

“Brief” Introduction to Stereo Matching

Chih-Ting Liu (劉致廷)

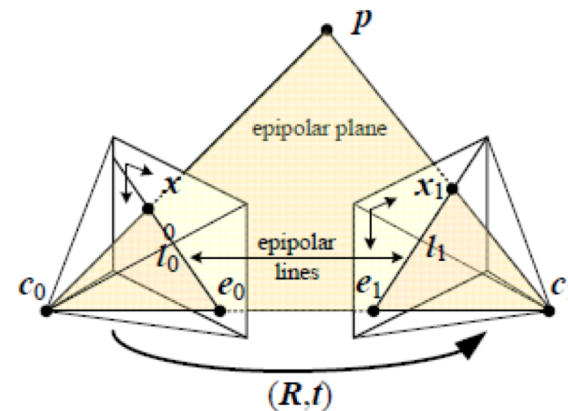
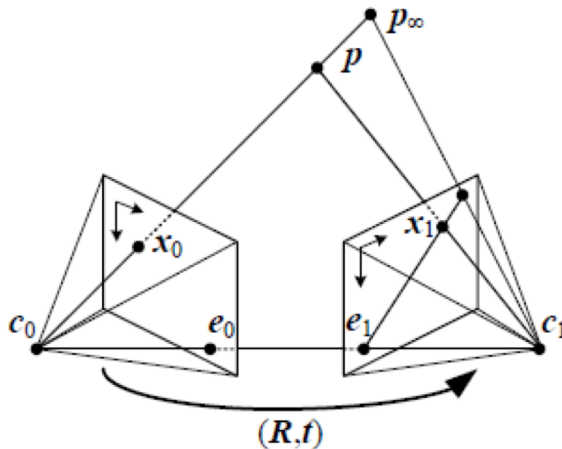
National Taiwan University

Fall 2019

slide credit : 塗偉志

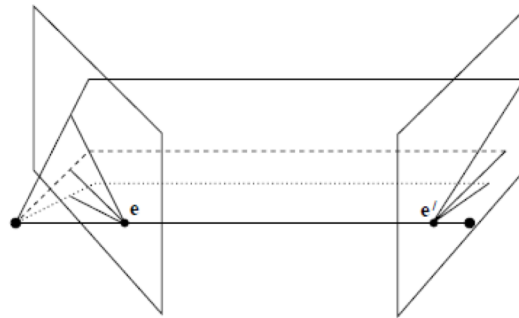
Stereo Matching

- 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 (next lectures...)
 - A **matching** problem

