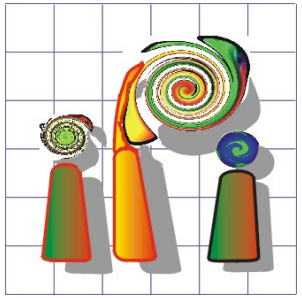
Current status of image matching for Earth observation



Christian Heipke

IPI - Institute for Photogrammetry and GeoInformation

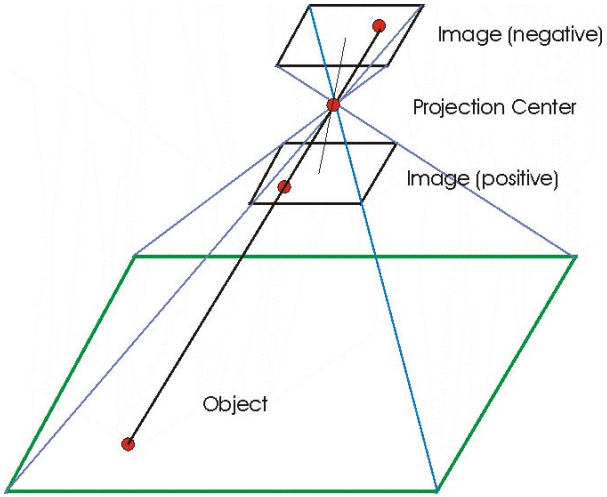
Leibniz Universität Hannover

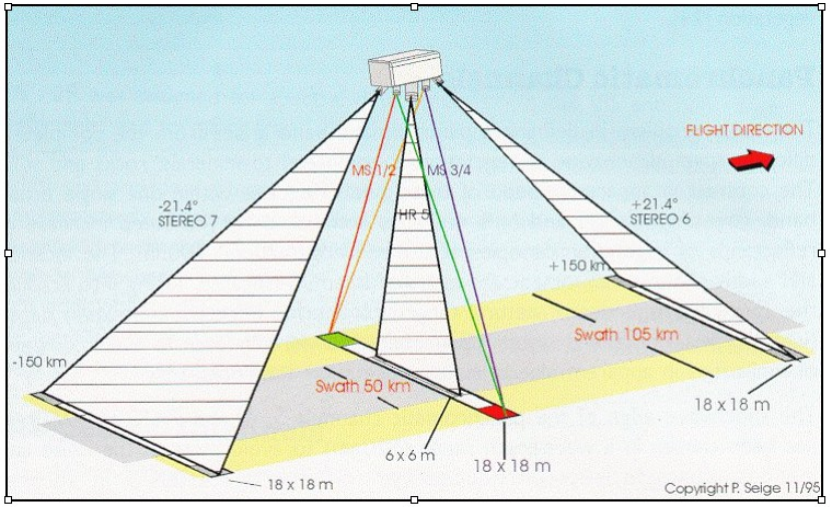
Secretary General, ISPRS

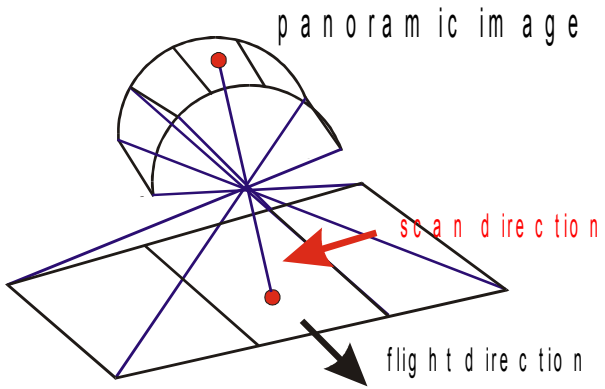
Content

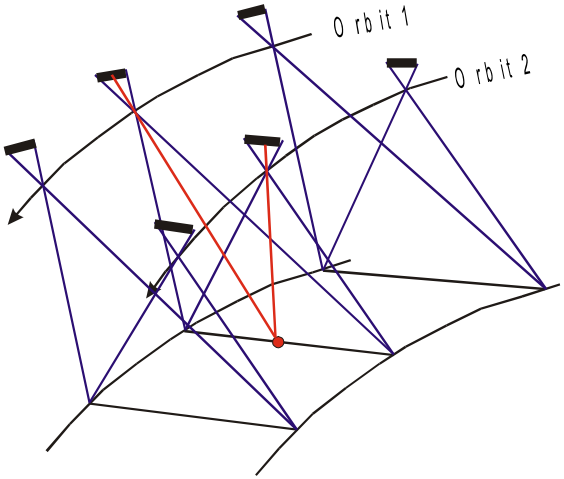
- Introduction
- Dense matching
- Dynamic programming and semi-global matching
- Examples
- Conclusions

