Robot Vision:
Stereo Matching

Ass.Prof. Friedrich Fraundorfer

SS 2019

Outline

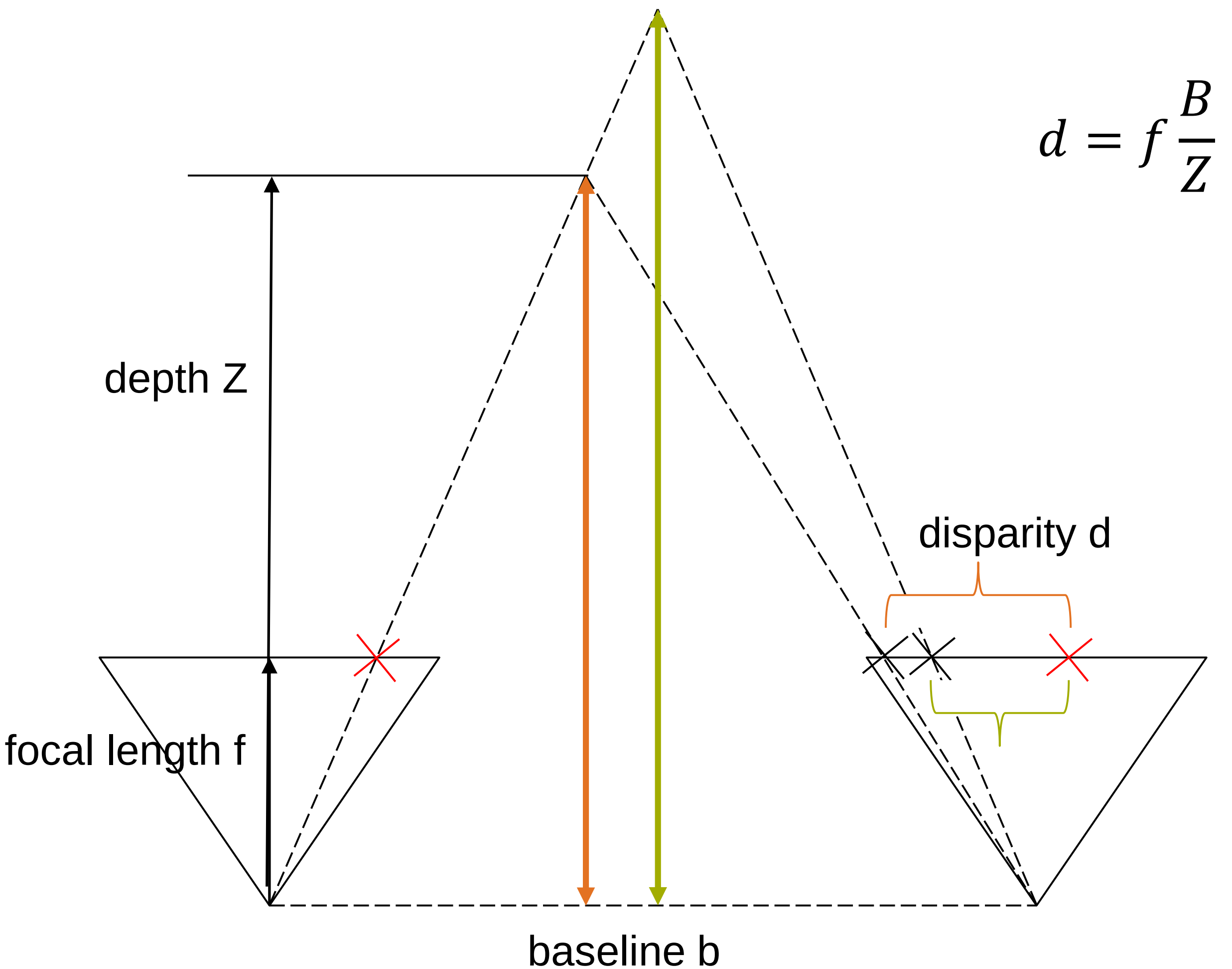
- Geometric relations for stereo matching
- Dense matching process
- Census Transform
- Dynamic programming
- Semiglobal matching

