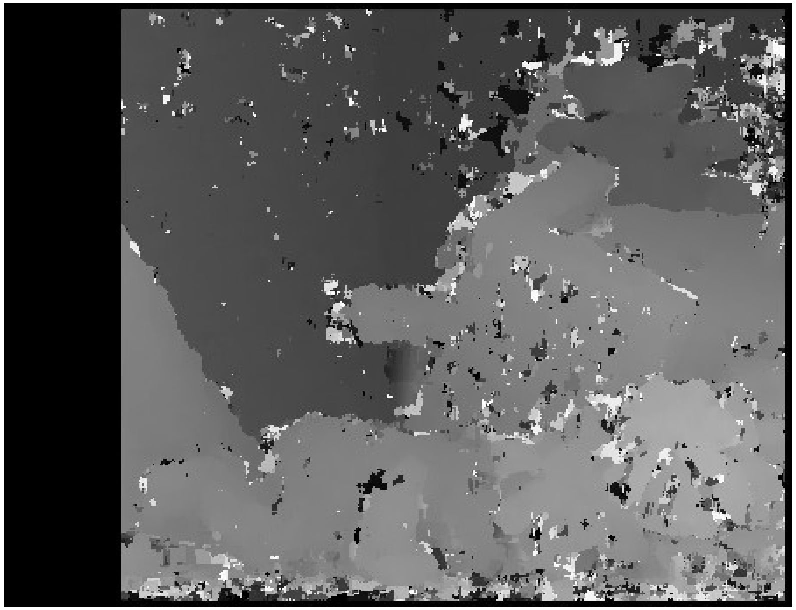
Computer Vision I

Dense Stereo Correspondences

A Basic Stereo Algorithm: Blockmatching

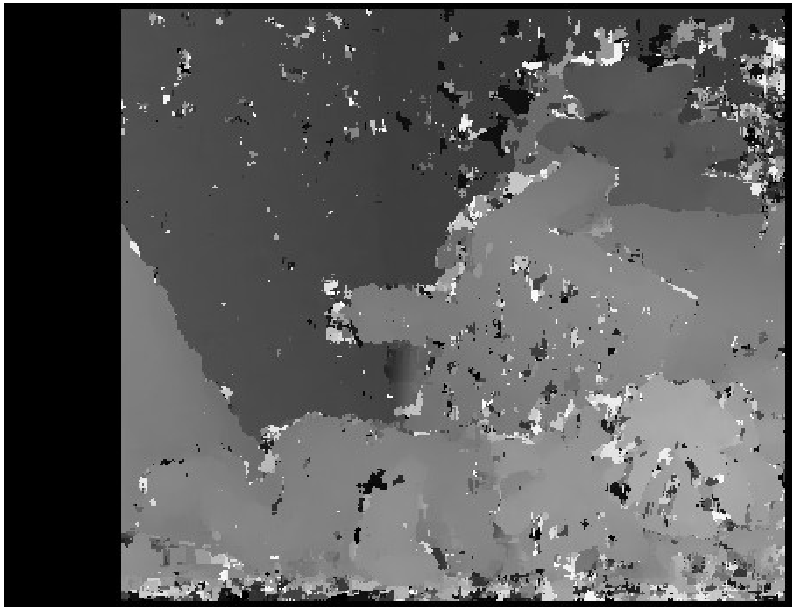
- For each pixel
  - For each disparity value
    - Compare patch similarity
    - Assign disparity with highest similarity



Result: Noisy, no smooth surfaces

A Basic Stereo Algorithm: Blockmatching

$$\forall (x,y) \in \Omega \quad \hat{d}(x,y) = argmin_{d \in D} C(x,y,d)$$



Result: Noisy, no smooth surfaces

Stereo Algorithm: Sanity Check

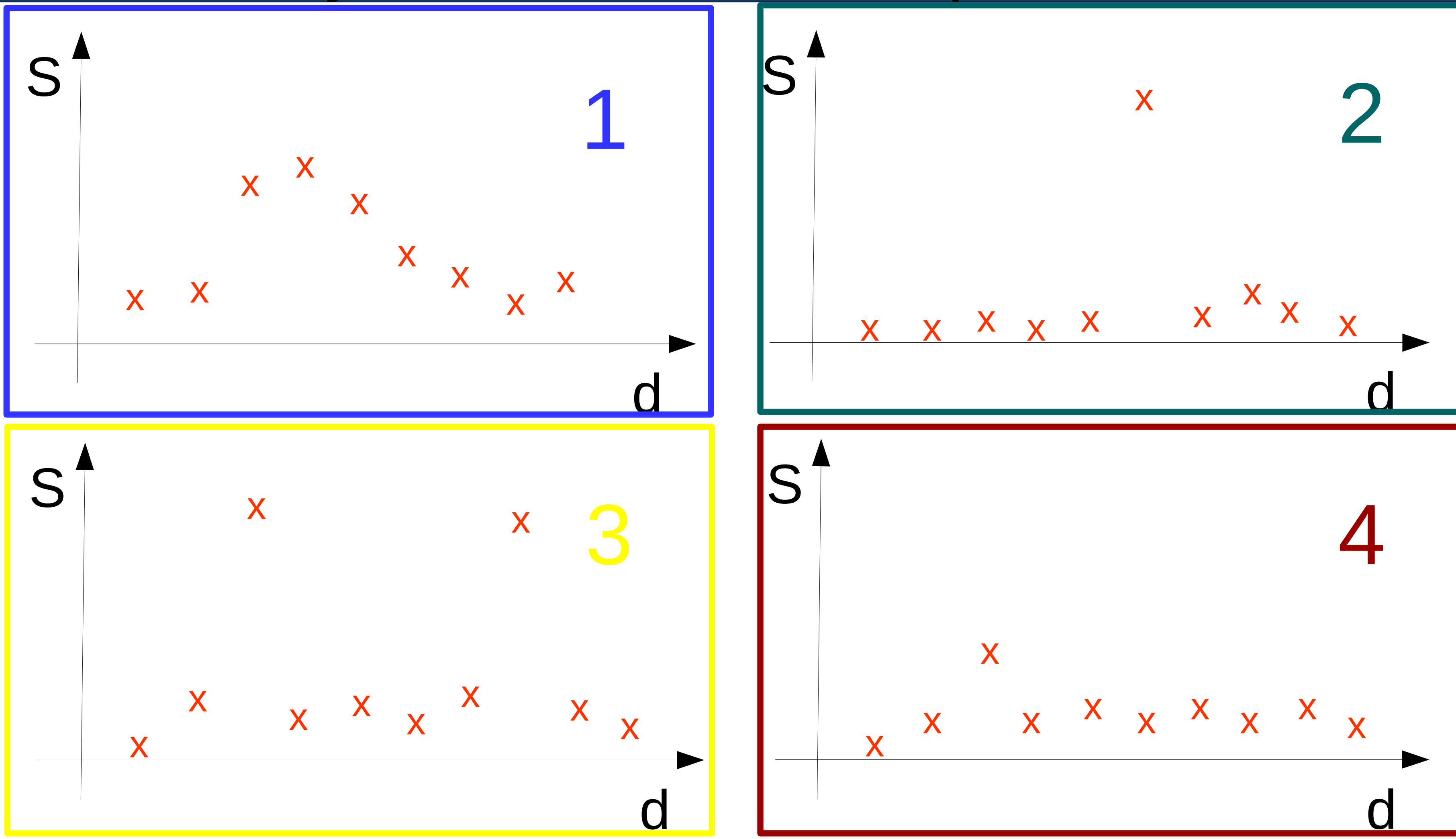
- Uniqueness constraint
  - Only one pixel should arrive at each target location

Stereo Algorithm: Sanity Check

- Left-Right consistency
  - Compute disparity with left reference image d_l
  - Compute disparity with right reference image d_r

$$|d_l(p) - d_r(q)| > 1 \qquad q = d_l(p)$$

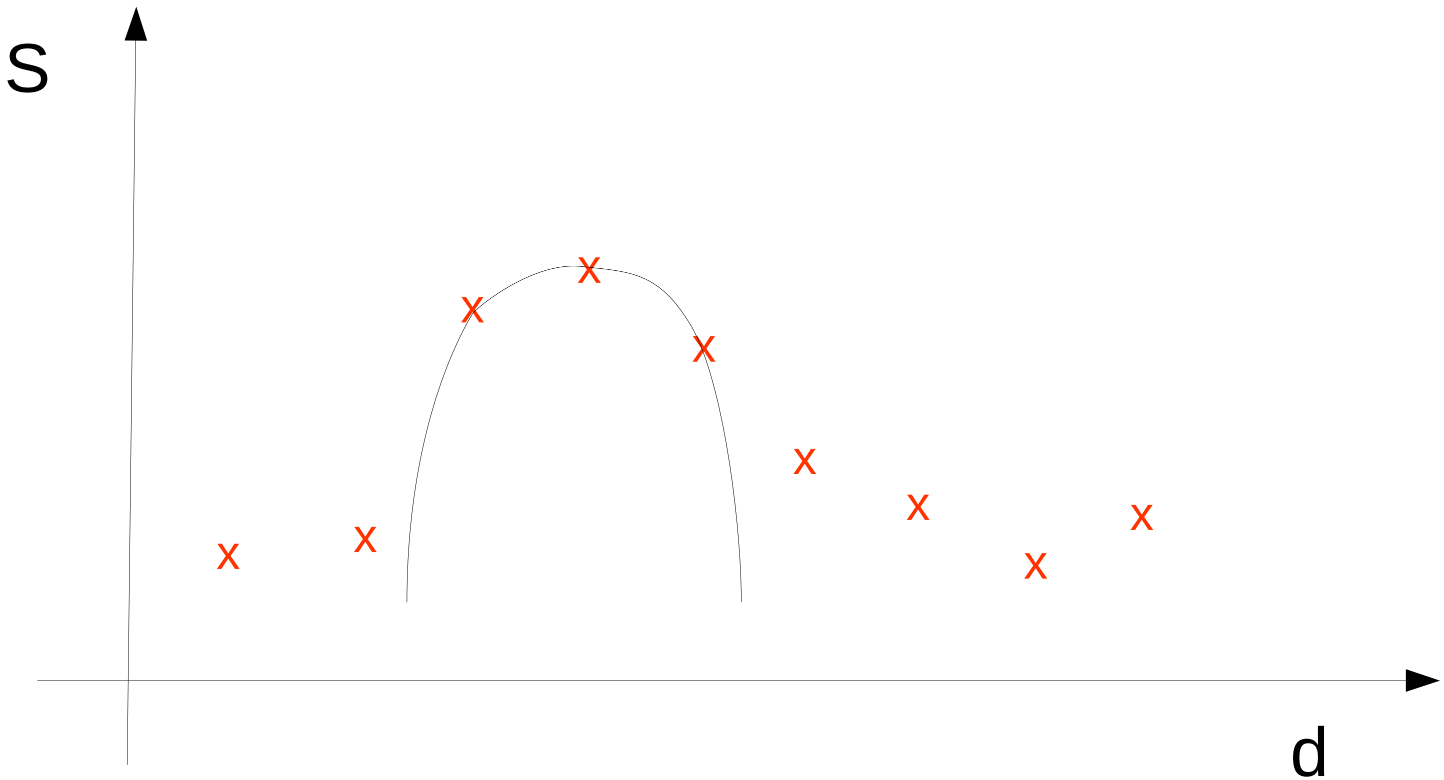
Quiz: In which situation are you confident with your stereo correspondence?