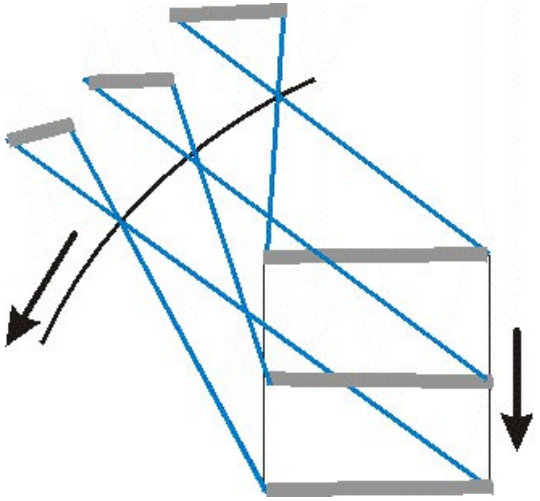
Satellite sensor geometry











Window Based Matching

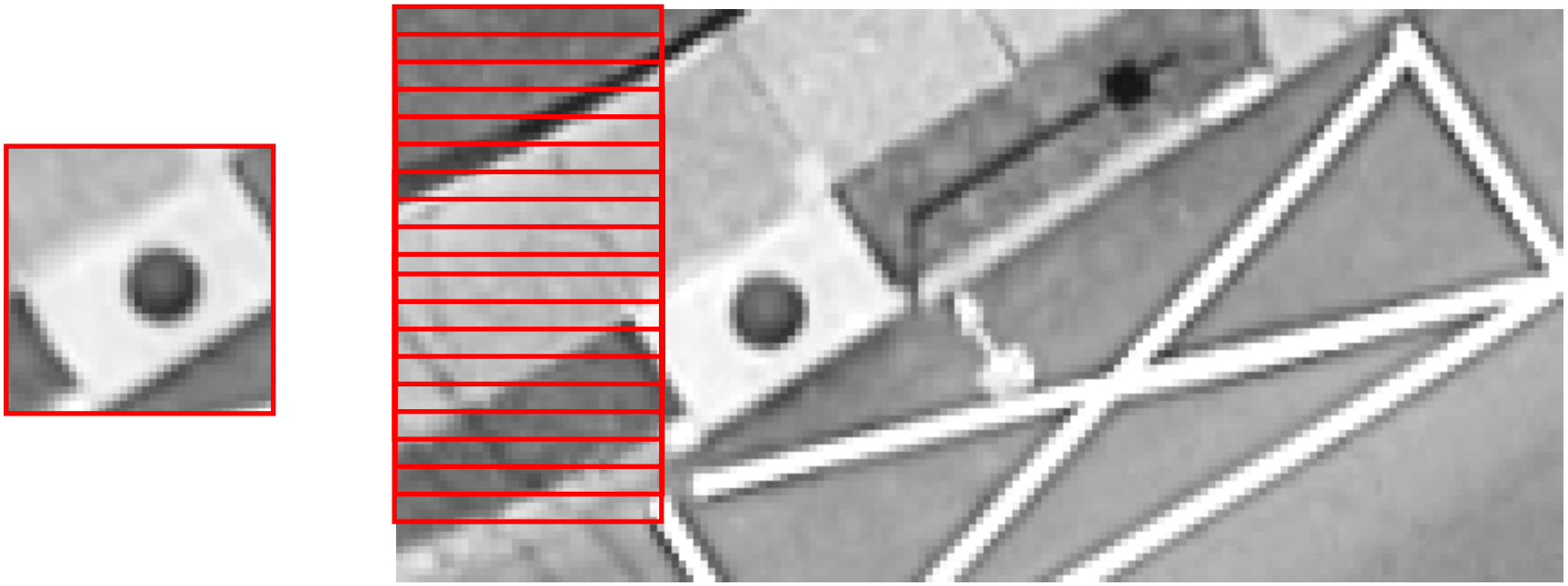
- Image matching via cross correlatin:
  - Template matrix around candidate point in reference image
  - Search matrix around approximate position in right image

Reference image

Partner image

Template and Search Matrix

➔ Template matrix is shifted across search matrix pixel per pixel

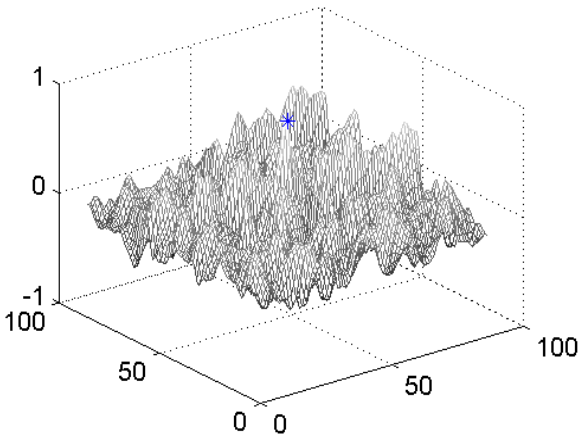


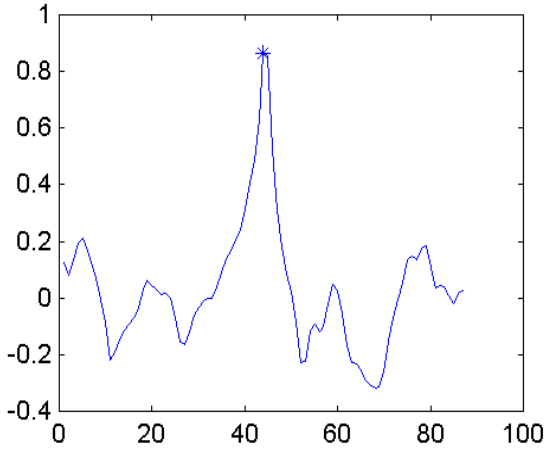
Template matrix
37×37 Pixel

Search matrix
161×79 Pixel

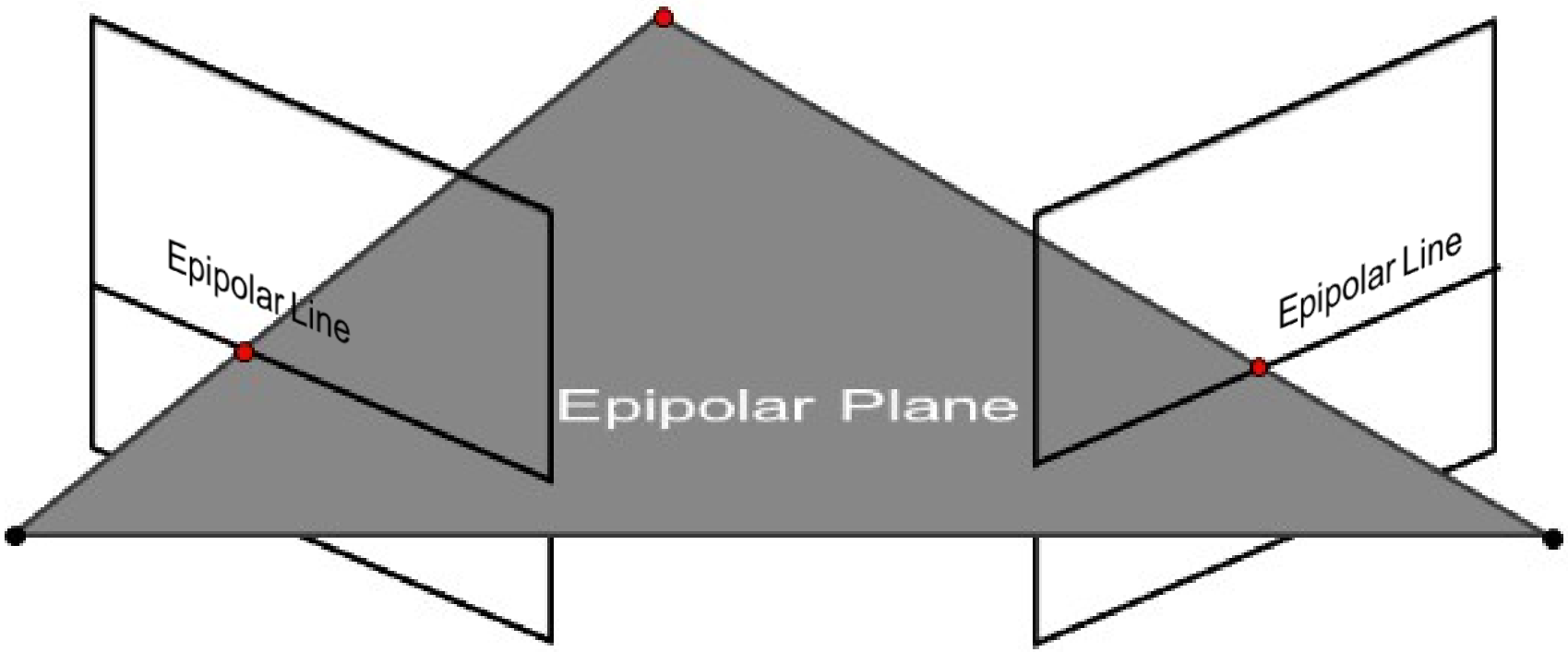
Selection of the Corresponding Point

- Search for the maximum in the correlation function (“Winner takes all strategy”)