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

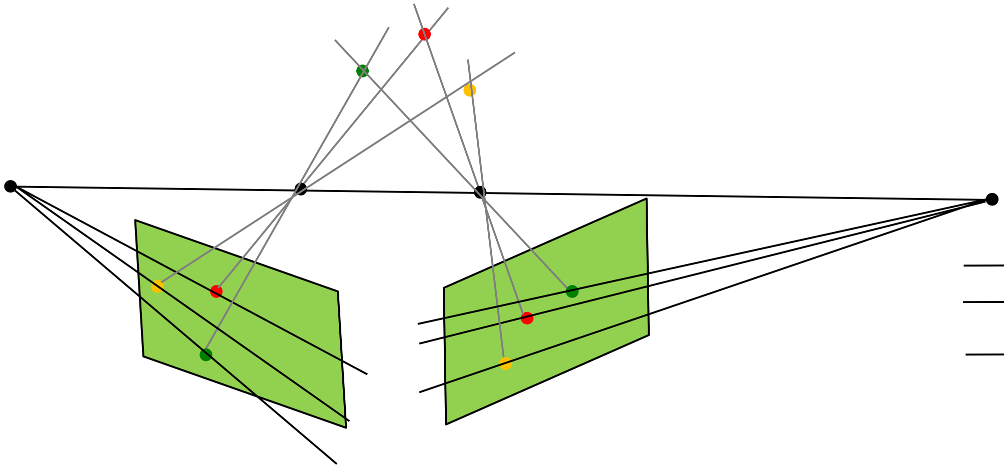
Geometric relation

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

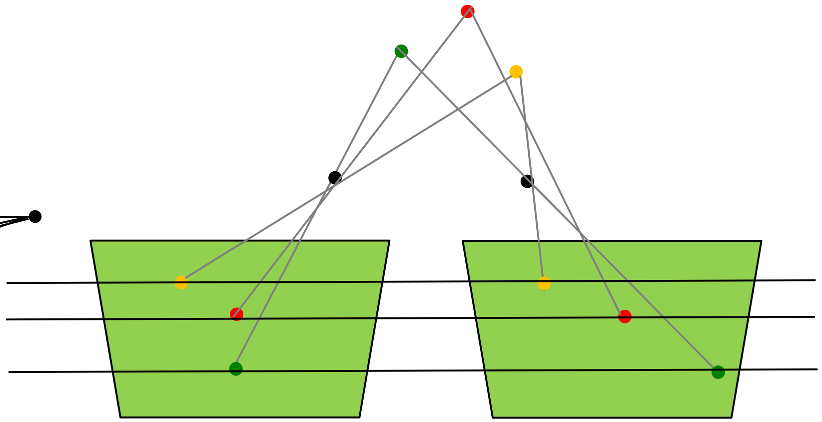


Rectification

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



Before rectification



„Stereo normal case“

Dense matching process

Left View

I_l

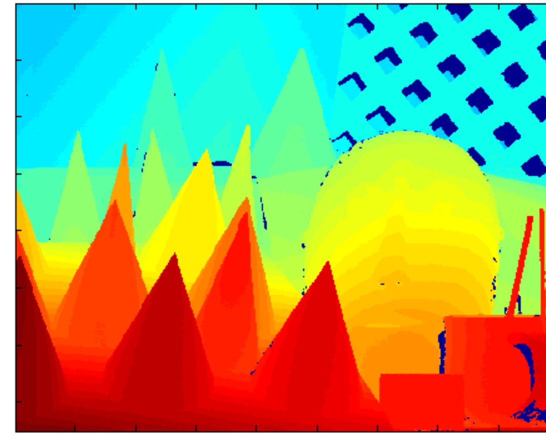
+

Right View

I_r

=

*



Disparity image

D

- Estimate disparity (depth) for all pixels in image left.
  - Evaluate correspondence 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.