5: None

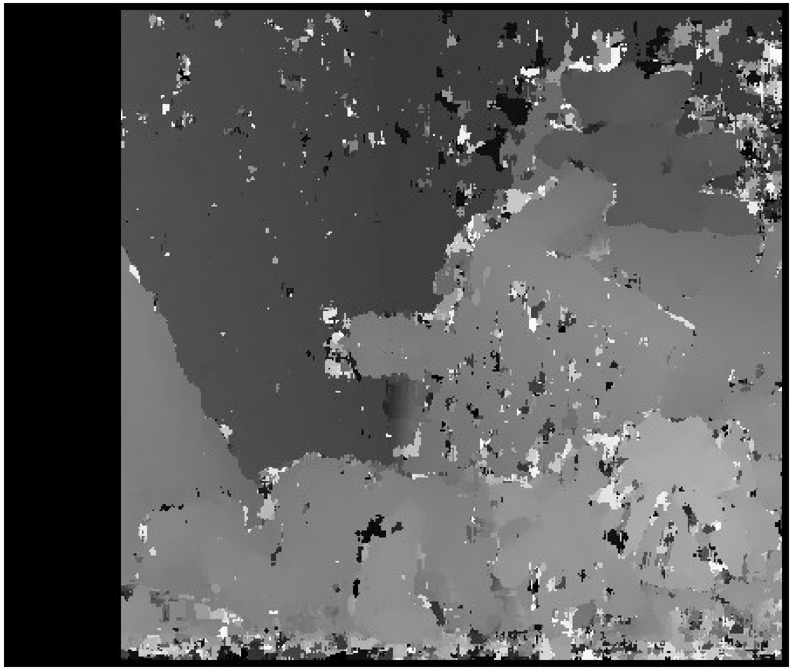
6: Don't know

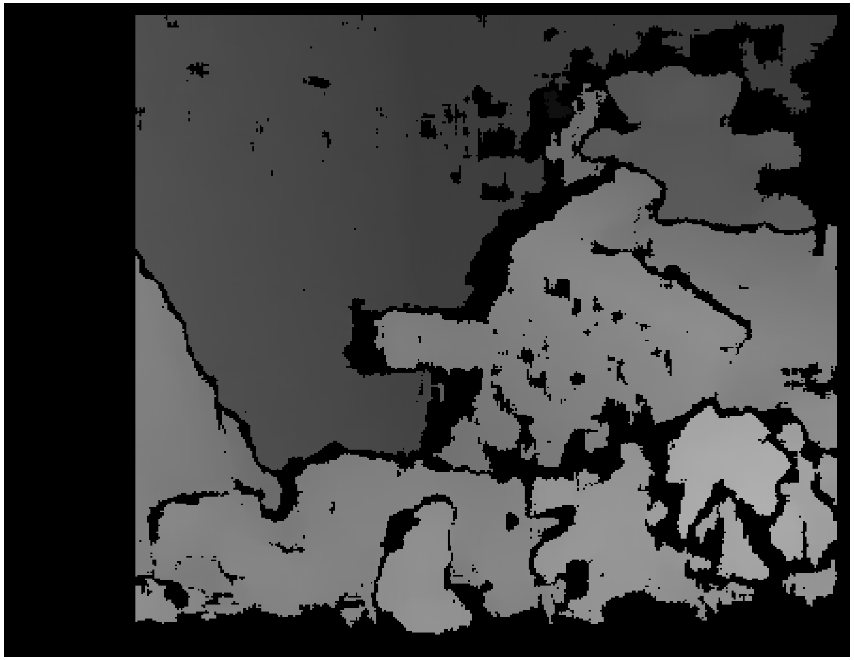
Disparity Refinement



Stereo Algorithm: Sanity Check

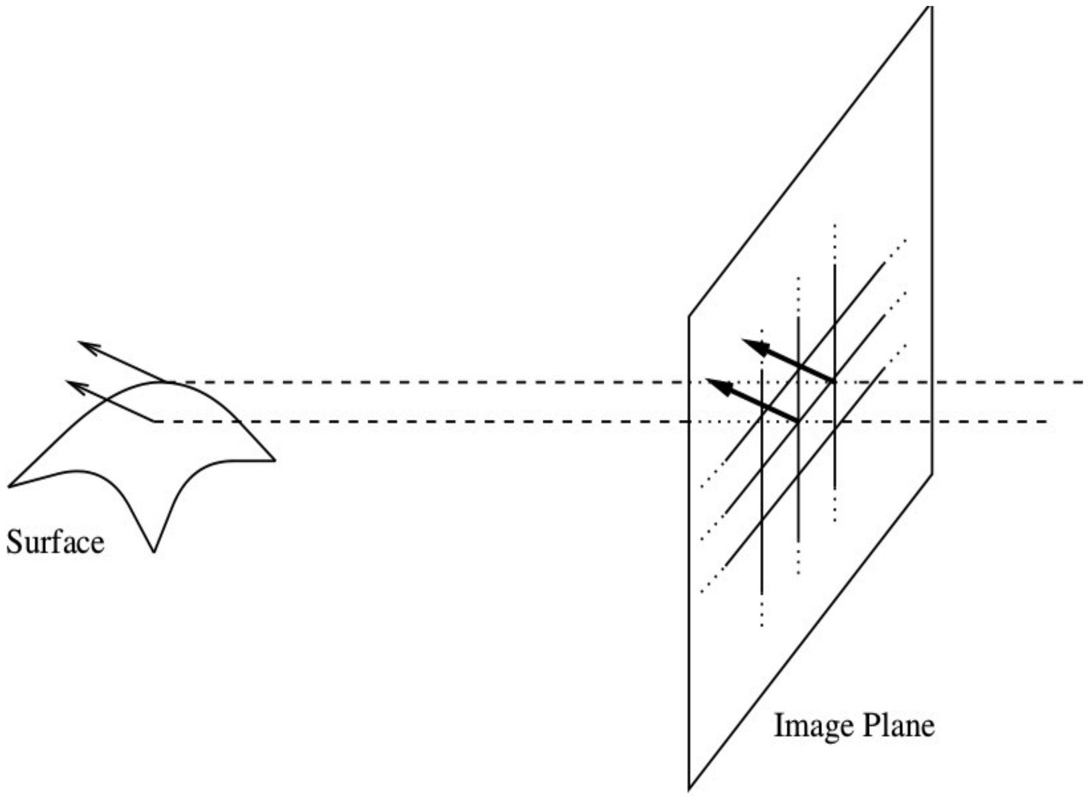
- Confidence considerations





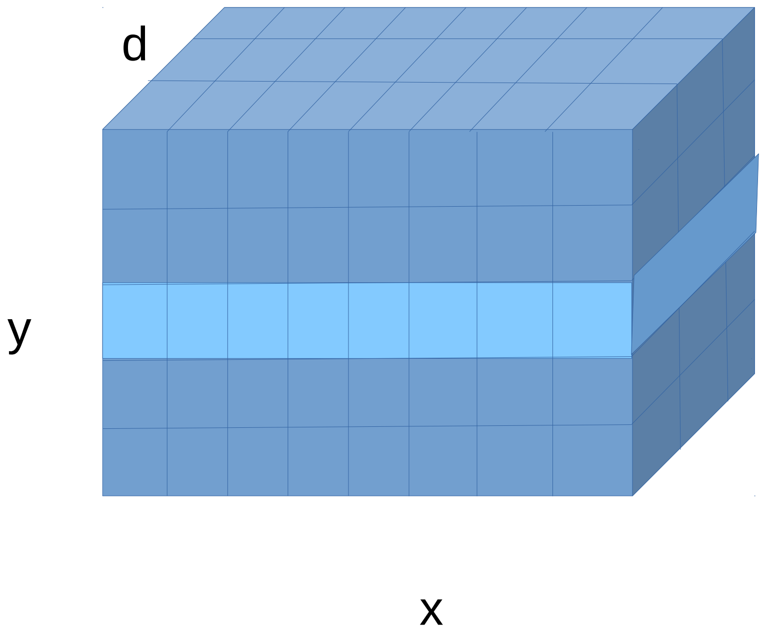
Disparity Estimation: Assumption II

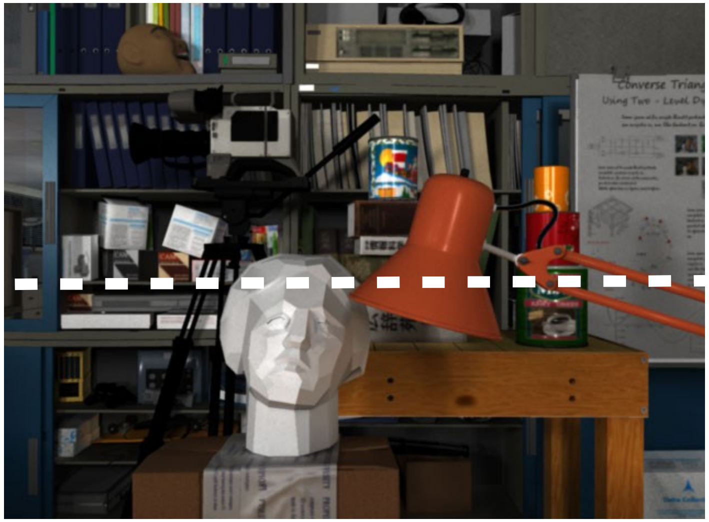
- Neighboring pixels belong to the same surface



$$d(x, y) \approx d(x + \epsilon_x, y + \epsilon_y)$$

Disparity Space Image



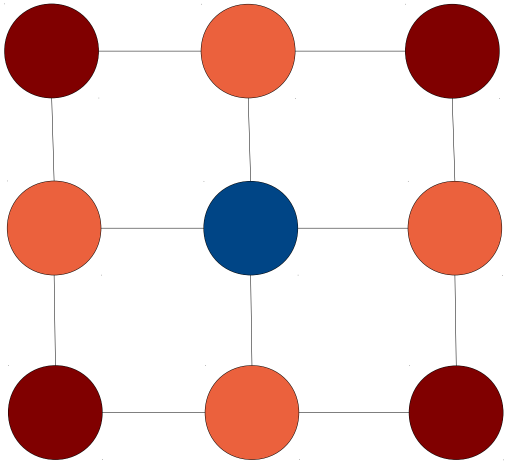


Disparity Fields

- Neighboring pixel should have similar disparity

$$\hat{d}:\Omega \rightarrow D, \quad (x,y) \rightarrow \hat{d}(x,y)$$

$$\hat{d} = argmin_{d:\Omega \rightarrow D} \sum_{(x,y)\in\Omega} \left(C(x,y,d(x,y)) + \sum_{(s,t)\in N(x,y)} \rho(d(x,y),d(s,t)) \right)$$

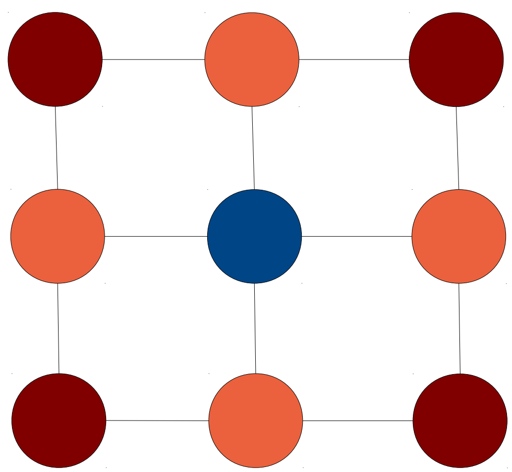


Disparity Fields

- Neighboring pixel should have similar disparity

$$\hat{d}:\Omega \rightarrow D,(x,y)\rightarrow \hat{d}(x,y)$$

$$\hat{d}=argmin_{d:\Omega\rightarrow D}\sum_{(x,y)\in\Omega}\left(C(x,y,d(x,y))+\sum_{(s,t)\in N(x,y)}\rho(d(x,y),d(s,t))\right)$$

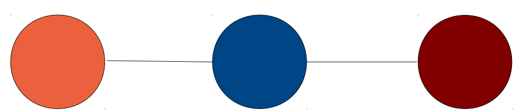


- Discrete Optimization Problem
- In general NP hard $|\Omega|^{|D|}$

Disparity Chains

- Neighboring pixel should have similar disparity
  - On an epipolar line

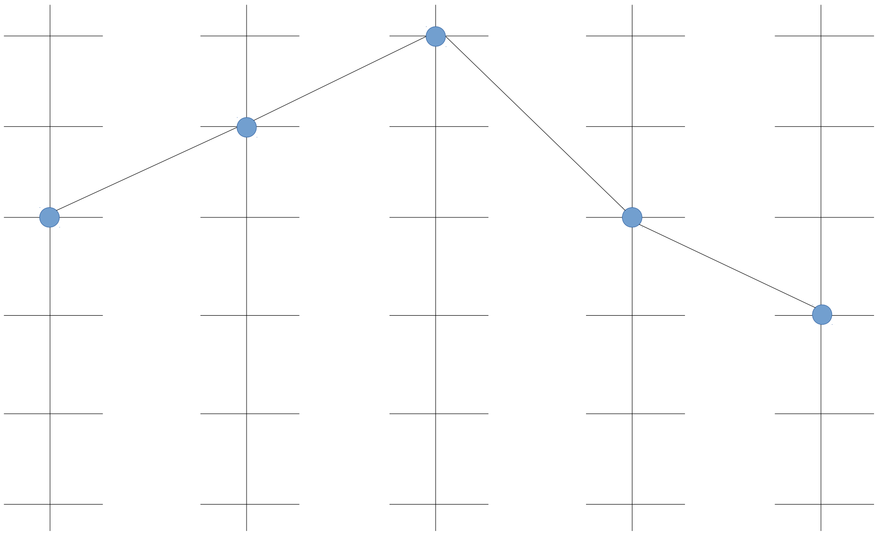
$$\hat{d} = argmin_{d:\Omega \rightarrow D} \sum_{(x,y)\in\Omega} \left(C(x,y,d(x,y)) + \rho\left(d(x,y), d(x-1,y)\right) \right)$$



Dynamic Programming

- Break the optimization down to sub-optimizations

$$E(\boldsymbol{x}) = \sum_{i=1}^{n} C(d_i) + \sum_{i=1}^{n} \rho(d_{i-1}, d_i)$$



