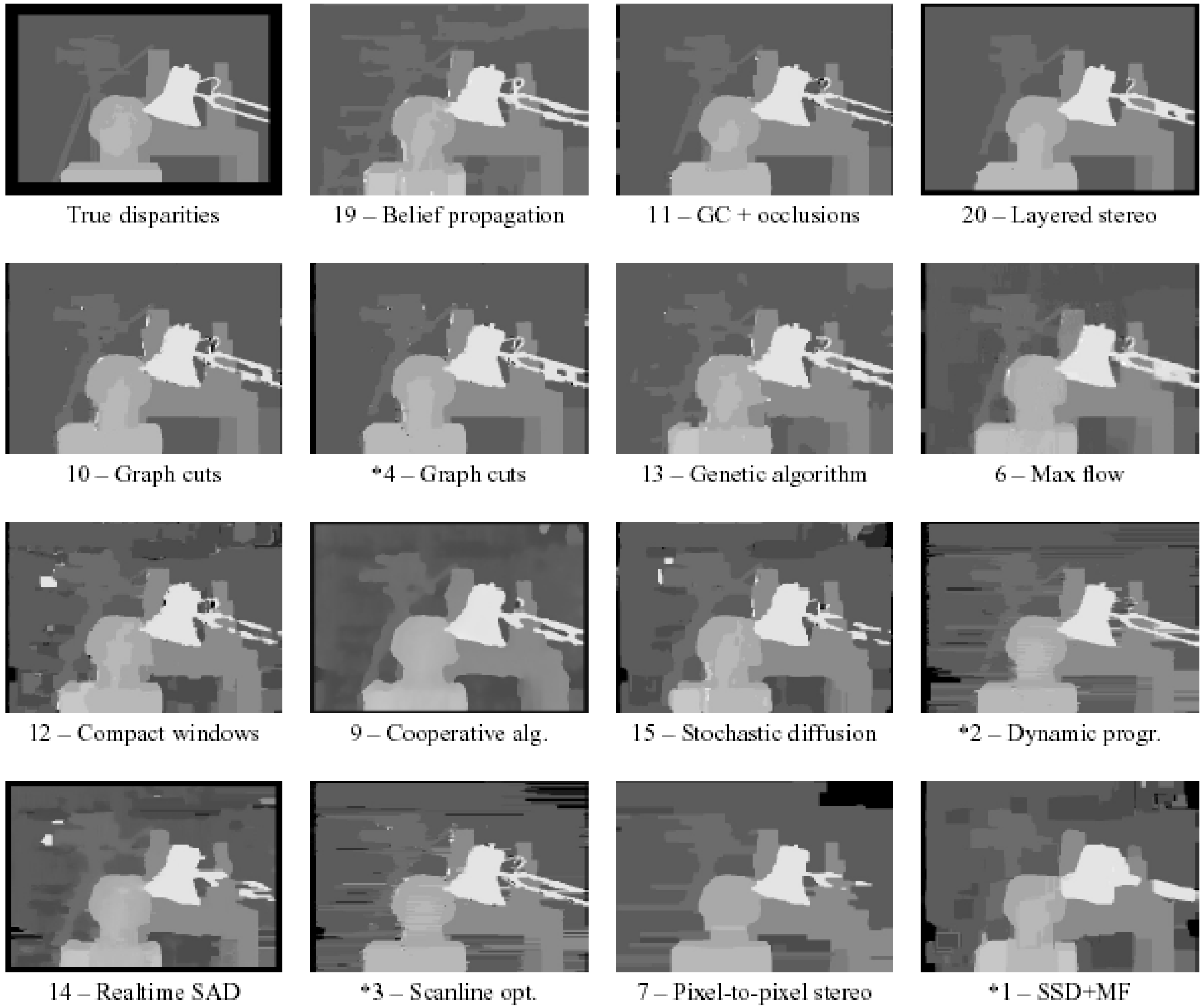
D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision,* 47 (2002), pp. 7-42.