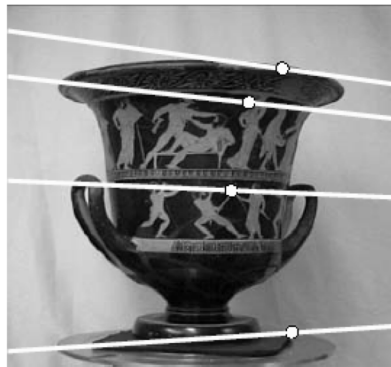


Epipolar Geometry for Converging Cameras

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



a



b



c

Image Rectification

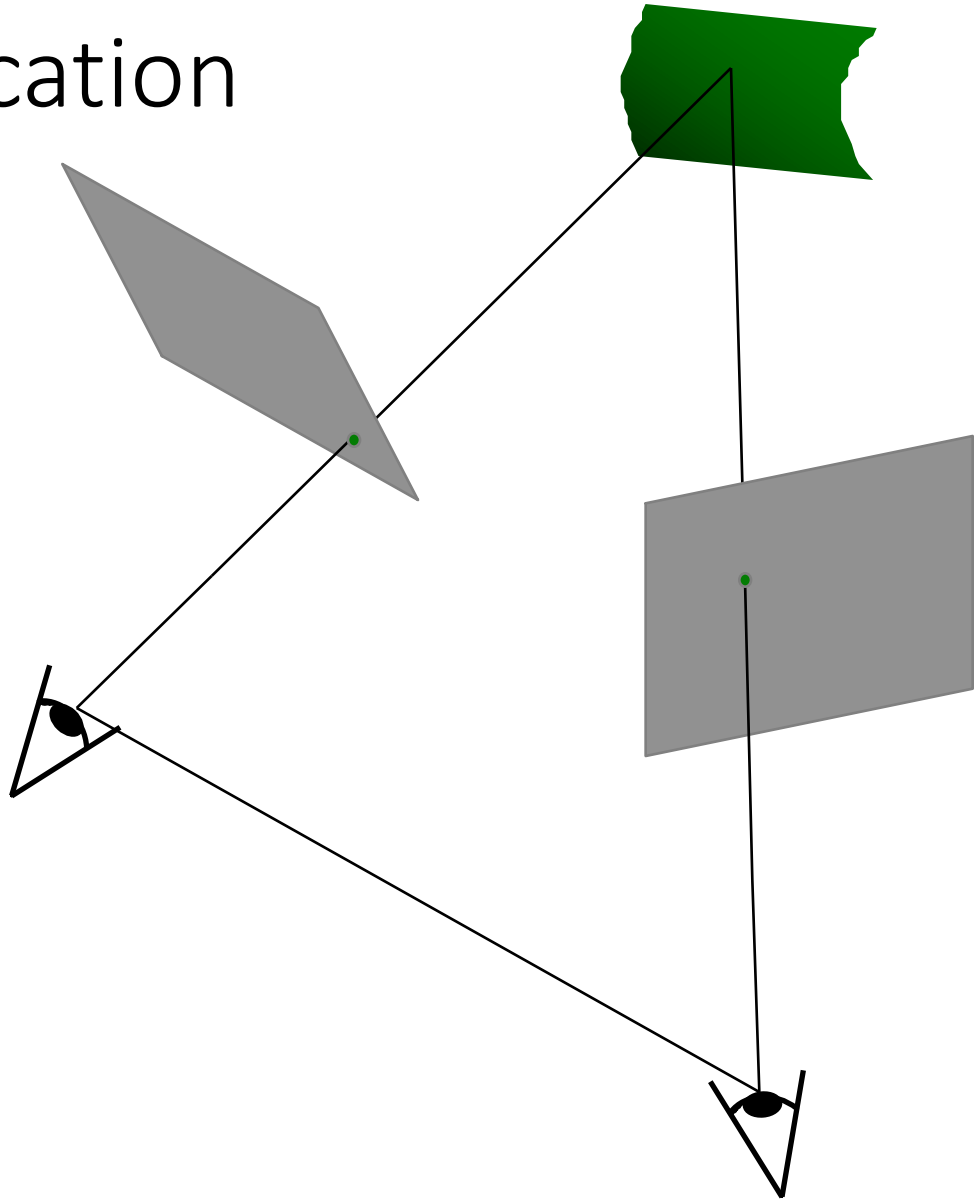


Image Rectification

- 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

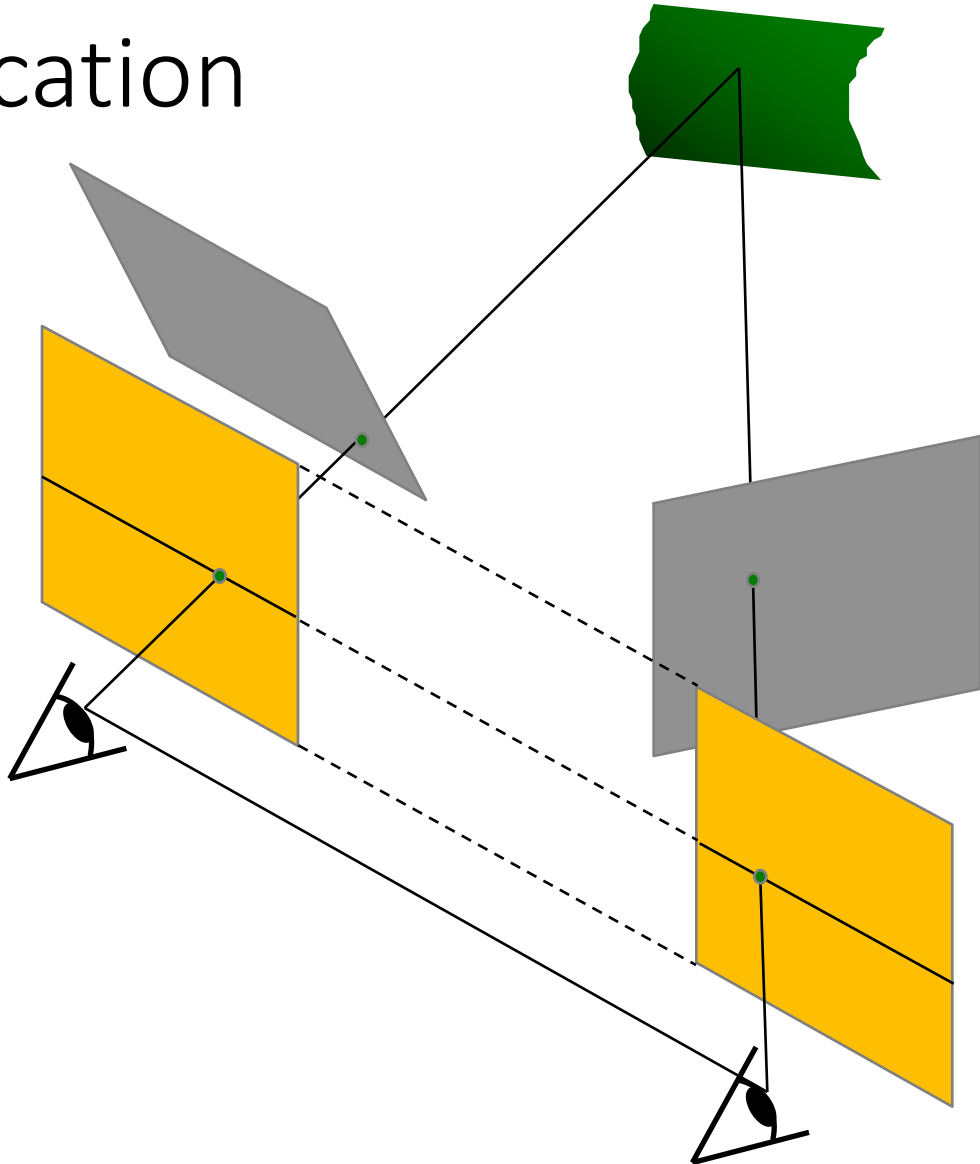
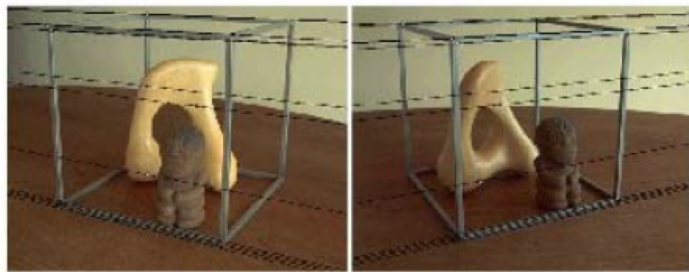


Image Rectification

- [Loop and Zhang 1999]



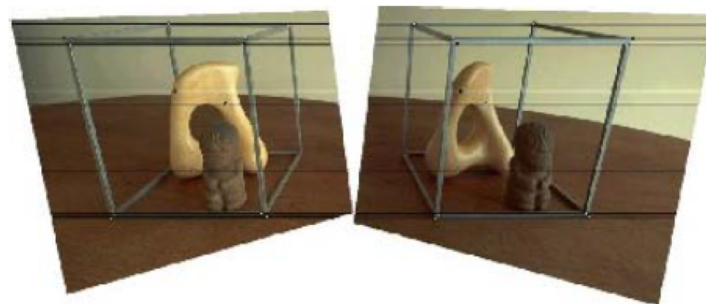
Original image pair overlaid with several epipolar lines.



Images transformed so that epipolar lines are parallel.



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



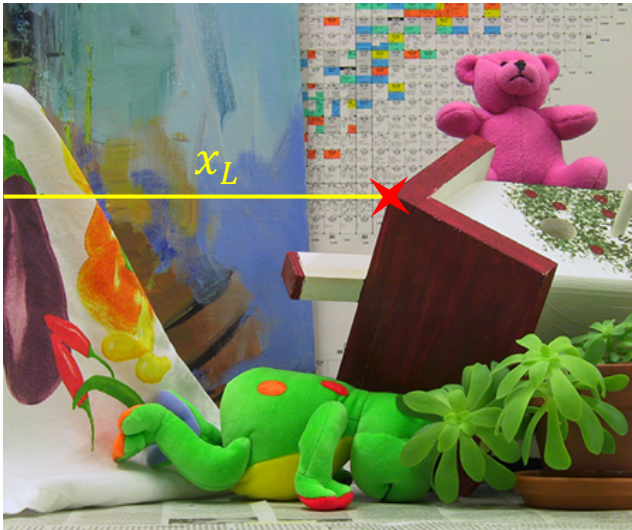
Final rectification that minimizes horizontal distortions. (Shearing)

You don't need to know image rectification now....

For the task of Stereo Matching in recent research, we consider two images that are well-rectified.