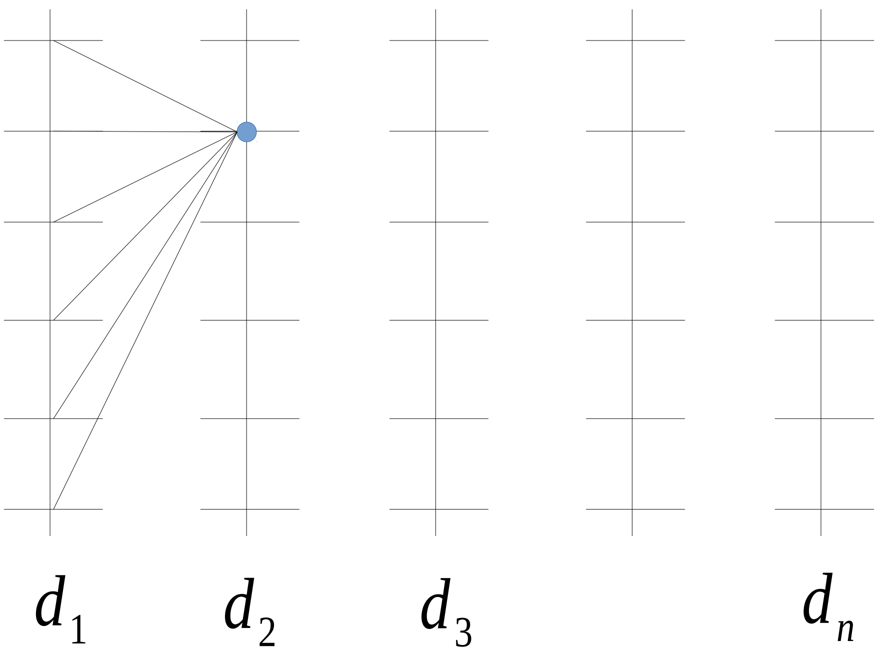
$$|\Omega|^{|D|}$$

Dynamic Programming

- Break the optimization down to sub-optimizations

$$S_2(d_2) = C(d_2) + min_{d_1}\left\{ C(d_1) + \rho(d_1, d_2) \right\}$$

Remember d_1 for which $S_2(d_2)$ is a minimum



$$|\Omega|^{|D|}$$

vs

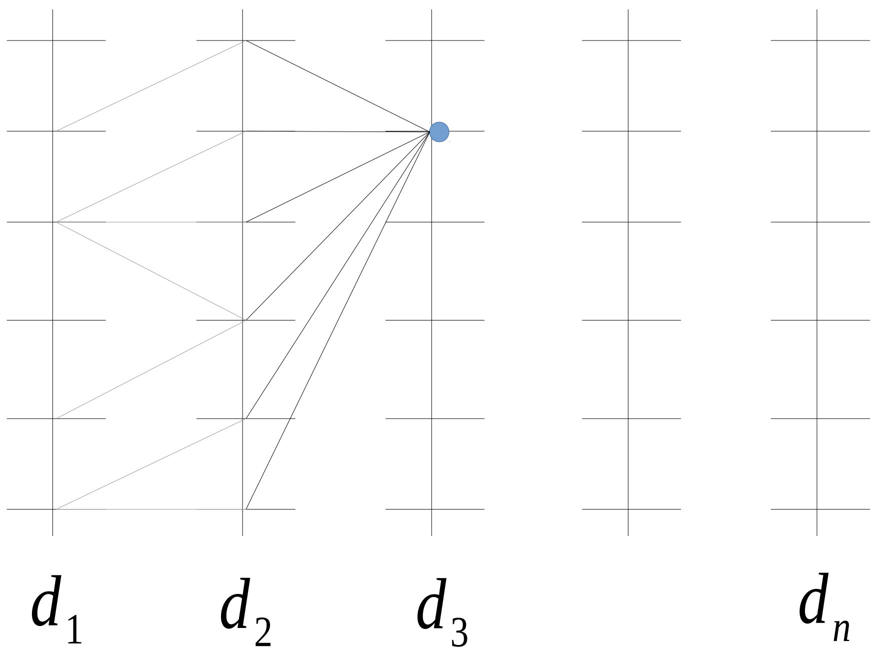
$$|D|^2 + \ldots$$

Dynamic Programming

- Break the optimization down to sub-optimizations

$$S_3(d_3) = C(d_3) + min_{d_2}\{S_2(d_2) + \rho(d_{2,}\, d_3)\}$$

Remember d_2 for which $S(d_3)$ is a minimum



$$|\Omega|^{|D|}$$

vs

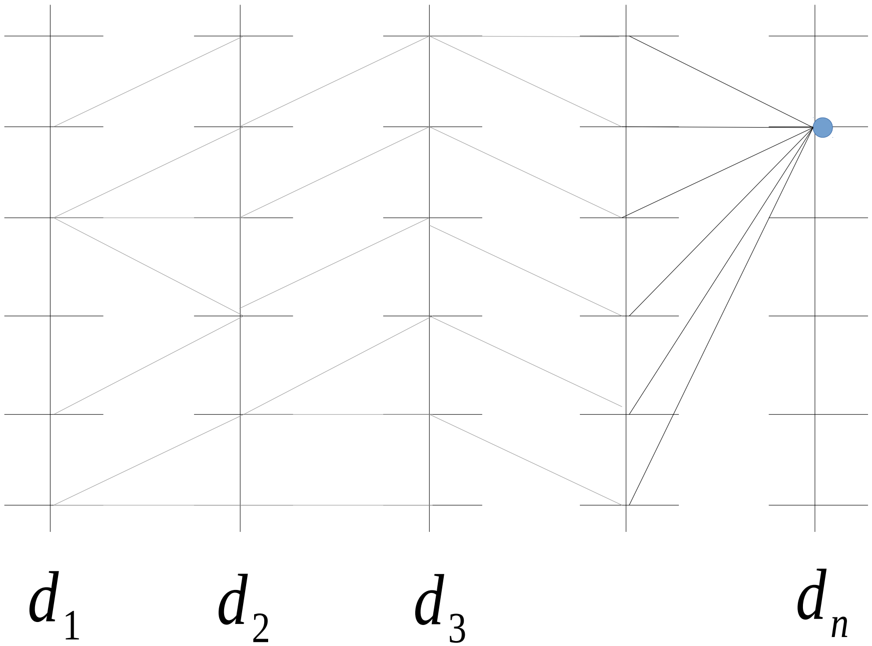
$$2|D|^2 + \ldots$$

Dynamic Programming

- Break the optimization down to sub-optimizations

$$S_n(d_n) = C(d_n) + min_{d_{n-1}} \{ S_{n-1}(d_{n-1}) + \rho(d_{n-1}, d_n) \}$$

Remember d_{n-1} for which $S_n(d_n)$ is a minimum



$$\hat{d}_n = argmin_{d_n} S_n(d_n)$$

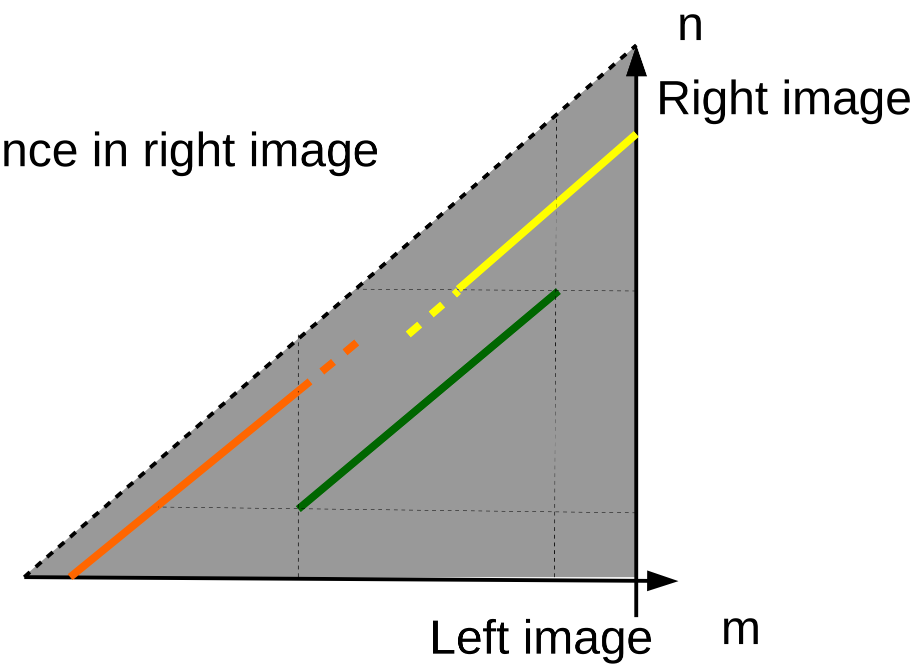
Dynamic Programming Approach

- Work on a single epipolar line

Pixels can at most have disparity 0

A point in the left image has a correspondence in right image with a smaller x-coordinate-index

Objects in front have larger disparity

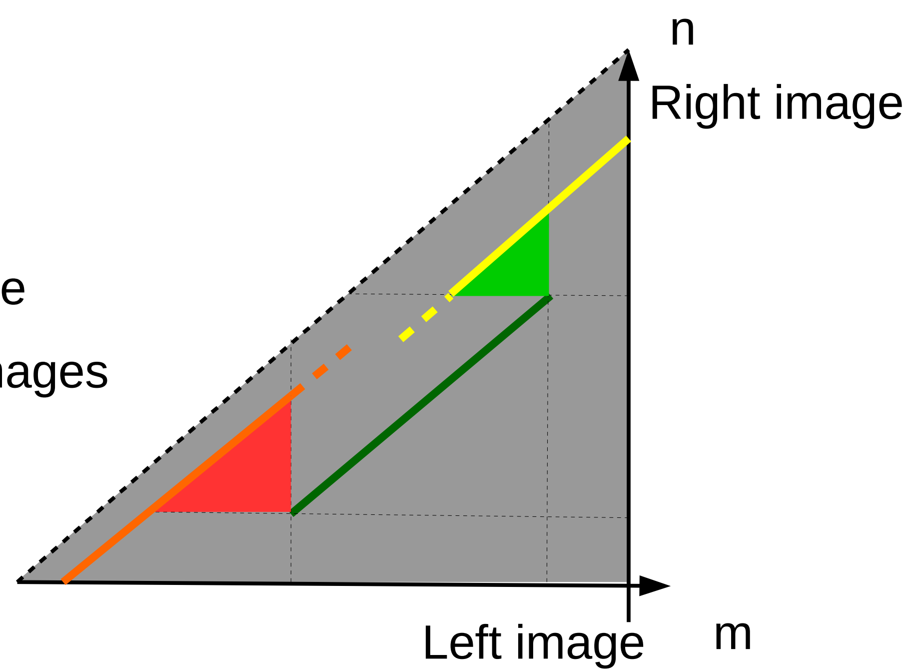


Dynamic Programming Approach

- Work on a single epipolar line

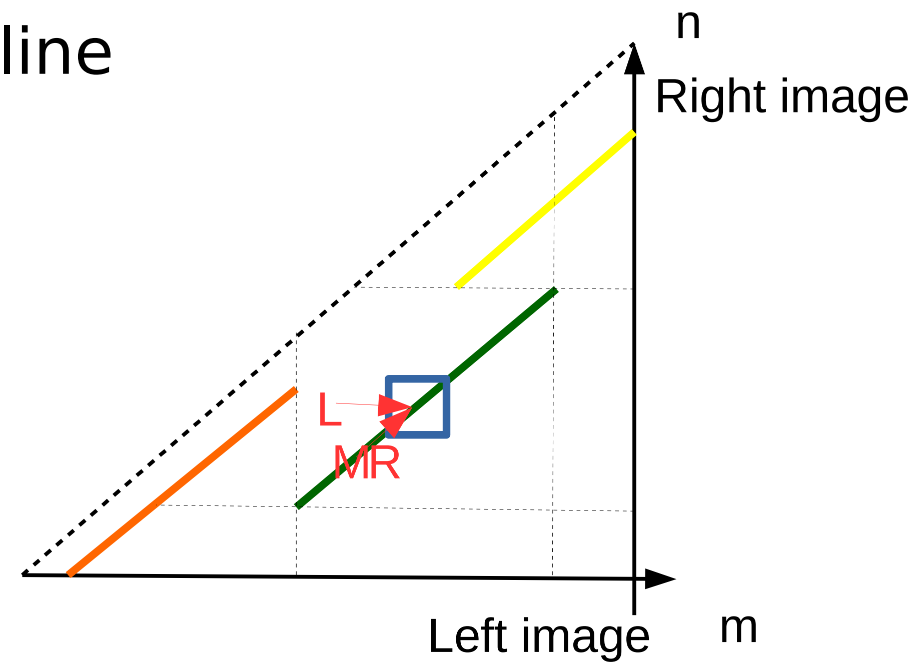
Part of the scene is visible in only one image