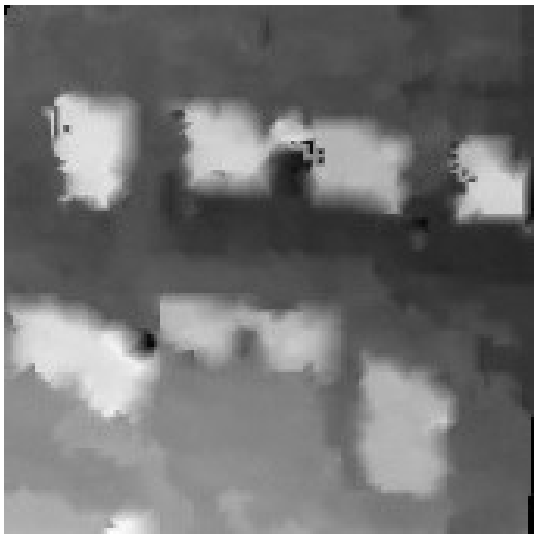
Epipolar Geometry and 1D matching

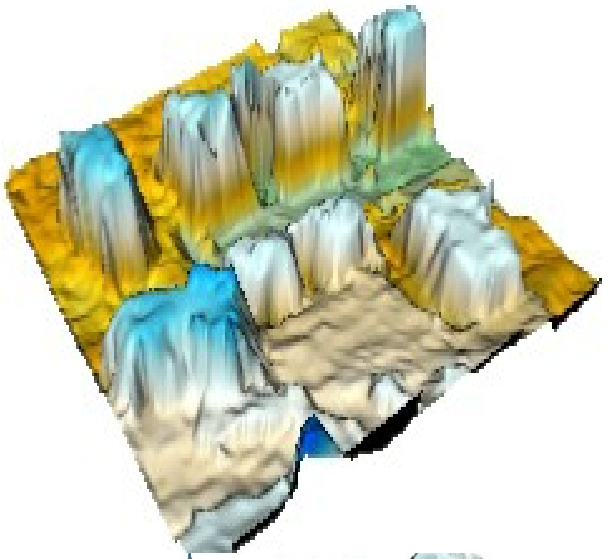


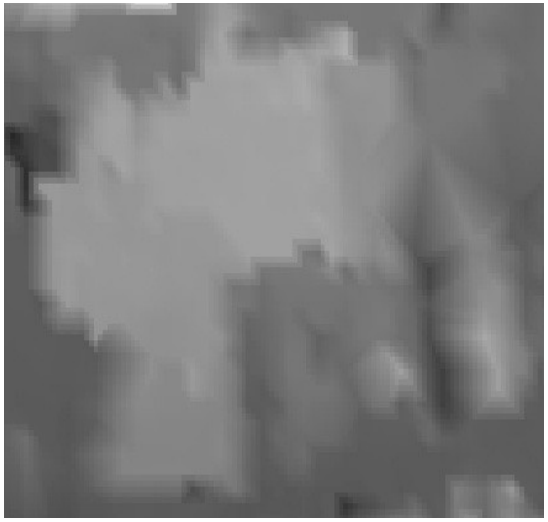
Correlation at Depth Discontinuities

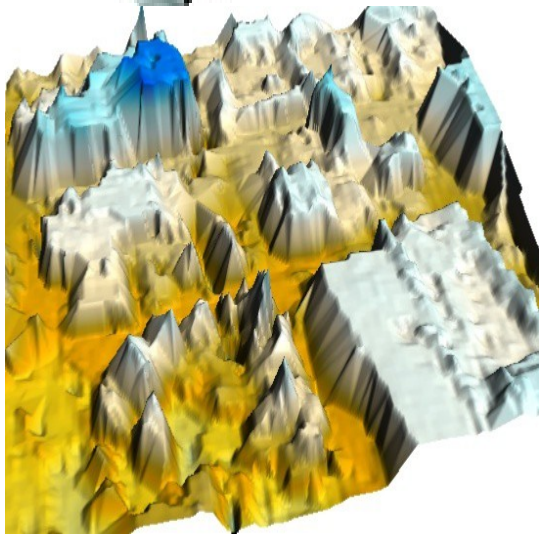
➔ Depth discontinuity in the window results in errors

Example result









Dense matching

- Match individual pixels (will result in very ambiguous matches and noisy results)
- Introduce constraints to make sure that surface changes occur smoothly ("smoothness constraint")
- Smoothness constraints cover a large part of the image (reason for "global" in the name)
- Use energy minimisation to find optimum solution:

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

D disparity map (disparity = parallax, i.e. d = p = x'' – x')

$E_{data}(D)$ sum of local matching costs

$E_{smooth}(D)$ encodes smoothness assumptions measuring differences between disparities (parallaxes) of neighboring pixels

Data term

- Measures the pixel-wise colour dissimilarity for each pixel *p* of the left image *I*:

$$E_{data}(D) = \sum_{p \in I} m(p, p - d_p)$$

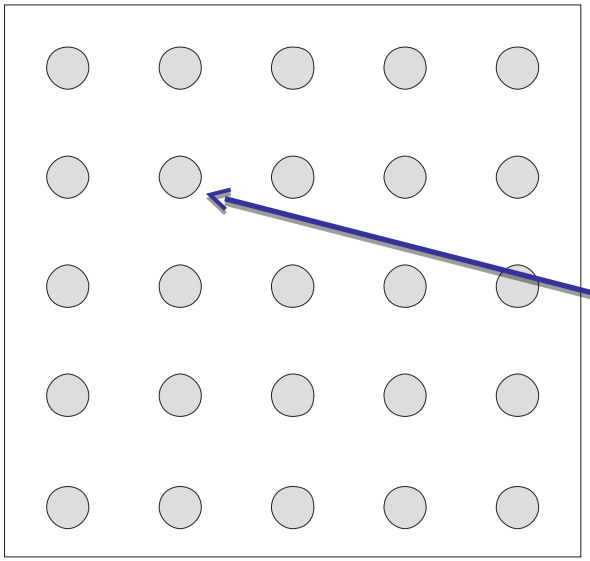
where

  - d_p is the disparity of *p* in the disparity map *D*
  - *m*() is a function computing the colour dissimilarity between pixels of left and right images

    e. g. cross correlation, sum of squared differences (SSD), sum of absolue differences (SAD), mutual information (MI)

Smoothness term in a graph

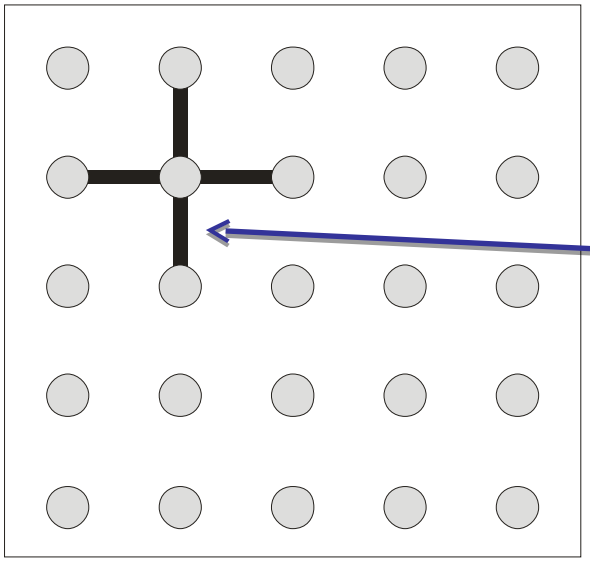
- The smoothness assumption states that neighbouring pixels should be assigned similar disparities.
- *G = {V,E}*, Graph with set of vertices (nodes) *V* and edges (links, arcs) *E*



Nodes correspond to pixels of the left image.

