



Stereo matching

Reading: Chapter 11 In Szeliski's book



Schedule (tentative)

#	date	topic
1	Sep.22	Introduction and geometry
2	Sep.29	Invariant features
3	Oct.6	Camera models and calibration
4	Oct.13	Multiple-view geometry
5	Oct.20	Model fitting (RANSAC, EM, ...)
6	Oct.27	Stereo Matching
7	Nov.3	2D Shape Matching
8	Nov.10	Segmentation
9	Nov.17	Shape from Silhouettes
10	Nov.24	Structure from motion
11	Dec.1	Optical Flow
12	Dec.8	Tracking (Kalman, particle filter)
13	Dec.15	Object category recognition
14	Dec.22	Specific object recognition



An Application: Mobile Robot Navigation



The Stanford Cart,
H. Moravec, 1979.

The INRIA Mobile Robot, 1990.

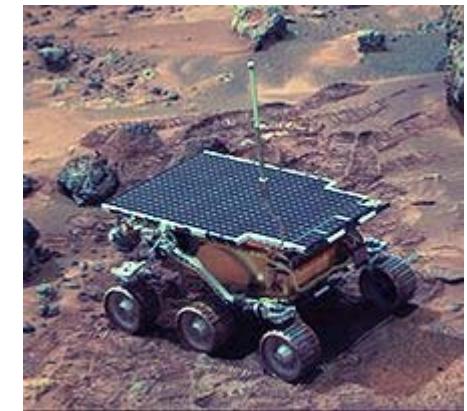
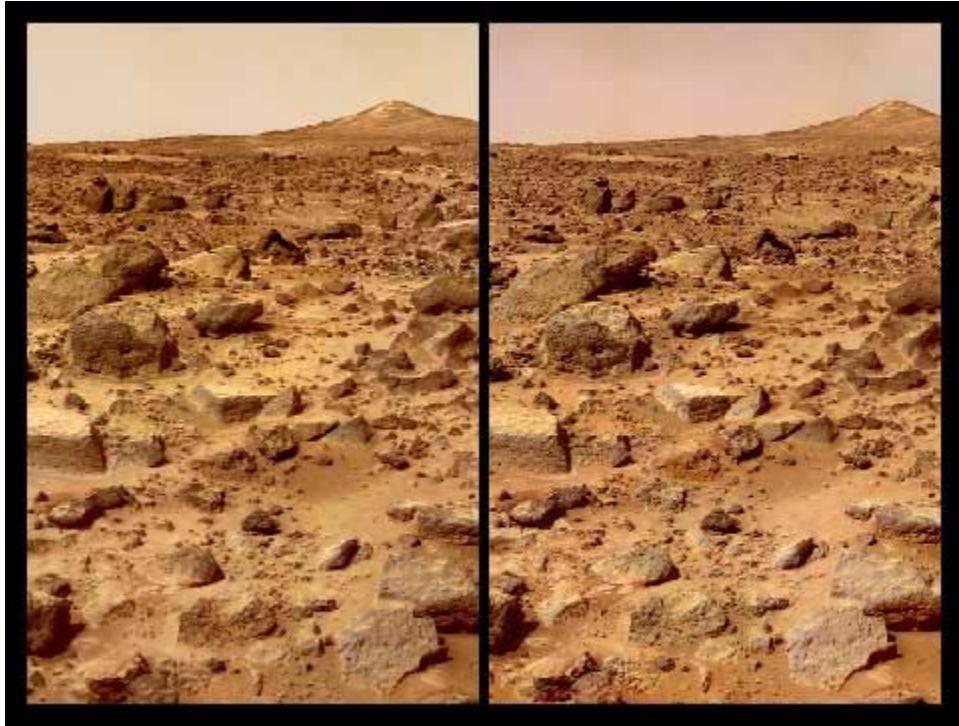


Courtesy O. Faugeras and H. Moravec.



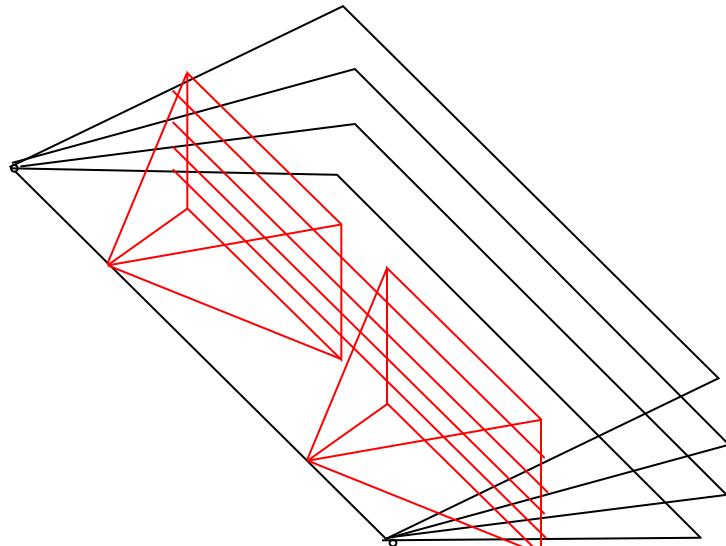
Stereo

NASA Mars Rover





Standard stereo geometry



PointGrey Bumblebee

pure translation along X-axis

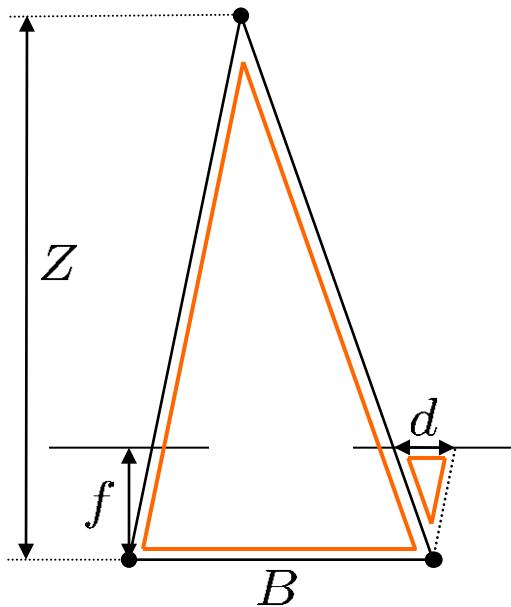
$$\mathbf{F} = [t]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{F}_x = \begin{bmatrix} 0 \\ 1 \\ -y \end{bmatrix}$$

$$\mathbf{F}^{\top} \mathbf{x}' = \begin{bmatrix} 0 \\ 1 \\ -y' \end{bmatrix}$$



Standard stereo geometry

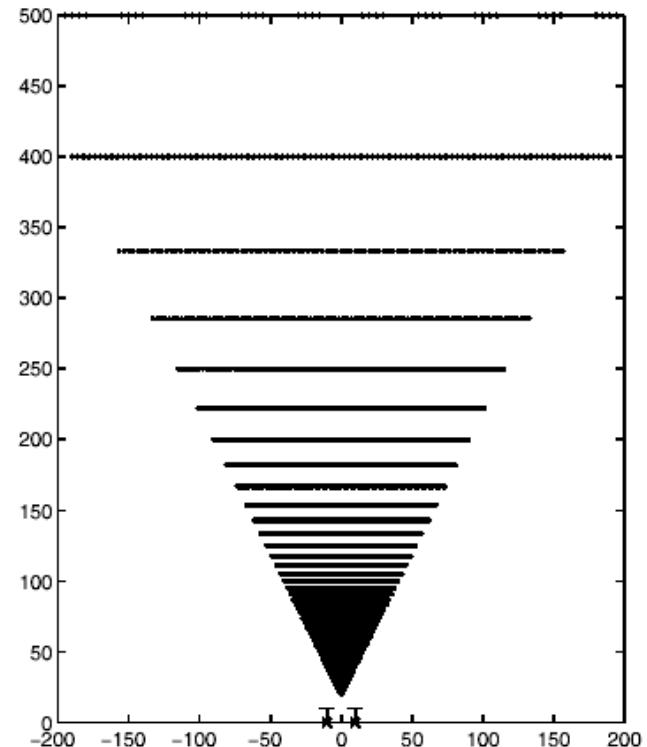


$$\frac{B}{Z} = \frac{d}{f}$$

$$d = -\frac{Bf}{Z}$$

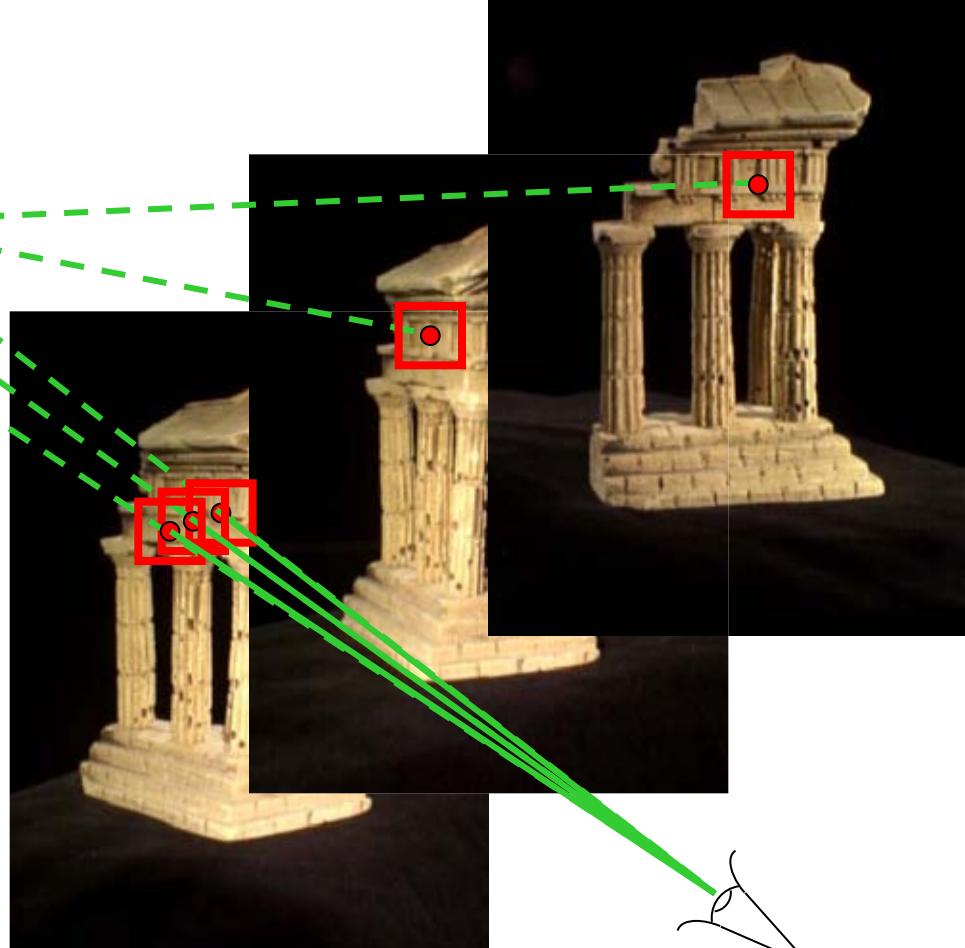
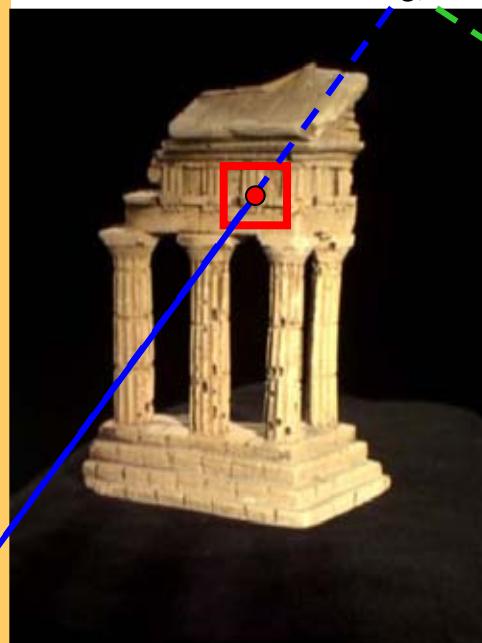
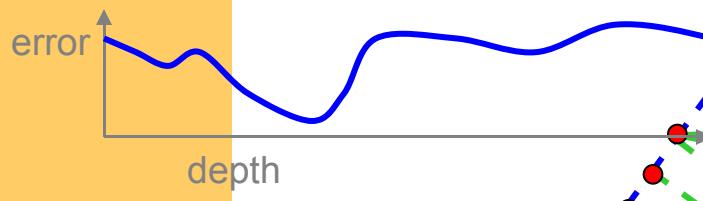
$$\frac{dd}{dZ} = \frac{Bf}{Z^2}$$

$$\Delta Z = \frac{Z^2}{Bf} \Delta d$$





Stereo: basic idea





Stereo matching

- Search is limited to epipolar line (1D)
- Look for most similar pixel

```
for x=1:w,  
    for y=1:h,  
        bestdist=inf;  
        for i=-dr:0,  
            if (dist(pix(x,y),pix(x+i,y))<bestdist)  
                d(x,y)=i; best=sim(pix(x,y),pix(x+i,y)); end  
        end  
    end  
end
```





Stereopsis

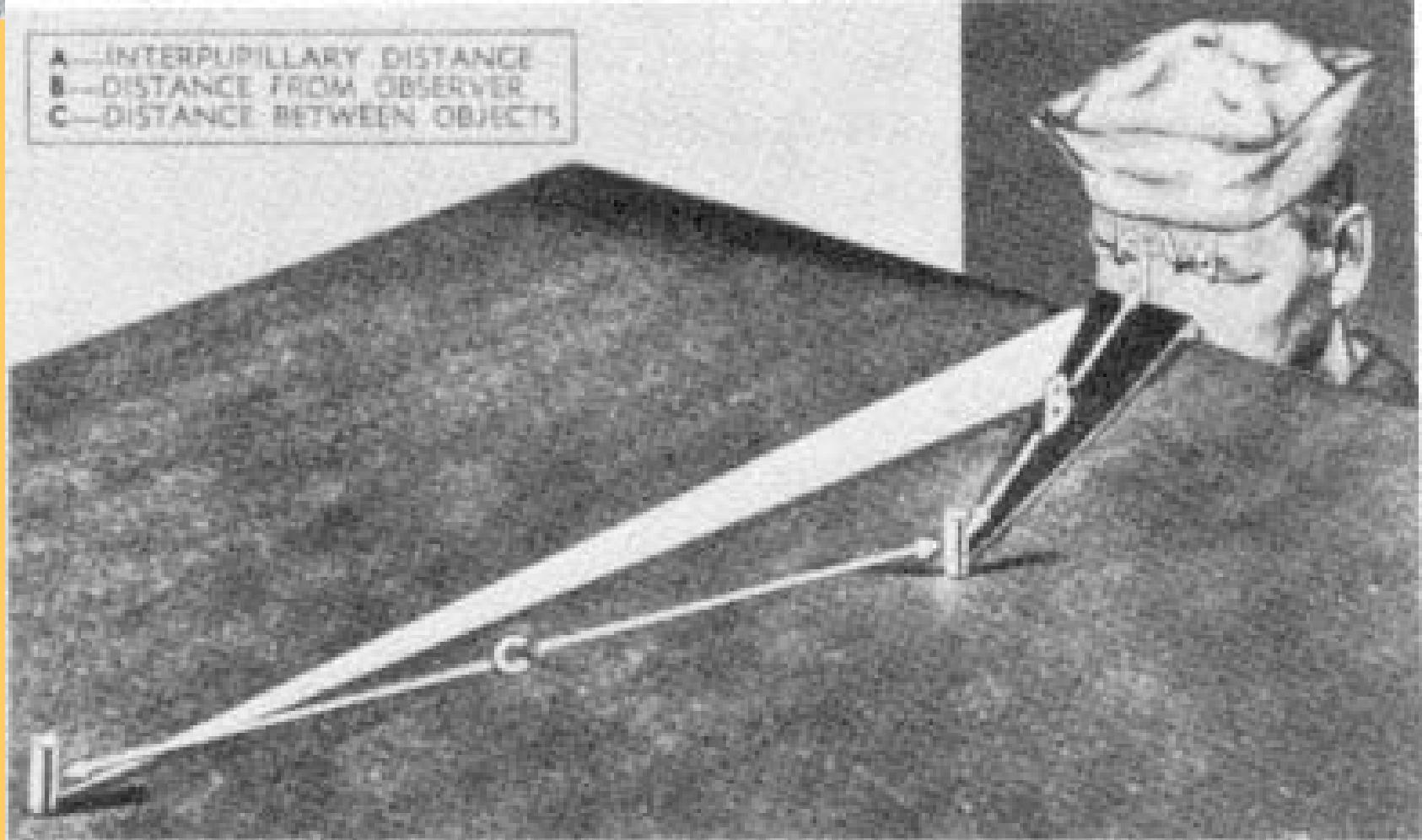
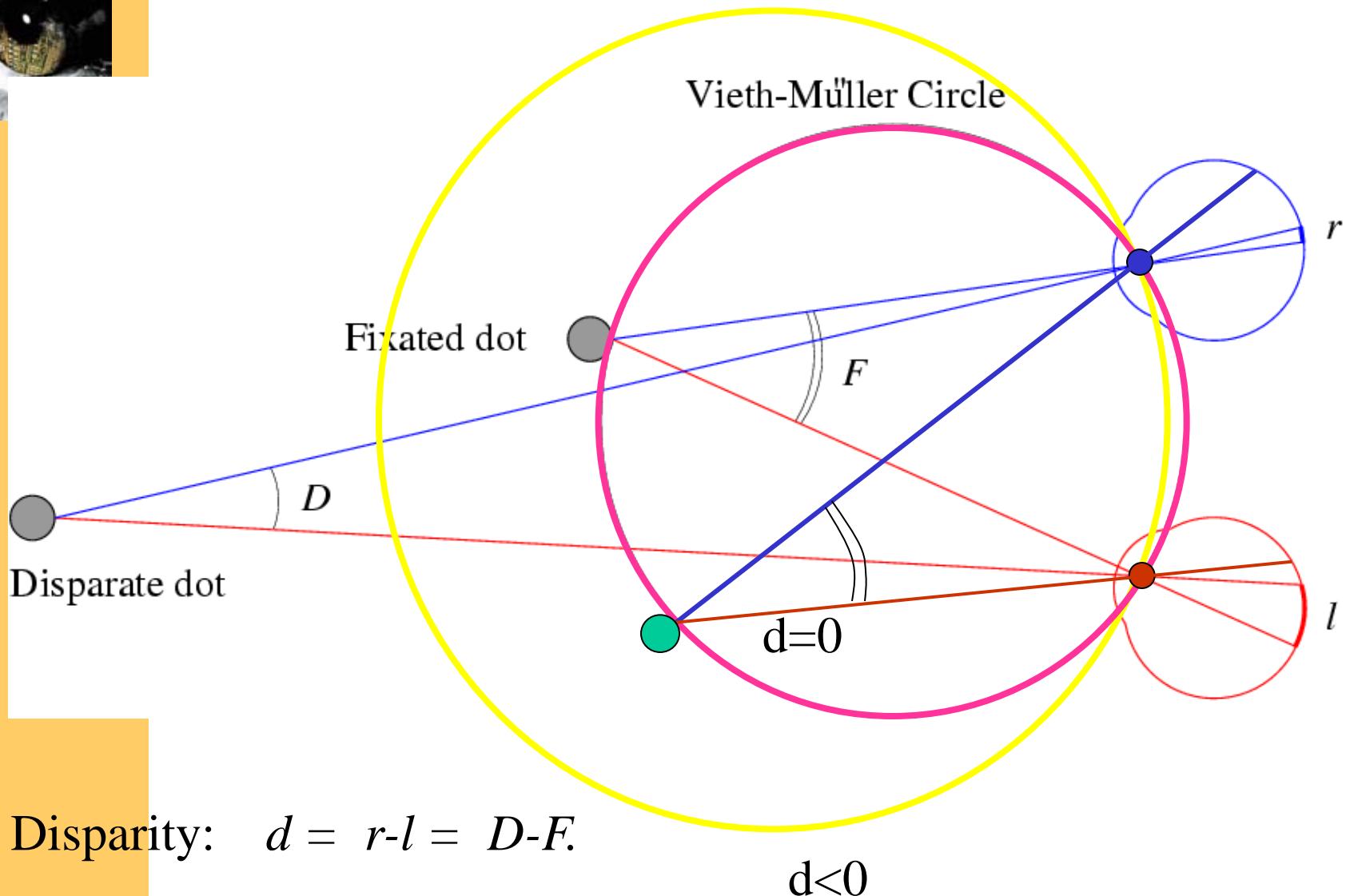


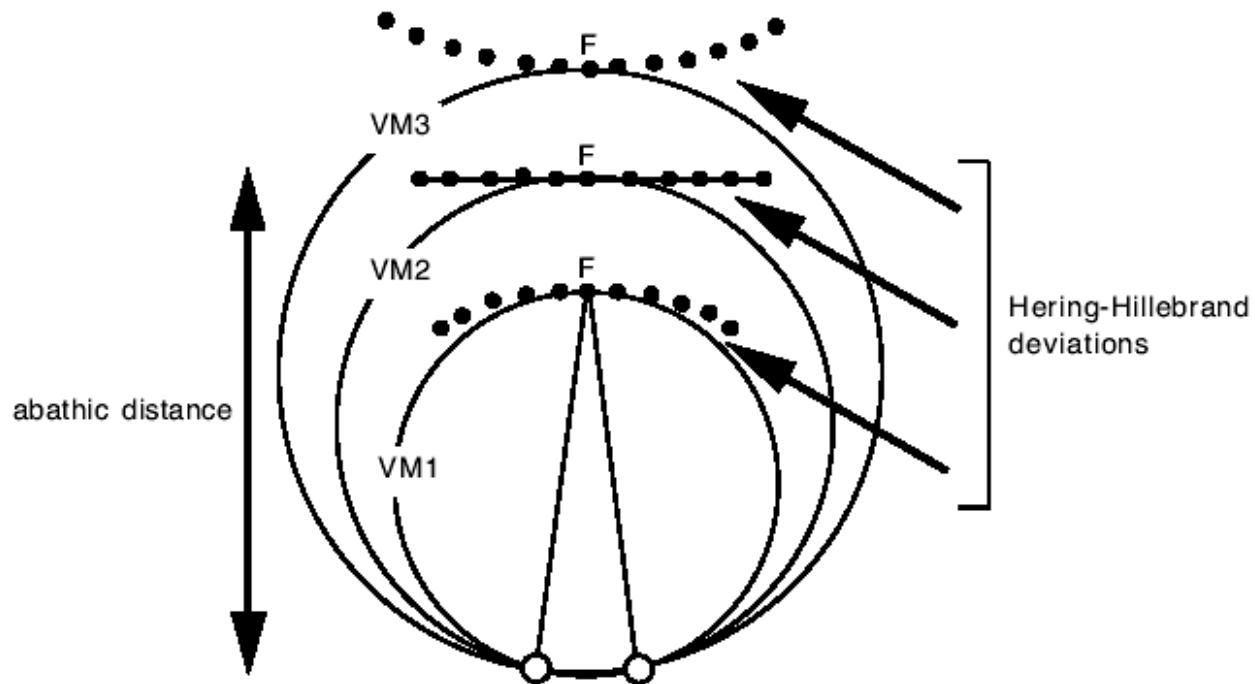
Figure from US Navy Manual of Basic Optics and Optical Instruments, prepared by Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.

Human Stereopsis: Reconstruction



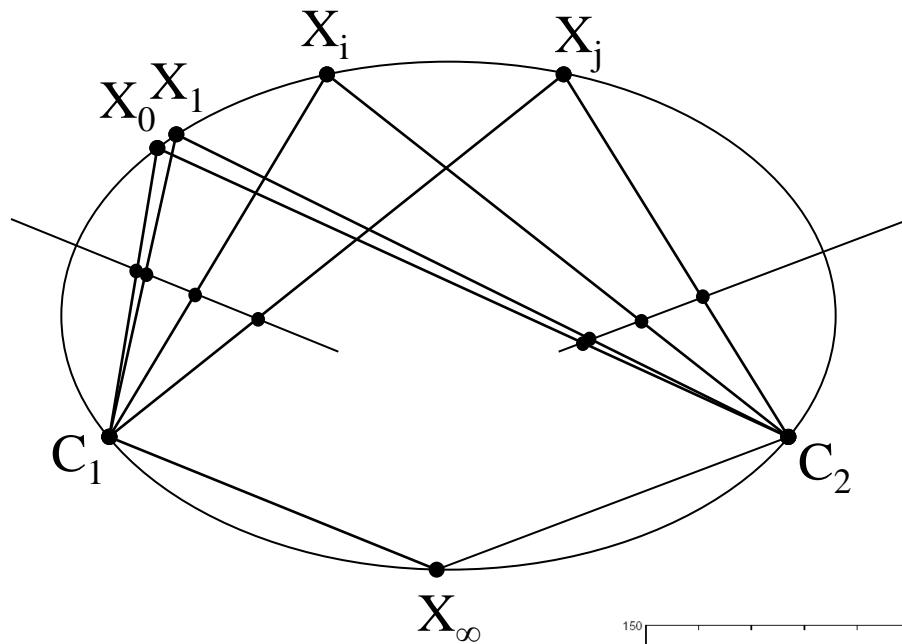
In 3D, the horopter.

Human Stereopsis: experimental horopter...

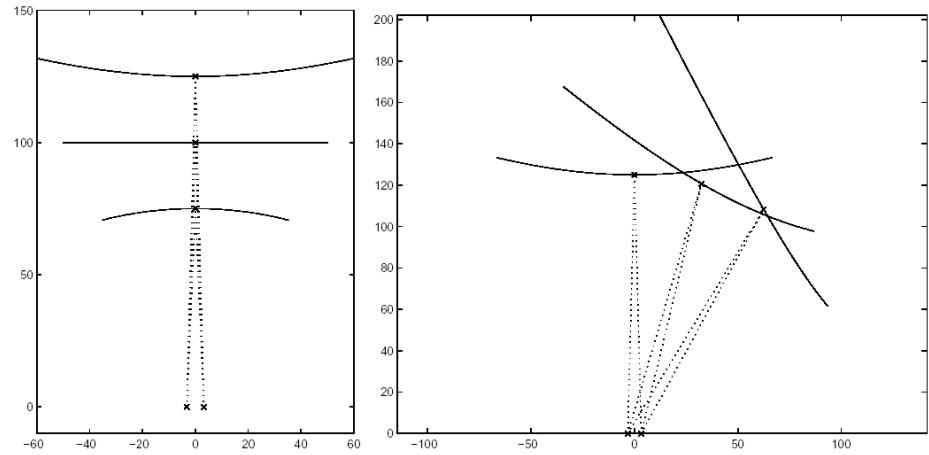




Iso-disparity curves: planar retinas



$$\frac{\begin{vmatrix} 0 & i \\ 1 & 1 \end{vmatrix}}{\begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix}} : \frac{\begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}}{\begin{vmatrix} i & 1 \\ 1 & 0 \end{vmatrix}} = \frac{-i}{-1} : \frac{-1}{-1} = i$$





Human Stereopsis: Reconstruction

What if F is not known?

Helmoltz (1909):

- There is evidence showing the vergence angles cannot be measured precisely.
- Humans get fooled by bas-relief sculptures.
- There is an analytical explanation for this.
- Relative depth can be judged accurately.

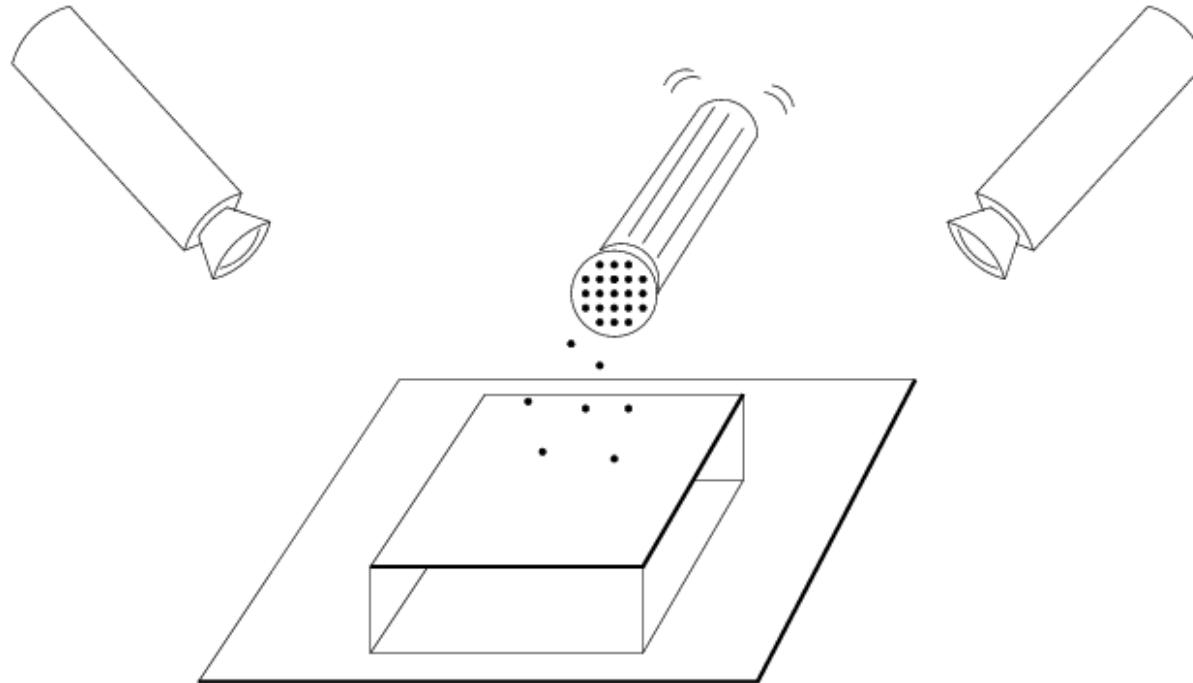


Human Stereopsis: Binocular Fusion

How are the correspondences established?

