

---

# Robot Vision: Stereo Matching

Prof. Friedrich Fraundorfer

SS 2022

# Outline

---

- Geometric relations for stereo matching
- Dense matching process
- Census Transform
- Dynamic programming
- Semiglobal matching
- Stereo matching with CNN's
- Monocular depth estimation

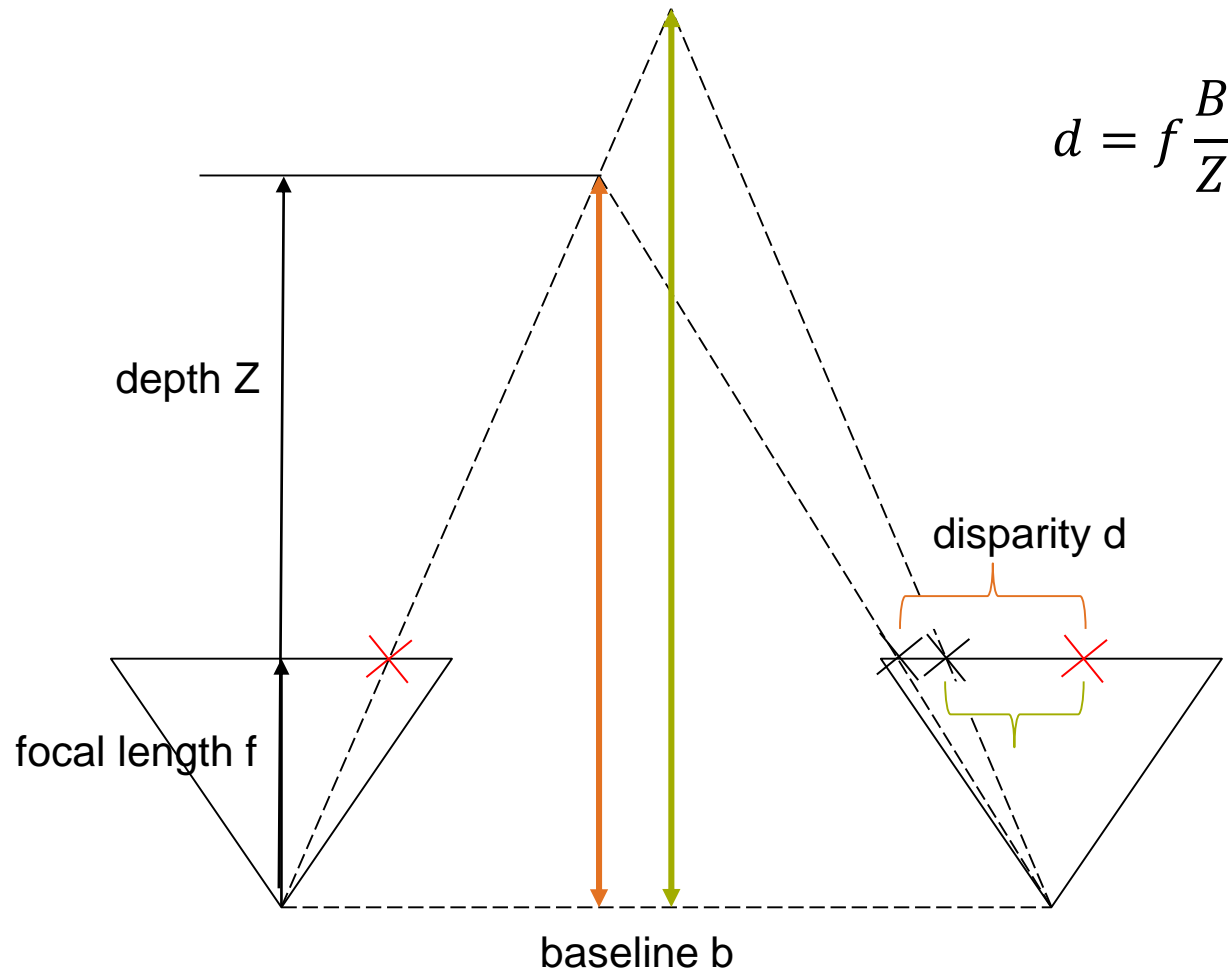
# Dense matching

---

- SfM only gives sparse 3D data
- Only feature points (e.g. SURF) are triangulated – for most pixel no 3D data is computed
- Dense image matching computes a 3D point for every pixel in the image (1MP image leads to 1 million 3D points)
- Dense matching algorithms need camera poses as prerequisite

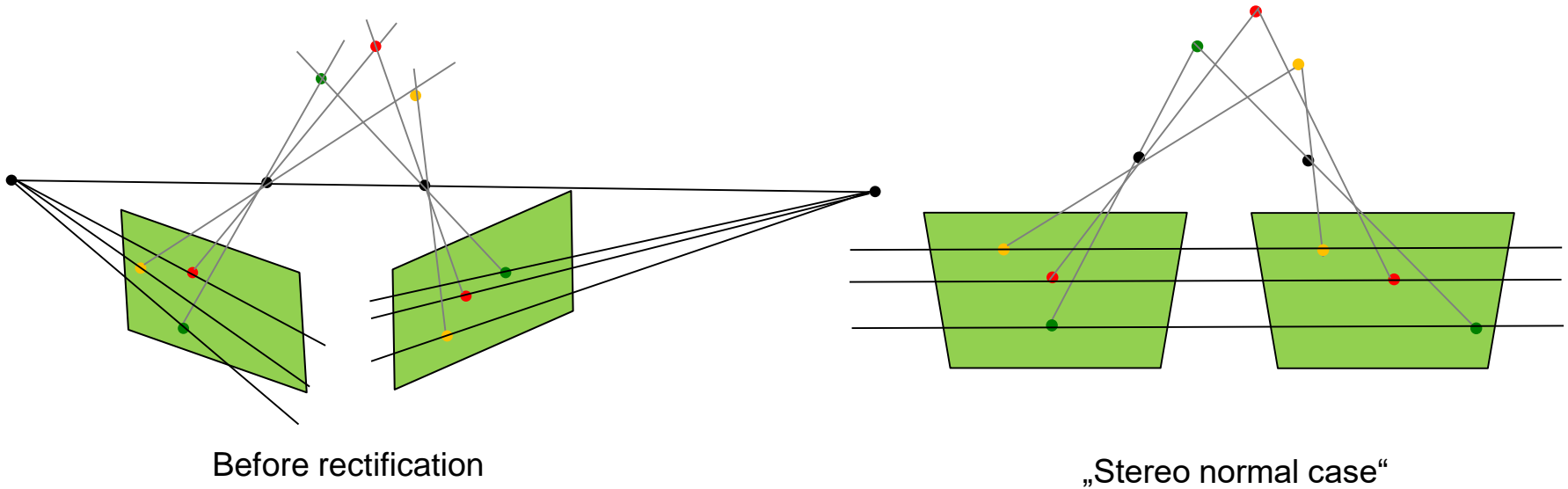
# Geometric relation

- Stereo normal case
- Depth  $Z$  [m] can be computed from disparity  $d$  [pixel]