Smoothness term in a graph

- The smoothness assumption states that neighbouring pixels should be assigned to the same (or similar) disparities.
- *G = {V,E}*, Graph with set of vertices (nodes) *V* and edges (links, arcs) *E*

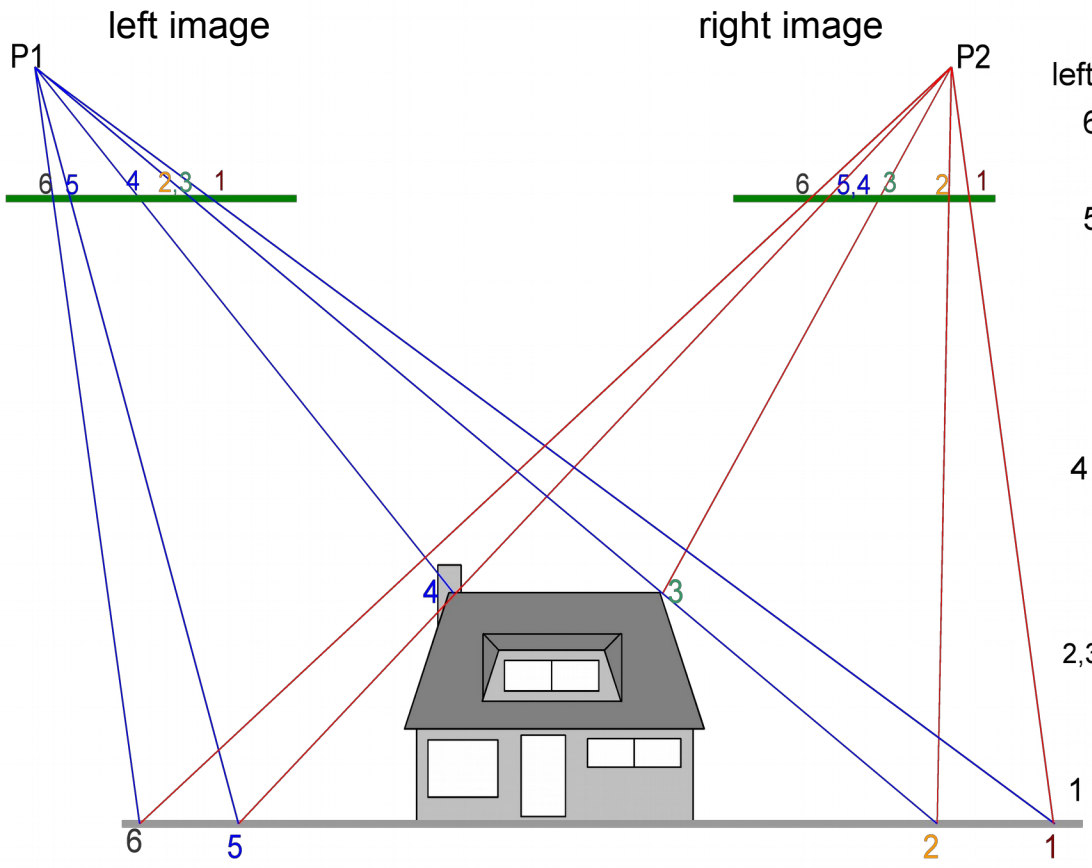


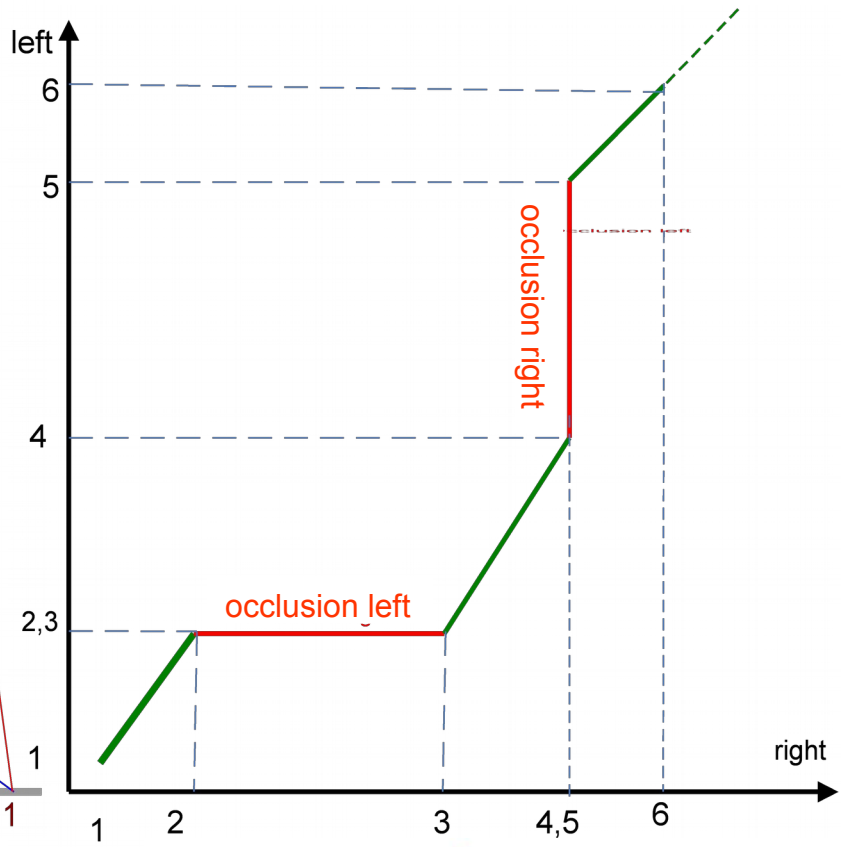
Edges represent interactions between pixels.

Often interactions occur between a pixel and its 4 spatial neighbours.

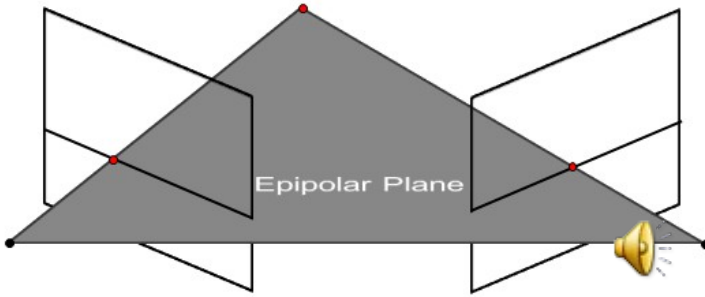
Pixel should have the same disparity as its neighbours.