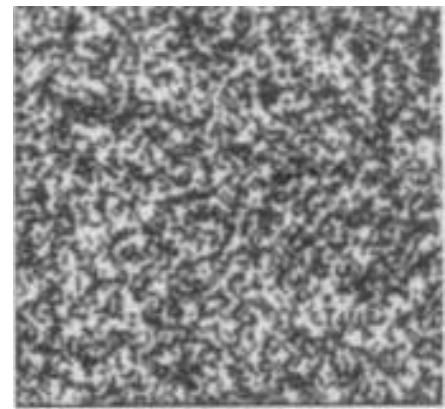
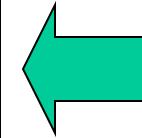
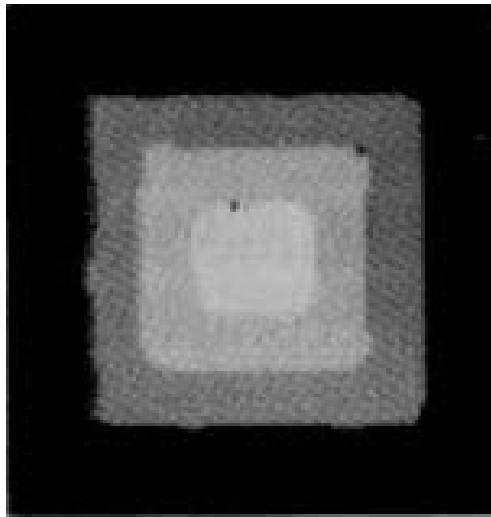
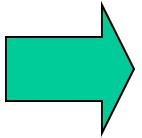
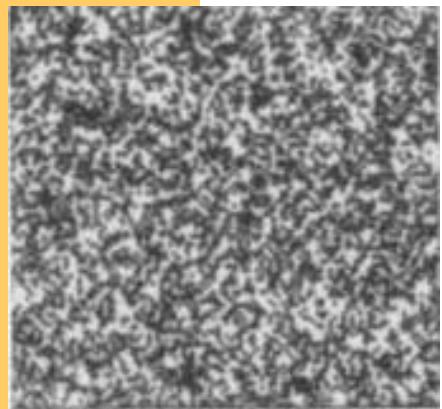
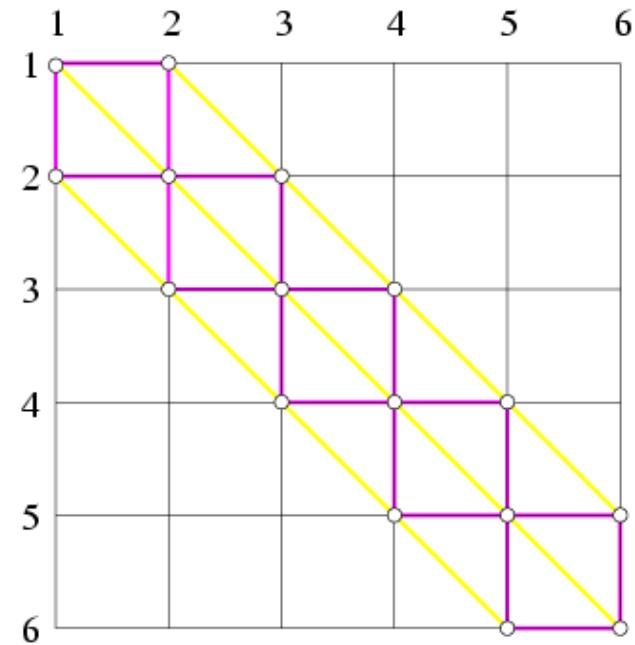
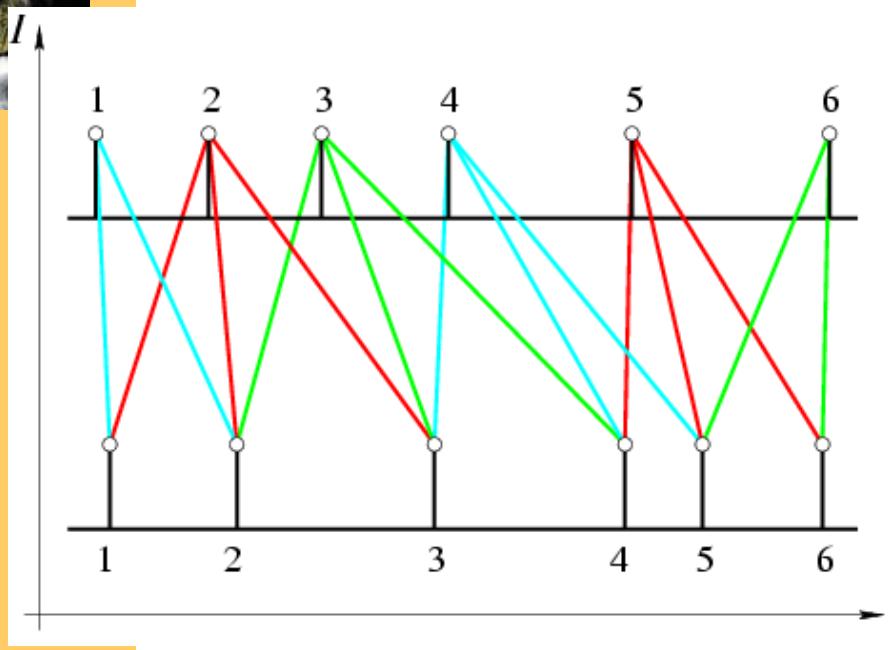
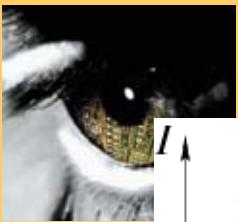
Julesz (1971): Is the mechanism for binocular fusion a monocular process or a binocular one??

- There is anecdotal evidence for the latter (camouflage).

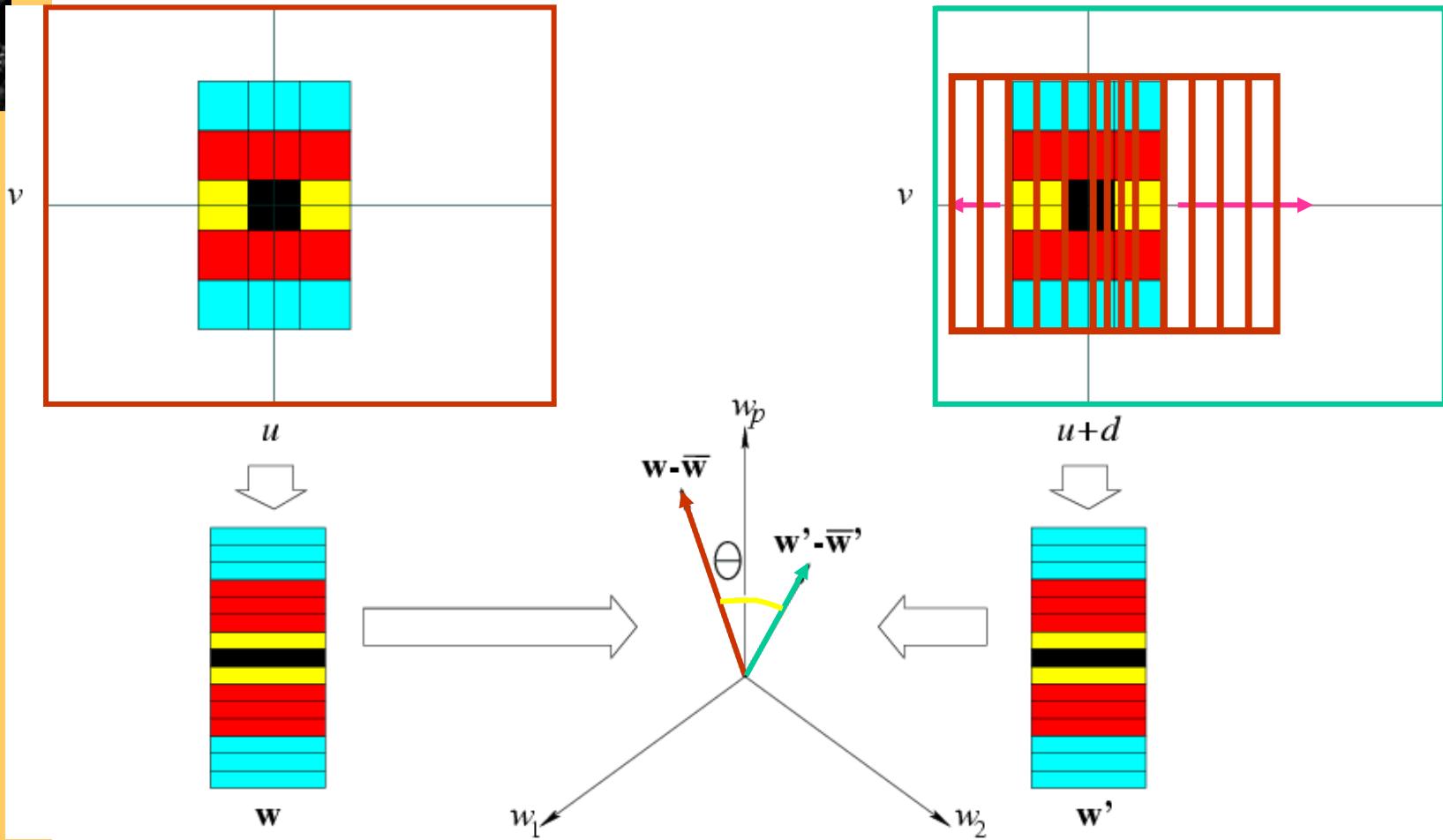


- Random dot stereograms provide an objective answer

A Cooperative Model (Marr and Poggio, 1976)



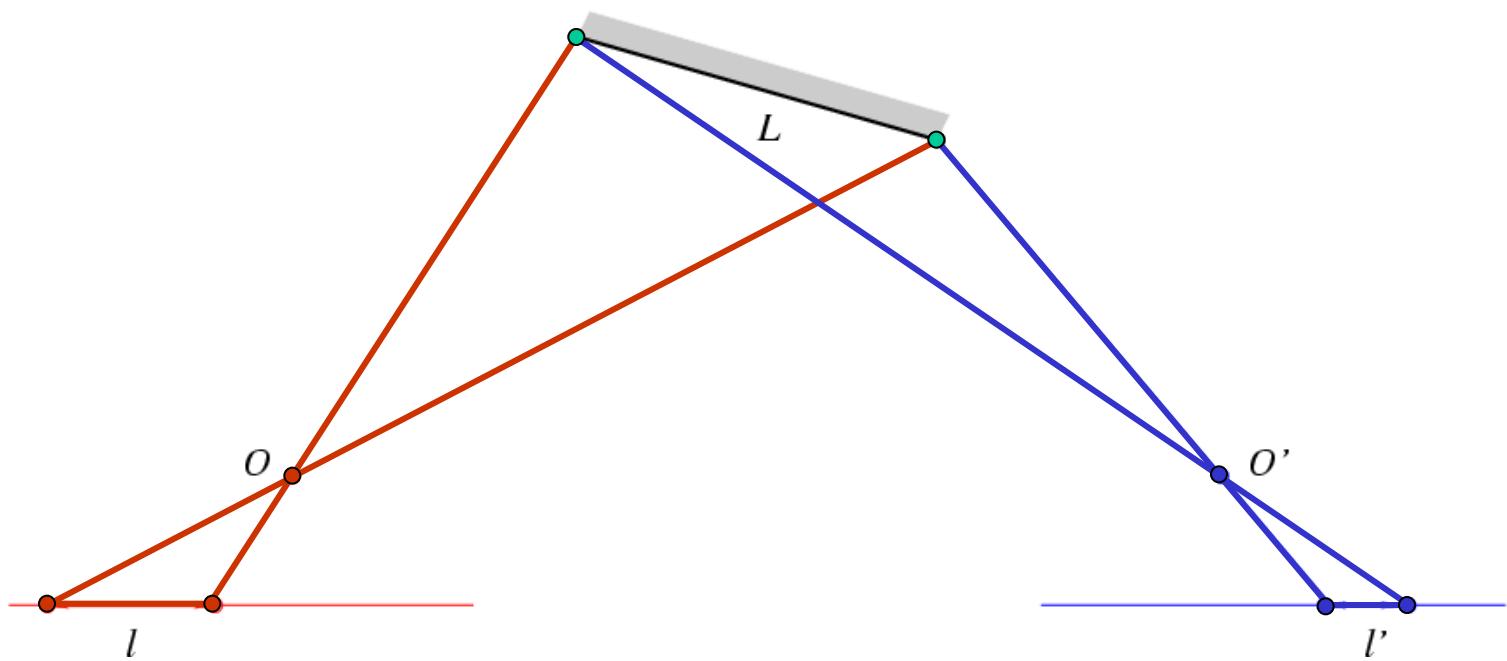
Correlation Methods (1970--)



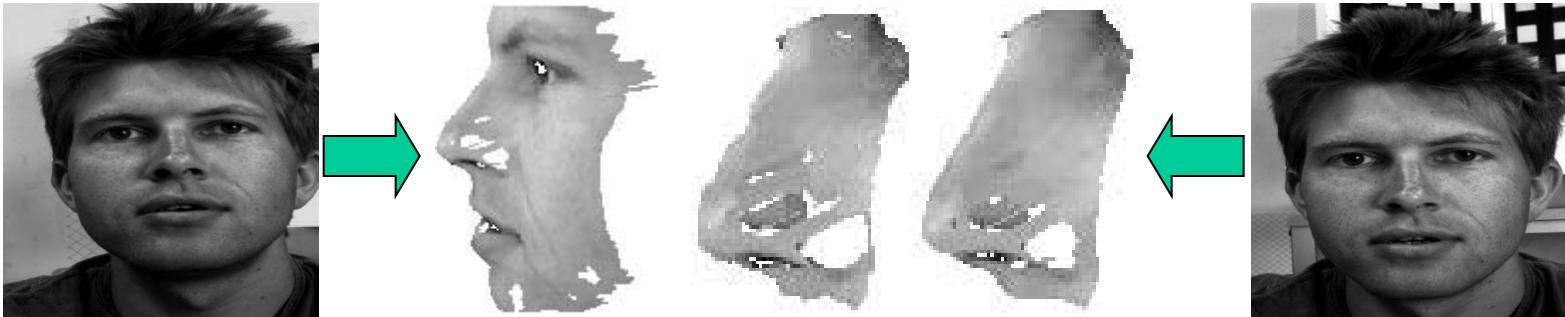
Slide the window along the epipolar line until $w \cdot w'$ is maximized.

Normalized Correlation: minimize θ instead. \leftrightarrow Minimize $|w - w'|^2$

Correlation Methods: Foreshortening Problems



Solution: add a second pass using disparity estimates to warp the correlation windows, e.g. Devernay and Faugeras (1994).

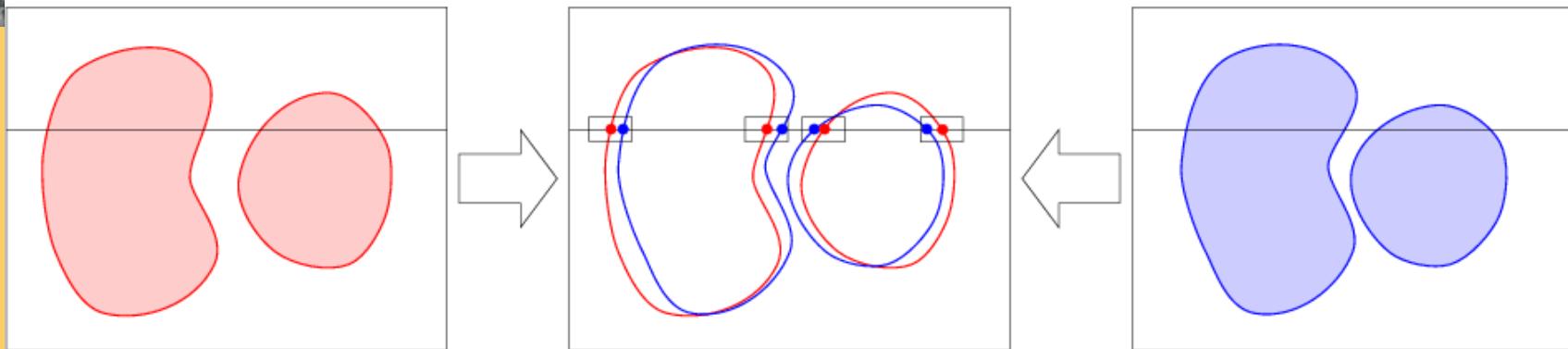


Reprinted from "Computing Differential Properties of 3D Shapes from Stereopsis without 3D Models," by F. Devernay and O. Faugeras, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1994). © 1994 IEEE.

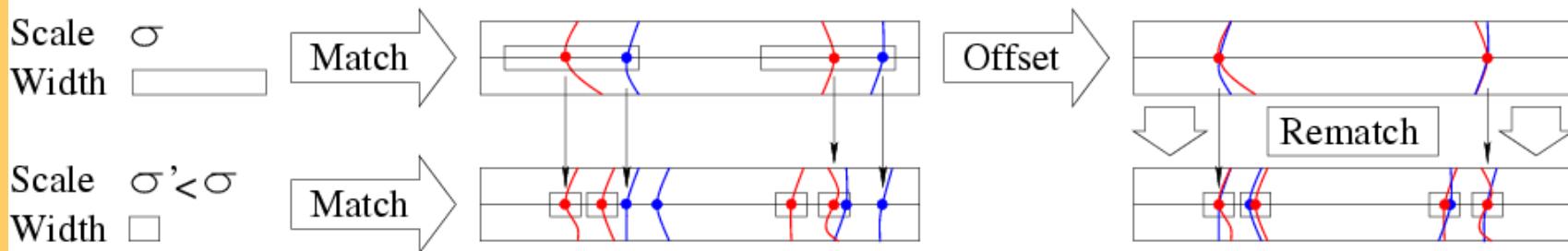


Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)

Matching zero-crossings at a single scale

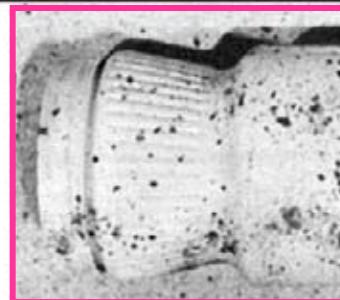


Matching zero-crossings at multiple scales



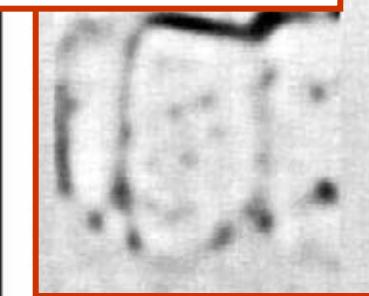
- Edges are found by repeatedly smoothing the image and detecting the zero crossings of the second derivative (Laplacian).
- Matches at coarse scales are used to offset the search for matches at fine scales (equivalent to eye movements).

Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)



One of the two input images

Image Laplacian

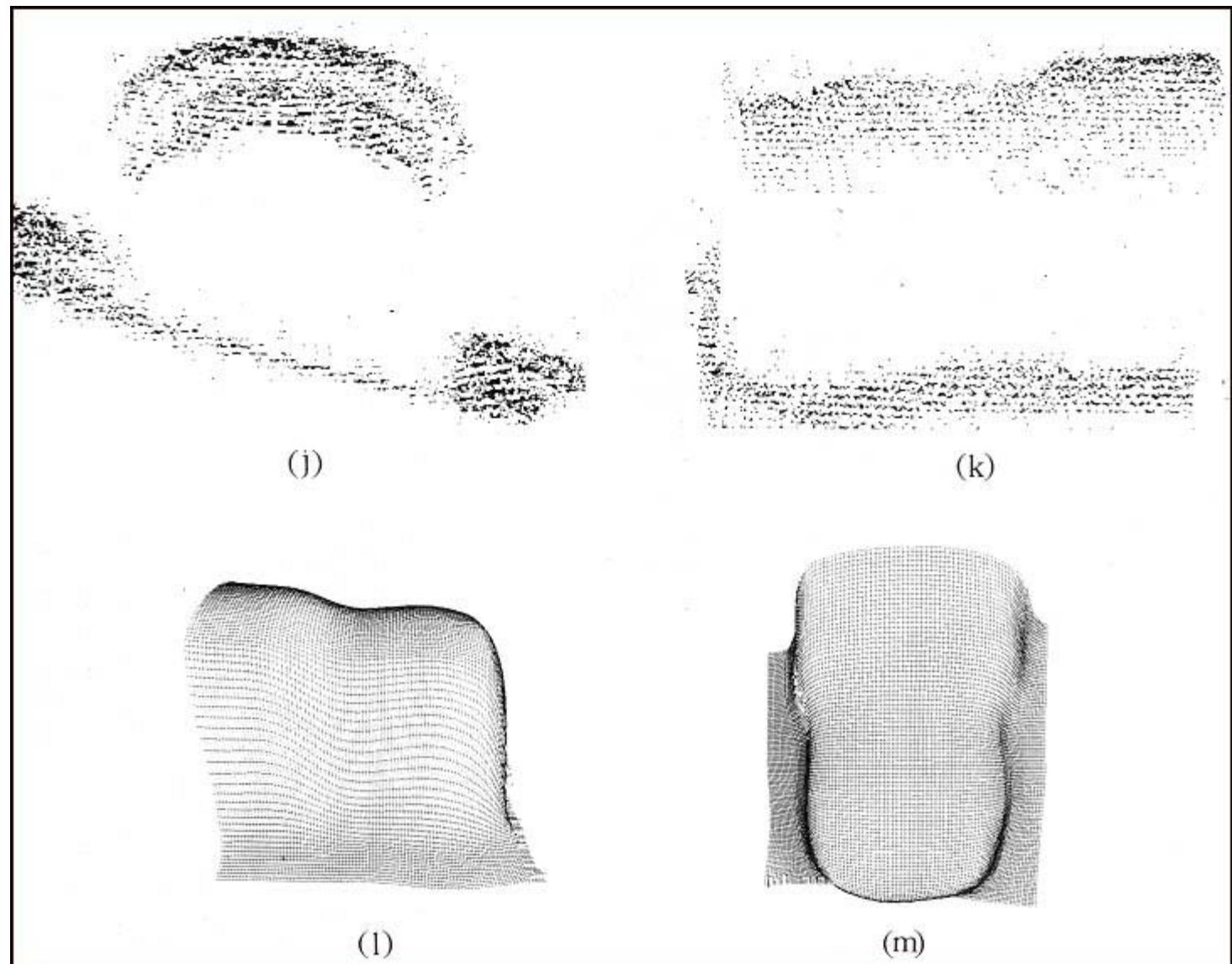


Zeros of the Laplacian





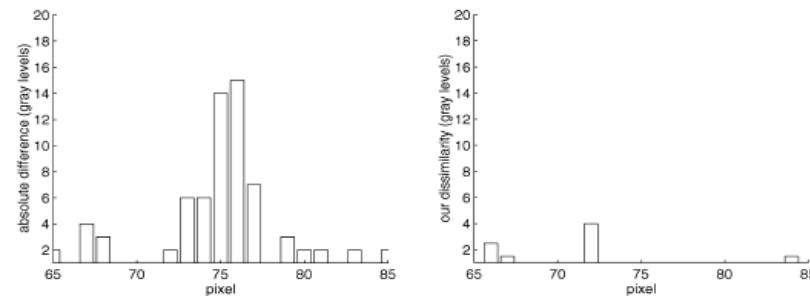
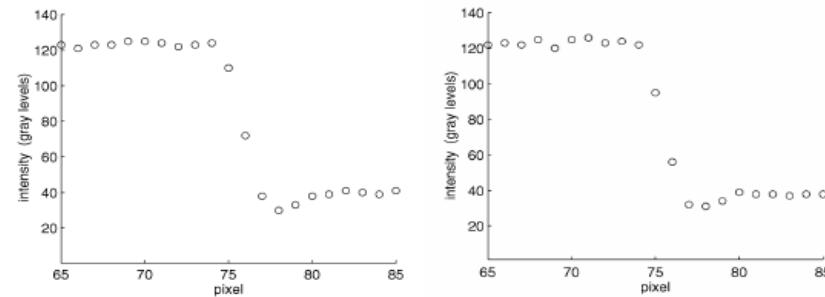
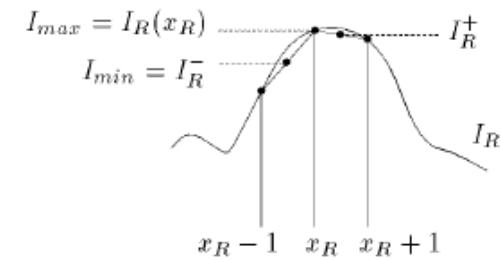
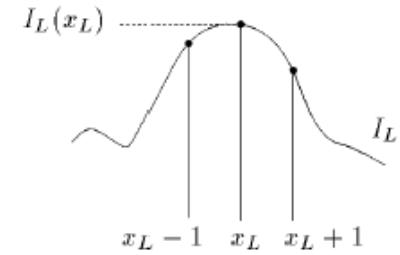
Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)





Pixel Dissimilarity

- Absolute difference of intensities
 $c = |I_1(x,y) - I_2(x-d,y)|$
- Interval matching [Birchfield]
 - Considers sensor integration
 - Represents pixels as intervals





Alternative Dissimilarity Measures

- Rank and Census transforms [Zabih ECCV94]
- Rank transform:
 - Define window containing R pixels around each pixel
 - Count the number of pixels with lower intensities than center pixel in the window
 - Replace intensity with rank (0..R-1)
 - Compute SAD on rank-transformed images
- Census transform:
 - Use bit string, defined by neighbors, instead of scalar rank
- Robust against illumination changes



Rank and Census Transform Results

- Noise free, random dot stereograms
- Different gain and bias

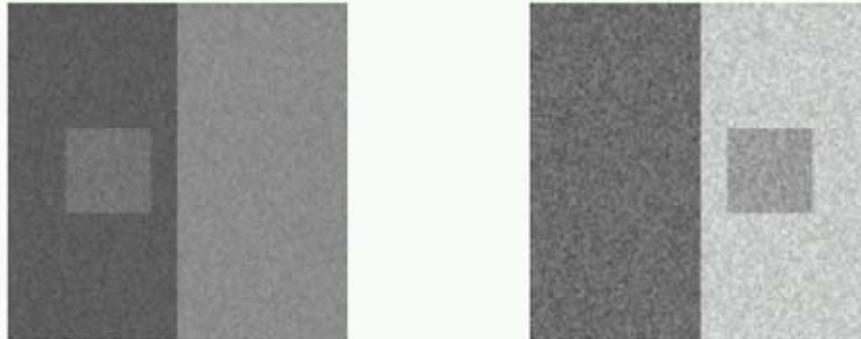


Fig. 2. Right and left random-dot stereograms

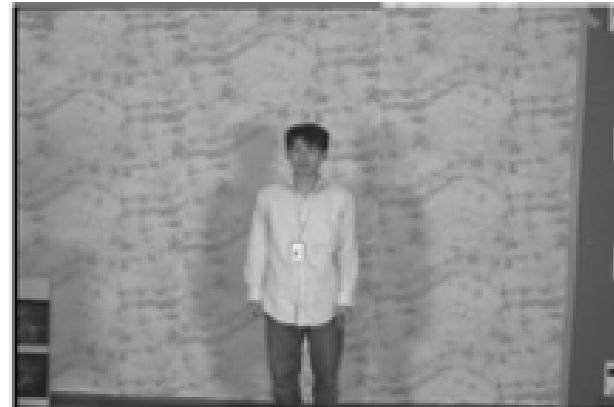


Fig. 3. Disparities from normalized correlation, rank and census transforms



Systematic Errors of Area-based Stereo

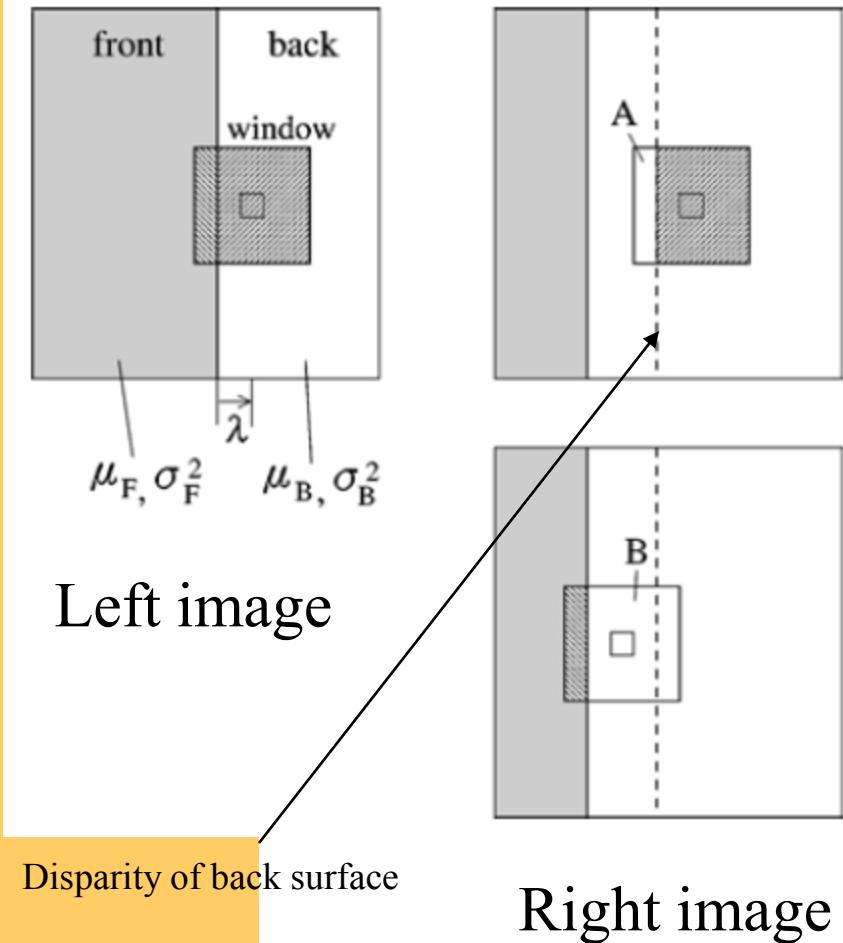
- Ambiguous matches in textureless regions
- []



