Computer Vision: from Recognition to Geometry
Lecture 14

Stereo Matching

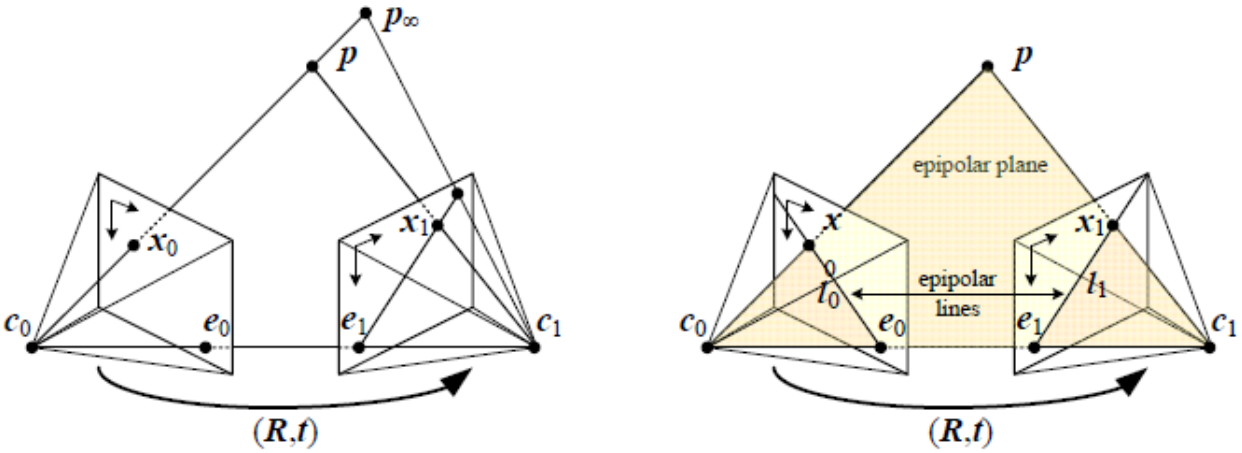
Wei-Chih Tu (塗偉志)

National Taiwan University

Fall 2018

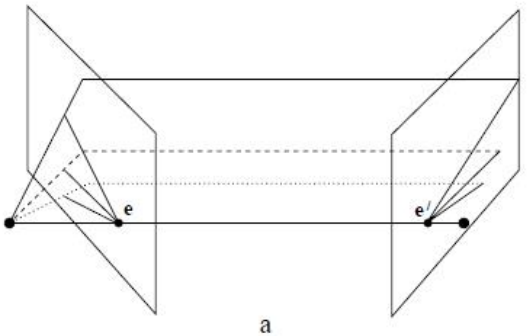
Stereo Matching

- For pixel x_0 in one image, where is the corresponding point x_1 in another image?
  - Stereo: two or more input views
- Based on the epipolar geometry, corresponding points lie on the epipolar lines
  - A matching problem



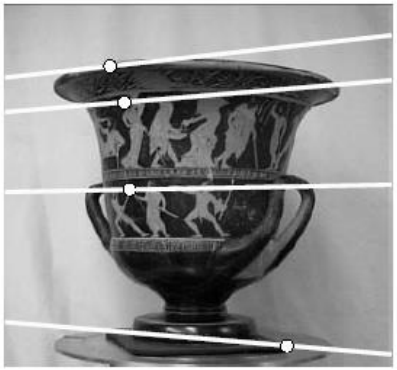
Epipolar Geometry for Converging Cameras

- Still difficult
  - Need to trace different epipolar lines for every point



a

b



c

Image Rectification

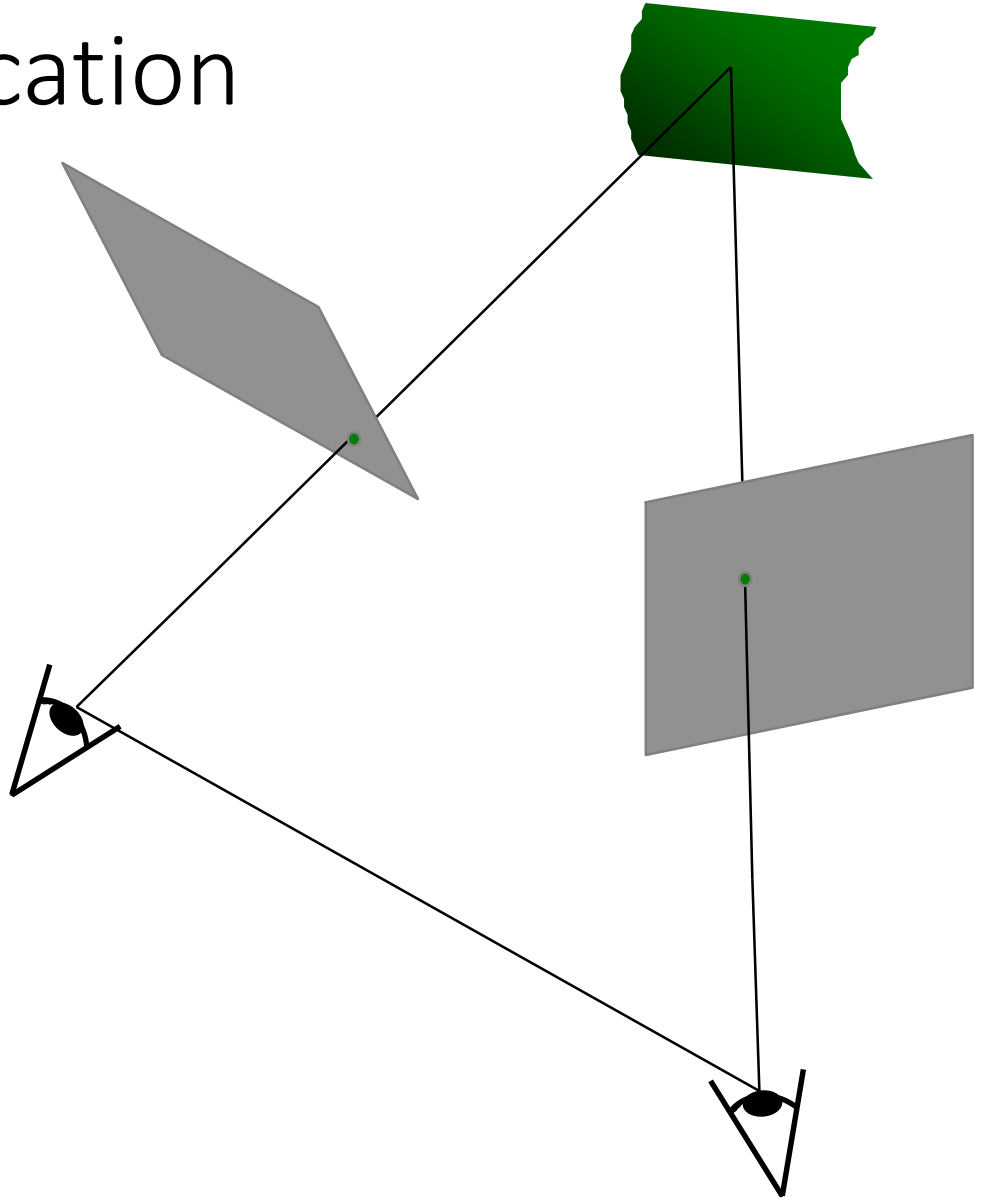


Image Rectification

- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each image

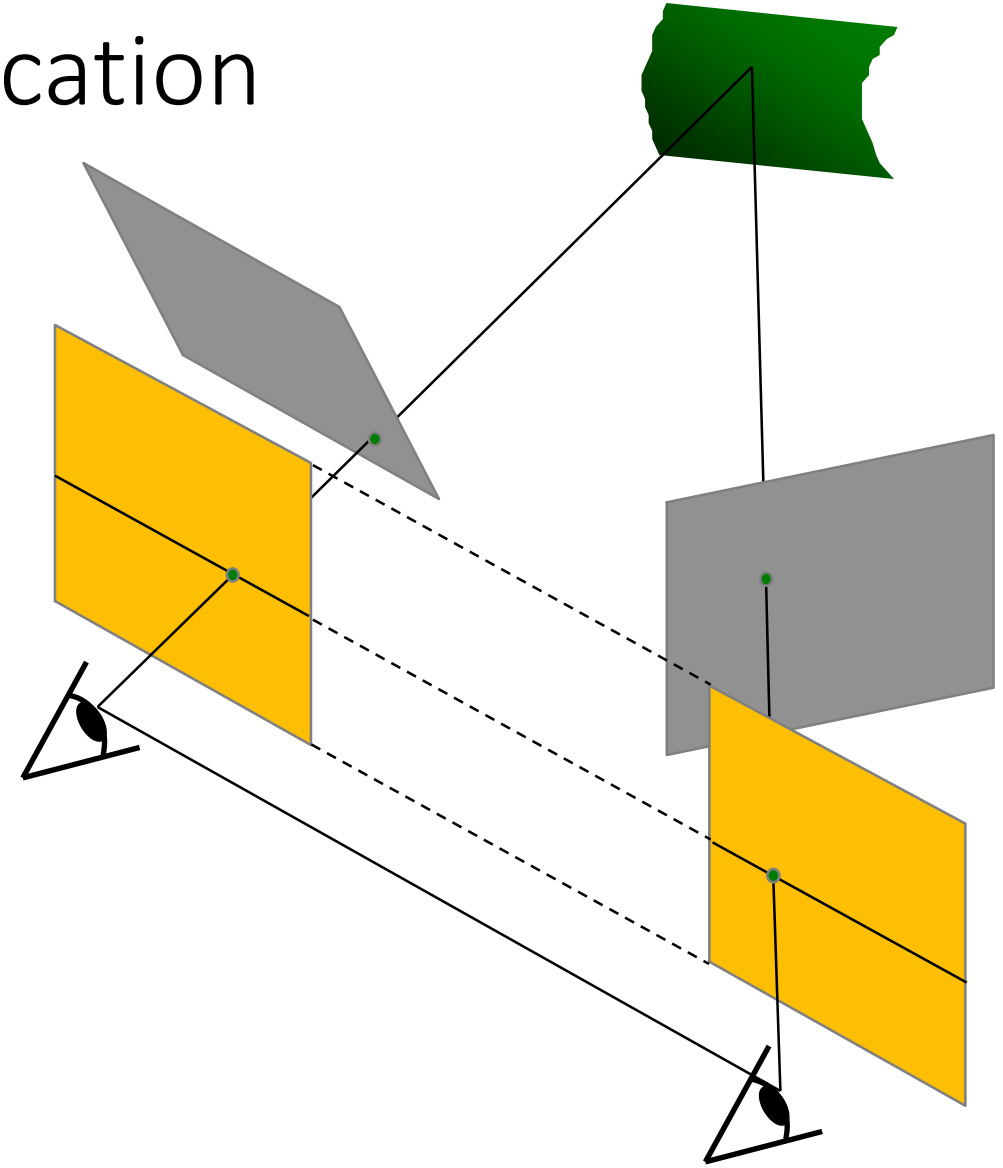


Image Rectification

- [Loop and Zhang 1999]

Original image pair overlaid with several epipolar lines.

