3D BAG

GEO1004: 3D modelling of the built environment

Ravi Peters

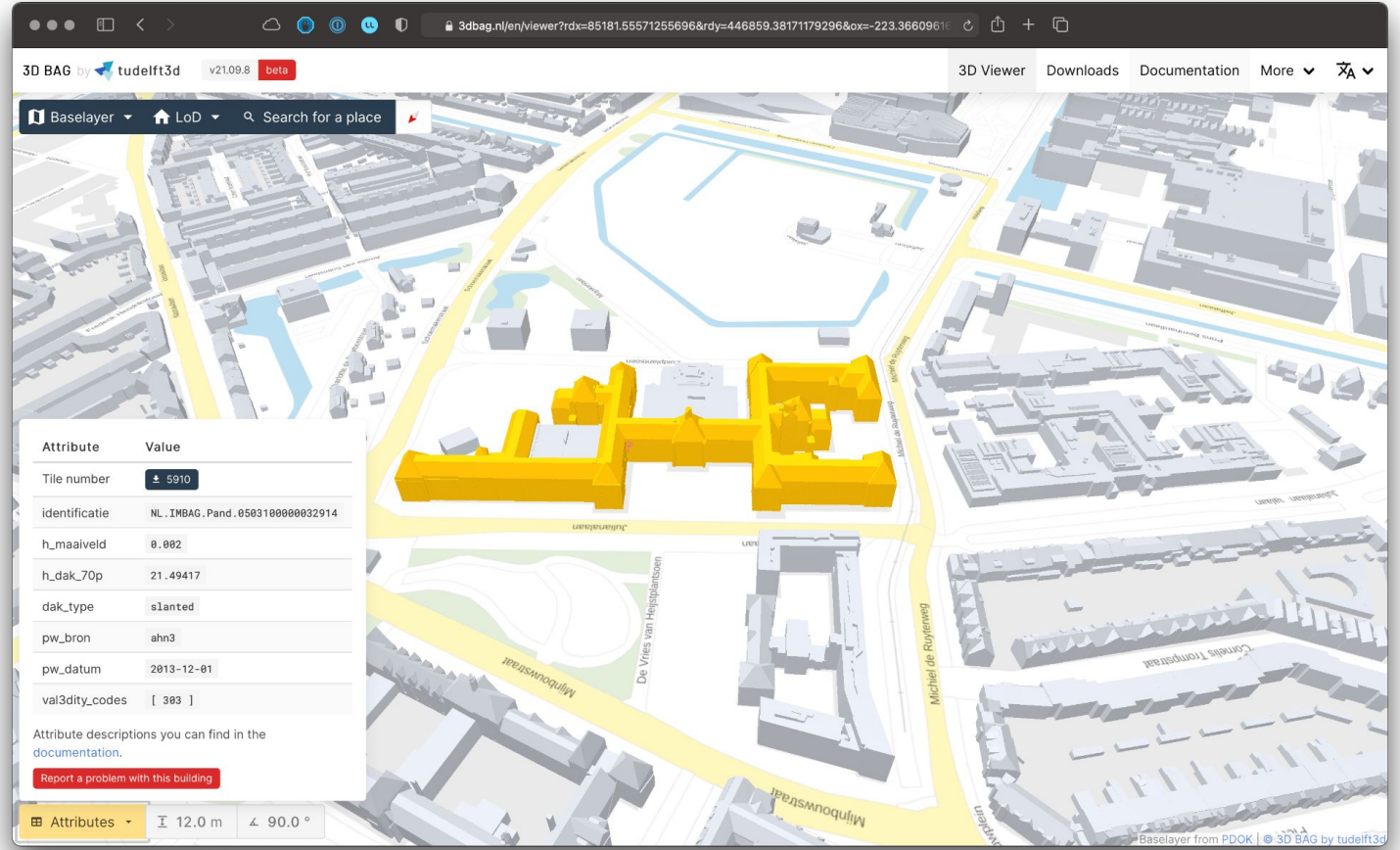
02-06-2025

3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

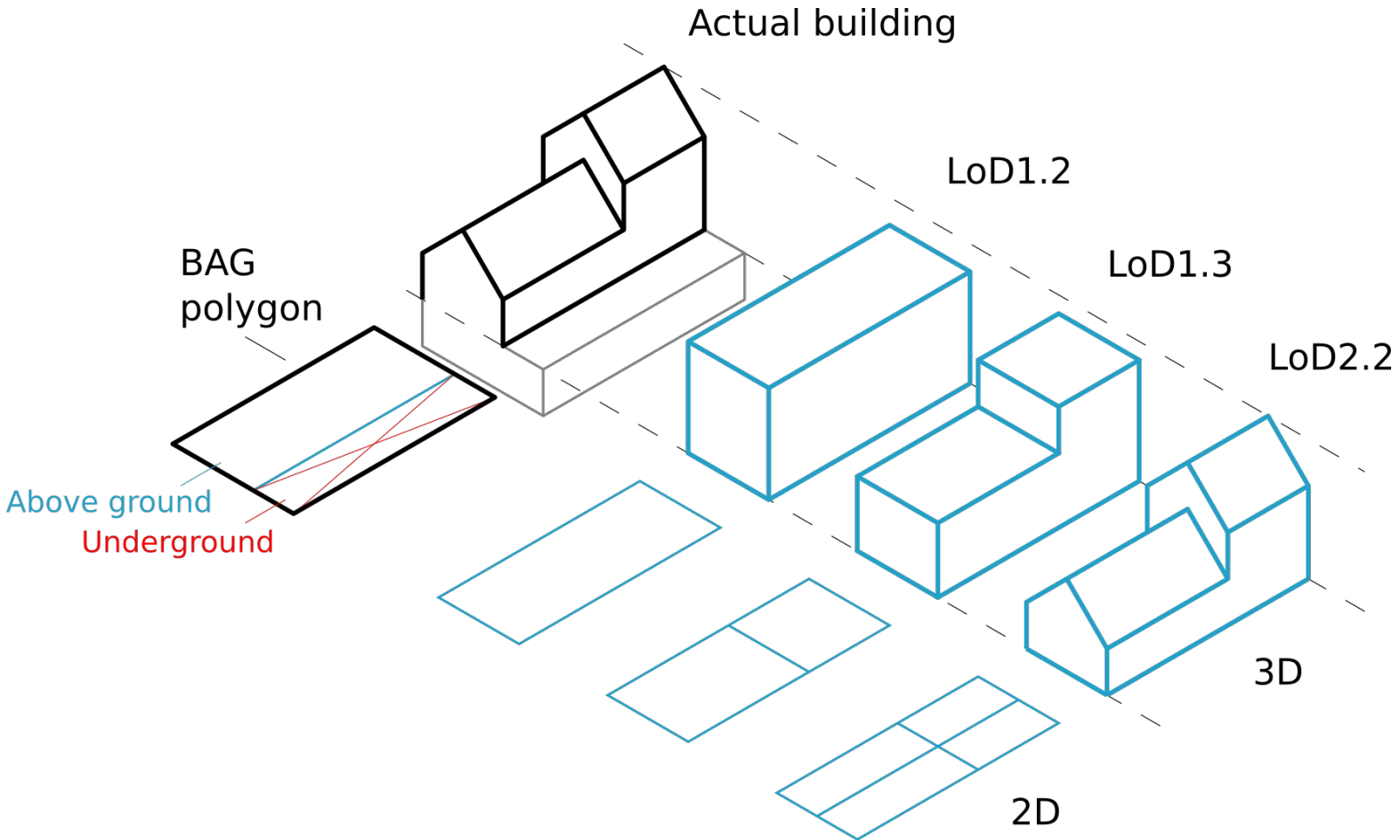
The 3D BAG

3dbag.nl

Data:

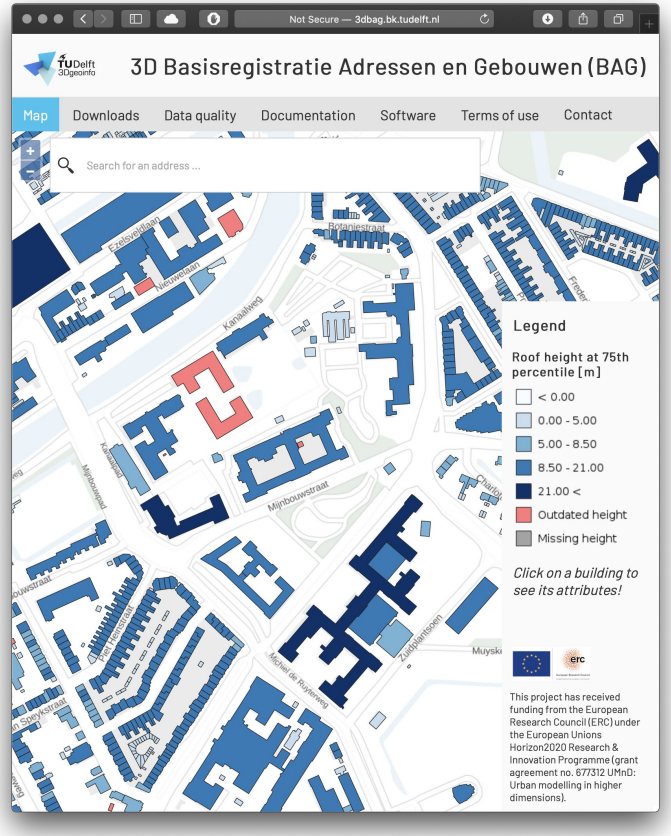


3D BAG layers: 3 LoDs

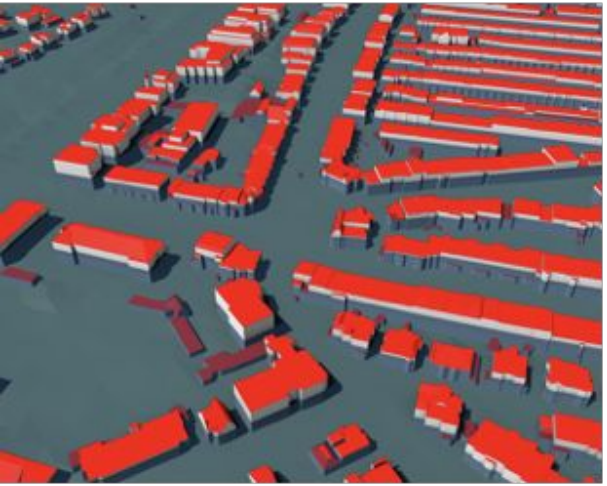


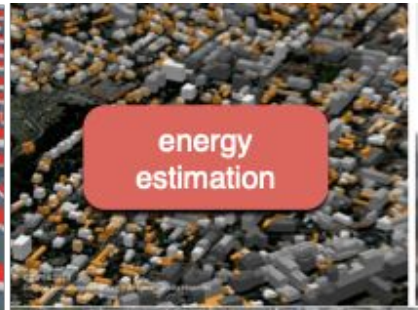
A bit of background...

- Developed in 3D geoinformation group
  - To offer reliable & uniform height data for buildings
- Started as simple height attributes
  - Only LoD1.2
  - Used by practitioners, much feedback
- Co-developments within several research projects
  - Initial request for LoD1.3 models for Noise simulation NL
- First version of 3DBAG with LoD2.2 in 2021