[]



Surface Over-extension



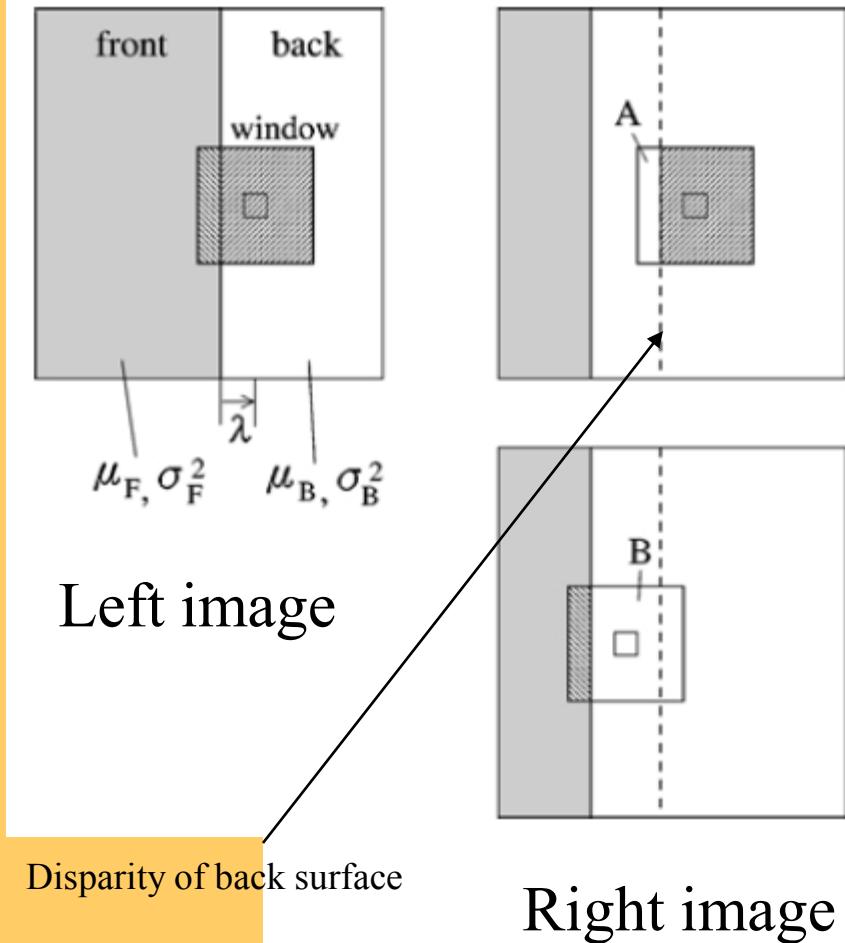
- Expected value of $E[(x-y)^2]$ for x in left and y in right image is:

Case A: $\sigma_F^2 + \sigma_B^2 + (\mu_F - \mu_B)^2$ for $w/2 - \lambda$ pixels in each row

Case B: $2 \sigma_B^2$ for $w/2 + \lambda$ pixels in each row



Surface Over-extension



- Discontinuity perpendicular to epipolar

$$\begin{aligned} & (\sigma_F^2 + \sigma_B^2 + (\mu_F - \mu_B)^2) \left(\frac{w}{2} - \lambda \right) w \\ &= 2\sigma_B^2 \left(\frac{w}{2} + \lambda \right) w. \end{aligned}$$

$$\lambda = \frac{\sigma_F^2 - \sigma_B^2 + (\mu_F - \mu_B)^2}{\sigma_F^2 + 3\sigma_B^2 + (\mu_F - \mu_B)^2} \frac{w}{2}.$$

- Discontinuity parallel to epipolar lines

$$\lambda = \frac{\sigma_F^2 - \sigma_B^2}{\sigma_F^2 + \sigma_B^2} \frac{w}{2}.$$



Over-extension and shrinkage

- Turns out that:

for discontinuities perpendicular to epipolar lines

$$-\frac{w}{2} \leq \lambda \leq \frac{w}{2}$$

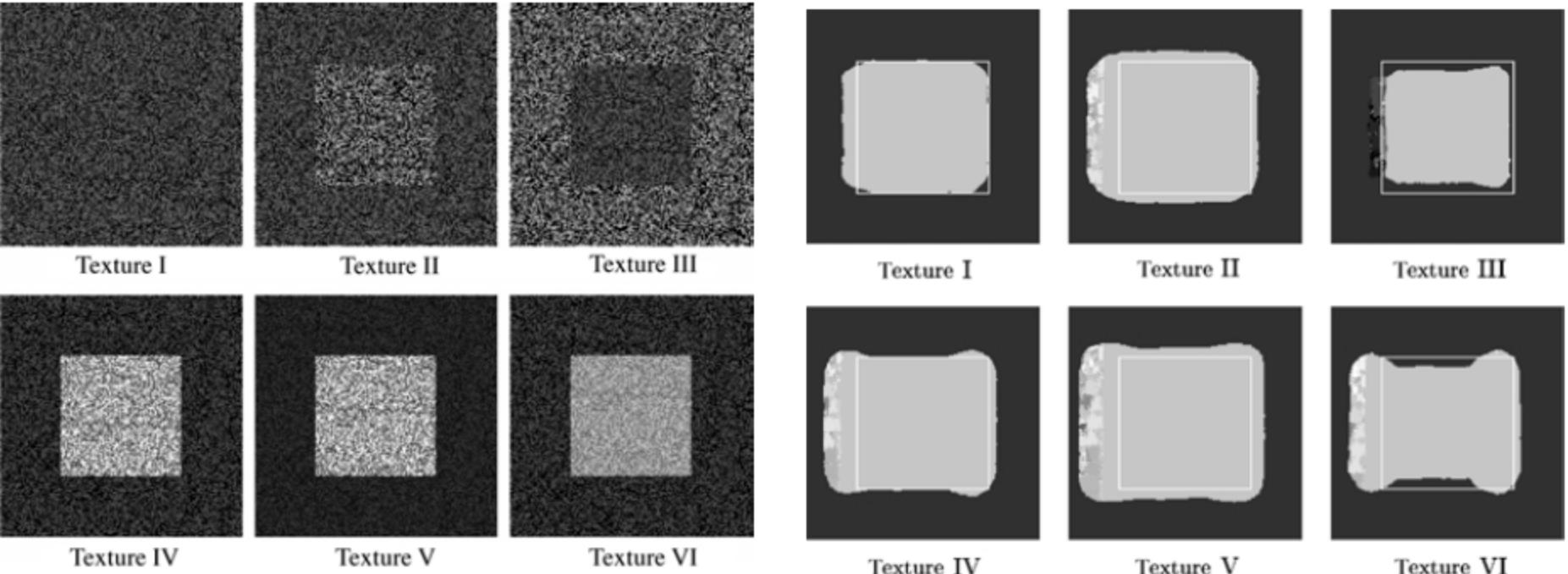
- And:

for discontinuities parallel to epipolar lines

$$-\frac{w}{2} \leq \lambda \leq \frac{w}{2}$$



Random Dot Stereogram Experiments

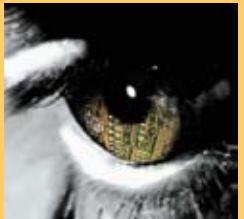


	Front surface		Back surface	
	μ_F	σ_F	μ_B	σ_B
I	100	50	100	50
II	100	100	100	50
III	100	50	100	100
IV	200	50	100	50
V	200	50	100	25
VI	200	25	100	50

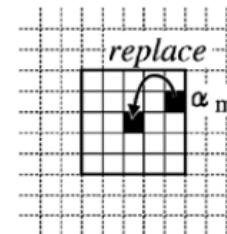
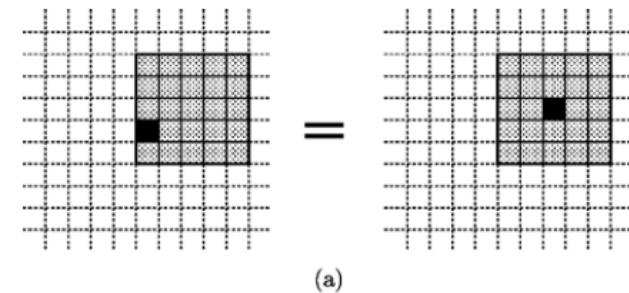
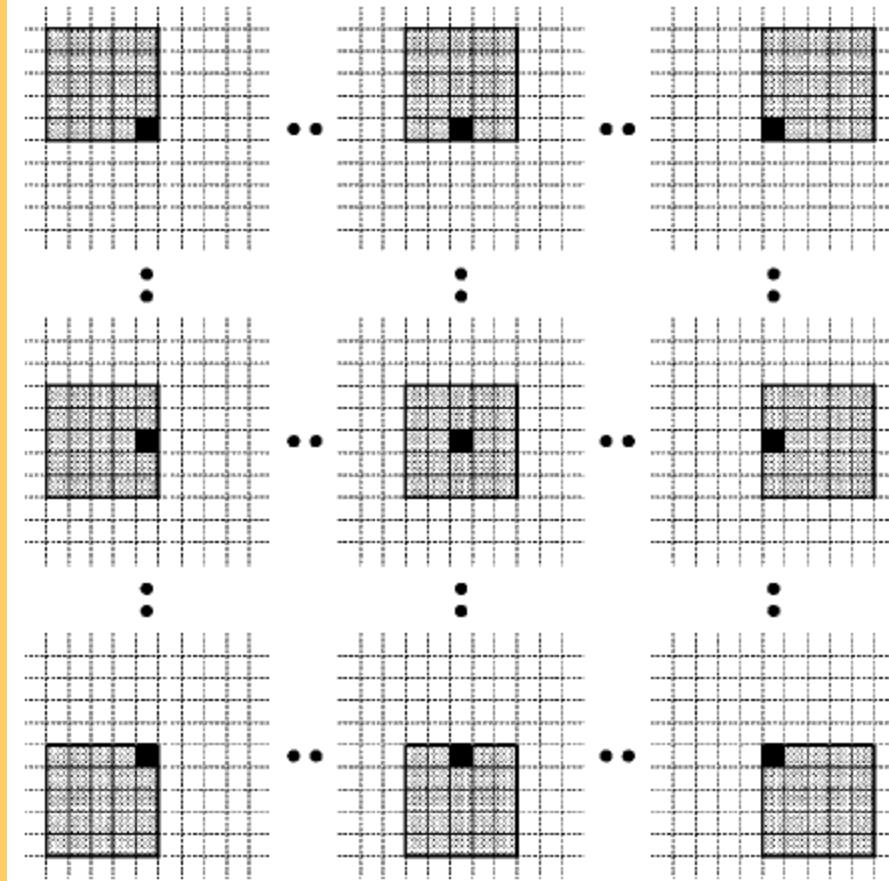


Random Dot Stereogram Experiments

w	Perpendicular		Parallel		Perpendicular		Parallel		Perpendicular		Parallel	
	Theoretical	Actual	Theoretical	Actual	Theoretical	Actual	Theoretical	Actual	Theoretical	Actual	Theoretical	Actual
Texture I				Texture II				Texture III				
7	0.00	0.03	0.00	0.05	1.50	1.53	2.10	2.08	-0.81	-0.69	-2.10	-1.87
11	0.00	0.08	0.00	0.13	2.36	2.35	3.30	3.32	-1.27	-1.04	-3.30	-3.09
17	0.00	0.09	0.00	0.13	3.64	3.75	5.10	5.29	-1.96	-2.00	-5.10	-5.00
25	0.00	0.12	0.00	0.42	5.36	5.20	7.50	7.73	-2.88	-3.00	-7.50	-7.53
35	0.00	0.40	0.00	-0.33	7.50	7.50	10.50	10.25	-4.04	-4.50	-10.50	-10.75
Texture IV				Texture V				Texture VI				
7	1.75	1.81	0.00	0.23	2.89	2.87	2.10	2.27	1.57	1.49	-2.10	-1.87
11	2.75	2.74	0.00	0.56	4.54	4.60	3.30	3.61	2.47	2.52	-3.30	-3.02
17	4.25	4.32	0.00	0.68	7.02	7.11	5.10	5.47	3.81	3.88	-5.10	-4.87
25	6.25	6.15	0.00	0.65	10.33	10.20	7.50	7.75	5.50	5.80	-7.50	-7.43
35	8.75	9.00	0.00	0.90	14.46	14.45	10.50	10.80	7.84	8.00	-10.50	-10.35



Offset Windows



- ❑ Equivalent to using min nearby cost
- ❑ Result: loss of depth accuracy



Discontinuity Detection

- Use offset windows only where appropriate
 - Bi-modal distribution of SSD
 - Pixel of interest different than mode within window





Compact Windows

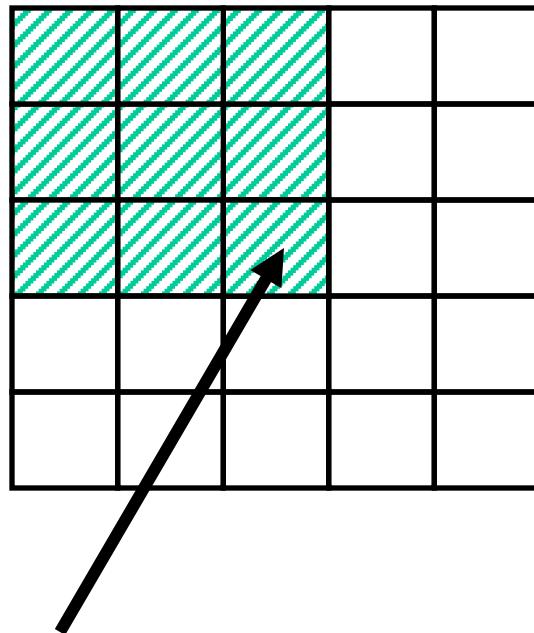
- [Veksler CVPR03]: Adapt windows size based on:
 - Average matching error per pixel
 - Variance of matching error
 - Window size (to bias towards larger windows)

$$C_d(W) = \bar{e} + \alpha \cdot var(e) + \frac{\beta}{\sqrt{W} + \gamma}.$$

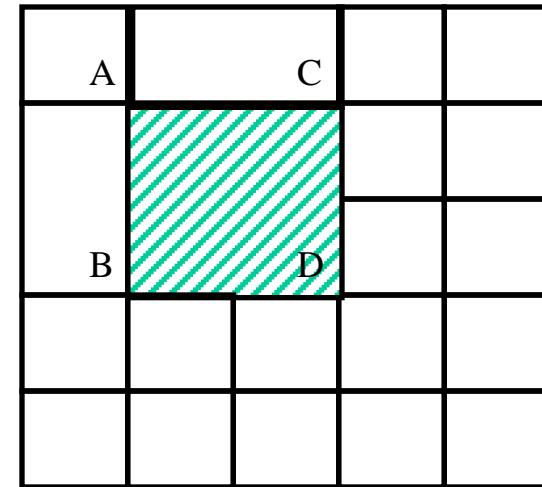
- Pick window that minimizes cost



Integral Image



Sum of shaded part



Shaded area = $A + D - B - C$
Independent of size

Compute an integral image for pixel dissimilarity at each possible disparity



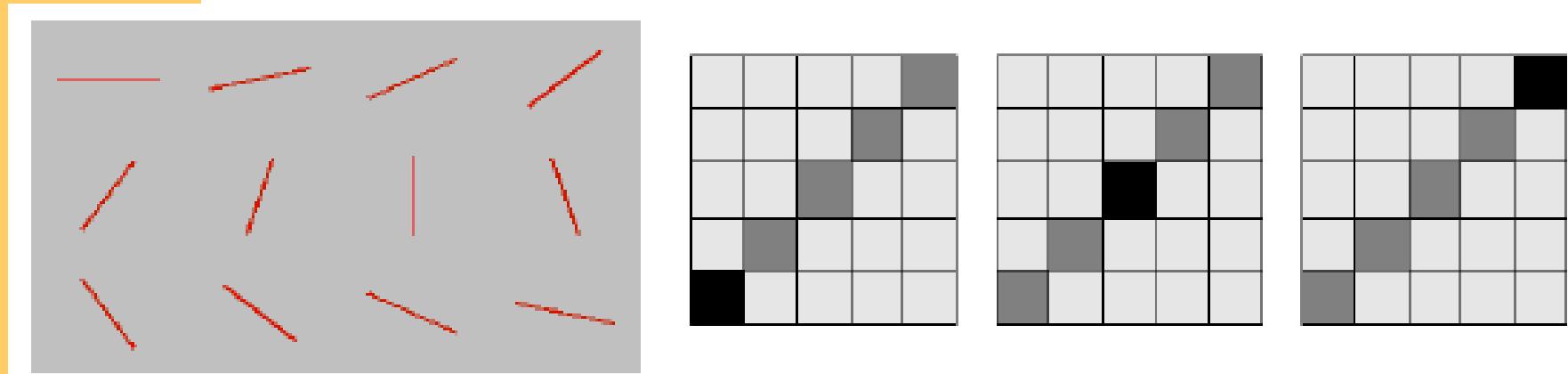
Results using Compact Windows





Rod-shaped filters

- Instead of square windows aggregate cost in rod-shaped shiftable windows [Kim CVPR05]
- Search for one that minimizes the cost (assume that it is an iso-disparity curve)
- Typically use 36 orientations





Locally Adaptive Support

Apply weights to contributions of neighboring pixels according to similarity and proximity
[Yoon CVPR05]



(a) left support win- (b) right support win- (c) color difference
dow dow between (a) and (b)



Locally Adaptive Support

- Similarity in CIE Lab color space:

$$\Delta c_{pq} = \sqrt{(L_p - L_q)^2 + (a_p - a_q)^2 + (b_p - b_q)^2}$$

- Proximity: Euclidean distance
- Weights:

$$w(p, q) = k \cdot \exp \left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p} \right) \right)$$



Locally Adaptive Support: Results



(a) left image



(b) ground truth



(c) shiftable win. [7]



(d) compact win. [3]



(e) variable win. [4]



(f) Bay. diff. [19]



Locally Adaptive Support



(g) bad pixels (error > 1)

Locally Adaptive Support: Results



(a) left image



(b) ground truth



(c) our result

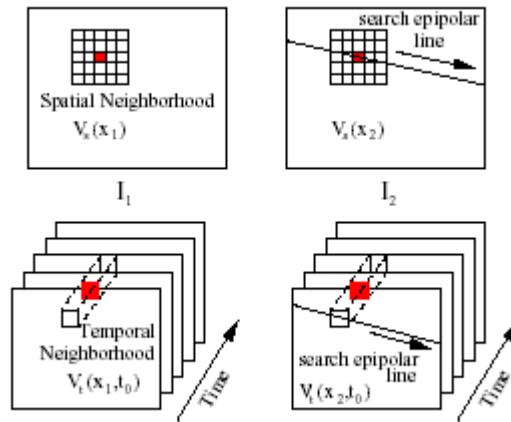


(d) bad pixels ($\text{error} > 1$)



Cool ideas

- Space-time stereo
(varying illumination, not shape)





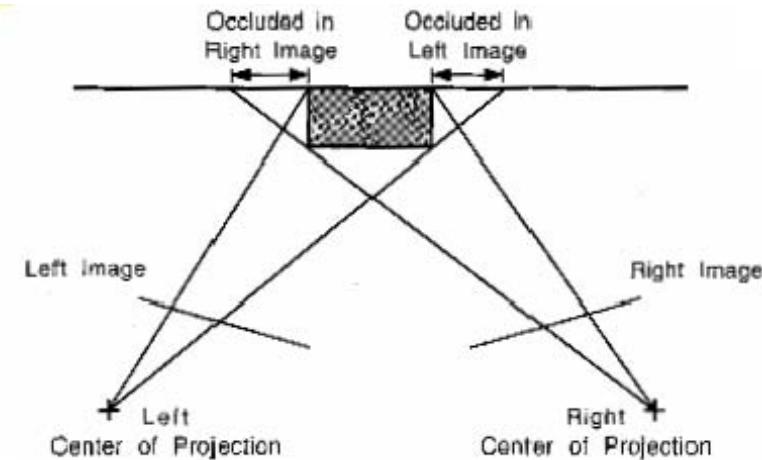
Challenges

- Ill-posed inverse problem
 - Recover 3-D structure from 2-D information
- Difficulties
 - Uniform regions
 - Half-occluded pixels

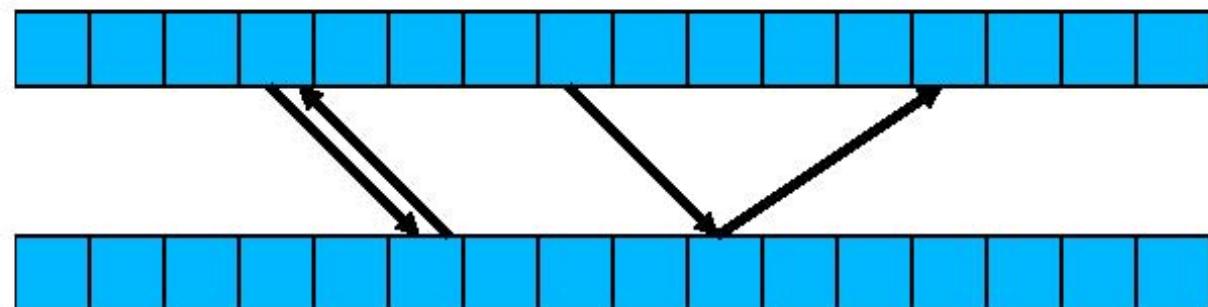




Occlusions

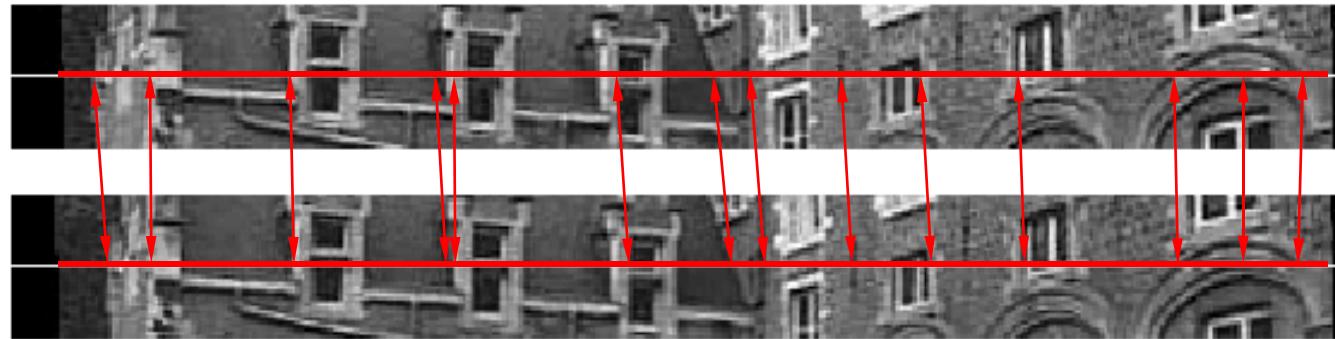


→ Consistency test



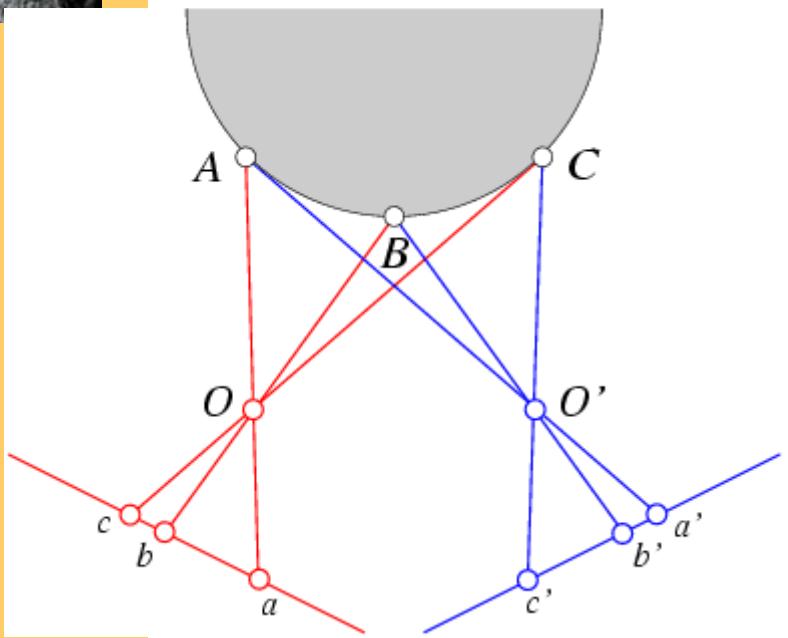


Exploiting scene constraints





The Ordering Constraint



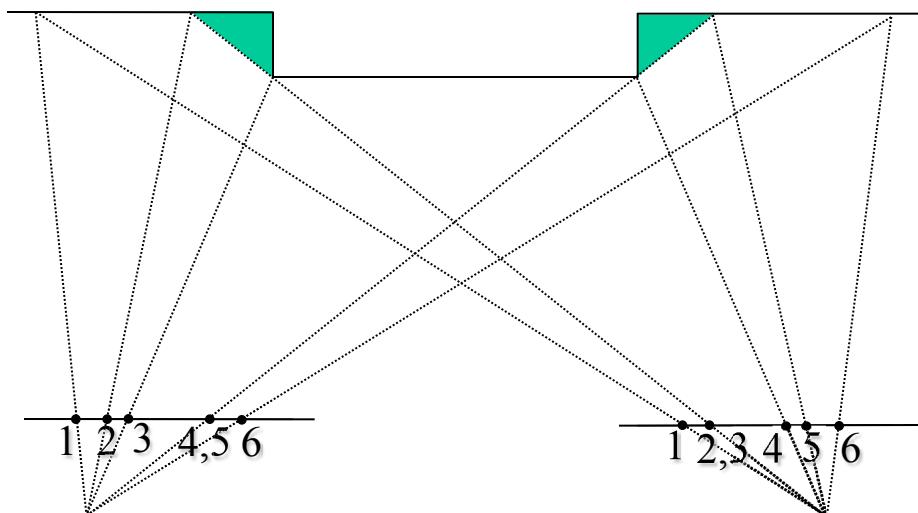
In general the points
are in the same order
on both epipolar lines.

But it is not always the case..