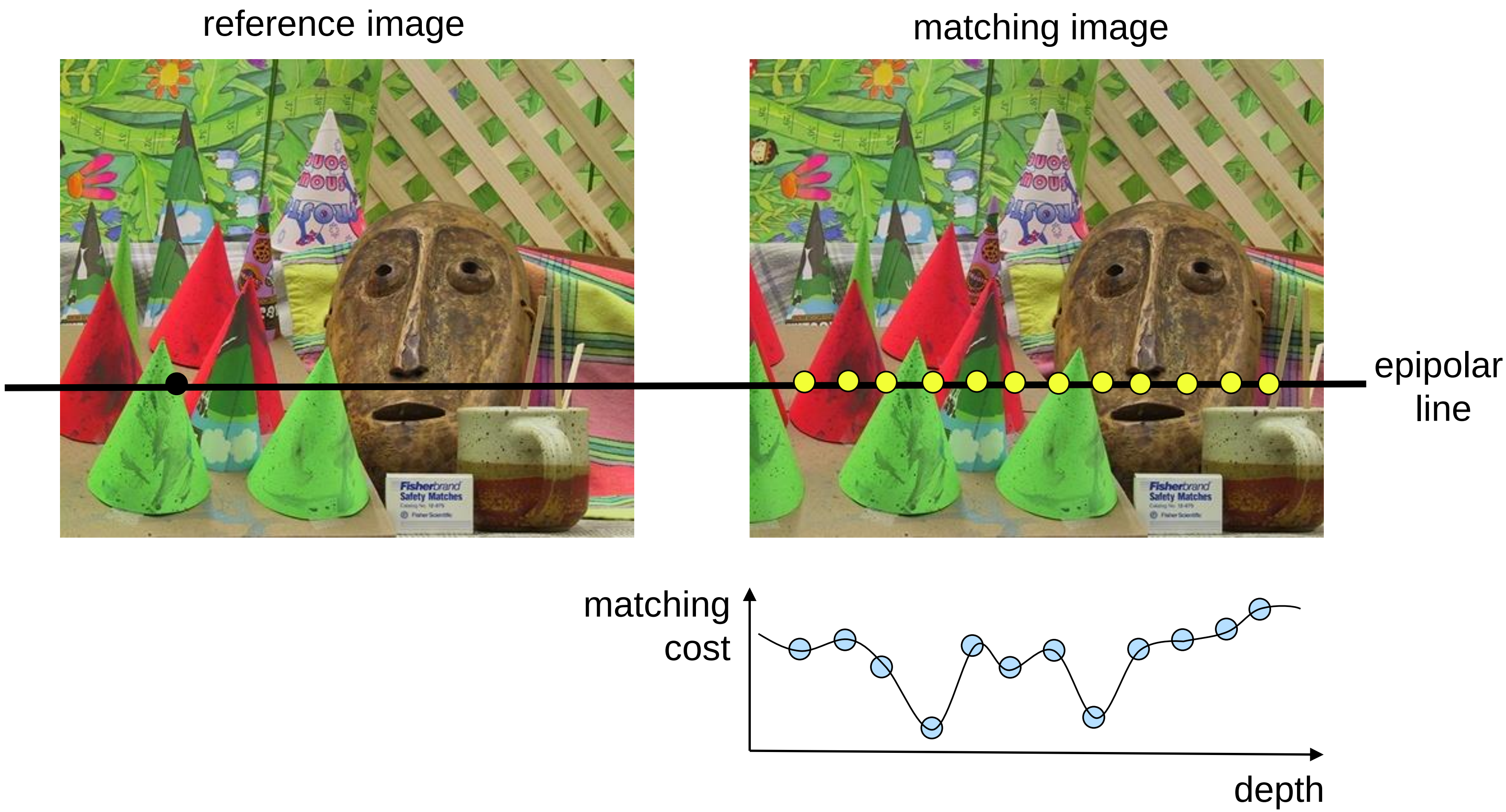
Census Transform

- 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)

89	63	72
67	**55**	64
58	51	49

00000011

Dense matching process



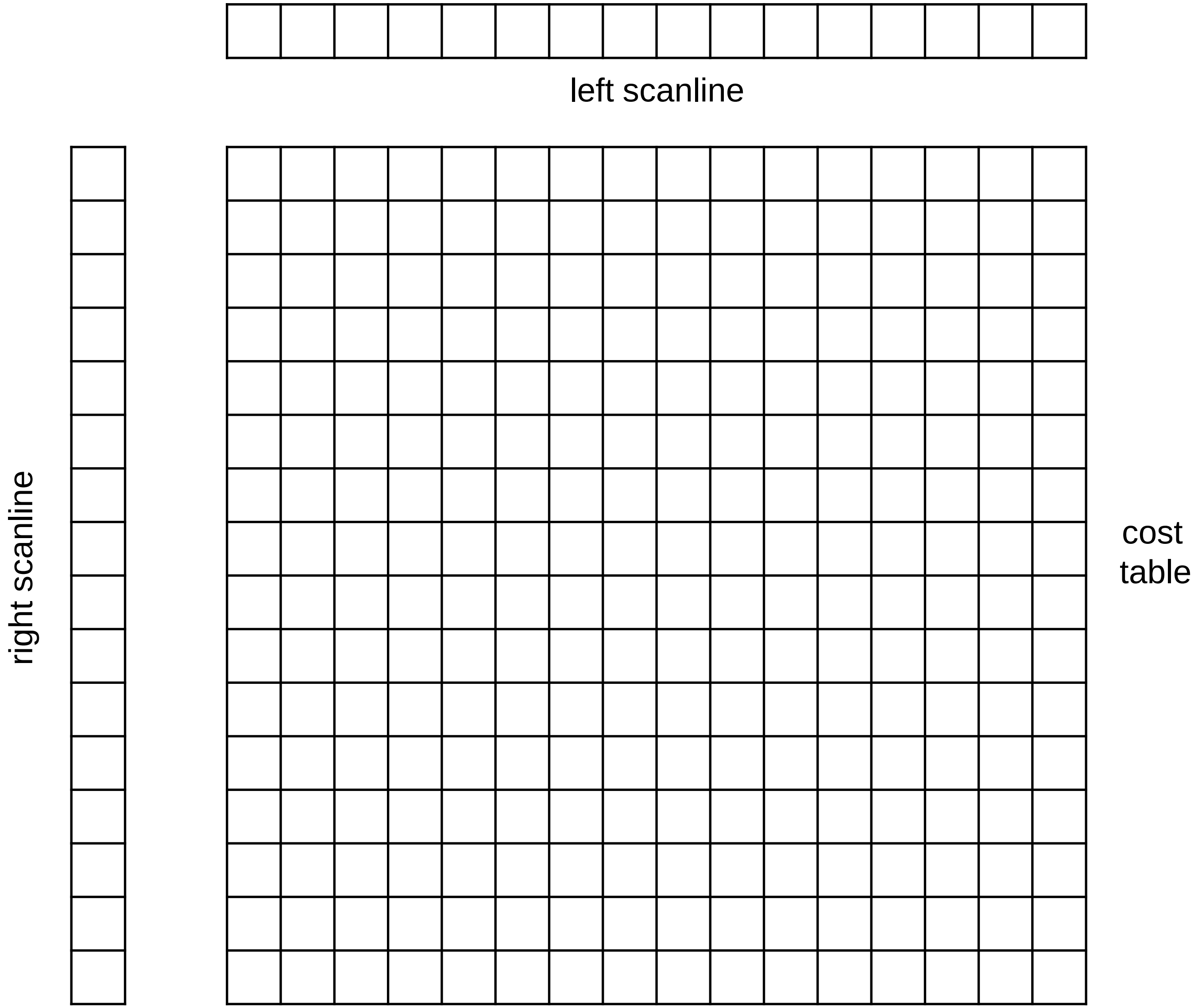
Disparity selection

- 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

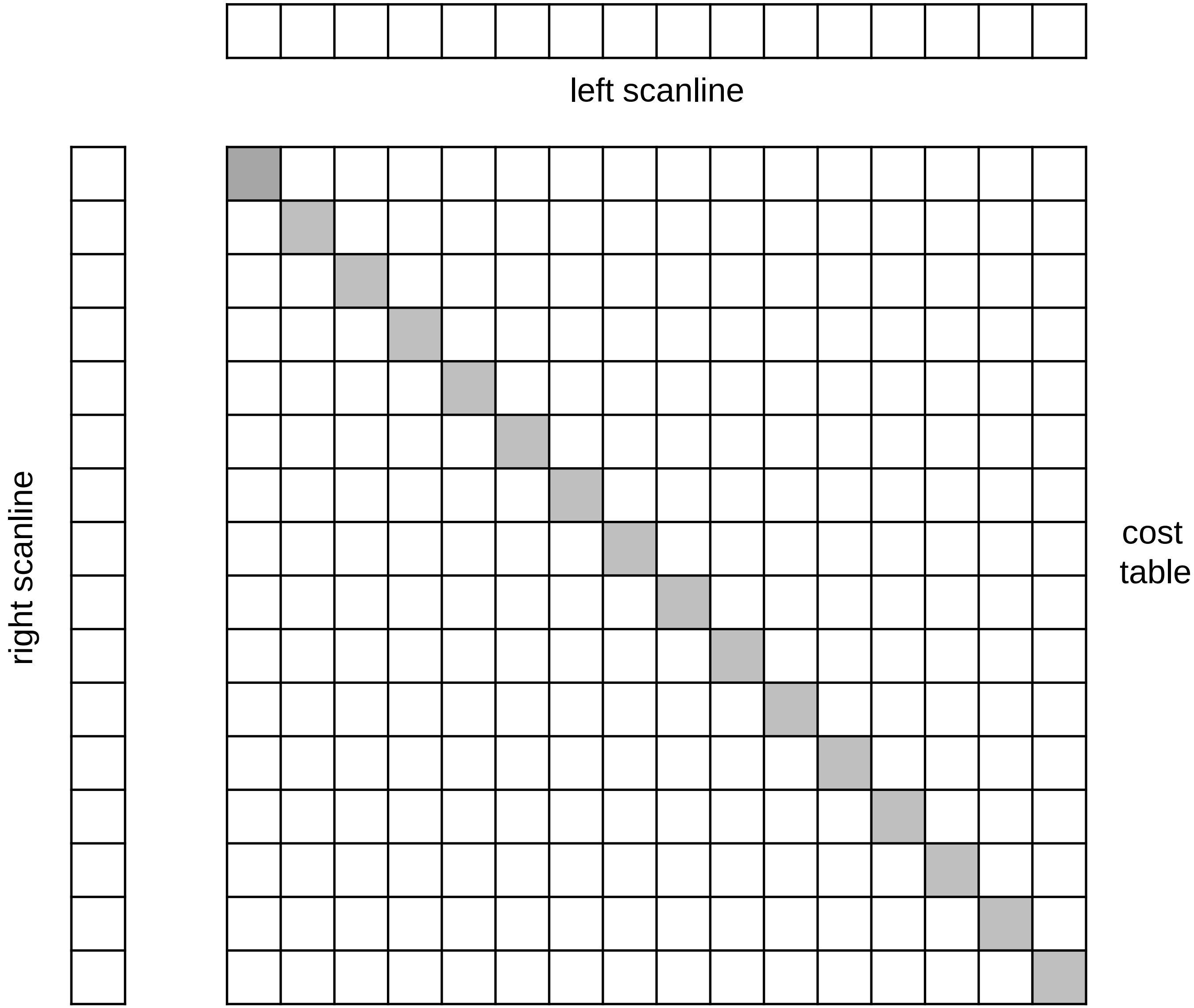
Scanline optimization

- 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

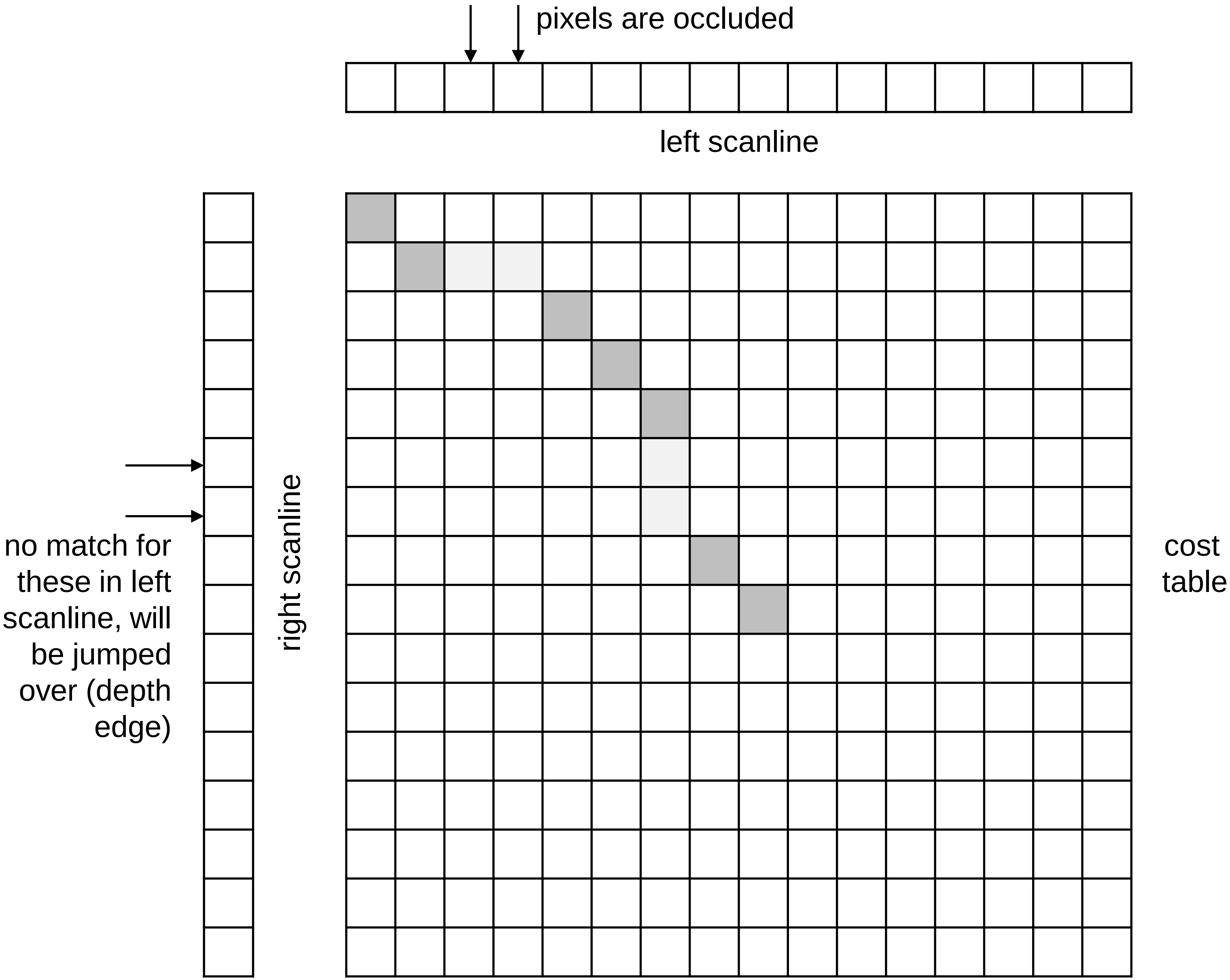
Scanline optimization



Scanline optimization



Scanline optimization

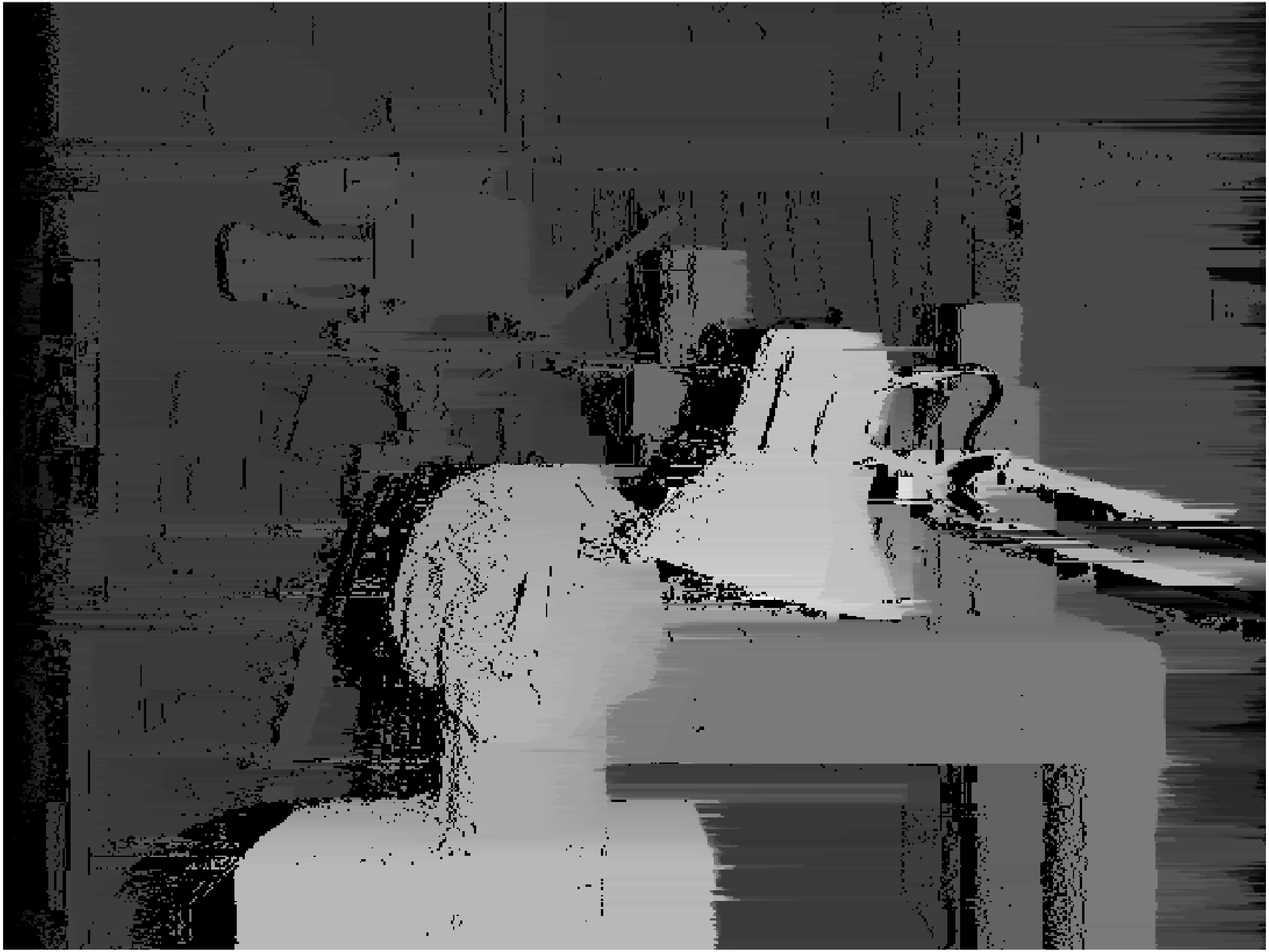


Scanline optimization complexity