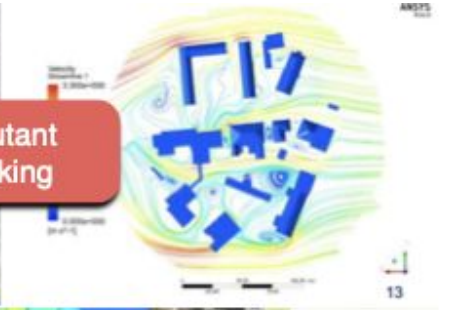


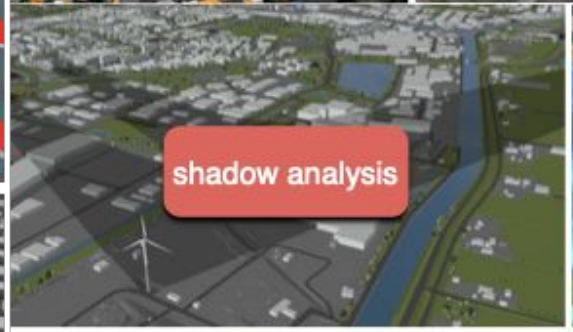
3DBAG 2018

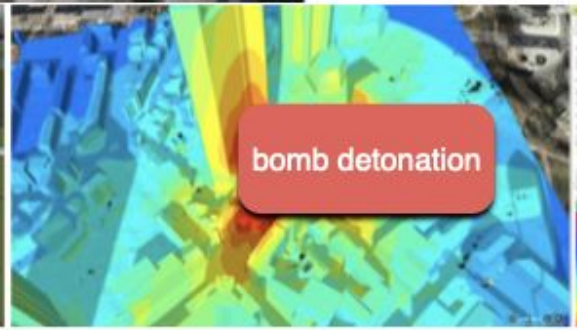


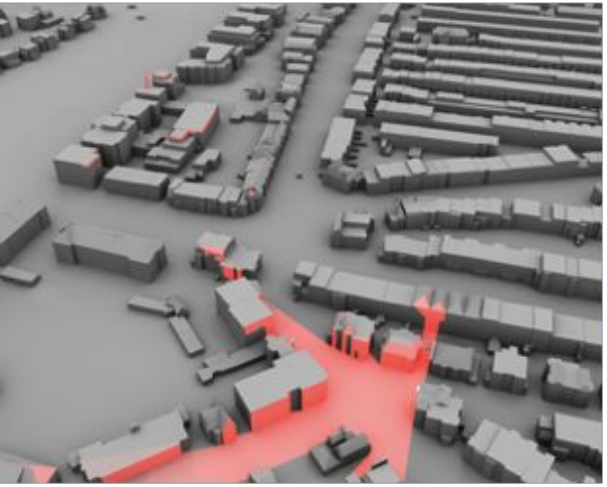

energy
estimation

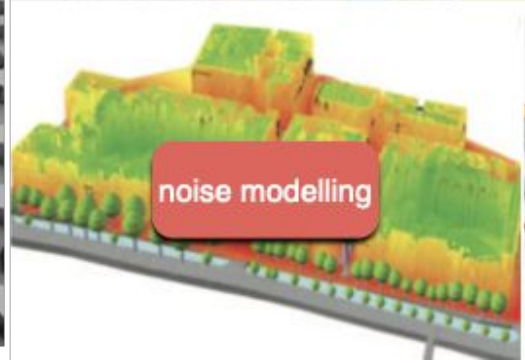

pollutant
tracking


13

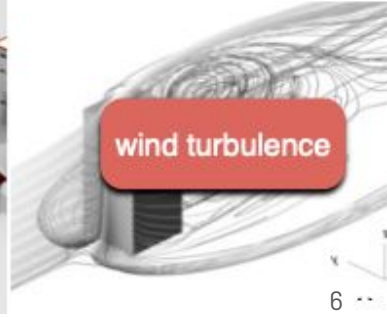

shadow analysis


bomb detonation




noise modelling


solar potential

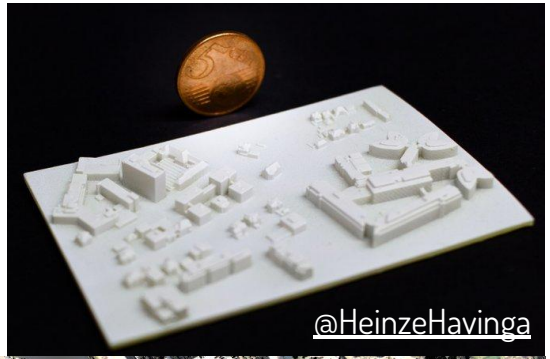

wind turbulence

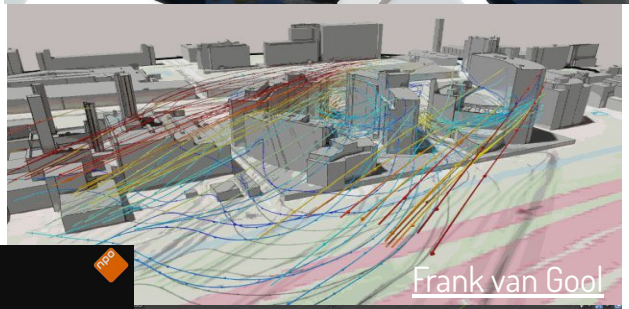

Heating energy demand
Computational Fluid Dynamics
Solar panel potential

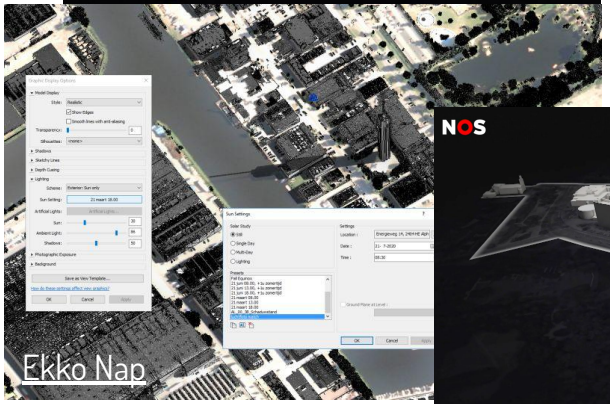

Heating energy demand
Computational Fluid Dynamics
Solar panel potential

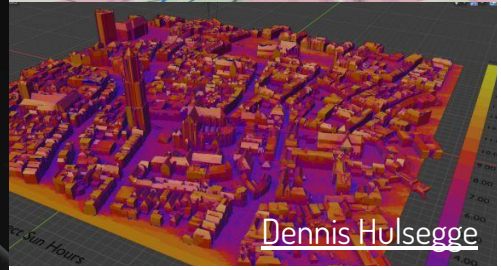
3D BAG in practice

https://docs.3dbag.nl/en/overview/media/





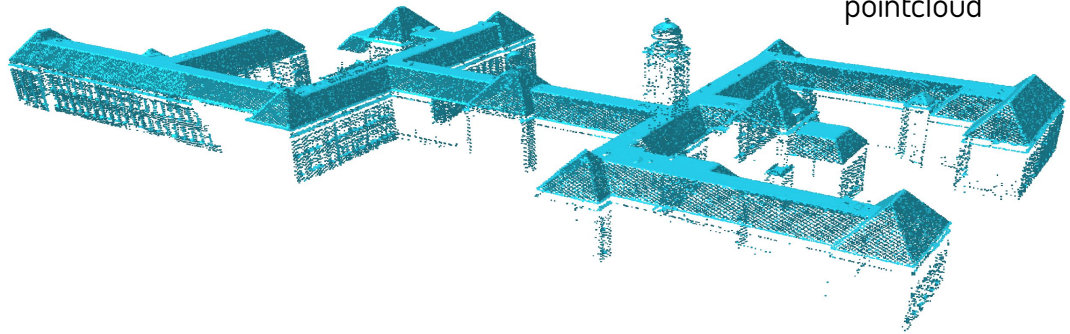




Reconstruction algorithm

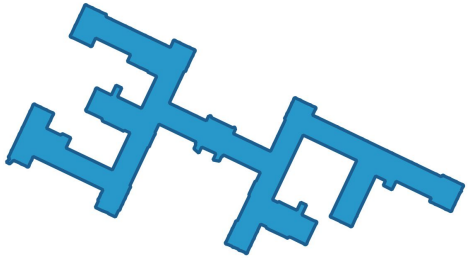
(https://github.com/3DBAG/roofer)