Scene

Ground truth

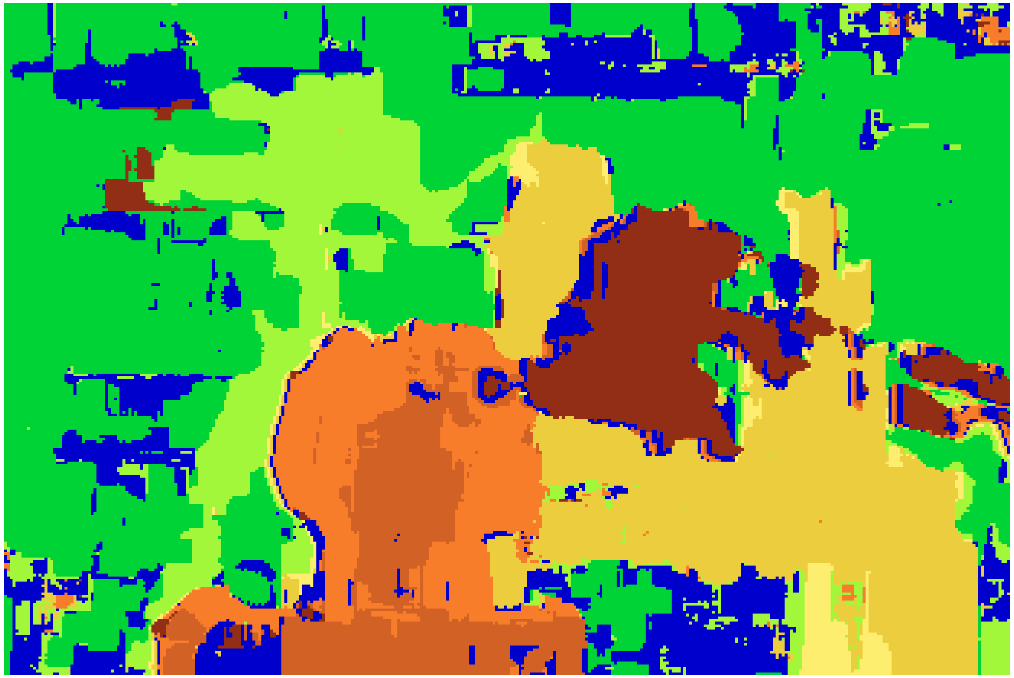
E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Testing and Comparison [1]



E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Testing and Comparison [1]

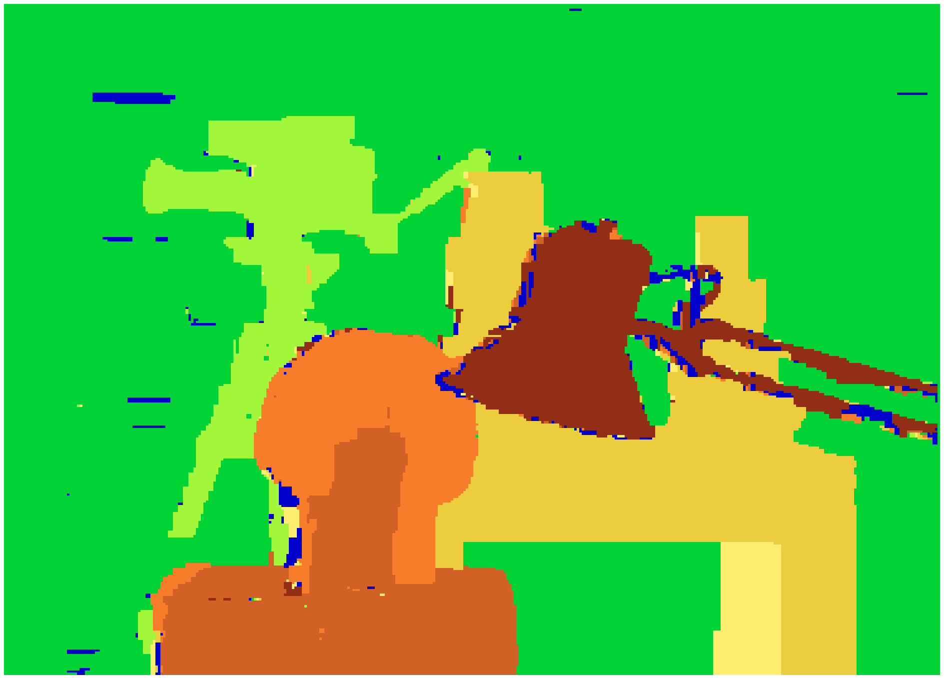


Window-based matching
(best window size)

Ground truth

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Stereo Testing and Comparison [1]