Images transformed so that epipolar lines are parallel.

Images rectified so that epipolar lines are horizontal and aligned in vertical.

Final rectification that minimizes horizontal distortions. (Shearing)

Loop and Zhang. Computing Rectifying Homographies for Stereo Vision. In CVPR 1999.

Disparity Estimation

- After rectification, stereo matching becomes the disparity estimation problem
- Disparity = horizontal displacement of corresponding points in the two images
  - Disparity of ✕ = $x_L - x_R$

Disparity Estimation

- The “hello world” algorithm: block matching
  - Consider SSD as matching cost

Winner take all (WTA)

d	0	1	2	3	…	33	…	59	60
SAD	100	90	88	88	…	12	…	77	85

Left view

Right view

Disparity Estimation

- The “hello world” algorithm: block matching

```
• For each pixel in the left image
    • For each disparity level
        • For each pixel in window
            • Compute matching cost
    • Find disparity with minimum matching cost
```

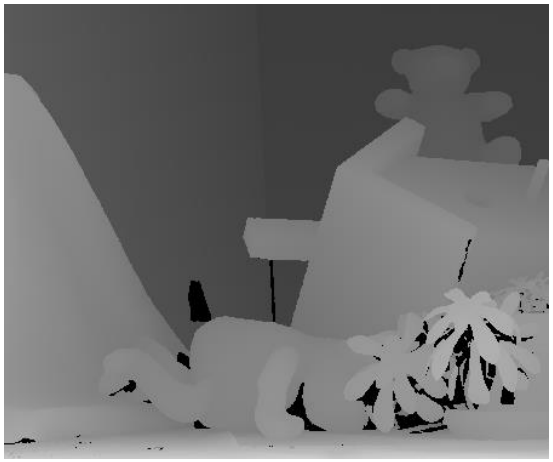
Disparity Estimation

- Reverse order of loops

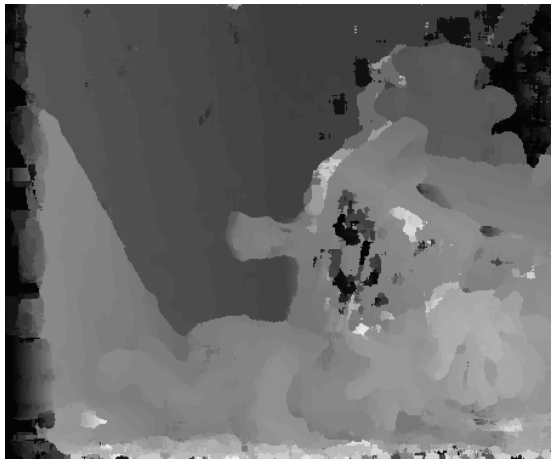
```
• For each disparity in the left image
    • For each pixel
        • For each pixel in window
            • Compute matching cost
• Find disparity with minimum matching cost at each pixel
```

Disparity Estimation

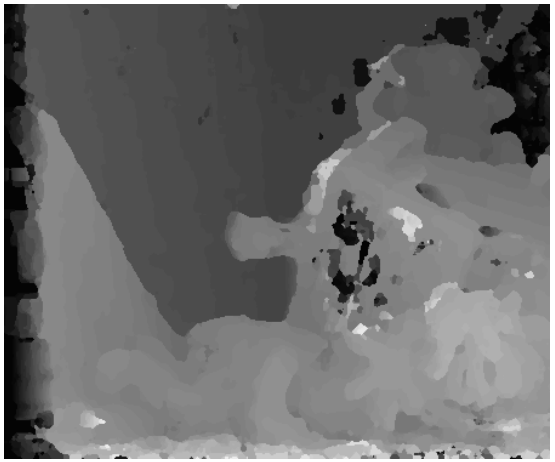
- Block matching result



Ground-truth

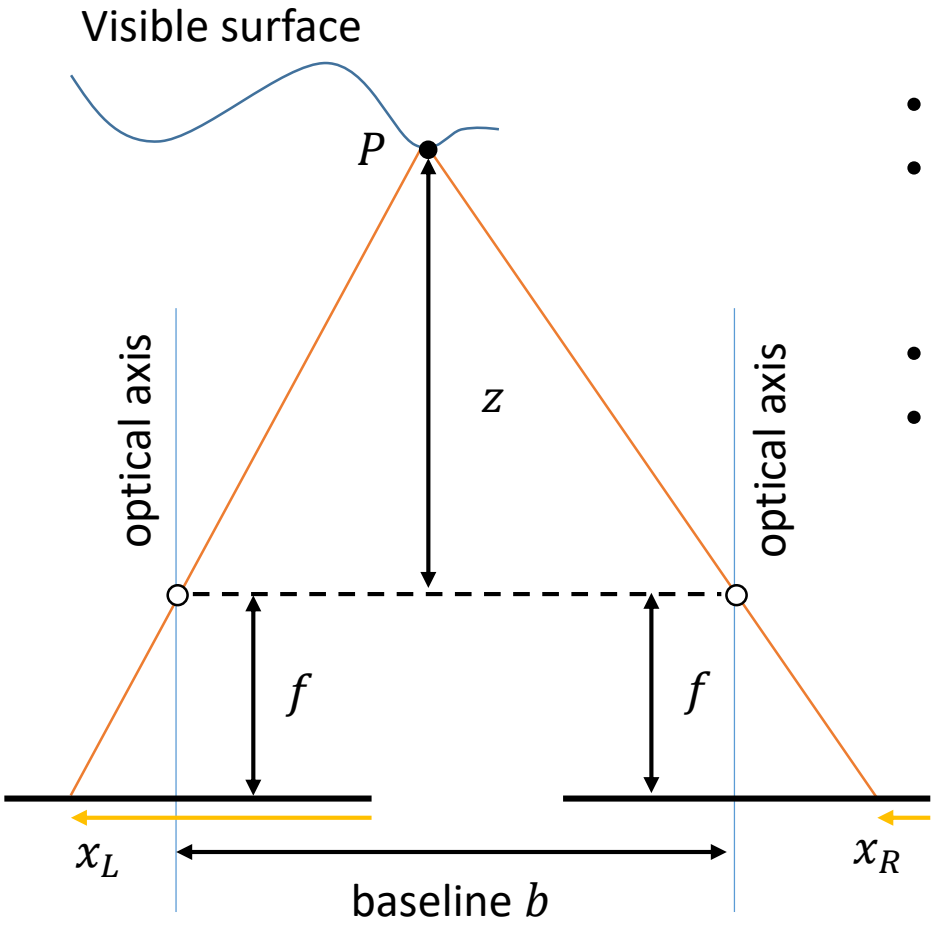


Window 5x5



After 3x3 median filter

Depth from Disparity



- Disparity $d = x_L - x_R$
- It can be derived that

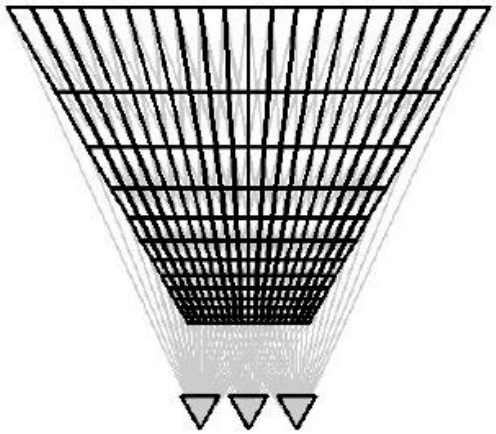
$$d = \frac{f \cdot b}{z}$$

- Disparity = 0 for distant points
- Larger disparity for closer points

Depth Error from Disparity

- From above equation, we can also derive the depth error w.r.t. the disparity error is:

$$\epsilon_z = \frac{z^2}{f \cdot b} \epsilon_d$$



Gallup et al. Variable baseline/resolution stereo. In CVPR 2008.

Components of a Stereo Vision System

- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

Components of a Stereo Vision System

- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

Most stereo matching papers mainly focus on disparity estimation

More on Disparity Estimation

- Typical pipeline
- Matching cost
- Local methods
  - Adaptive support window weight
  - Cost-volume filtering
- Global methods
  - Belief propagation
  - Dynamic programming
  - Graph cut
- Better disparity refinement
- More challenges

Typical Stereo Pipeline

- Cost computation
- Cost (support) aggregation
- Disparity optimization
- Disparity refinement

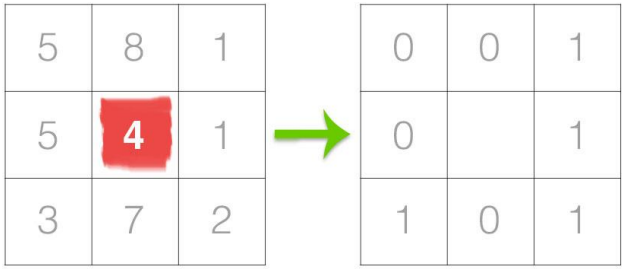
Block matching algorithm

Method	Matching cost	Aggregation	Optimization
SSD (traditional)	squared difference	square window	WTA
Hannah [51]	cross-correlation	(square window)	WTA
Nishihara [82]	binarized filters	square window	WTA
Kass [63]	filter banks	-none-	WTA
Fleet *et al.* [40]	phase	-none-	phase-matching
Jones and Malik [57]	filter banks	-none-	WTA
Kanade [58]	absolute difference	square window	WTA
Scharstein [95]	gradient-based	Gaussian	WTA
Zabih and Woodfill [129]	rank transform	(square window)	WTA
Cox *et al.* [32]	histogram eq.	-none-	DP
Frohlinghaus and Buhmann [41]	wavelet phase	-none-	phase-matching
Birchfield and Tomasi [12]	shifted abs. diff	-none-	DP
Marr and Poggio [73]	binary images	iterative aggregation	WTA
Prazdny [89]	binary images	3D aggregation	WTA
Szeliski and Hinton [114]	binary images	iterative 3D aggregation	WTA
Okutomi and Kanade [84]	squared difference	adaptive window	WTA
Yang *et al.* [127]	cross-correlation	non-linear filtering	hier. WTA
Shah [103]	squared difference	non-linear diffusion	regularization
Boykov *et al.* [22]	thresh. abs. diff.	connected-component	WTA
Scharstein and Szeliski [97]	robust sq. diff.	iterative 3D aggregation	mean-field
Zitnick and Kanade [132]	squared difference	iterative aggregation	WTA
Veksler [124]	abs. diff - avg.	adaptive window	WTA
Quam [90]	cross-correlation	-none-	hier. warp
Barnard [6]	squared difference	-none-	SA
Geiger *et al.* [46]	squared difference	shiftable window	DP
Belhumeur [9]	squared difference	-none-	DP
Cox *et al.* [31]	squared difference	-none-	DP
Ishikawa and Geiger [55]	squared difference	-none-	graph cut
Roy and Cox [92]	squared difference	-none-	graph cut
Bobick and Intille [18]	absolute difference	shiftable window	DP
Boykov *et al.* [23]	squared difference	-none-	graph cut
Kolmogorov and Zabih [65]	squared difference	-none-	graph cut
Birchfield and Tomasi [14]	shifted abs. diff.	-none-	GC + planes
Tao *et al.* [117]	squared difference	(color segmentation)	WTA + regions

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV 2002.