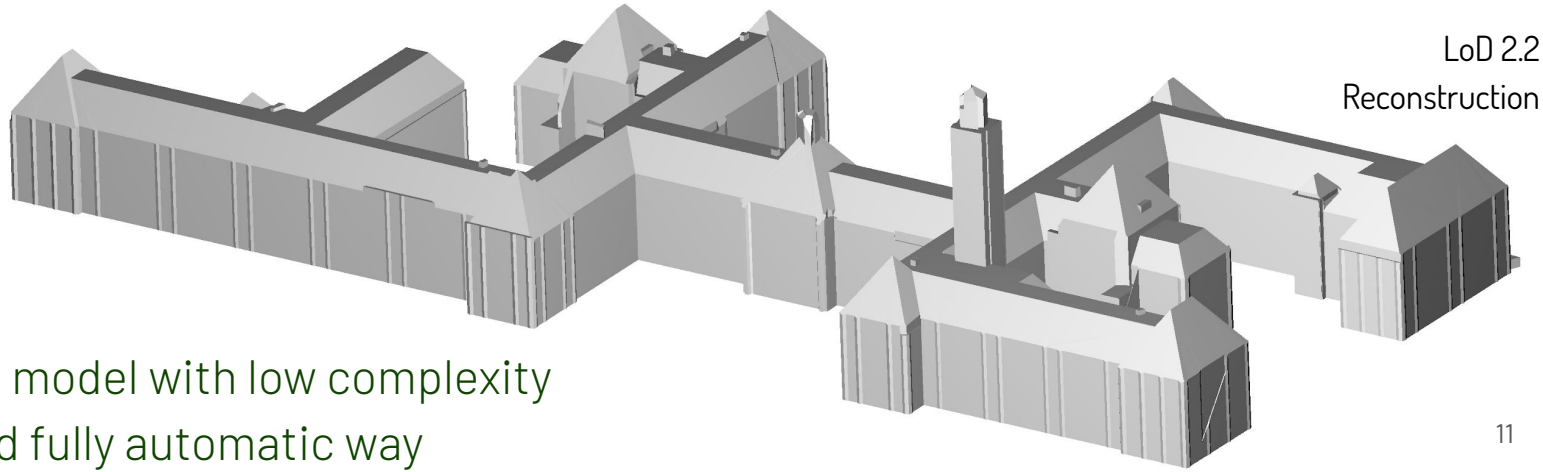
Building reconstruction



pointcloud

footprint polygon





LoD 2.2
Reconstruction

- Accurate 3D model with low complexity
- In robust and fully automatic way

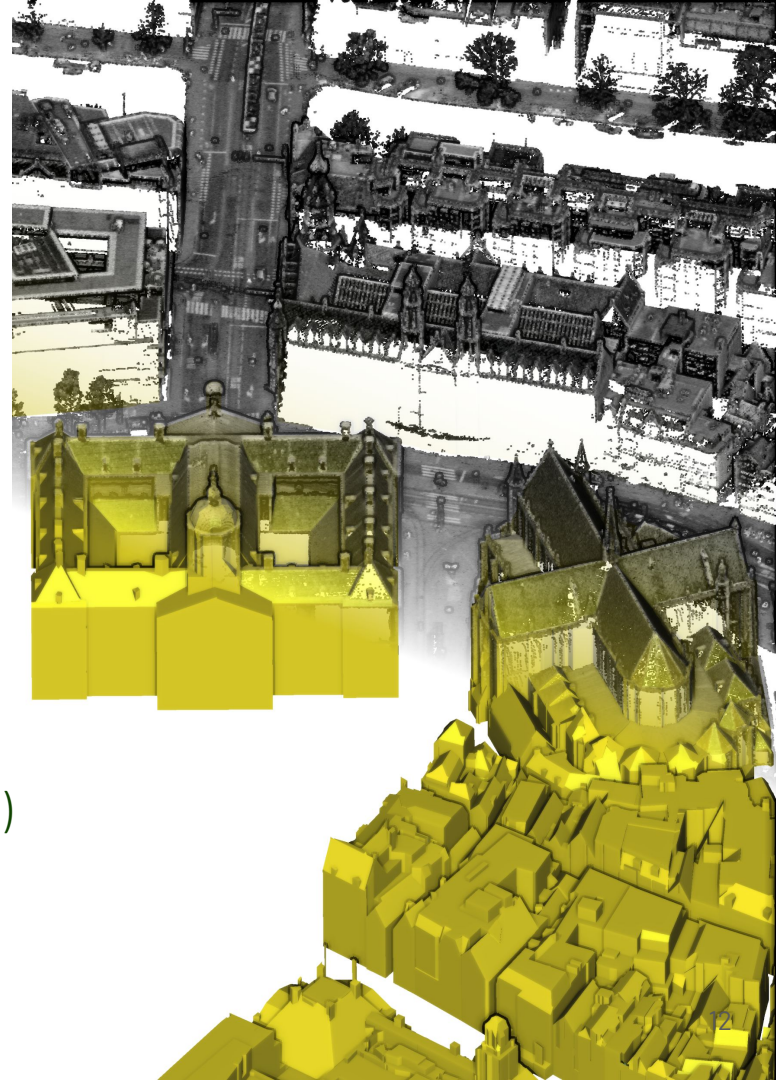
Open data in the Netherlands

BAG https://www.kadaster.nl/bag

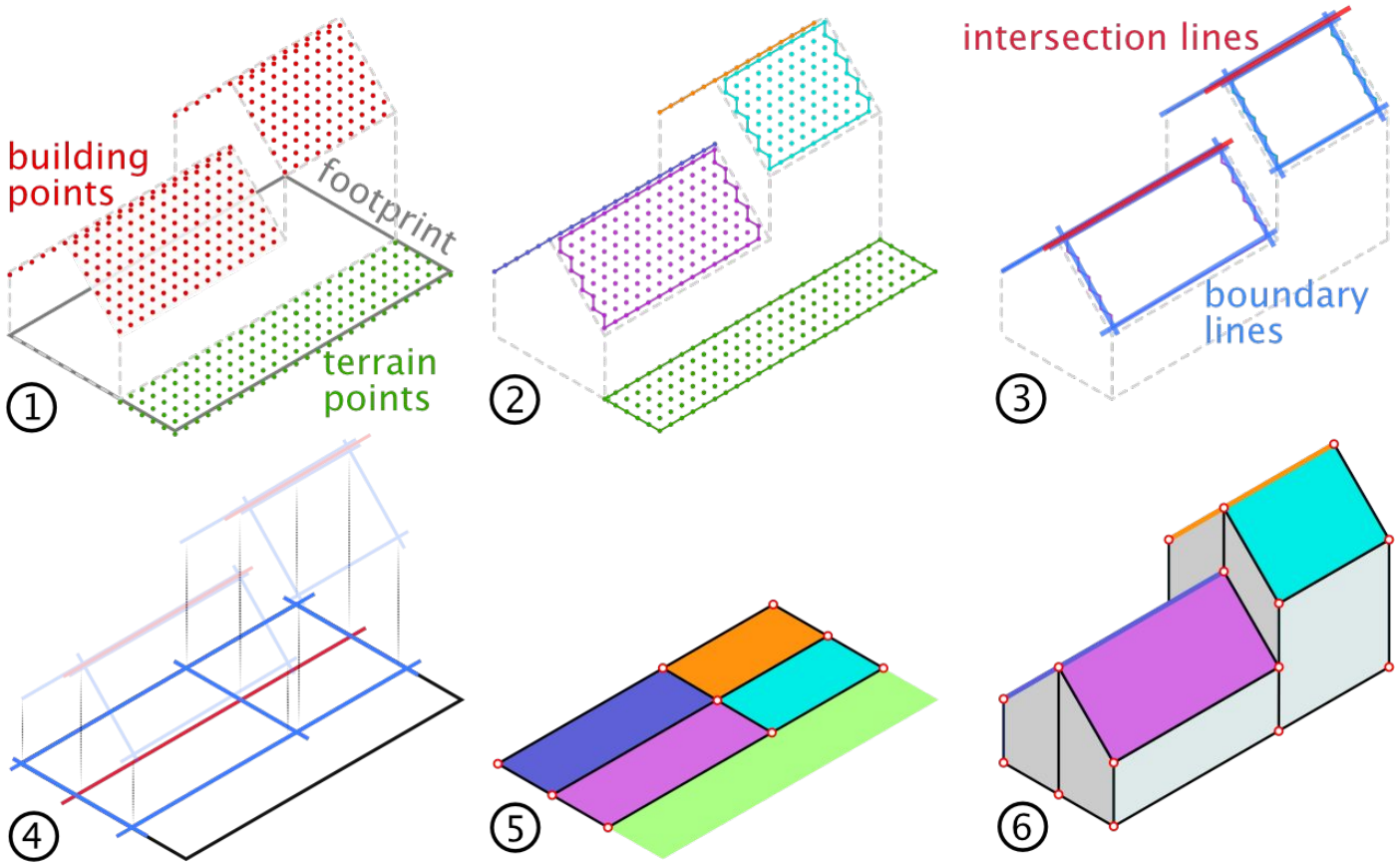
up-to-date building polygons + attributes
largest extent → roofprint + underground
positional accuracy 30cm

AHN https://ahn.nl

classified point cloud
positional accuracy ~23cm (height and planimetric)
8-15 pts/m2 for buildings
occlusion and other no-data areas

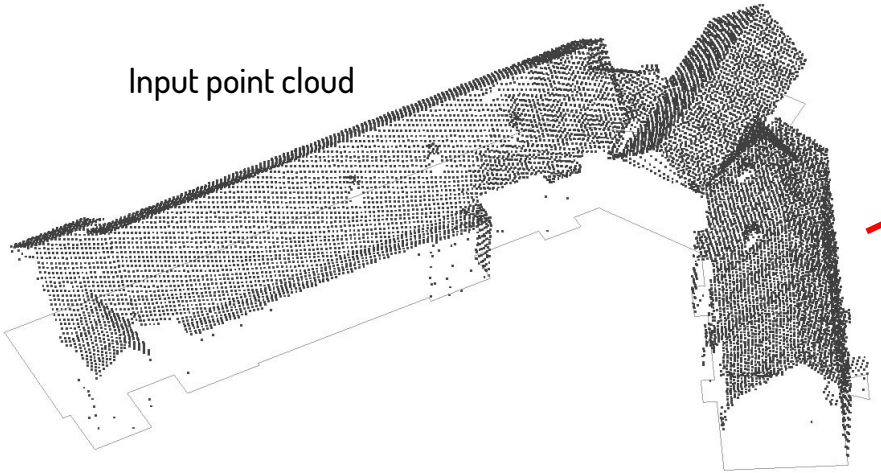


Overview building reconstruction method

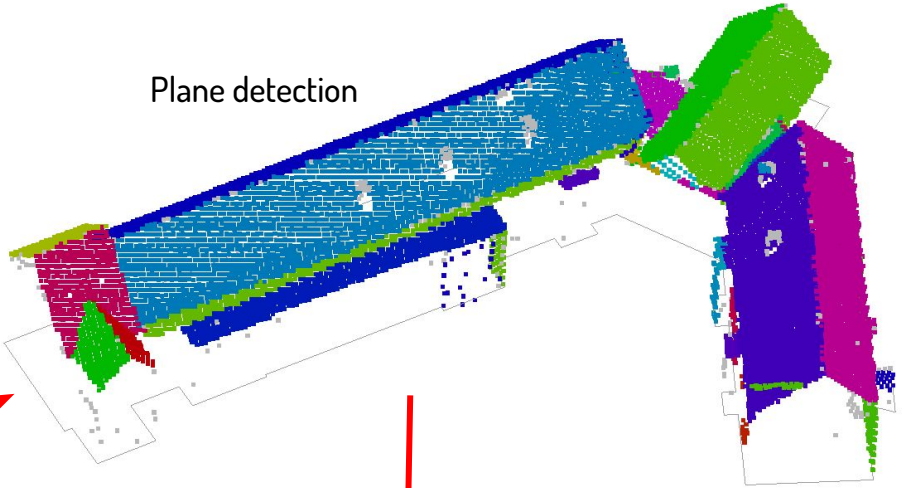


Feature extraction

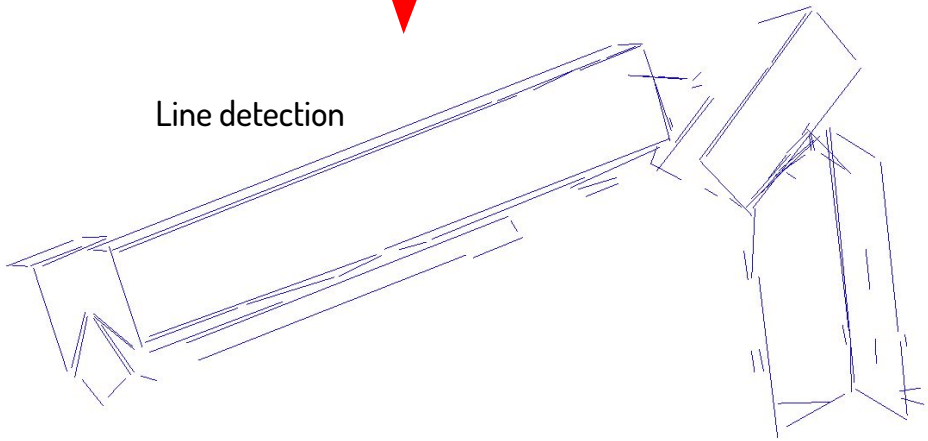
Input point cloud



Plane detection

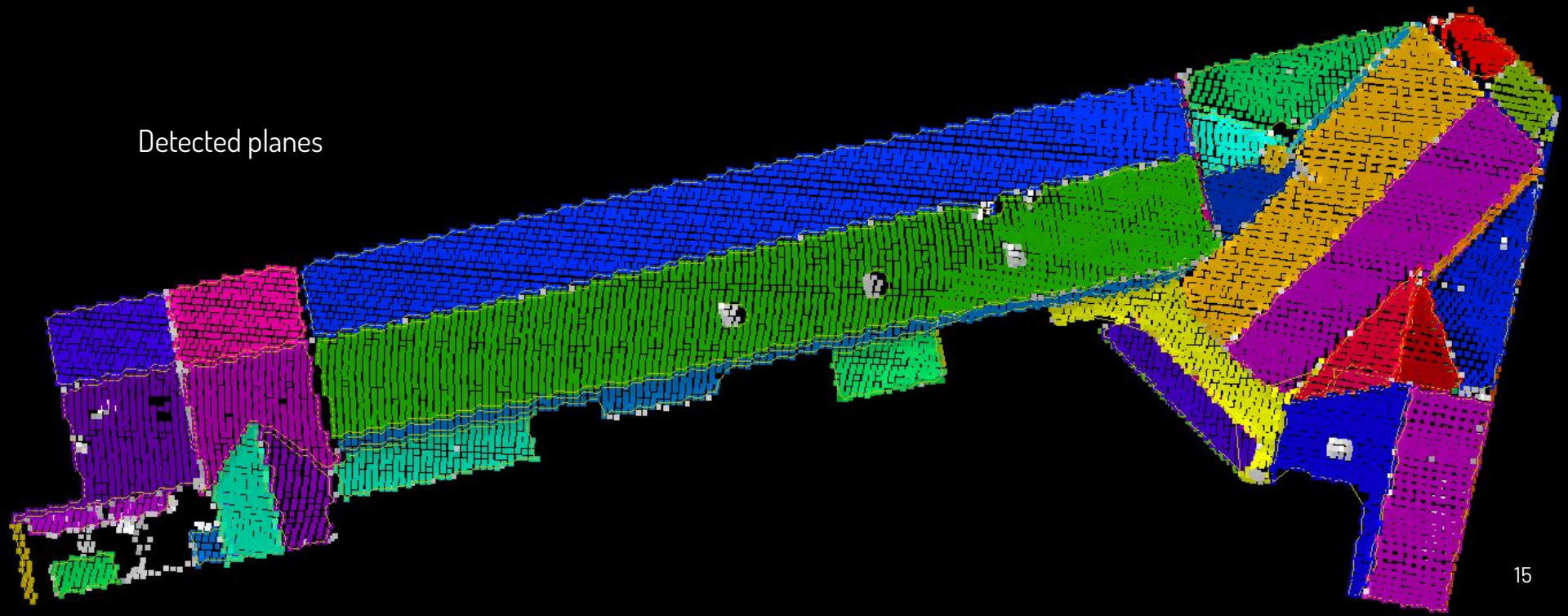


Line detection

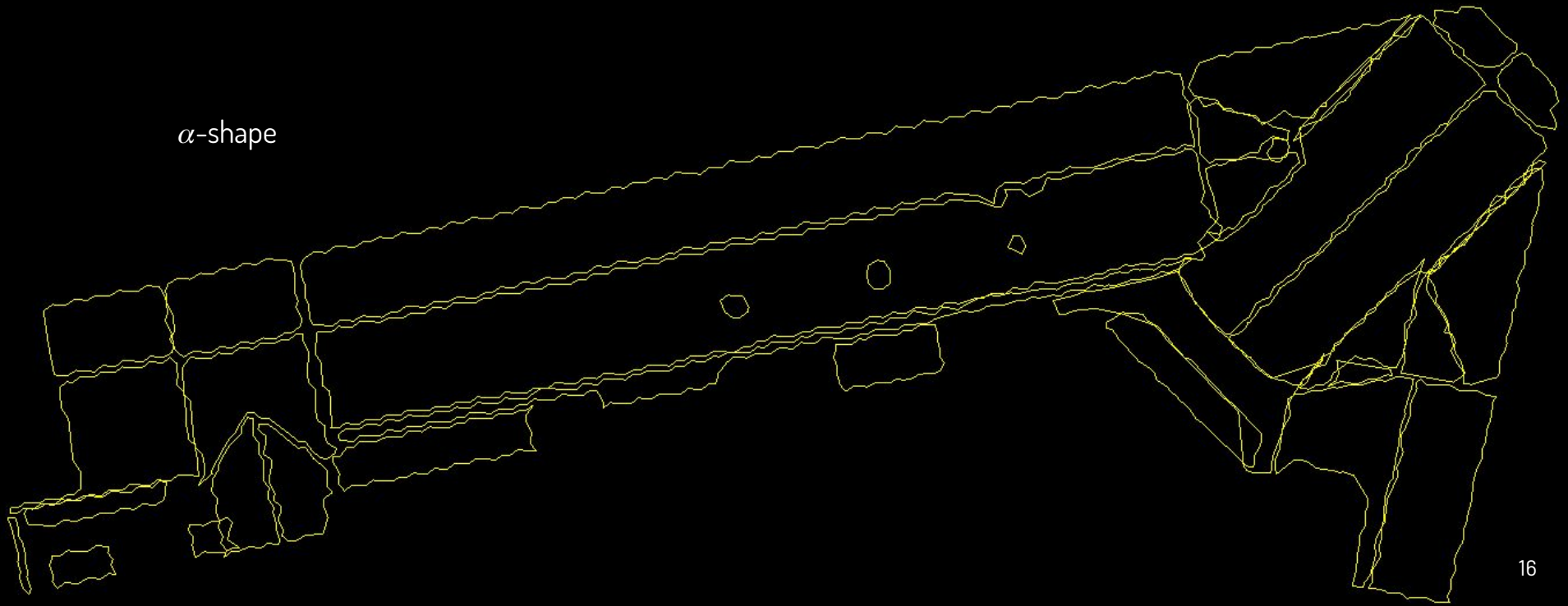


Feature extraction

Detected planes

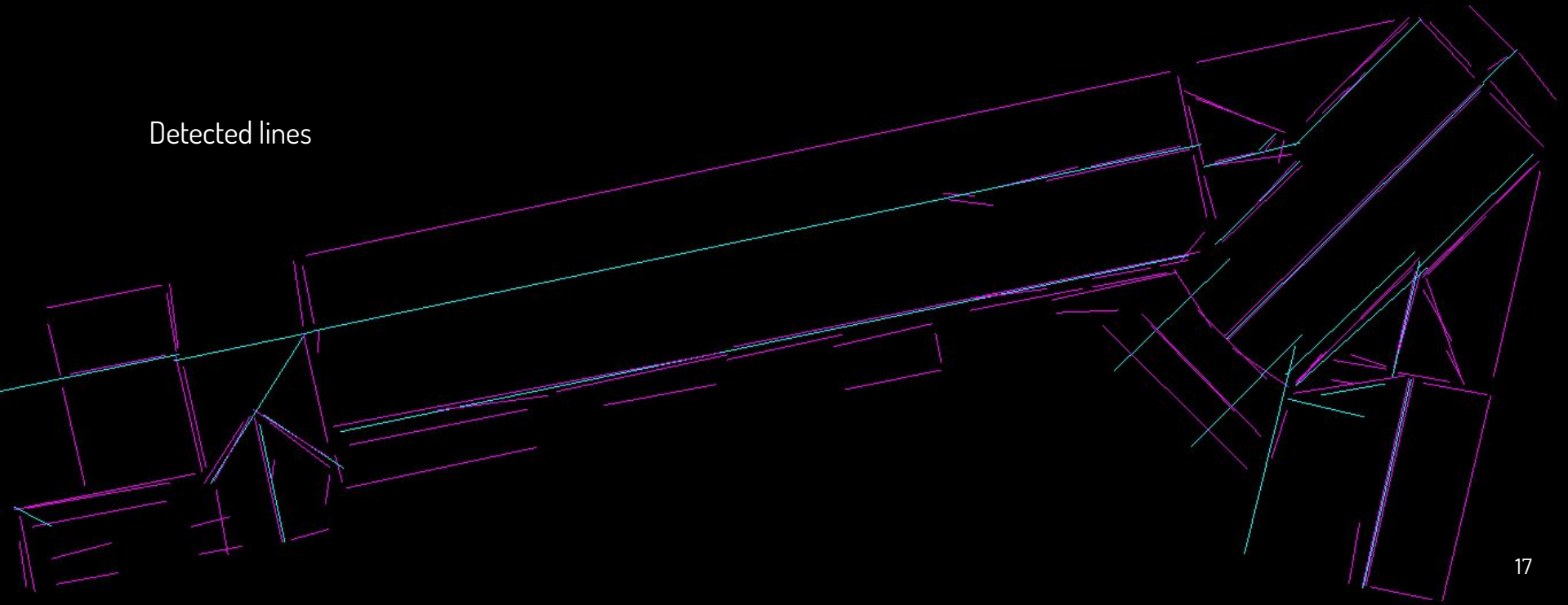


Feature extraction



Feature extraction

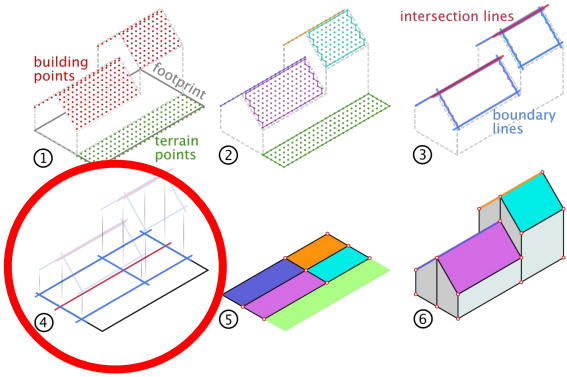
Detected lines

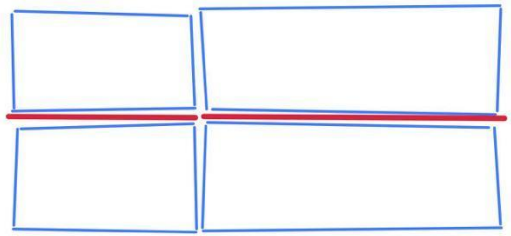


Line regularisation

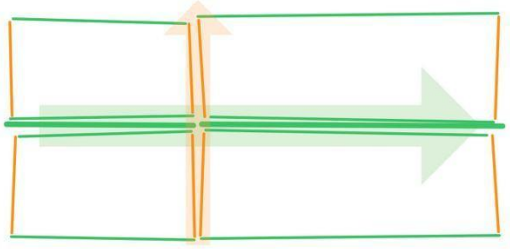
Using 2-step hierarchical clustering

1. Based on line orientation
2. Based on euclidean distance within orientation clusters

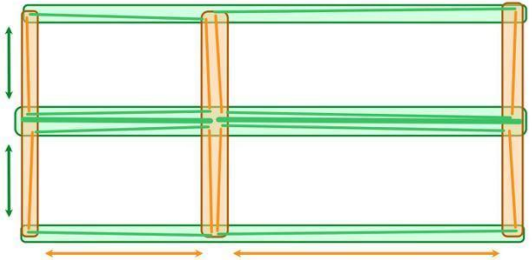




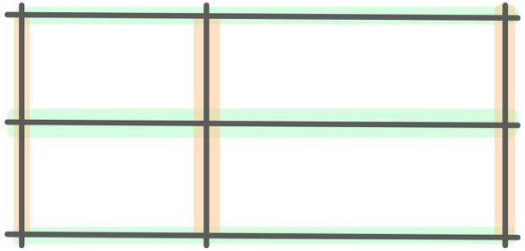
(a) Detected lines



(b) Orientation clustering



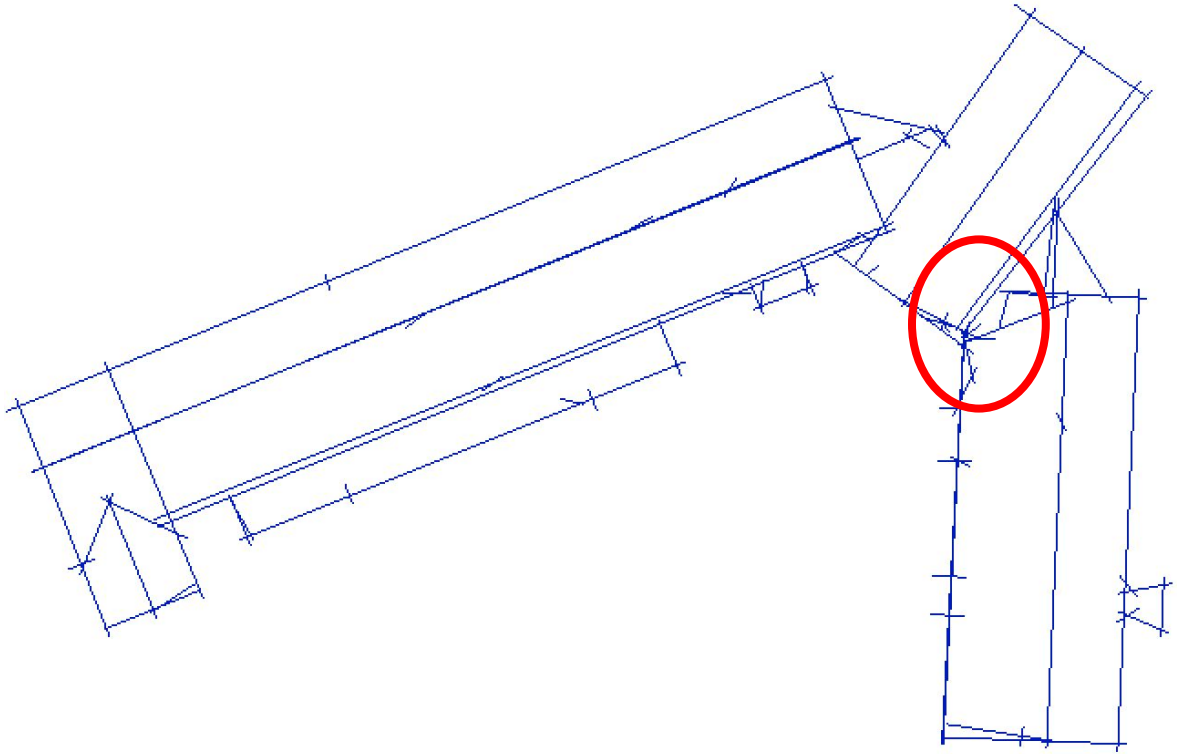
(c) Distance clustering

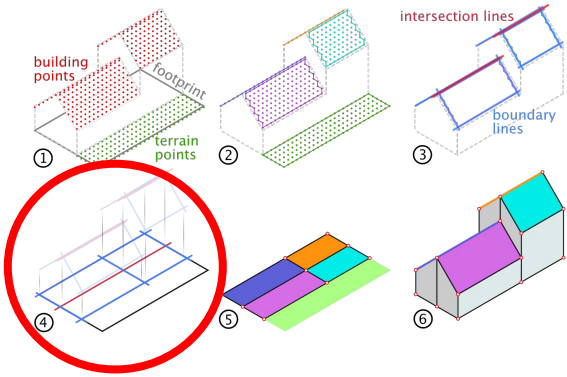


(d) Regularised lines

Initial roof partition

Still many small faces

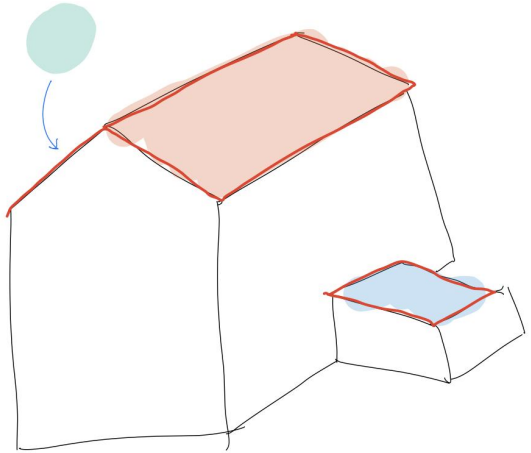




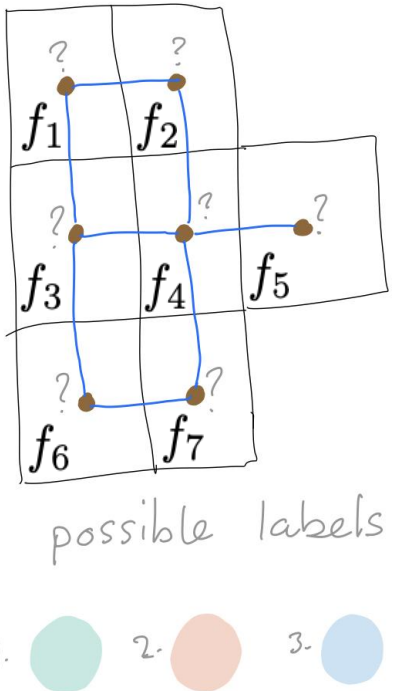
Graph-cut optimisation

Zebedin et al. (2008)

$$E(f) = \lambda \sum_{p \in P} D_p(f_p) + (1-\lambda) \sum_{\{p,q\} \in N} S_{p,q}(f_p, f_q)$$



Dual graph of planar arrangement

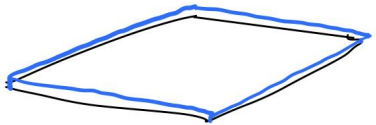


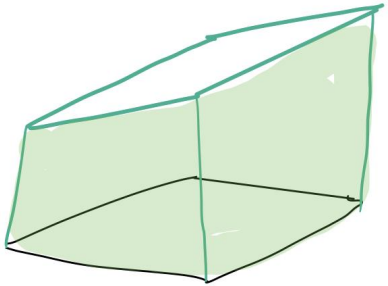
Graph-cut optimisation

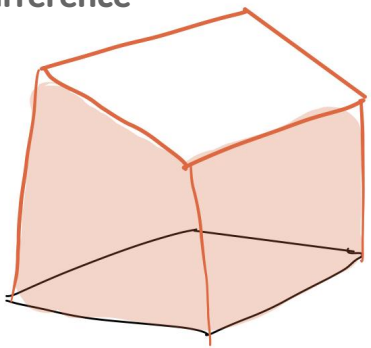
$$D_p(f_p) = \sum_{x \in p} \left| height_{\mathrm{PC}}(x_i) - height_{\mathrm{f_p}}(x_i) \right|$$

Data term:
Volume between 2.5D heightfield of point cloud at a face and candidate planes

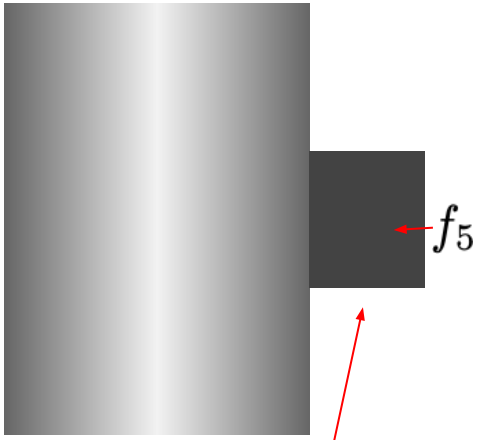
Encourages good data fit
by selecting candidate plane with smallest volumetric difference