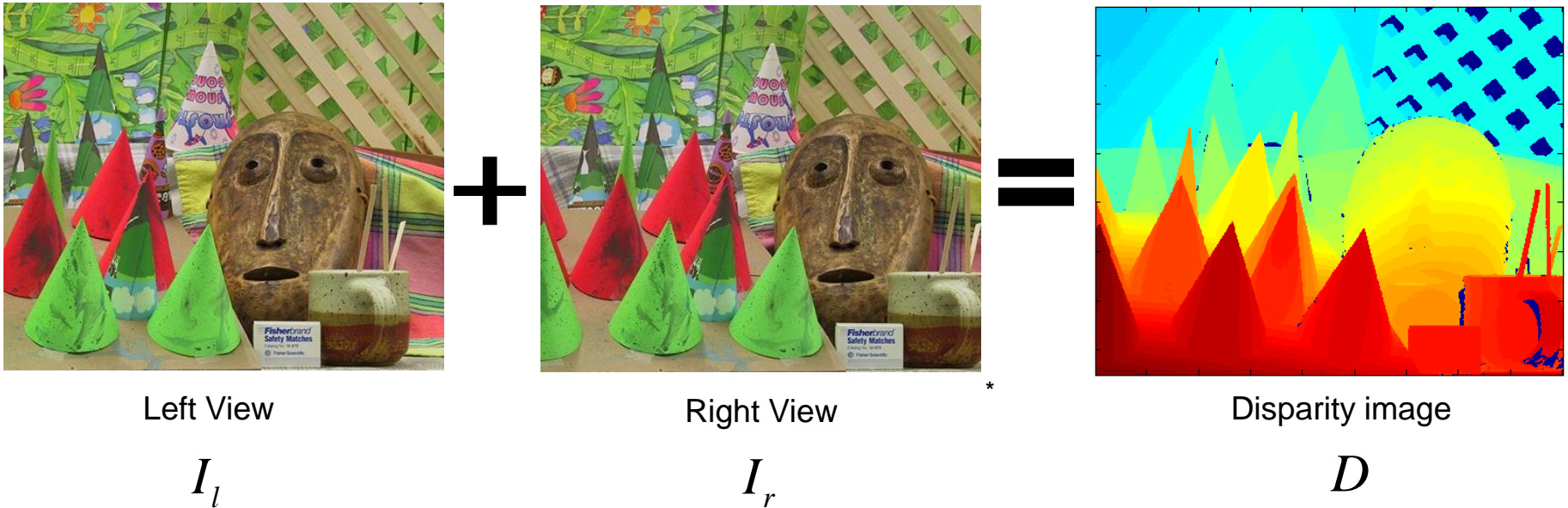


# Rectification

- Image transformation to simplify the correspondence search
  - Makes all epipolar lines parallel
  - Image x-axis parallel to epipolar line
  - Corresponds to parallel camera configuration



# Dense matching process

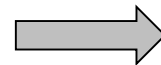


- Estimate disparity (depth) for all pixels in the left image.
  - Evaluate similarity measure for every possible pixel location on the line (e.g. NCC, SAD)
- Disparity  $d$ : Offset between pixel  $p$  in the left image and its corresponding pixel  $q$  in the right image.

# Census Transform

- A popular block matching cost
- Good robustness to image changes (e.g. brightness)
- Matching cost is computed by comparing bit strings using the Hamming distance (**efficient**)
- Bit strings encode if a pixel within a window is greater or less than the central pixel (0 .. if center pixel is smaller, 1 .. if center pixel is larger or equal)

89	63	72
67	<b>55</b>	64
58	51	49



00000011

# Dense matching process

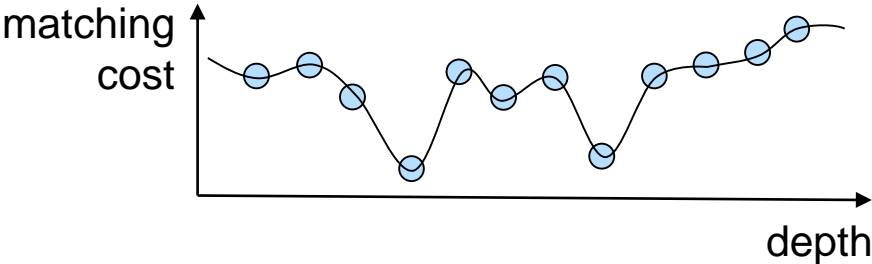
reference image



matching image



epipolar line





# Disparity selection

---

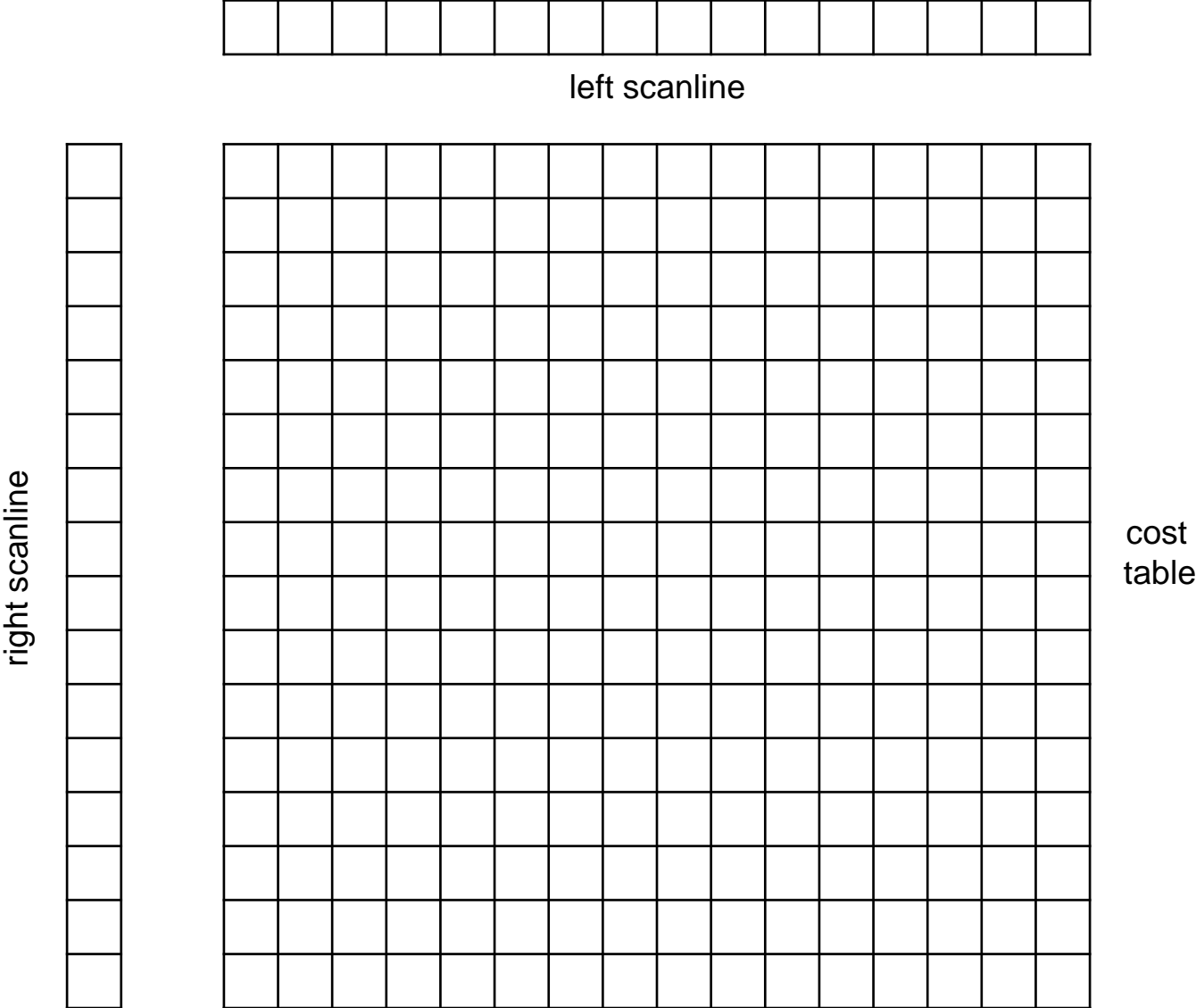
- Single scanline based
  - Winner takes all (WTA)  
Select the disparity with the lowest cost (i.e. the highest similarity)
  - Scanline optimization (Dynamic programming)  
Select the disparities of the whole scanline such that the total (added up) costs for a scanline is minimal
- Global methods (Cost volume optimization)
  - Belief propagation  
Selects the disparities such that the total cost for the whole image is minimal
  - Semi-global Matching  
Approximates the optimization of the whole disparity image