Matching Cost

- Squared difference (SD): $(I_p - I_q)^2$
- Absolute difference (AD): $|I_p - I_q|$
- Normalized cross-correlation (NCC)
- Zero-mean NCC (ZNCC)
- Hierarchical mutual information (HMI)
- Census cost
- Truncated cost
  - $C = \min(C_0, \tau)$

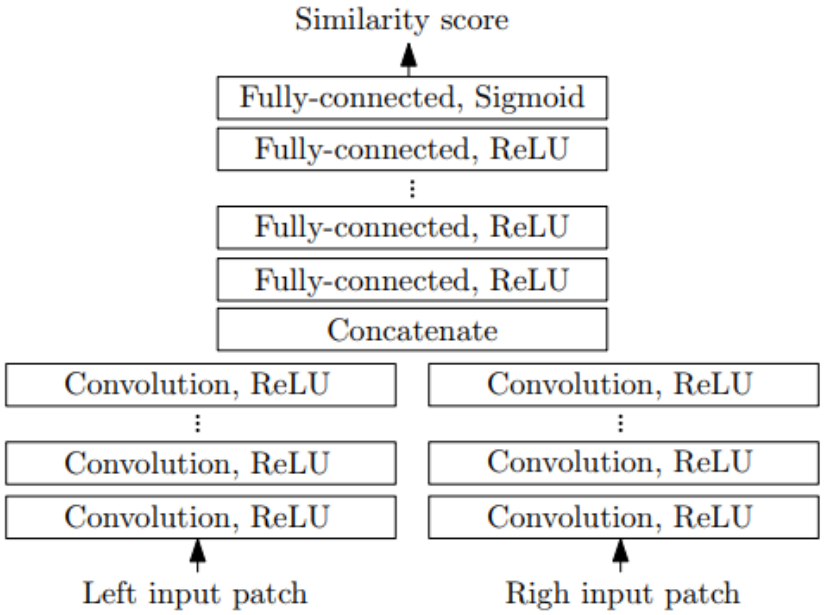


Local binary pattern

Hirschmuller and Scharstein. Evaluation of stereo matching costs on images with radiometric differences. PAMI 2008.

Matching Cost

- Deep matching cost (MC-CNN)



Date	Name	Res	bad 2.0 (%) Weight Avg	Aus MP: nd: 2 im0 i G nonc
05/26/18	NOSS_ROB	H	5.01 1	3.57
03/06/18	NOSS	H	5.04 2	3.57
06/22/17	LocalExp	H	5.43 3	3.65
06/01/18	NaN_ROB	H	5.73 4	3.41
01/24/17	3DMST	H	5.92 5	3.71
03/10/17	MC-CNN+TDSR	F	6.35 6	5.45
05/12/16	PMSC	H	6.71 7	3.46
10/19/16	LW-CNN	H	7.04 8	4.65
04/12/16	MeshStereoExt	H	7.08 9	4.41
10/12/17	FEN-D2DRR	H	7.23 10	4.68
05/28/16	APAP-Stereo	H	7.26 11	5.43
03/11/18	SGM-Forest	H	7.37 12	4.71
01/19/16	NTDE	H	7.44 13	5.72
05/31/18	CBMV_ROB	H	7.65 14	3.48
02/28/18	FDR	H	7.69 15	5.41
11/28/18	MSFNetA	H	7.96 16	6.21
10/29/18	Dense-CNN	H	7.98 17	5.59
08/28/15	MC-CNN-acrt	H	8.08 18	5.59
11/03/15	MC-CNN+RBS	H	8.42 19	6.05
09/13/16	SNP-RSM	H	8.75 20	5.46
12/11/17	OVOD	H	8.87 21	4.74
01/21/16	MCCNN_Layout	H	8.94 22	5.53
01/26/16	MC-CNN-fst	H	9.47 23	7.35

Snapshot from Middlebury v3

Zbontar and LeCun. Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research. 2016.
https://github.com/jzbontar/mc-cnn

More on Disparity Estimation

- Typical pipeline
- Matching cost
- Local methods
  - Adaptive support window weight
  - Cost-volume filtering
- Global methods
  - Belief propagation
  - Dynamic programming
  - Graph cut
- Better disparity refinement
- More challenges

Local Methods

- Cost computation
- Cost (support) aggregation
  - Adaptive support weight
  - Adaptive support shape
- Disparity optimization: winner-take-all
- Disparity refinement

Adaptive Support Weight

- Not all pixels are equal
  - Larger weight for near pixels
  - Larger weight for pixels with similar color

(a) left support window (b) right support window (c) color difference between (a) and (b)

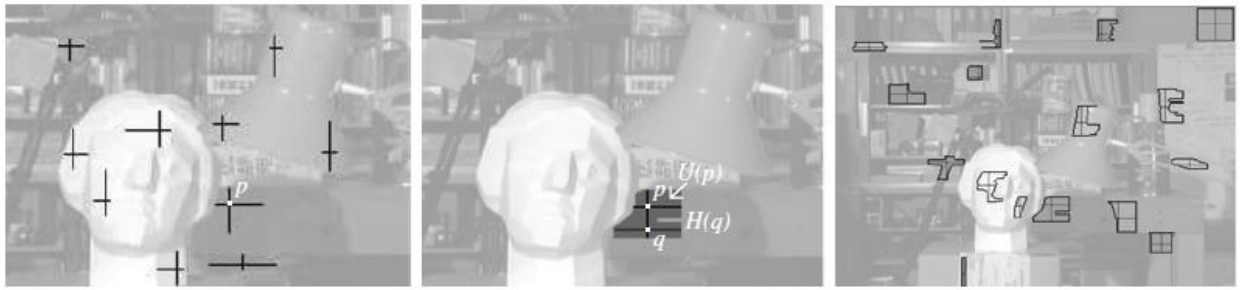
$$w(p,q) = k \cdot \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p}\right)\right)$$

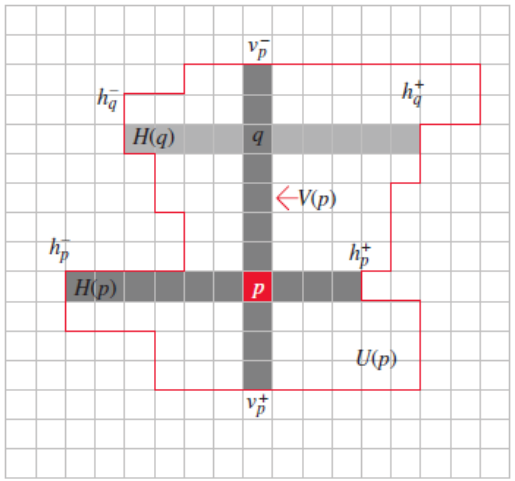
It's bilateral kernel! Computationally expensive ☹

Kuk-Jin Yoon and In-So Kweon. Locally adaptive support-weight approach for visual correspondence search. In CVPR 2005.

Adaptive Support Shape

- Cross-based cost aggregation





Find the largest arm span:

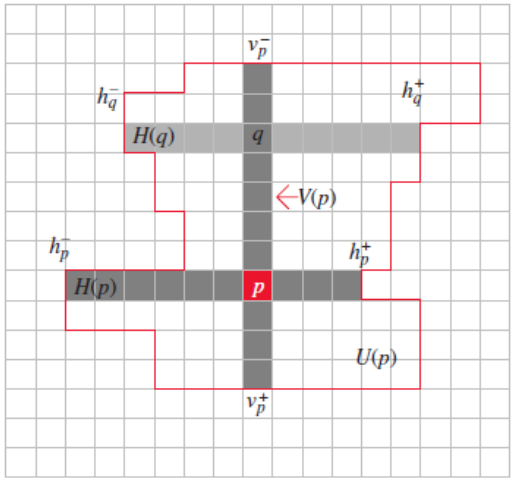
$$r^* = \max_{r \in [1,L]} \left(r \prod_{i \in [1,r]} \delta(p, p_i) \right)$$

$$\delta(p_1, p_2) = \begin{cases} 1, & \max_{c \in \{R,G,B\}} \left(|I_c(p_1) - I_c(p_2)| \right) \leq \tau \\ 0, & \text{otherwise} \end{cases}$$

Zhang et al. Cross-based local stereo matching using orthogonal integral images. CSVT 2009.

Adaptive Support Shape

- Cross-based cost aggregation



$$\overline{E}_d(p) = \frac{1}{\|U_d(p)\|} E_d(p) = \frac{1}{\|U_d(p)\|} \sum_{s \in U_d(p)} e_d(s)$$

with

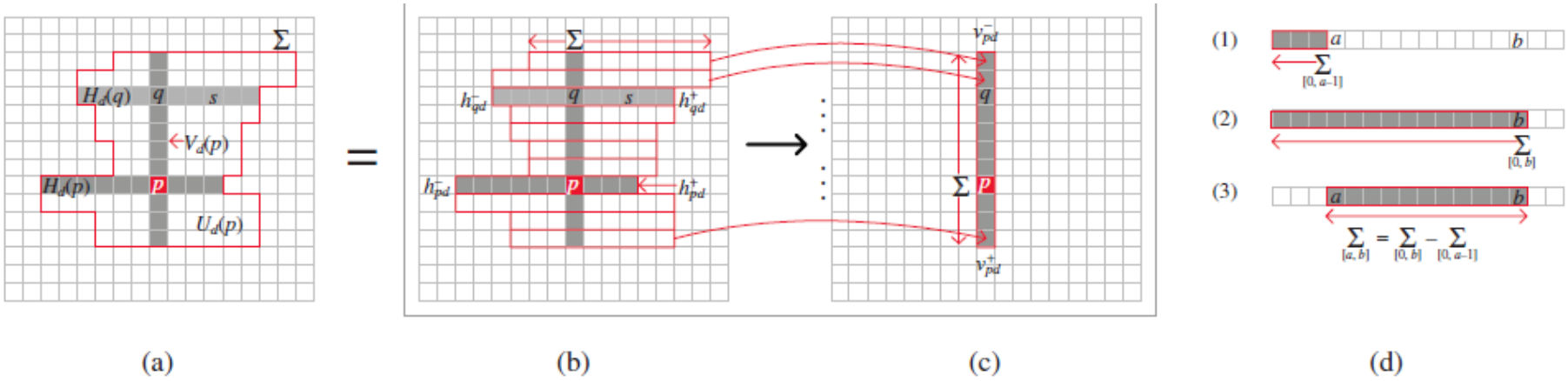
$$U_d(p) = \{(x, y) \mid (x, y) \in U(p), (x - d, y) \in U'(p')\}.$$

$$e_d(s) = \min \left(\sum_{c \in \{R,G,B\}} \left| I_c(s) - I'_c(s') \right|, T \right)$$

Zhang et al. Cross-based local stereo matching using orthogonal integral images. CSVT 2009.