Part of the scene is visible in none of the images



Dynamic Programming Approach

- Work on a single epipolar line

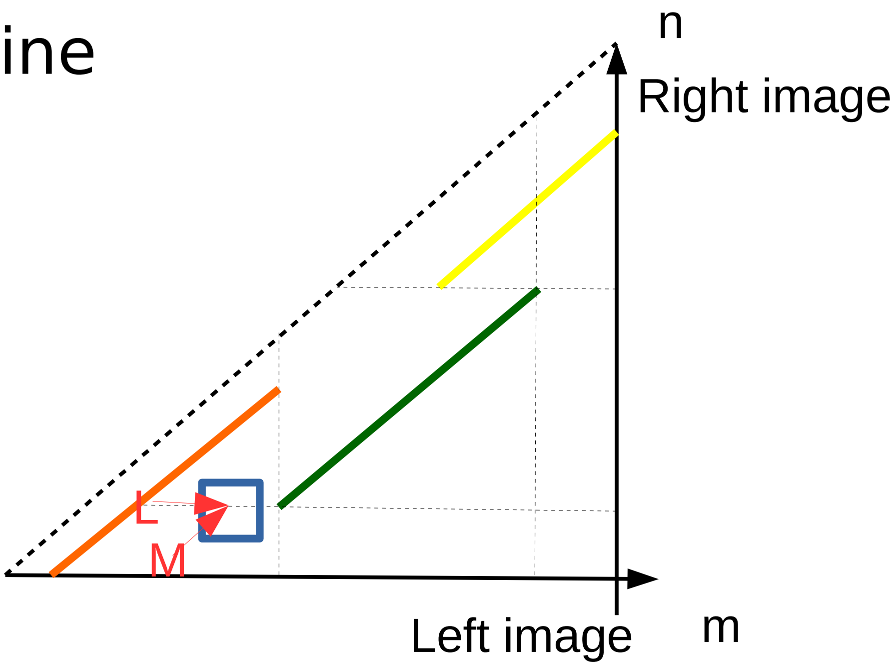


- Design cost function with case distinction
  - Match

$$S(m,n,M) =$$
$$\min\left(S(m-1,n-1,M), S(m-1,n,L), S(m-1,n-1,R)\right)$$
$$+C(m,n)$$

Dynamic Programming Approach

- Work on a single epipolar line

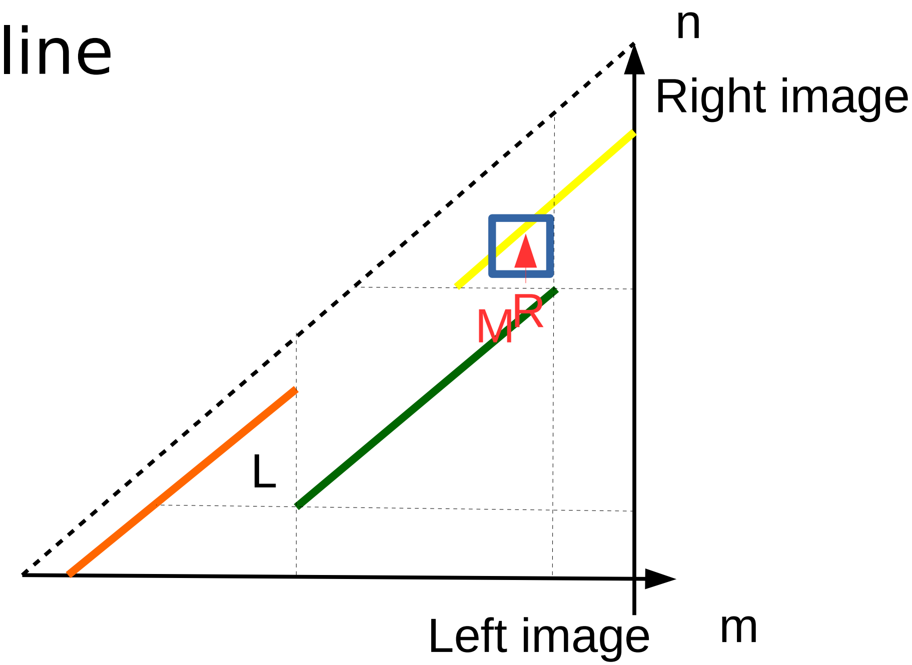


- Recursive cost function
  - Left Occlusion

$$S(m,n,L) =$$
$$\min\left(S(m-1,n-1,M),\ S(m-1,n,L)\right)$$
$$+O$$

Dynamic Programming Approach

- Work on a single epipolar line

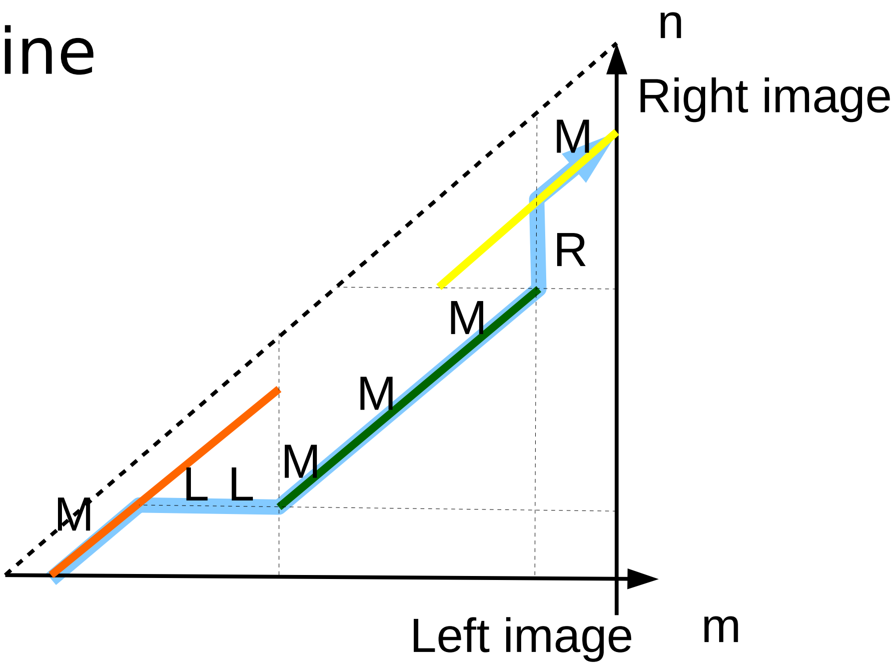


- Recursive cost function
  - Right Occlusion

$$S(m,n,R) =$$
$$\min\left(S(m,n-1,M),S(m,n-1,R)\right)$$
$$+O$$

Dynamic Programming Approach

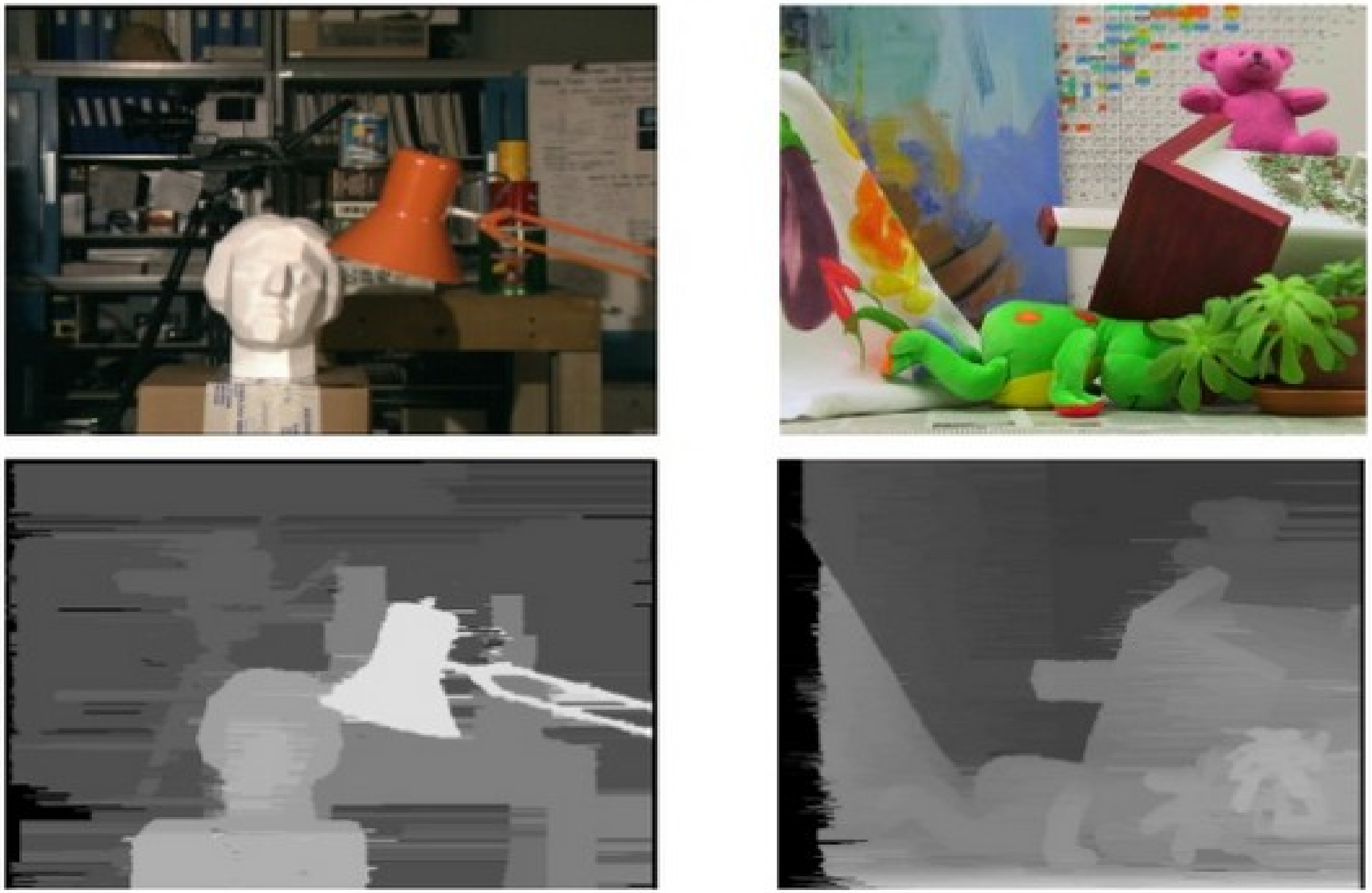
- Work on a single epipolar line



- Recursive cost function
  - Ordering constraint:
    not all constellations are possible

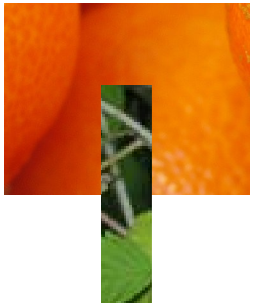
Dynamic Programming

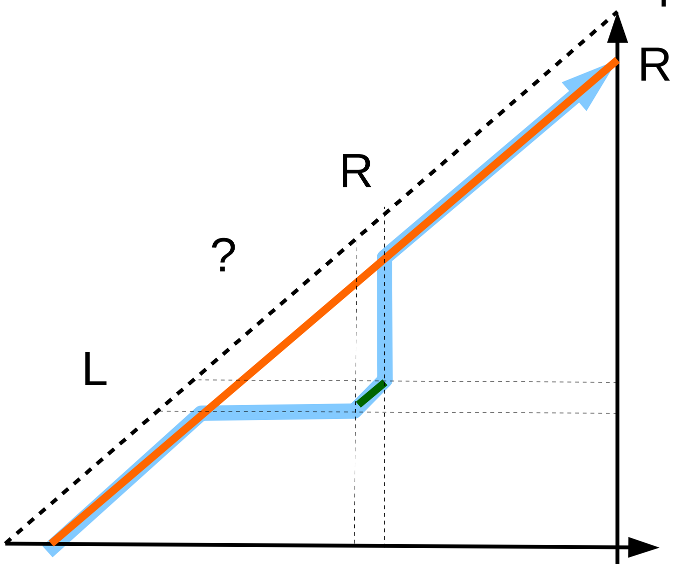
- Streaking Artifacts



Dynamic Programming Approach

- Thin Objects



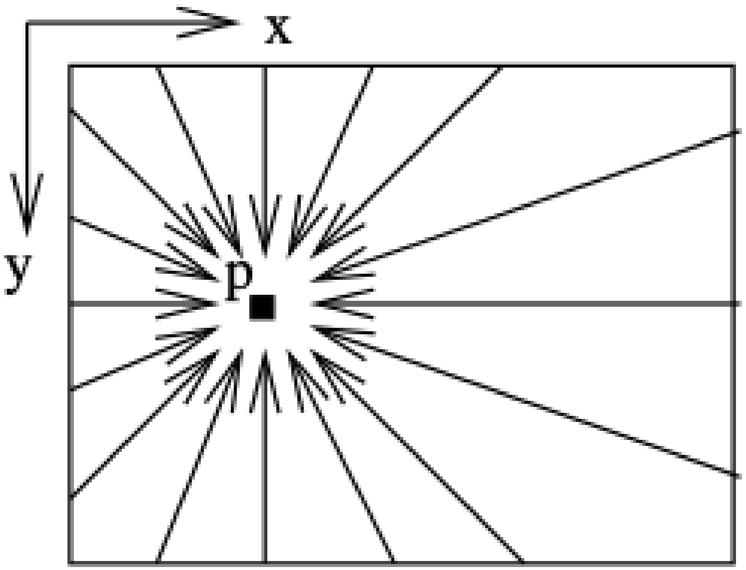


Dynamic Programming

- Pros:
  - Can be optimized in polynomial time
  - Incorporates consistency and ordering constraint
- Cons:
  - Streaking artifacts
  - Thin objects that violate ordering constraint