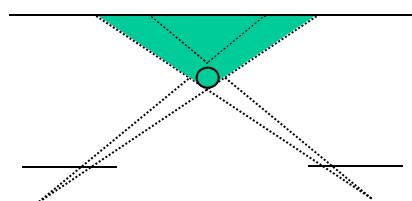
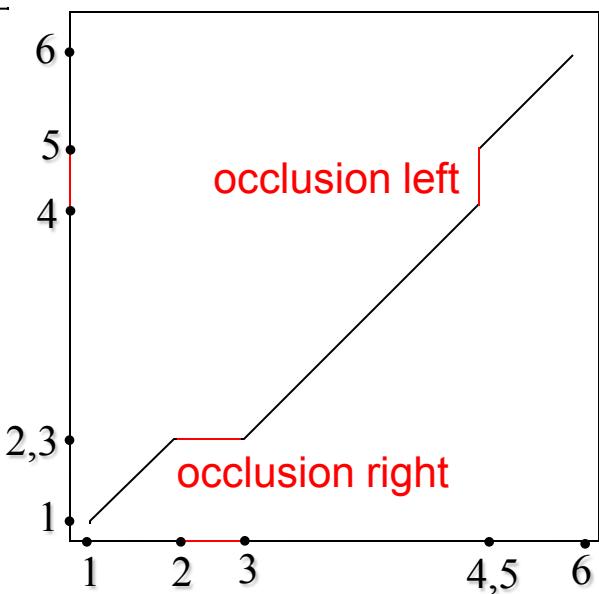


Ordering constraint

surface slice



surface as a path





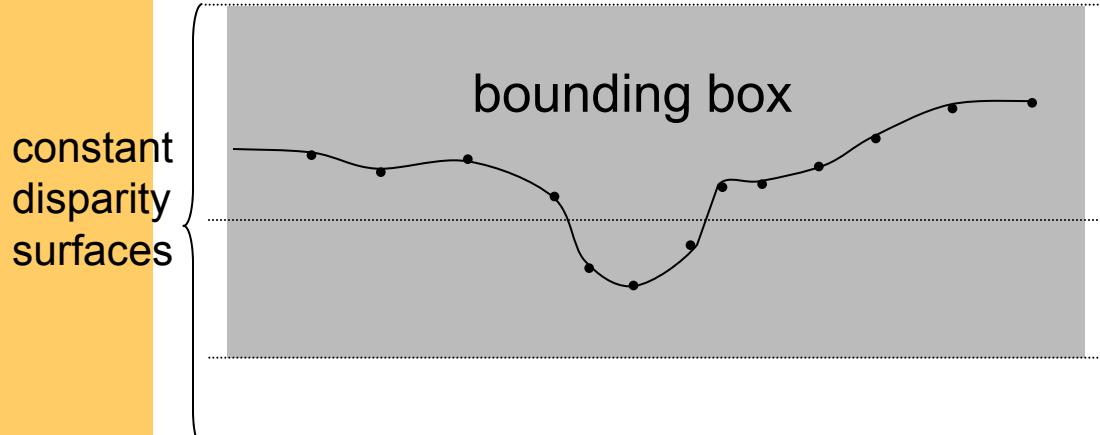
Uniqueness constraint

- In an image pair each pixel has at most one corresponding pixel
 - In general one corresponding pixel
 - In case of occlusion there is none

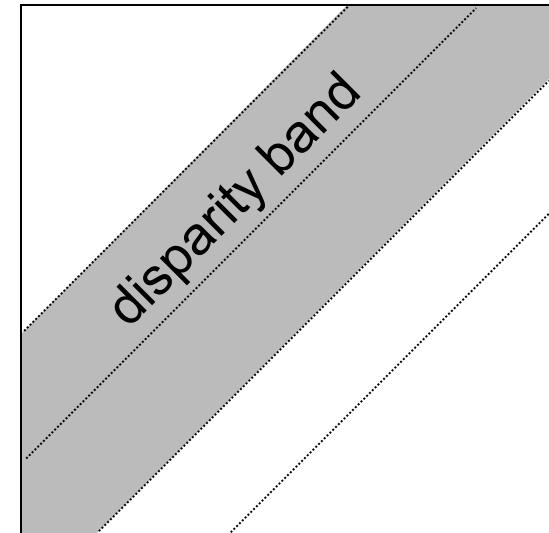


Disparity constraint

surface slice



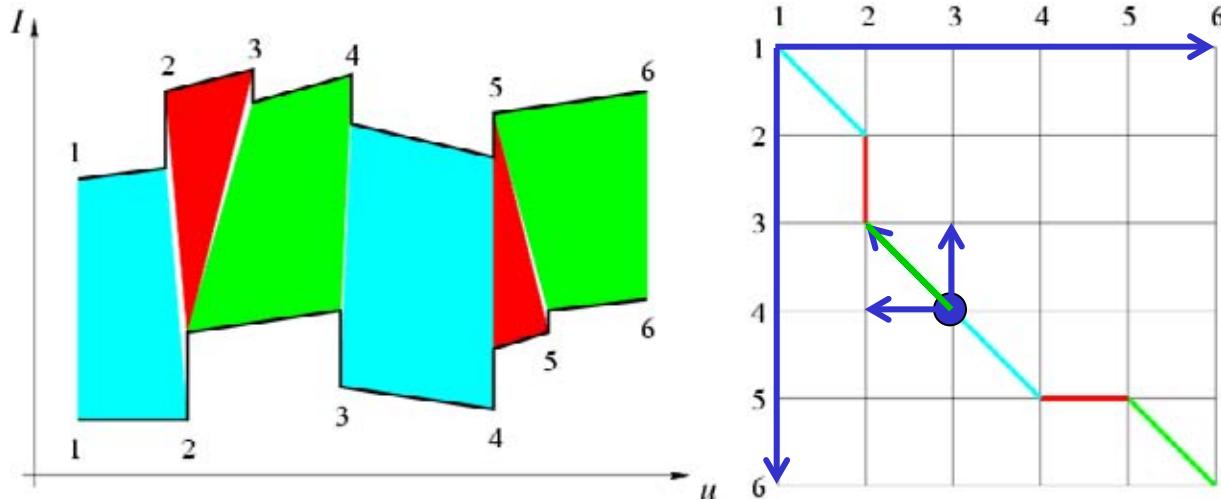
surface as a path



use reconstructed features
to determine bounding box



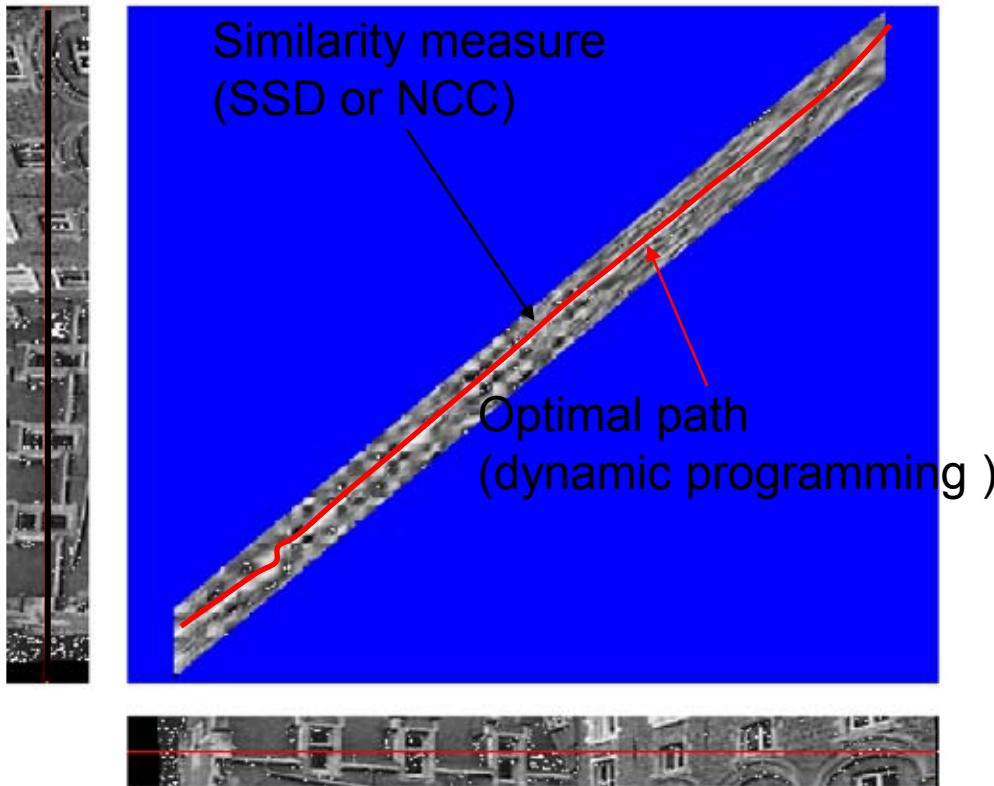
Dynamic Programming (Baker and Binford, 1981)



```
% Loop over all nodes  $(k, l)$  in ascending order.  
for  $k = 1$  to  $m$  do  
    for  $l = 1$  to  $n$  do  
        % Initialize optimal cost  $C(k, l)$  and backward pointer  $B(k, l)$ .  
         $C(k, l) \leftarrow +\infty$ ;  $B(k, l) \leftarrow \text{nil}$ ;  
        % Loop over all inferior neighbors  $(i, j)$  of  $(k, l)$ .  
        for  $(i, j) \in \text{Inferior - Neighbors}(k, l)$  do  
            % Compute new path cost and update backward pointer if necessary.  
             $d \leftarrow C(i, j) + \text{Arc - Cost}(i, j, k, l)$ ;  
            if  $d < C(k, l)$  then  $C(k, l) \leftarrow d$ ;  $B(k, l) \leftarrow (i, j)$  endif;  
            endfor;  
        endfor;  
    endfor;  
% Construct optimal path by following backward pointers from  $(m, n)$ .  
 $P \leftarrow \{(m, n)\}; (i, j) \leftarrow (m, n)$ ;  
while  $B(i, j) \neq \text{nil}$  do  $(i, j) \leftarrow B(i, j)$ ;  $P \leftarrow \{(i, j)\} \cup P$  endwhile.
```



Stereo matching



Constraints

- epipolar
- ordering
- uniqueness
- disparity limit
- disparity gradient limit

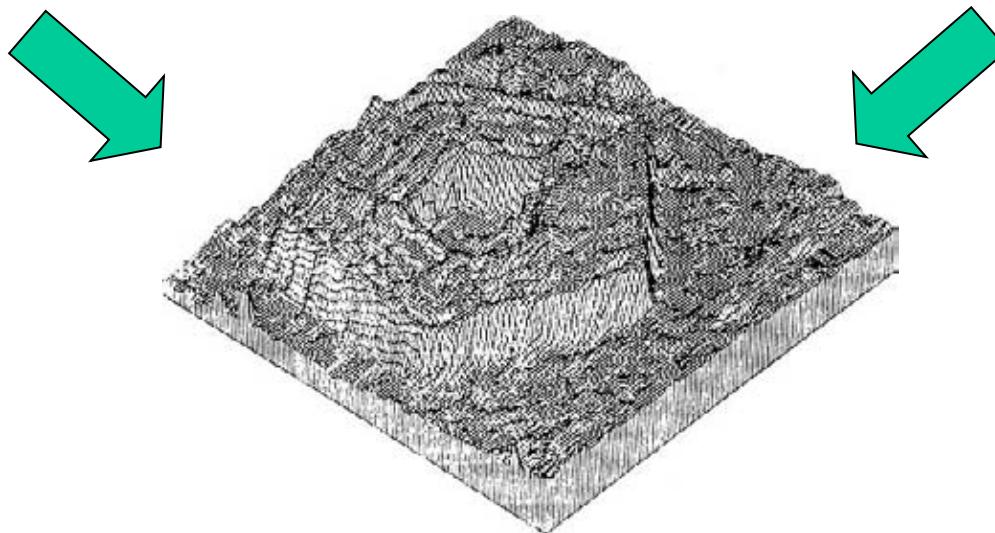
Trade-off

- Matching cost (data)
- Discontinuities (prior)

(Cox et al. CVGIP'96; Koch'96; Falkenhagen '97;
Van Meerbergen, Vergauwen, Pollefeys, VanGool IJCV'02)



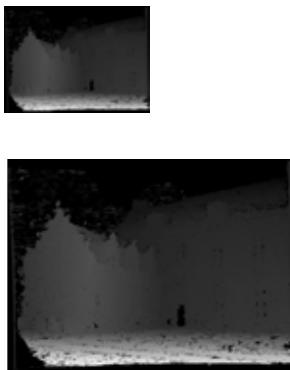
Dynamic Programming (Ohta and Kanade, 1985)





Hierarchical stereo matching

Downsampling
(Gaussian pyramid)



Disparity propagation

Allows faster computation

Deals with large disparity ranges



Disparity map

image $I(x,y)$



Disparity map $D(x,y)$

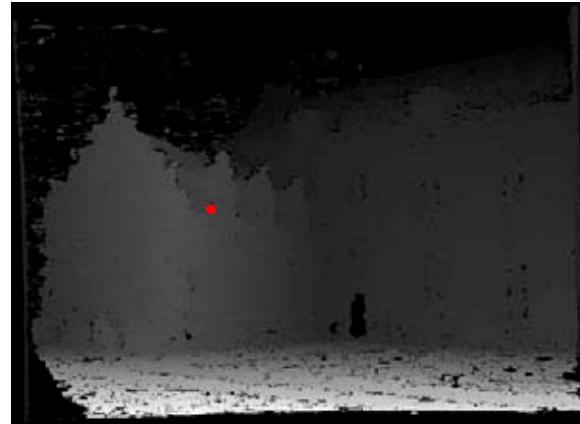


image $I'(x',y')$



$$(x', y') = (x + D(x, y), y)$$



Example: reconstruct image from neighboring images

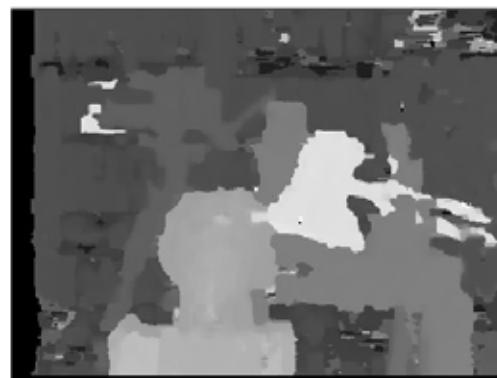




True disparities



*2 – Dynamic progr.

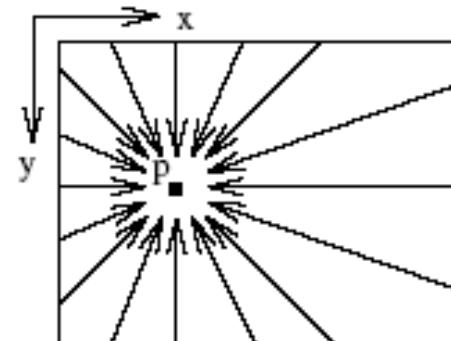


16 – Fast Correlation



Semi-global optimization

- Optimize: $E = E_{\text{data}} + E(|D_p - D_q| = 1) + E(|D_p - D_q| > 1)$ [Hirschmüller CVPR05]
 - Use mutual information as cost
- NP-hard using graph cuts or belief propagation (2-D optimization)
- Instead do dynamic programming along many directions
 - Don't use visibility or ordering constraints
 - Enforce uniqueness
 - Add costs



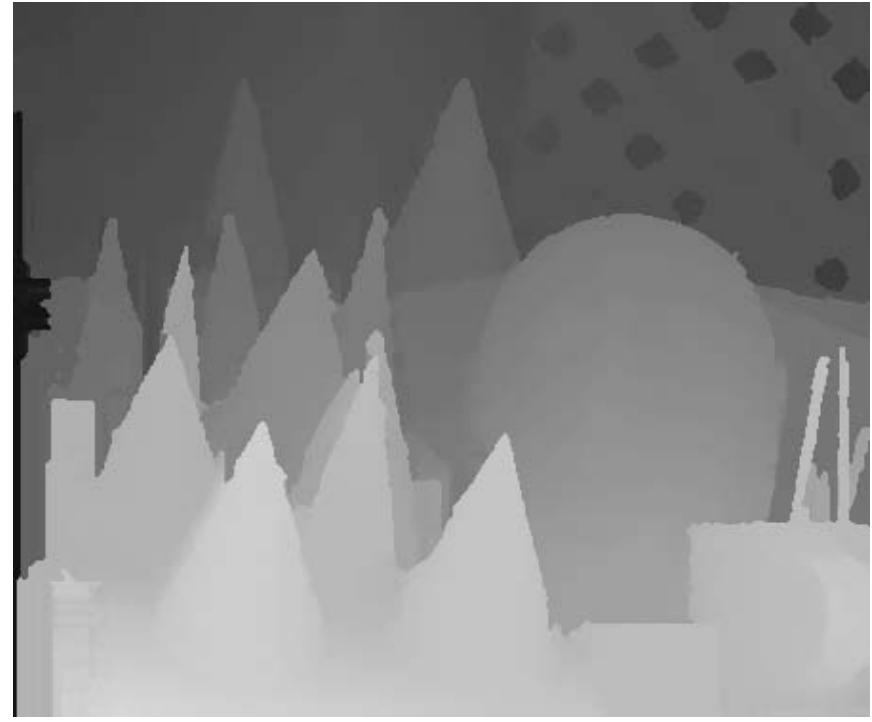


Results of Semi-global optimization





Results of Semi-global optimization

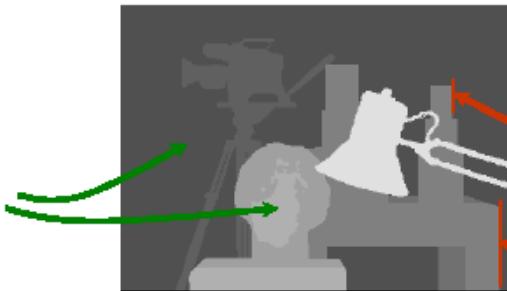


No. 1 overall in Middlebury evaluation
(at 0.5 error threshold as of Sep. 2006)



Energy minimization

Disparity
continuous
in most
places,



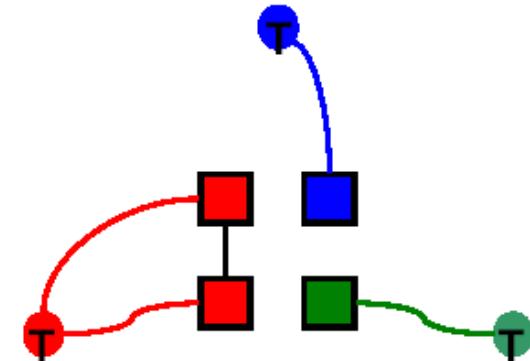
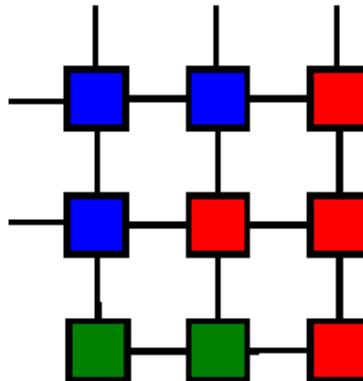
except at
depth
discontinuities

1. Matching pixels should have similar intensities.
2. Most nearby pixels should have similar disparities

$$\begin{aligned} \rightarrow \text{Minimize} \quad & \sum [I_1(x + D(x, y), y) - I_2(x, y)]^2 \\ & + \lambda \sum [D(x + 1, y) - D(x, y)]^2 \\ & + \mu \sum [D(x, y + 1) - D(x, y)]^2 \end{aligned}$$



Graph Cut

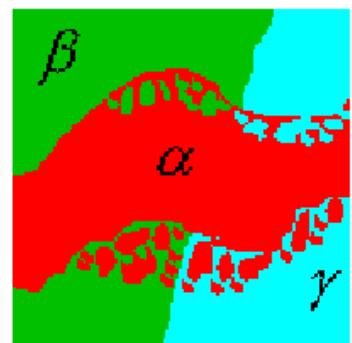
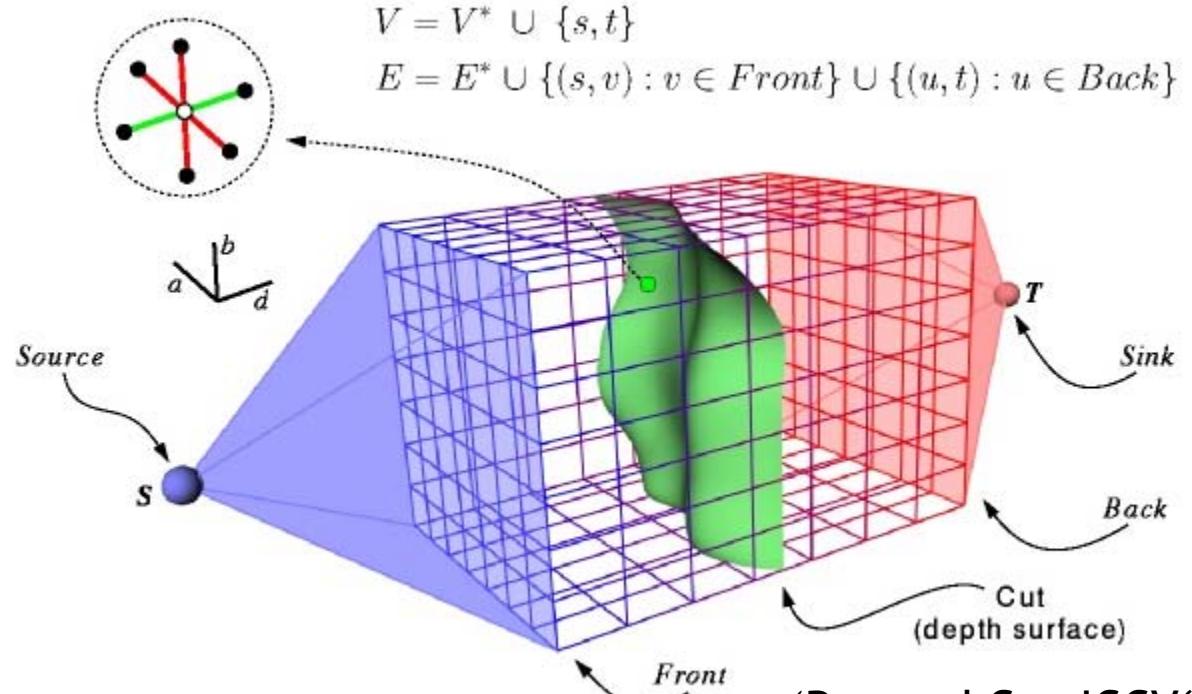


1. Stereo is a labeling problem
2. Graph cut corresponds to a labeling.
→ **Assign edge weights cleverly so that the min-weight cut gives the minimum energy!**

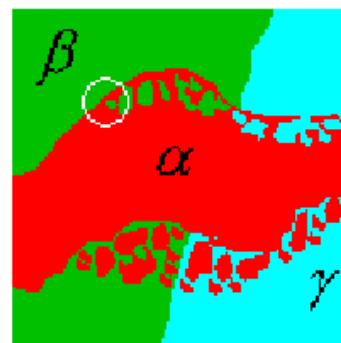
(general formulation requires multi-way cut!)



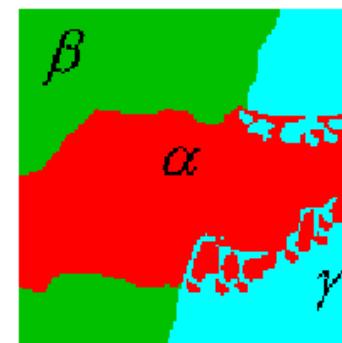
Simplified graph cut



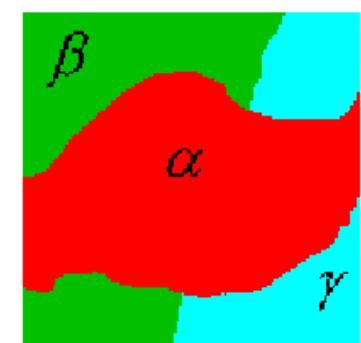
(a) initial labeling



(b) standard move



(c) α - β -swap



(d) α -expansion
(Boykov et al ICCV'99)



Belief Propagation

Belief of one node about another gets propagated through messages (full pdf, not just most likely state)

first iteration

per pixel



+left



+right



+up



+down



subsequent iterations

2



3



4



5



...

20



(adapted from J. Coughlan slides)



True disparities



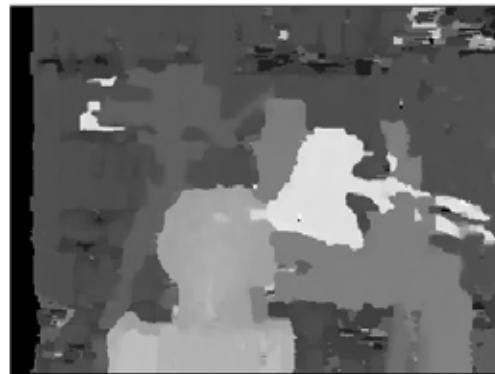
11 – GC + occlusions



19 – Belief propagation



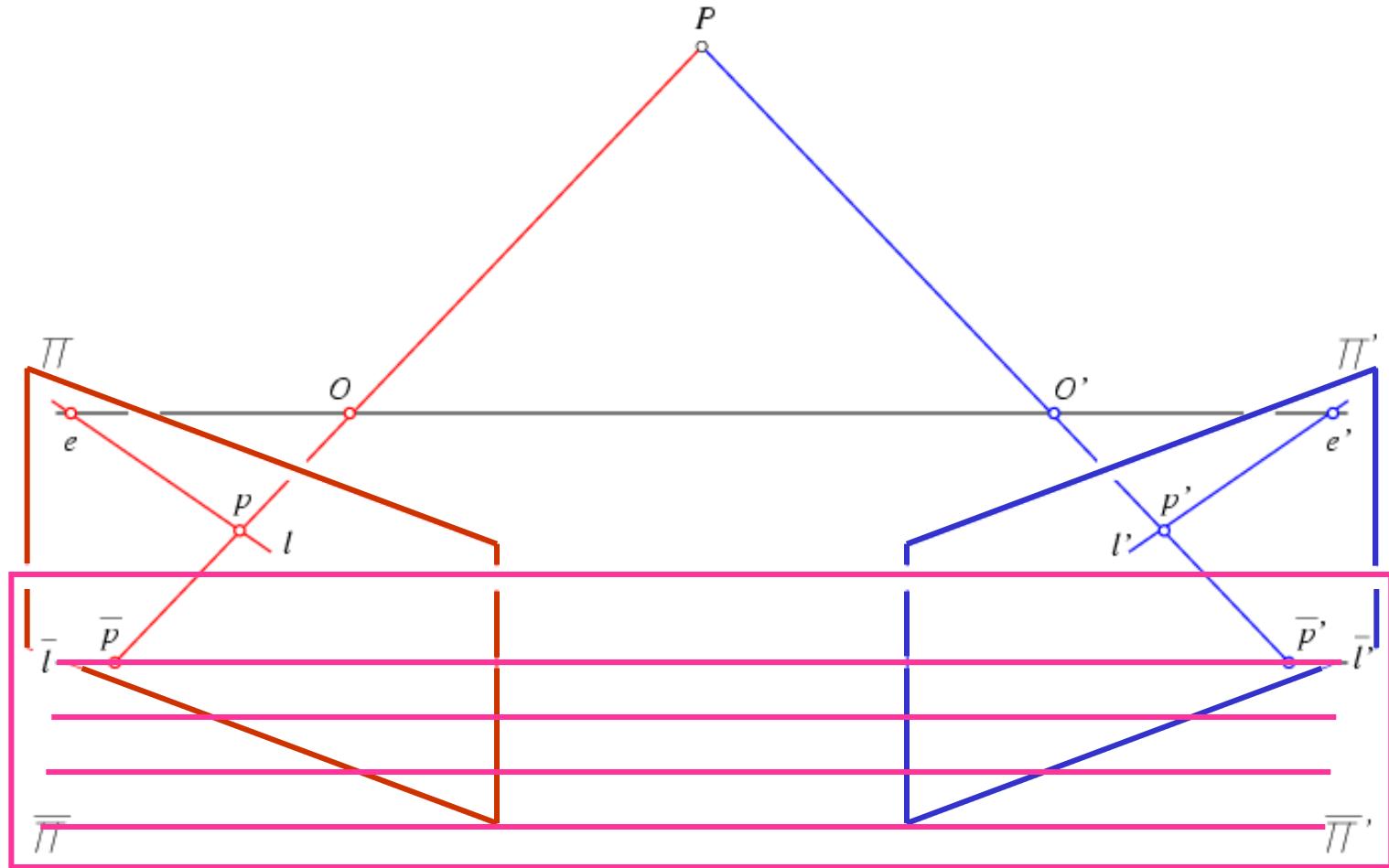
*2 – Dynamic progr.



16 – Fast Correlation



Rectification



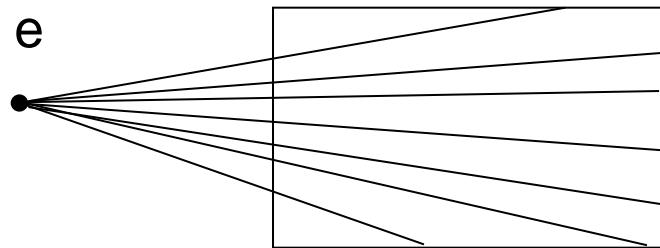
All epipolar lines are parallel in the rectified image plane.