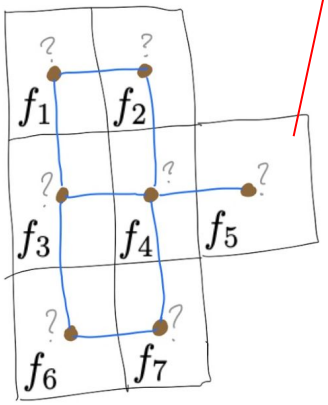




Volume wrt each candidate plane



heightfield



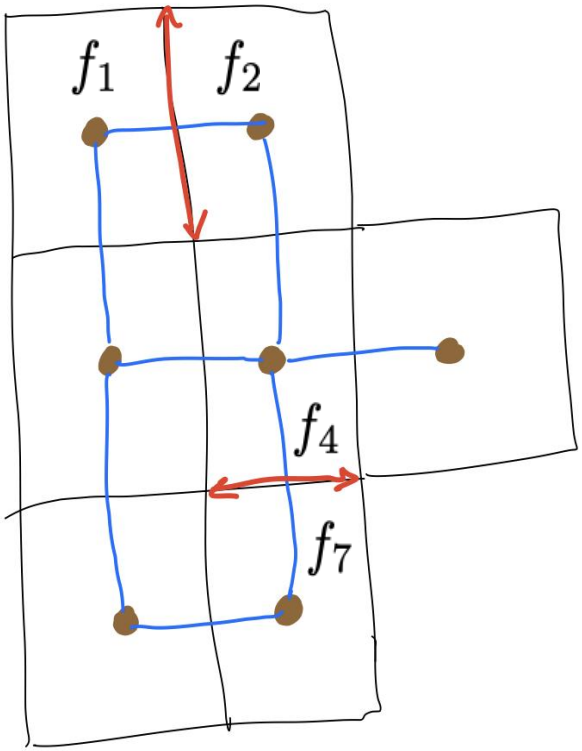
Graph-cut optimisation

$$S_{p,q}\left(f_p, f_q\right) = \begin{cases} length\left(f_p, f_q\right) & \text{if } f_p \neq f_q \\ 0 & \text{if } f_p = f_q \end{cases}$$

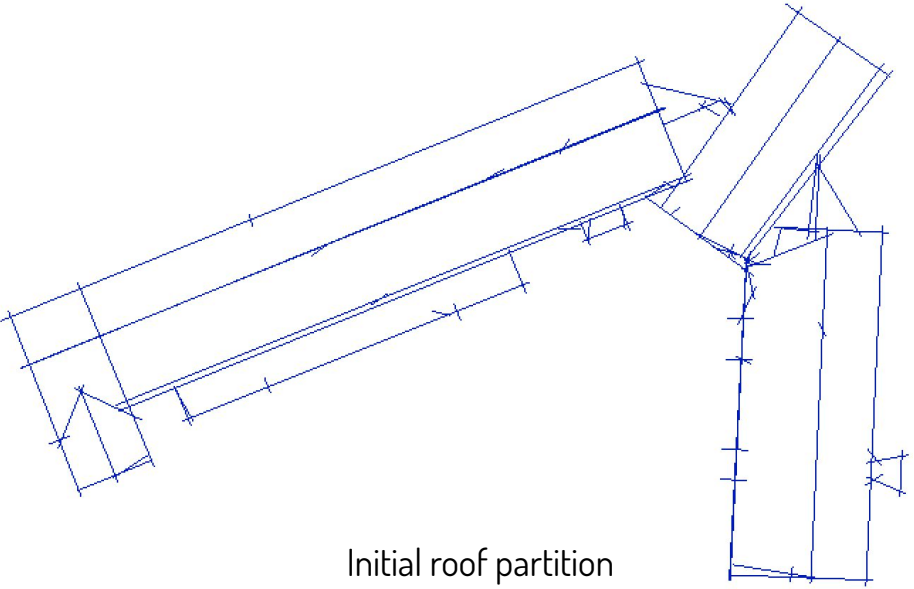
Smoothness term:
Edge length between adjacent faces

Minimisation entails:

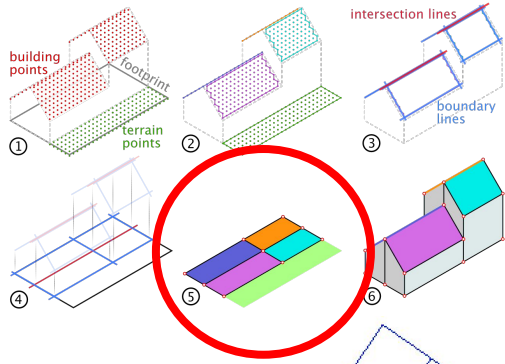
- **Short and simple edges**
- **'Merging' of faces,**

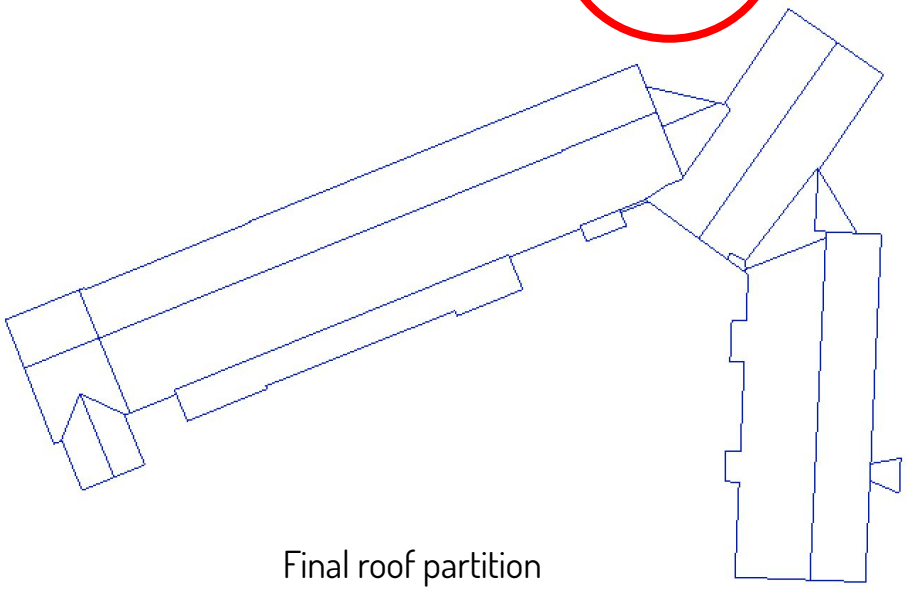


Final roof partition



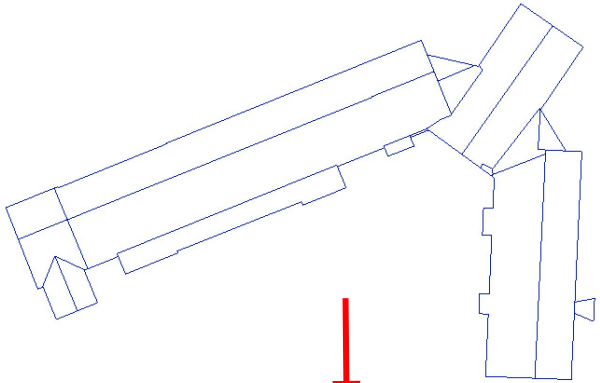
Initial roof partition

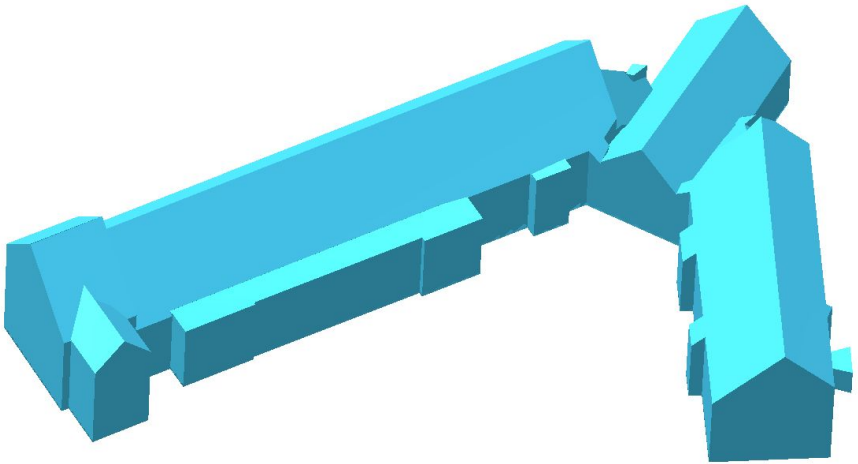


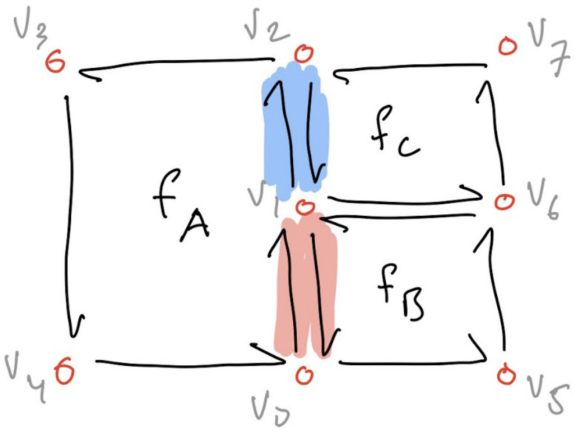


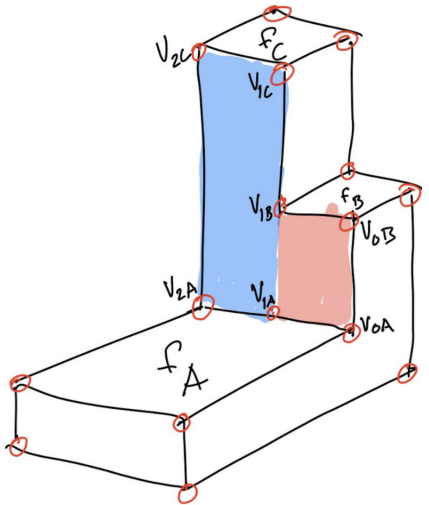
Final roof partition
(edges between equal plane labels dissolved)