Semi Global Matching

- Idea: Consider more than one scanline



[Hirschmüller (CVPR 2005): Accurate and efficient stereo processing by semi-global matching and mutual information]

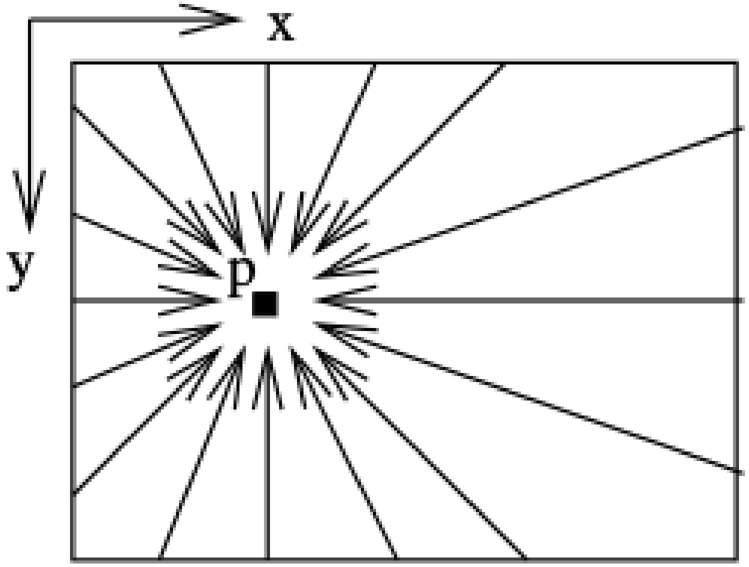
Semi Global Matching

- Idea: Consider more than one scanline
- Drop ordering constraint and allow for all possible disparity changes

$$\rho(d_p, d_q) = \begin{cases} P_1 if \, |d(p) - d(q)| = 1 \\ P_2 if \, |d(p) - d(q)| > 1 \\ 0 \quad else \end{cases}$$

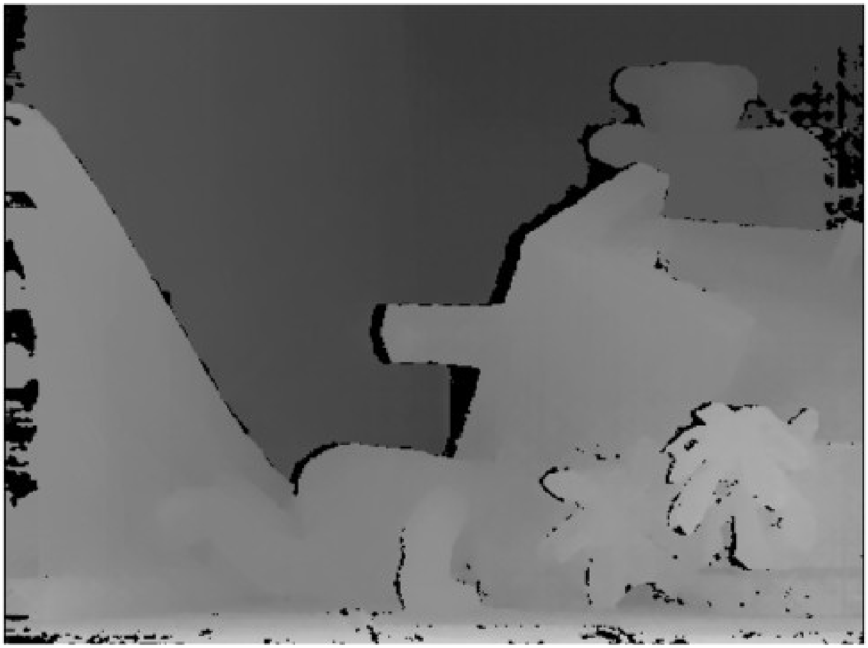
[Hirschmüller (CVPR 2005): Accurate and efficient stereo processing by semi-global matching and mutual information]

Semi Global Matching



$$L_r(p,d) = C(p,d) + min \begin{cases} L_r(p-r,d) \\ L_r(p-r,d-1) + P_1 \\ L_r(p-r,d+1) + P_1 \\ min_i L_r(p-r,i) + P_2 \end{cases}$$

Semi Global Matching



$$\hat{d}(p) = argmin \sum_{r} L_r(p, d)$$

A Global Stereo Algorithm

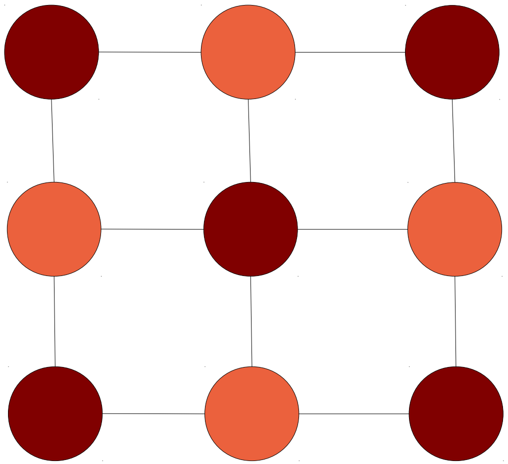
- Make the disparity of a pixel dependent on all other disparities!
  - Avoid streaking artifacts

$$\hat{d} = argmin_{d:\Omega \rightarrow D} \sum_{(x,y)\in\Omega} \Bigg(\underbrace{C(x,y,d(x,y))}_{\text{Pixel/ Patch Similarity}} + \sum_{(s,t)\in N(x,y)} \underbrace{\rho(d(x,y),d(s,t))}_{\text{Smoothness Prior}} \Bigg)$$

A Global Stereo Algorithm

- Make the disparity of a pixel dependent on all other disparities!
  - Avoid streaking artifacts

$$\hat{d} = argmin_{d:\Omega \rightarrow D} \sum_{(x,y)\in\Omega} \Big(\underbrace{C(x,y,d(x,y))}_{\text{Pixel/ Patch Similarity}} + \sum_{(s,t)\in N(x,y)} \underbrace{\rho(d(x,y),d(s,t))}_{\text{Smoothness Prior}} \Big)$$



A Global Stereo Algorithm

- Make the disparity of a pixel dependent on all other disparities!
  - Avoid streaking artifacts
  - NP-hard problem
  - Find approximate solution by using
    - Graph Cut algorithm
    - Belief Propagation

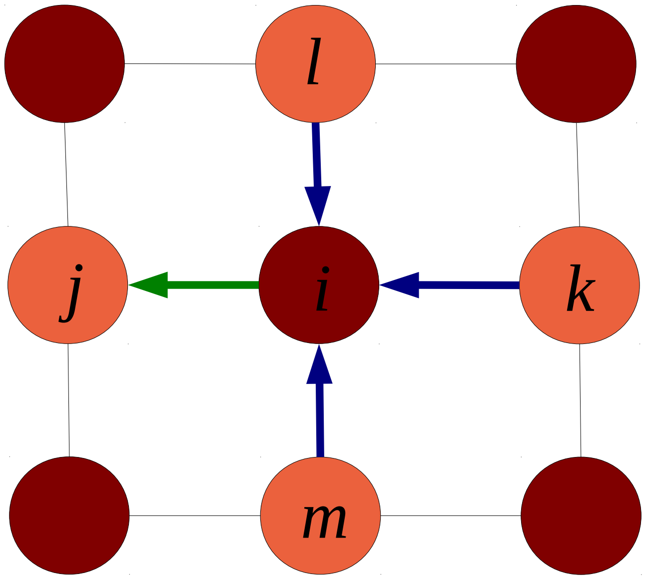
Belief Propagation Approach

- Make the disparity of a pixel dependent on all other disparities!
  - Avoid streaking artifacts
  - NP-hard problem
  - Find approximate solution by using
    - Graph Cut algorithm
    - Belief Propagation

[Tappen, Freeman: Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters]

Belief Propagation Approach

What node i beliefs about each possible state of node j

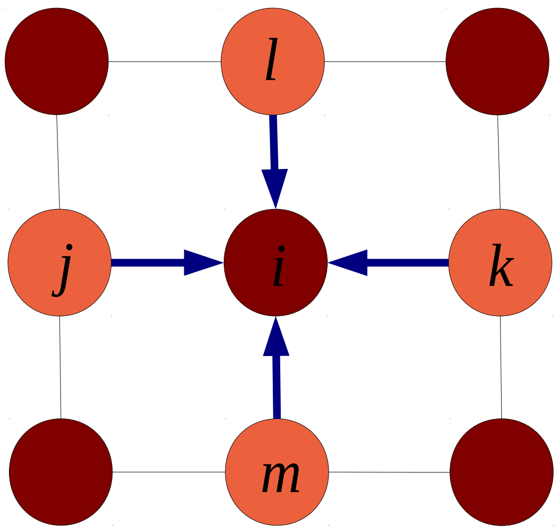


No message from j!

$$M_{i\rightarrow j}(d_j)=min_{d_i} C(d_i)+\rho(d_i,d_j)+M_{l\rightarrow i}(d_i)+M_{k\rightarrow i}(d_i)+M_{m\rightarrow i}(d_i)$$

Belief Propagation Schedule I

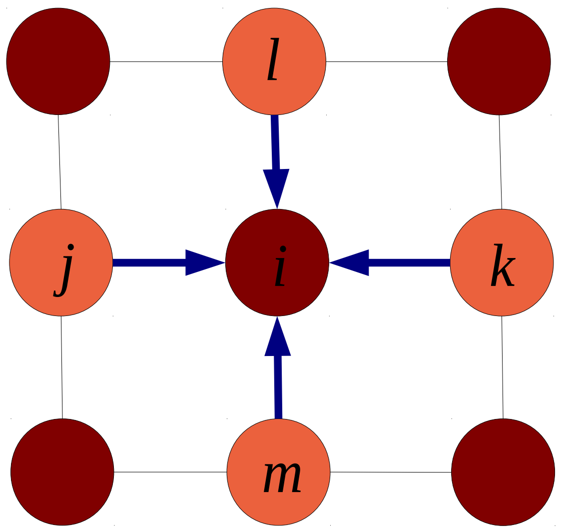
- Initialize all messages as 0
- Iterate
  - For each node: Compute all outgoing messages
  - Send messages
- Pick solution with min energy



$$\hat{d}_i = min_{d_i} C(d_i) + M_{j \to i}(d_i) + M_{l \to i}(d_i) + M_{k \to i}(d_i) + M_{m \to i}(d_i)$$

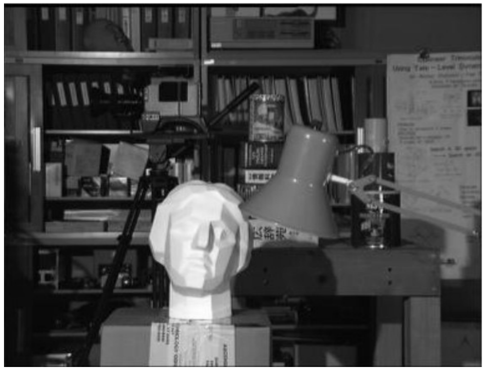
Belief Propagation Schedule II

- Initialize all messages as 0
- Iterate (change proceeding direction)
  - For each node
    - Compute all outgoing messages
    - Send message
- Pick solution with min energy



$$\hat{d}_i = min_{d_i} C(d_i) + M_{j \to i}(d_i) + M_{l \to i}(d_i) + M_{k \to i}(d_i) + M_{m \to i}(d_i)$$

Belief Propagation



Semi Global Matching

Belief Propagation

Belief Propagation Stereo

- Advantages
  - Global energy formulation
  - Arbitrary data/ smoothness term
- Disadvantages
  - Iterative
  - Approximative algorithm for energy

Steps in Correspondence Estimation

1. Matching cost computation
2. Cost aggregation
3. Disparity computation / optimization
4. Disparity refinement

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

[Scharstein, Szeliski (IJCV 2002): A taxonomy and evaluation of dense two-frame stereo correspondence algorithms]