Adaptive Support Shape

- Cross-based cost aggregation
  - Fast algorithm using the orthogonal integral image (OII) technique

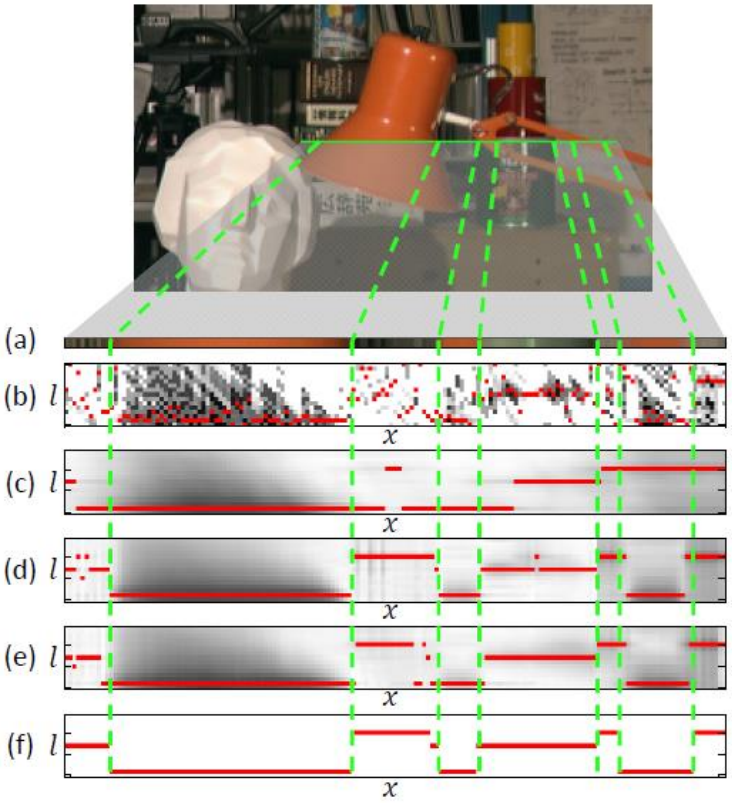


We only need four additions/subtractions for an anchor pixel
to aggregate raw matching costs over any arbitrary shaped regions.

Zhang et al. Cross-based local stereo matching using orthogonal integral images. CSVT 2009.

Cost-Volume Filtering

- Illustration of the matching cost



Raw cost

Smoothed by box filter

Smoothed by bilateral filter

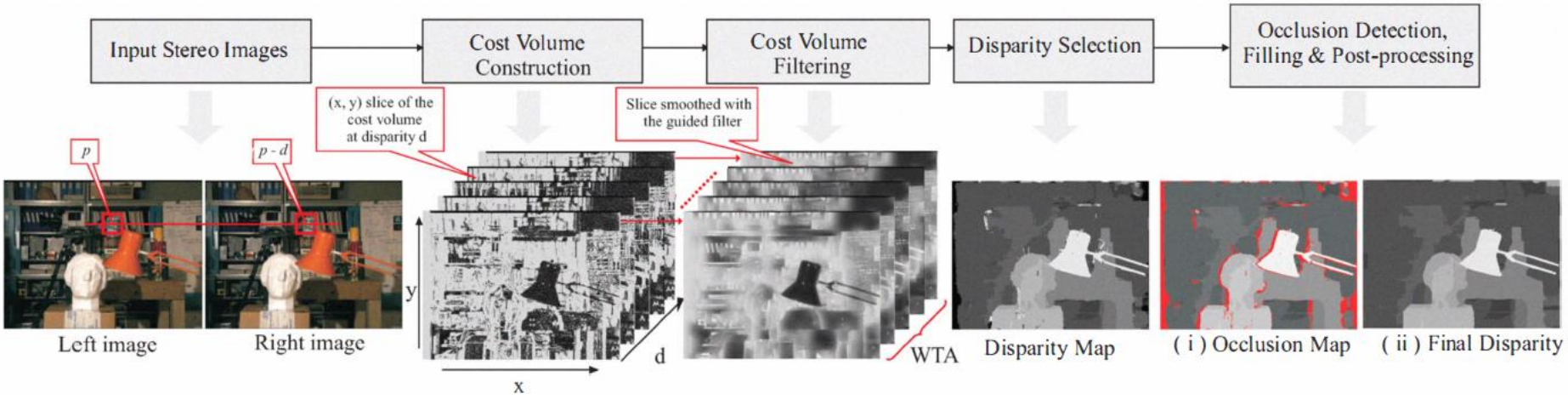
Smoothed by guided filter

Ground-truth

Rhemann et al. Fast cost-volume filtering for visual correspondence and beyond. In CVPR 2011.

Cost-Volume Filtering

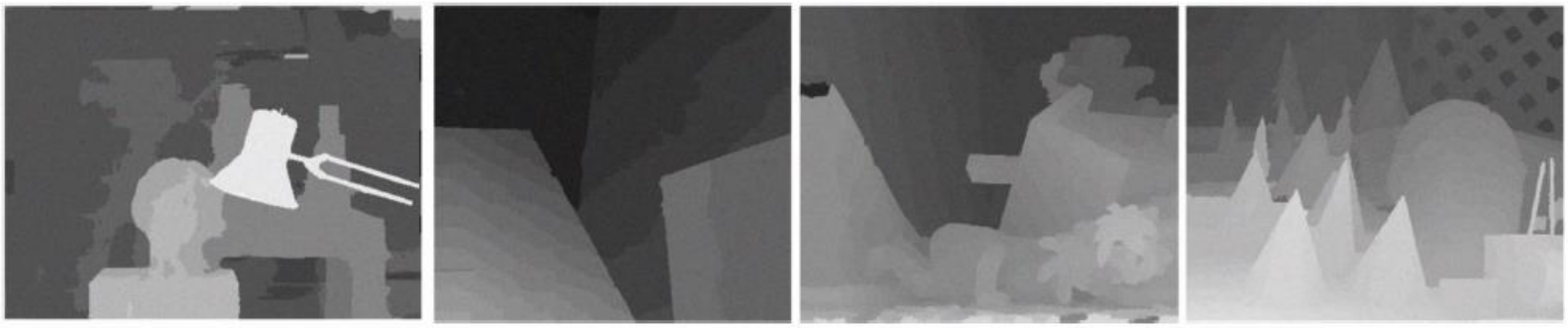
- The cost spans a $H \times W \times L$ volume
- Local cost aggregation can be regarded as filtering the volume to obtain more reliable matching costs
- Choose O(1) edge-preserving filters so that the overall complexity is regardless of the window size
- Easy to parallelize



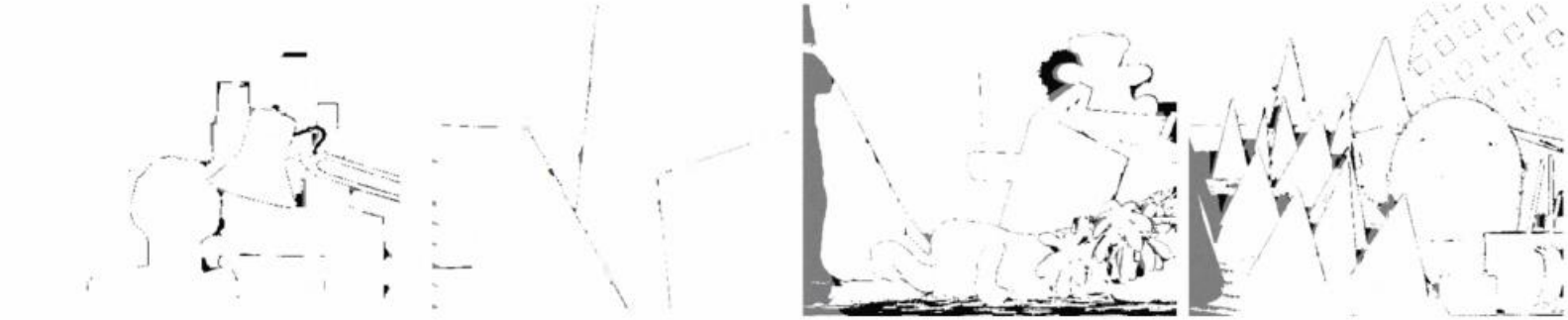
Rhemann et al. Fast cost-volume filtering for visual correspondence and beyond. In CVPR 2011.