Dynamic programming for match sequences

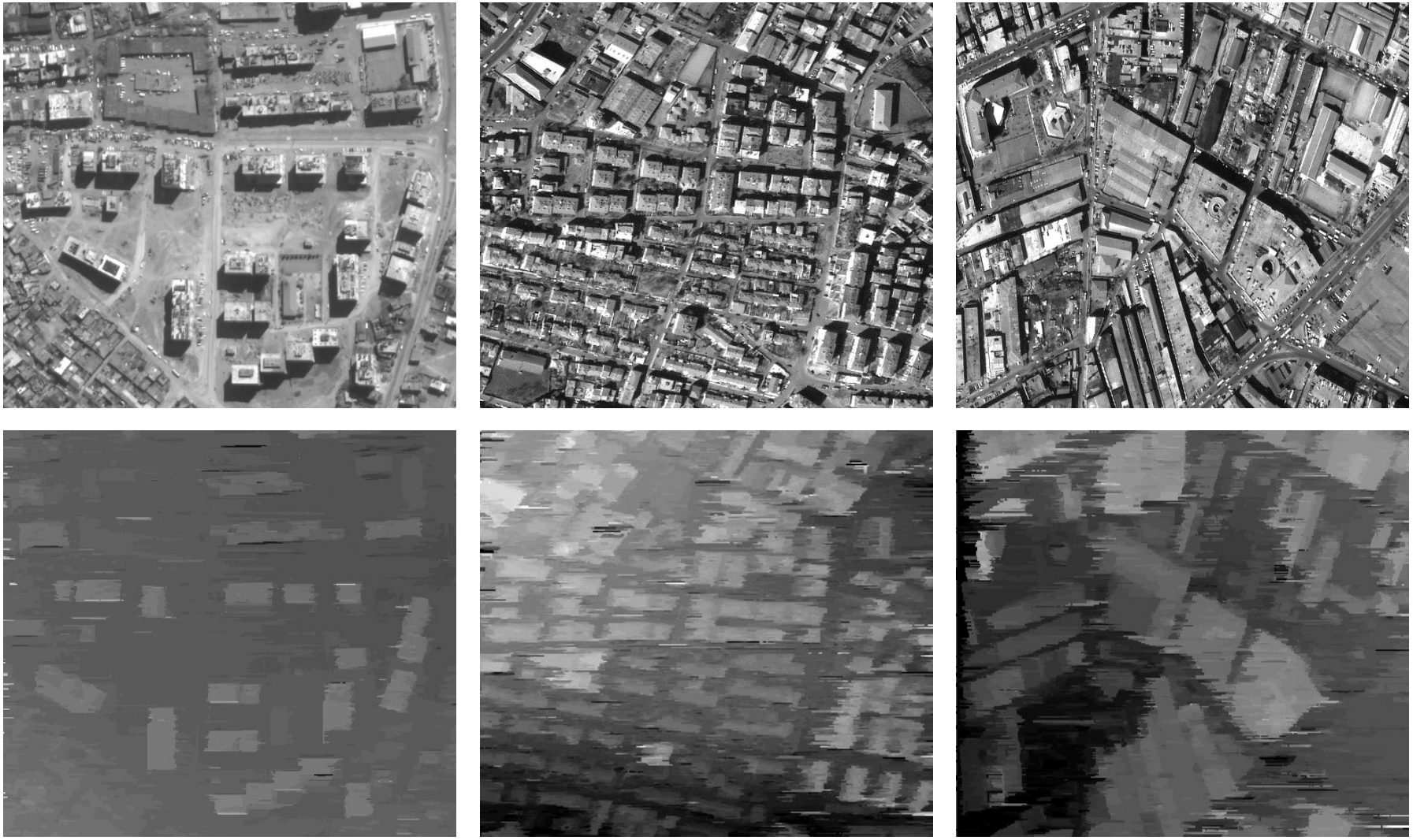




green parts of match sequence: matched pixel
red parts: occlusions



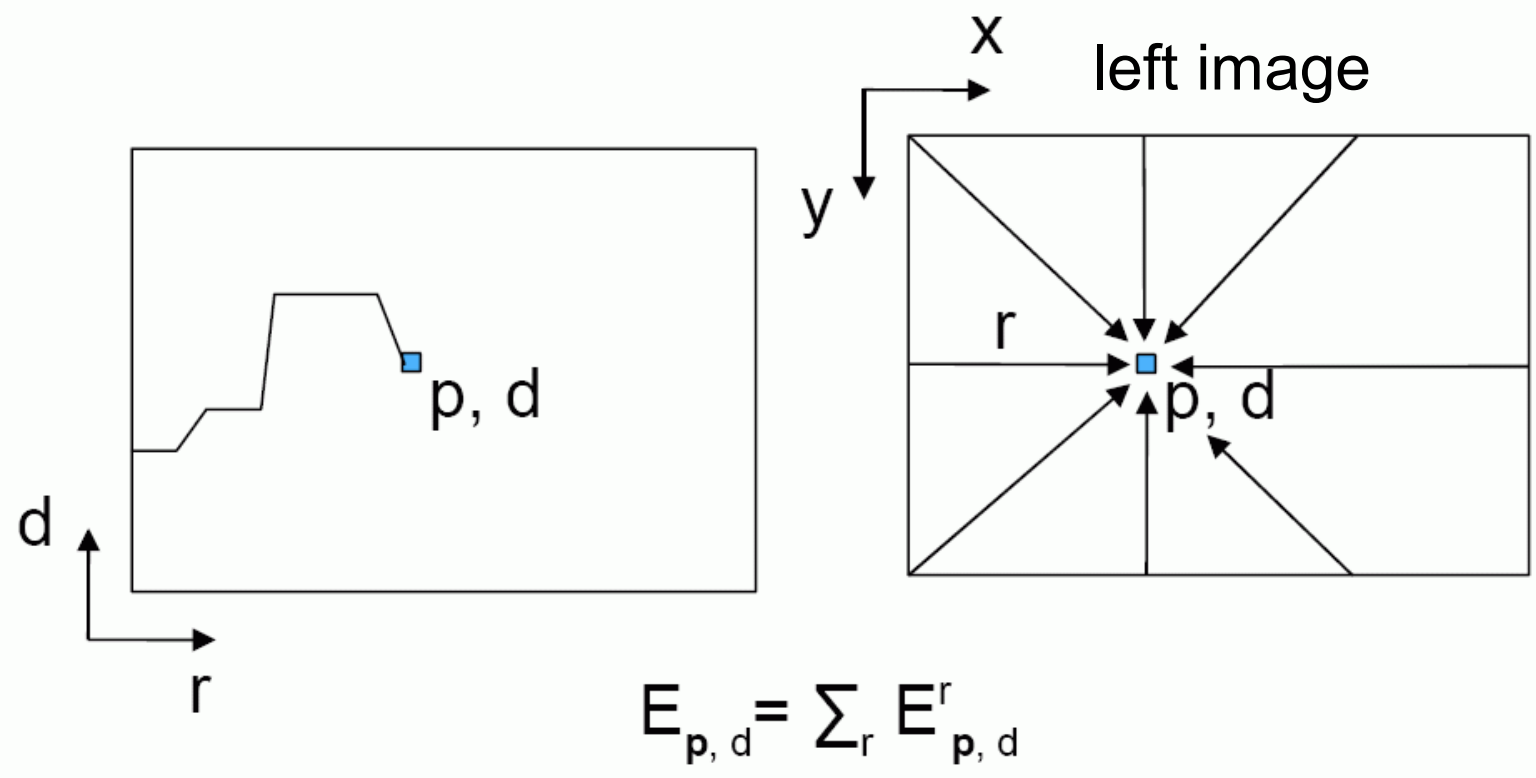
Dynamic programming (DP)