The Limits of Stereo Computation

- Brightness constancy per pixel

or

- Brightness constancy on patch
- Occlusion
- Suitable Baseline
- Texture

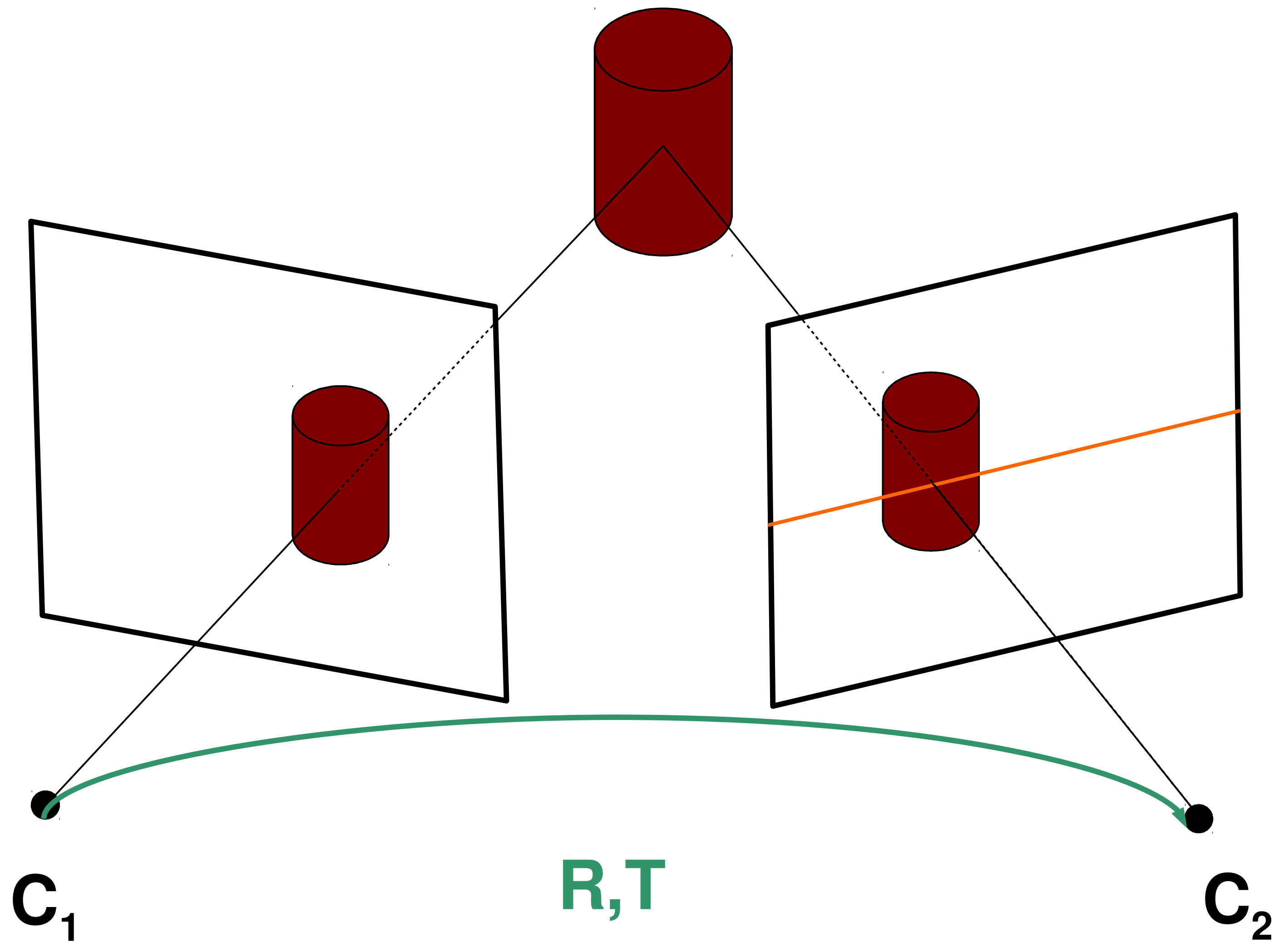
Active Depth Cameras

[Zeng (2012): Microsoft kinect sensor and its effect]

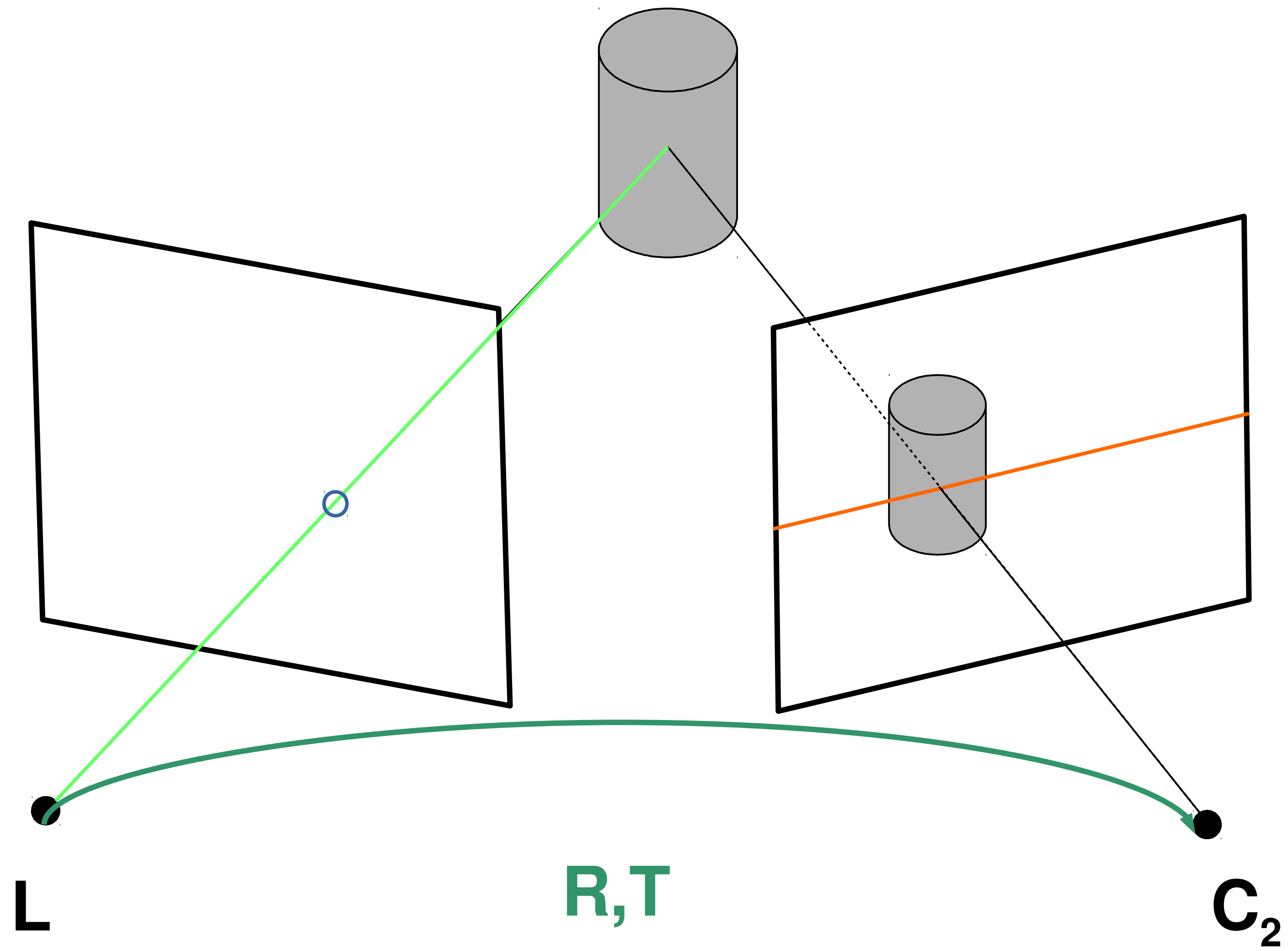
Active Depth Cameras

- Impact on consumer market
  - 10 million units sold with first 3 months
- Impact on scientific world
  - Numerous publications with RGBD images
  - Increased interest in depth cameras