Disparity Estimation

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



Disparity Estimation

SSD :Sum of Squared Distance

- The “hello world” algorithm: block matching

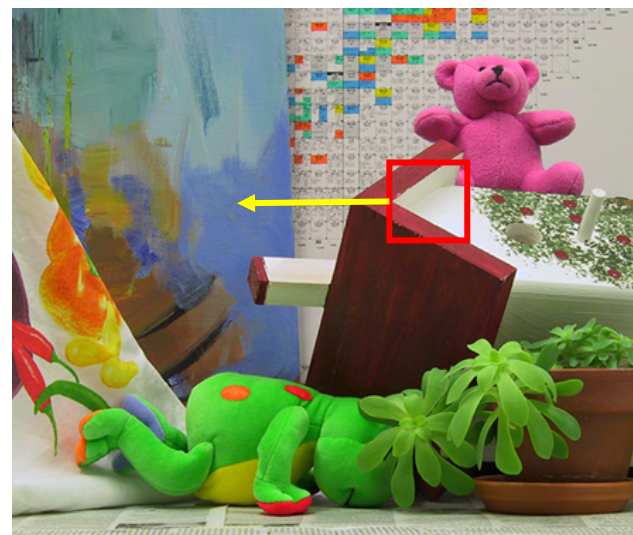
- Consider SSD as matching cost

Winner take all (WTA)

d	0	1	2	3	...	33	...	59	60
SSD	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

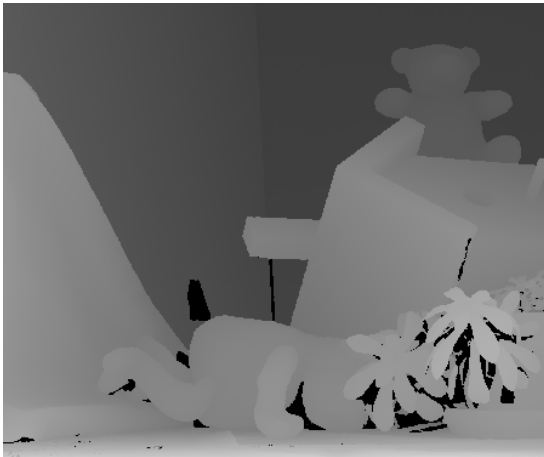
Disparity Estimation

- Reverse order of loops

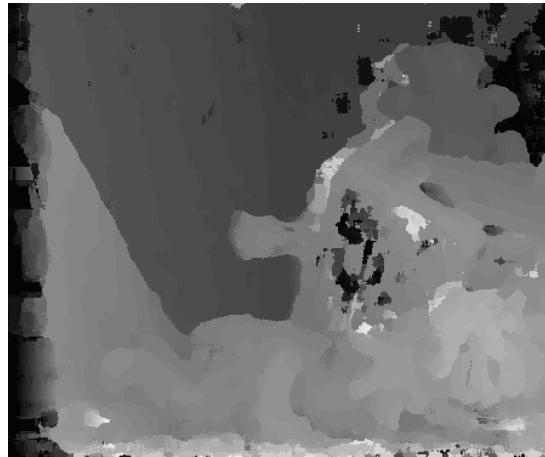
- For each disparity in the left image
 - For each pixel
 - For each pixel in window
 - Compute matching cost
- Find disparity with minimum matching cost at each pixel

Disparity Estimation

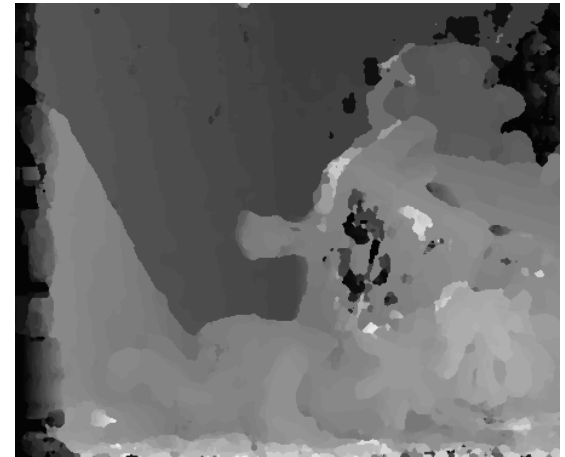
- Block matching result



Ground-truth

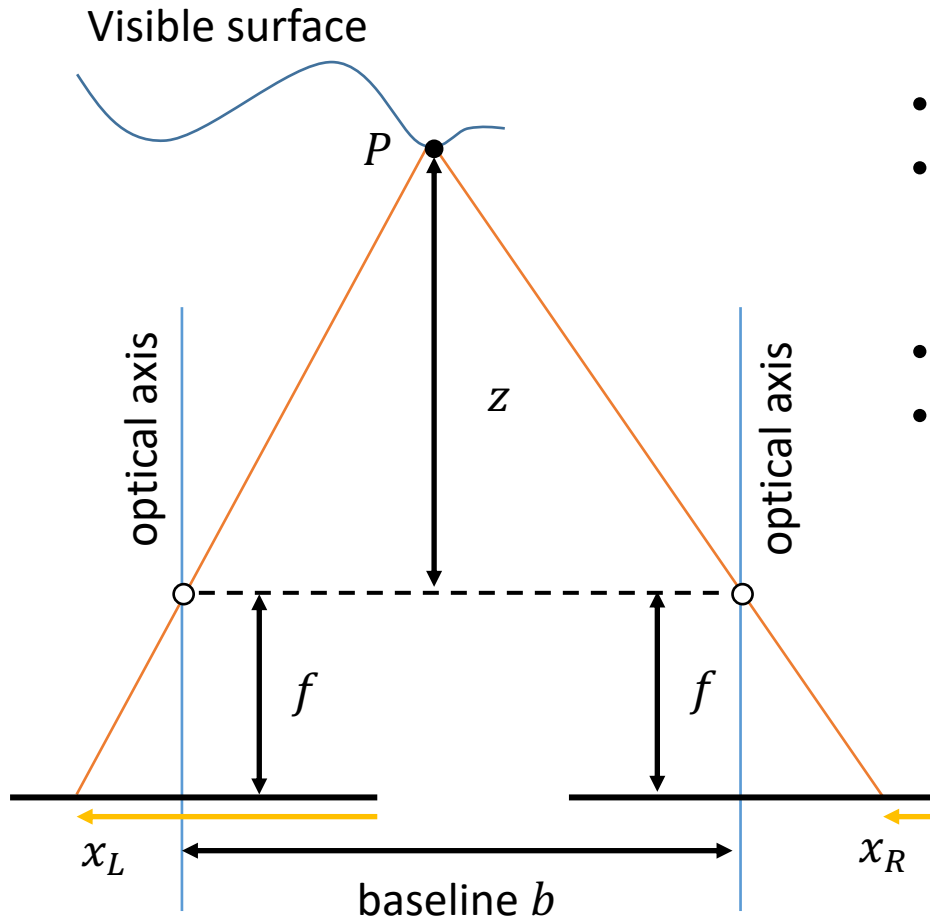


Window 5x5



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

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

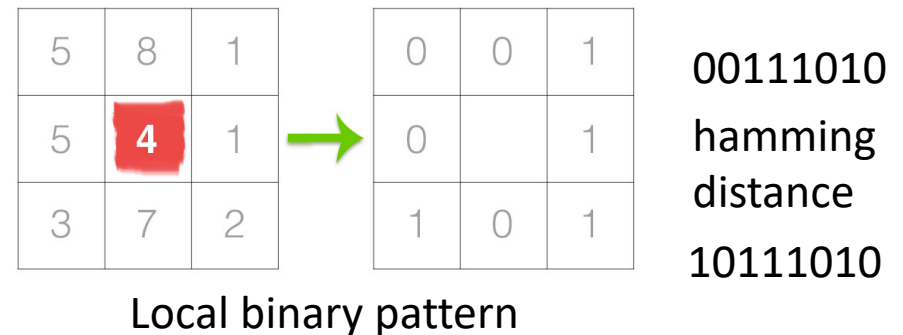
- Cost computation
- Cost (support) aggregation
- Disparity optimization
- Disparity refinement