Cost-Volume Filtering

Method	Rank	Avg. Error (%)	Avg. Runtime (ms)
Ours	9	5.55	65
GeoSup [12]	12	5.80	16000
Plane-fit BP	13	5.78	650
Ours using AdaptWeight [31]	15	5.86	15000
AdaptWeight [31]	32	6.67	8550
Real-time GPU	66	9.82	122.5
Reliability DP	69	10.7	187.8
DCB Grid [19]	76	10.9	95.4*



Disparity Maps

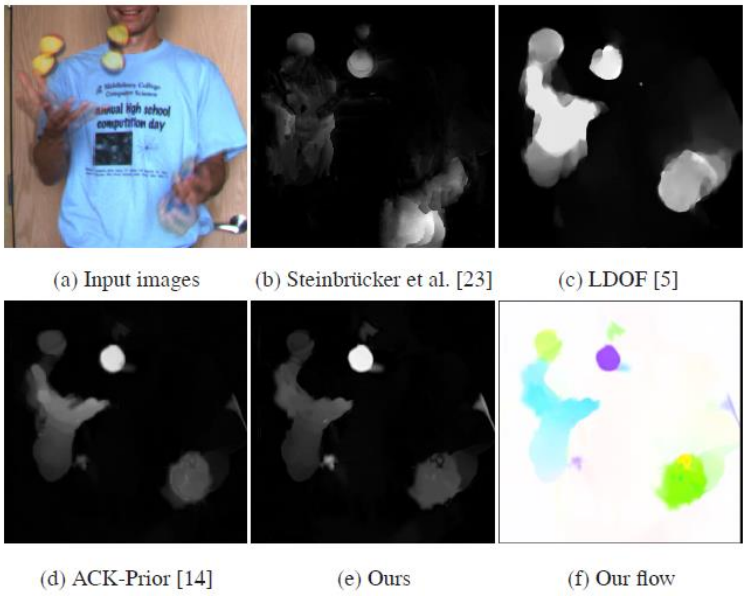


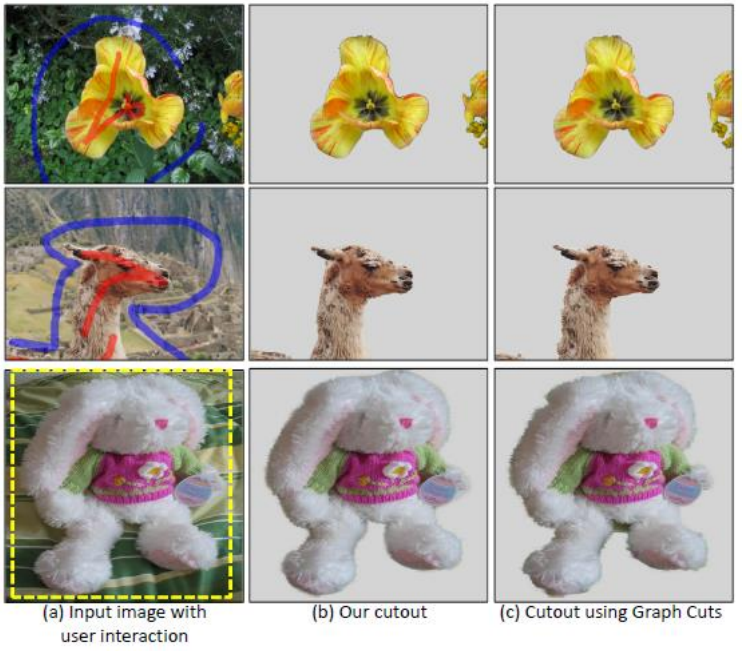
Error Maps

Rhemann et al. Fast cost-volume filtering for visual correspondence and beyond. In CVPR 2011.

Cost-Volume Filtering

- Cost-volume filtering is a general framework and can be applied to other discrete labeling problems
  - Optical flow: labels are displacements
  - Segmentation: labels are foreground/background

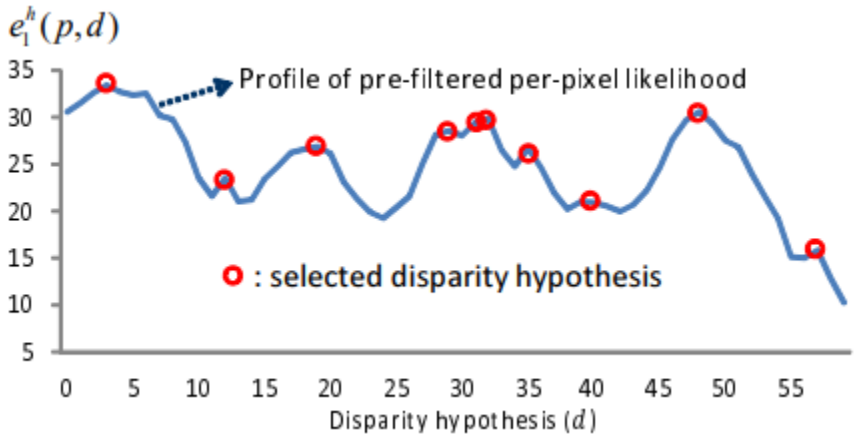




Rhemann et al. Fast cost-volume filtering for visual correspondence and beyond. In CVPR 2011.

Reduce Redundancy

- Two-pass cost aggregation
  - Pass 1: 5x5 box filter
  - Pass 2: adaptive weight filter



Min et al. A Revisit to Cost Aggregation in Stereo Matching: How Far Can We Reduce Its Computational Redundancy? In ICCV 2011.

More on Disparity Estimation

- Typical pipeline
- Matching cost
- Local methods
  - Adaptive support window weight
  - Cost-volume filtering
- Global methods
  - Belief propagation
  - Dynamic programming
  - Graph cut
- Better disparity refinement
- More challenges

Global Methods

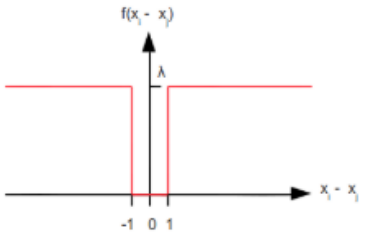
- A good stereo correspondence
  - Match quality: each pixel finds a good match in the other image
  - Smoothness: disparity usually changes smoothly
- Mathematically, we want to minimize:

$$E(d) = \sum_{p} D(d_p) + \lambda \sum_{p,q} V(d_p, d_q)$$

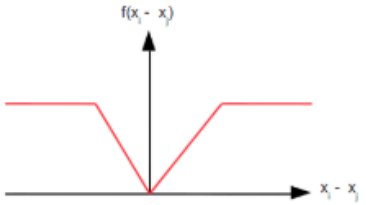
  - D_p is the data term, which is the cost of assigning label d_p to pixel p. D_p can be the raw cost or the aggregated cost.
  - V is the smoothness term or discontinuity cost.
    It measures the cost of assigning labels d_p and d_q to two adjacent pixels.

Global Methods

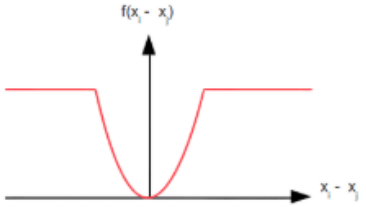
- Choice of the Smoothness Cost
  - Consider V as $V(d_p - d_q)$
  - Make $E(d)$ non-smooth
- Optimizing $E(d)$ is hard
  - Non-smooth
  - Many local minima
  - Provably NP-hard
- Practical algorithms find approx. minima
  - Belief propagation, graph cut, dynamic programming, …



$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ \lambda & \text{otherwise} \end{cases}$$ Also known as the Potts model.



$f(n) = \lambda \times \min(|n|, K)$
Truncated linear model.



$f(n) = \lambda \times \min(n^2, K)$
Truncated quadratic model.

http://nghiaho.com/?page_id=1366

Belief Propagation

It takes $O(L^2)$ time to compute each message

- BP is a message passing algorithm
  - Message on each node is a vector sized L

$$m_{pq}^t(f_q) = \min_{f_p} \left(V(f_p, f_q) + D_p(f_p) + \sum_{s \in \mathcal{N}(p) \setminus q} m_{sp}^{t-1}(f_p) \right)$$

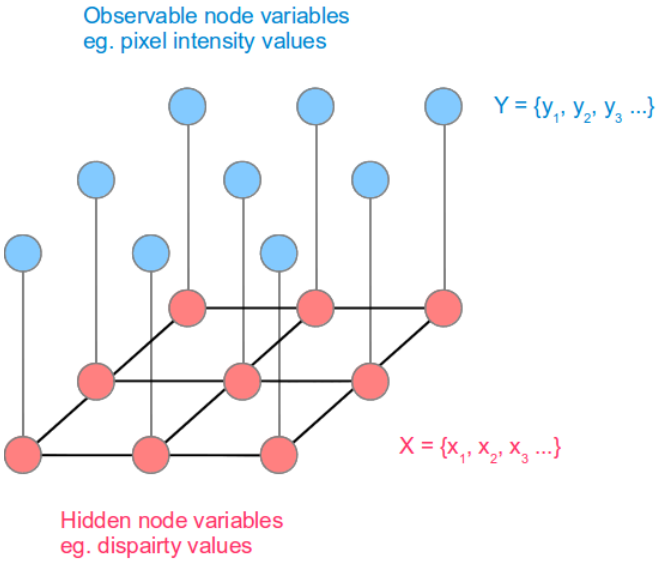
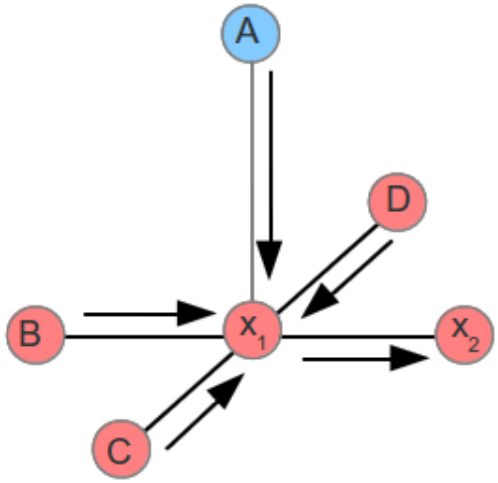