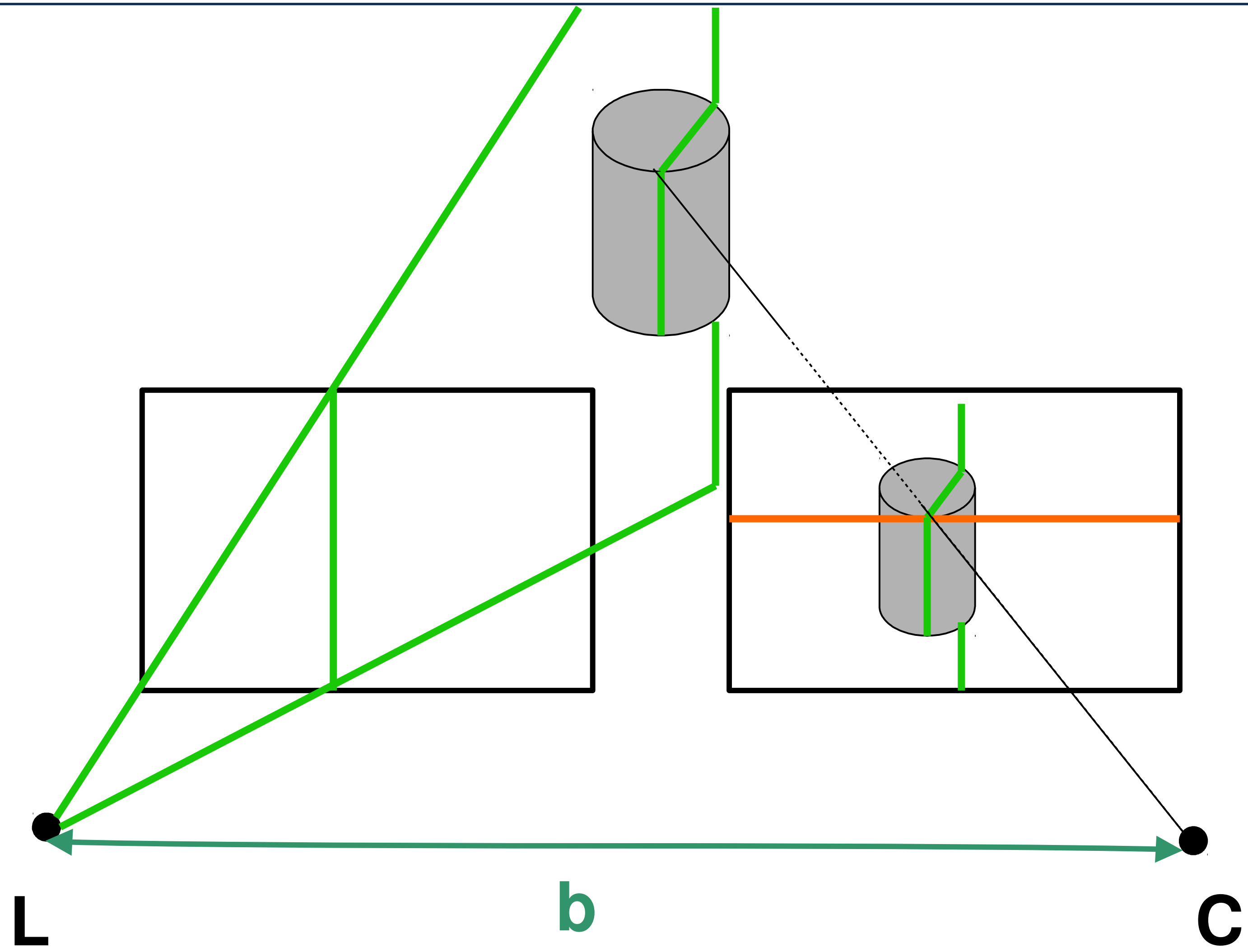
Triangulation



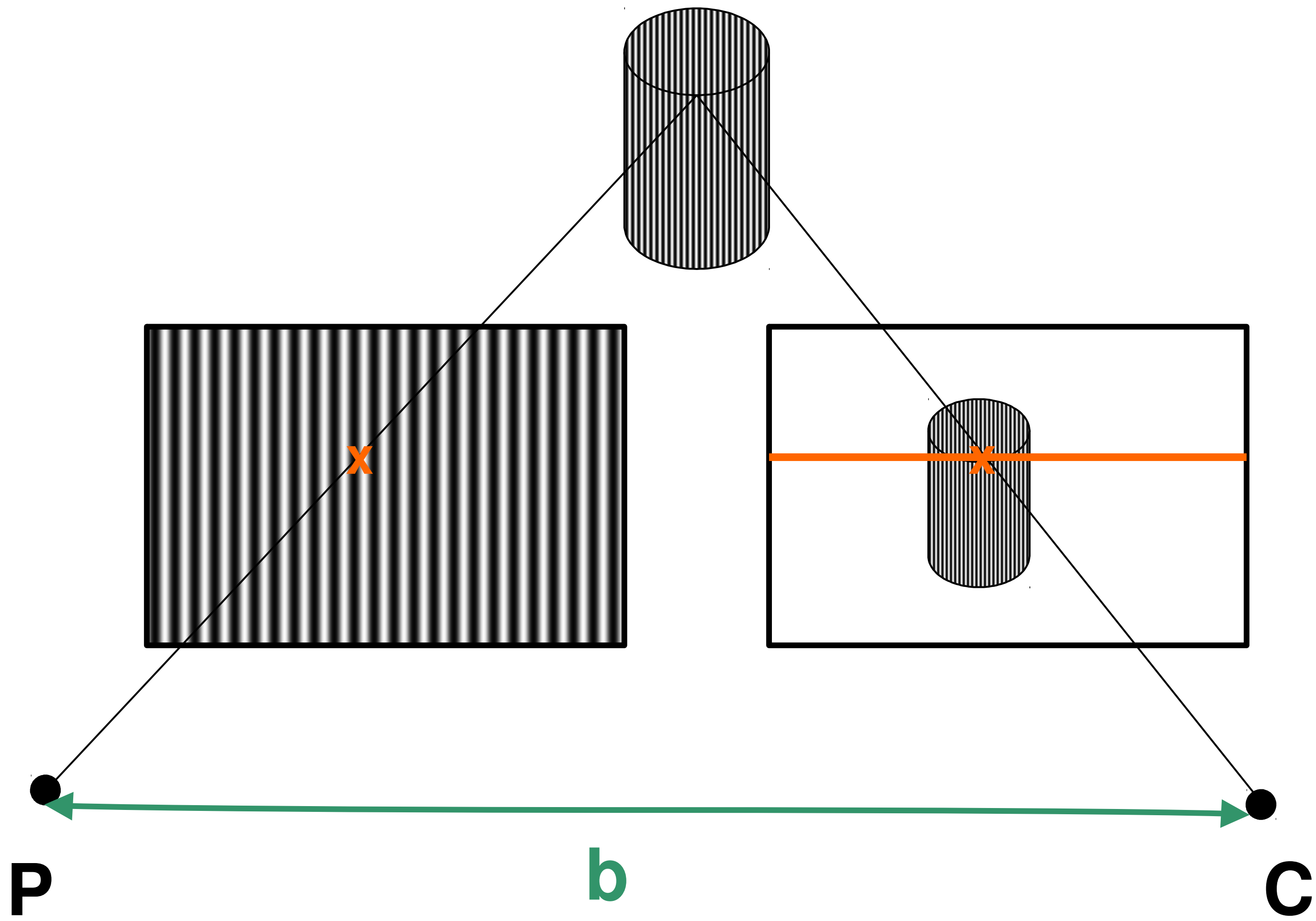
Laser Scanner



Laser Scanner



Active Depth Estimation

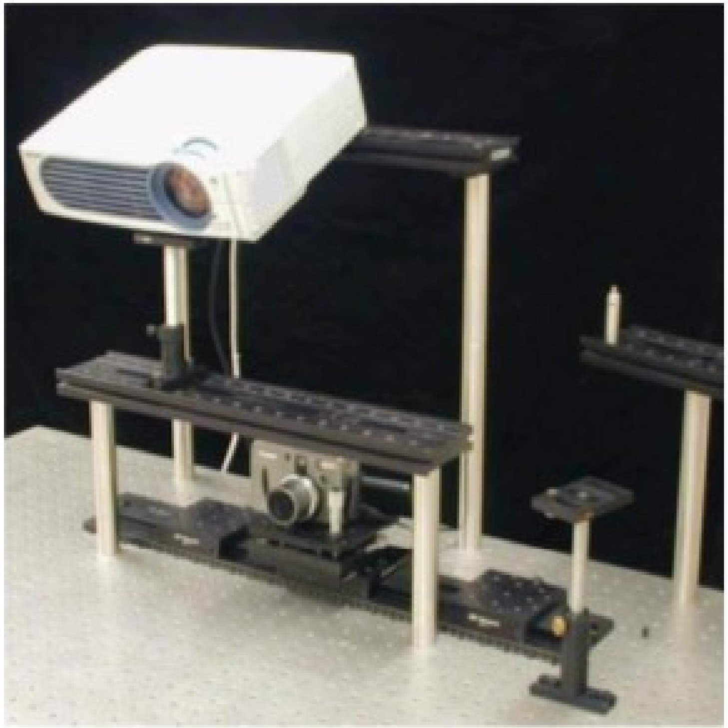


Active Depth Estimation

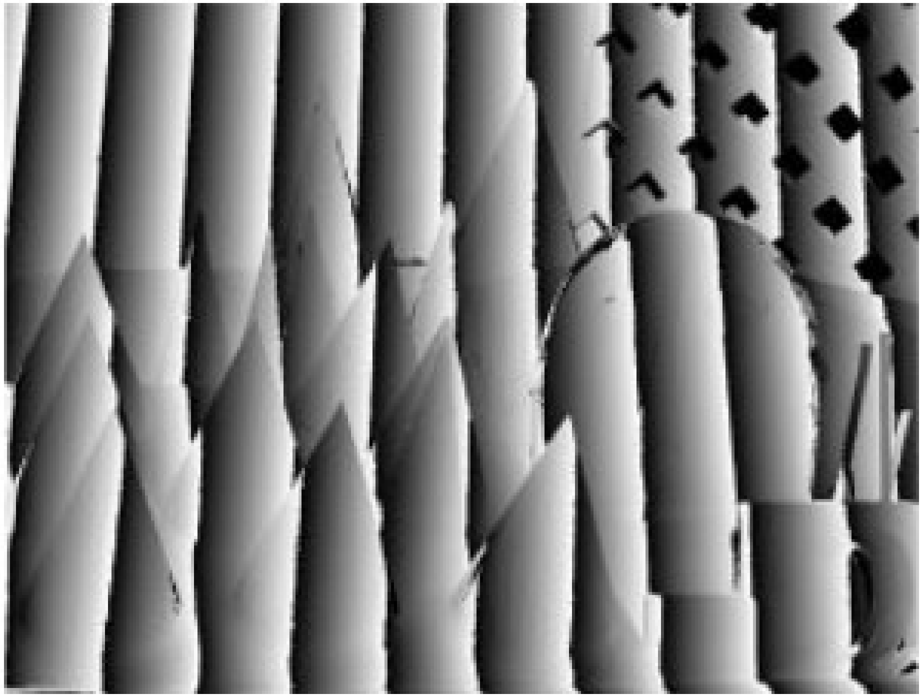
- Advantages
  - Works on un-textured objects !!!
  - Works in dark environments
- Disadvantages
  - Sensible to incident radiation (e.g. sunlight)
  - Need to generate radiation

Active Depth Estimation

Active Depth Estimation

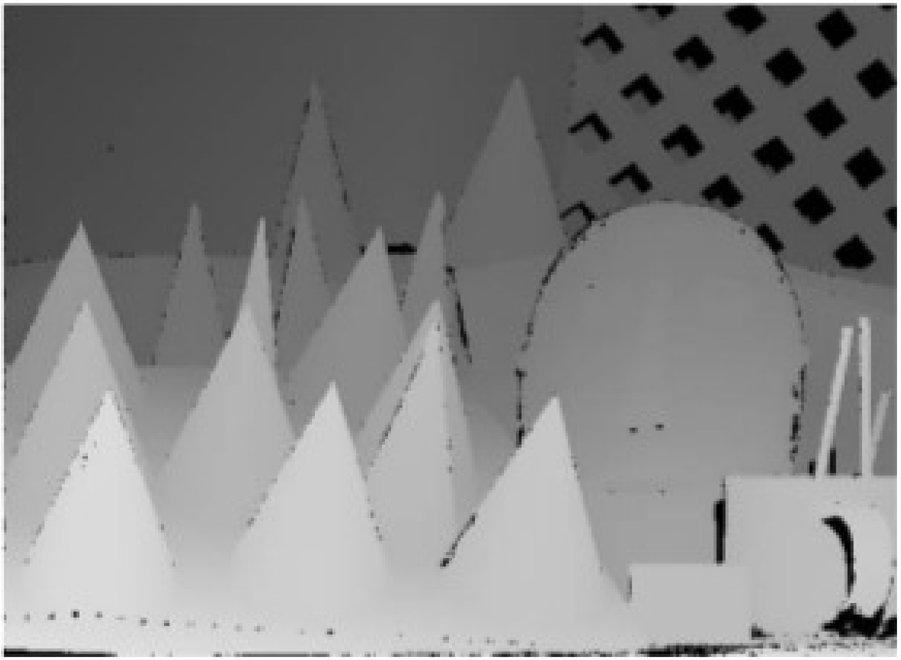


Active Depth Estimation



Active Depth Estimation

Active Depth Estimation



Active Depth Estimation

Active Depth Estimation

Stereo Algorithm Benchmark

Set: **test dense** **test sparse** **training dense** **training sparse**

Metric: **bad 0.5** **bad 1.0** **bad 2.0** **bad 4.0** **avgerr** **rms** **A50** **A90** **A95** **A99** **time** **time/MP** **time/GD**

Mask: **nonocc** **all**

plot selected **show invalid** **Reset sort**

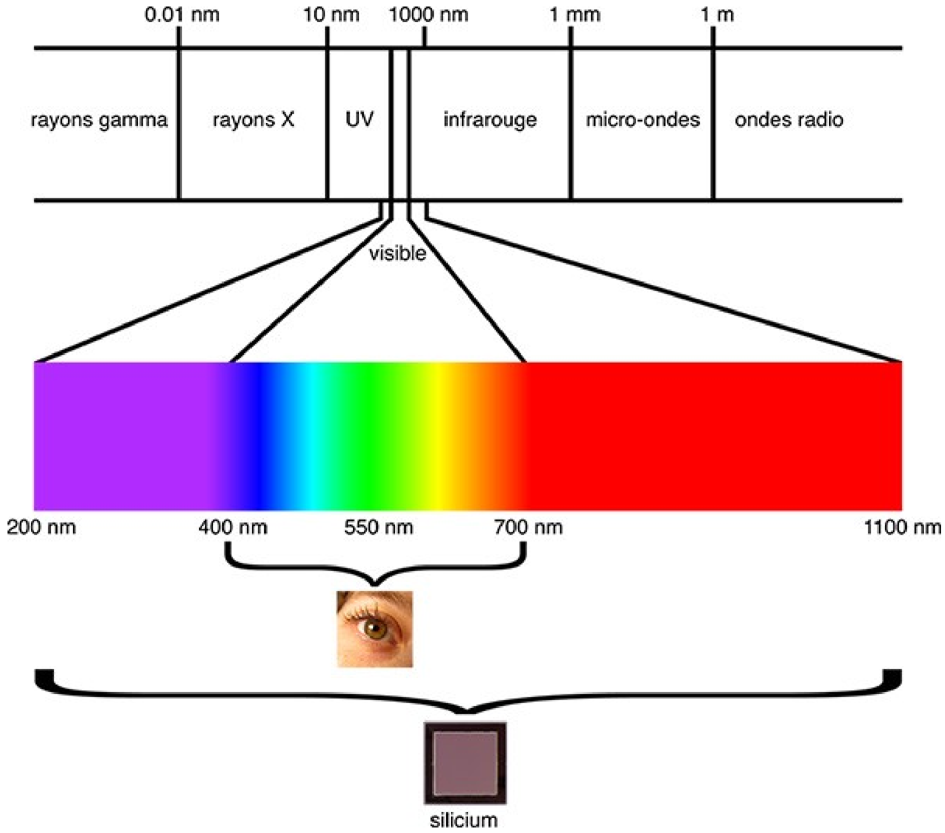
bad 2.0 (%)

Date	Name	Res	Weight Avg	Austr MP: 5.6 nd: 290 im0 im1 GT nonocc	AustrP MP: 5.6 nd: 290 im0 im1 GT nonocc	Bicyc2 MP: 5.6 nd: 250 im0 im1 GT nonocc	Class MP: 5.7 nd: 610 im0 im1 GT nonocc	ClassE MP: 5.7 nd: 610 im0 im1 GT nonocc	Compu MP: 1.5 nd: 256 im0 im1 GT nonocc	Crusa MP: 5.5 nd: 800 im0 im1 GT nonocc	CrusaP MP: 5.5 nd: 800 im0 im1 GT nonocc	Djemb MP: 5.7 nd: 320 im0 im1 GT nonocc	DjembL MP: 5.7 nd: 320 im0 im1 GT nonocc	Hoops MP: 5.7 nd: 410 im0 im1 GT nonocc	Livgrm MP: 5.9 nd: 320 im0 im1 GT nonocc	Nkuba MP: 5.5 nd: 570 im0 im1 GT nonocc	Plants MP: 5.6 nd: 320 im0 im1 GT nonocc	Stairs MP: 5.2 nd: 450 im0 im1 GT nonocc
01/19/16	NTDE	H	**7.62** 1	5.72 2	4.36 4	**5.92** 1	**2.83** 1	**10.4** 1	**8.02** 1	**5.30** 1	**5.54** 1	**2.40** 1	**13.5** 1	**14.1** 1	12.6 2	13.9 3	**6.39** 1	12.2 3
08/28/15	MC-CNN-acrt	H	8.29 2	**5.59** 1	4.55 7	5.96 2	**2.83** 1	11.4 4	8.44 3	8.32 2	8.89 4	2.71 2	16.3 3	14.1 2	13.2 4	**13.0** 1	6.40 2	11.1 2
11/03/15	MC-CNN+FBS	H	8.62 3	6.05 4	5.16 11	6.24 3	3.27 3	11.1 3	8.91 4	8.87 3	9.83 7	3.21 5	15.1 2	15.9 3	12.8 3	13.5 2	7.04 3	**9.99** 1
10/13/15	MDP	H	12.6 4	14.4 8	4.99 10	10.6 14	10.7 6	27.2 6	8.11 2	12.5 8	8.07 3	4.27 6	30.4 11	20.5 5	**12.6** 1	17.8 5	13.4 7	17.3 5
04/19/15	MeshStereo	H	13.4 5	5.90 3	4.88 8	10.8 15	12.9 11	10.6 2	13.6 8	12.2 7	9.01 5	5.39 10	27.4 5	23.5 8	17.7 6	21.0 12	15.4 11	20.9 8
11/06/15	SOU4P-net	H	13.5 6	23.1 12	5.41 13	6.39 4	13.1 12	30.5 10	11.1 5	16.4 11	12.7 12	3.13 3	28.9 7	17.1 4	16.4 5	16.9 4	10.7 5	14.5 4
12/18/15	INTS	H	14.8 7	20.2 10	4.52 5	8.62 10	11.6 7	29.5 8	13.7 9	16.4 10	10.3 9	4.69 7	27.6 6	22.5 6	20.7 9	20.5 10	11.5 6	24.9 11
11/05/15	GCSVR	H	14.8 8	17.1 9	**3.50** 1	8.22 9	16.5 17	47.4 17	11.4 6	9.75 5	7.06 2	3.17 4	34.4 15	27.1 10	18.3 7	19.2 8	16.0 12	19.3 7
11/12/14	LCU	Q	17.0 9	24.7 13	7.59 16	11.6 17	11.9 8	27.9 7	14.0 10	19.3 12	15.8 14	8.10 20	36.1 16	29.1 13	21.3 11	18.4 6	14.1 8	23.8 10
04/17/15	TMAP	H	17.1 10	20.2 11	4.94 9	8.13 8	12.8 10	30.0 9	14.1 12	27.9 17	20.4 18	5.09 8	31.5 13	23.1 7	20.9 10	19.0 7	18.8 16	18.0 6
10/31/15	SPS	F	18.2 11	12.1 7	11.4 22	13.3 21	12.1 9	15.7 5	17.0 21	14.0 9	14.3 13	8.31 21	30.3 10	32.3 15	30.0 26	25.2 21	23.7 21	26.2 14
10/07/14	IDR	H	18.4 12	37.5 20	4.08 2	7.49 6	23.3 20	40.6 13	15.7 19	24.5 13	11.3 11	5.46 12	33.1 14	26.0 9	21.5 12	21.7 13	15.3 10	21.2 9