# Scanline optimization

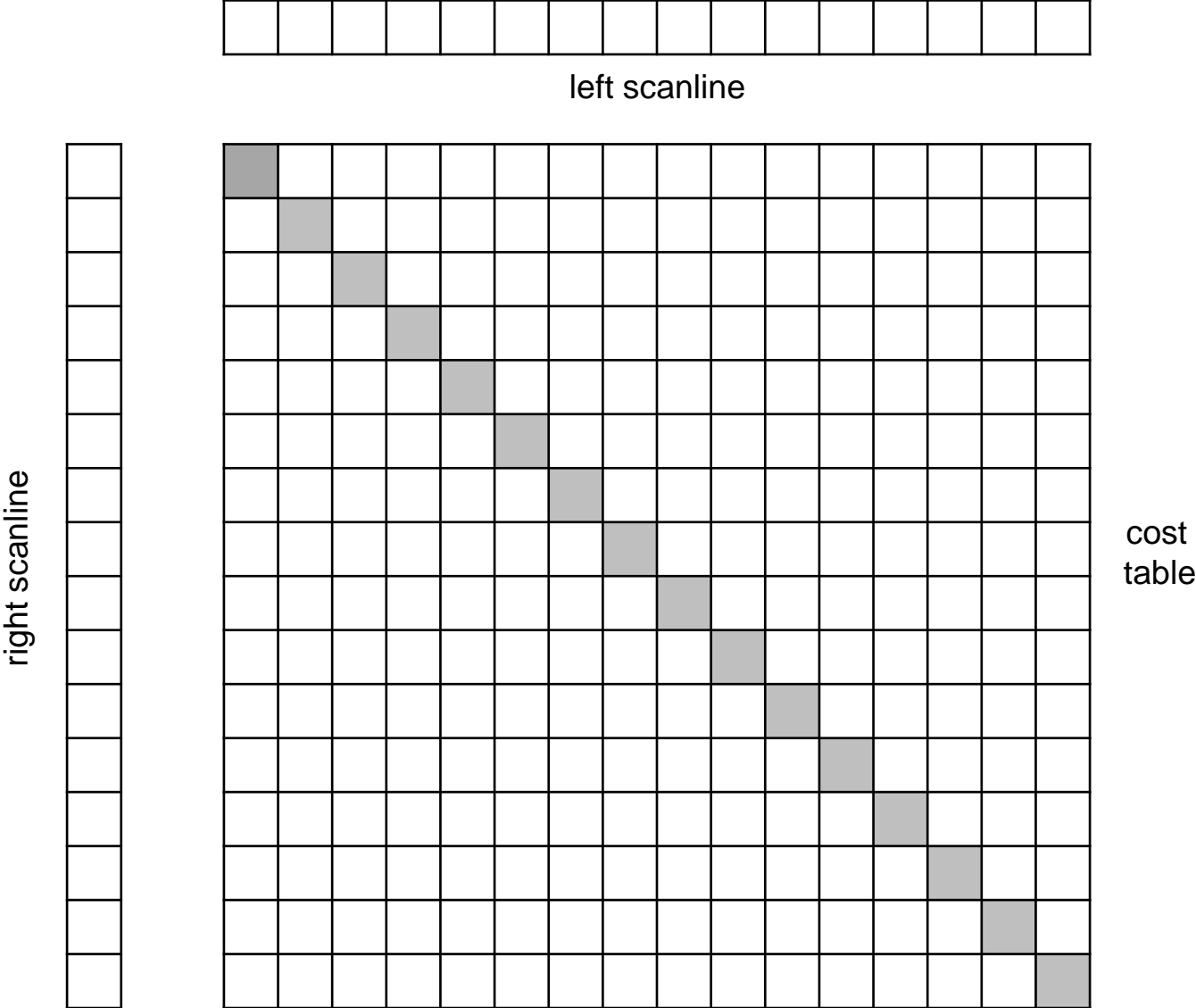
---

- Frequently called “dynamic programming” because of the programming scheme for efficient cost calculation. This naming is historic and does not reflect the method well. In fact it is an application of the Viterbi-Algorithm.
- Cost calculation based on a 2D grid

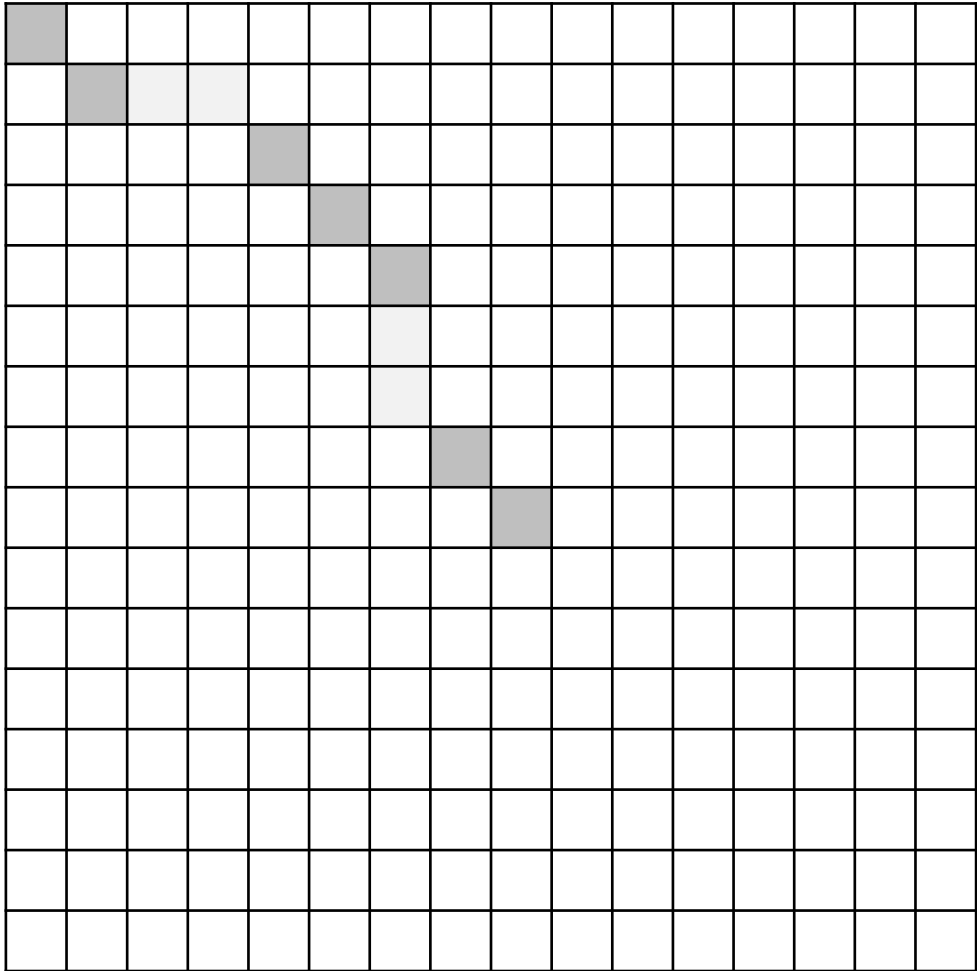
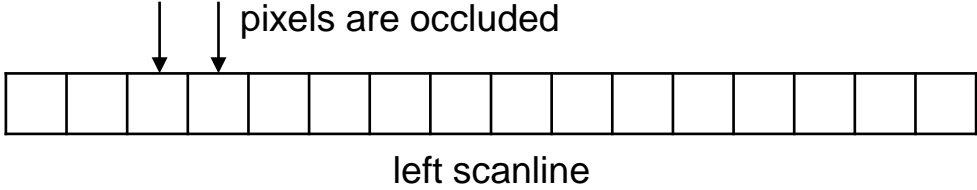
# Scanline optimization



# Scanline optimization



# Scanline optimization



no match for these in left scanline, will be jumped over (depth edge)



# Scanline optimization complexity

---

- Exhaustive search:  $O(h^n)$   
Example: scanline of length  $n=512$  with  $h=100$  disparities:  $100^{512}$
- Dynamic programming:  $O(nh^2)$   
Example: scanline of length  $n=512$  with  $h=100$  disparities:  
 $512 * 100 * 100 = 5,12$  million operations