Illustration of a 3x3 MRF



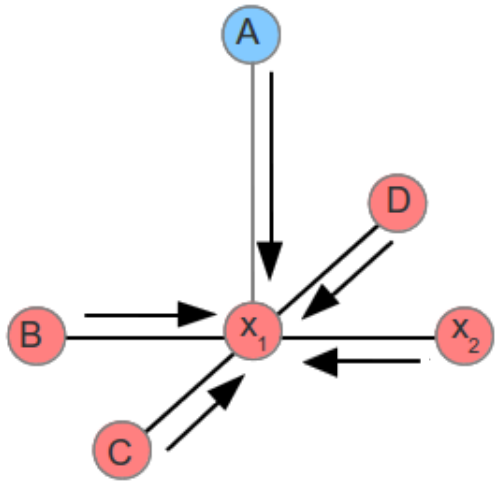
Message passing to the right

http://nghiaho.com/?page_id=1366

Belief Propagation

- Loopy BP (LBP): BP applied to graphs that contain loops

$$b_q(f_q) = D_q(f_q) + \sum_{p \in \mathcal{N}(q)} m_{pq}^T(f_q).$$



Calculating belief

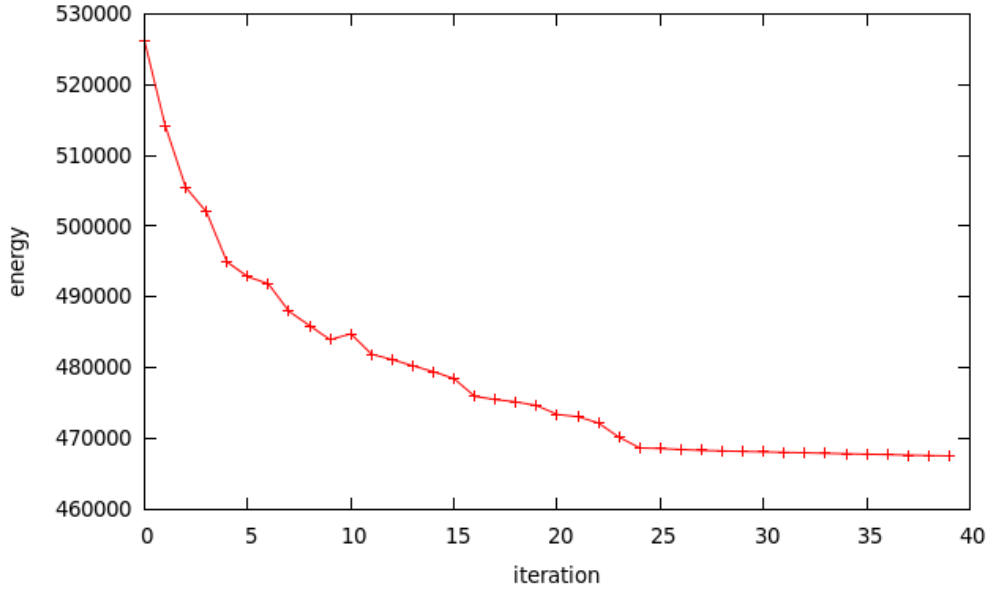
```
function LoopyBeliefPropgation
    initialise all messages

    for t iterations
        at every pixel pass messages right
        at every pixel pass messages left
        at every pixel pass messages up
        at every pixel pass messages down
    end for

    find the best label at every pixel i by calculating belief
end
```

Overall time complexity is $O(L^2TN)$

http://nghiaho.com/?page_id=1366

Belief Propagation

- Loopy BP is not guaranteed to converge
- Empirically it converges to good approximate minima.



http://nghiaho.com/?page_id=1366

Efficient Belief Propagation

- Multiscale BP (coarse-to-fine)
- $O(L)$ time complexity message passing

$$m_{pq}^{t}(f_q) = \min_{f_p} \left(V(f_p, f_q) + D_p(f_p) + \sum_{s \in \mathcal{N}(p) \setminus q} m_{sp}^{t-1}(f_p) \right)$$

Rewrite as: $m_{pq}^{t}(f_q) = \min_{f_p} \left(V(f_p, f_q) + h(f_p) \right)$

Truncated linear model

$$V(f_p, f_q) = \min(s||f_p - f_q||, d)$$

Felzenszwalb and Huttenlocher. Efficient belief propagation for early vision. IJCV 2006.

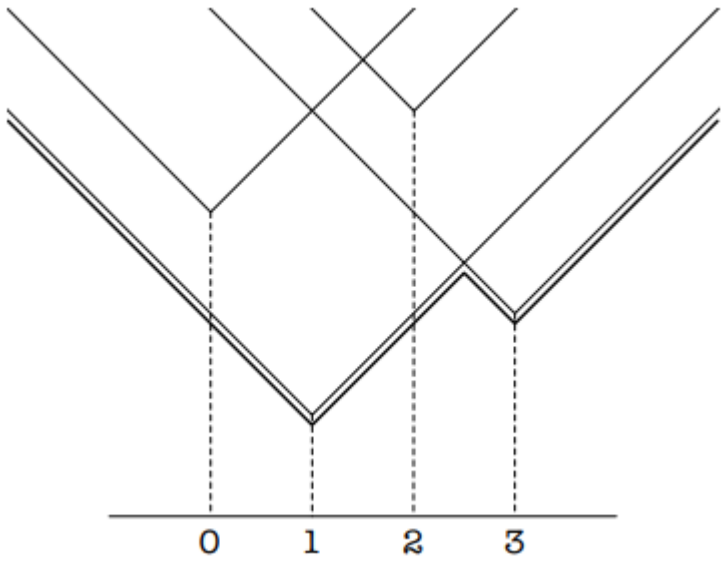
Efficient Belief Propagation

- Let's see the $O(L^2)$ message passing first

$$m_{pq}^t(f_q) = \min_{f_p}(V(f_p, f_q) + h(f_p))$$

$$V(f_p, f_q) = \min(s||f_p - f_q||, d)$$

$$m_{pq}^t(f_q) = \min_{f_p}(s||f_p - f_q|| + h(f_p))$$



Efficient Belief Propagation

- $O(L)$ time complexity message passing

$$m_{pq}^t(f_q) = \min_{f_p}(V(f_p, f_q) + h(f_p))$$

$$V(f_p, f_q) = \min(s||f_p - f_q||, d)$$

Fast algorithm:

$$m_{pq}^t(f_q) = \min_{f_p}(s||f_p - f_q|| + h(f_p))$$

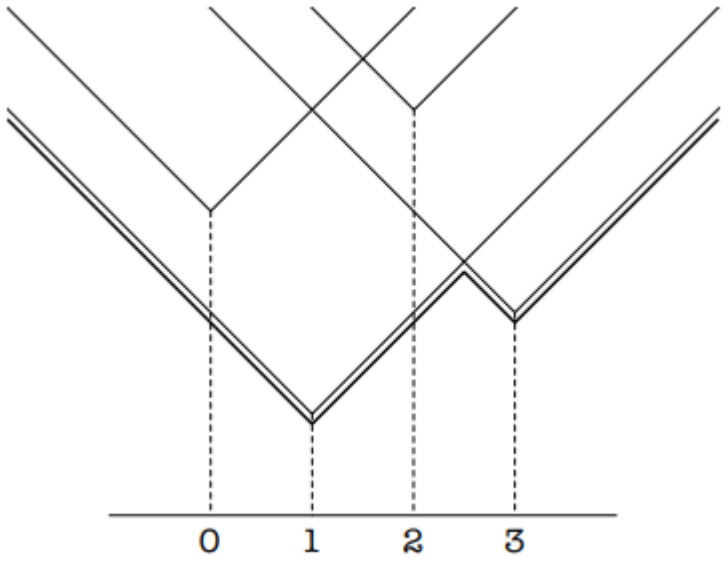
for f_q **from** 1 **to** $k-1$:

$$m(f_q) \leftarrow \min(m(f_q), m(f_q - 1) + s).$$

for f_q **from** $k-2$ **to** 0 :

$$m(f_q) \leftarrow \min(m(f_q), m(f_q + 1) + s).$$

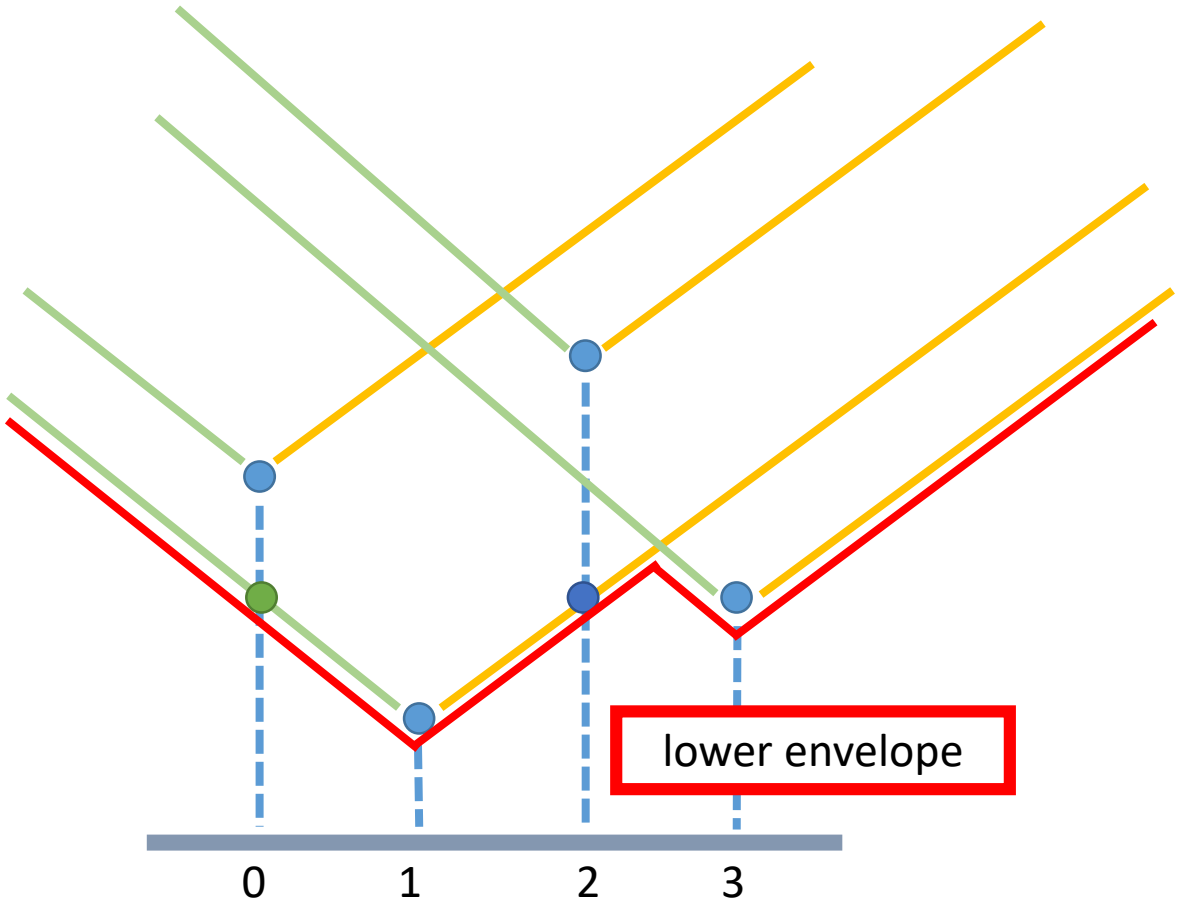
$$m_{pq}(f_q) = \min(m(f_q), \min_{f_p} h(f_p) + d).$$



Felzenszwalb and Huttenlocher. Efficient belief propagation for early vision. IJCV 2006.