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 <i>et al.</i> [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 <i>et al.</i> [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 <i>et al.</i> [127]	cross-correlation	non-linear filtering	hier. WTA
Shah [103]	squared difference	non-linear diffusion	regularization
Boykov <i>et al.</i> [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 <i>et al.</i> [46]	squared difference	shiftable window	DP
Belhumeur [9]	squared difference	-none-	DP
Cox <i>et al.</i> [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 <i>et al.</i> [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 <i>et al.</i> [117]	squared difference	(color segmentation)	WTA + regions

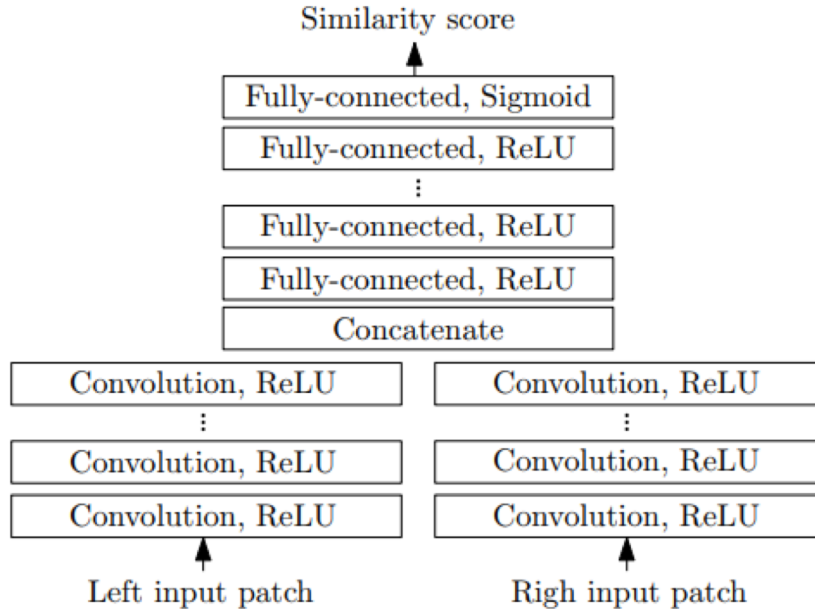
Matching Cost

- 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
 - $C = \min(C_0, \tau)$



Matching Cost

- Deep matching cost (MC-CNN)



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

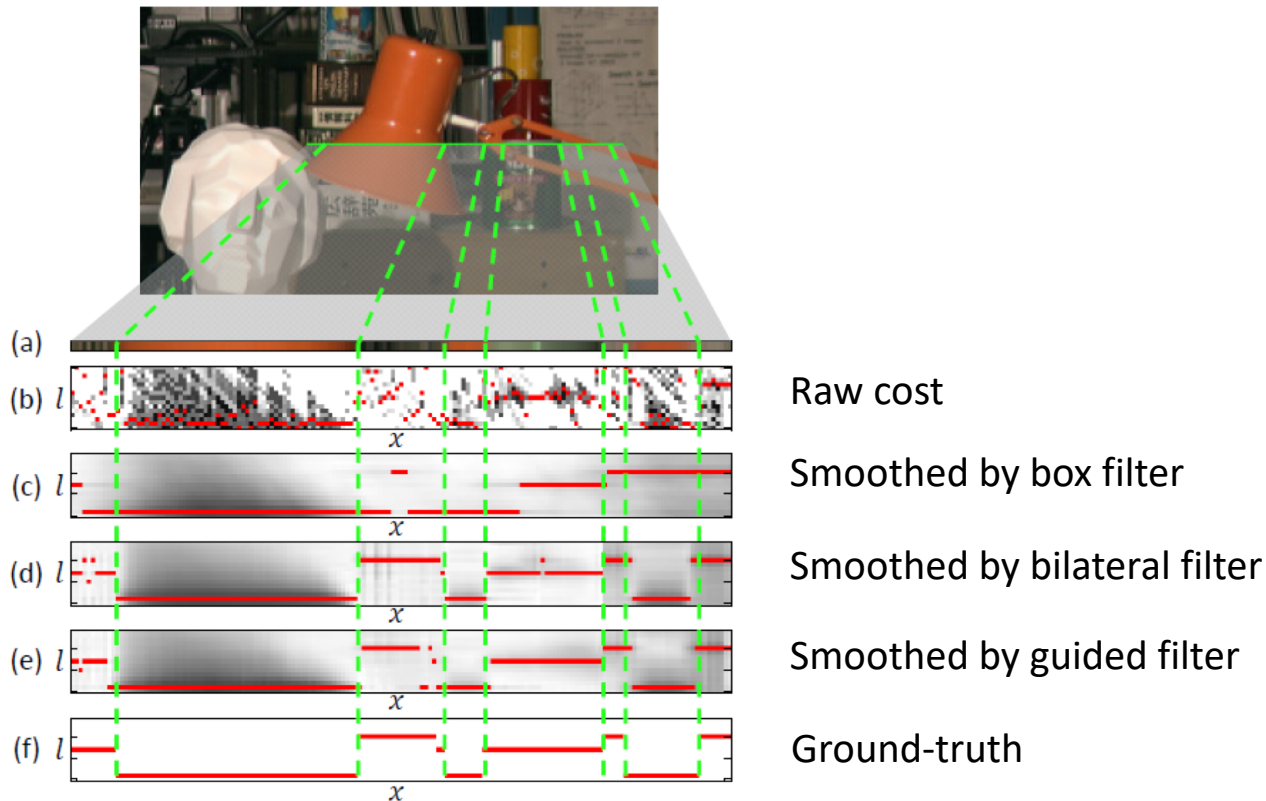
<https://github.com/jzbontar/mc-cnn>

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

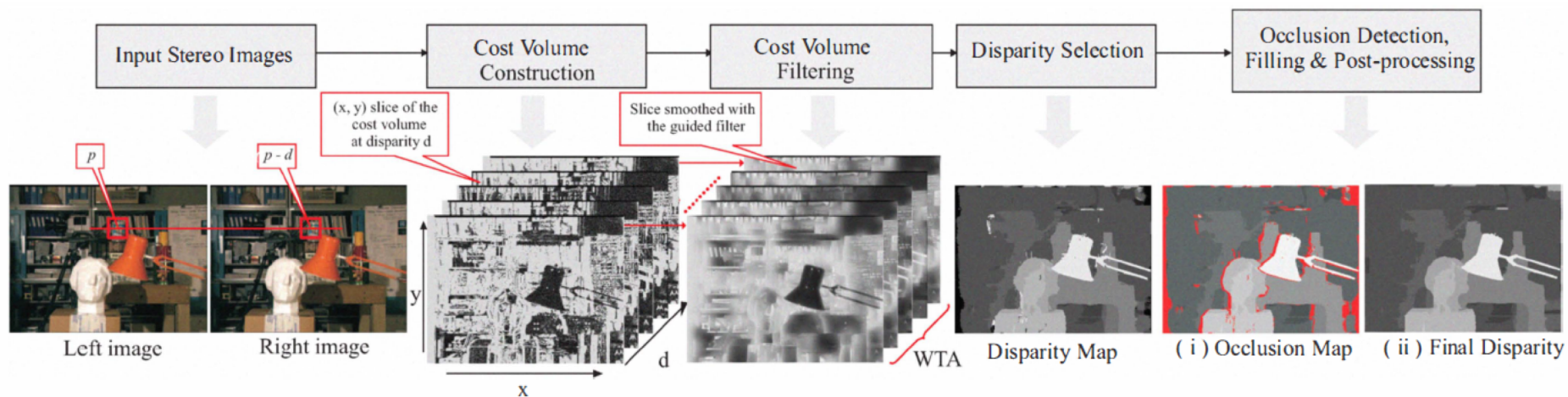
Cost-Volume Filtering

- Illustration of the matching cost



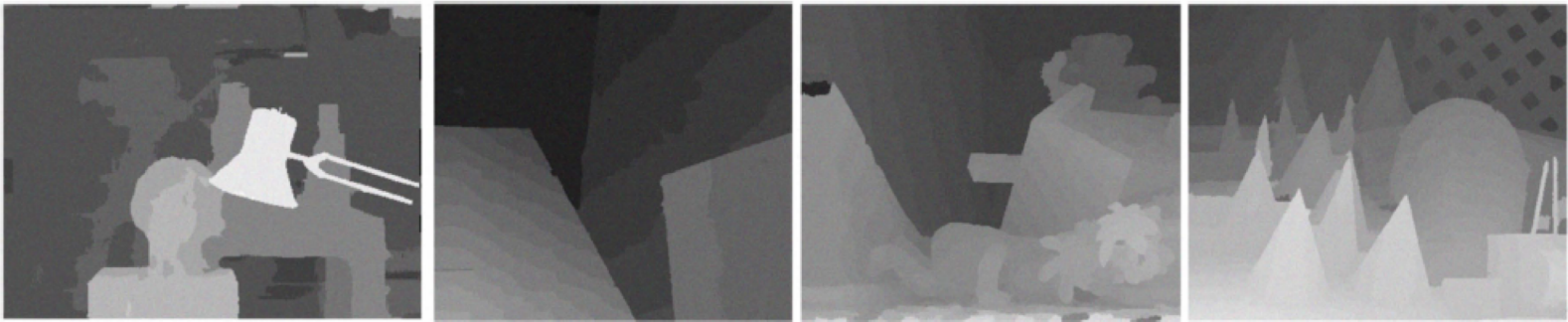
Cost-Volume Filtering

- 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

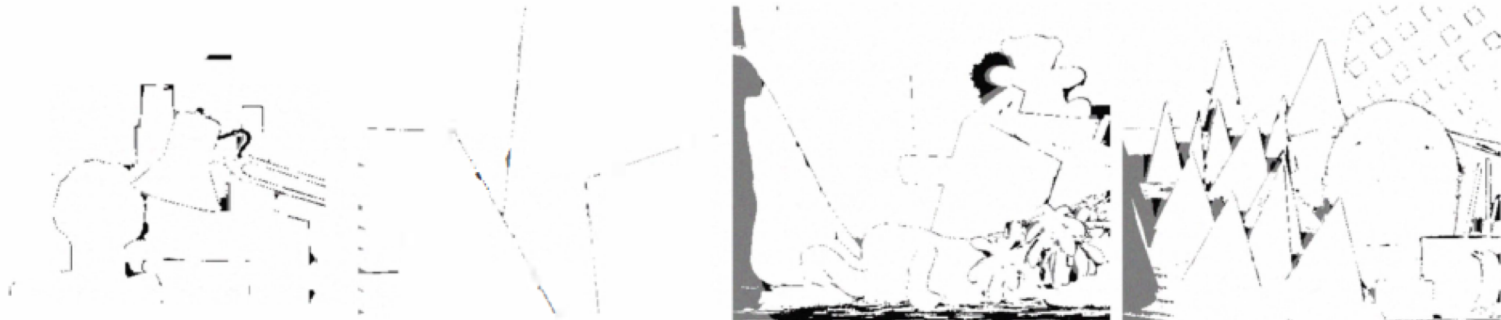


Cost-Volume Filtering

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*



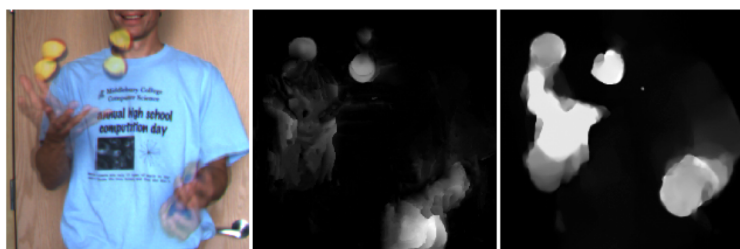
Disparity Maps



Error Maps

Cost-Volume Filtering

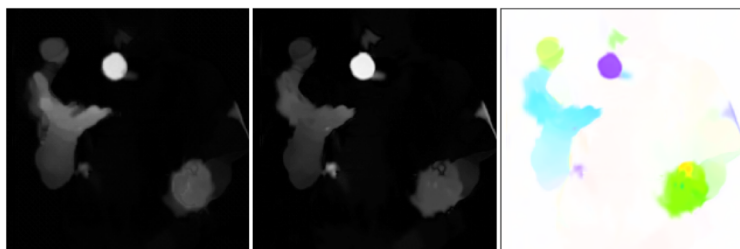
- Cost-volume filtering is a general framework and can be applied to other discrete labeling problems
 - Optical flow: labels are displacements
 - Segmentation: labels are foreground/background



(a) Input images

(b) Steinbrücker et al. [23]

(c) LDOF [5]



(d) ACK-Prior [14]

(e) Ours

(f) Our flow



(a) Input image with user interaction

(b) Our cutout

(c) Cutout using Graph Cuts

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

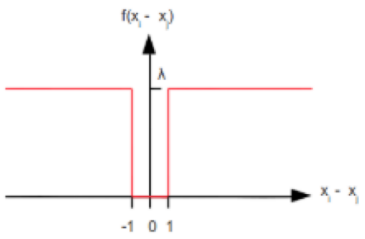
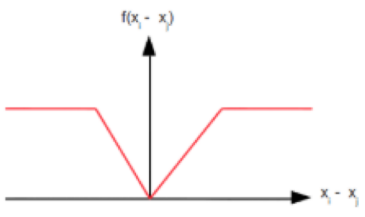
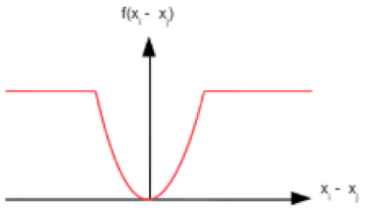
- A good stereo correspondence
 - Match quality: each pixel finds a good match in the other image
 - Smoothness: disparity usually changes smoothly
- Mathematically, we want to minimize:

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

- 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, ...

	$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ \lambda & \text{otherwise} \end{cases}$ <p>Also known as the Potts model.</p>
	$f(n) = \lambda \times \min(n , K)$ <p>Truncated linear model.</p>
	$f(n) = \lambda \times \min(n^2, K)$ <p>Truncated quadratic model.</p>

Belief Propagation

Algorithm	Avg. Rank	
IGSM [155]	10.4	0
TSGO [141]	13.4	1
JSOSP+GCP [149]	14.8	1
KADI [164]	15.2	1
SSCBP [157]	17.6	1
ADCensus [82]	18.2	1
AdaptingBP [16]	22.2	1
CoopRegion [39]	22.2	1
CCRADAR [150]	26.8	1
PM-Forest [162]	27.2	1
RDP [87]	28.7	0
MultiRBF [129]	28.7	1
DoubleBP [34]	29.0	1
OutlierConf [40]	30.0	1
SegAggr [144]	30.2	1
CVW-RM [146]	30.4	1
GC+LocalExp [158]	32.0	1
SOS [135]	35.0	1
SubPixSearch [109]	35.4	2
AdaptiveGF [127]	35.8	1
GcKuwaGrad [161]	36.5	1
SurfaceStereo [71]	37.0	1
SubPixDoubleBP [29]	37.2	1

Snapshot from Middlebury v2

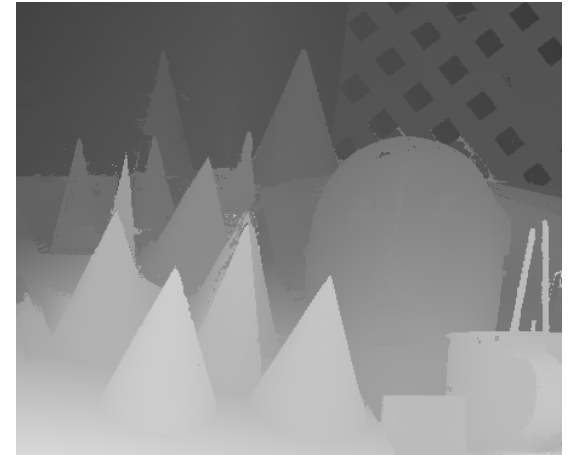
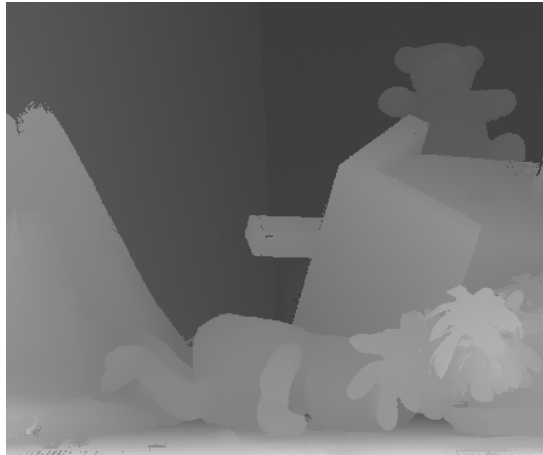
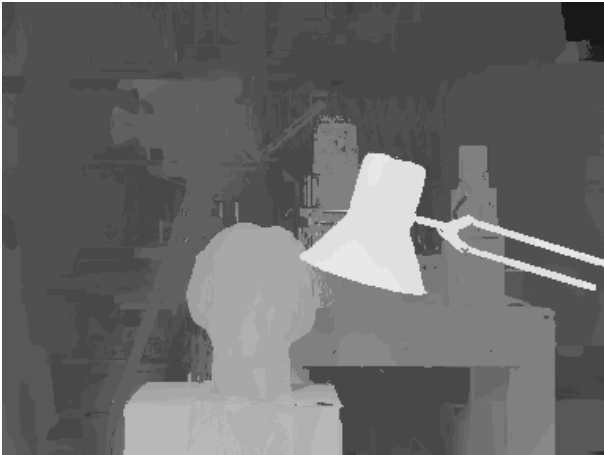
Yang et al. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In CVPR 2006.

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 Tanai et al.

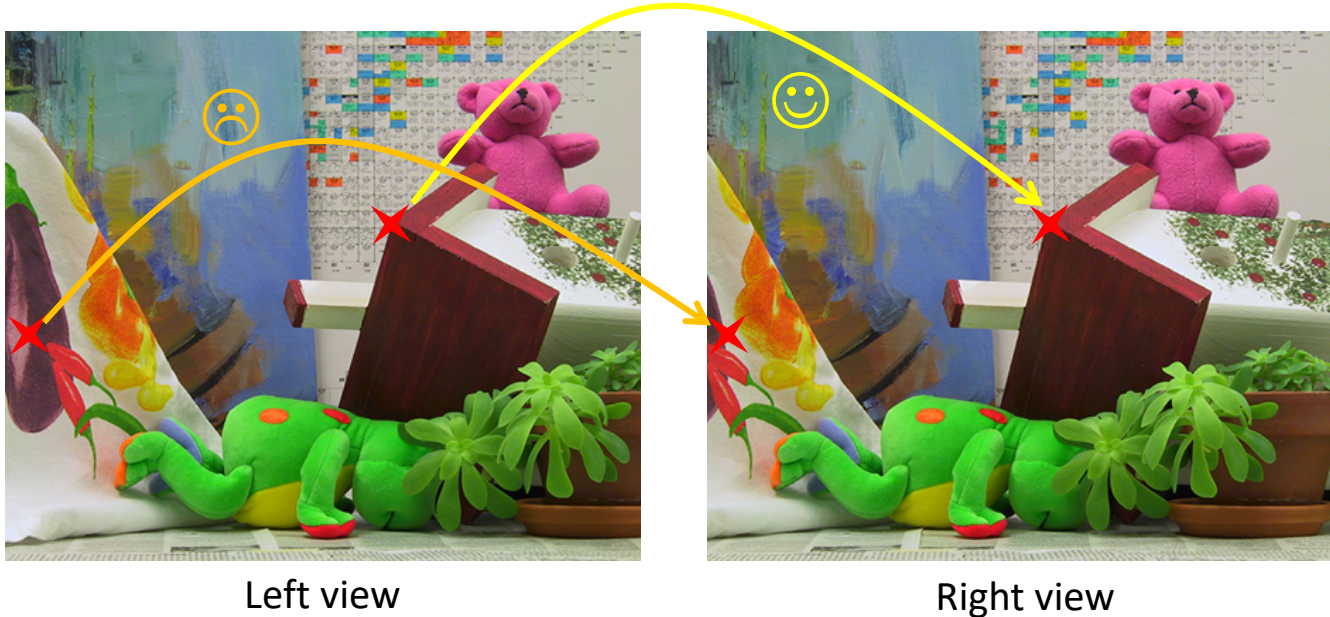
Tanai et al. Graph cut based continuous stereo matching using locally shared labels. In CVPR 2014.
Kolmogorov et al. What energy functions can be minimized via graph cuts? PAMI 2004
Boykov et al. Fast approximate energy minimization via graph cuts. In ICCV 1999.