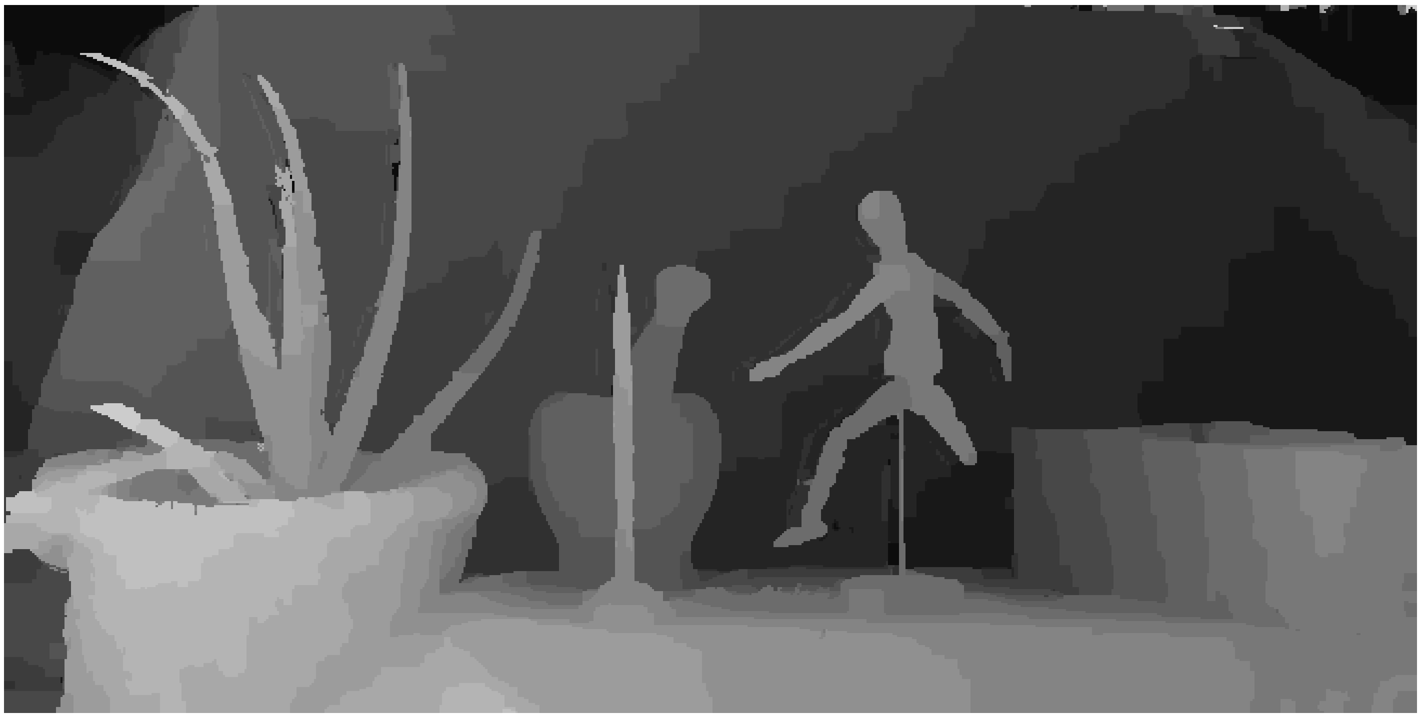
State of the art method

Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Intermediate View Reconstruction [1]



Right Image
Left Image
Disparity

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Intermediate View Reconstruction [1]

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

Summary of Different Stereo Methods

- **Constraints:**
  - Geometry, epipolar constraint.
  - Photometric: Brightness constancy, only partly true.
  - Ordering: only partly true.
  - Smoothness of objects: only partly true.

- **Algorithms:**
  - What you compare: points, regions, features?

- **How you optimize:**
  - Local greedy matches.
  - 1D search.
  - 2D search.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung

References

1. David Lowe, “Stereo,” UBC(Univ. of British Columbia) Lecture Material of Computer Vision (CPSC 425), Spring 2007.
2. Sebastian Thrun, Rick Szeliski, Hendrik Dahlkamp and Dan Morris, “Stereo 2,” Stanford Lecture Material of Computer Vision (CS 223B), Winter 2005.
3. Chandra Kambhamettu, “Multiple Views1” and “Multiple View2,” Univ. of Delawave Lecture Material of Computer Vision (CISC 4/689), Spring 2007.
4. J. Diebel and S. Thrun, “An Application of Markov Random Fields to Range Sensing,” Proc. Neural Information Processing Systems (NIPS), Cmbridge, MA, 2005. MIT Press.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung