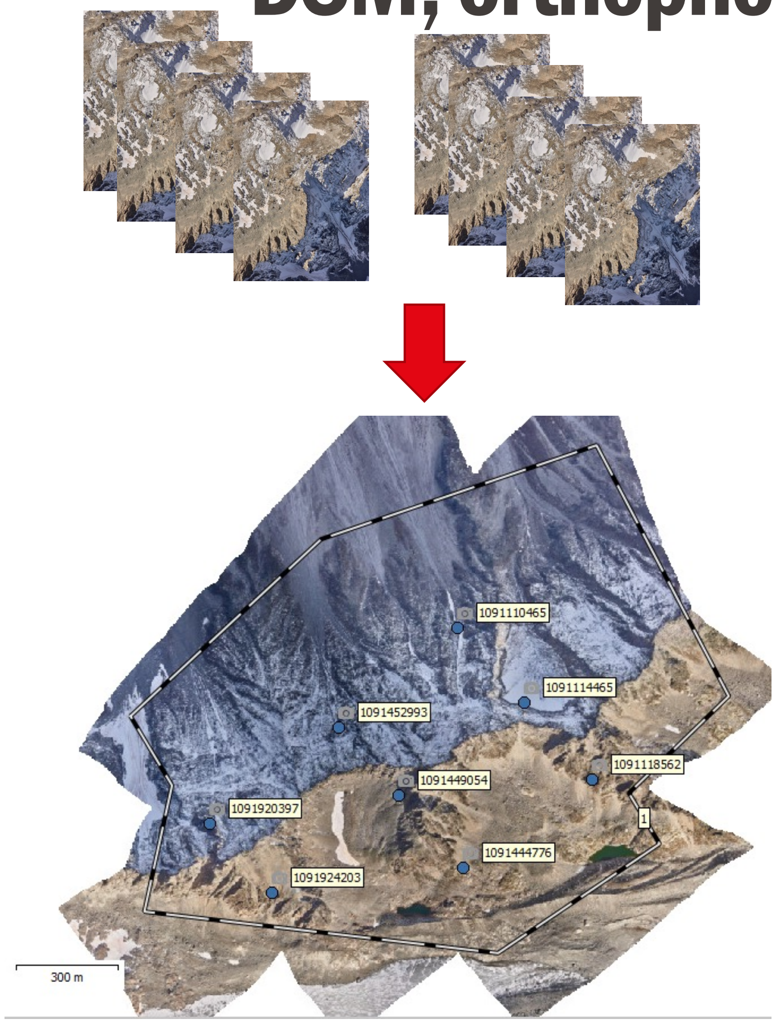
Lecture 5
Optimization & Mapping

ENV408: Optical Sensing & Modeling for Earth Observations

Jan Skaloud

Ex 5 – Optimization, DSM, orthophoto

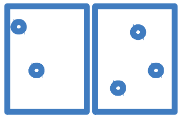


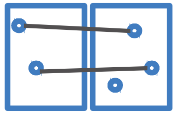
- Tomorrow: use 8 (possibly 45) neighboring photographs (to those used in Ex. 1,2,3,4) together with navigation-derived poses to obtain (via a professional software) a digital surface model with orthorectified image.

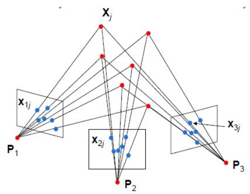
- Today: understand the concepts behind this process and the influencing parameters.

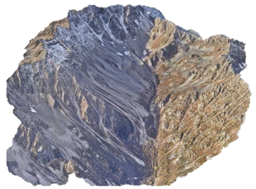
Flight planning | Image acquisition | (Trajectory e.g. navigation sensors) | Images

Feature detection | Feature matching | Orientation 3D reconstruction | Optimization Production surface orthophoto









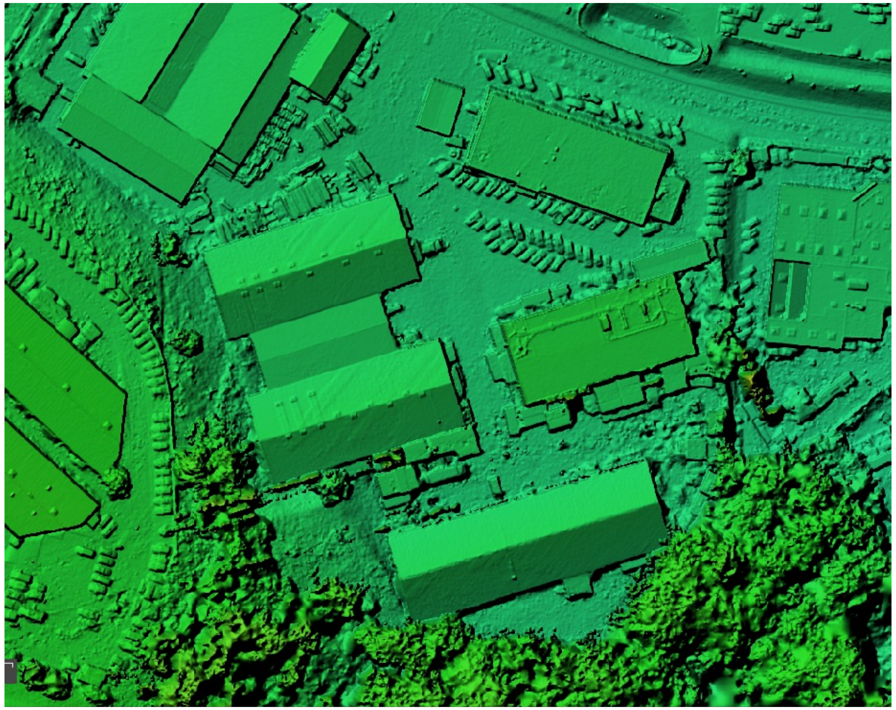
Lectures

- Image primes (L1)
- Salient *features* (L2)
- Image *orientation* (L3)
- Stereo vision (L4)
- Optimization & Mapping (L5)

Exercises

- Image ‘corrections’ (Lab01)
- Detection & matching (Lab02)
- Approx. absolute orientation (Lab03)
- Approx. relative orientation (Lab04)
- Calibration, DEM, ortho-photo (Lab05)

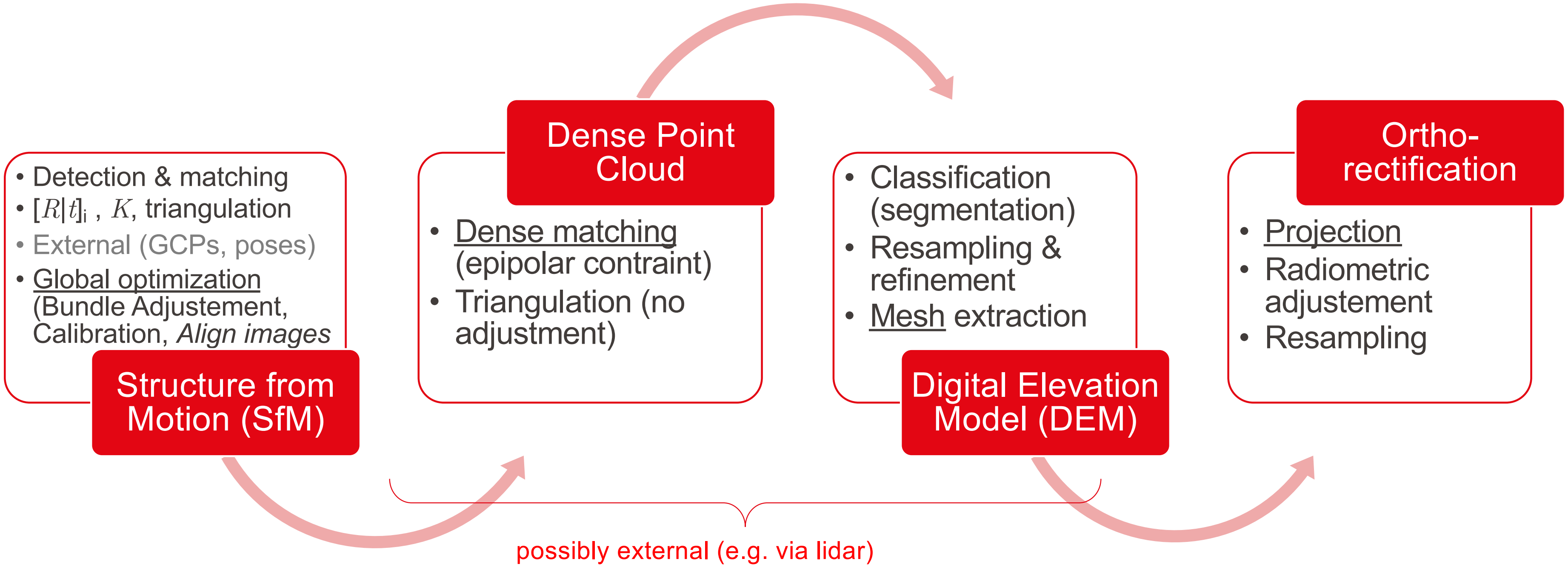
Main mapping products



Digital elevation model (DEM)
colour represents a height above reference surface

Ortho-rectified image (needs DEM)
pixels correspond to Δx, Δy (m) on a projected surface

Main stages


• Detection & matching
• $[R|t]_i$, K, triangulation
• External (GCPs, poses)
• Global optimization (Bundle Adjustement, Calibration, *Align images*
Structure from Motion (SfM)
Dense Point Cloud
• Dense matching (epipolar contraint)
• Triangulation (no adjustment)
• Classification (segmentation)
• Resampling & refinement
• Mesh extraction
Digital Elevation Model (DEM)
Ortho-rectification
• Projection
• Radiometric adjustement
• Resampling
possibly external (e.g. via lidar)

Optimization

Dense matching / 3D point cloud

3D meshes and elevation models

Orthorectification

Appendix

EPFL

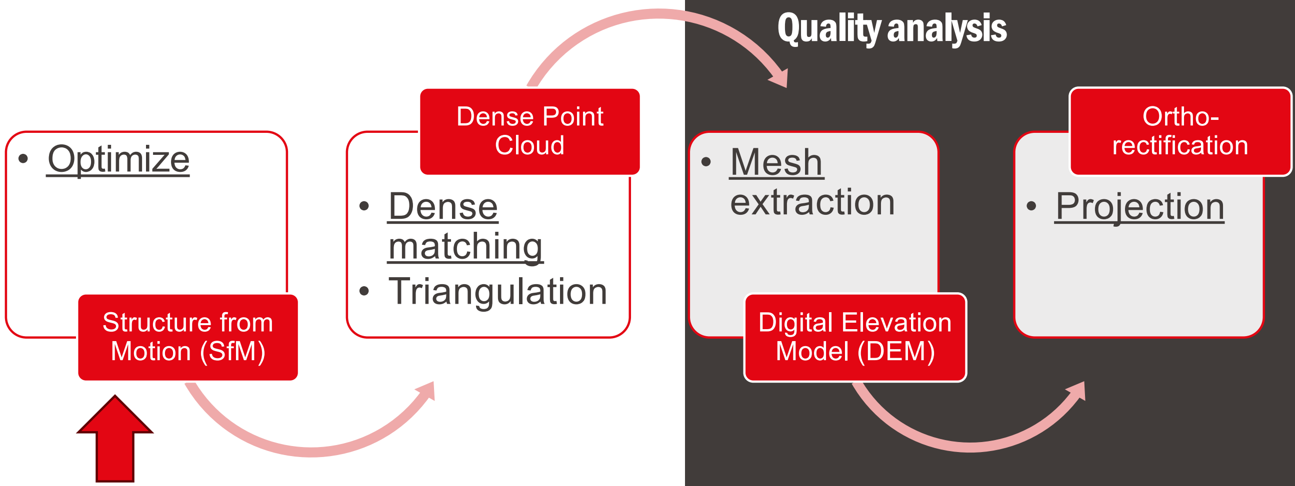
Global optimization

Incremental SfM

Effect of random errors

Additional observations

Quality analysis

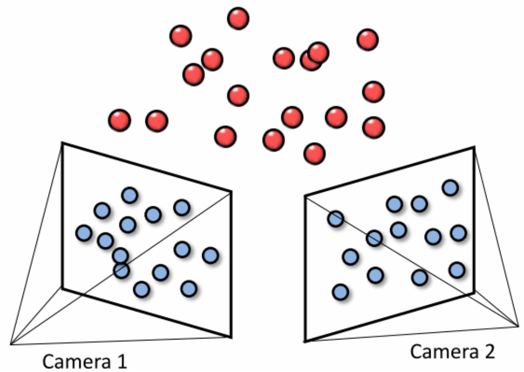


Incremental SfM (Structure from Motion)

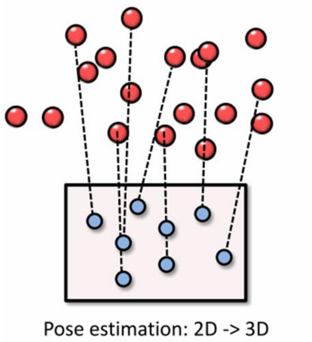
I. Describe & match images (SIFT, Lab02), deduce connectivity matrix (which image matches which image)

II. Initialize 3D model with suitable image pair (coplanarity, Lab04)

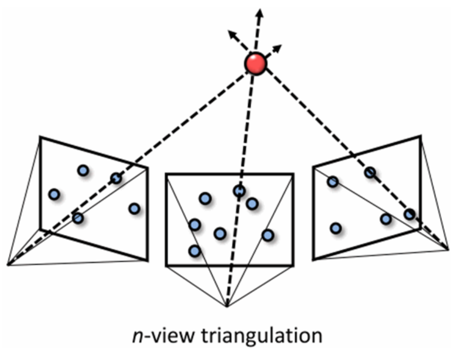
III. Expand model to all images, iterate steps below:
   a. Add connected image
   b. Compute its pose from previously triangulated points (e.g. via DLT, Lab03)
   c. Triangulate new points (coplanarity, Lab04)
   d. Optimize (minimize reprojection error) via collinearity for <u>all</u> images



II.



III.b



III.c

Image: P. Moulon

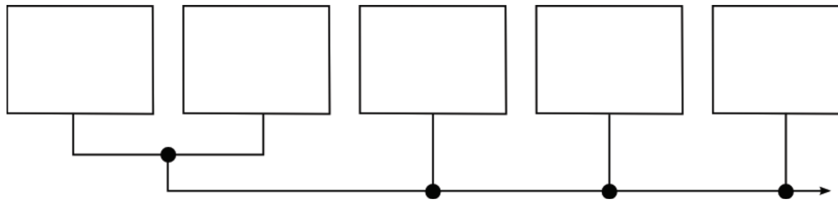
Scene reconstruction without scale*

Estimate essential matrix $\mathbb{E}$ (>4 points)

- Filter wrong matches (RANSAC, lecture 1.2) + extract $[R|t]$

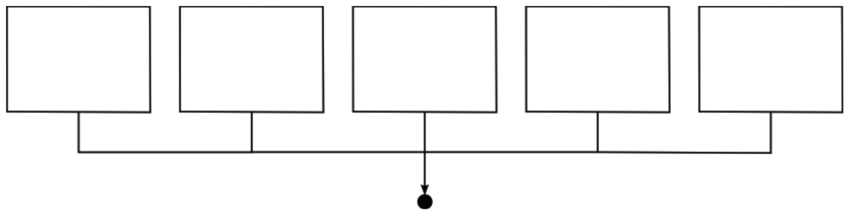
Sequentially (one image pair at a time)

- Good for unknown camera param, pose, poor overlap, etc.
- Initial step/approximation



Global (all)

- Good for scenes with known image distortions, large overlap, sufficient texture, etc.
- Final step



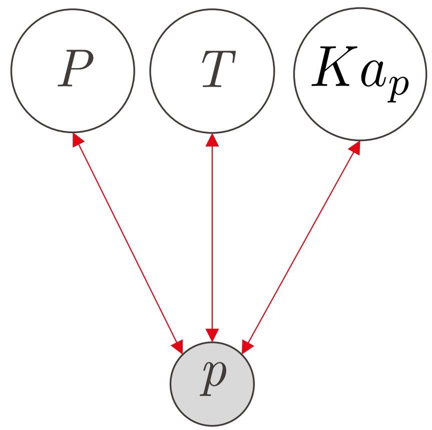
*Scale reconstruction: several possibilities: a. GCPs, b. pose via GPS(/INS), c. a+b (details later)

Scene reconstruction – global optimization

Goal:

increase reconstruction accuracy

- Refine: exterior $\mathbf{T}=(\mathbf{R} \mid \mathbf{t})$ & interior $(\mathbf{K} \mid \mathbf{a_p})$ parameters + image coordinates p_{ij}
- How: using collinearity conditions involving all images j in a set J
- Better identification of extreme (faulty) observations

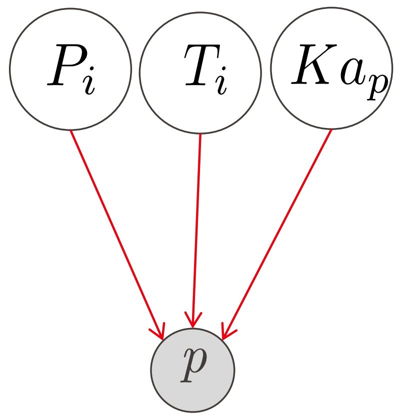


$(\mathbf{P_i}\ ,\ \mathbf{T_j}=[\mathbf{R_j}|\mathbf{t_j}\ ]$ in local-frame, no scale!)

Optimization – starting point

With *minimum* inputs:

- Observed image coordinates $\mathbf{p_i}$
- Keypoints 3D coordinates $\mathbf{P_i}$
- Camera poses $\mathbf{T_j} = [\mathbf{R}|\mathbf{t}\ ]$
- Camera additional parameters (calibration) a_p



Via following observation model (collinearity equations)

$$p_{i,j} + v_{i,j} = \Pi(P_i, T_j, K, a_p)$$

With *optional* additional inputs:

- (possibly) observed point coordinates in object space (GCP) – later
- (possibly) observed pose (position and/or attitude) – later

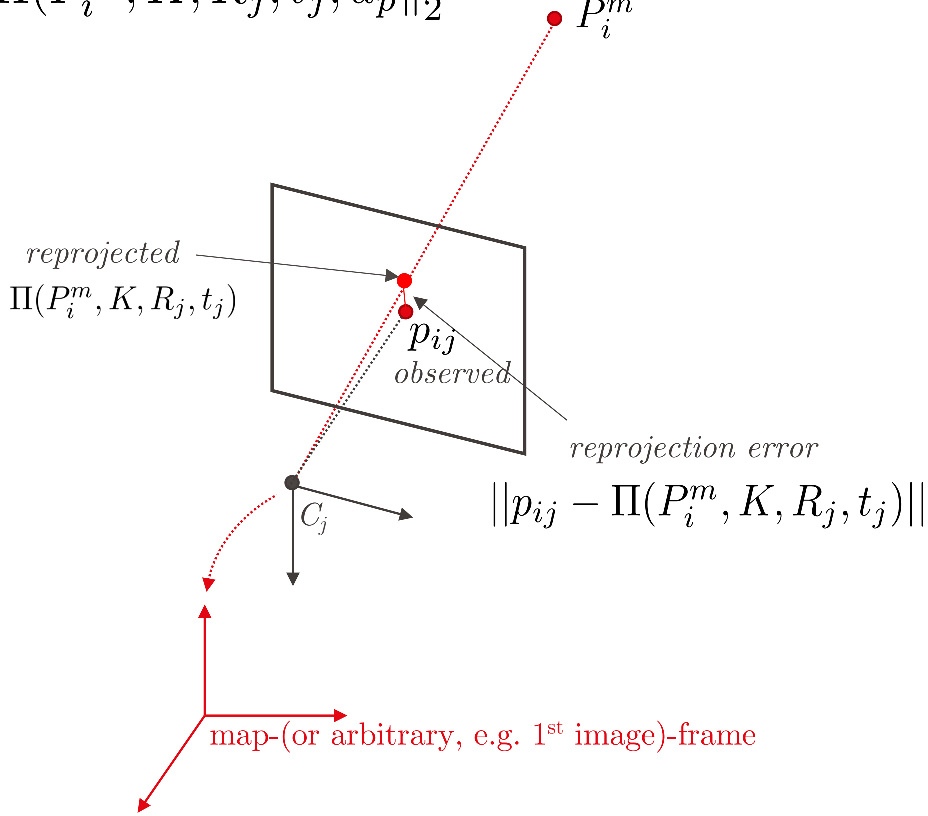
Optimization – general

Minimize errors via maximum likelihood estimate of

$$argmin_{K,R,t,a_p} v = argmin_{K,R,t,a_p} \sum_{i=1}^{n} \|p_{ij} - \Pi(P_i^m, K, R_j, t_j, a_p\|_2$$

With:

- Parameters P_i^m, K, R_j, t_j, a_p
- Joint covariance matrix $C_{\ell\ell}$



How to minimize errors: least-squares

- For errors following (multivariate) normal distribution
- Choose P_i^m, K, R_j, t_j, a_p that minimize the sum of squared errors v weighted by the inverse of their uncertainty (covariance matrix) $C_{\ell\ell}^{-1}$

Optimization – input dependency

- Input
  - Image coordinates (key-points)
  - :
  - (possibly) observed points in object space (later)
  - (possibly) observed camera poses (partial or full - later)
  - :
  - approximated values of exterior or interior orientation & 3D point coordinates!

- Output
  - Follows the fundamental I/O principal

    GARBAGE IN → GARBAGE OUT

  - Adjusted values of all observations + exterior and interior orientation & 3D points

EPFL

Effect of random errors – precision of points

Image block (forward+side overlap)

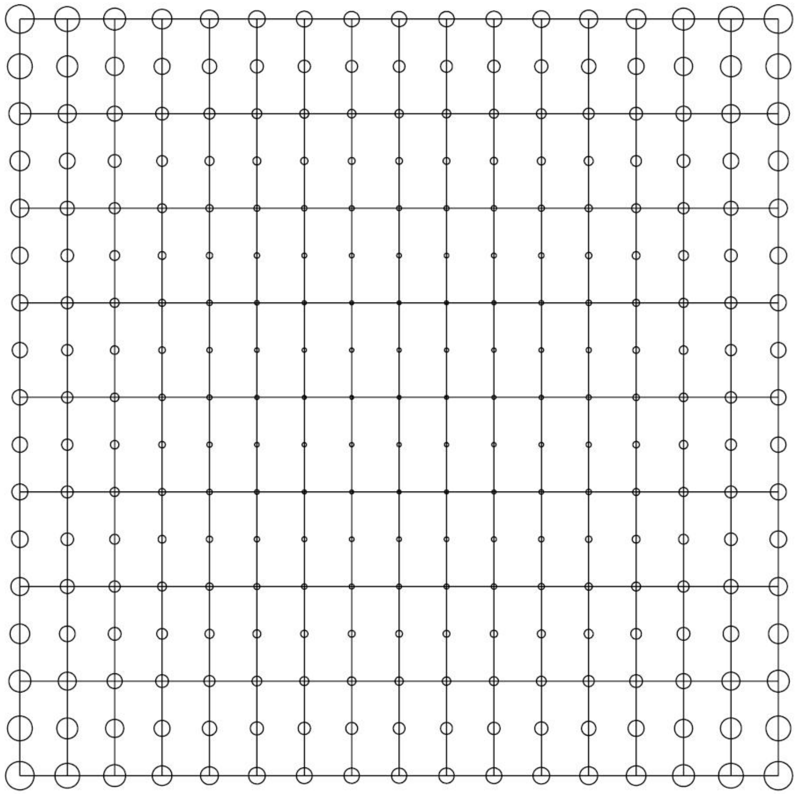
- Only image coordinates (p_i)
- Ideal (hypothetical) case

Prediction / covariance of points $\mathbf{X} P_i$

P_i are part of the parameter space

$$C_{\hat{x}\hat{x}} = (A^T C_{\ell\ell}^{-1} A)^{-1}$$

$$A = \left(\frac{\partial \Pi()}{\partial \mathbf{x}} \right)_{|(\ell, \mathbf{x}_0)}$$



Block

circle/ellipse = uncertainty

after W. Förstner, UAV-g 2017

Effect of random errors in image observations

Image overlap forward + side (if in block*)

When no accurate prior knowledge on camera pose

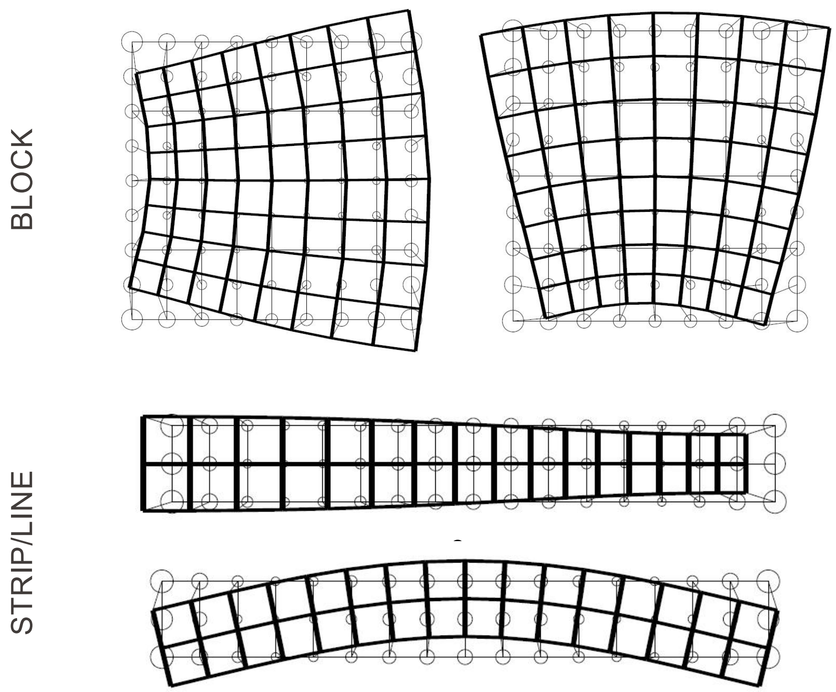
↓

Possible deformation caused by accumulation of random errors

↓

control is needed!

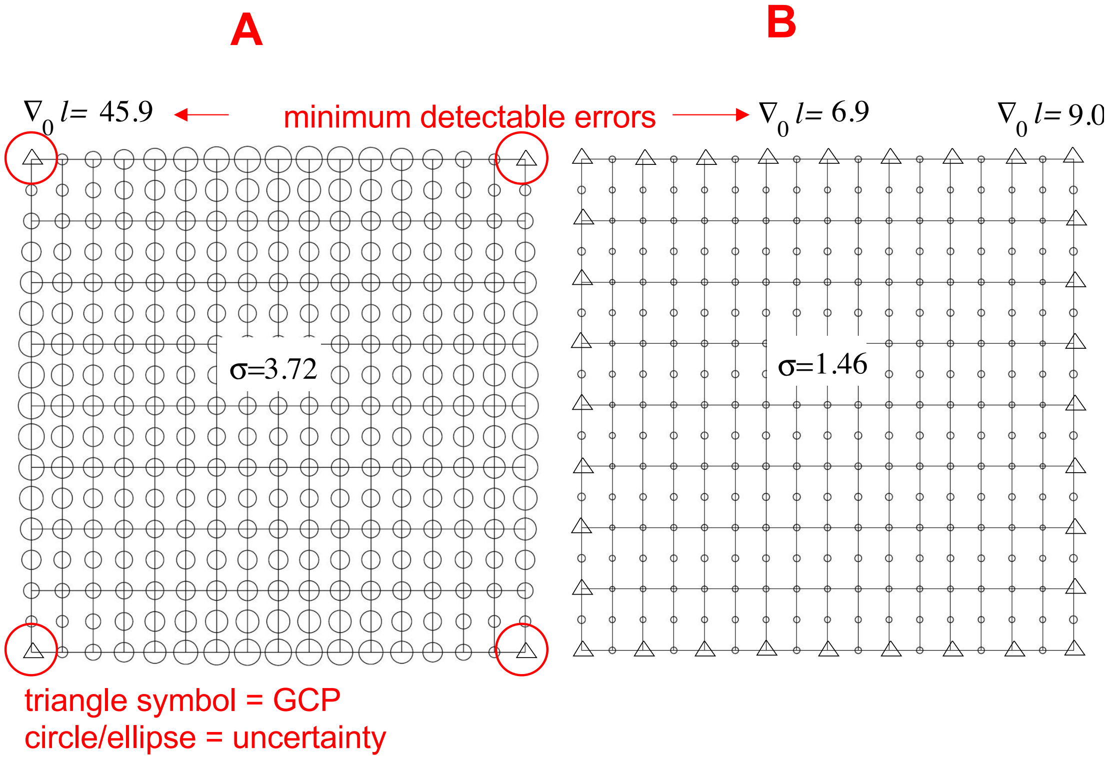
circle/ellipse = uncertainty



after W. Förstner, UAV-g 2017

*block is from from several strips (e.g. flight lines) where images overlap also laterally/sideways

Improving quality
e.g. via ground control points (GCPs)



GCP spatial distribution:

A. Only 4 GCPs at corners
   - Unstable boundaries, 6 sigma
   - GCP outliers (large errors) are not detectable

B. GCPs at boundaries
   - Homogenous precision, < 2 sigma
   - Small GCP outliers not detectable

- Note: a better setup is with precise aerial position (+ attitude)
  - Camera poses observed
  - Fewer or no GCPs are needed! (Lab05)

- After W. Förstner, UAV-g 2017

Optimization – additional observations
Ground control points (GCPs)

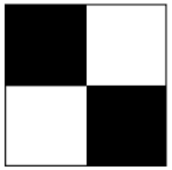
Points with know coordinates in mapping frame, measured on site P_i^m
and known image coordinates (measured manually or semi-automatically) p_i

Using GPS/GNSS - RTK/PPK:

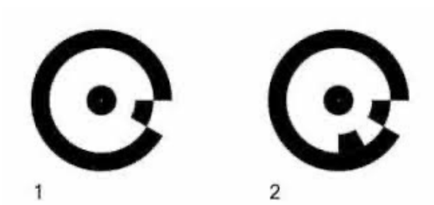
- *1.5-2 cm* horizontal uncertainty
- *2.5-3 cm* vertical uncertainty

Signalization

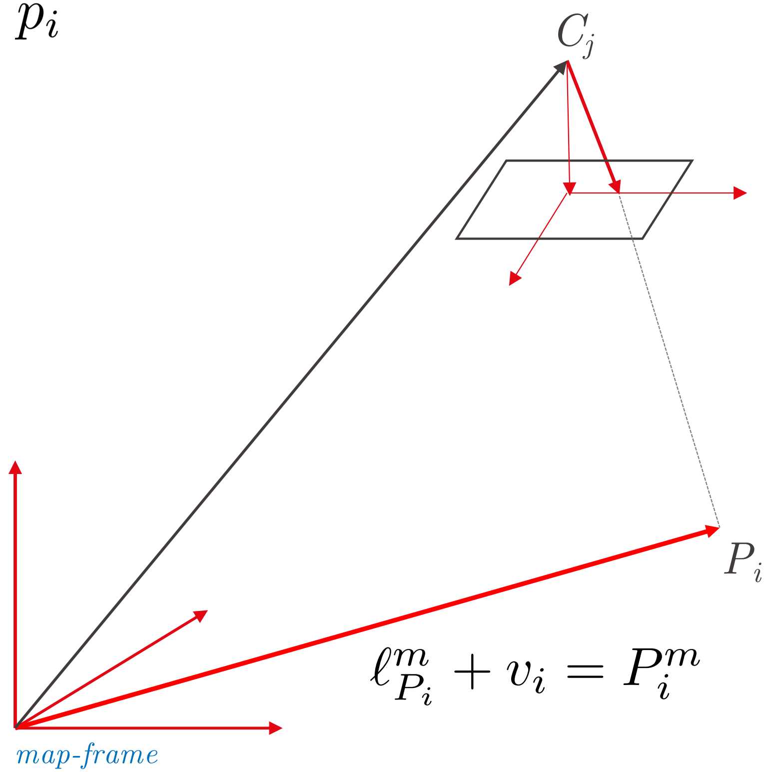
- Uncoded



- Coded



automated identification



Optimization – additional observations

Sensor aerial position + orientation

GPS / GNSS receiver + inertial measurement unit (IMU)

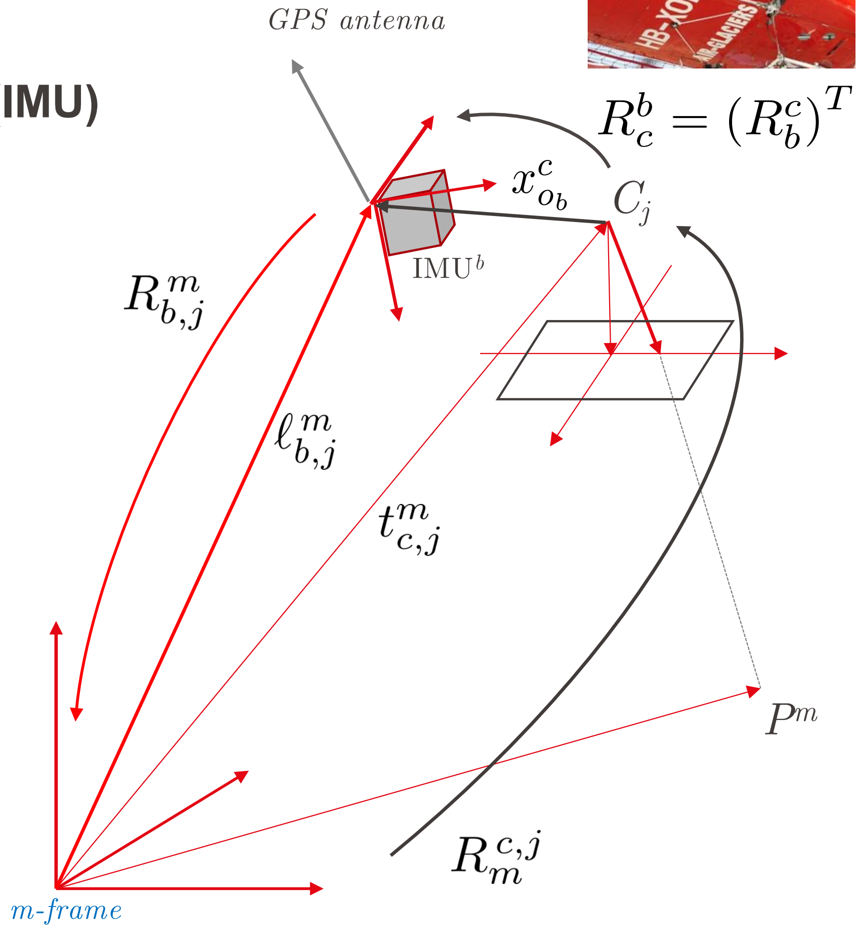
- IMU orientation, interpolated for C_j :$R^m_{b,j}$
- orientation offset (bore-sight): R^b_c
- IMU position, interpolated for C_j : $\ell^m_{b,j}$
- spatial offset to IMU origin (lever-arm): $x^c_{o_b}$

Position observation (GPS only or GPS/IMU)

$$\ell^m_{b,j} + v_{b,j} = t^m_{c,j} + R^m_{c,j} x^c_{o_b}$$

Attitude observation (requires IMU)

$$\ell_{R^{b,j}_m} + v_{R^{b,j}_m} \equiv \left(R^m_{b,j} R_{v,j}\right)^T = \left(R^m_{c,j} R^c_b\right)^T$$



Optimization – precision and accuracy

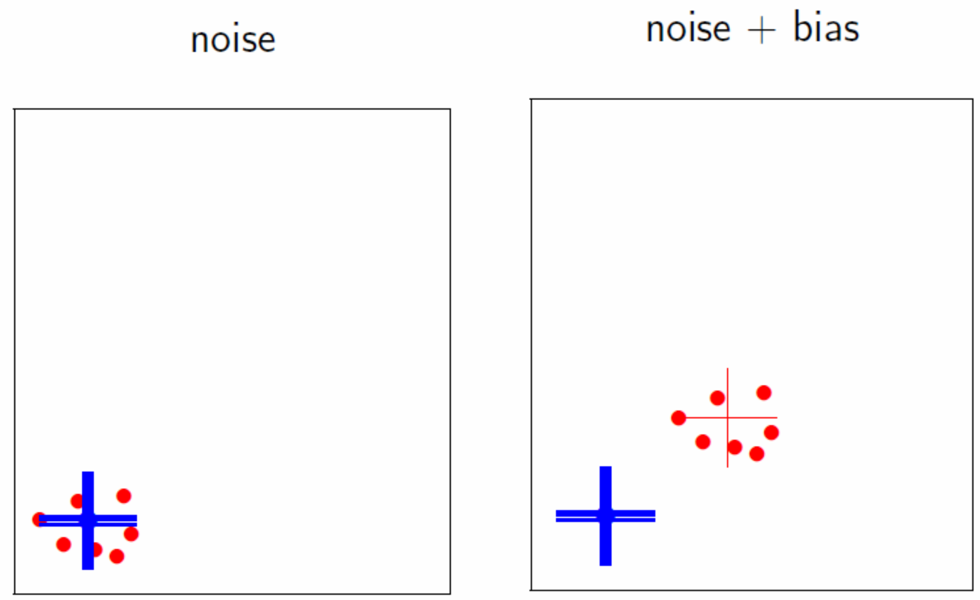
Quantify effect of random errors

Internal measures

- Control correctness
- Needs to be checked

External measures

- Control sufficiency
- Needs to be sufficient



Quality metrics example

for a drone flight

A. Internal

- Discordance between observations
- Size & distribution of residuals vs. estimated precision

B. External

- comparison with ground truth, e.g. < 2 x GSD (ground sampling density)
- e.g. using reference values, points (check points = coordinates in object space observed, but not used in optimization)

Metric - Mean square error (MSE) of:

- reprojection error (all image observations)
- aerial pose (position & attitude)
- check points (in object space, metric)
- check points (in image space, pixels)

Example – summary

Number of images:	440
Flying altitude:	155 m
Ground resolution:	2.66 cm/pix
Coverage area:	0.547 km²
Camera stations:	440
Tie points:	381,168
Projections:	1,698,383
Reprojection error:	0.445 pix

Quality report example
Camera calibration

Type	Resolution	Focal Length	Pixel Size
Frame	7920 x 6004	35 mm eq.	4.6 10E-6 m

K (c, Cx, Cy) — a_p (B1, B2, K1, K2, K3, P1, P2)

	Value	**Error**	**F**	**Cx**	**Cy**	**B1**	**B2**	**K1**	**K2**	**K3**	**P1**	**P2**
c	**5607.25**	0.018	1.00	0.00	0.04	-0.13	0.01	-0.35	0.32	-0.29	0.01	-0.00
Cx	**30.2154**	0.021		1.00	-0.01	0.02	-0.09	-0.00	-0.00	0.00	0.82	-0.01
Cy	**52.4225**	0.018			1.00	0.14	0.04	0.01	-0.00	0.00	-0.00	0.66
B1	**-0.0246013**	0.003				1.00	-0.00	0.04	-0.06	0.07	-0.02	-0.08
B2	**0.120007**	0.003					1.00	-0.00	0.00	-0.00	0.08	0.00
K1	**0.0161593**	3e-005						1.00	-0.97	0.92	0.00	0.02
K2	**-0.0621043**	0.00022							1.00	-0.99	-0.00	-0.01
K3	**0.063624**	0.00048								1.00	0.01	0.01
P1	**-0.000934223**	1.2e-006									1.00	0.01
P2	**0.00104344**	9e-007										1.00

Table 2. Calibration coefficients and correlation matrix.

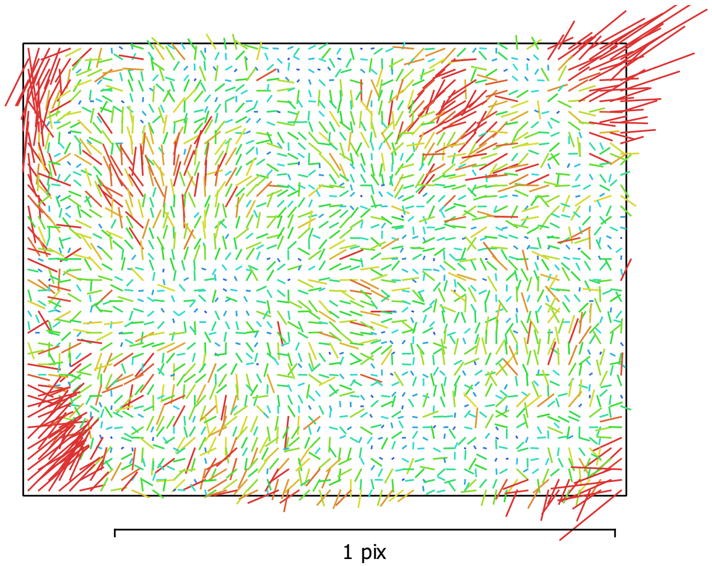


Fig. 2. Image residuals for unknown.

Quality report example
Camera location
internal metric - prediction

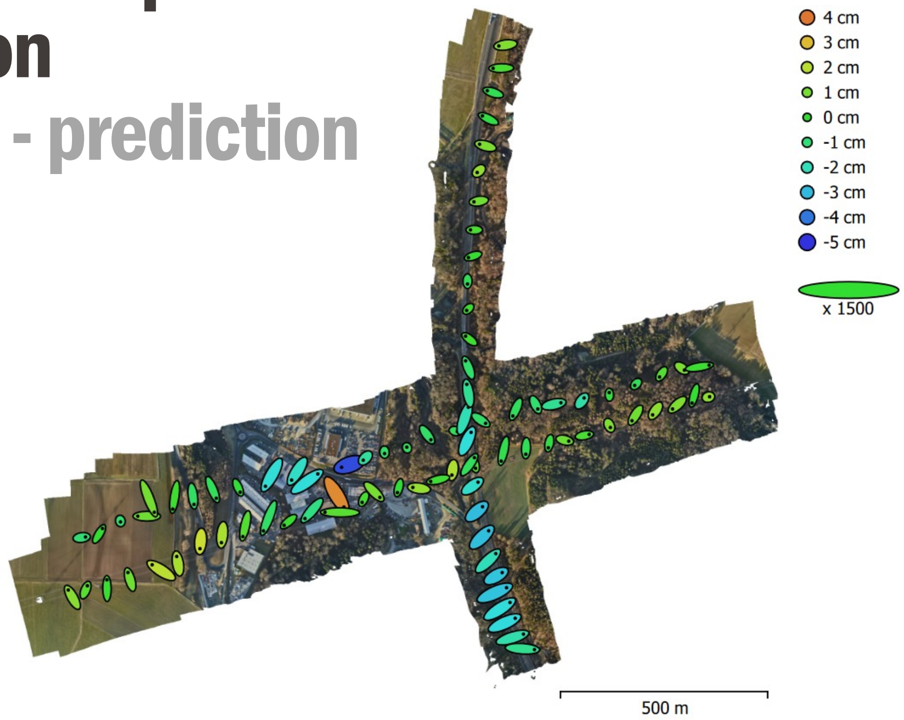


Fig. 3. Camera locations and error estimates.

X error (cm)	Y error (cm)	Z error (cm)	XY error (cm)	Total error (cm)
1.73	1.8	1.4	2.5	2.9

Table 3. Average camera location error.
X - Easting, Y - Northing, Z - Altitude.

Quality report example Check* points

external metric evaluation

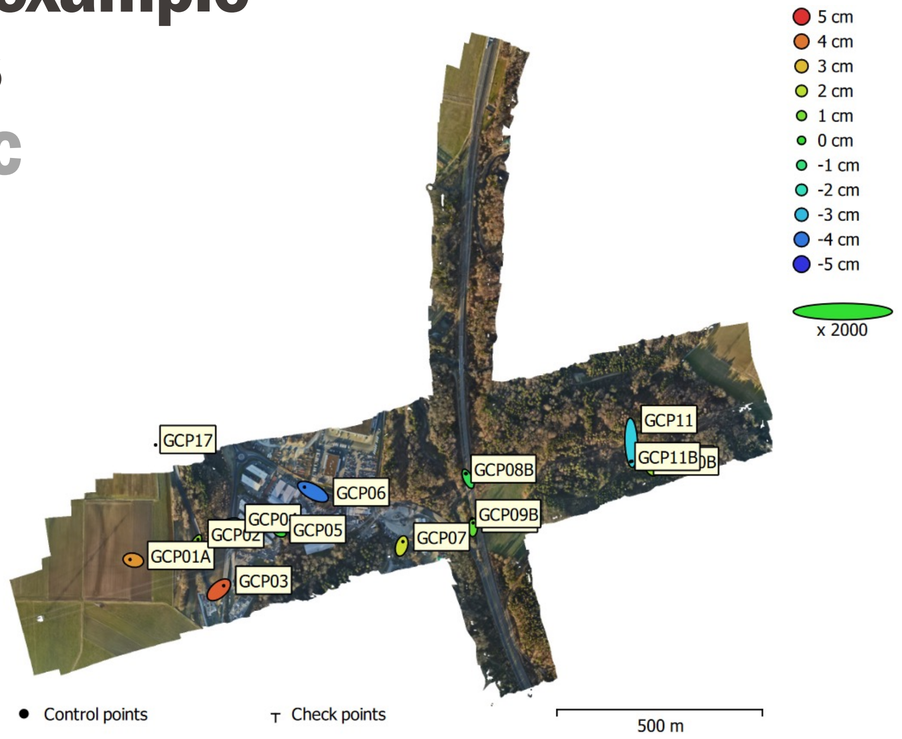


Fig. 5. GCP locations and error estimates.

*check points
are GCP that are used as tie-points, i) position of which is determined via photos & evaluated against known position.
(i.e. their known object coordinates are NOT used to constrain optimizations)

Count	X error (cm)	Y error (cm)	Z error (cm)	XY error (cm)	Total (cm)
11	0.9	1.7	2.5	1.9	3.18

Table 5. Control points RMSE.
X - Easting, Y - Northing, Z - Altitude.

What are the principal factors Influencing precision/accuracy?
→ Appendix II.

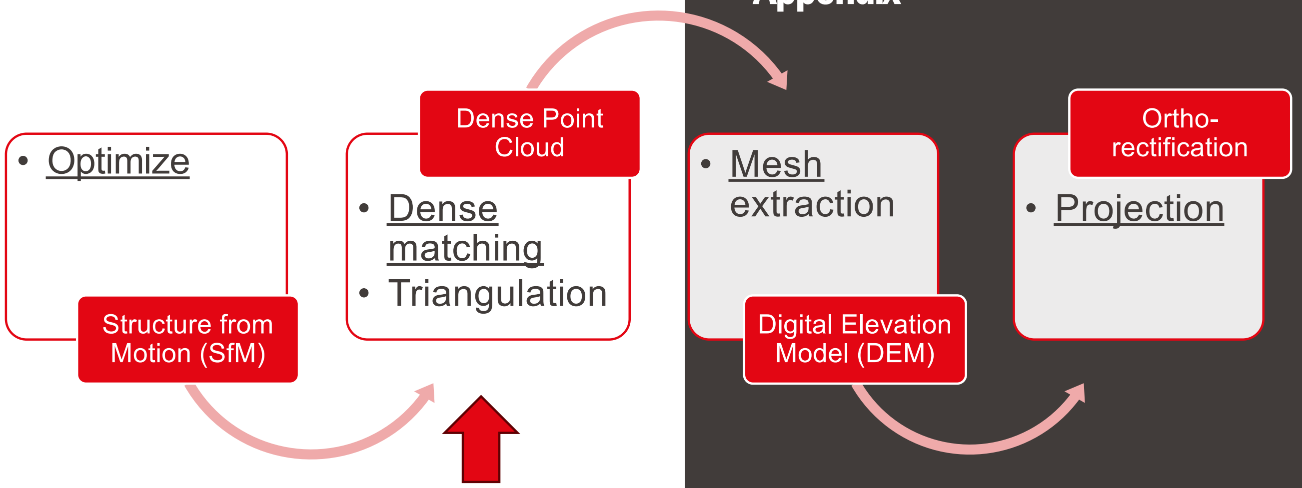
Optimization

Dense matching / 3D point cloud

3D meshes and elevation models

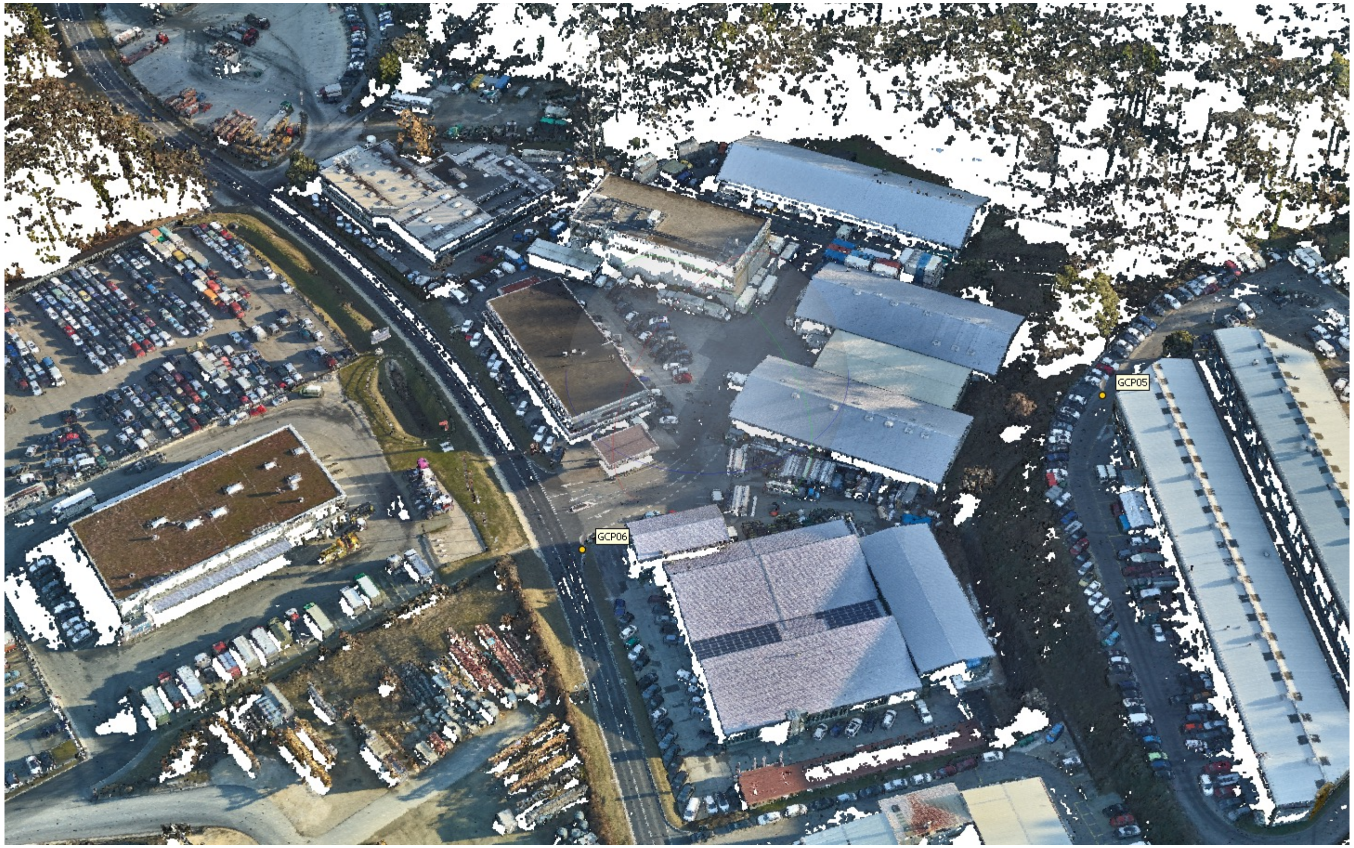
Orthorectification

Appendix



3D point cloud from multi-view stereo

example



Potential accuracy of 3D point cloud 0.5-2 pixels (at image scale) so 0.5- 2x GSD.
Challenges: occlusion (see Appendix IV.), semi-transparent structures, homogenous texture.

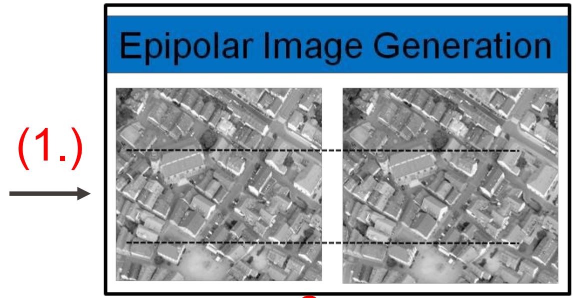
High-density image matching stages

Required
Image orientation parameters (+global optimization), Lab 03 + 04 + 05

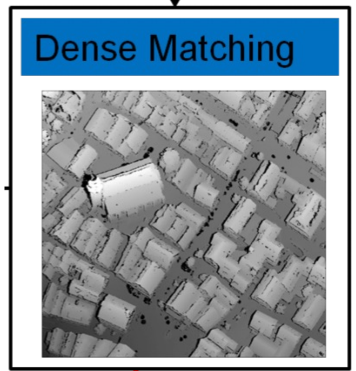
Steps

1. Epipolar image generation (optional, appendix III.)
   i.e. image transformation for simplified matching
2. Dense matching
   disparity (per pixel) along epipolar lines
3. Triangulation
   we know how, Lab 04

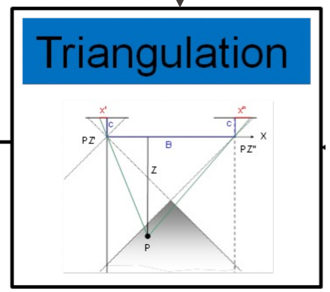
Further process
Digital elevation model – see below



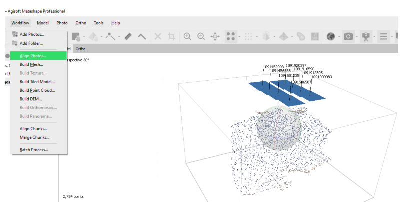
2.



3.



Agisoft – Align Photos



dense point cloud

Source: N. Haala, EuroSDR 2019

Epipolar lines = rows in the epipolar images

What it does ?
- 2D Transformation
- Epipolar lines = rows

Why doing that ?
- Simplification for dense matching
- Region algoritm algo

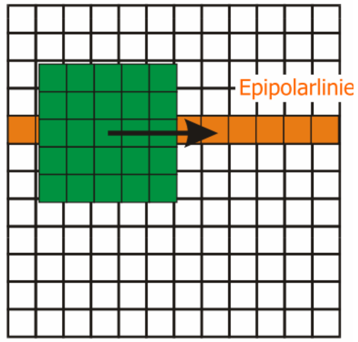
How ?
- 5 slides on “epipolar image generation” are in Appendix III.

Source: N. Haala, EuroSDR 2019

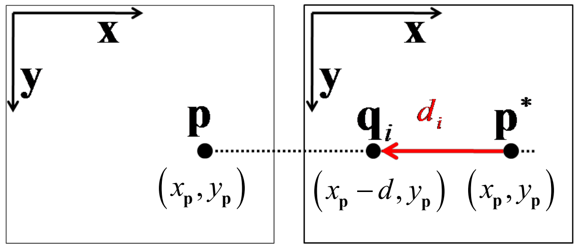
Dense pixel matching

Epipolar images: pairs of epipolar lines are parallel to rows

- This simplifies the matching to 1D search
- Correspondences per each pixel can be stored in disparity (parallax) images (shade = distance)



Base image Match image Parallax image

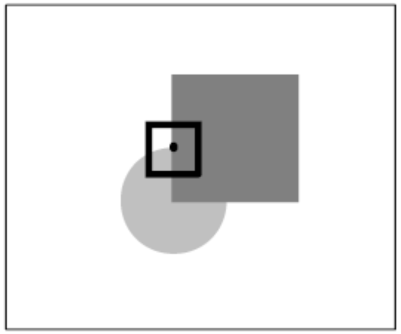


Base image I_1 Match image I_2

Source: N. Haala, EuroSDR 2019

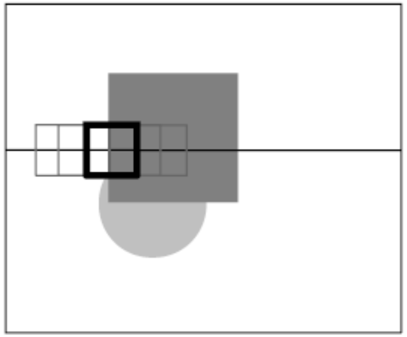
Window based matching

Challenge: Homogenous texture results in matching ambiguity (*same* gray values for many *different* object points)



Goal: Increase robustness of similarity measure by comparing matching windows (correlation).

Condition: implies constant disparity (depth) within the mask



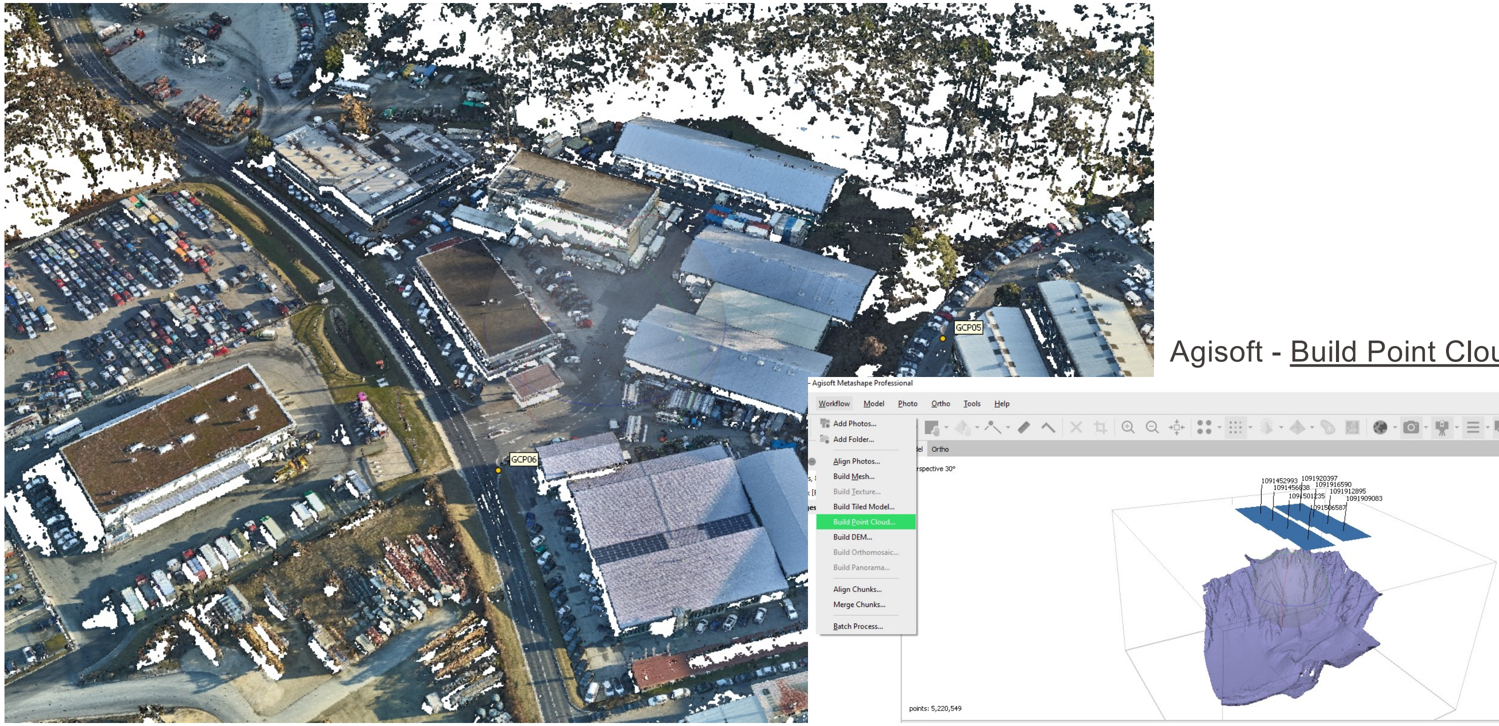
Side effects: Blurred edges, loss of resolution (small objects)

Other approaches: hierarchical stereo matching, segmentation by Markov fields, graph cut labeling, multi-stereo (across several images) matching, deep learning …

- Source: Hirschmuller, Heiko 2005, Accurate and efficient stereo processing by semi global matching and mutual information.

Dense matching + triangulation = 3D cloud / image pair

Union of stereo pairs: combined 3D cloud (in mapping frame)



Agisoft - Build Point Cloud

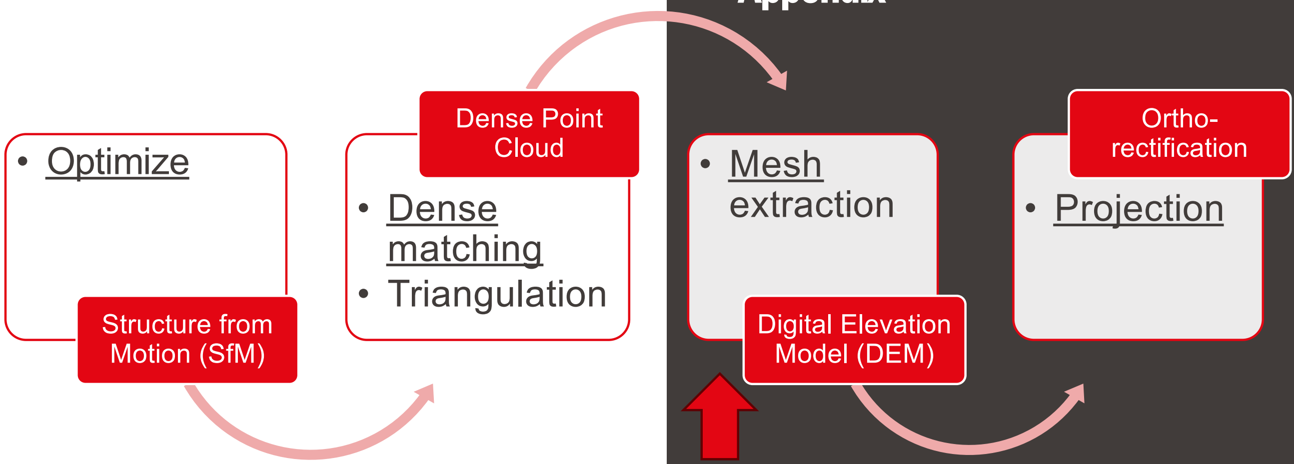
Optimization

Dense matching / 3D point cloud

3D meshes and elevation models

Orthorectification

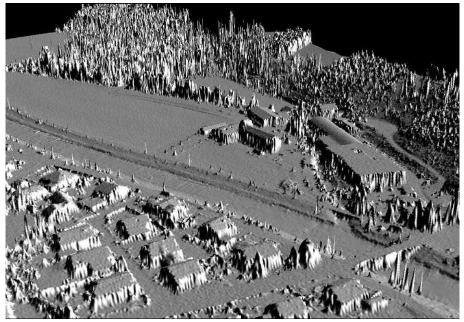
Appendix



Types of digital elevation models (DEM)

Digital surface model (DSM)

Representation of Earths surface including buildings and vegetation

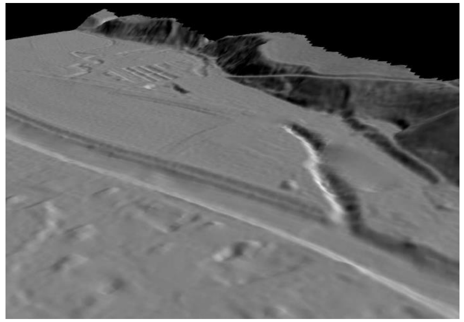


© Ressl, TU Wien, 2007

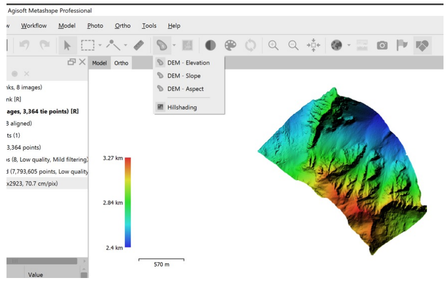
Filtering

Digital terrain model (DTM)

Representation of Earth surface without buildings and vegetation



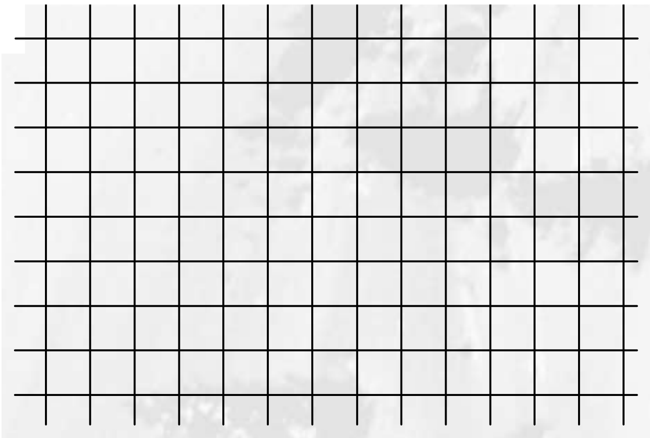
Agisoft – Build DEM



Elevation model representation types (2.5D)

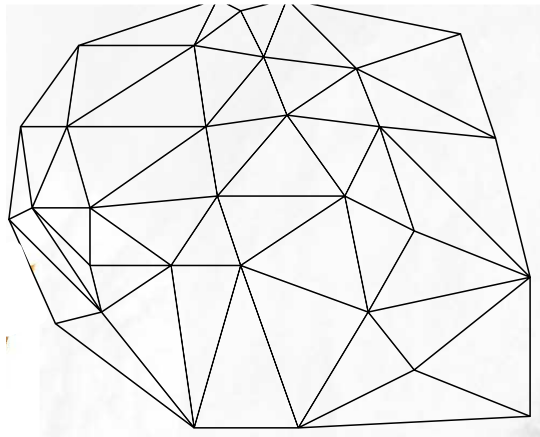
Raster (2D vector)

- ✓ easy to use (format)
- ✓ Once uper-left and lower-right cornes only defined with spacing, only sequence of height needed (as an image)
- ✓ Choice for national models at different resolutions



Triangulated irregular network (TIN)

- ✓ More efficient in respresenting varying levels of details
- ✓ Preserves break-lines
- ✓ Can be extended to 3D



Point cloud organisation

explicit surface representation priors

Properties

- Massive point clouds (>10^9 points)
- Not all can be in computer memory
- Not all needs to be treated together

Strategies

- Sub-sample
- Divide into blocks (Tiles)
- Efficient spatial data structure (K-d Tree)

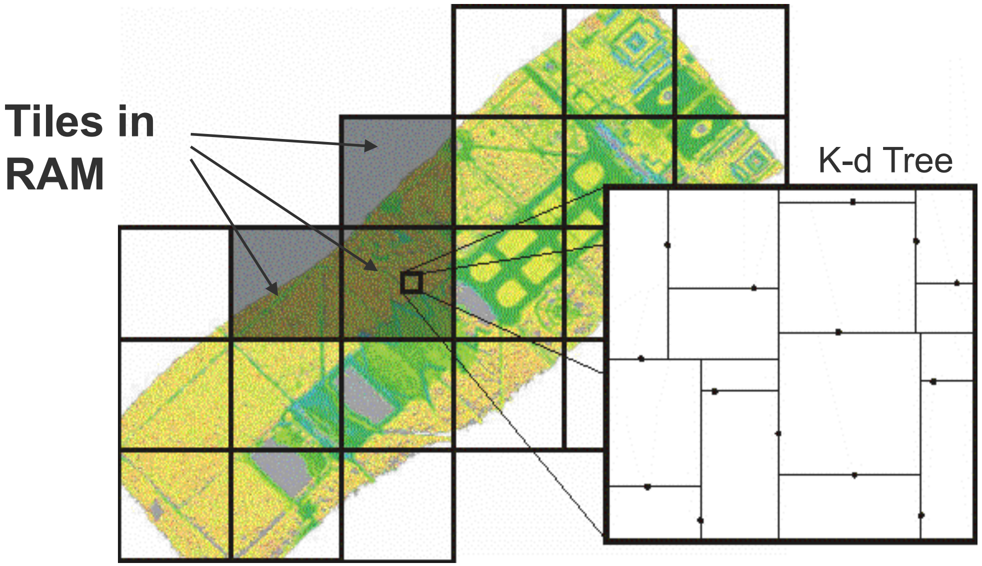


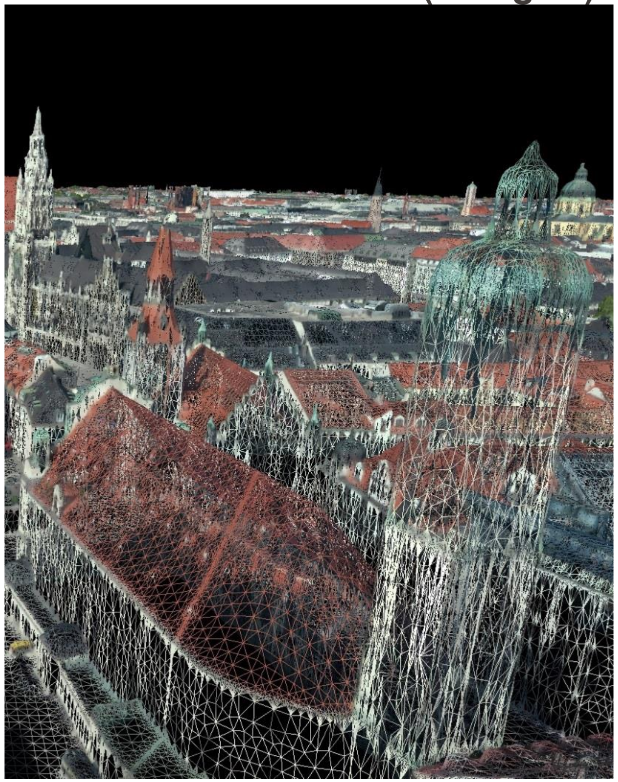
Image credit: Haala, 2019

3D Meshing usage

3D model

3D model with texture

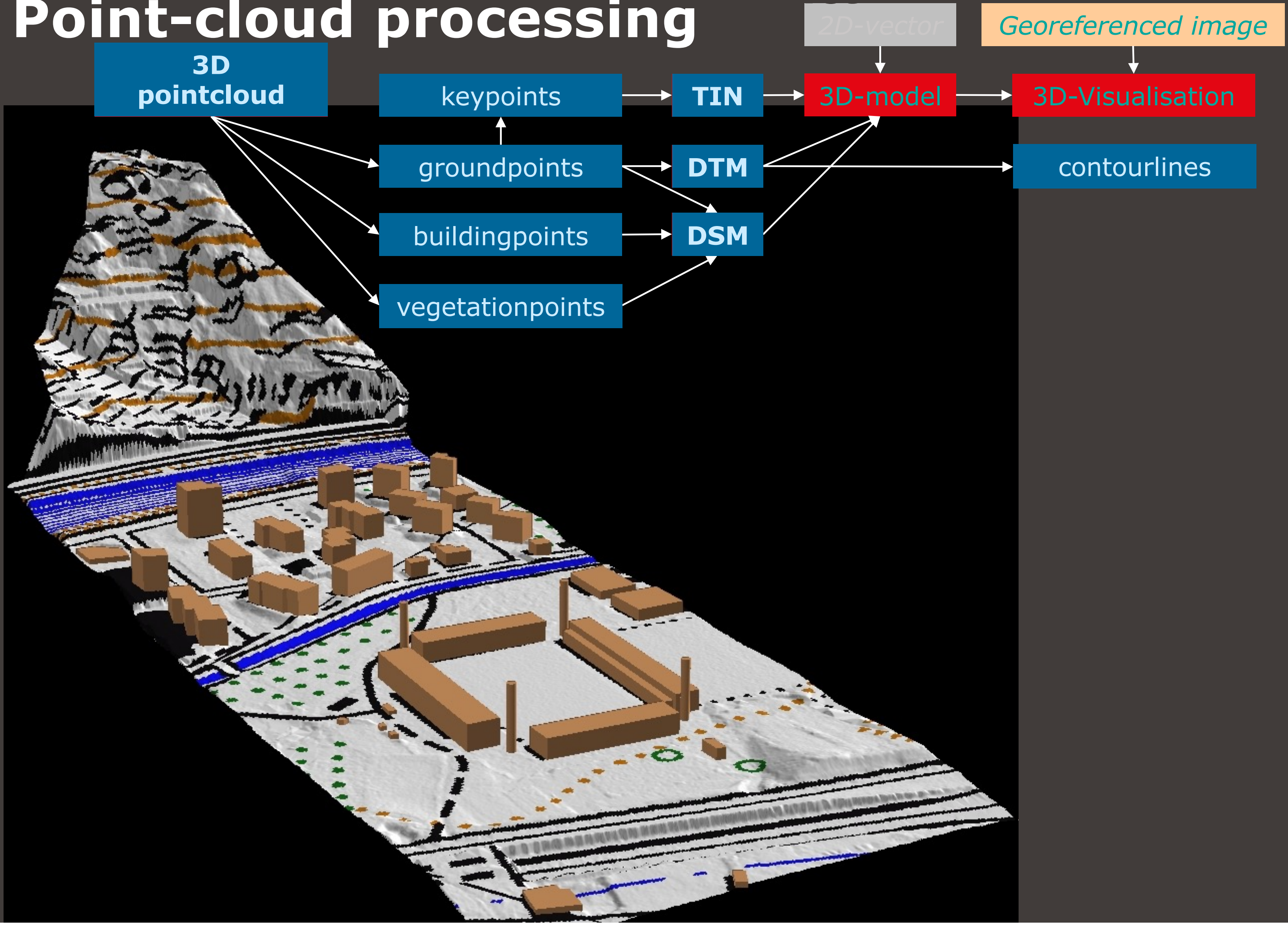
3D model as a 3D mesh (triangles)



Source: N. Haala, EuroSDR 2019

EPFL

Point-cloud processing



Triangle meshes
explicit surface representation

Consists of geometric and topological component

- Set of vertices
- Set of triagular faces connecting them (often respresented as *edges of a graph*)

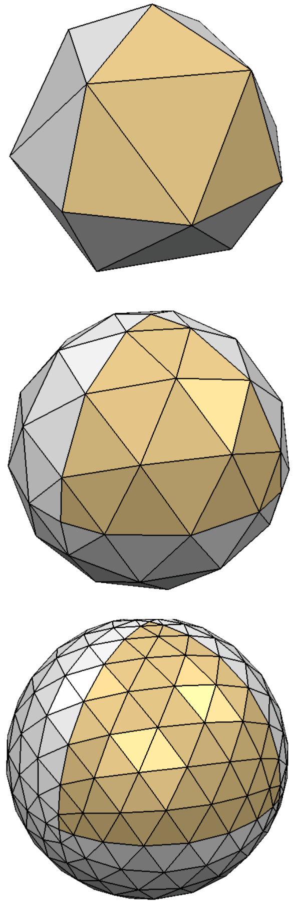
Continuous piecewise linear surface

- Approximation error of order $O(h^2)$, h = max. edge length
- Error reduced by ¼ for ½ h

Properties

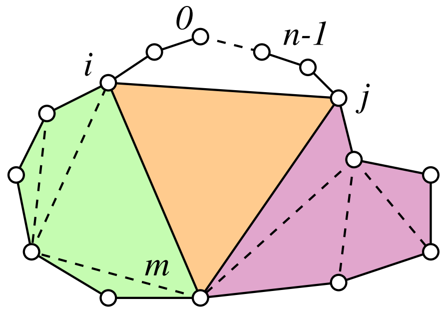
- No. triangles = 2x No. of vertices
- No. edges = 3 x No. vertices
- Average No. of incident edges* = 6

*incident edges are sharing common vertex



Botsch,et al. 2006

Mesh data structures



Data structures

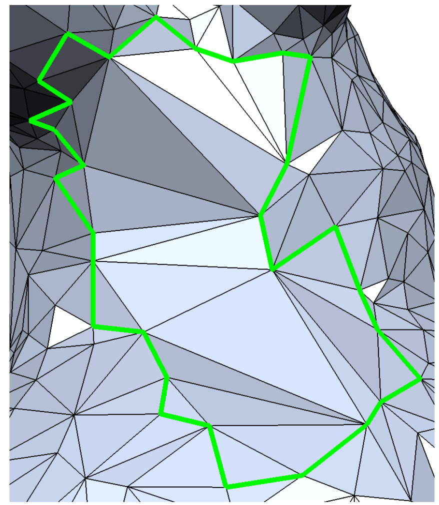
- Face Vertex List - direct edges (straight forward implementation, complex processing)[1]
- Doubly Connected Edge List (DCEL) – half edges (processing efficient, enforces manifold structure, implementation complex)[1]

Mesh libraries

- CGAL (Computational Geometry Algorithms Library) – robust, efficient, scalable
- OpenMesh – efficient for processing based on halfedge data structures

Computation

- Purpose: visualisation, 2.5D models, 3D models, quality, breaklines, smoothness, level of detail, topology (clustering)
- Implies filtering, editing, corrections …

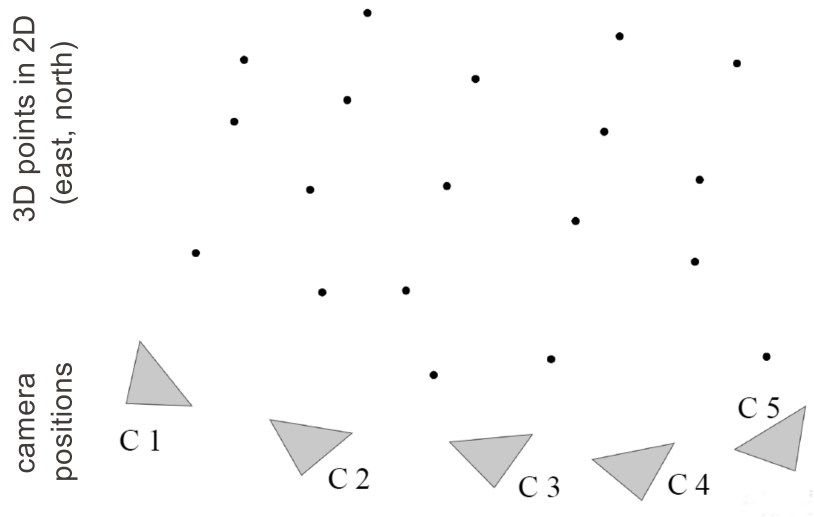


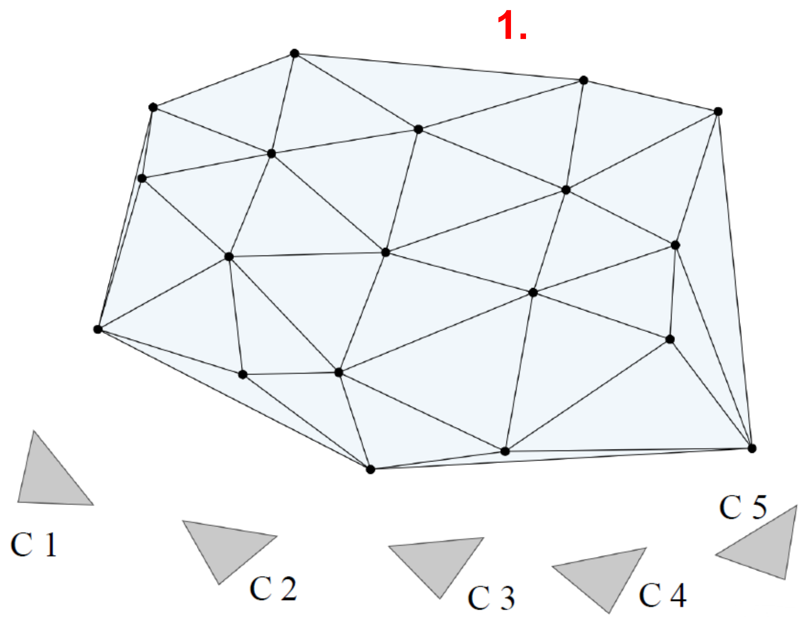
Goal: minimizes normal variation and total area.

Source: Botsch et al. 2006

[1] – https://observablehq.com/@2talltim/mesh-data-structures-traversal

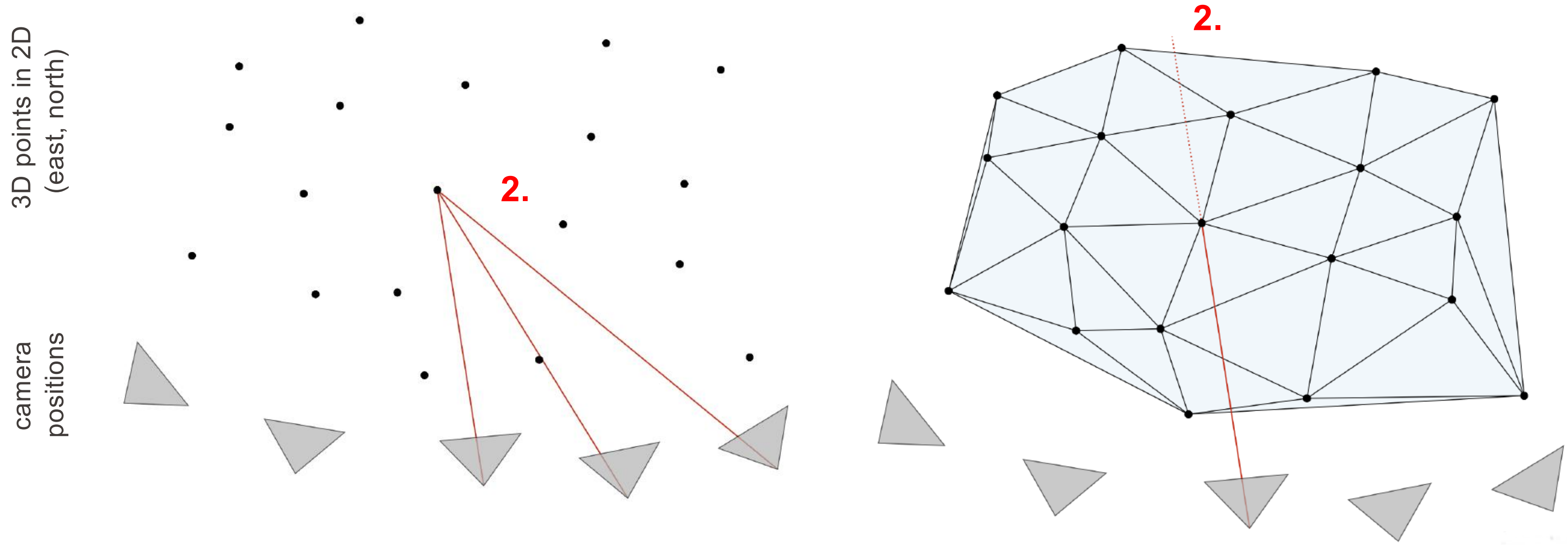
Example of 2D triangulation





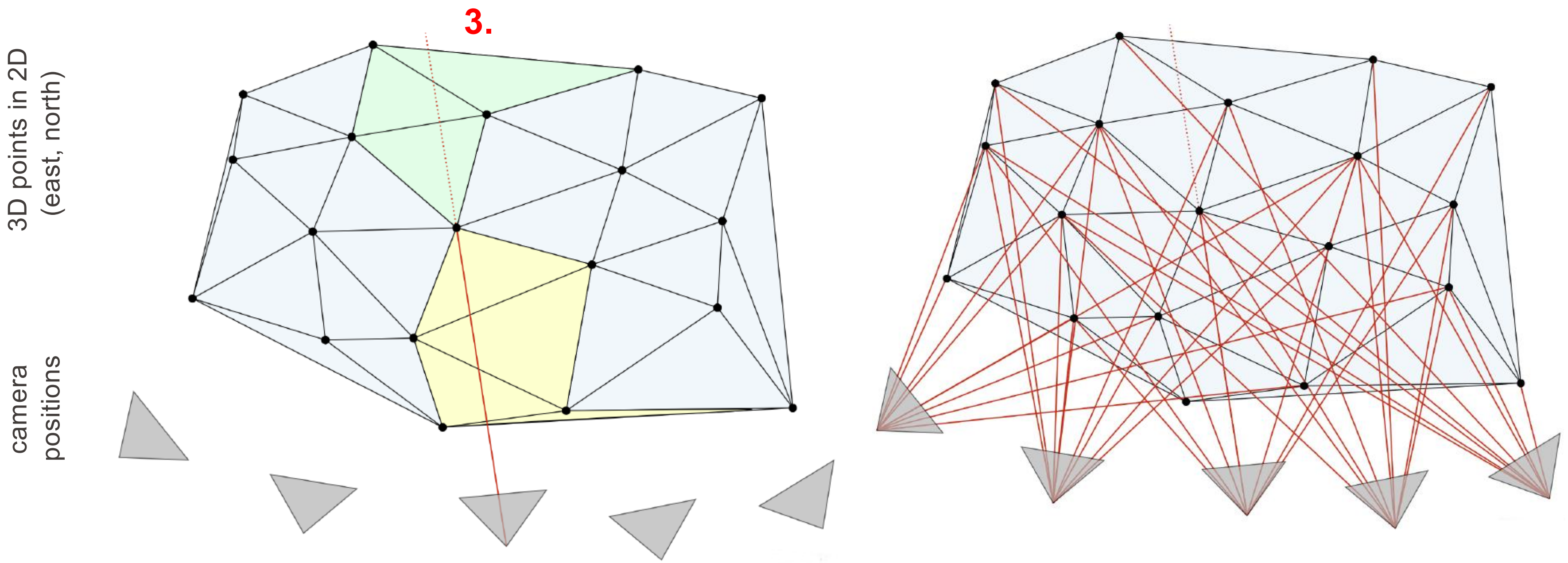
1. Delaunay-Triangulation of point cloud (in 2D / horizontal plaine)

Example of 2D triangulation



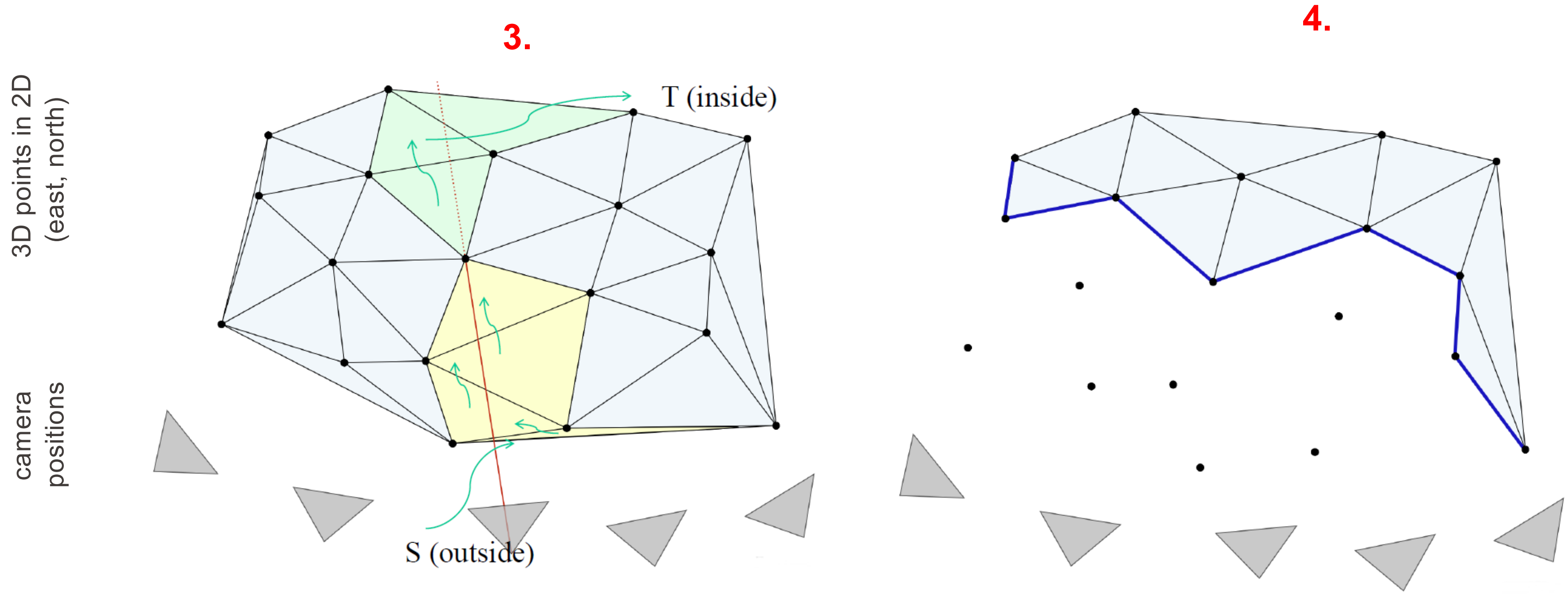
1. Delaunay-Triangulation of point cloud
2. Each 3D point defines (free) line of sight to the corresponding camera station

Example of 2D triangulation



1. Delaunay-Triangulation of point cloud
2. Each 3D point defines (free) line of sight to the corresponding camera station
3. Label each Delaunay cell as outside or inside of the observed object (while considering visibility and geometric quality)

Example of 2D triangulation



1. Delaunay-Triangulation of point cloud
2. Each 3D point defines (free) line of sight to the corresponding camera station
3. Label each Delaunay cell as outside or inside of the observed object (while considering visibility and geometric quality) – optimization by Graph-cut
4. Extract surface (ok for 2.5D representation)

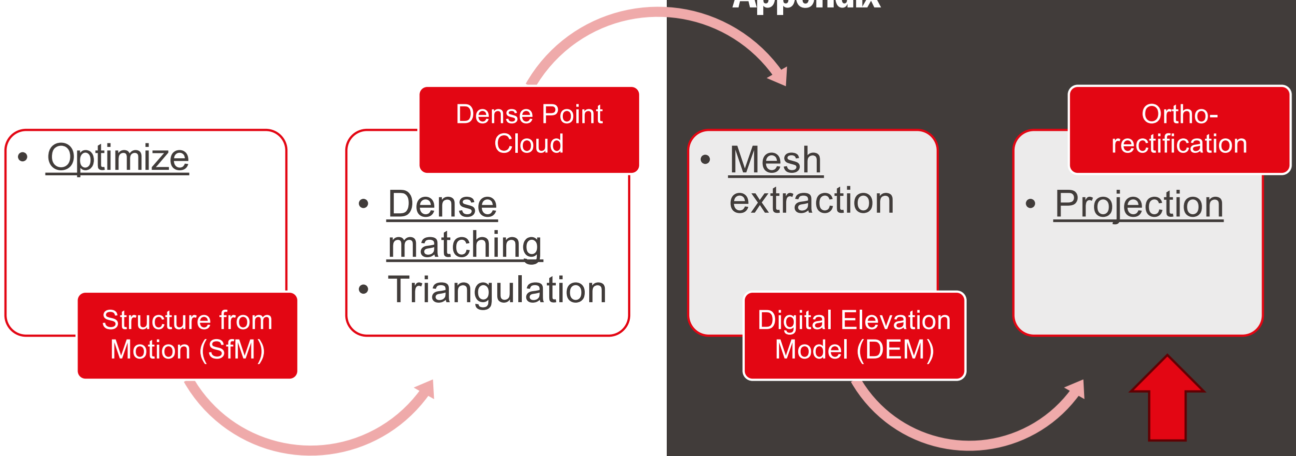
Optimization

Dense matching / 3D point cloud

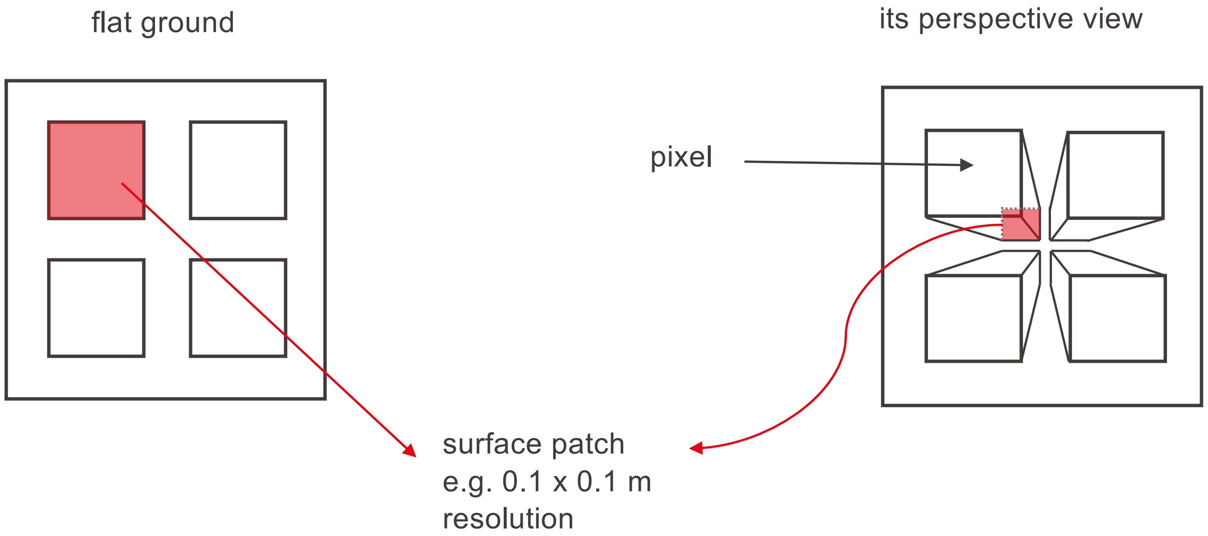
3D meshes and elevation models

Orthorectification

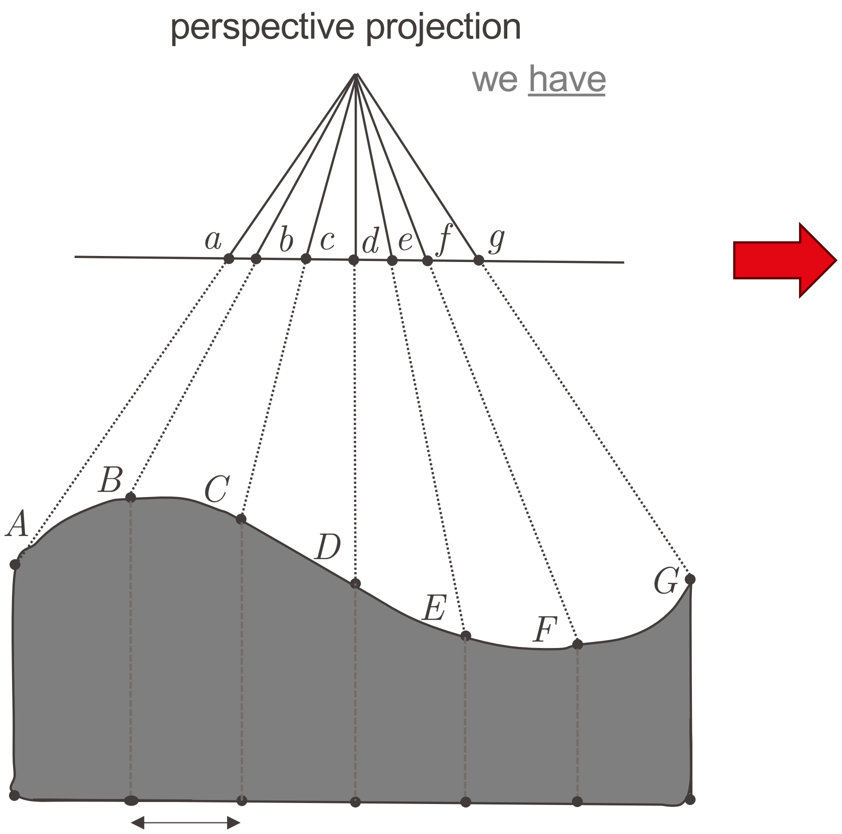
Appendix

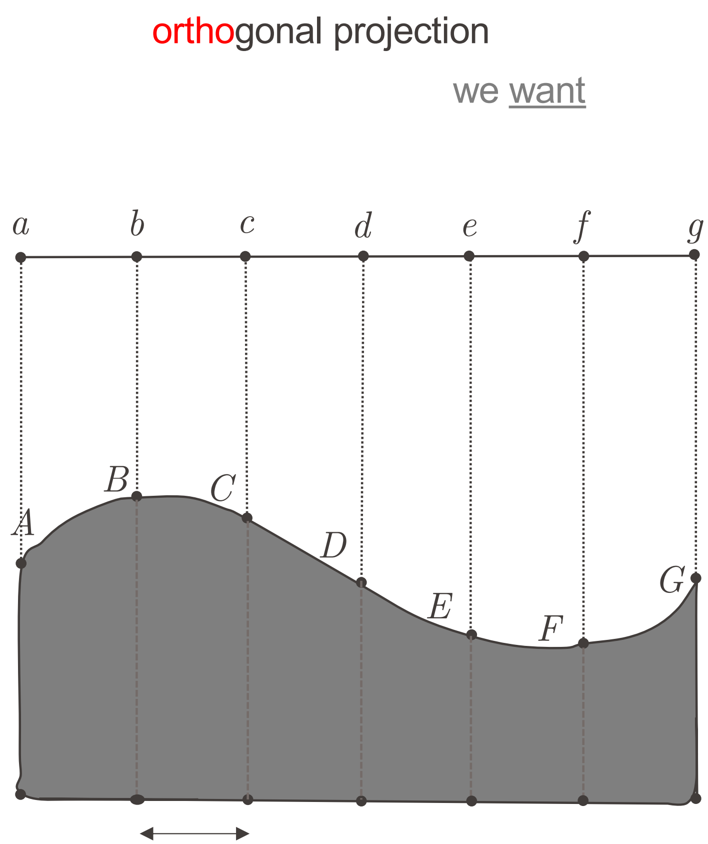


Why ortho-rectification?



Why ortho-rectification?

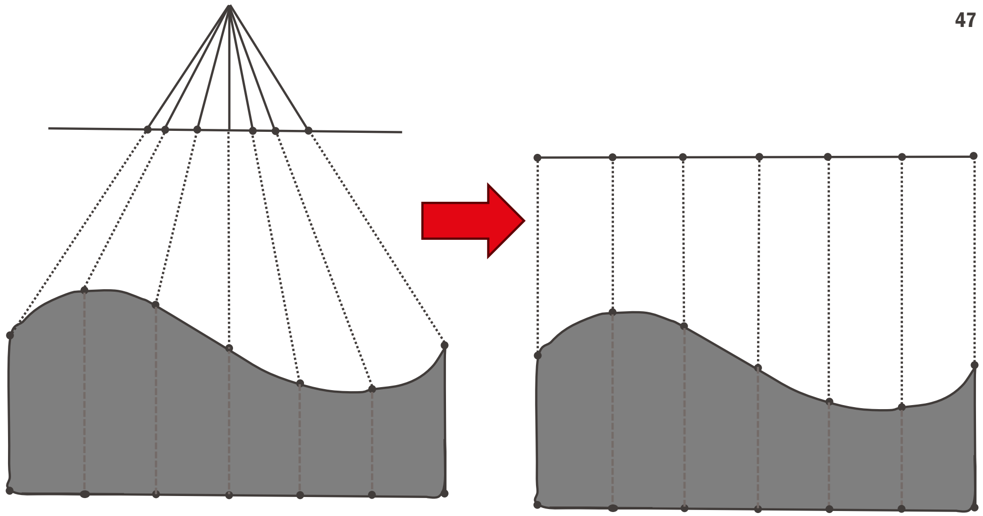




We aim to have an equidistant resolution …

… at a chosen projection surface (map)!

Ortho-rectification



Perspective projection

- Causes displacement of reality (radialy) per each pixel (also due to relief).
- Does not account for Earth curvature.

Orthographic projection

- Removes this displacement and produces planimetric photo from which features can be mapped.

Prerequisites

- *Orientation* (external & internal) known.
- Existance of a digital elevation model *DEM* – external or via photo-derived point cloud. (resolution, accuracy, format)

Ortho-photo

Is an image where each point is derived from an *orthogonal projection* on a *planar* surface.

Characteristics

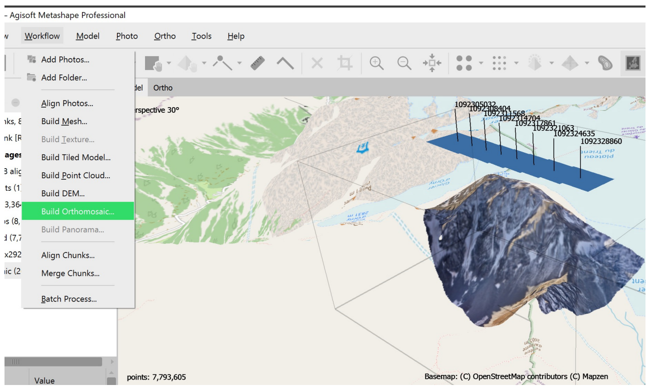
- Constant **scale** (as in a map)
- Inferring on **distances (or angles)** is possible
- Overlay of the photo on **a map** is possible
- ~~Stereo-viewing~~ is NOT possible

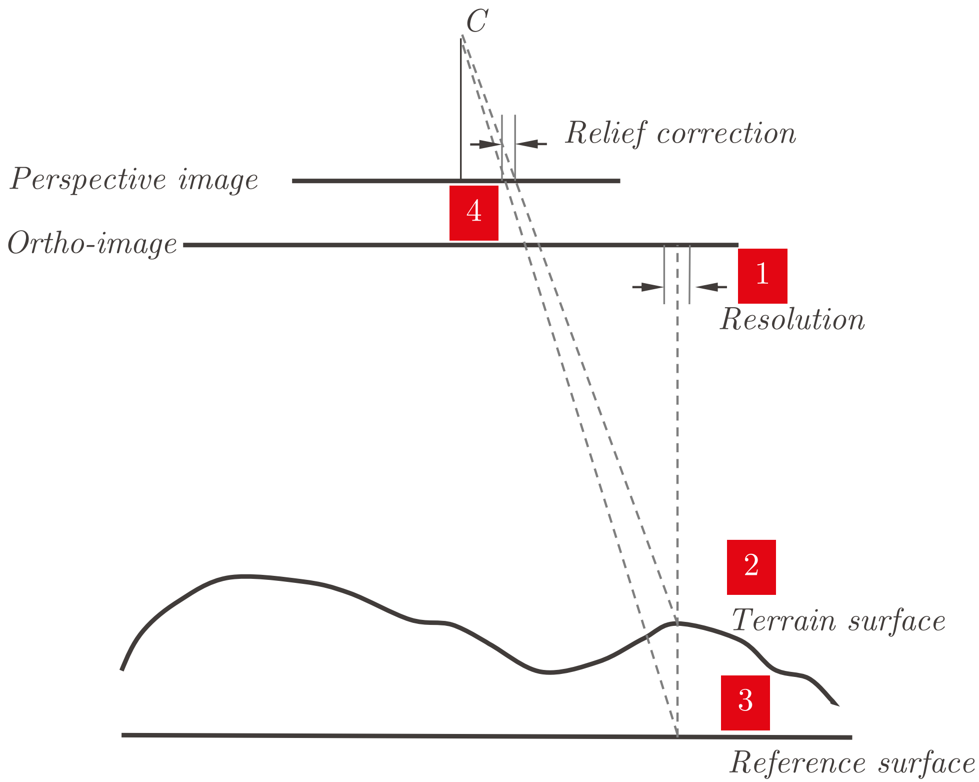
Process of ortho-rectification

For each $X,\ Y$

1. on ortho-photo (grid) coordinates
2. Project to DEM
3. Project to image coordinates (x,y)
4. Correction due to terrain height
5. Interpolate (radiometrically) from surrounding pixels (and photos)

Agisoft – Build Orthomosaic





Process of ortho-rectification

Characteristics per stage

A. Re-projection of pixel coordinates is uneven.

B. New RGB values obtained via interpolation (e.g. bilinear, bi-cubic, nearest-neighbour).

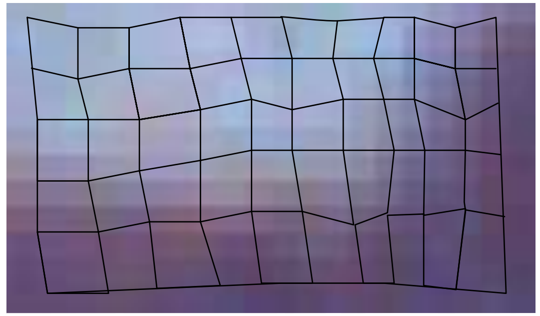
C. Position is redressed on the grid of resolution at a <u>reference height</u> (GSD - ground sampling density):

Characteristics – general

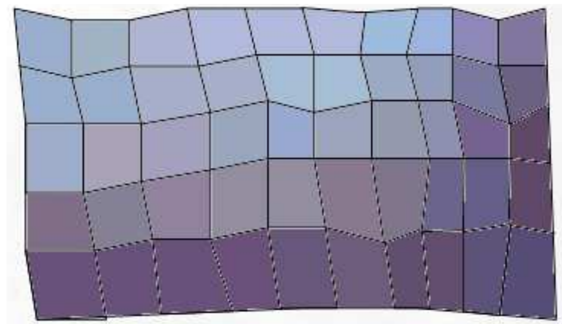
- The distance between points on orthophoto = (accounting for photo scale)

= the horizontal distance between the corresponding points on the ground.

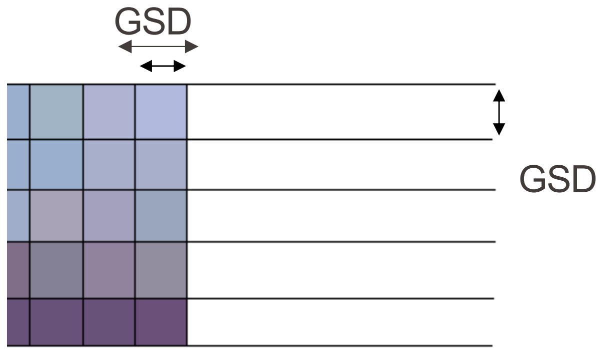
A. x,y



B. RGB



C.

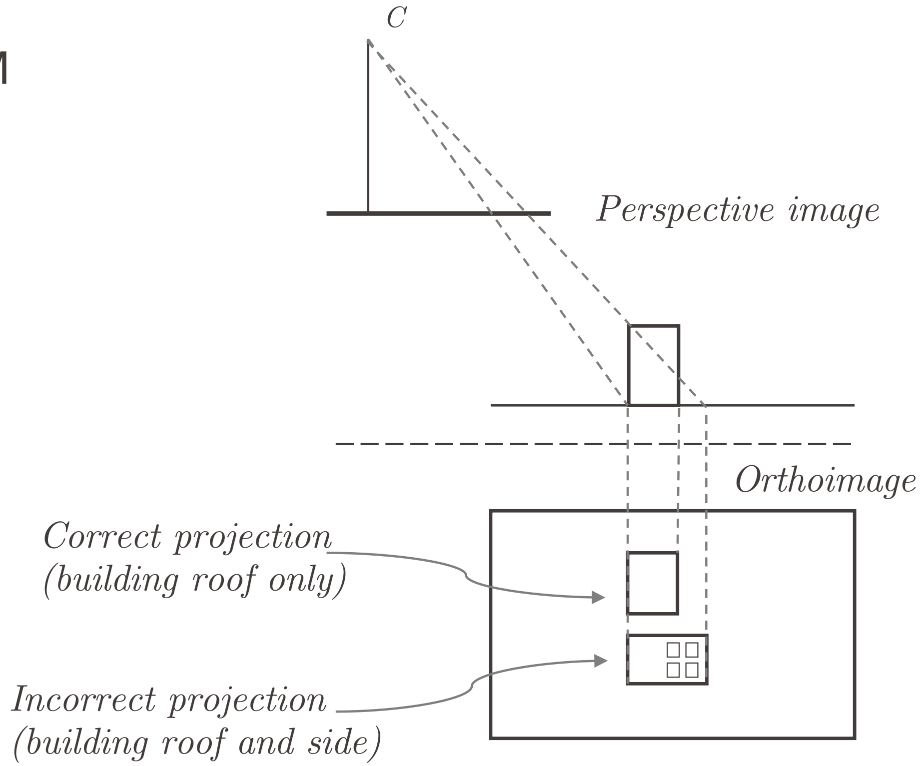


Ortho-rectification in cities …

Building correction

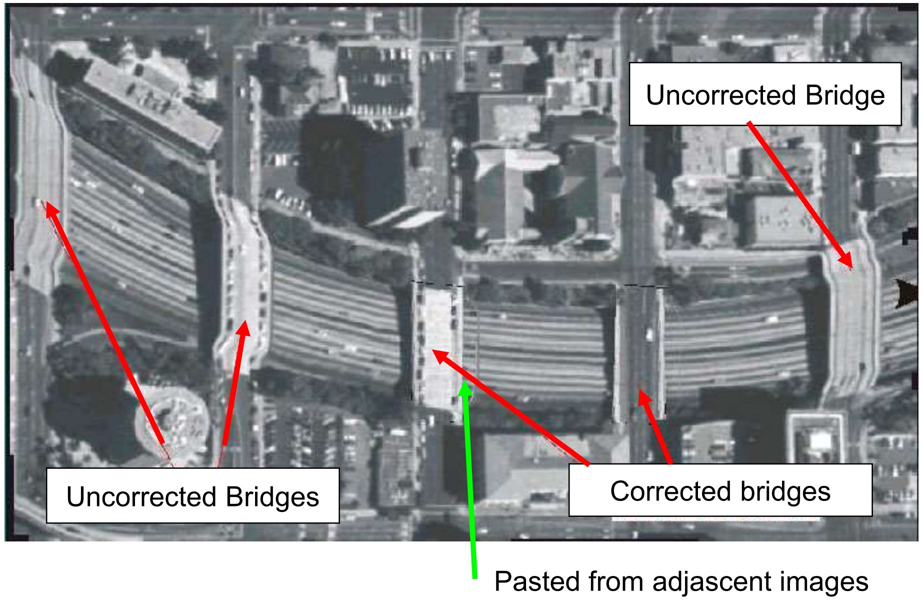
- If buildings are not included in the DEM (normal case of DTM), they are not orthorectified and remain "leaning"
- Use of <u>DSM is recommended</u>

Same building in different perspective projections



Ortho-rectification in cities ...

Idem for 'bridge correction' …



True ortho-photo

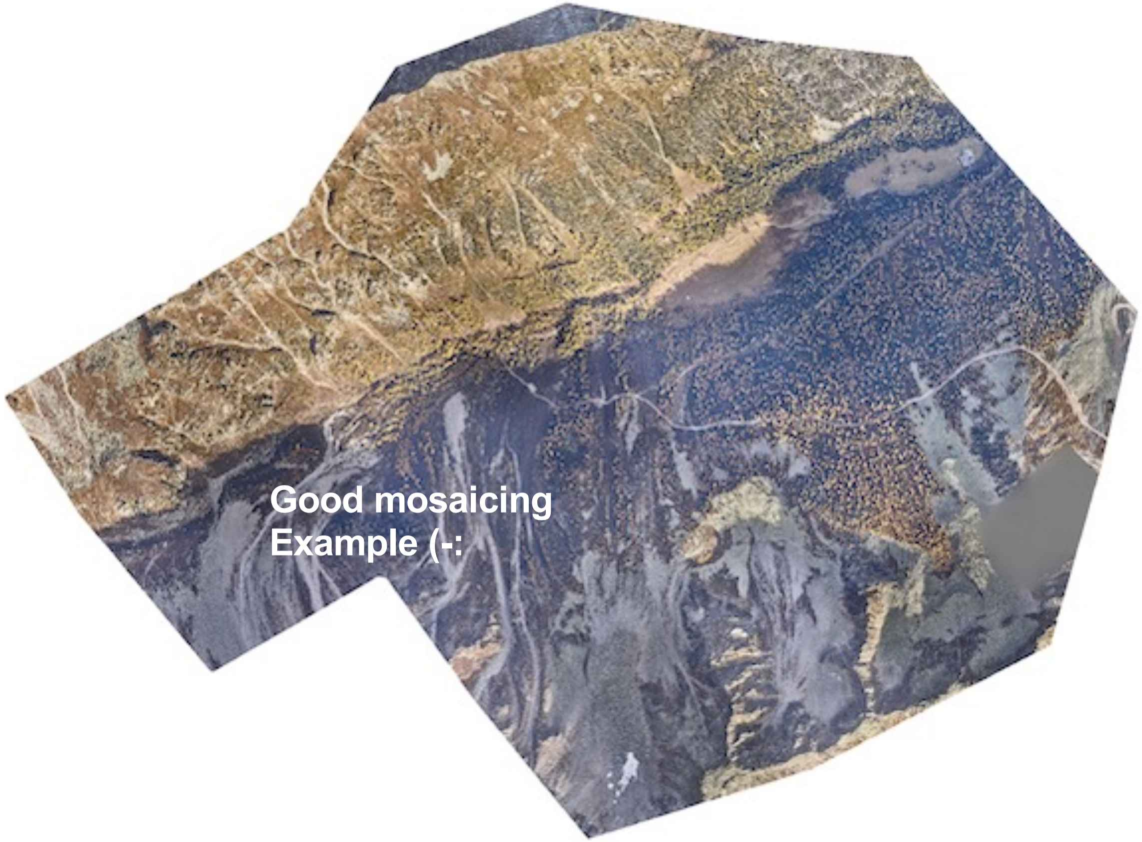
- Detailed DSM
- All buildings & bridges

Challenges

- Visibility afects DSM quality

  (Appendix IV: image vs lidar)
- Along-track 80% …
- Cross-track 80% …
- Productivity!

Orthophoto – mosaicing



Goal

- From image collection – all geometric *distortions corrected,* and imagery *colour balanced* to produce seamless image

Pixel

- can have several representations – choice!

Strategy

- radiometric differences are less visible on borders: seamlines or cutlines
- Urban: geometry (building unity)
- Rural: radiometry (field unity)

Algorithm

- Graph-cut [1]

- [1] https://www.sciencedirect.com/science/article/pii/S0924271615002774, http://web.eecs.umich.edu/~pettie/matching/Edmonds-Karp-network-flow.pdf

Understanding - self assessment

- What is main idea of “incremental” structure from motion (SfM)?
- Why is ”incremental” SfM performed in steps?
- What are the pre-requisites before global optimization can be performed?
- What is the objective function to minimize in the global optimization?
- How are random errors influencing feature coordinates?
- What are the possible approaches to mitigate the influence of random errors on points?
- What a should and optimization quality report contain?
- What should the criteria of acceptance be?
- What type of analysis could be done prior to a mission to determine the chance that its objectives can be fulfilled? (What are the influencing factors?)

Understanding - self assessment

- What are the processing stages for creating 3D point clouds from image pixels?
- What is necessary to known before dense-matching can be started and how the pixel-to-pixel correspondence search can be simplified?
- What are the main challenges in performing dense-matching for 3D point cloud and which can be overcome by using a different technology?
- How DTM is created from a 3D point cloud?
- What are the differences between DSM and DTM?
- What are the respective pros & cons for DEM in raster and TIN representations?
- What is the ortho-rectification of images and what is needed for its creation?
- What is needed that building's facades are not visible on an ortho-photo?
- What is the true orthophoto?

EPFL

Optimization

Dense matching / 3D point cloud

3D meshes and elevation models

Orthorectification

Appendix I. planning exercise 4 exam!

II. optimization factors

III. epipolar image generation

IV. image vs. lidar point-cloud