# Scanline optimization streaking artifacts



# Global methods

---

- Global methods

- Global cost optimization in energy-minimization framework

$$E(D) = E_{data}(D) + \lambda E_{smooth}(D)$$

- Data term:  
Agreement between cost function and input image pair

$$E_{data}(D) = \sum_{(p)} c(p, d)$$

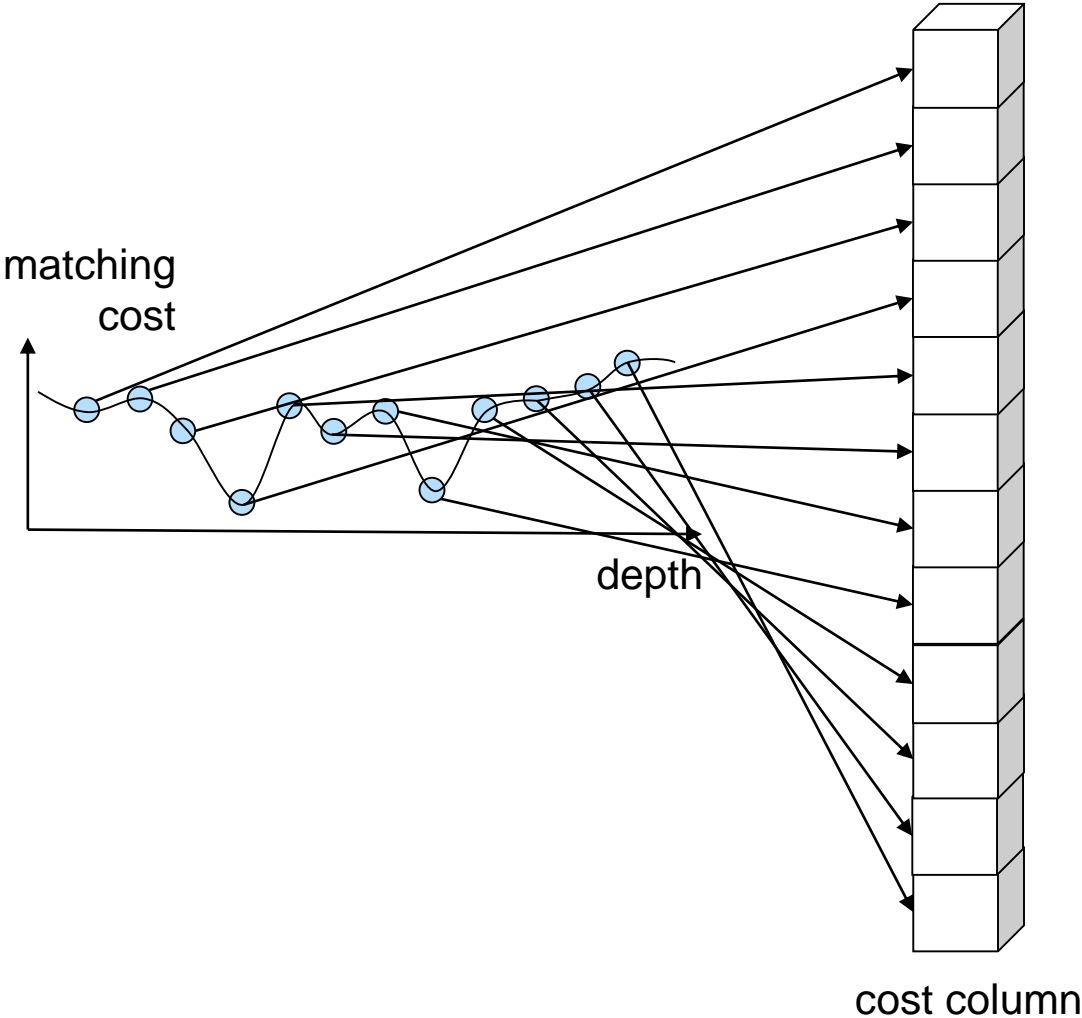
- Smoothness term:  
Encoding the smoothness assumptions

$$E_{smooth}(D) = \sum_{(p)} \rho(d(u, v) - d(u + 1, v))$$



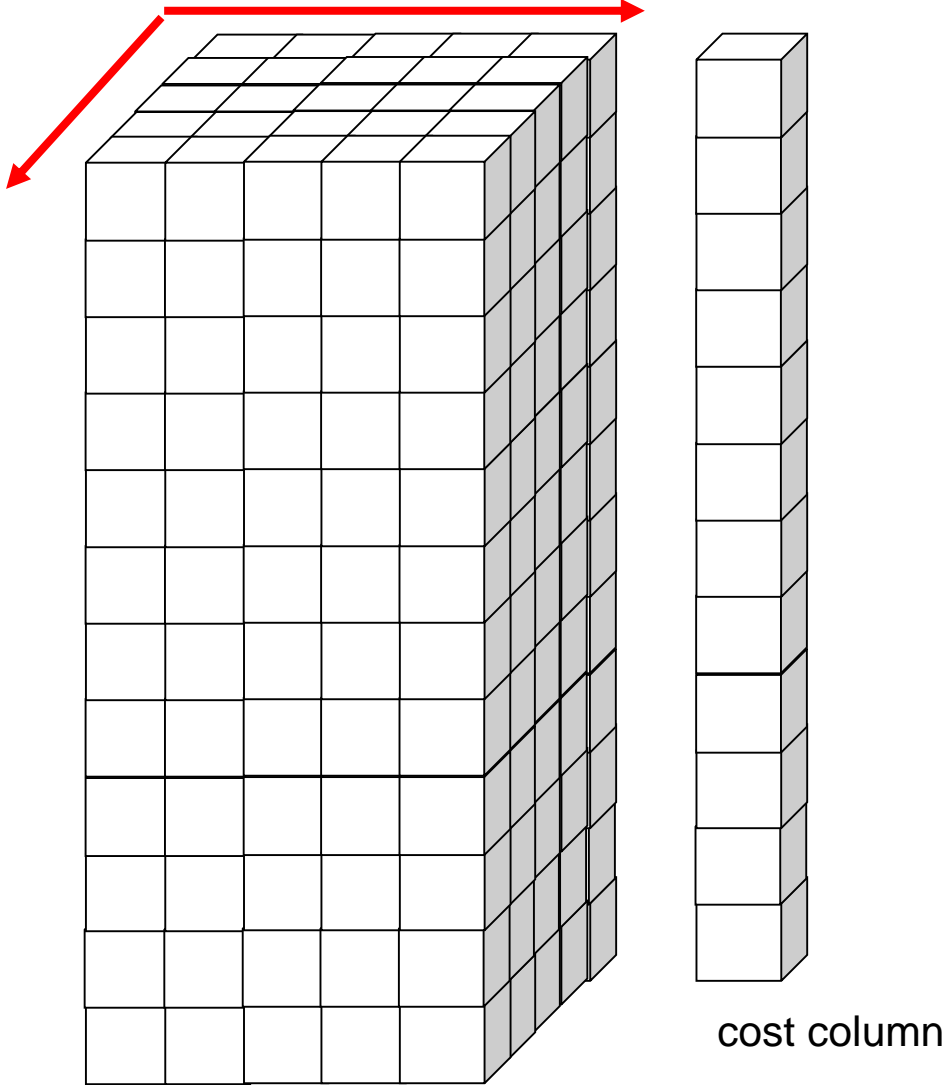
# Cost volume

matching image



# Cost volume

matching image

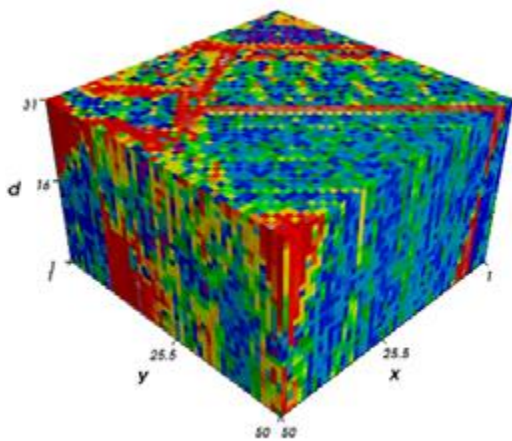


cost volume

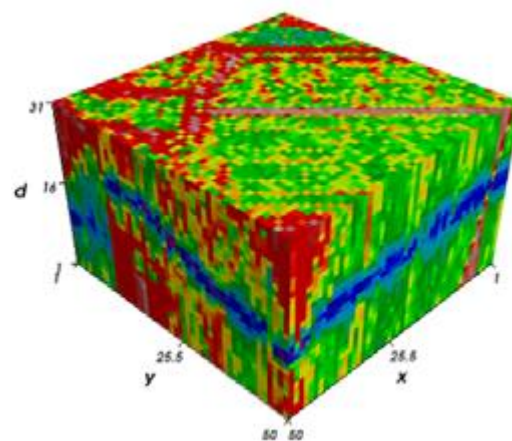
cost column

# Semiglobal matching

- Cost Aggregation (Cost Optimization)