Image pair rectification

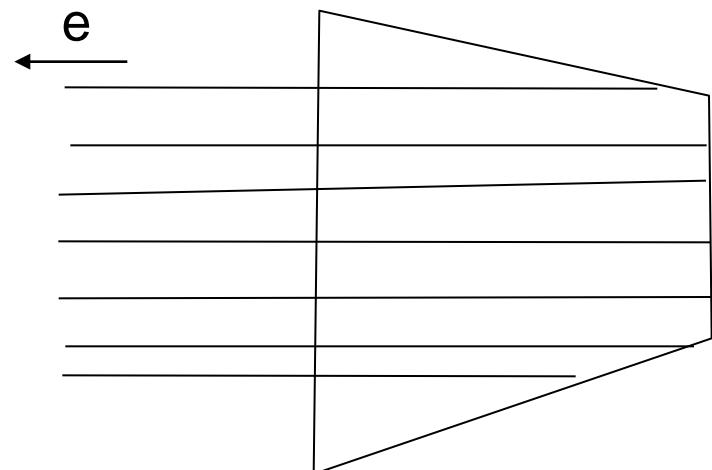
simplify stereo matching
by warping the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines



$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = H\mathbf{e}$$

map epipole \mathbf{e} to $(1,0,0)$
try to minimize image distortion

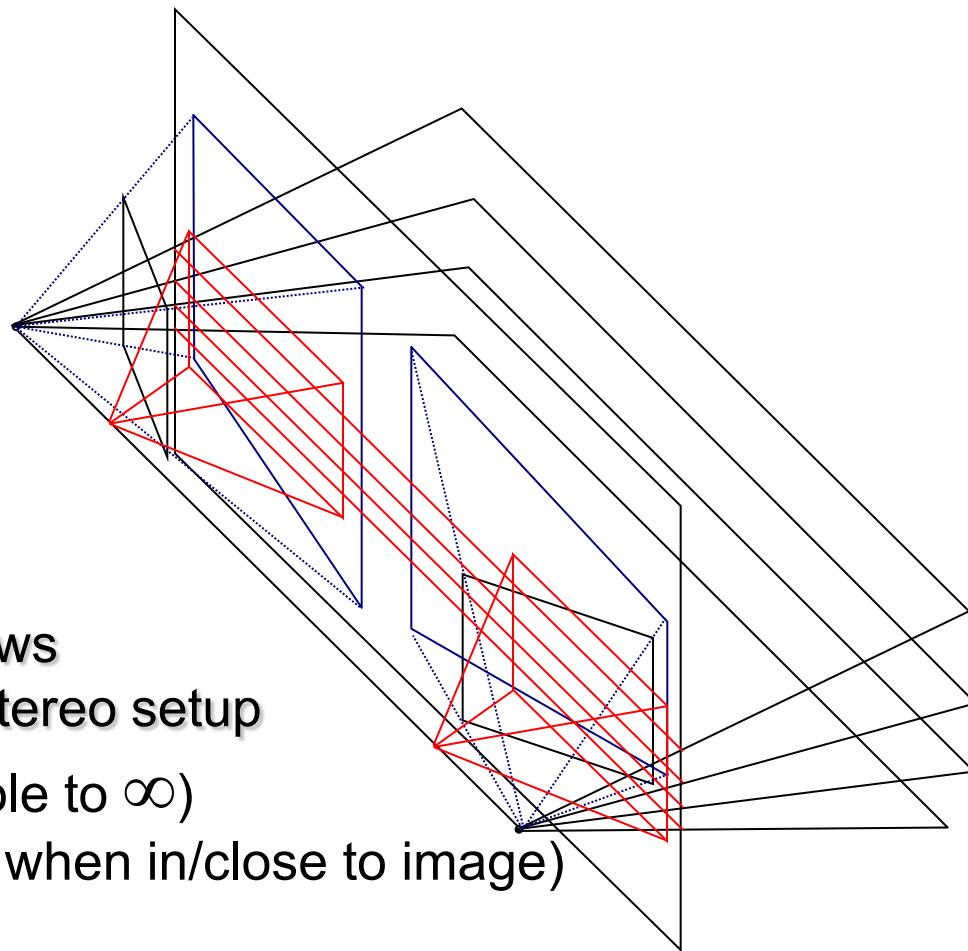


problem when epipole in (or close to) the image

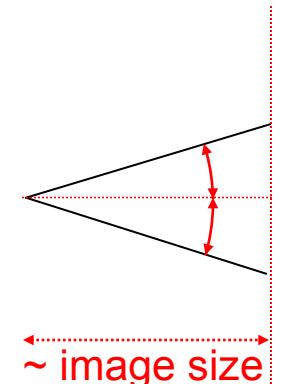


Planar rectification

(standard approach)

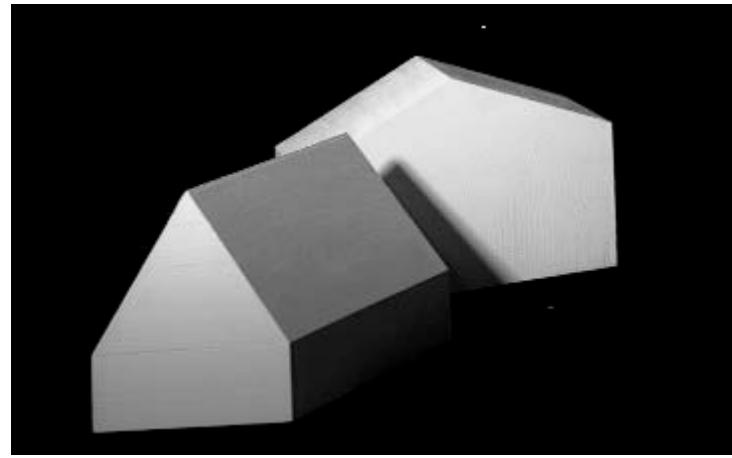
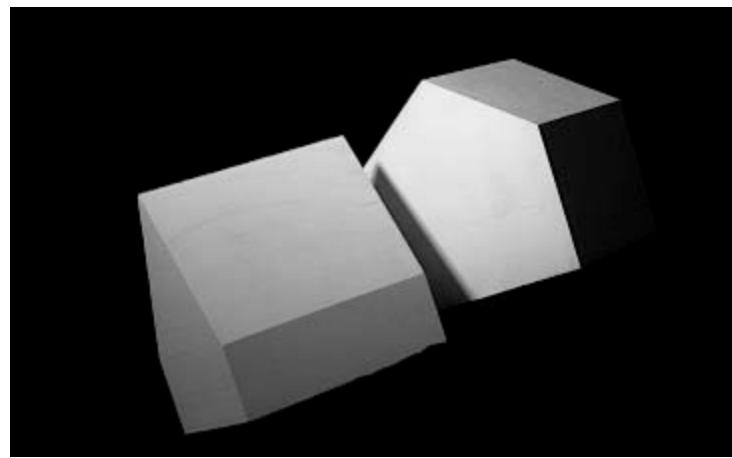
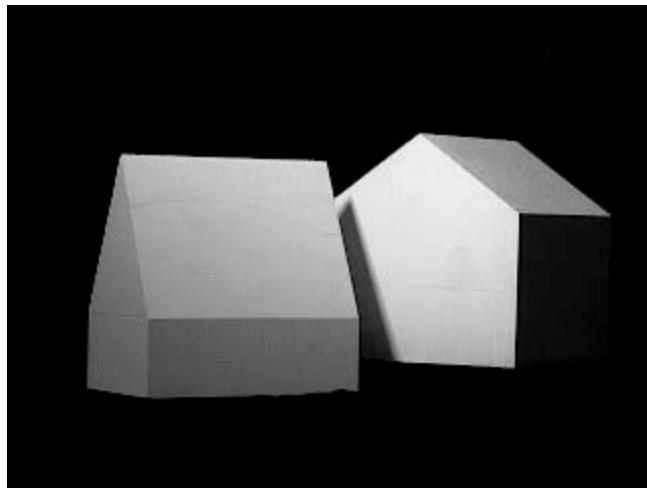


Bring two views
to standard stereo setup
(moves epipole to ∞)
(not possible when in/close to image)



(calibrated)

Distortion minimization
(uncalibrated)



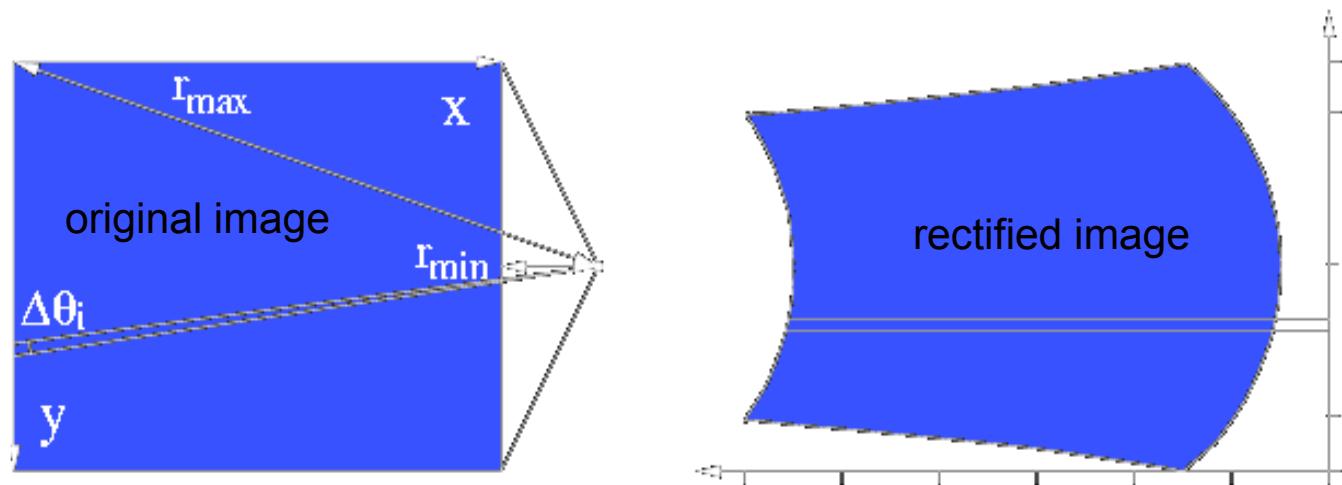




Polar rectification

(Pollefeys et al. ICCV'99)

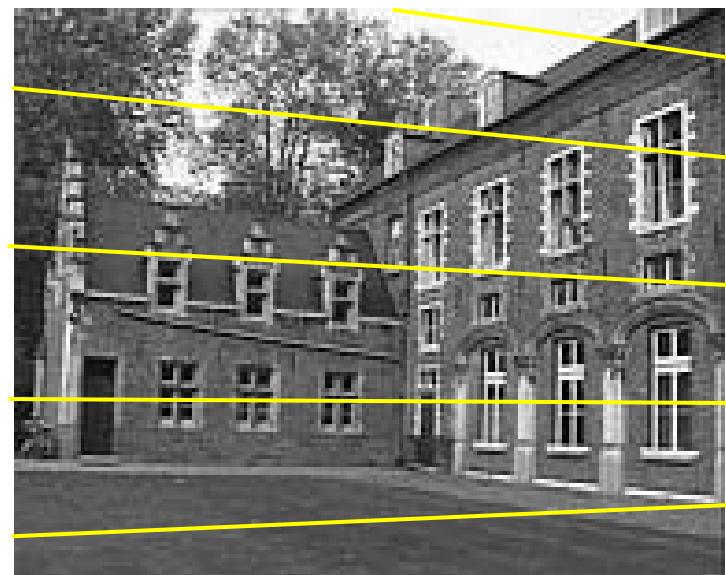
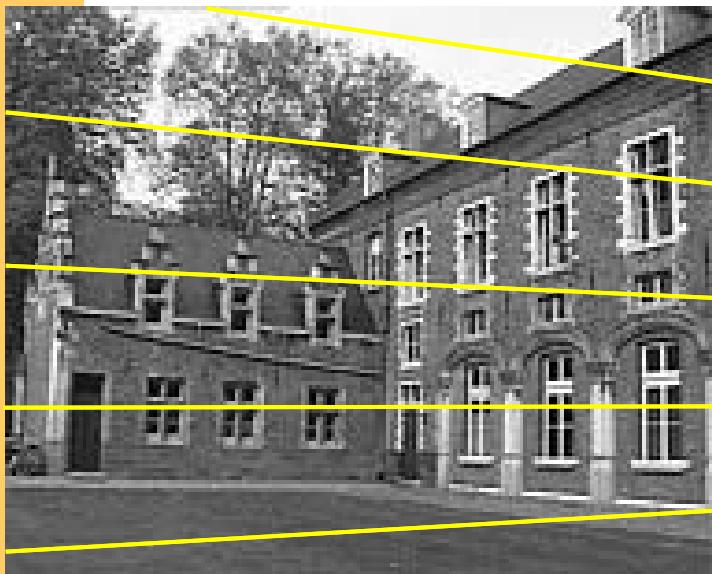
- Polar re-parameterization around epipoles
- Requires only (oriented) epipolar geometry
- Preserve length of epipolar lines
- Choose $\Delta\theta$ so that no pixels are compressed



Works for all relative motions
Guarantees minimal image size

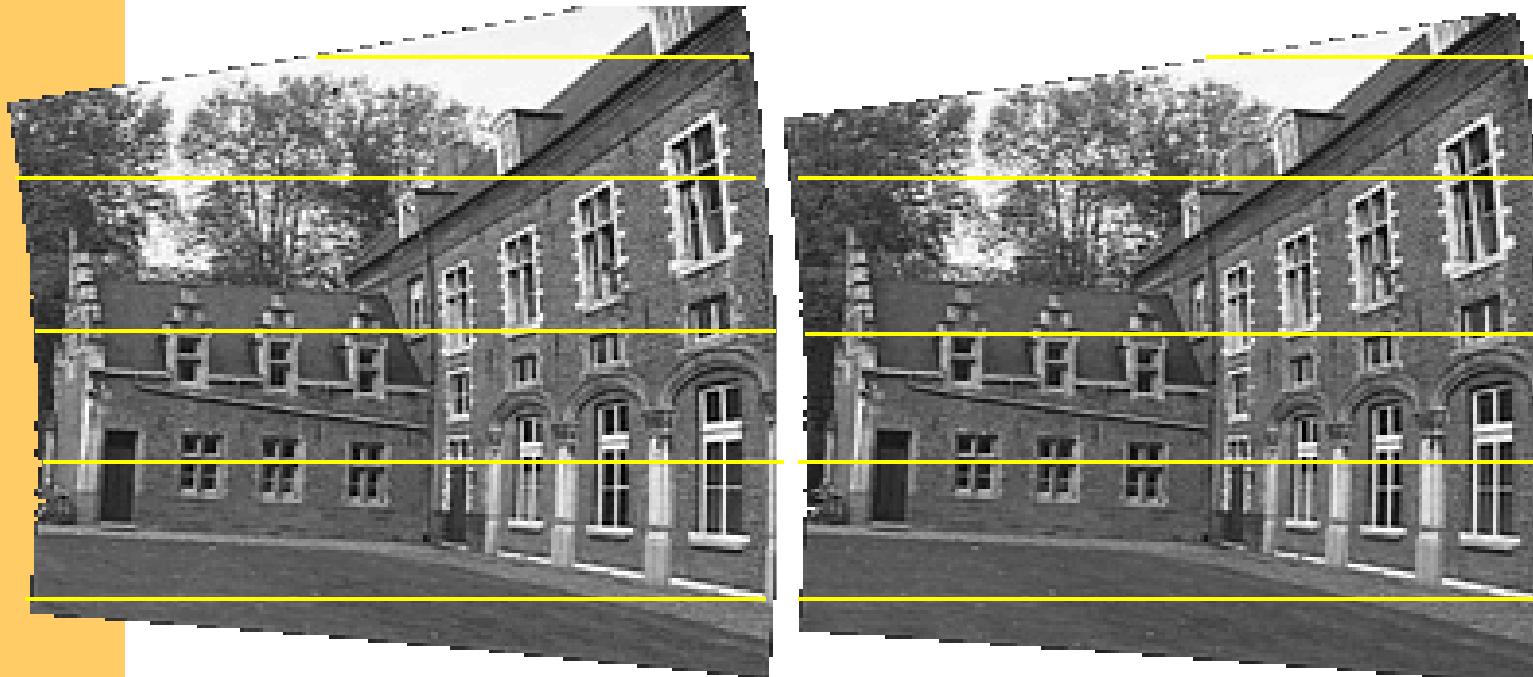


polar rectification: example





polar rectification: example





Example: Béguinage of Leuven



Does not work with standard
Homography-based approaches

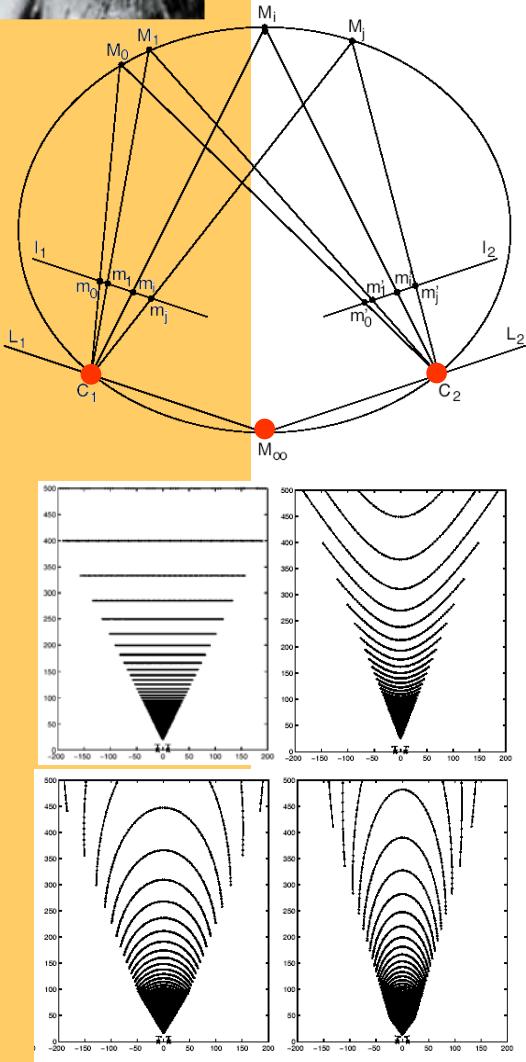


Example: Béguinage of Leuven

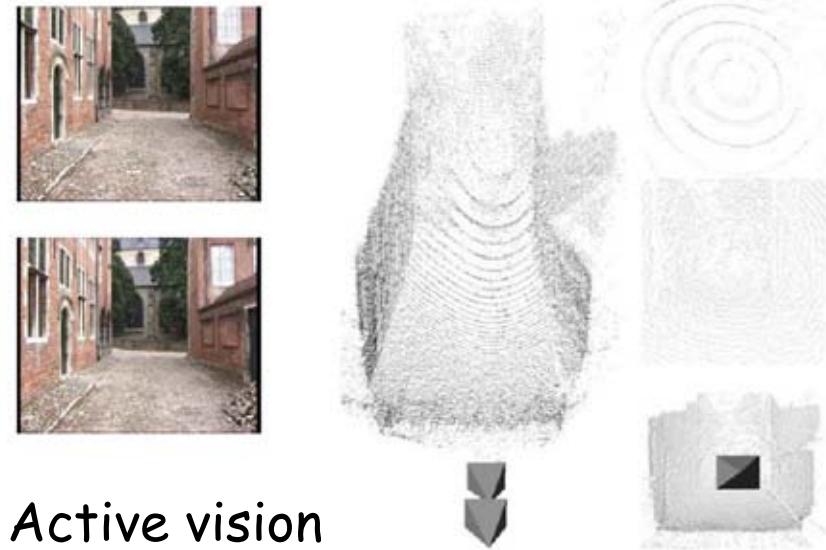




General iso-disparity surfaces



(Pollefeys and Sinha, ECCV'04)
Example: polar rectification preserves disp.

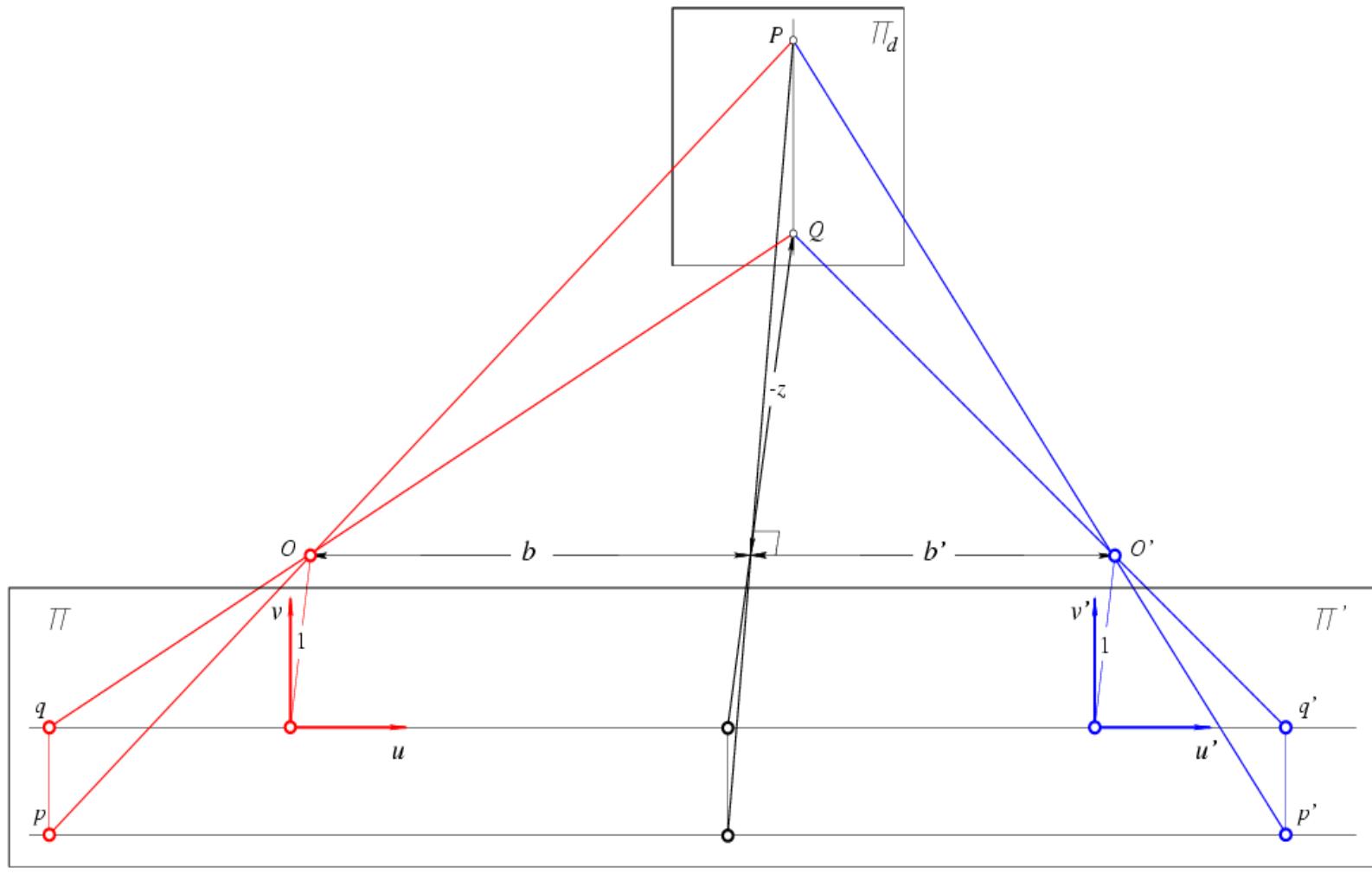


Application: Active vision



Also interesting relation to human horopter

Reconstruction from Rectified Images



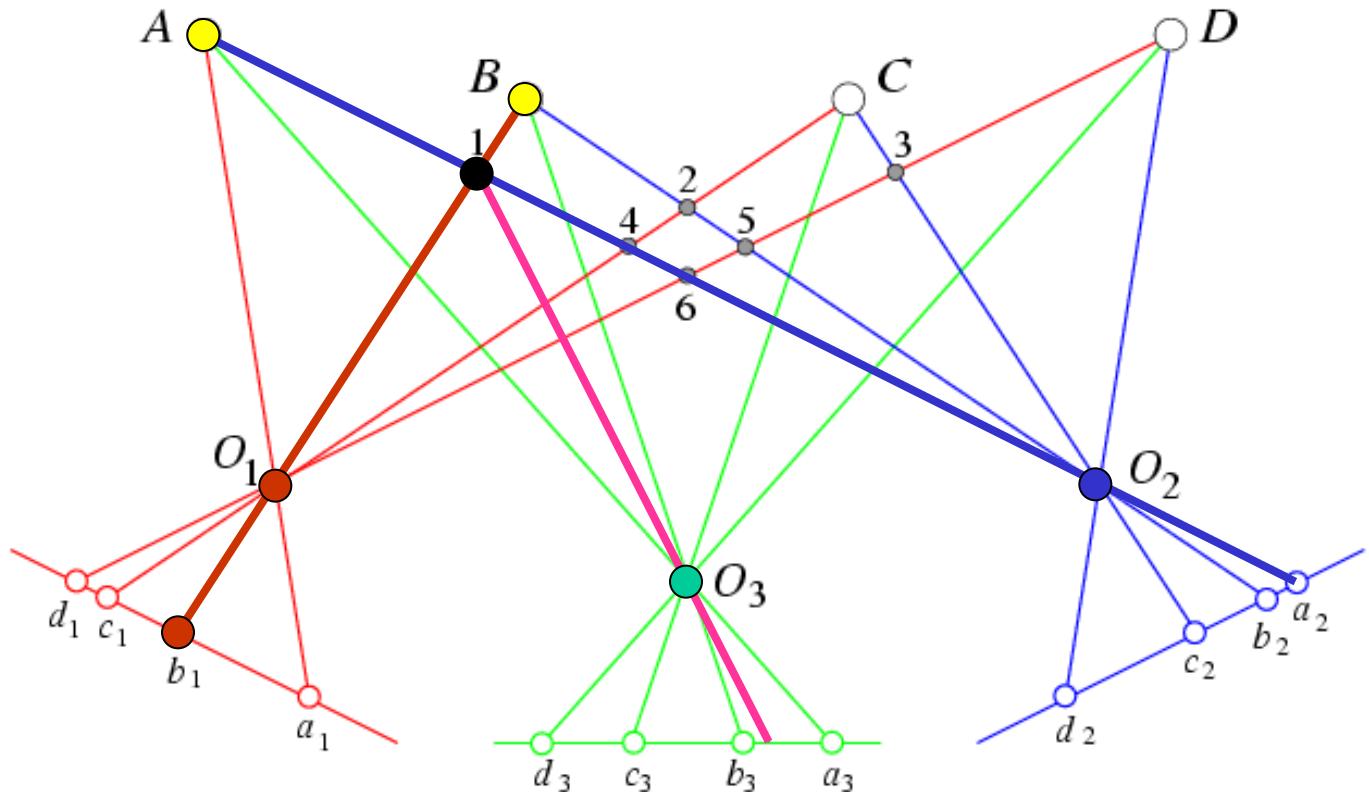
Disparity: $d = u' - u$.



Depth: $z = -B/d$.



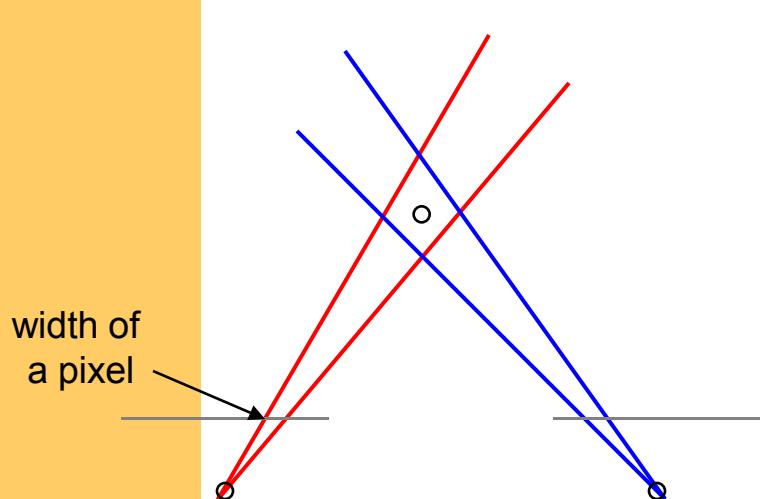
Three Views



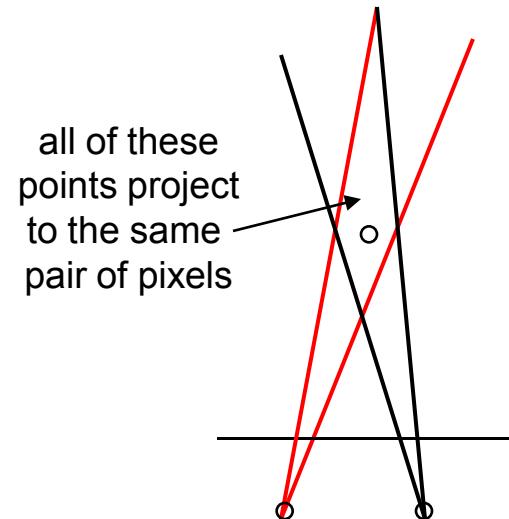
The third eye can be used for verification..



Choosing the stereo baseline



Large Baseline



Small Baseline

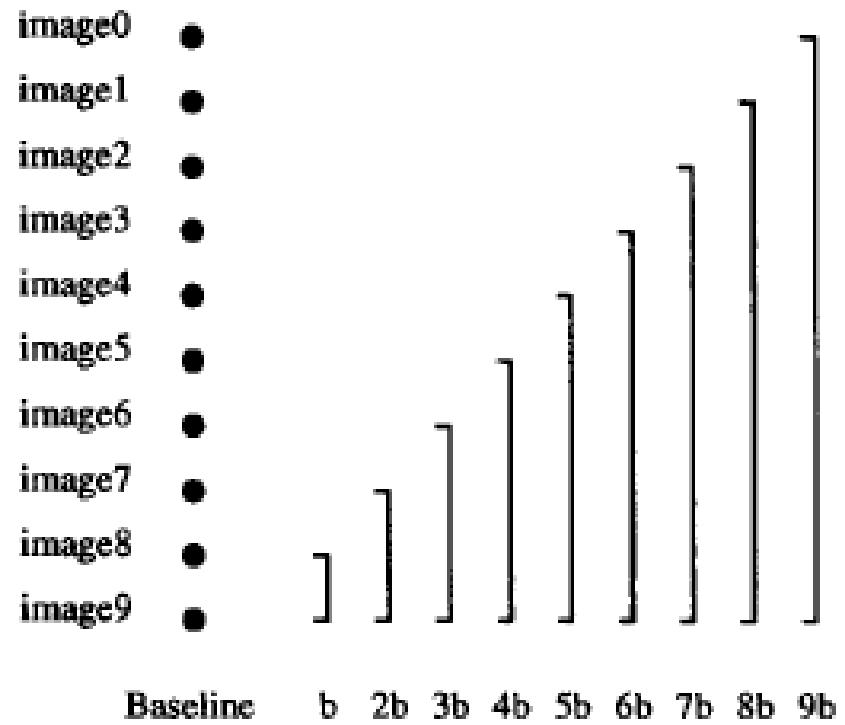
- What's the optimal baseline?
 - Too small: large depth error
 - Too large: difficult search problem, more occlusions



The Effect of Baseline on Depth Estimation



Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.



(Okutami and Kanade PAMI'93)

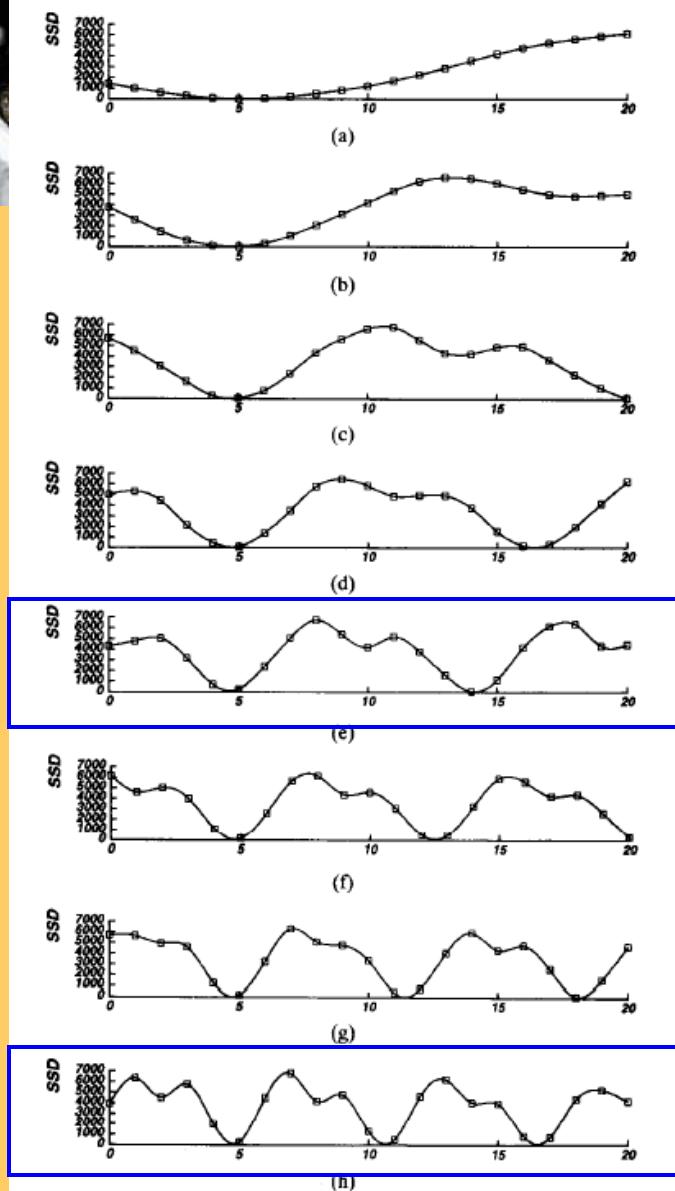


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

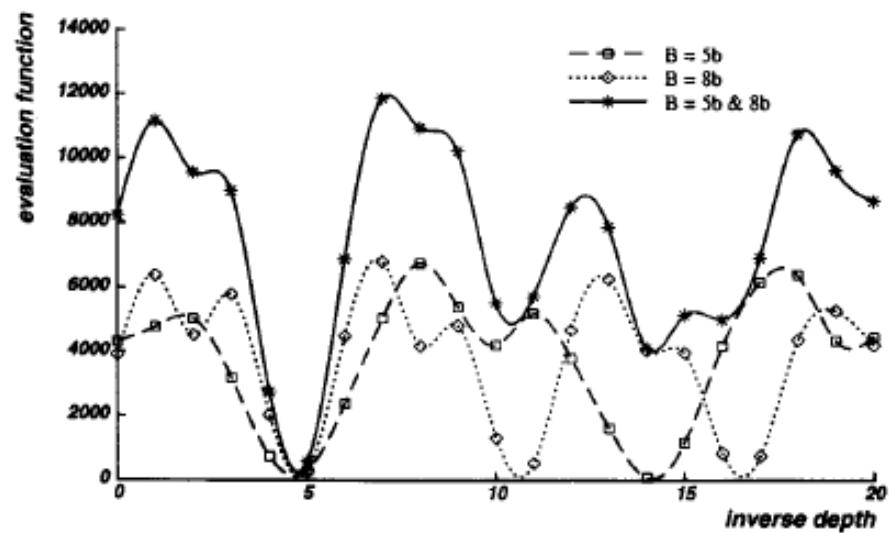


Fig. 6. Combining two stereo pairs with different baselines.