- 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

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

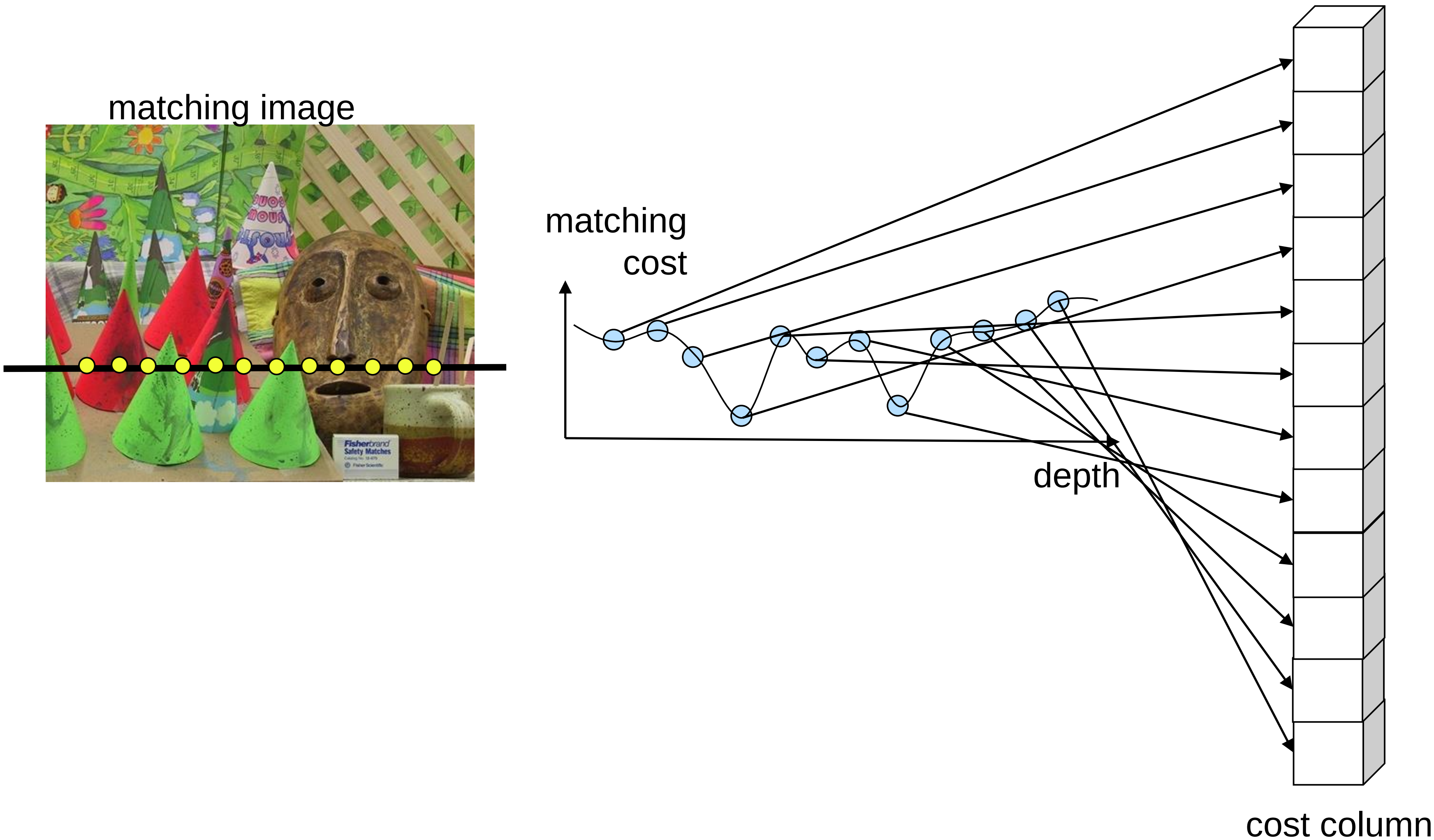
  - Data term:
    Agreement between cost function and input image pair