Cost Cube



Optimized Cost Cube

Goal: global minimization of

$$E(D) = \underbrace{\sum_P (C(p, D_p))}_{\text{Data term}} + \underbrace{\sum_{q \in N_p} P_1 [|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 [|D_p - D_q| > 1]}_{\text{Regularization term}}$$

Data term

Regularization term

$P_1$  : Penalty factor for small jump

$P_2$  : Penalty factor for large jump

$N_p$  : Neighborhood of p

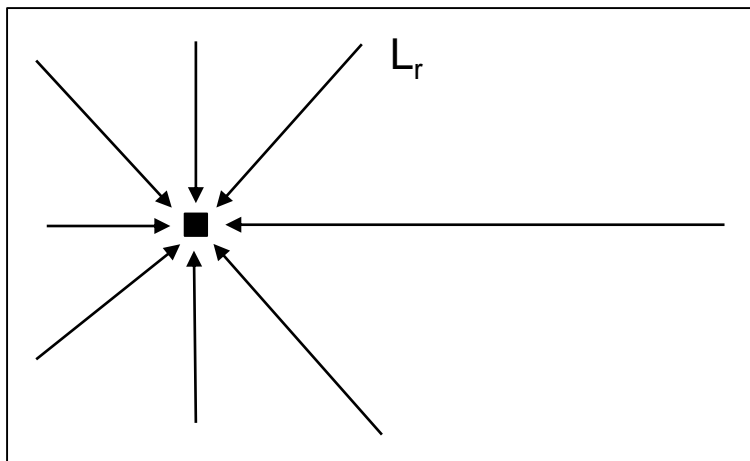
# Semiglobal matching

- Path-wise approximation of aggregation

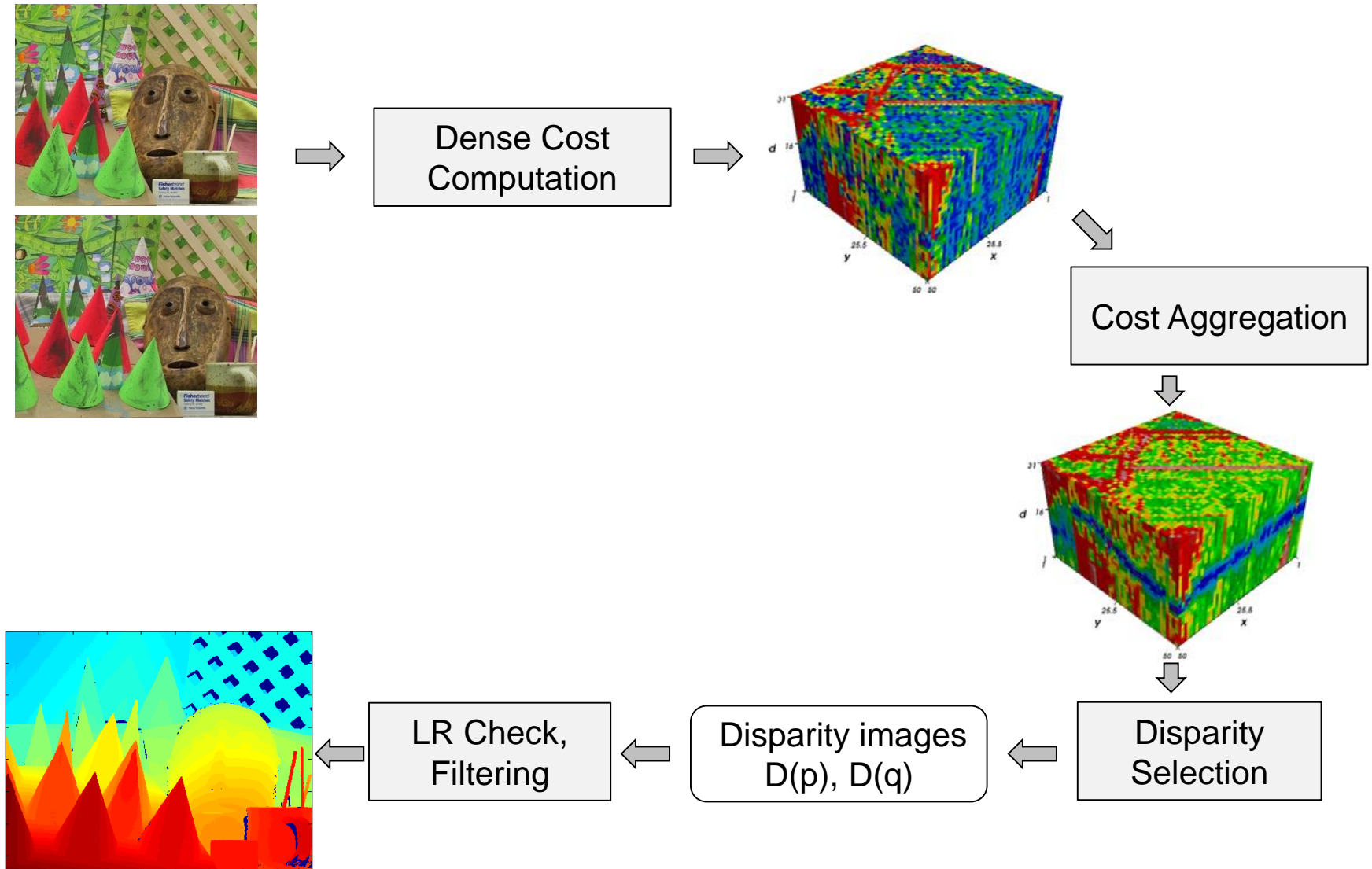
$$L_r(p, d) = C(p, d) + \min \begin{pmatrix} L_r(p-r, d), \\ P_1 + L_r(p-r, d-1), \\ P_1 + L_r(p-r, d+1), \\ P_2 + \min_i L_r(p-r, i) \end{pmatrix}$$

$p$  Image coordinates  
 $P_1$  Cost for small height jump  
 $P_2$  Cost for large height jump  
 $r$  Path direction  
 $L_r$  Aggregated costs along  $r$   
 $d$  Disparity

- Summation of  $L$  along 8 or 16 directions  $r$   $S(p, d) = \sum_r L_r(p, d)$