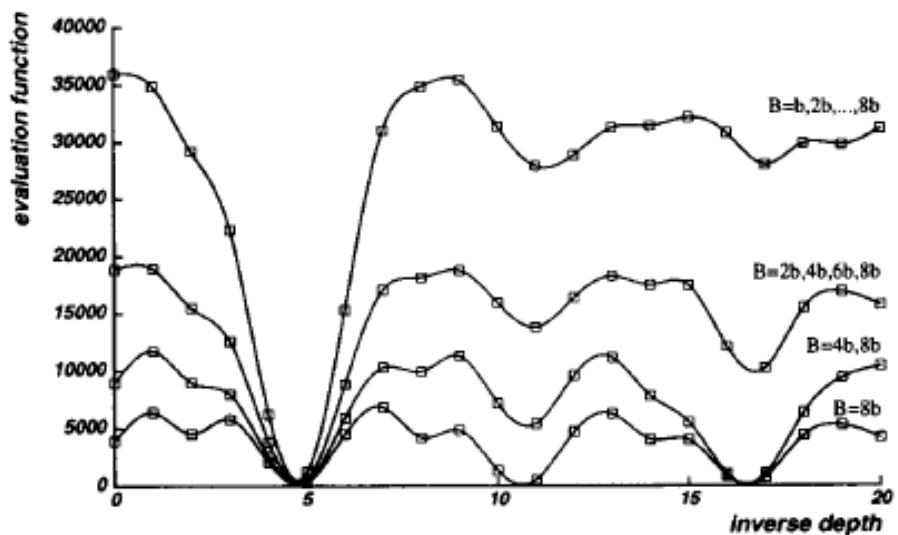


Fig. 7. Combining multiple baseline stereo pairs.



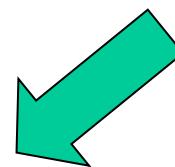
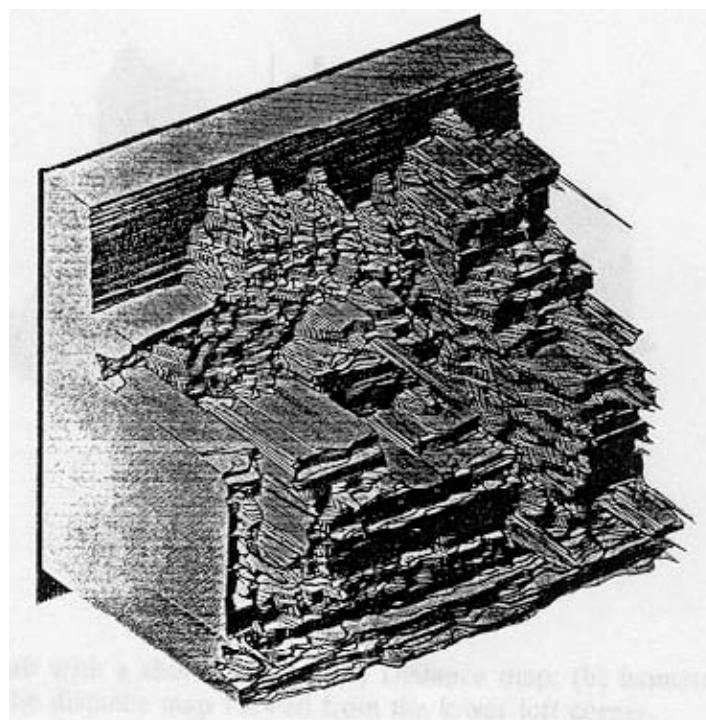
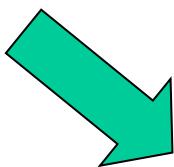
I1



I2



I10



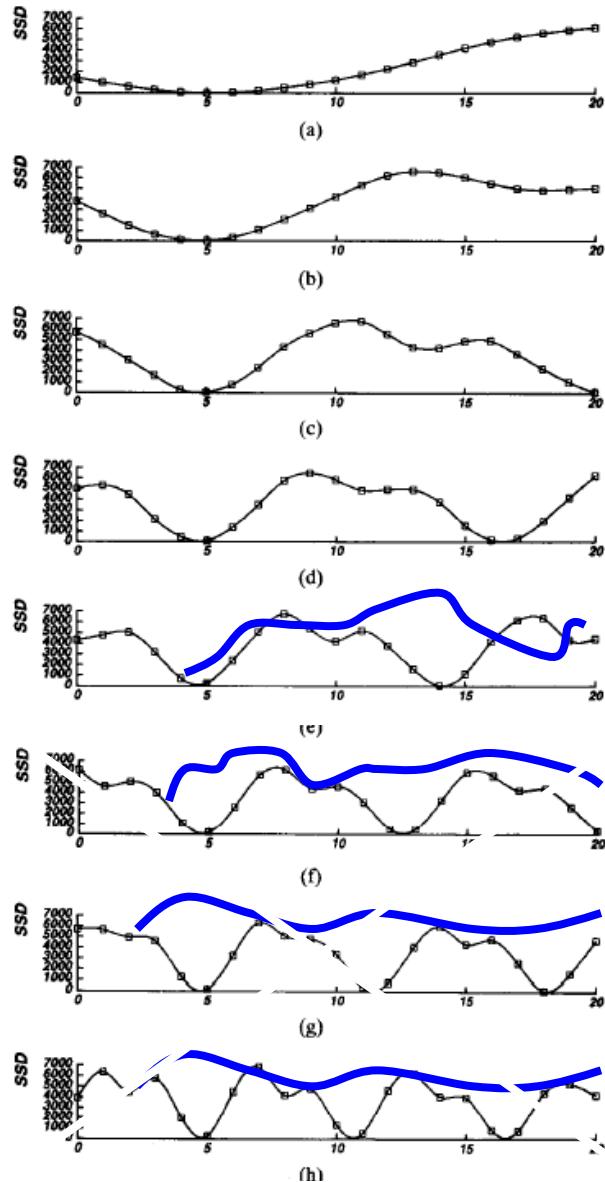


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

Problem: *visibility*

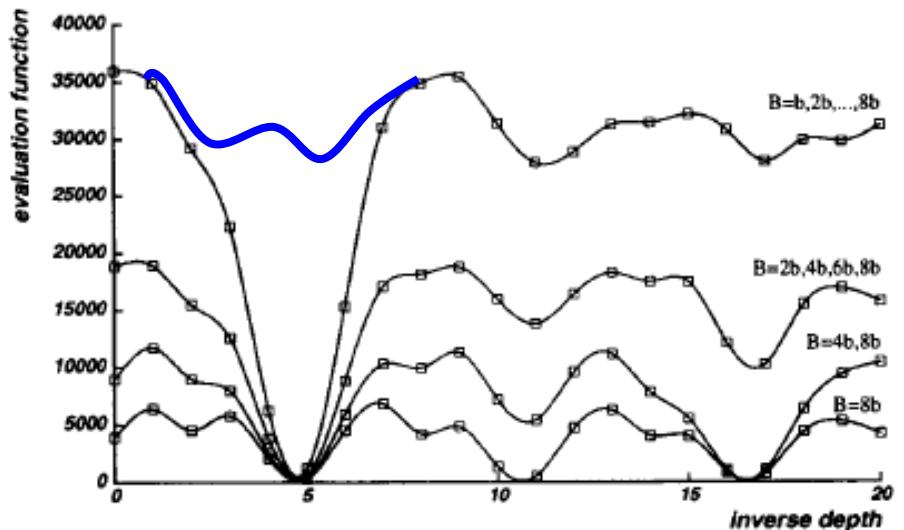


Fig. 7. Combining multiple baseline stereo pairs.

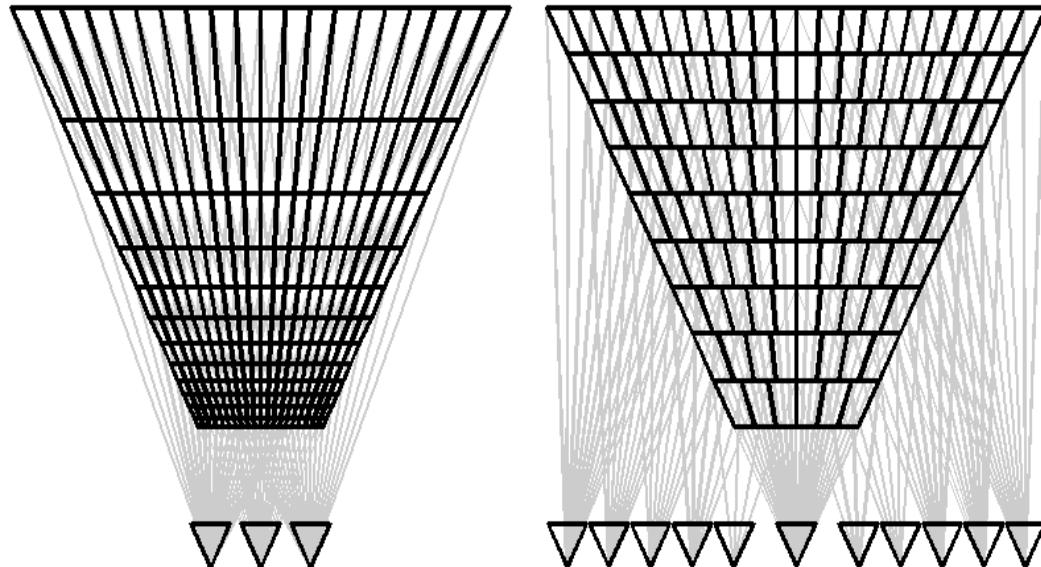
Some Solutions

- Multi-view linking ([Koch et al. ECCV98](#))
- Best of left or right ([Kang et al. CVPR01](#))
- Best k out of n , ...



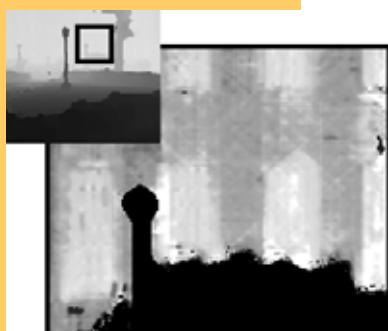
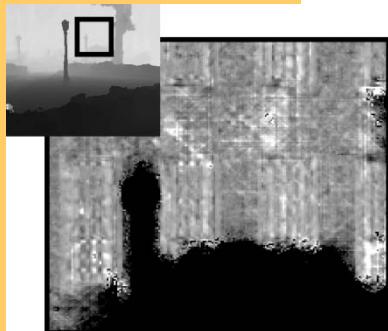
Variable Baseline/Resolution Stereo

(Gallup et al., CVPR08)



- Multi-baseline, multi-resolution
- At each depth, baseline and resolution selected proportional to that depth
- Allows to keep depth accuracy constant!

Variable Baseline/Resolution Stereo: comparison

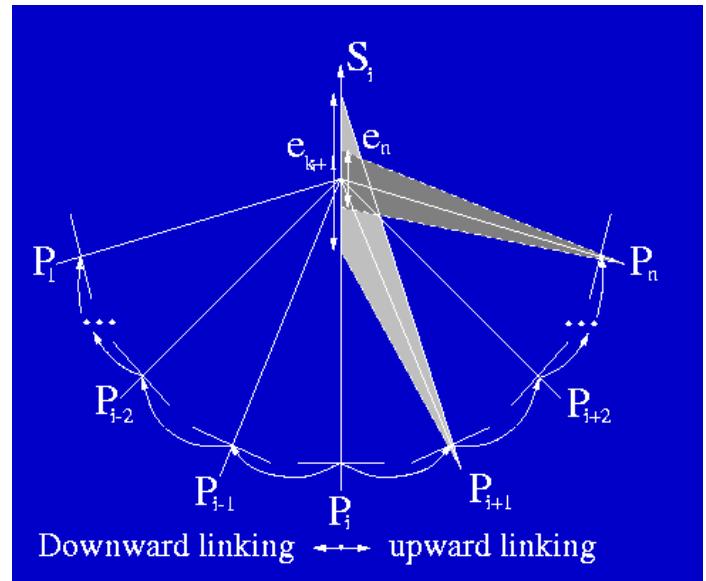




Multi-view depth fusion

(Koch, Pollefeys and Van Gool. ECCV'98)

- Compute depth for every pixel of reference image
 - Triangulation
 - Use multiple views
 - Up- and down sequence
 - Use Kalman filter



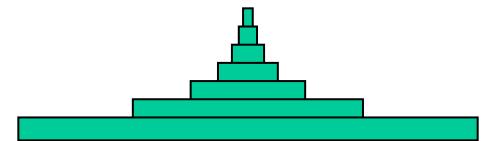
Allows to compute robust texture



Real-time stereo on graphics hardware

Yang and Pollefeys CVPR03

- Computes Sum-of-Square-Differences
- Hardware mip-map generation used to aggregate results over support region
- Trade-off between small and large support window

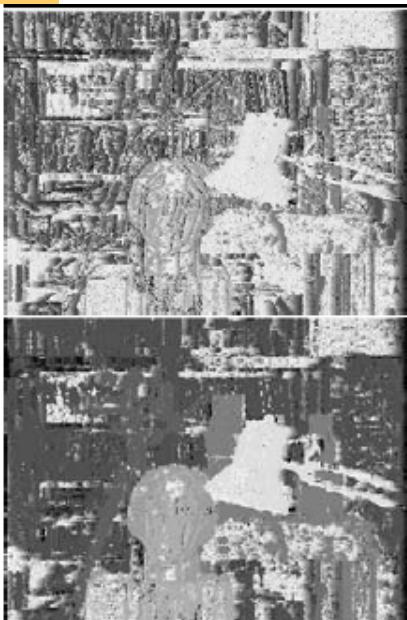


Shape of a kernel
for summing up 6 levels

140M disparity hypothesis/sec on Radeon 9700pro
e.g. 512x512x20disparities at 30Hz

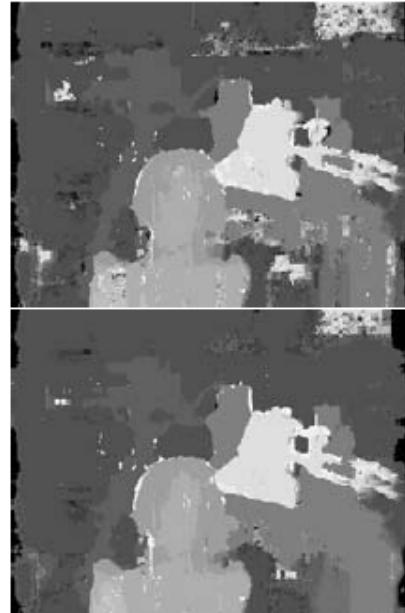


Combine multiple aggregation windows using hardware mipmap and multiple texture units in single pass



(1x1)

(1x1+2x2)

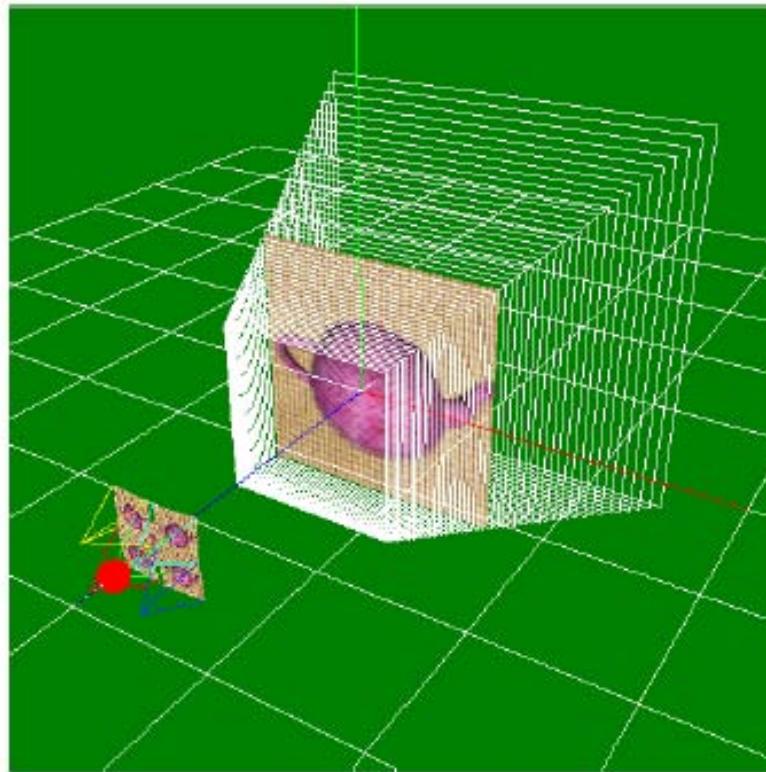


(1x1+2x2
+4x4+8x8)

(1x1+2x2
+4x4+8x8
+16x16)



Plane-sweep multi-view matching



- Simple algorithm for multiple cameras
- no rectification necessary
- doesn't deal with occlusions

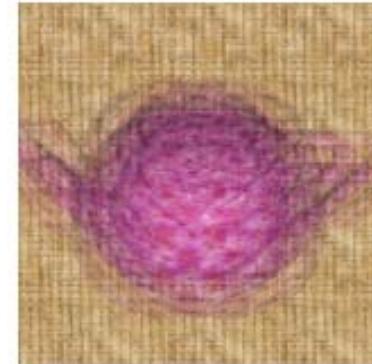
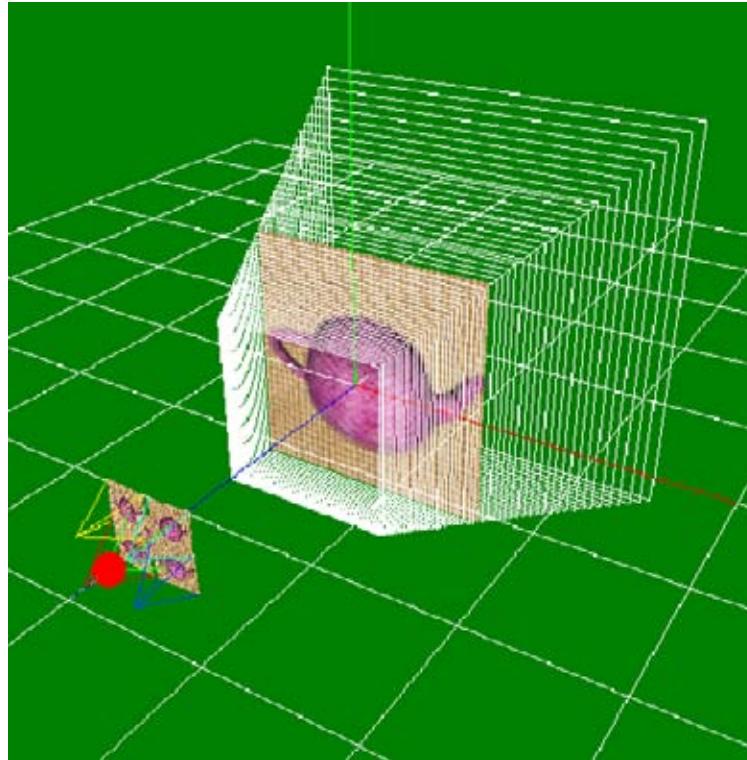
Collins'96; Roy and Cox'98 (GC); Yang et al.'02/'03 (GPU)



Fast GPU-based plane-sweeping stereo

Plane-sweep multi-view depth estimation

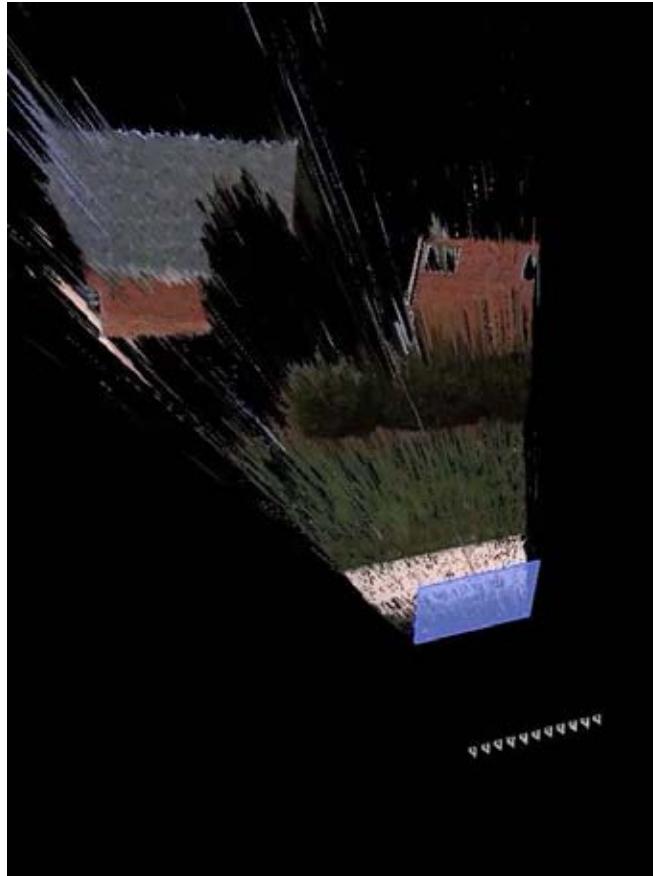
(Yang & Pollefeys, CVPR'03)





Plane Sweep Stereo

Images Projected on Plane



Plane Sweep



Matching Score



Plane Sweep Stereo Result



- Ideal for GPU processing
- 512x384 depthmap, 50 planes, 5 images: 50 Hz



Plane Sweep Stereo



Plane Sweep

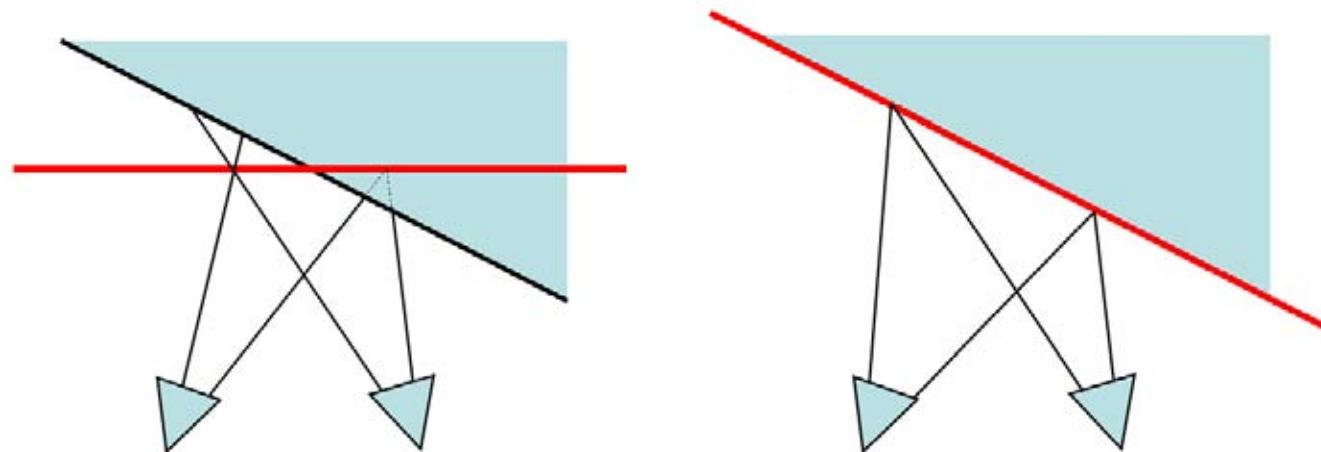


Matching Score (SAD)



Window-Based Matching - Local Stereo

- Pixel-to-pixel matching is ambiguous for local stereo
- Use a matching window (SAD/SSD/NCC)



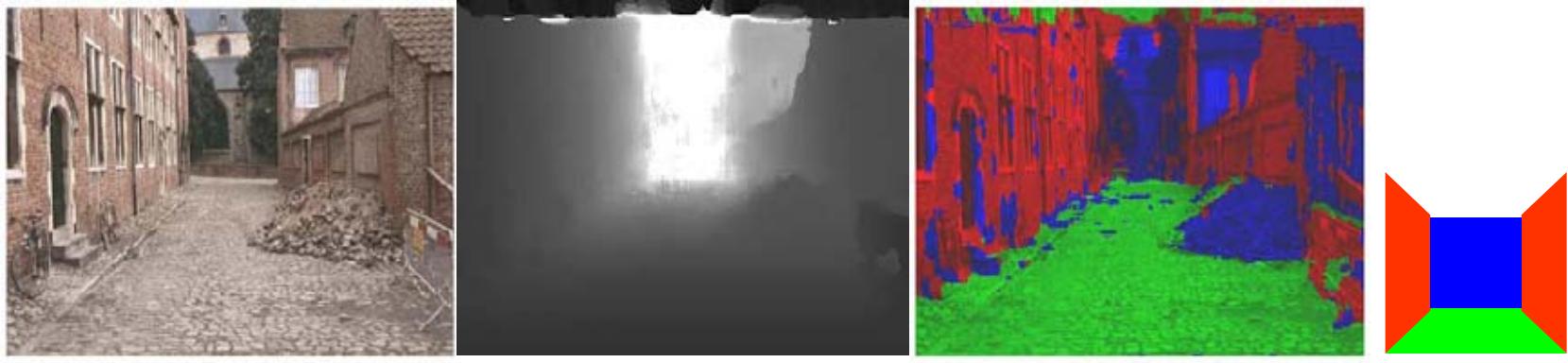
- Assumes all pixels in window are on the plane



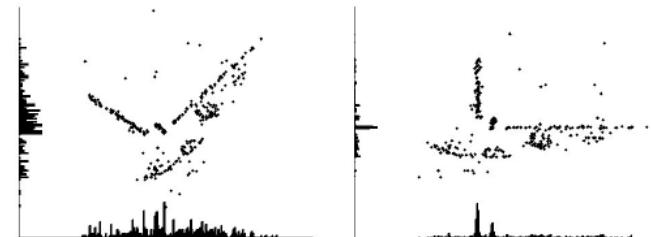
Multi-directional plane-sweeping stereo

(Gallup, Frahm & Pollefeys, CVPR07)

- Select best-cost over depths AND orientation



3D model from
11 video frames
(hand-held)

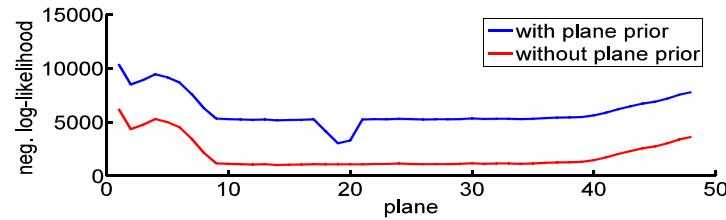


Automatic computations of dominant orientations



Plane Priors and Quicksweep

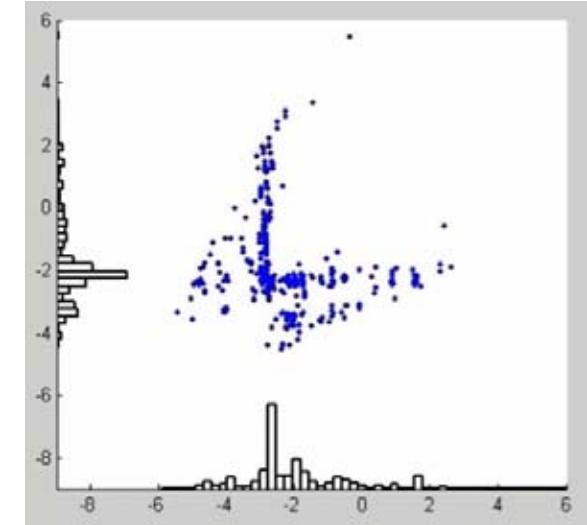
Use SfM points as a prior for stereo



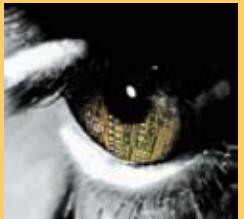
without plane prior



with plane prior

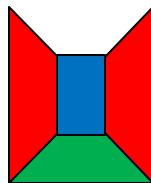


For real-time:
test only the planes with high prior probability

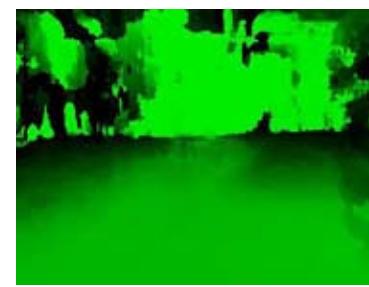


Selecting Best Depth / Surface Normal

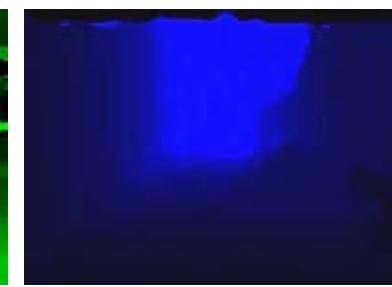
- Combining multiple sweeps



■ Sweep 1

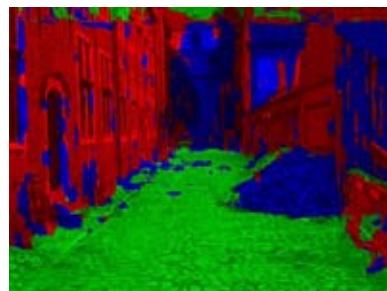


■ Sweep 2



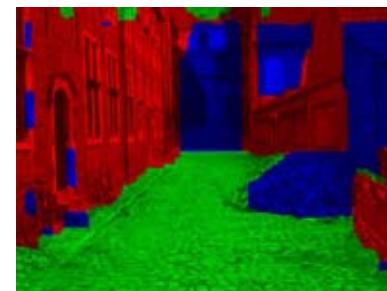
■ Sweep 3

How to
combine
sweeps?