Illustration

$$m_{pq}^{t}(f_q) = \min_{f_p} (s||f_p - f_q|| + h(f_p))$$



Let $L = 4$:

$m = (3,1,4,2)$

forward pass

$m = (3,1,2,2)$

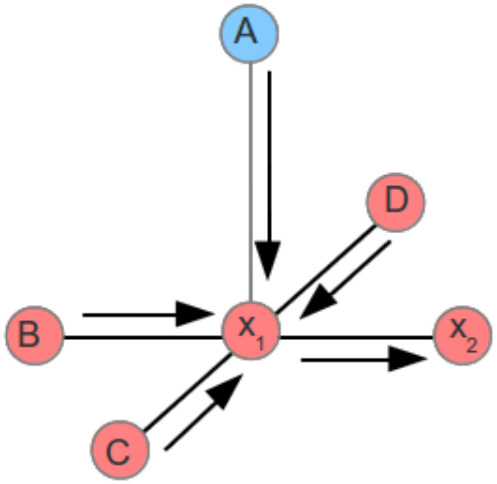
backward pass

$m = (2,1,2,2)$

Felzenszwalb and Huttenlocher. Efficient belief propagation for early vision. IJCV 2006.

Color Weighted BP

- The message is reliable when x_1 and x_2 have similar color.



Color weighted smoothness cost:

$$E_S = \rho_{bp}\rho_s|D(\mathrm{x}) - D(\mathrm{y})|$$

$$\delta_{\mathbf{xy}} = |I(\mathrm{x}) - I(\mathrm{y})|$$

$$\rho_s = 1 - \delta_{norm}$$

Yang et al. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In CVPR 2006.

Belief Propagation

- Other strong variants exists
  - Double BP
  - Constant-space BP
  - Hardware-efficient BP

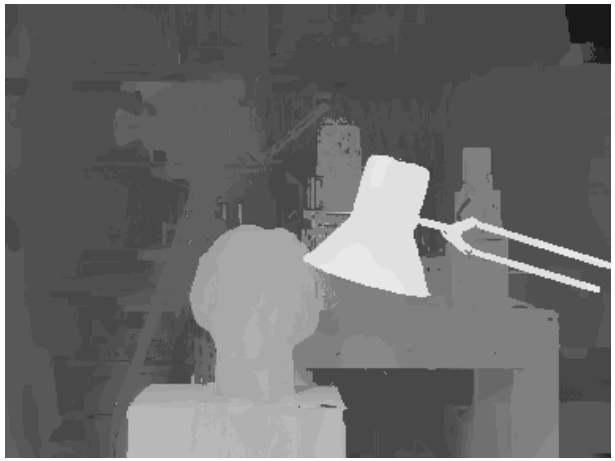
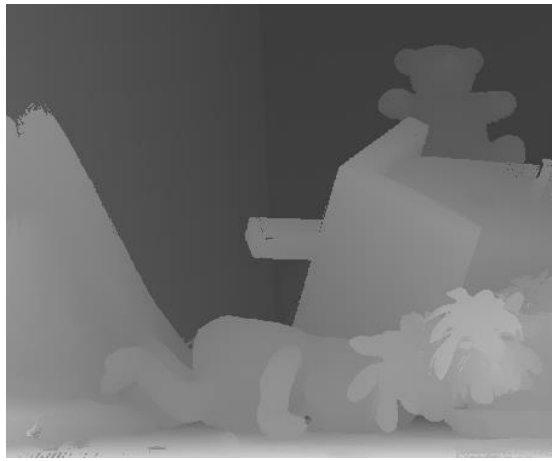
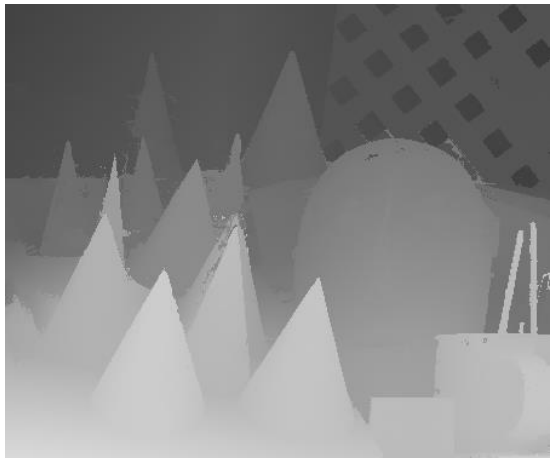
Algorithm	Avg. Rank
IGSM [155]	10.4
TSGO [141]	13.4
JSOSP+GCP [149]	14.8
KADI [164]	15.2
SSCBP [157]	17.6
ADCensus [82]	18.2
AdaptingBP [16]	22.2
CoopRegion [39]	22.2
CCRADAR [150]	26.8
PM-Forest [162]	27.2
RDP [87]	28.7
MultiRBF [129]	28.7
DoubleBP [34]	29.0
OutlierConf [40]	30.0
SegAggr [144]	30.2
CVW-RM [146]	30.4
GC+LocalExp [158]	32.0
SOS [135]	35.0
SubPixSearch [109]	35.4
AdaptiveGF [127]	35.8
GcKuwaGrad [161]	36.5
SurfaceStereo [71]	37.0
SubPixDoubleBP [29]	37.2

Snapshot from Middlebury v2

Yang et al. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In CVPR 2006.
Yang et al. A constant-space belief propagation algorithm for stereo matching. In CVPR 2010.
Liang et al. Hardware-efficient belief propagation. In CVPR 2009.

Graph Cut

- GC can also be used to minimize $E(d) = \sum_{p} D(d_p) + \lambda \sum_{p,q} V(d_p, d_q)$

Results from Taniai et al.

Taniai et al. Graph cut based continuous stereo matching using locally shared labels. In CVPR 2014.
Kolmogorov et al. What energy functions can be minimized via graph cuts? PAMI 2004
Boykov et al. Fast approximate energy minimization via graph cuts. In ICCV 1999.

More on Disparity Estimation

- Typical pipeline
- Matching cost
- Local methods
  - Adaptive support window weight
  - Cost-volume filtering
- Global methods
  - Belief propagation
  - Dynamic programming
  - Graph cut
- Better disparity refinement
- More challenges

Disparity Refinement

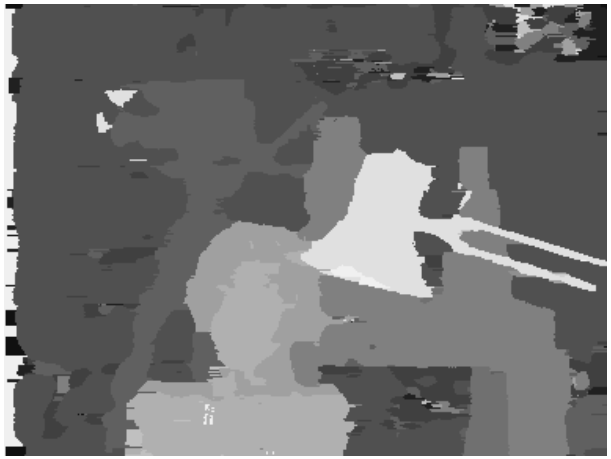
- Left-right consistency check
  - Compute disparity map D_L for left image
  - Compute disparity map D_R for right image
  - Check if $D_L(x, y) = D_R(x - D_L(x, y), y)$

Left view

Right view

Disparity Refinement

- Hole filling
  - F_L, the disparity map filled by closest valid disparity from left
  - F_R, the disparity map filled by closest valid disparity from right
  - Final filled disparity map $D = \min(F_L, F_R)$ (pixel-wise minimum)
  - Why?



The above steps do not guarantee coherency between scanlines

Disparity Refinement

- Weighted median filtering

Denoting $I(q)$ as the value at pixel q in image I and $n = (2r+1)^2$ as the number of pixels in $\mathcal{R}(p)$, we express the value and weight elements for all pixels in $\mathcal{R}(p)$ as $\{(I(q), w_{pq})\}$. By sorting values in an ascending order, the weighted median operator for pixel p returns a new pixel p^* and $I(p)$ is replaced by $I(p^*)$. This process to get p^* is

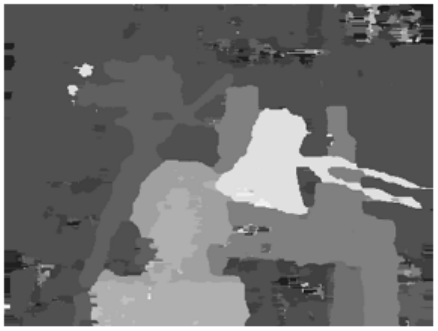
$$p^* = \min k \quad \text{s.t.} \quad \sum_{q=1}^{k} w_{pq} \geq \frac{1}{2} \sum_{q=1}^{n} w_{pq}. \tag{2}$$

(a) left image (b) box agg.+LR check+hole-filling (c) median filtering of (b) (d) weighted median filtering of (b) (e) zoom-in

Ma et al. Constant time weighted median filtering for stereo matching and beyond. In ICCV 2013.
Zhang et al. 100+ times faster weighted median filter. In CVPR 2014.

Disparity Refinement

- Weighted median filtering

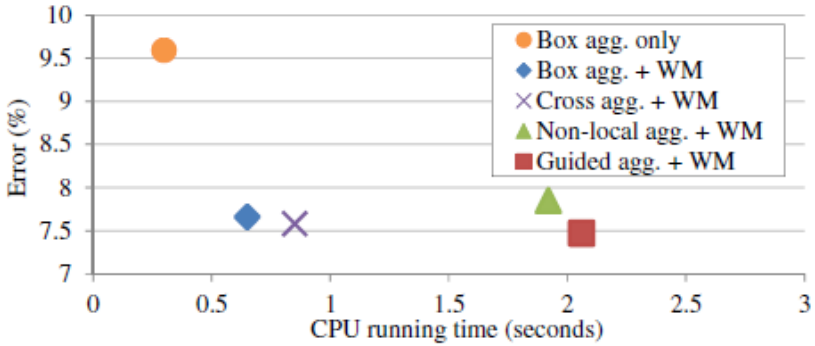


(a) box agg. + LR check + hole-filling
46ms, Err. 4.68

(b) guided agg. + LR check + hole-filling
246ms, Err. 2.08

(c) weighted median filtering of (a)
134ms, Err. 1.66

(d) weighted median filtering of (b)
334ms, Err. **1.62**



Ours and previous work	Avg. error	GPU time
NonLocalFilter [30]	5.48†	n/a
Guided agg. + **WM**	5.50	**54ms**
CostFilter [22]	5.55†	65ms†
Cross agg. + **WM**	5.63	n/a
Non-local agg. + **WM**	6.04	n/a
Box agg. + **WM**	6.19	**22ms**
VariableCross [34]	7.60†	n/a

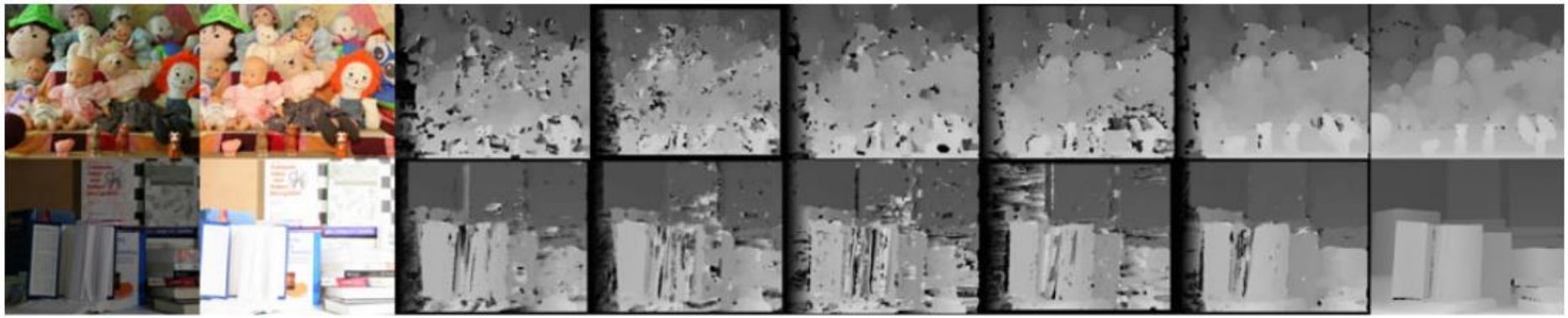
Ma et al. Constant time weighted median filtering for stereo matching and beyond. In ICCV 2013.