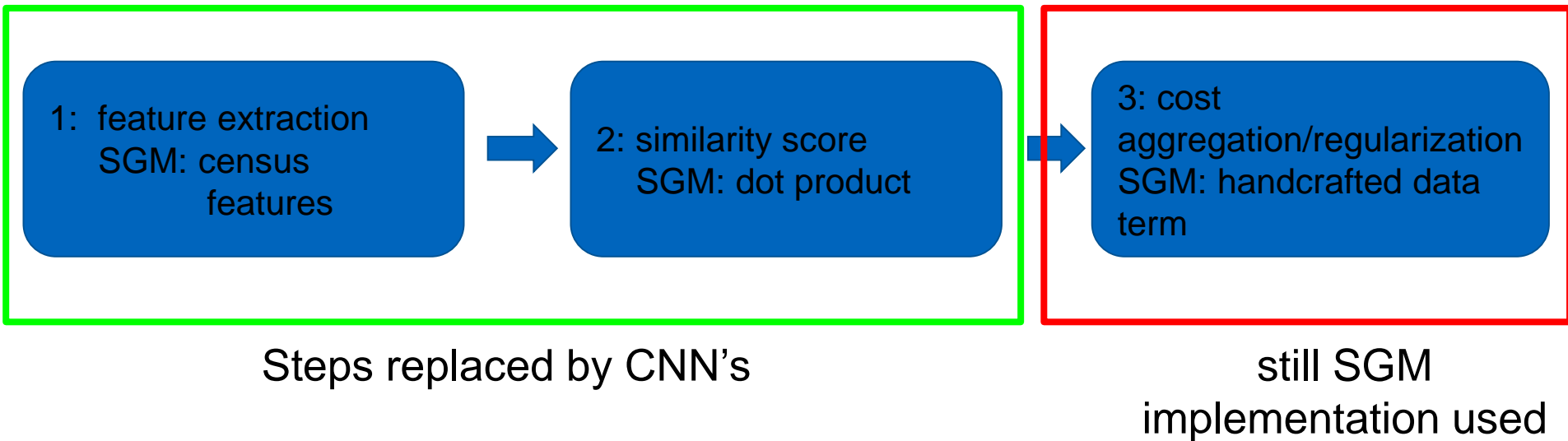


# Semiglobal matching

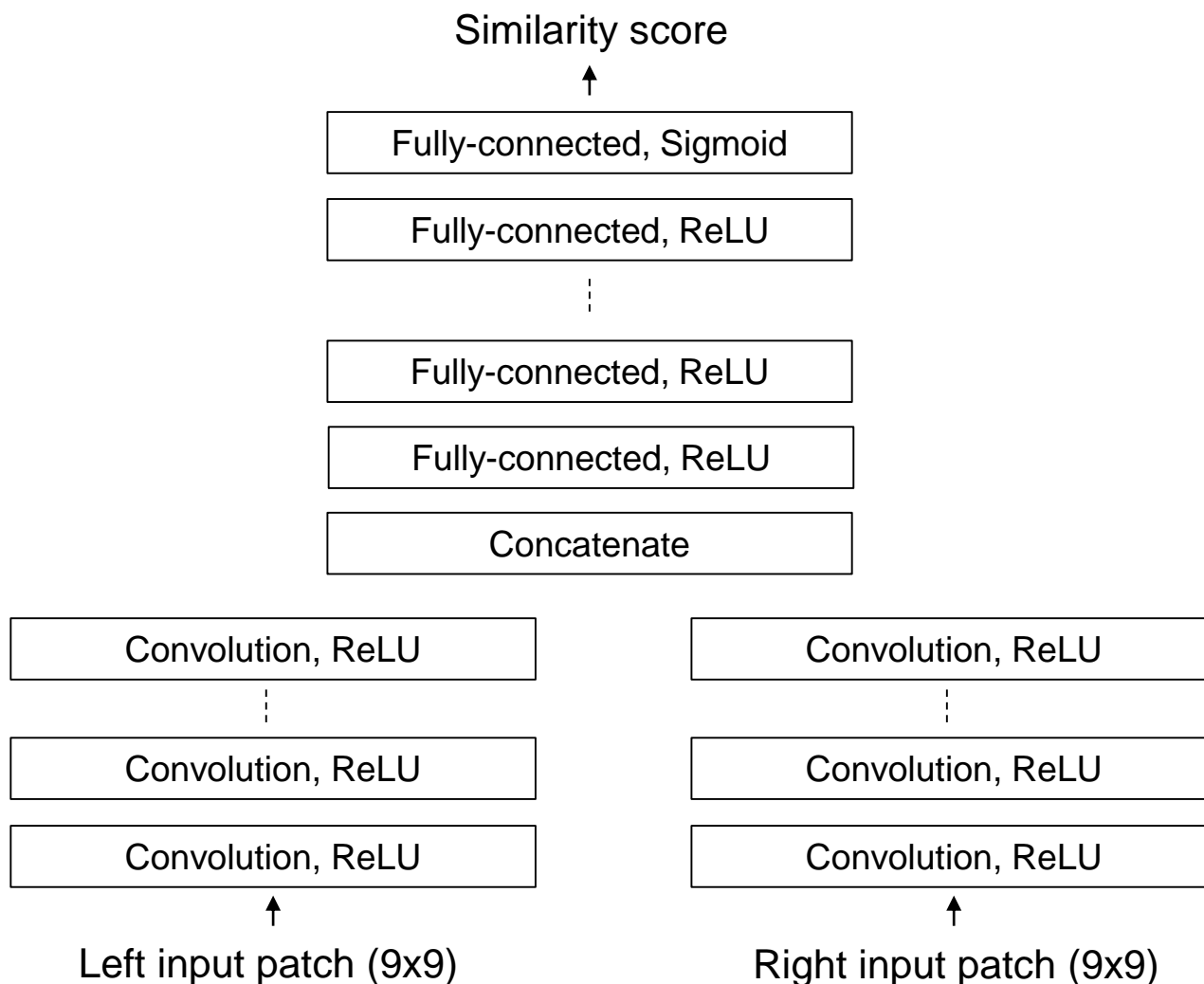


## Stereo matching using CNN's (Example MC-CNN)

- Traditionally disparity estimation works along 3 steps
- CNN's can be used to replace these parts which replaces handcrafted models and thresholds with data-driven algorithms



- Using deep neural networks for similarity estimation



# Performance of CNN based stereo

## Middlebury Stereo Evaluation - Version 3

Mouseover the table cells to see the produced disparity map. Clicking a cell will blink the ground truth for comparison. To change the table type, click the links below. For more information, please see the [description of new features](#).

[Submit and evaluate your own results](#). See [snapshots of previous results](#). See the [evaluation v.2](#) (no longer active).

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](#)

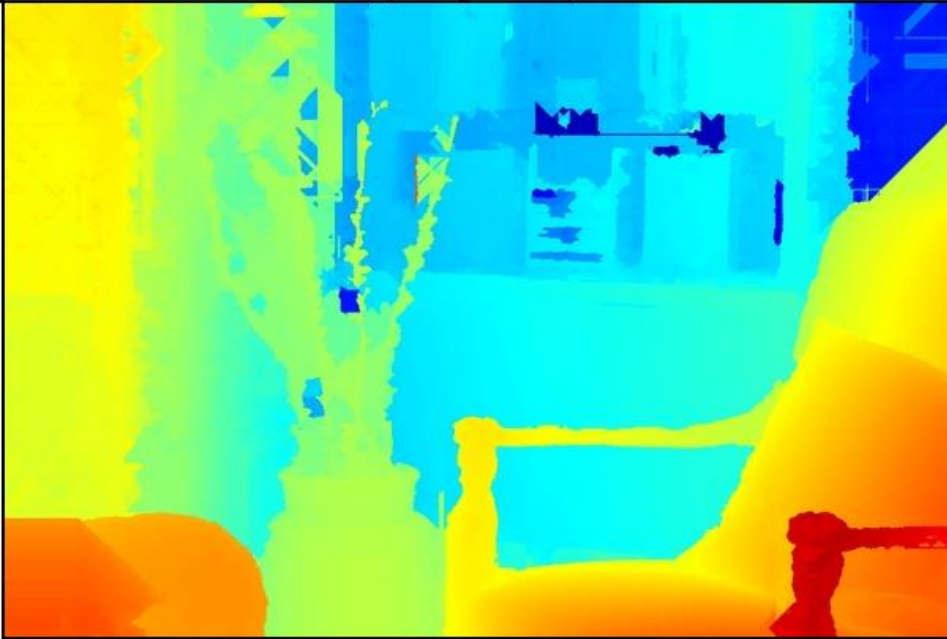
plot selected  show invalid [Reset sort](#) [Reference list](#)

Date	Name	Res	Weight Avg	bad 2.0 (%)															
				Austr	AustrP	Bicyc2	Class	ClassE	Compu	Crusa	CrusaP	Djemb	Djembl	Hoops	Livgrm	Nkuba	Plants	Stairs	
				MP: 5.8 nd: 290 im0 im1 GT nonocc	MP: 5.6 nd: 290 im0 im1 GT nonocc	MP: 5.6 nd: 250 im0 im1 GT nonocc	MP: 5.7 nd: 610 im0 im1 GT nonocc	MP: 5.7 nd: 610 im0 im1 GT nonocc	MP: 1.5 nd: 256 im0 im1 GT nonocc	MP: 5.5 nd: 800 im0 im1 GT nonocc	MP: 5.5 nd: 800 im0 im1 GT nonocc	MP: 5.7 nd: 320 im0 im1 GT nonocc	MP: 5.7 nd: 320 im0 im1 GT nonocc	MP: 5.7 nd: 410 im0 im1 GT nonocc	MP: 5.9 nd: 320 im0 im1 GT nonocc	MP: 5.5 nd: 570 im0 im1 GT nonocc	MP: 5.6 nd: 320 im0 im1 GT nonocc	MP: 5.2 nd: 450 im0 im1 GT nonocc	
↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑	↓↑
08/28/15	<input checked="" type="checkbox"/> MC-CNN-act	H	8.08	5.59 20	4.55 25	5.96 17	2.83 10	11.4 25	5.81 14	8.32 23	8.89 27	2.71 15	16.3 23	14.1 18	13.2 18	13.0 5	6.40 16	11.1 15	
07/28/14	<input checked="" type="checkbox"/> SGM	H	18.42	40.3 79	4.54 23	8.03 32	22.9 67	40.5 59	11.4 39	24.7 51	10.1 38	5.40 45	29.6 45	28.5 51	23.9 55	20.0 40	14.2 40	30.9 51	