Local best cost

OR



Global labeling

Graph Cut:
1-2 seconds

Continuous
Formulation: 45 ms

(Zach, Gallup, Frahm,
Niethammer VMV
2008)



Results



1474 frame reconstruction, 11 images in stereo,
512x384 grayscale, 48 plane hypotheses (quick sweep),
processing rate 20.0 Hz (Nvidia 8800 GTX)

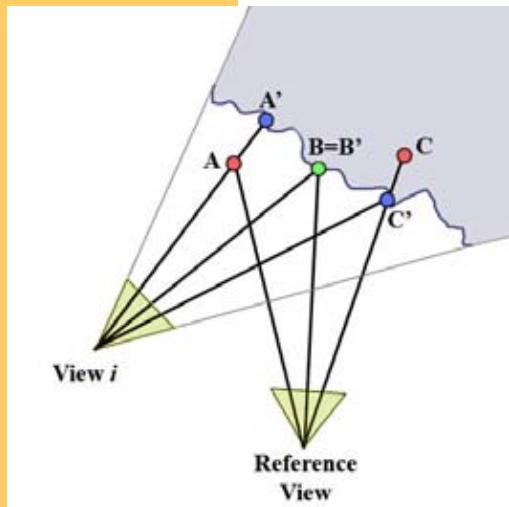


Fast visibility-based fusion of depth-maps

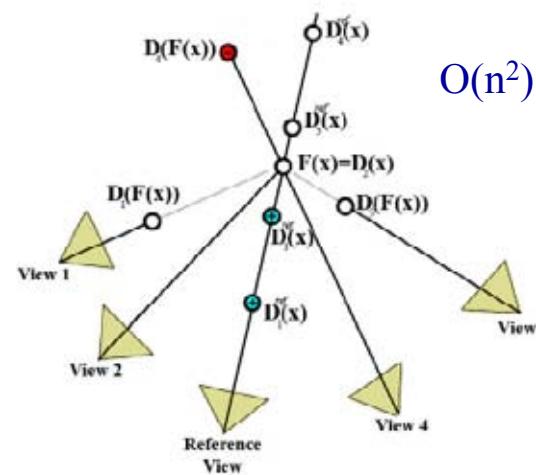
(Merrell , Akbarzadeh, Wang, Mordohai, Frahm, Yang, Nister, Pollefeys ICCV07)

- Complements very fast multi-view stereo

visibility constraints

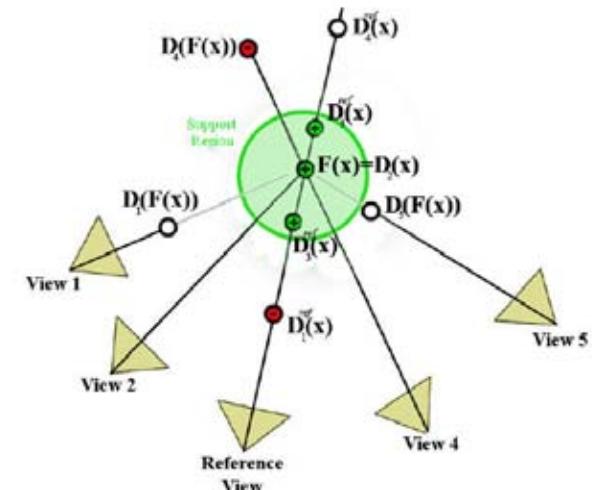


stability-based fusion



Two fusion approaches

confidence-based fusion



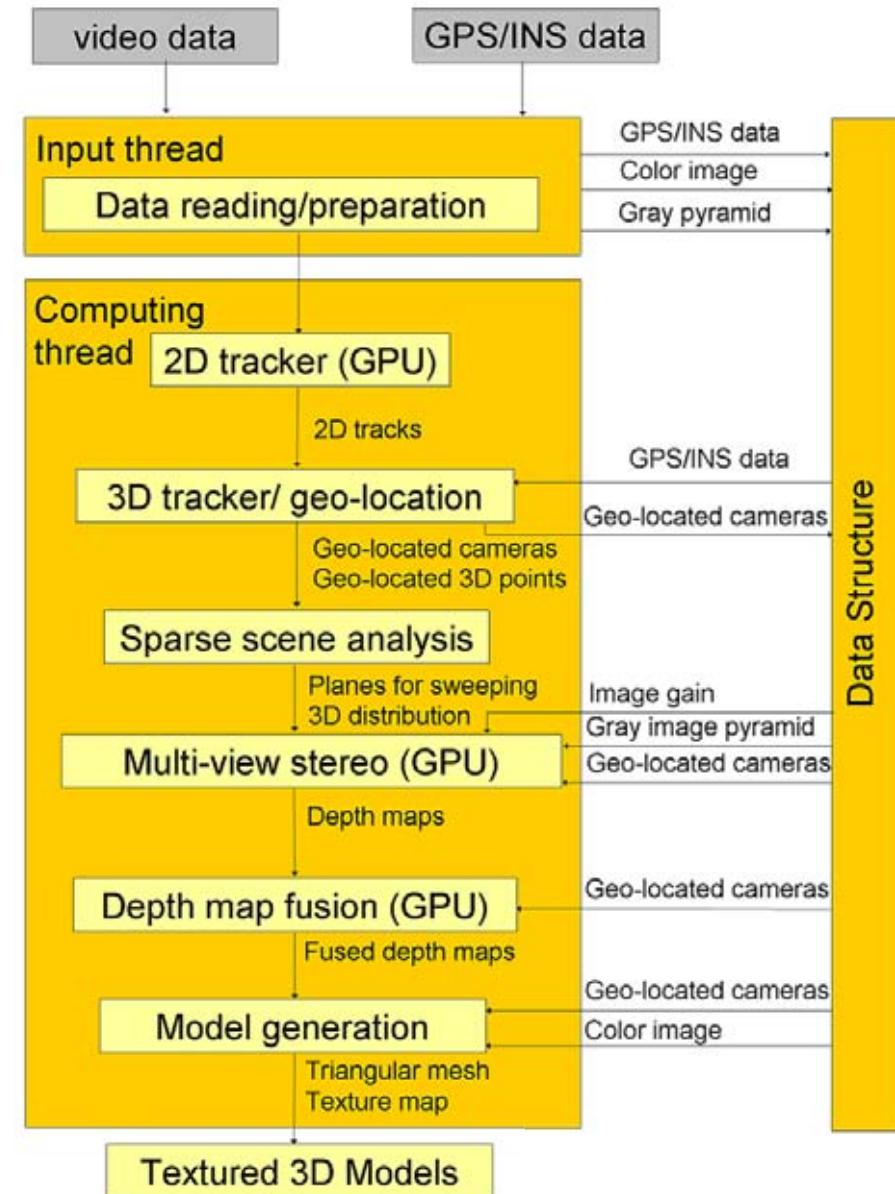
Fast on GPU as key operation is rendering depth-maps into other views



Fast video-based modeling of cities

- **Fast video processing pipeline**
- - up to 26Hz on single CPU/GPU
- - Most image processing on GPU
($\times 10\text{-}\times 100$ faster)
- - Exploits urban structure
- - Generates textured 3D mesh

(Pollefeys et al. IJCV, 2008)

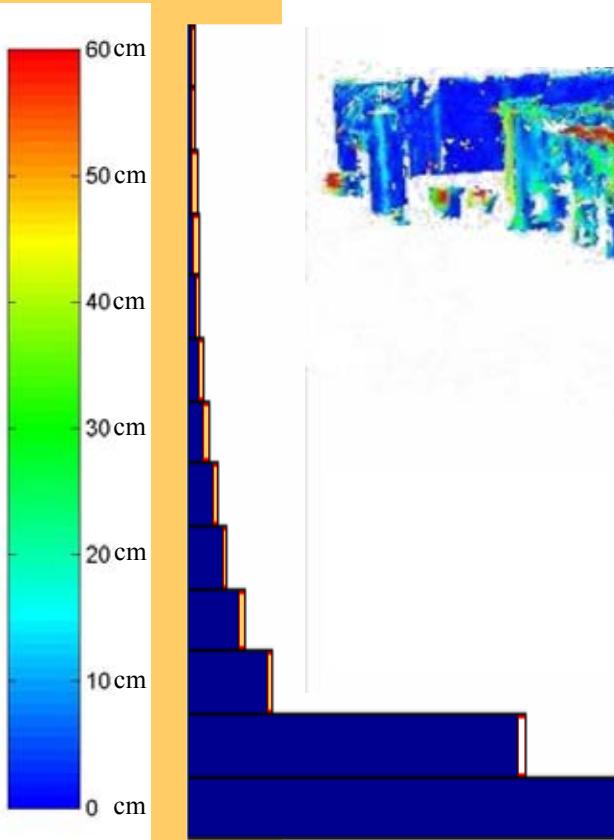




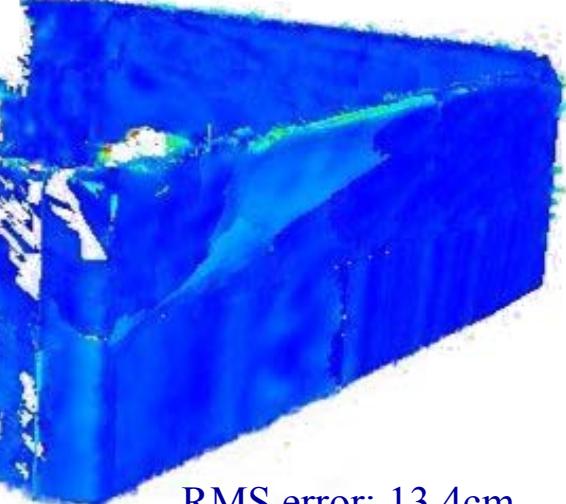
3D-from-video evaluation: Firestone building



building surveyed to 6mm



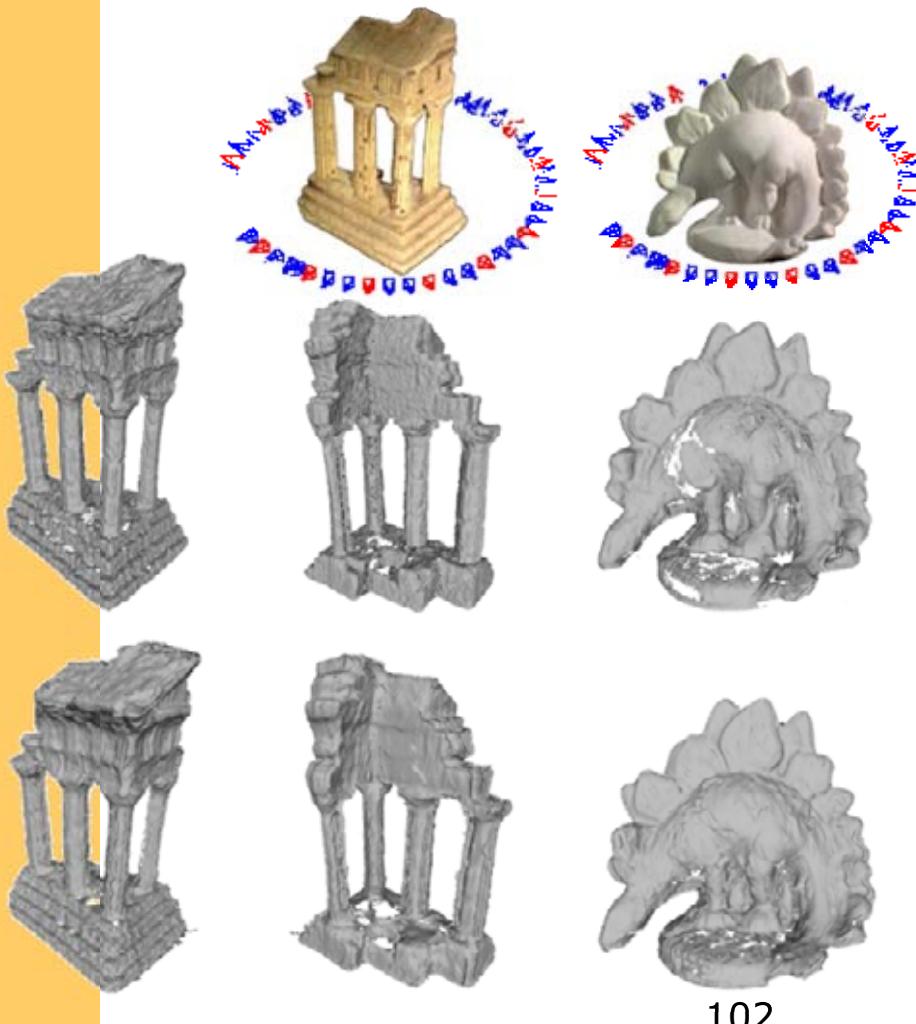
error histogram



RMS error: 13.4cm
mean error: 6.8cm
median error: 3.0cm

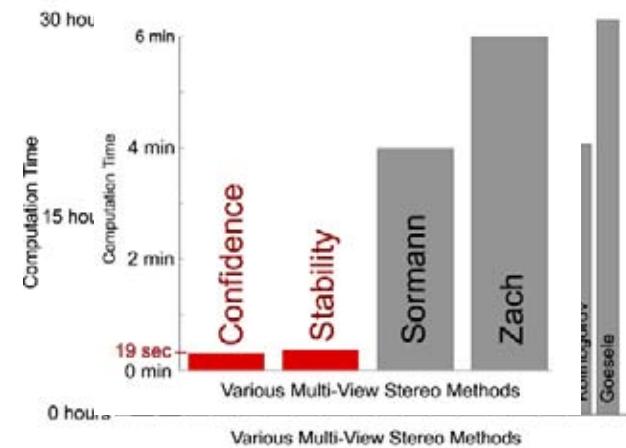


3D-from-video evaluation: Middlebury Multi-View Stereo Evaluation Benchmark

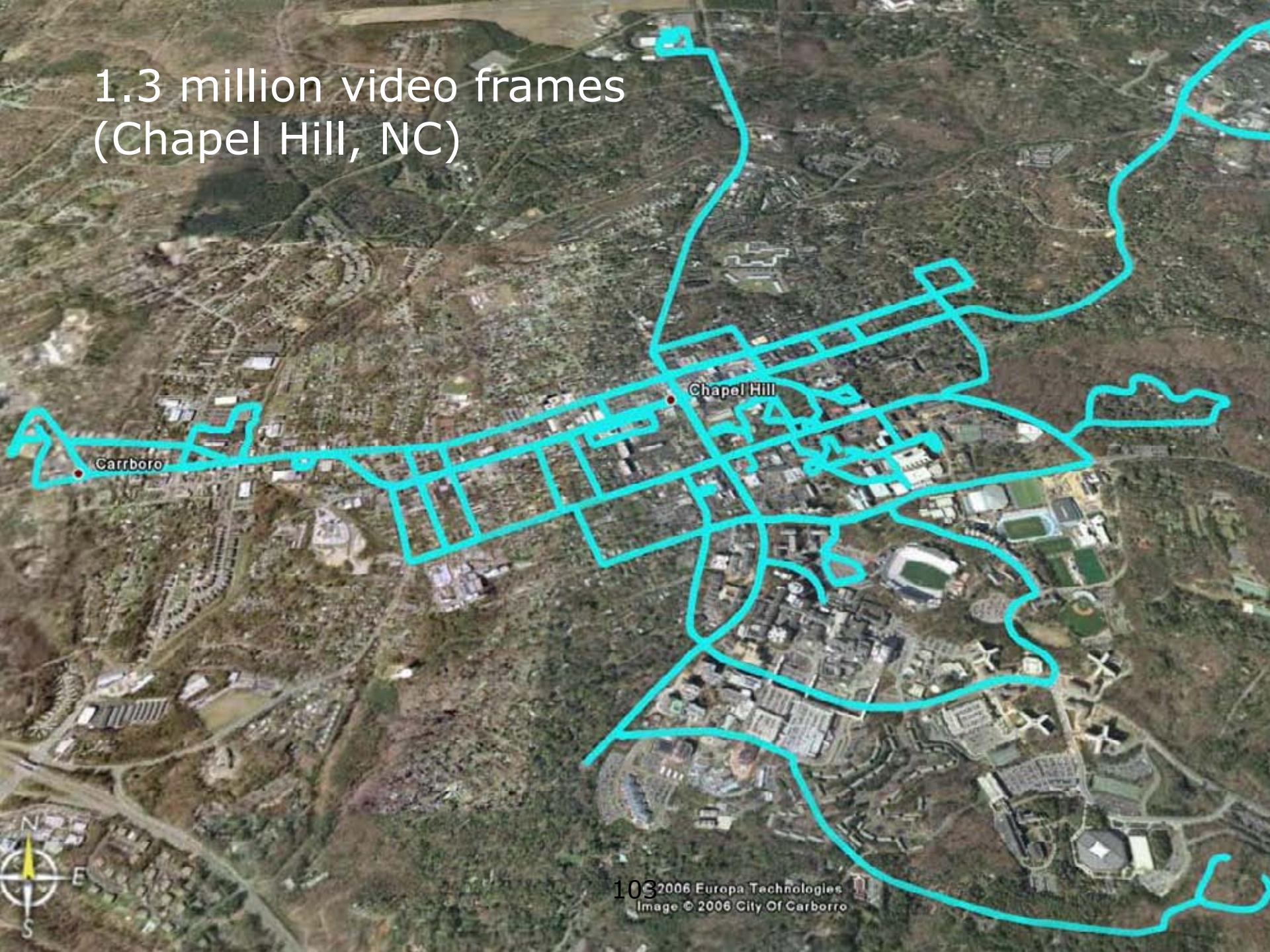


Ring datasets: 47 images

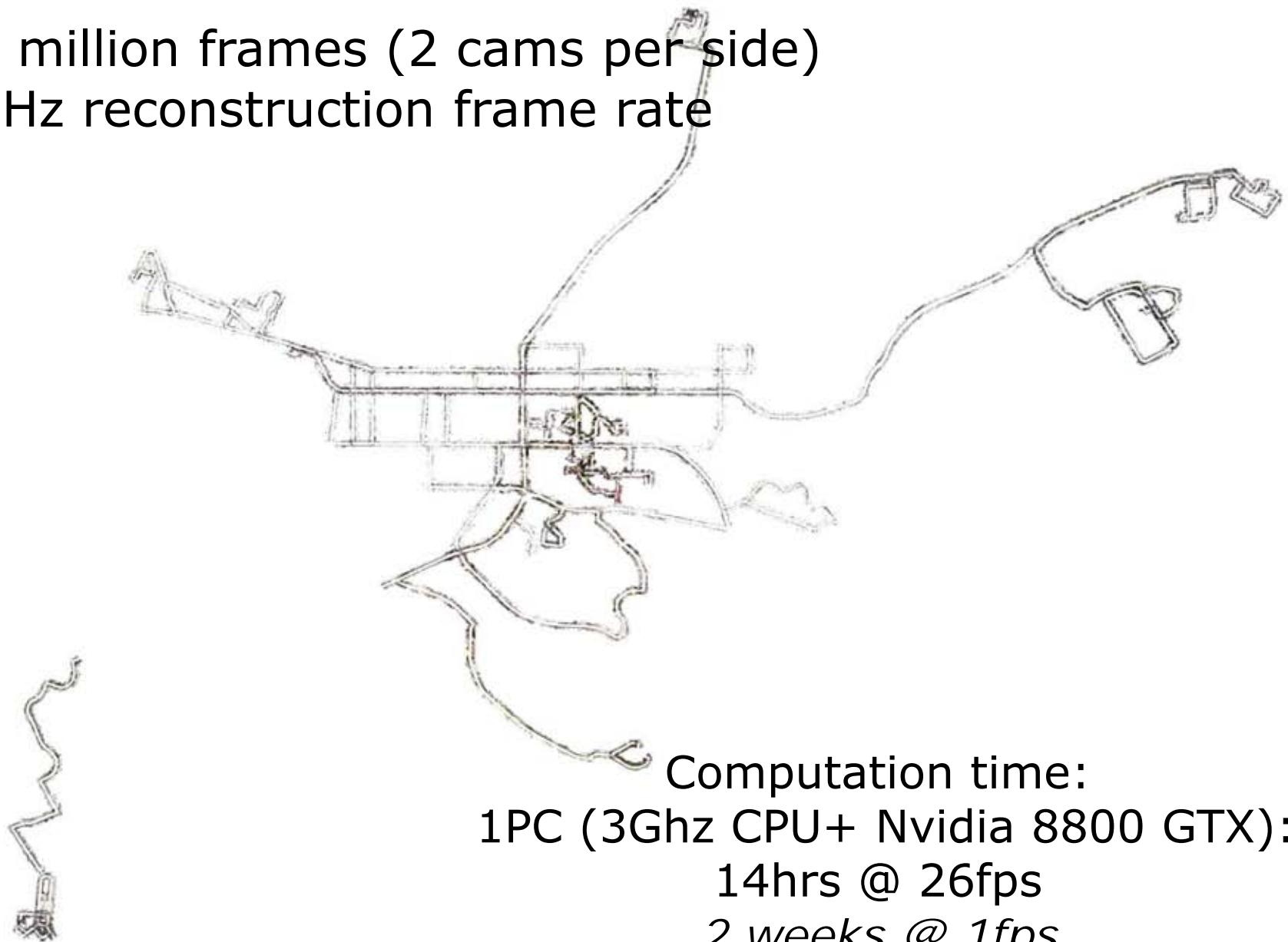
Results competitive
but much, much faster
(30 minutes → 30 seconds)



1.3 million video frames
(Chapel Hill, NC)

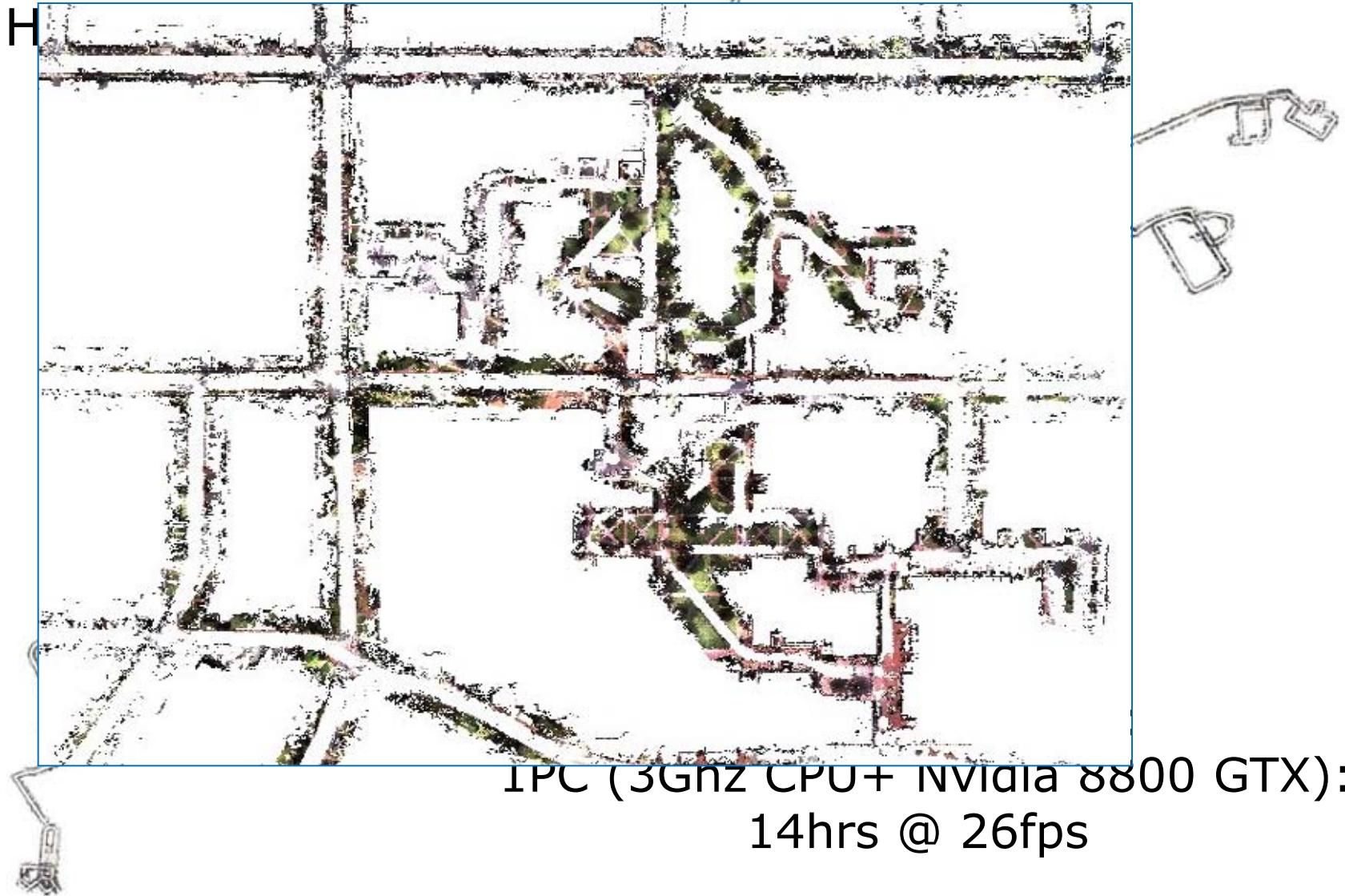


- 1.3 million frames (2 cams per side)
- 26 Hz reconstruction frame rate



Computation time:
1PC (3Ghz CPU+ Nvidia 8800 GTX):
14hrs @ 26fps
2 weeks @ 1fps
2.5 years @ 1fpm

- 1.3 million frames (2 cams per side)
- 26 H



IPC (3Ghz CPU + Nvidia 8800 GTX):
14hrs @ 26fps





Another Scene



view 1



...



...