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

  - Smooth term:
    Encoding the smoothness assumptions

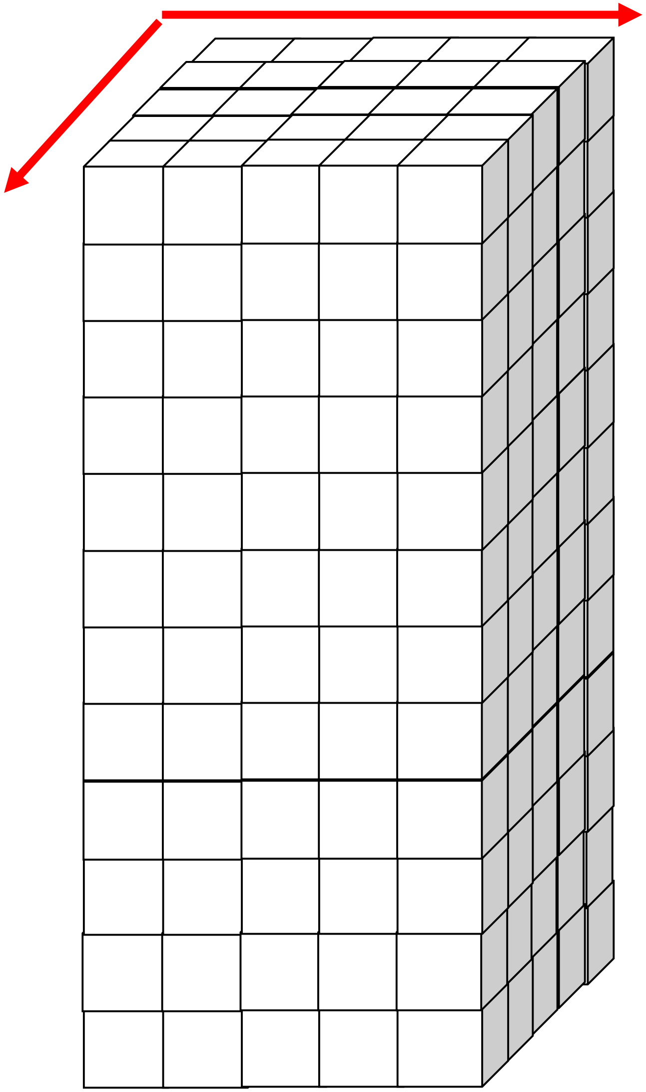
$$E_{smooth}(D) = \sum_{(p)} \rho(d(u,v) - d(u+1,v))$$

Cost volume

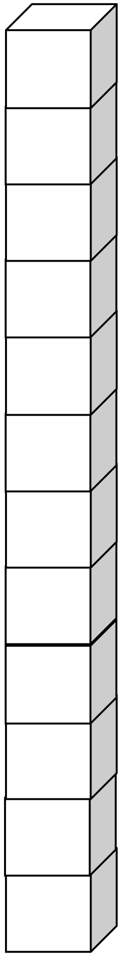


Cost volume

matching image



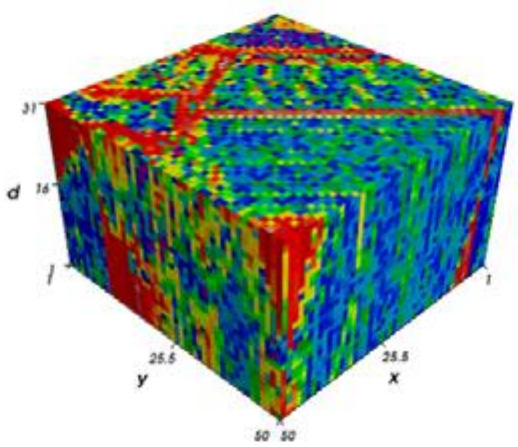
cost volume



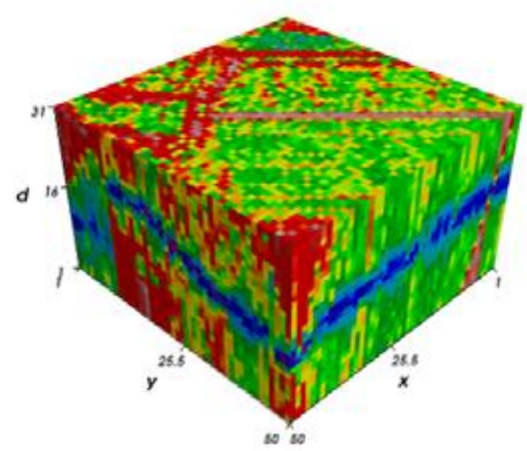
cost column

Semiglobal matching

- Cost Aggregation (Cost Optimization)