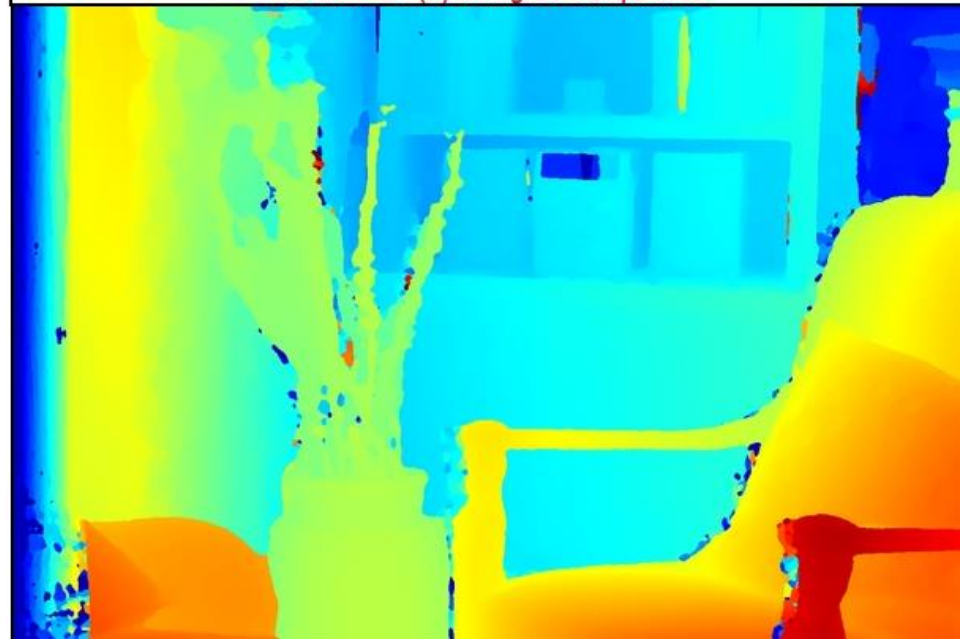


# Example results

SGM (H) Livingroom disparities

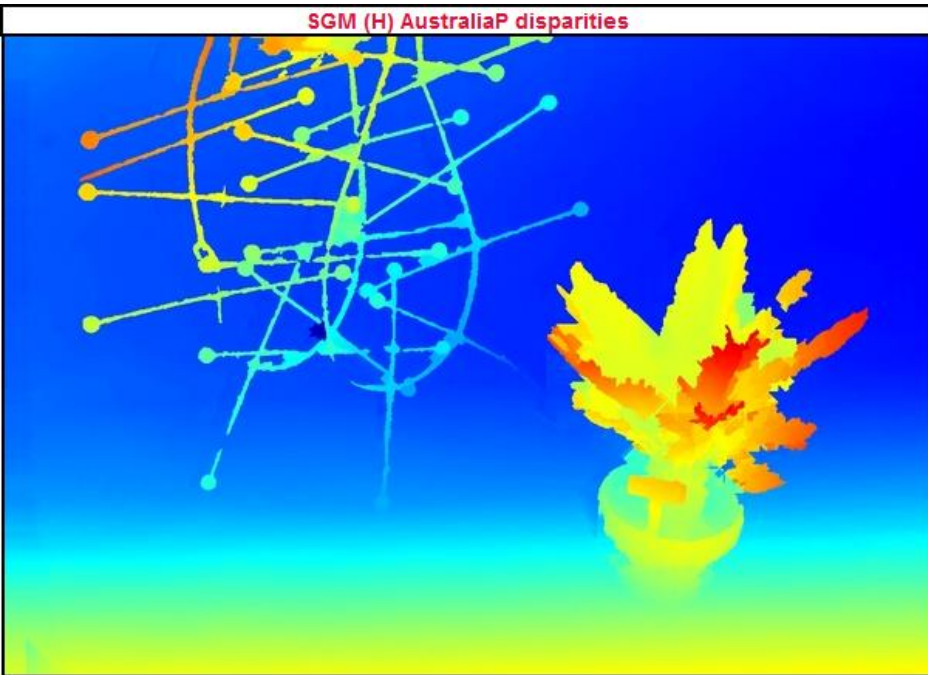


MC-CNN-acrt (H) Livingroom disparities

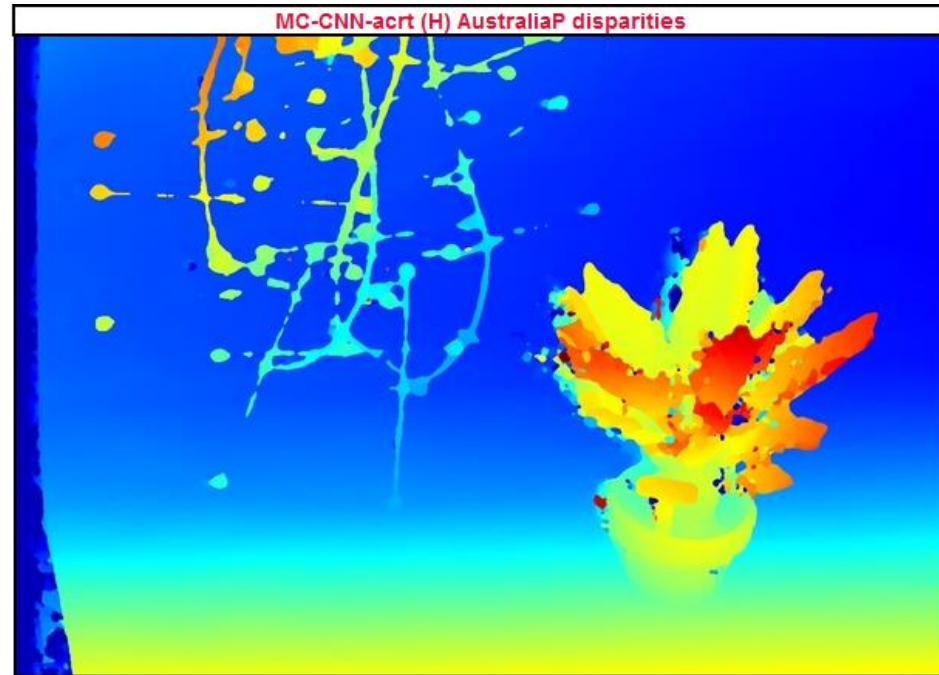


# Example results

SGM (H) AustraliaP disparities



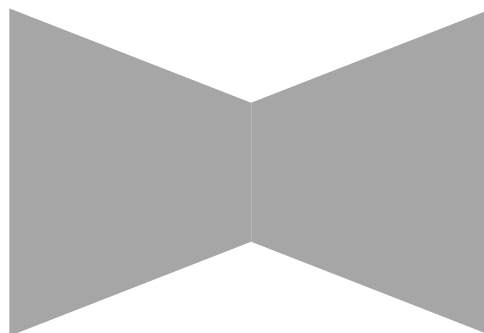
MC-CNN-acrt (H) AustraliaP disparities



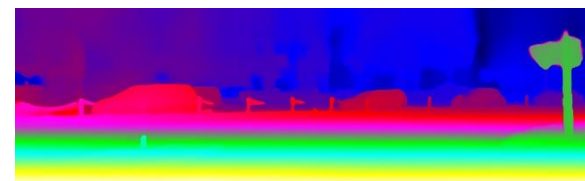
# Monocular depth estimation



single image

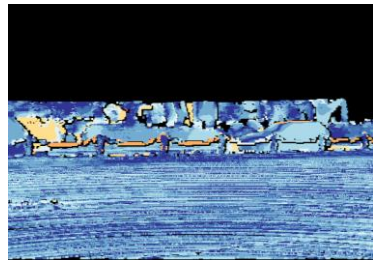


CNN

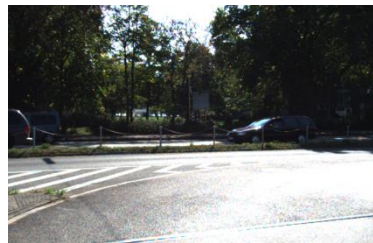


disparity image

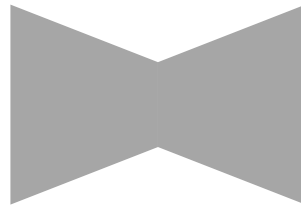
# Monocular depth estimation - Training



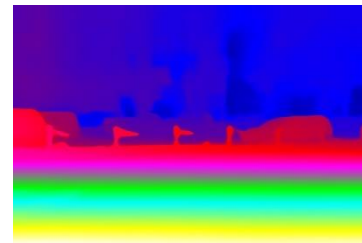
ground truth  
disparities