Upper: Ikonos images (Turkish test sites); lower: DSM before vertical median filtering

Semi Global Matching (SGM)

- 1D dynamic programming in different directions to overcome streaking effects



Semi Global Matching (SGM)

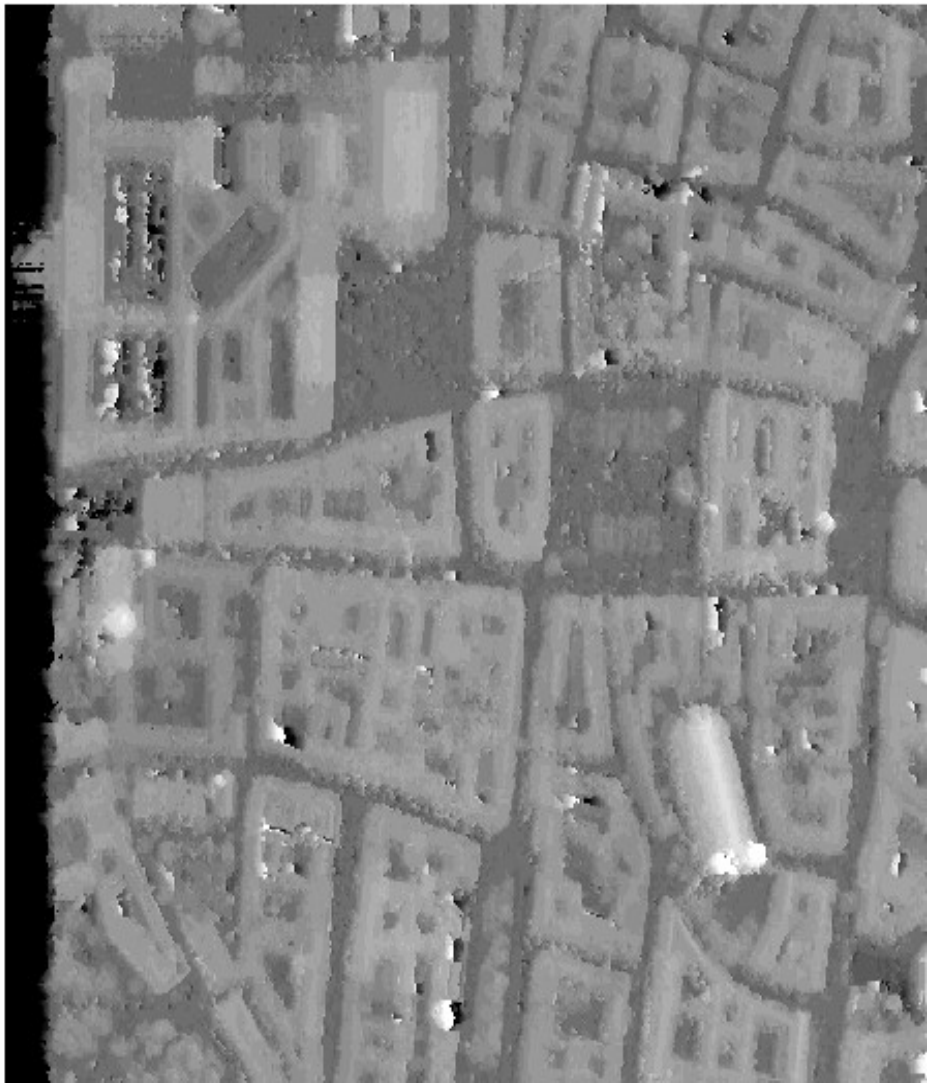
Hirschmüller, PAMI 2008

Cost Aggregation

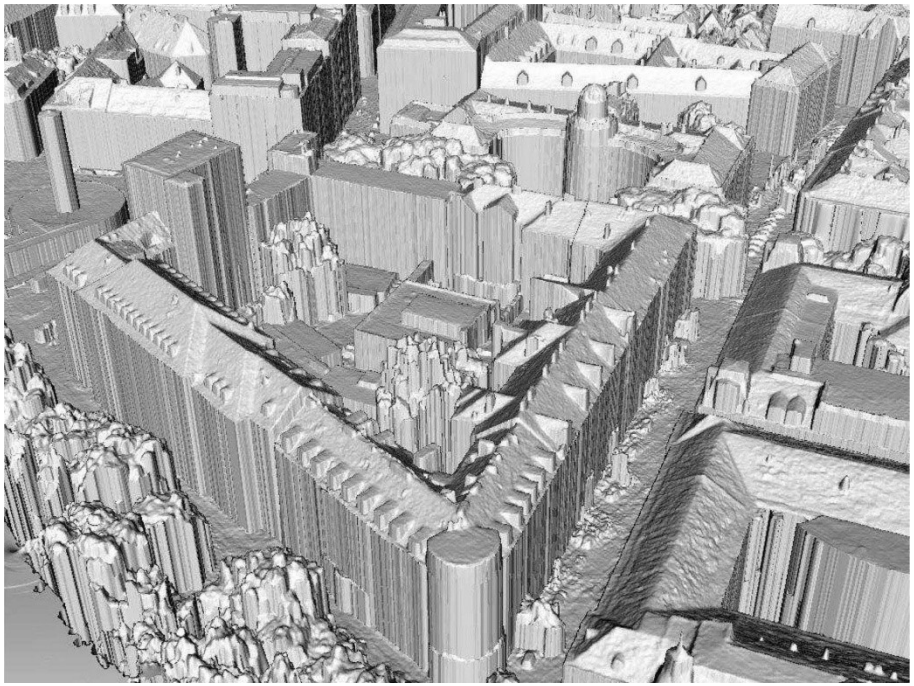
$$E(D) = \sum_p \left(C(p, D_p) + \sum_{q \in N_p} P_1 T\left[|D_p - D_q| = 1 \right] + \sum_{q \in N_p} P_2 T\left[|D_p - D_q| > 1 \right] \right)$$

- The first term is the sum of all pixel matching costs for the disparities of *D*.
- The second term adds a constant penalty P_1 for all pixels *q* in the neighborhood N_p of *p*, for which the disparity changes a little bit (that is, 1 pixel).
- The third term adds a larger constant penalty P_2, for all larger disparity changes.