Scanning Humans

Light Spectrum

- CCD chips have a broader range than eyes



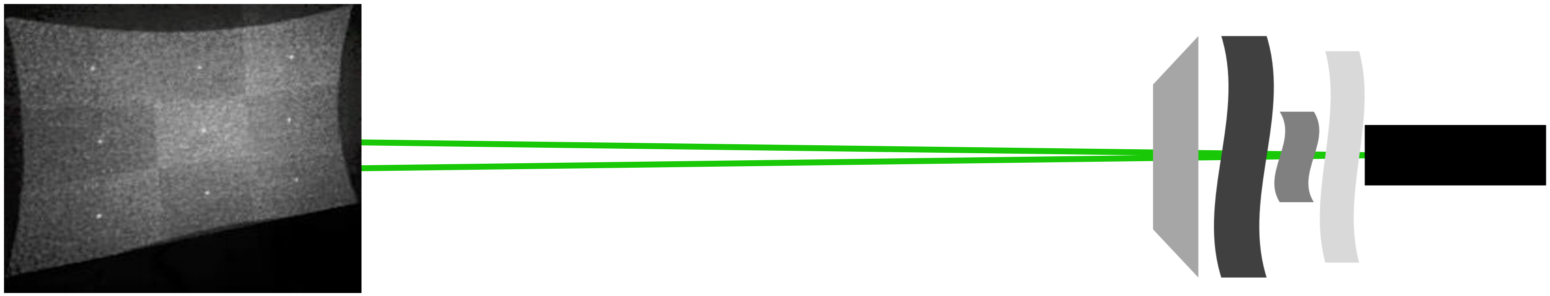
Near Infrared Imagery

[Image thanks to D. Armstrong]

The Hot Mirror

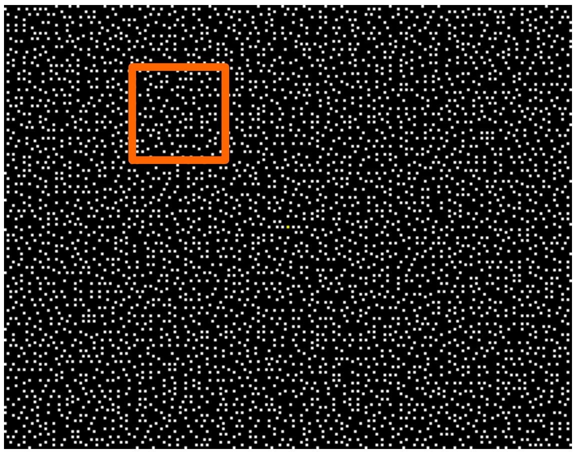
Active Depth Cameras

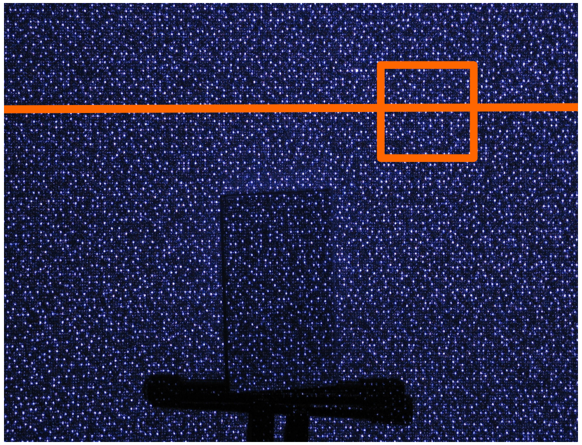
- Laser light source with coherent light
- Ground glass as diffuser → Random speckle pattern
- Optical devices → Constant pattern



Depth Estimation Algorithm

- Compare to reference pattern
- Normalized cross correlation on window (e.g. 16x16)

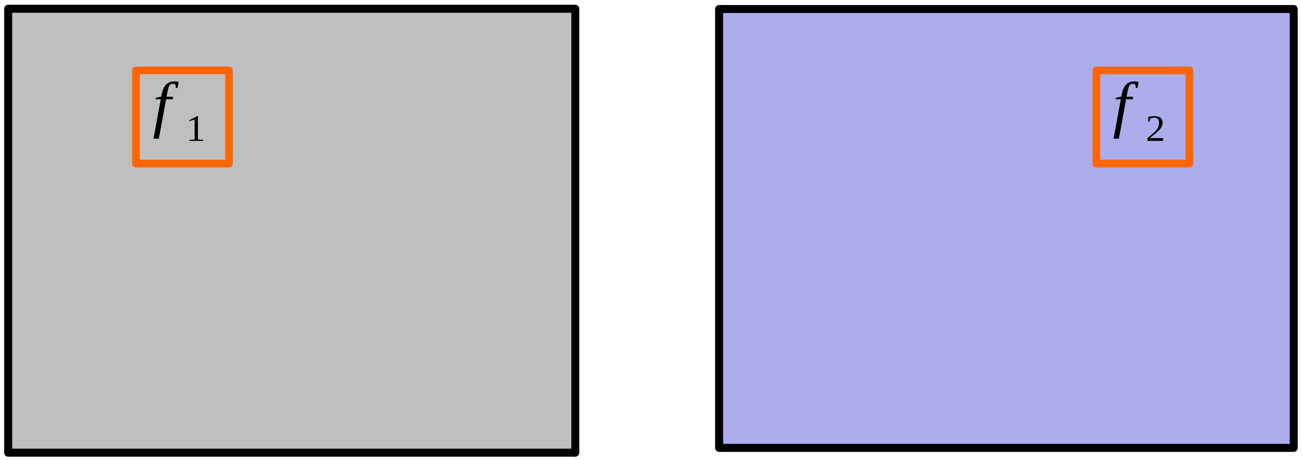




[R. Jain, K. Rangachar, B. Schunck (1995): *Machine vision*]

Depth Estimation Algorithm

- Compare to reference pattern
- Normalized cross correlation on window (e.g. 16x16)



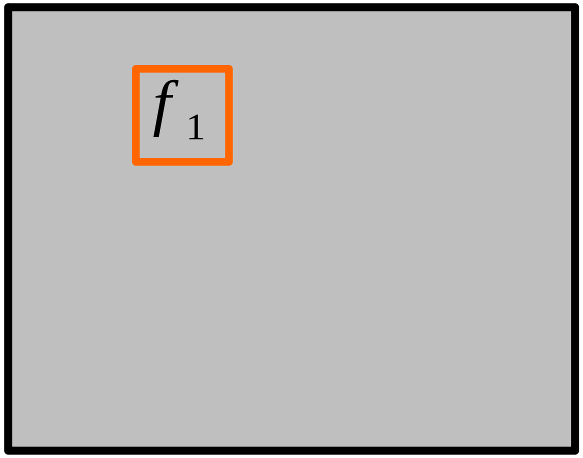
- Subtract mean $F = f_1 - \bar{f}$ $\quad G = f_2 - \bar{g}$
- Divide by variance $\sigma_f = \sqrt{\sum_i F_i^2}$ $\quad \sigma_g = \sqrt{\sum_i G_i^2}$

→ Invariant to brightness variations

[R. Jain, K. Rangachar, B. Schunck (1995): *Machine vision*]

Depth Estimation Algorithm

- Compare to reference pattern
- Normalized cross correlation on window (e.g. 16x16)

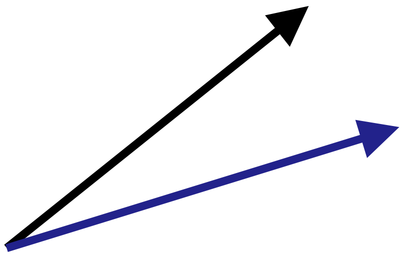


$$F = f_1 - \bar{f} \qquad G = f_2 - \bar{g}$$

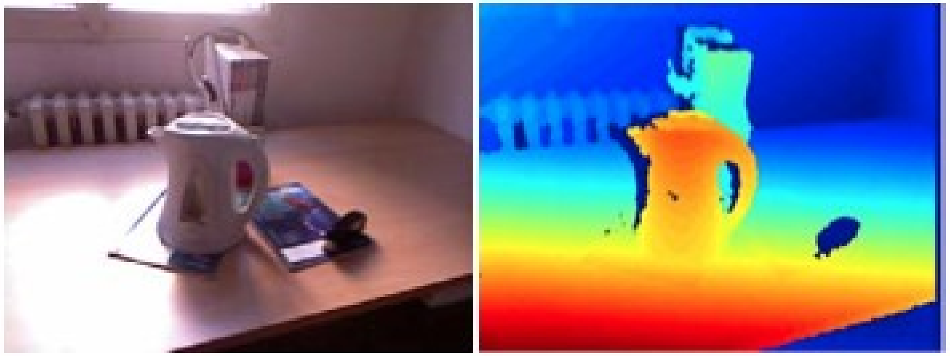
$$\sigma_f = \sqrt{\sum_i F_i^2} \qquad \sigma_g = \sqrt{\sum_i G_i^2}$$

$$NCC = \frac{\sum_i F_i G_i}{\sigma_f \sigma_g}$$

$$NCC = \frac{F}{\|F\|} \frac{G}{\|G\|}$$



Active Depth Estimation

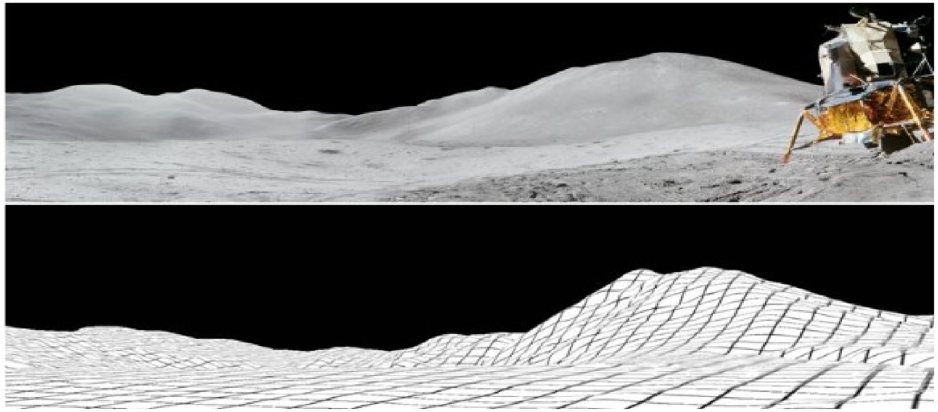


Active Depth Estimation

- Challenges


Depth Estimation with Images

- Stereo
- Active Stereo
- Time-of-flight
- Shape from shading
- Photometric stereo
- Depth from defocus



Summary

- Computing the disparity field
  - Blockmatching
  - Dynamic programming
  - SGM
  - Belief Propagation
- Alternative depth estimation methods

- Additional Literature:
  - Bradski, Kaehler: Learning OpenCV
  - Ponce, Forsyth: Computer Vision, Chp. 11