Cost Cube



Optimized Cost Cube

Goal: global minimization of

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

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

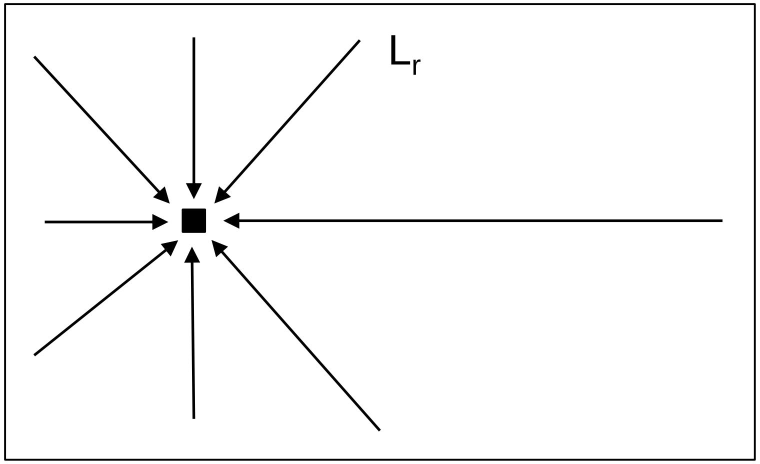
- Path-wise approximation of aggregation

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

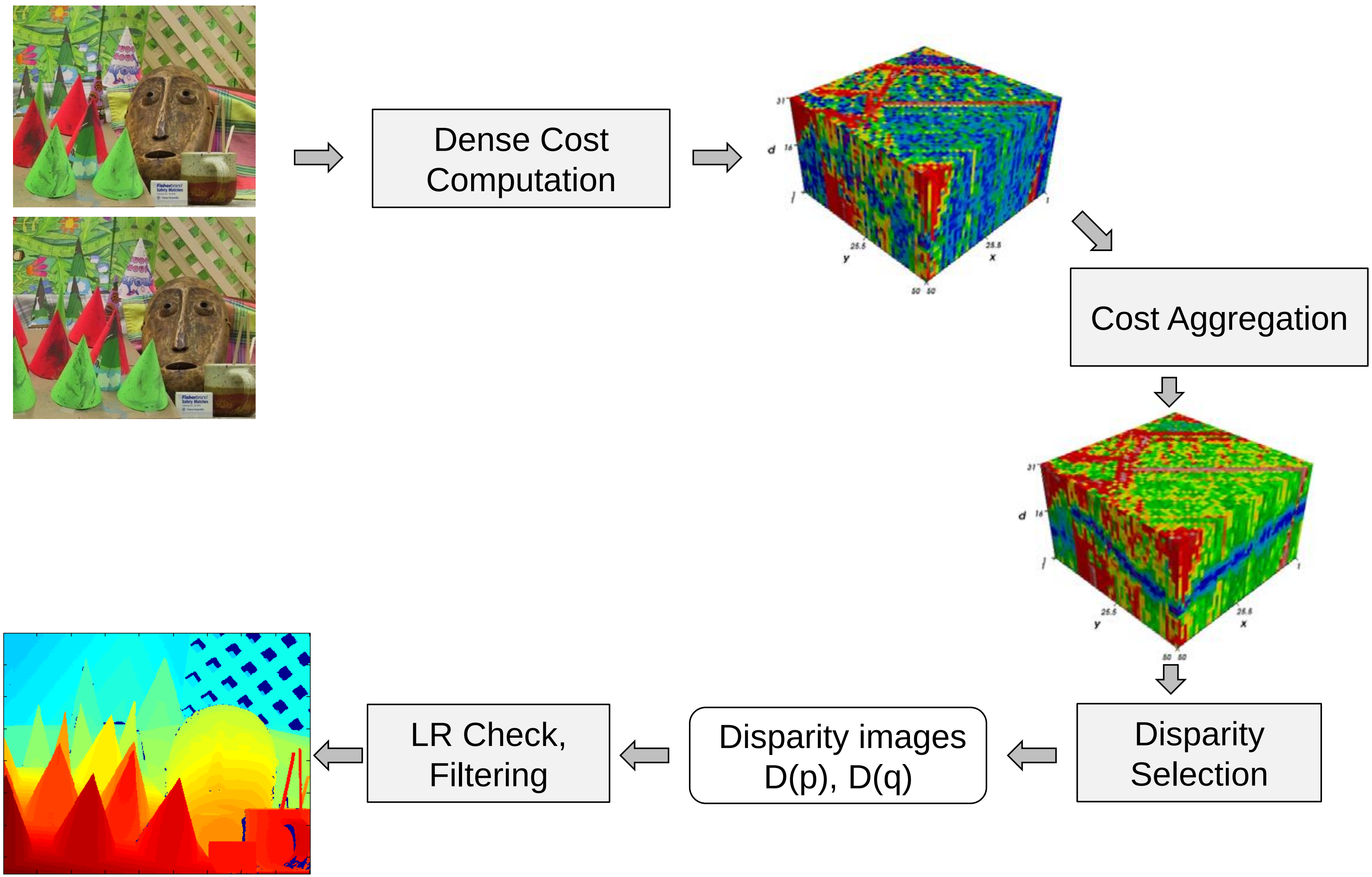
p	Image coordinates
P_1	Cost for small height jump
P_2	Cost for large height jump
r	Path direction
L_r	Aggregated costs along r
d	Disparity

- Summation of *L* along 8 or 16 directions *r*

$$S(p,d) = \sum_r L_r(p,d)$$



Semiglobal matching



[Heiko Hirschmüller (2008), Stereo Processing by Semi-Global Matching and Mutual Information, in IEEE PAMI, Volume 30(2), February 2008, pp. 328-341.]