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

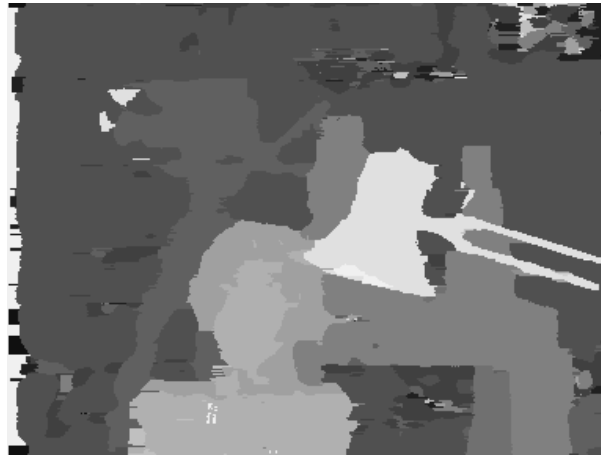
Disparity Refinement

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



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)
 - (You need to handle the boundary, pad maximum???)



The above steps do not guarantee coherency between scanlines

Disparity Refinement

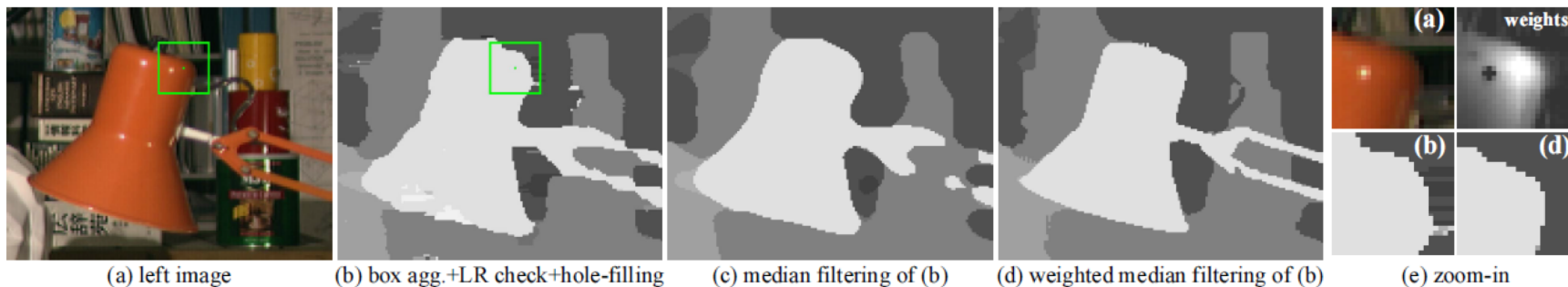
$$w_{pq} = g(\mathbf{f}(p), \mathbf{f}(q))$$

$$g = \exp\left(-\frac{\|\mathbf{f}(p) - \mathbf{f}(q)\|}{\sigma^2}\right)$$

- Weighted median filtering

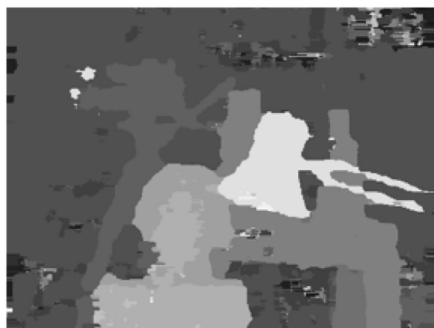
Denoting $I(q)$ as the value at pixel q in image I and $n = (2r + 1)^2$ as the number of pixels in $\mathcal{R}(p)$, we express the value and weight elements for all pixels in $\mathcal{R}(p)$ as $\{(I(q), w_{pq})\}$. By sorting values in an ascending order, the weighted median operator for pixel p returns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

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



Disparity Refinement

- Weighted median filtering



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



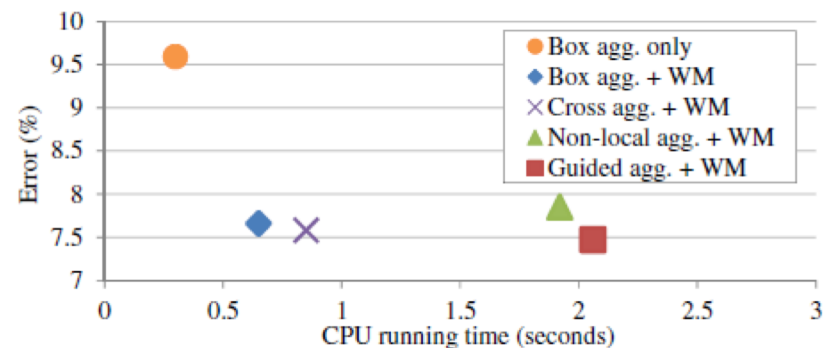
(b) guided agg. + LR check + hole-filling
246ms, Err. 2.08



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



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



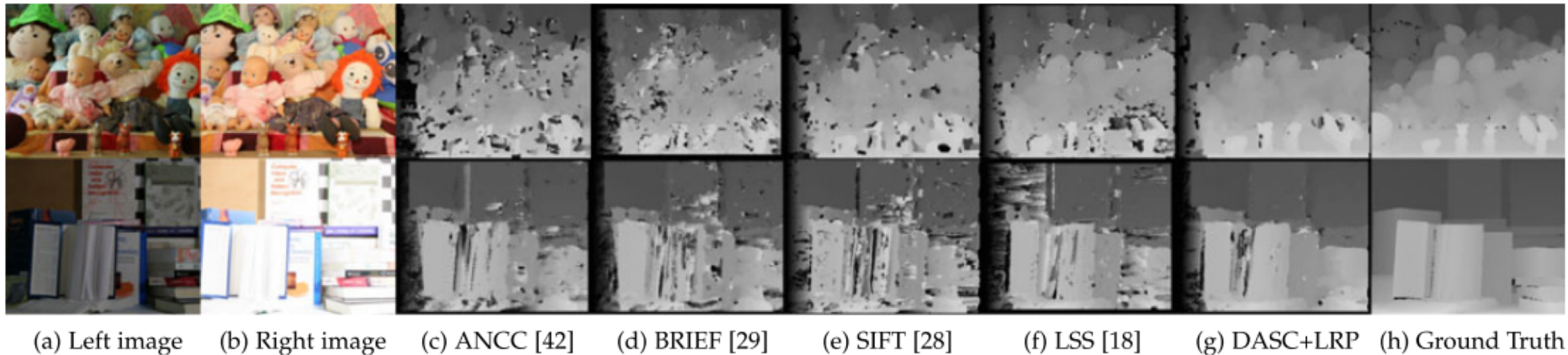
Ours and previous work	Avg. error	GPU time
NonLocalFilter [30]	5.48 [†]	n/a
Guided agg. + WM	5.50	54ms
CostFilter [22]	5.55 [†]	65ms [†]
Cross agg. + WM	5.63	n/a
Non-local agg. + WM	6.04	n/a
Box agg. + WM	6.19	22ms
VariableCross [34]	7.60 [†]	n/a

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)



Xu et al. Linear time illumination invariant stereo matching. IJCV 2016.

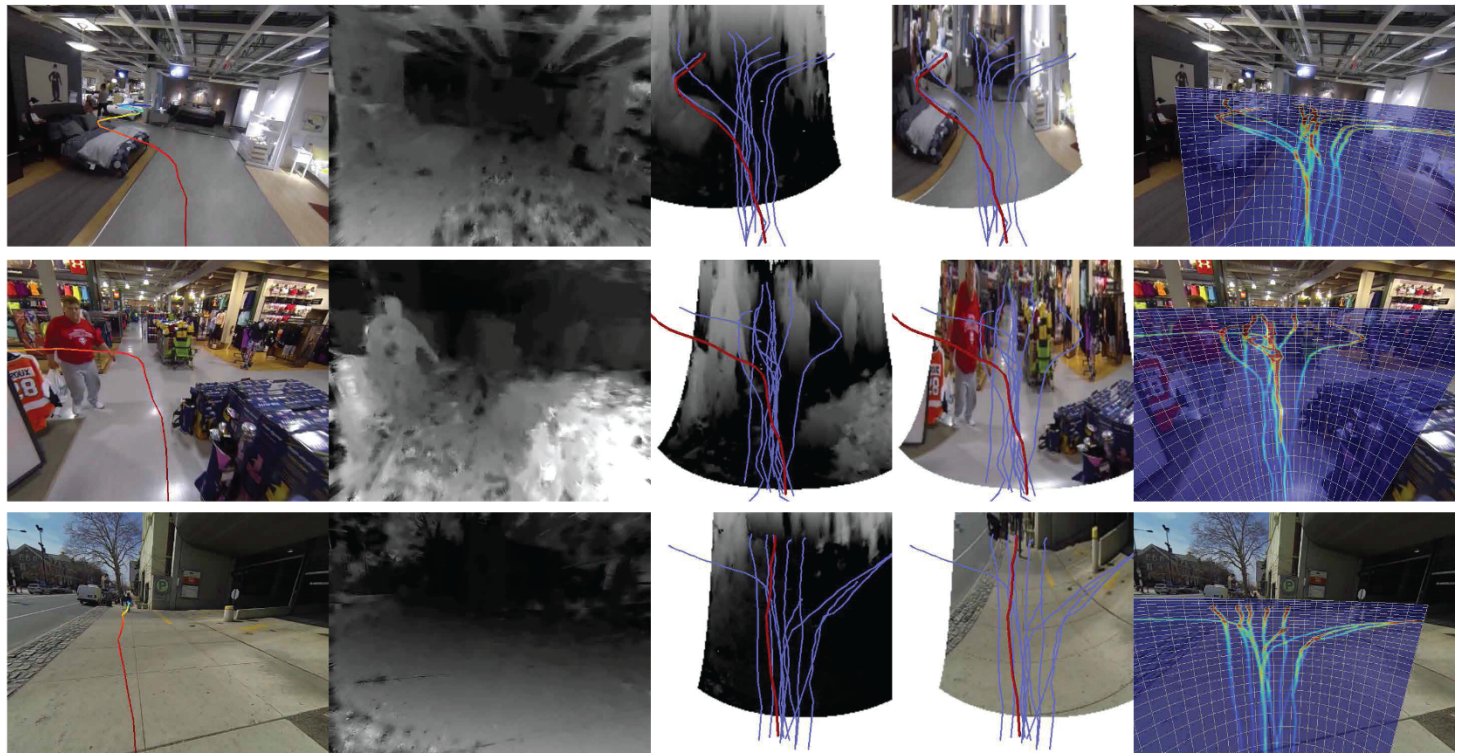
Kim et al. DASC: robust dense descriptor for multi-modal and multi-spectral correspondence estimation. PAMI 2017.

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



Input: RGB image

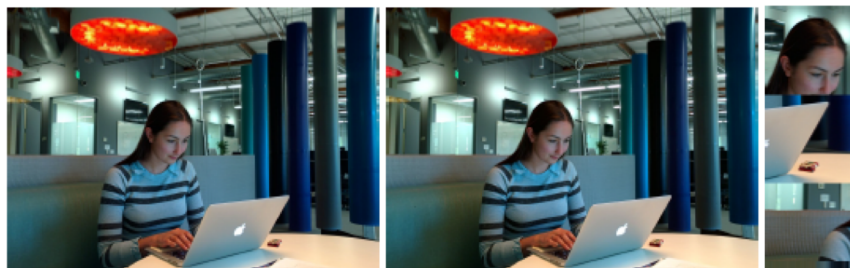
Input: depth image

EgoRetinal map

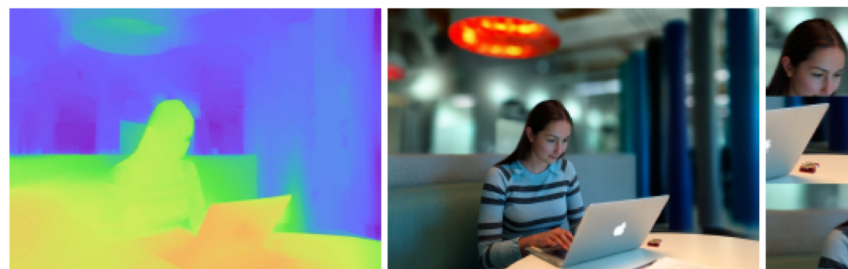
Egomotion prediction

Application: Synthetic Defocus

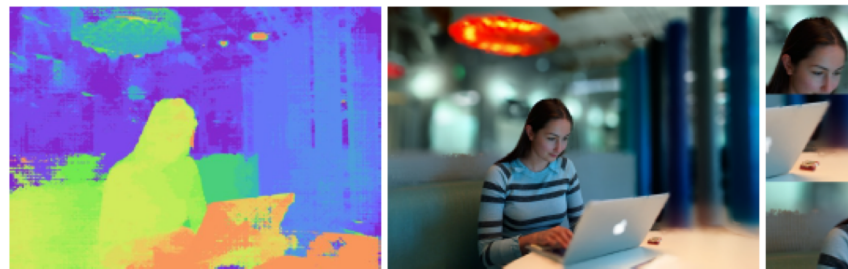
- A stereo algorithm tailored for synthetic defocus application



(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

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