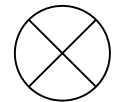
single image



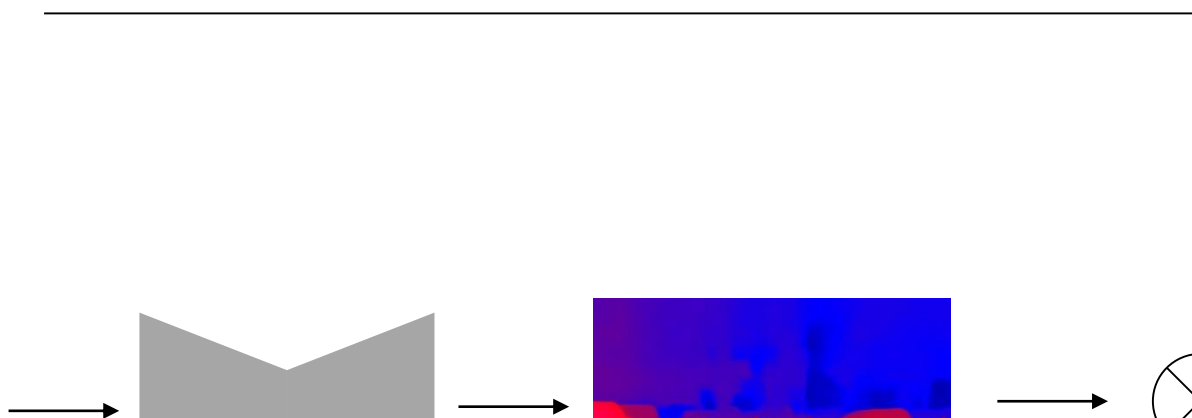
CNN



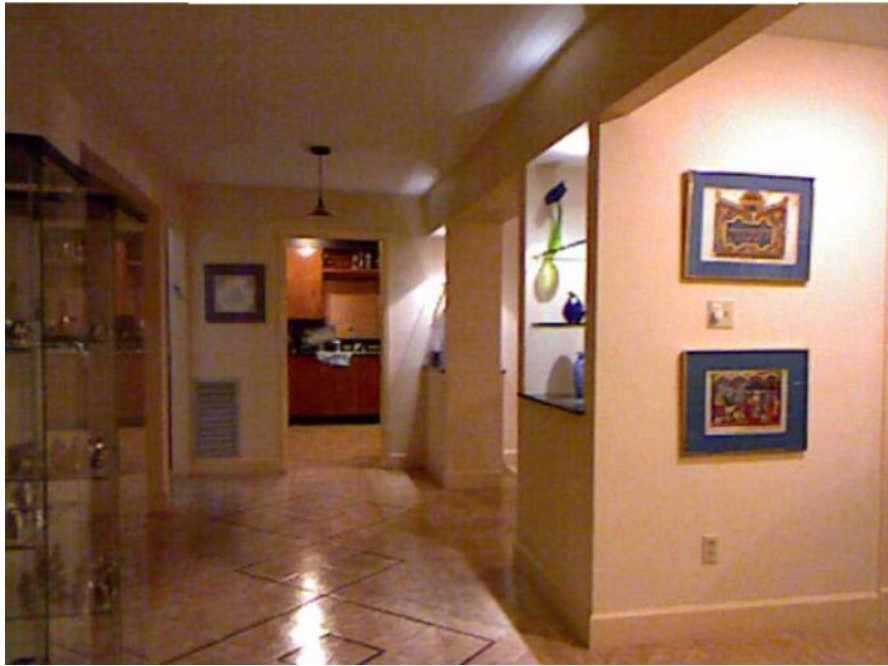
disparity image



cost function



# How well does it work?



# Planarity errors

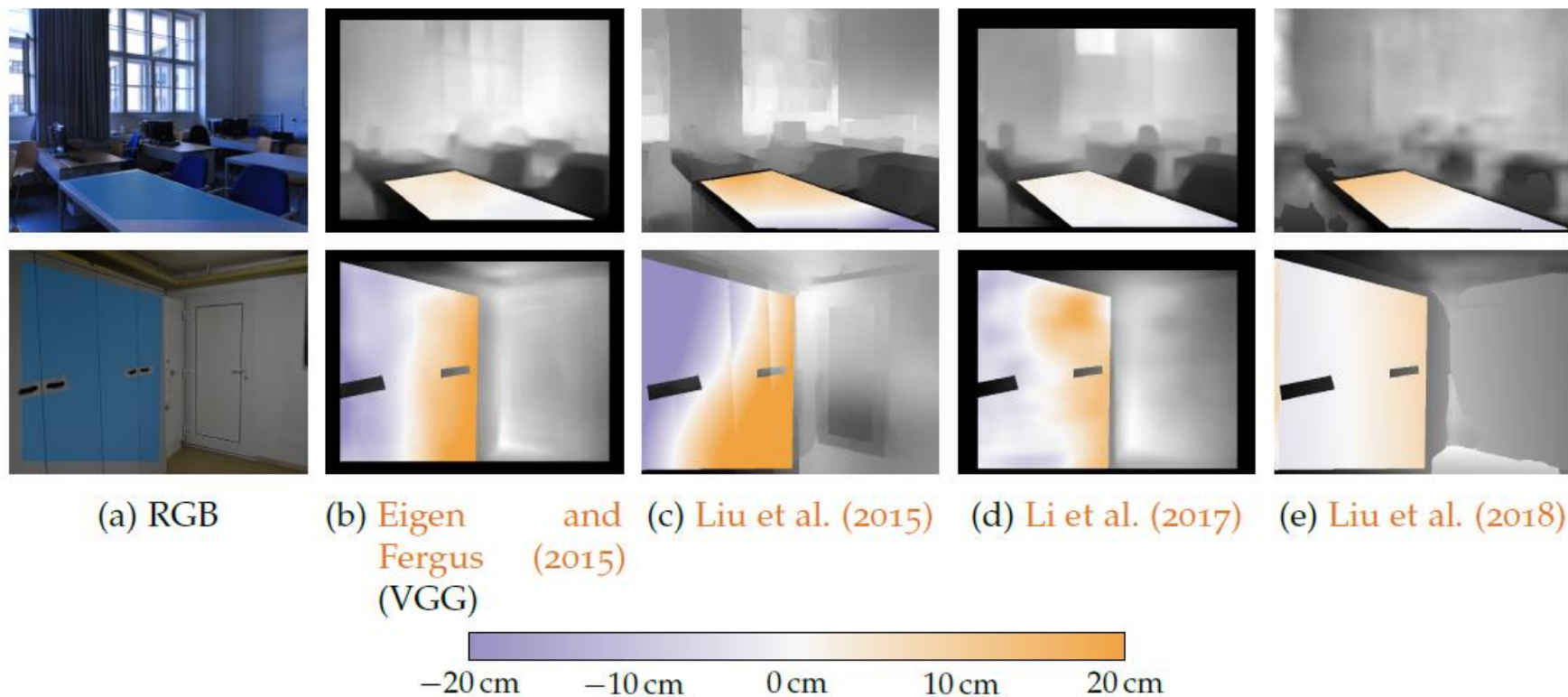


Figure 4.25: Visual results after applying *planarity errors* (PEs) on different planar regions (top: table, bottom: wall). RGB with corresponding plane masks (■) (a). Predictions using different methodologies (b-e). Colors in the predictions correspond to orthogonal differences of projected depths towards the reference plane

# Limits of current method

- Network estimates depth for a picture on a flat wall
- NO absolute scale measurements as in real stereo setup!