More on Disparity Estimation

- Typical pipeline
- Matching cost
- Local methods
  - Adaptive support window weight
  - Cost-volume filtering
- Global methods
  - Belief propagation
  - Dynamic programming
  - Graph cut
- Better disparity refinement
- More challenges

More Challenges

- Illumination invariance: most stereo algorithms assume the corresponding points share the same color/intensity.
  - This may not be true for specular reflection, transparent objects, …
- Solution: use illumination invariant features or explicit model the physics (reflection/transparency)



(a) Left image (b) Right image (c) ANCC [42] (d) BRIEF [29] (e) SIFT [28] (f) LSS [18] (g) DASC+LRP (h) Ground Truth

Xu et al. Linear time illumination invariant stereo matching. IJCV 2016.
Kim et al. DASC: robust dense descriptor for multi-modal and multi-spectral correspondence estimation. PAMI 2017.

More Challenges

- Frontal parallel assumption:
  Cost-volume filtering assumes the world is piecewise flat, so we believe mixing costs for similar pixels can refine the costs.
- In reality, there are many slanted surfaces in the world.

A sample view from KITTI 2012 dataset

More Challenges

- Breaking the frontal parallel assumption

(a) (b) (c)

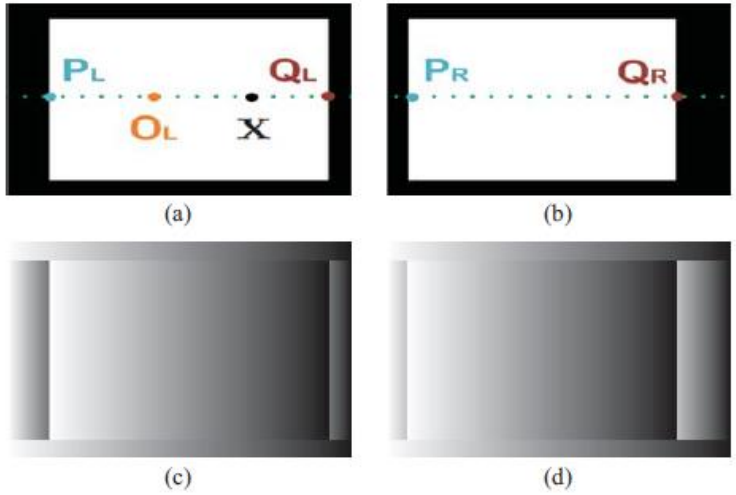
Local plane fitting:

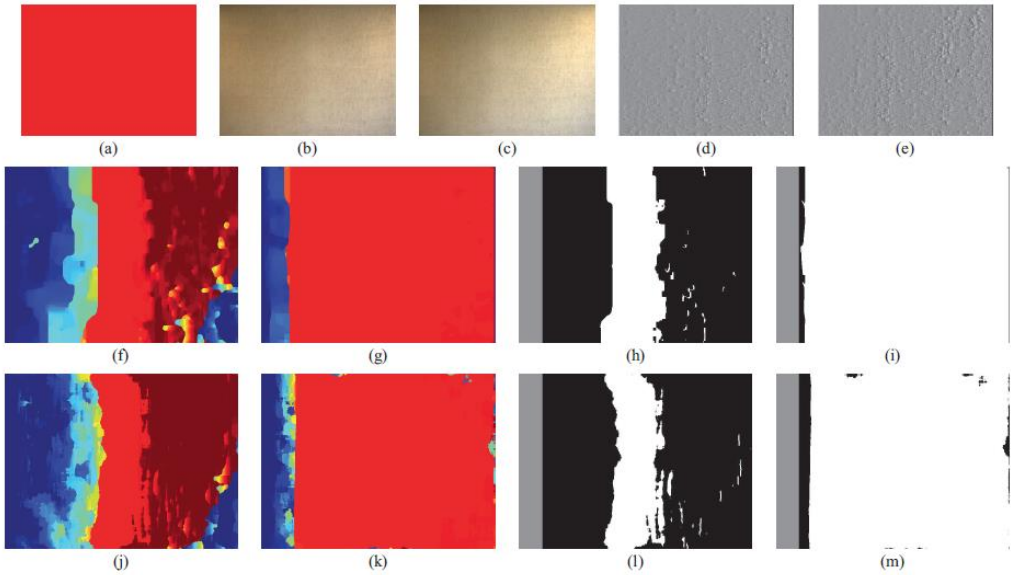
$$d_p = a_{f_p} p_x + b_{f_p} p_y + c_{f_p}$$

Bleyer et al. PatchMatch stereo – stereo matching with slanted support window. In BMVC 2014.

More Challenges

- Textureless: no reliable matching point
- Solution:
  - Use long range connection (larger window support) in cost aggregation may help
  - Transform to other domain





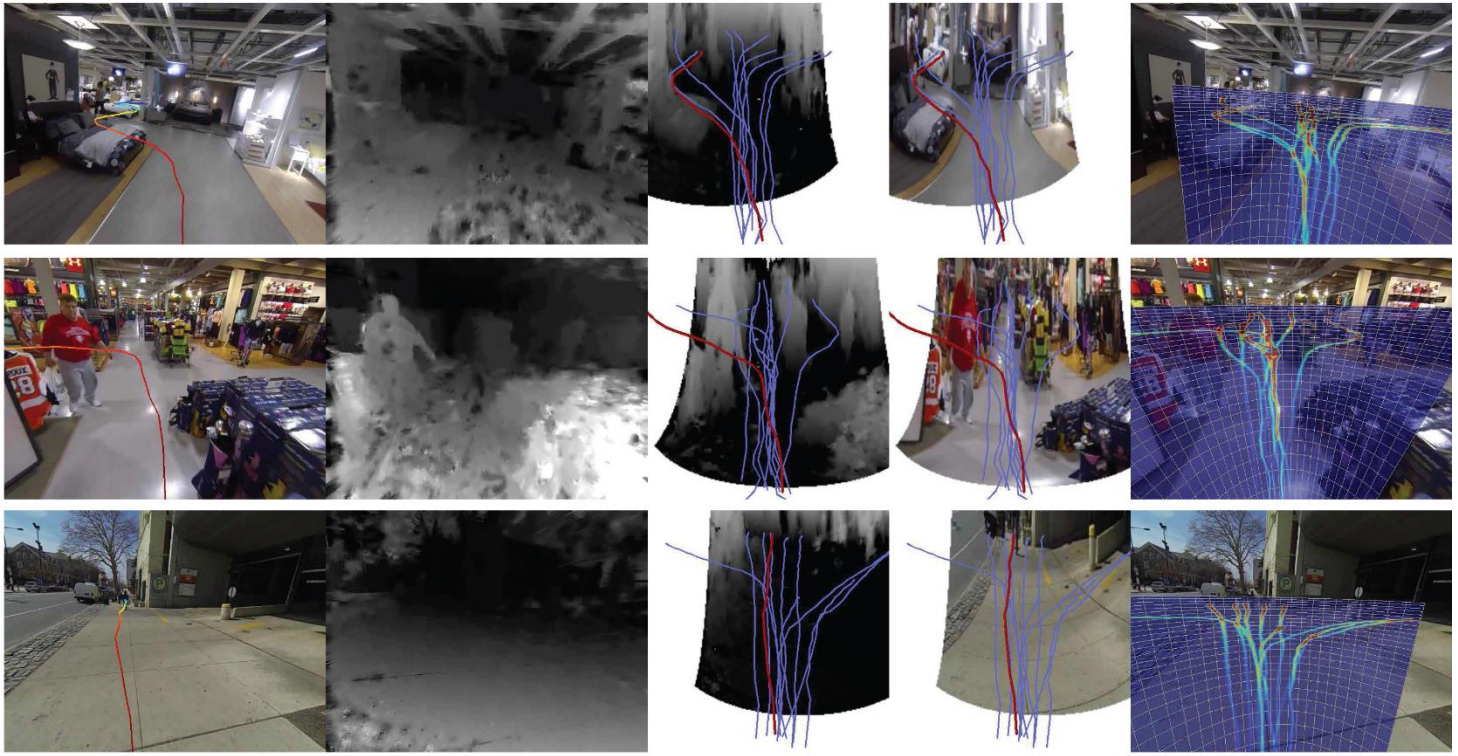
Yang et al. Stereo matching using epipolar distance transform. TIP 2012.

Benchmark

- For latest and greatest algorithms, check the following:
  - Middlebury v3
  - Middlebury v2 (still useful but no longer active)
  - KITTI 2012
  - KITTI 2015
  - ETH3D

Application: Scene Analysis

- Potential application for visual disability or robots



Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. Egocentric Future Localization. In CVPR 2016.

Application: Synthetic Defocus

- A stereo algorithm tailored for synthetic defocus application

(a) Input stereo pair with cropped subregions of the right image

(b) Our algorithm's disparity and defocused image / subregions

(c) SGM's [14] disparity and defocused image / subregions

Barron et al. Fast bilateral-space stereo for synthetic defocus. In CVPR 2015.

Summary

- Depth from disparity
- Standard stereo matching pipeline
- Stereo matching as a
  - correspondence problem
  - labeling problem
  - testbed for edge-preserving filtering
  - graph optimization problem
- Real world challenges and applications