SGM example München (Ikonos)



SGM example Graz

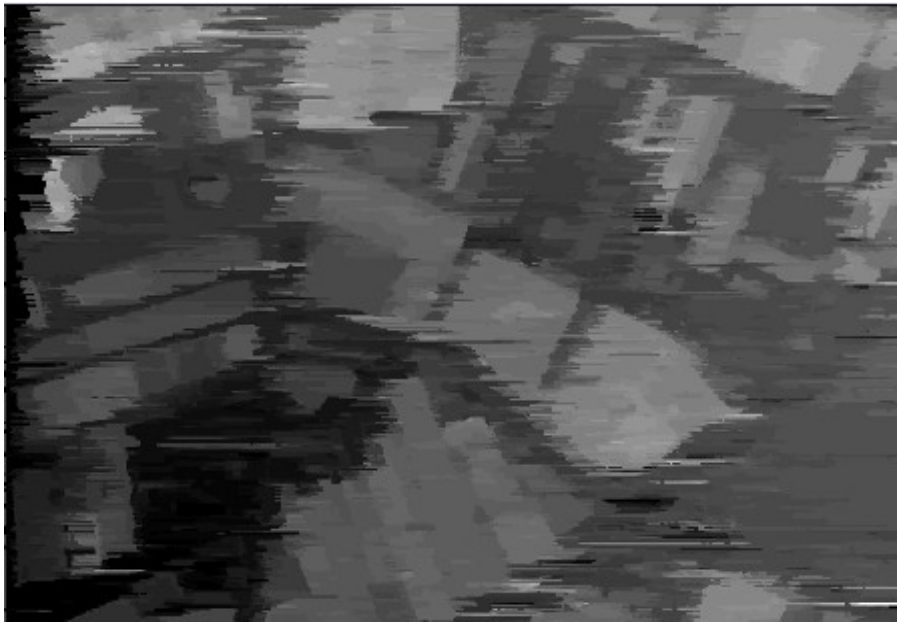


Untextured 3D reconstruction

Textured 3D reconstruction

Comparison DP/SGM

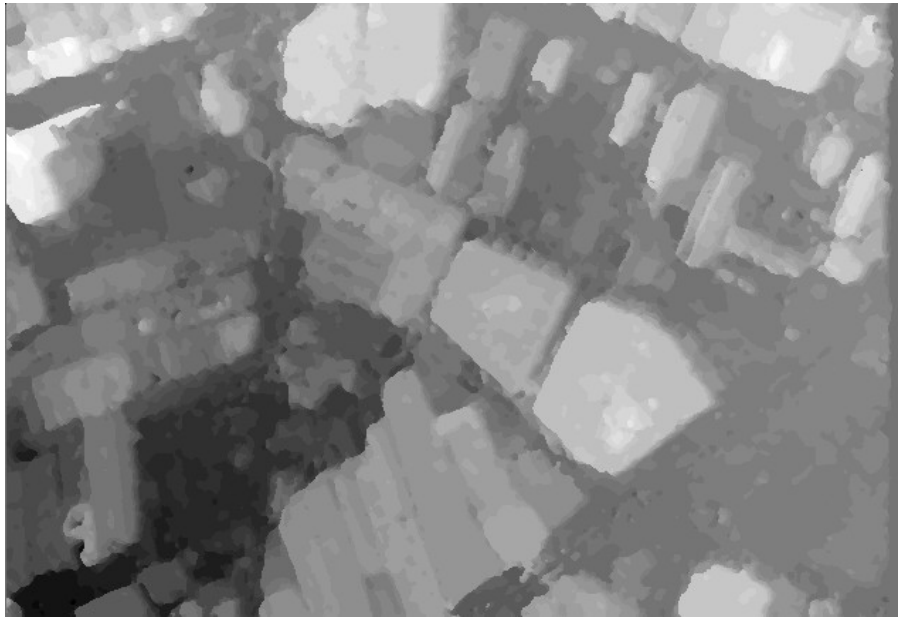
Istanbul test area, Ikonos images



dynamic programming

Comparison DP/SGM

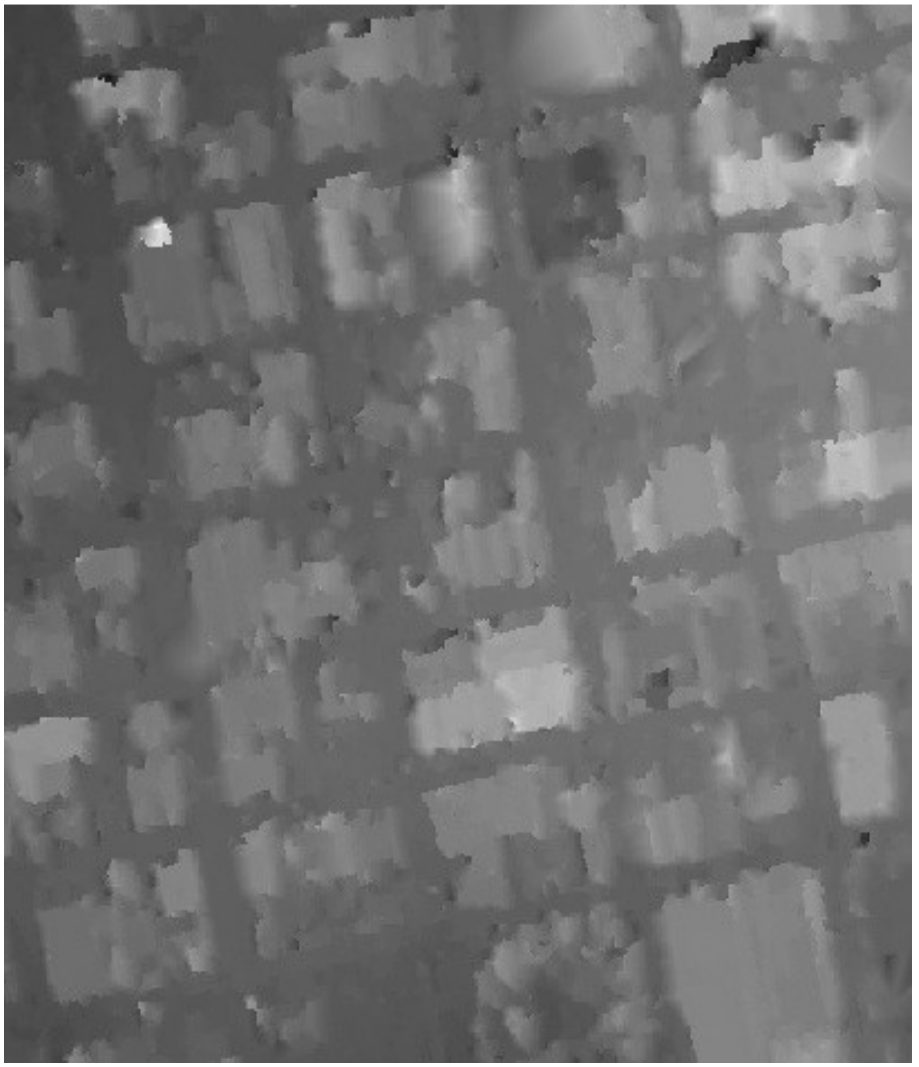
Istanbul test area, Ikonos images



semi global matching

Comparison LSM / DP / SGM

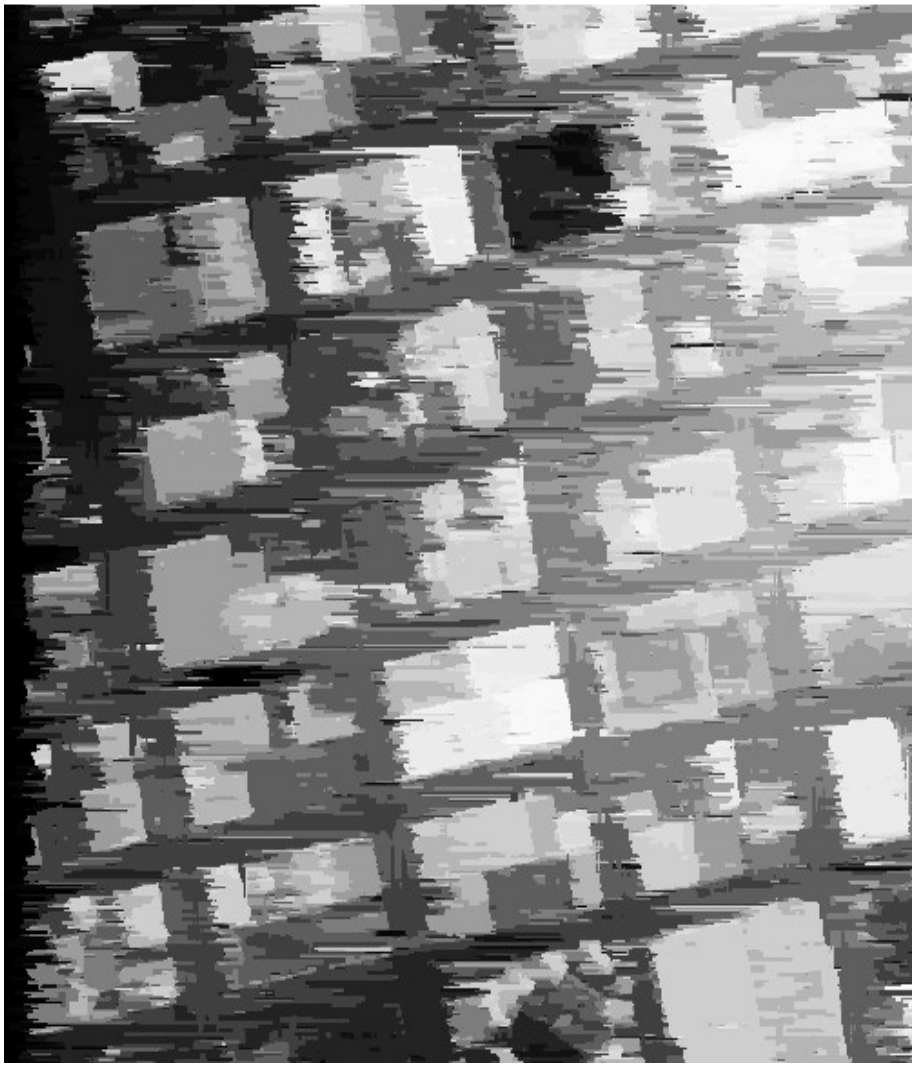
San Diego test area, Ikonos images



Least squares matching

Comparison LSM / DP / SGM

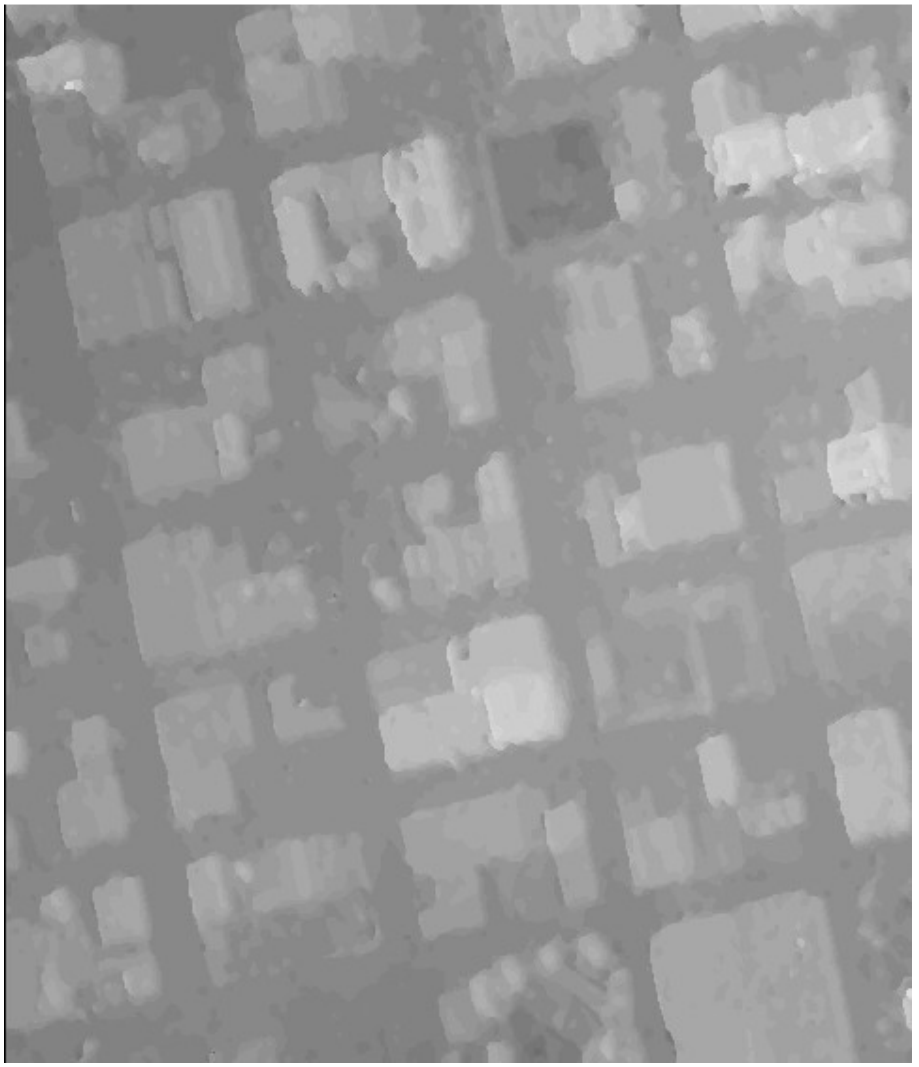
San Diego test area, Ikonos images



dynamic programming

Comparison LSM / DP / SGM

San Diego test area, Ikonos images



semi global matching

Conclusions

- Why dense matching?
  - More details in DSM, in particular in the vicinity of depth discontinuities
  - Useful additional information for automatic object recognition
  - Necessary for true orthophotos and 3D visualization, e. g. for cultural heritage documentation
  - Also used in automatic driver assistance systems (real-time version in FPGA)
- How?
  - Semi global matching or similar global algorithms (e. g. graph cuts)

Conclusions

- Status
  - Increasingly, companies offer software for dense matching, incl. Leica/Intergraph (ISAE), inpho (Match-T), BAE (SocetSet NGATE), Vexcel (UltraMap), Astrium (Pixel Factory), Racurs (PhotoMod)
  - Open source solutions (mainly close range): PMVS, Micmac, …
  - Accuracy in the range of ne pixel GSD

- Comparison to laser scanning
  - \+ Better resolution
  - \+ Faster data acquisition
  - \+ Pictorial documentation of the scene
  - \- Problems with low texture and repetitive texture
  - \- Cannot penetrate forest
  - \- Requires external light source (e. g. no night time data acquisition)