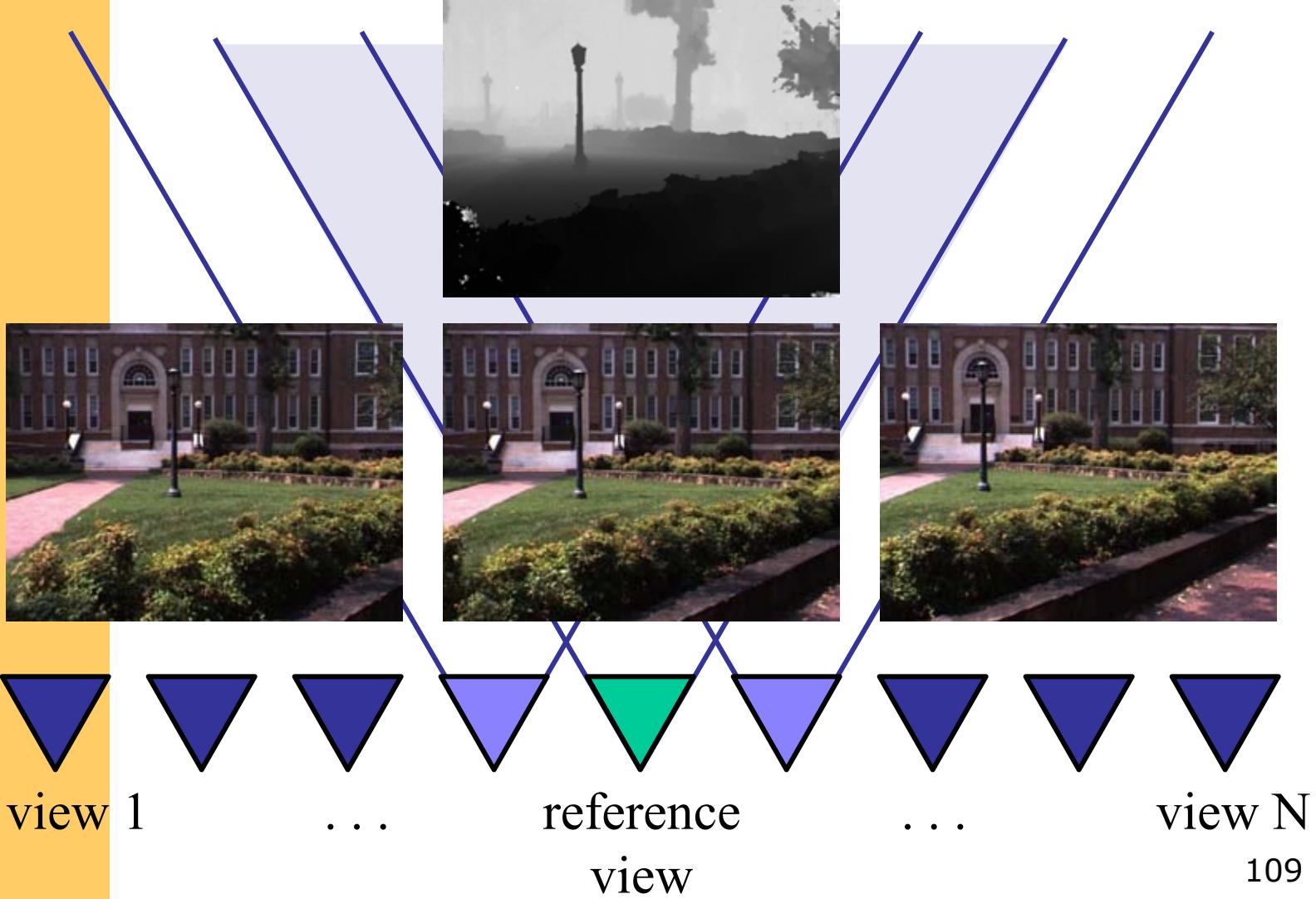


view N

108



Depthmap





3D Model: Standard Stereo





3D Model: Standard Stereo



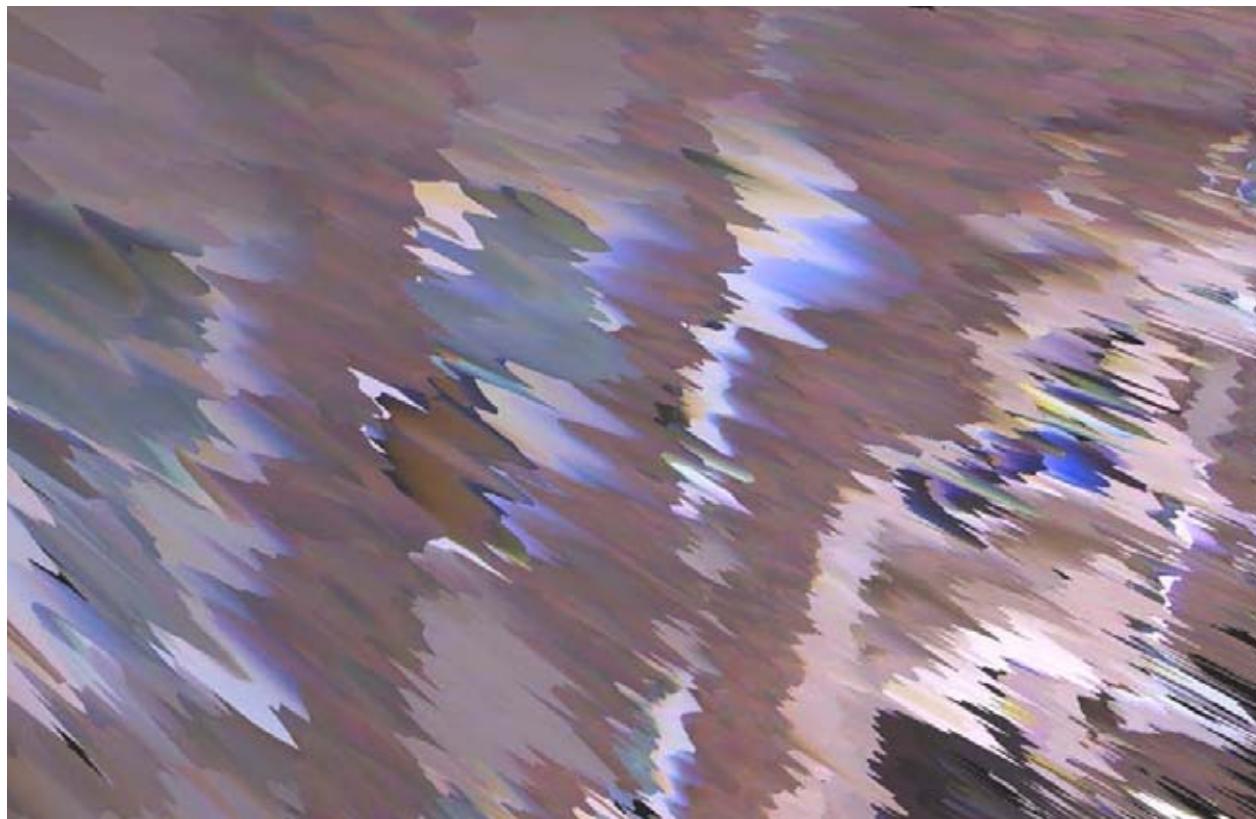


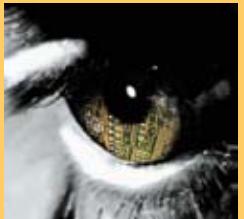
3D Model: Standard Stereo





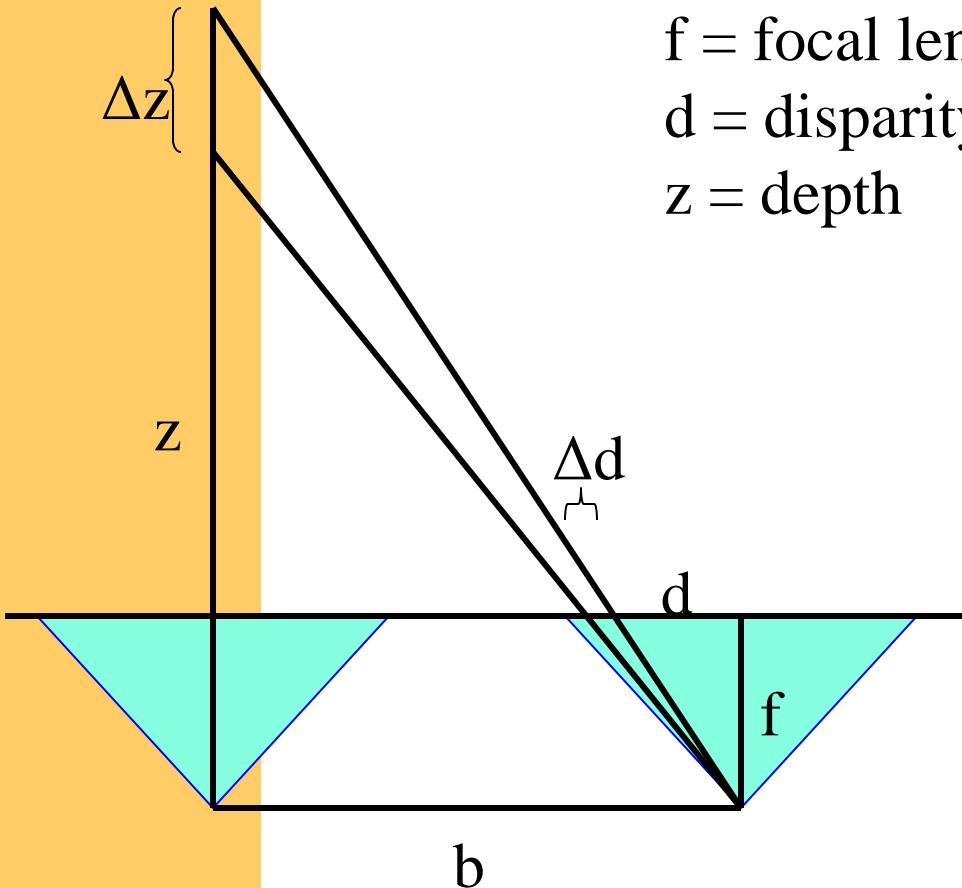
3D Model: Standard Stereo





How Accurate Is Stereo?

b = baseline
 f = focal length
 d = disparity
 z = depth



$$z = -\frac{bf}{d}$$

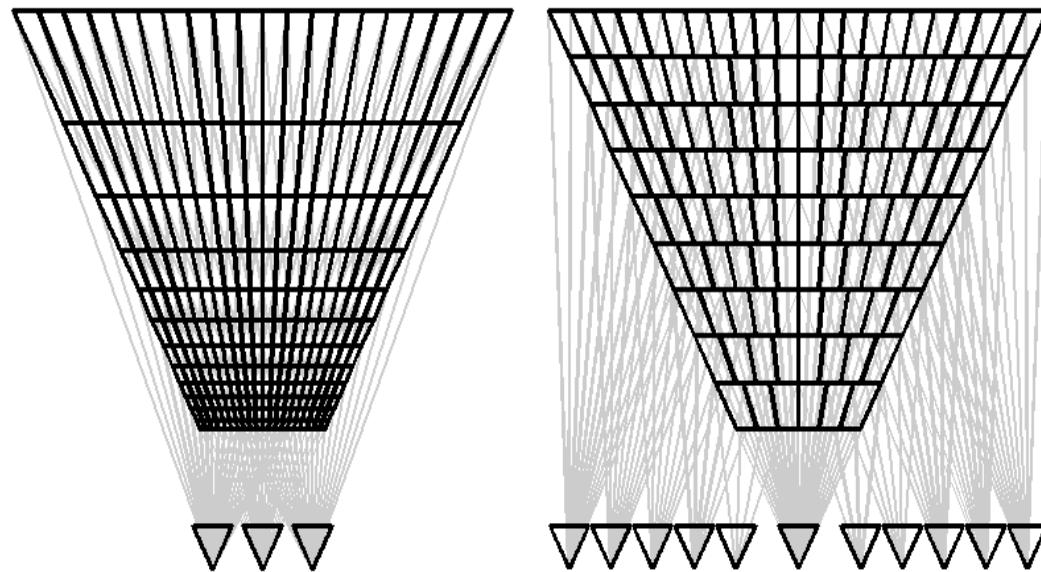
$$\Delta z \approx z' \Delta d = \frac{bf}{d^2} \Delta d$$

$$\Delta z \approx \frac{z^2}{bf} \Delta d$$



Variable Baseline/Resolution Stereo

(Gallup et al., CVPR08)

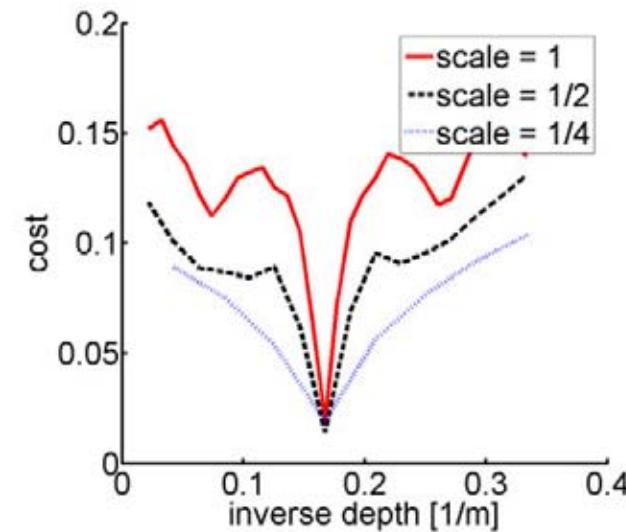
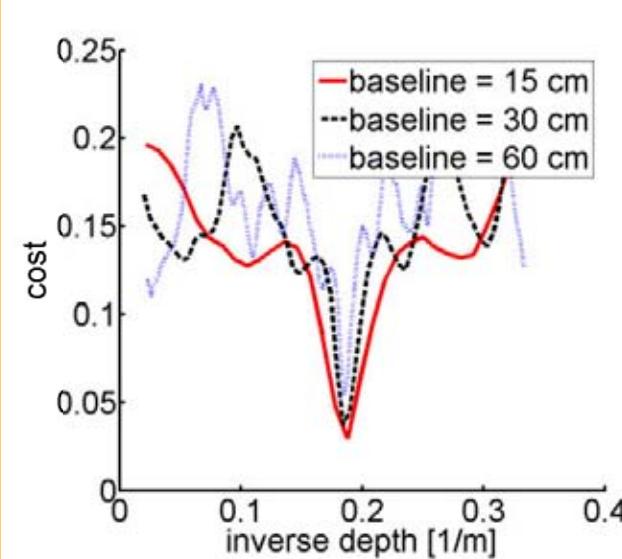


- Multi-baseline, multi-resolution
- At each depth, baseline and resolution selected proportional to that depth
- Allows to keep depth accuracy constant!

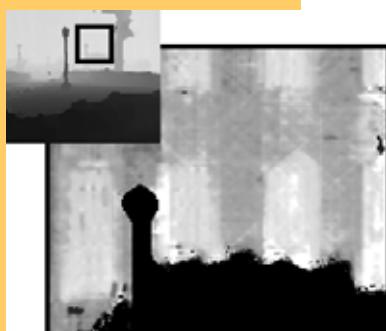
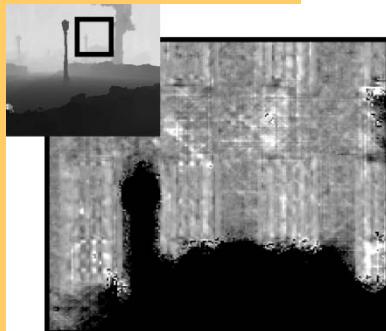


Verification of consistent results

- Analysis of matching scores at various baselines and scales
 - Location of global minimum unchanged
 - Value of global minimum preserved

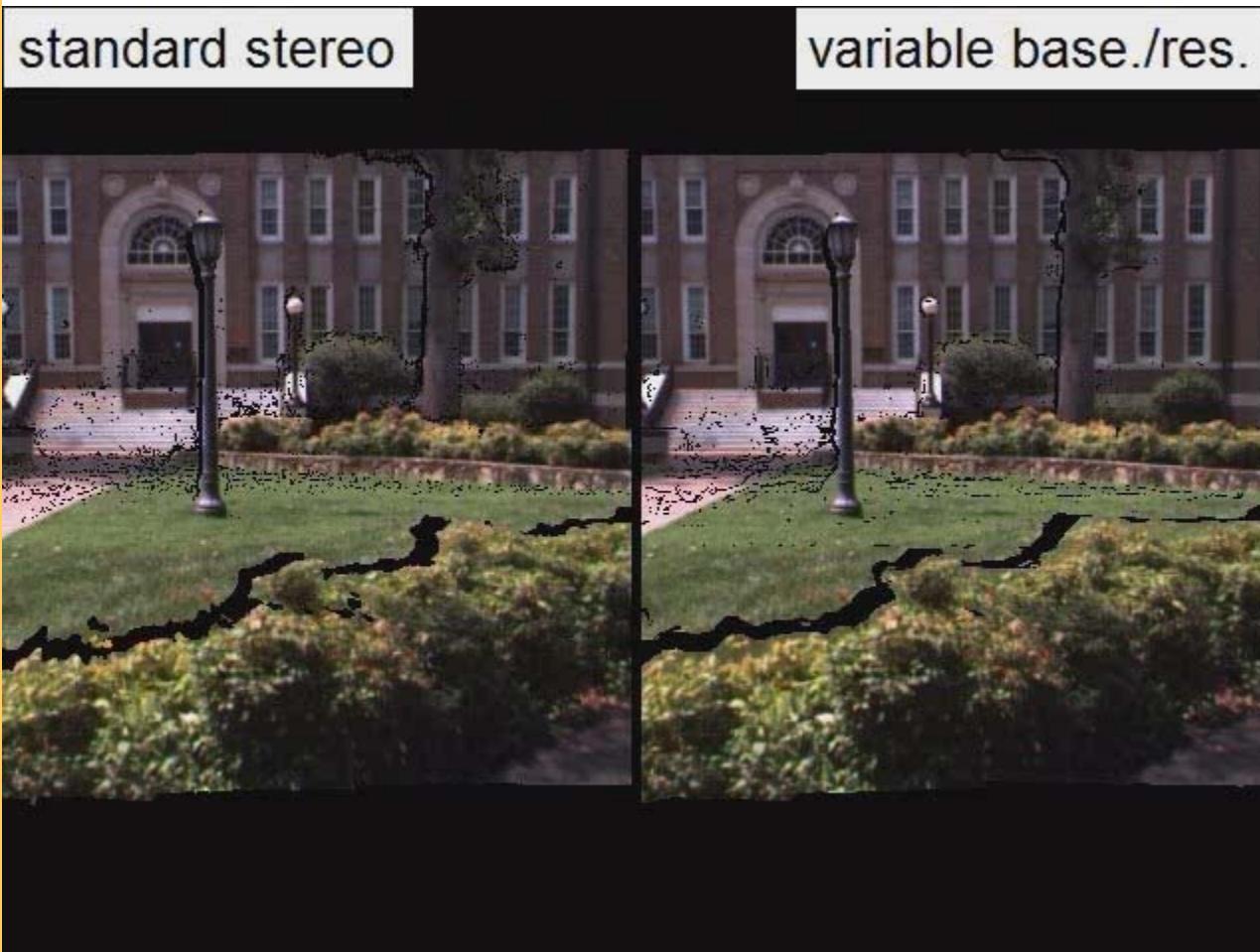


Variable Baseline/Resolution Stereo: comparison





Results





Depth Resolution Limits





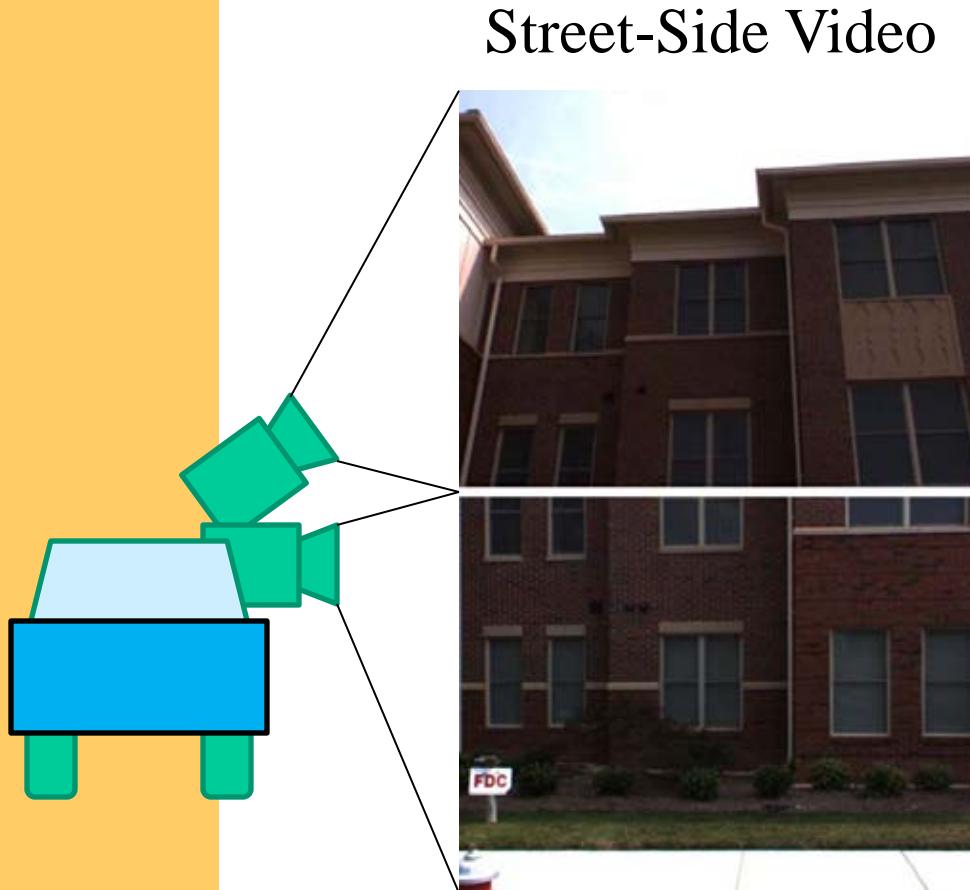
Piecewise Planar Stereo

- Fit planes to scene
- But also handle non-planar regions
- Use planar appearance
- Ensure consistency over video sequences

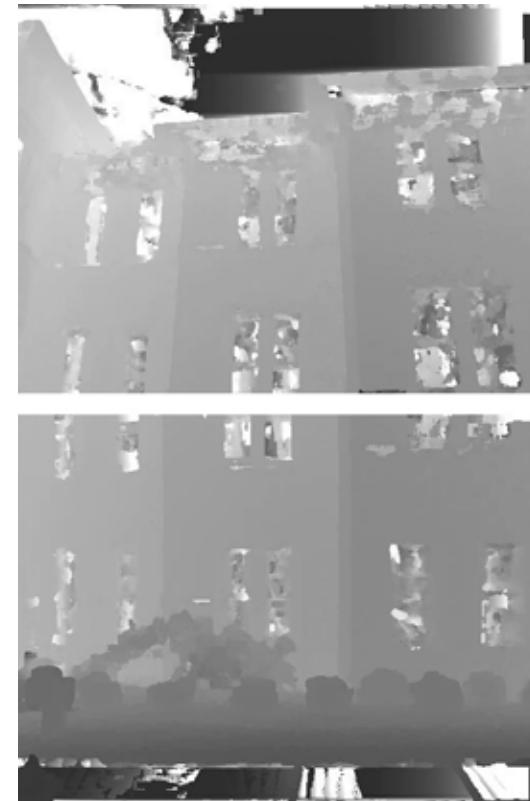




3D Reconstruction from Video



Real-Time Stereo



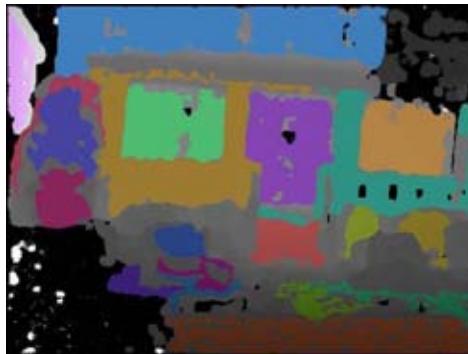


Method

(Gallup et al. CVPR 2010)



Video Frame



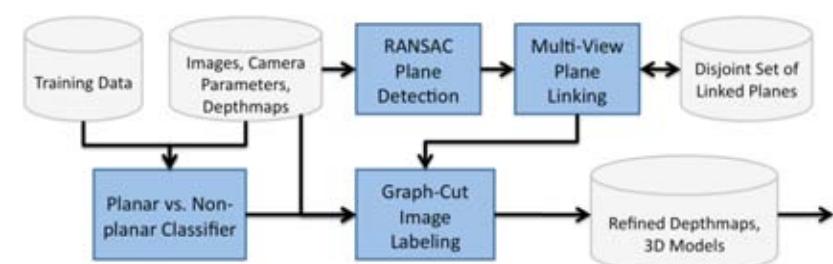
Depthmap with
RANSAC planes



Planar Class
Probability Map



Graph-Cut Labeling



Flowchart



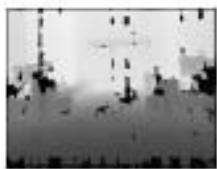
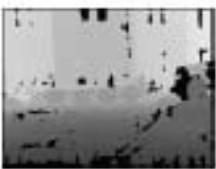
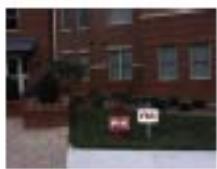
3D Model 142





Heightmap Model

(Gallup et al. 3DPVT 2010)

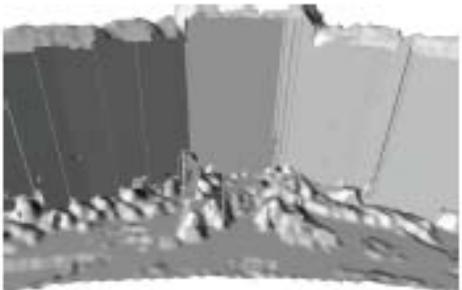


Sample Input Images

Sample Input Depthmaps



Heightmap



3D Model Geometry



Textured 3D Model

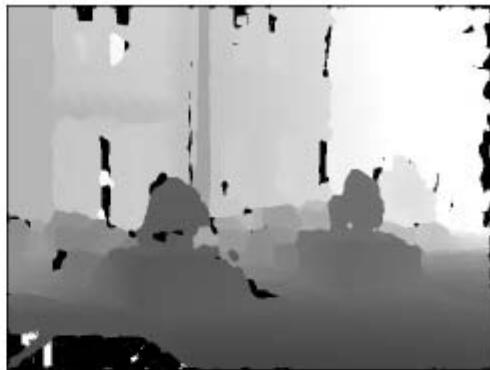
- Enforces vertical facades
- One continuous surface, no holes
- 2.5D: Fast to compute, easy to store



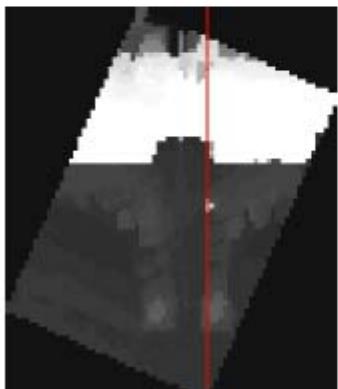
Occupancy Voting



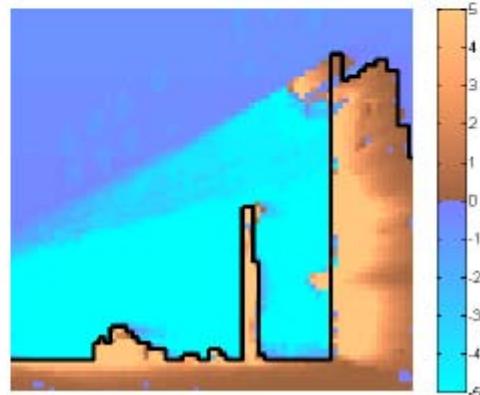
(a)



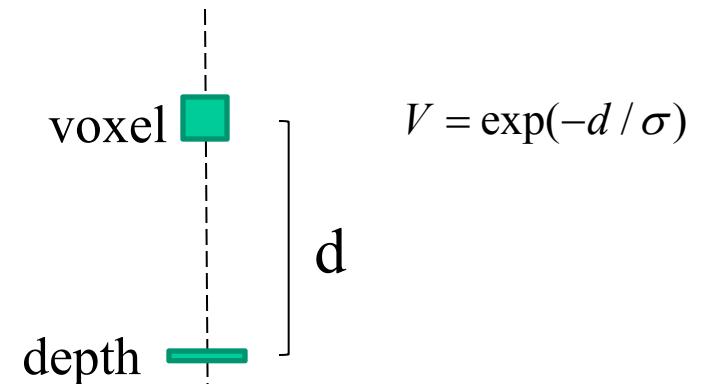
(b)



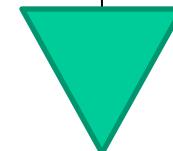
(c)



(d)



$$V = -\lambda_{empty}$$



camera

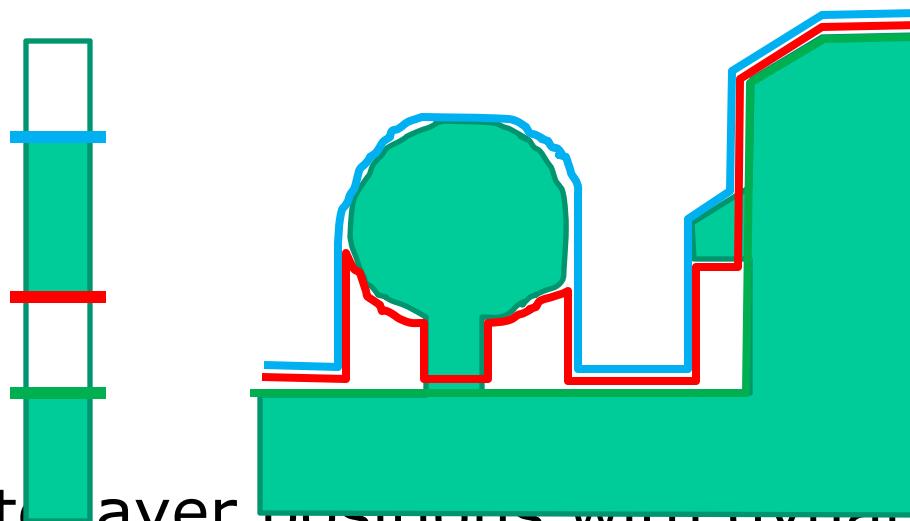
This is a
log-odds
likelihood



n-layer heightmap fusion

(Gallup et al. DAGM'10)

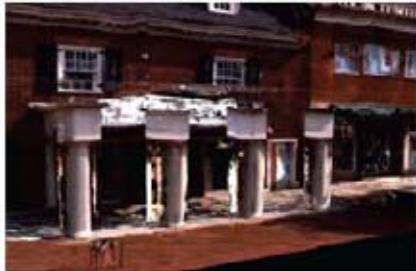
- Generalize to n-layer heightmap
- Each layer is a transition from full/empty or empty/full



- Compute layer positions with dynamic programming
- Use model selection (BIC) to determine number of layers



n-layer heightmap fusion



1 Layer



3 Layer



1 Layer



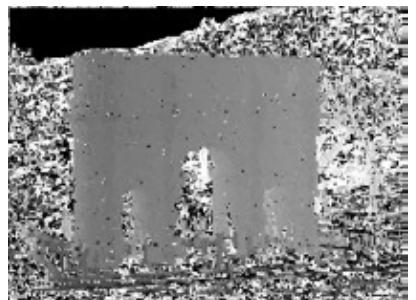
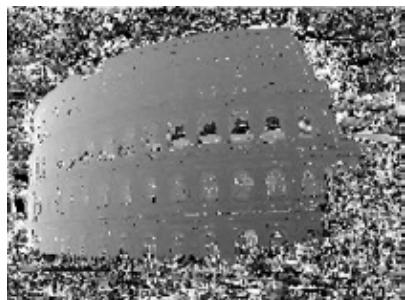
3 Layer





Reconstruction from Photo Collections

- Millions of images download from Flickr
- Camera poses computed using (Li et al. [ECCV'08](#))
- Depthmaps from GPU planesweep





Results





More on stereo ...

The Middlebury Stereo Vision Research Page

<http://cat.middlebury.edu/stereo/>

Recommended reading

D. Scharstein and R. Szeliski.

A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms.

IJCV 47(1/2/3):7-42, April-June 2002. [PDF file](#) (1.15 MB) - includes current evaluation.

Microsoft Research Technical Report MSR-TR-2001-81, November 2001. [PDF file](#) (1.27 MB).



Next week:
2D Shape Matching