Extrusion

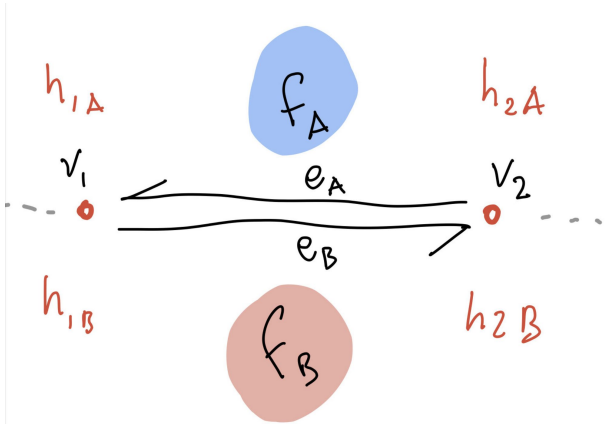


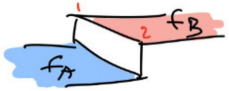
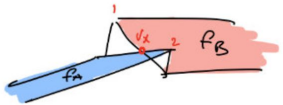
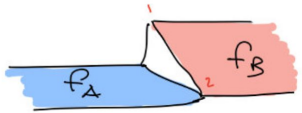






Extrusion



Case	condition	vertex order
	$h_{1A} < h_{1B}$ AND $h_{2A} < h_{2B}$	1. v_{1B}, v_{1A}, v_{2A}, v_{2B}
	$h_{1A} < h_{1B}$ AND $h_{2A} > h_{2B}$	1. v_{1B}, v_{1A}, v_X 2. v_{2A}, v_{2B}, v_X } 2 faces !
	$h_{1A} < h_{1B}$ AND $h_{2A} == h_{2B}$	1. v_{1B}, v_{1A}, v_{2A}

Effect of complexity term λ

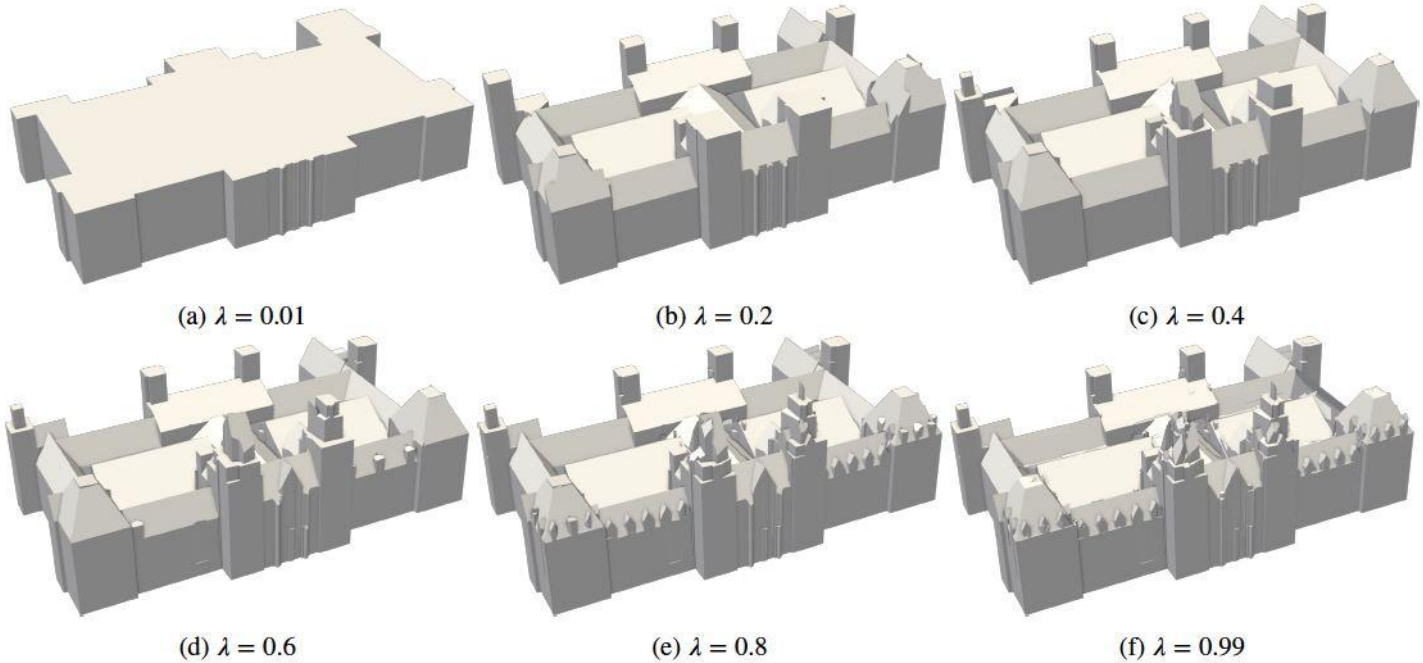


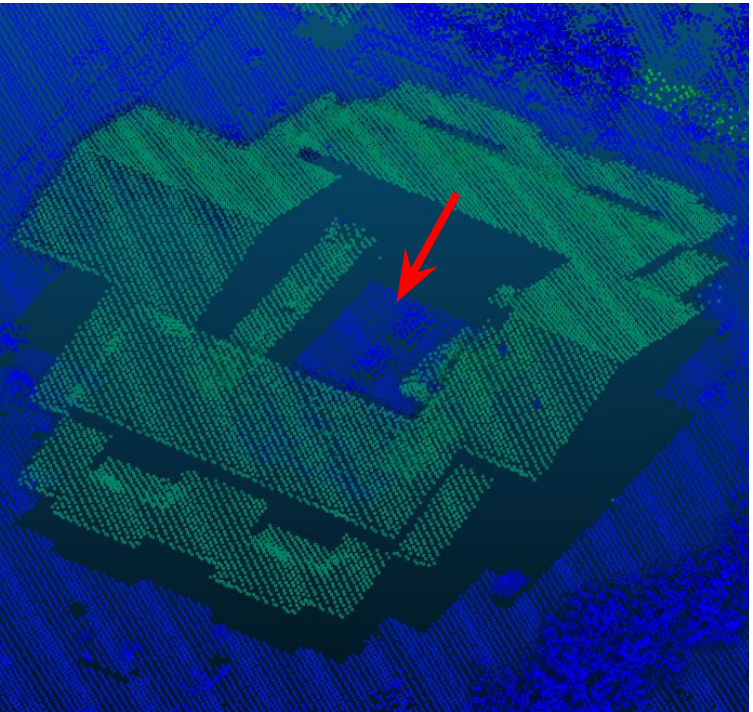
Image by Ivan Pađen

Special cases, Limitations

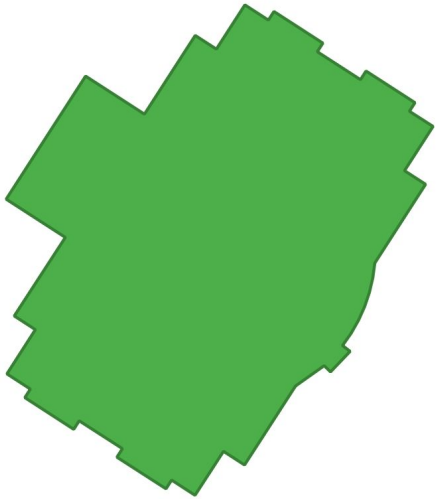
Groundparts

In some cases BAG footprint includes groundparts

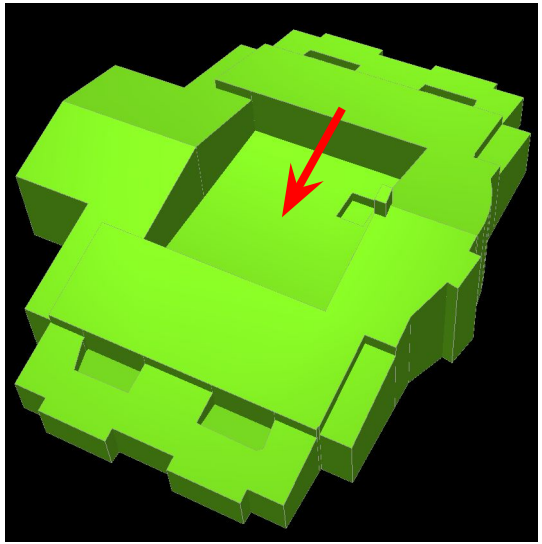
AHN3 ground and building class



BAG footprint



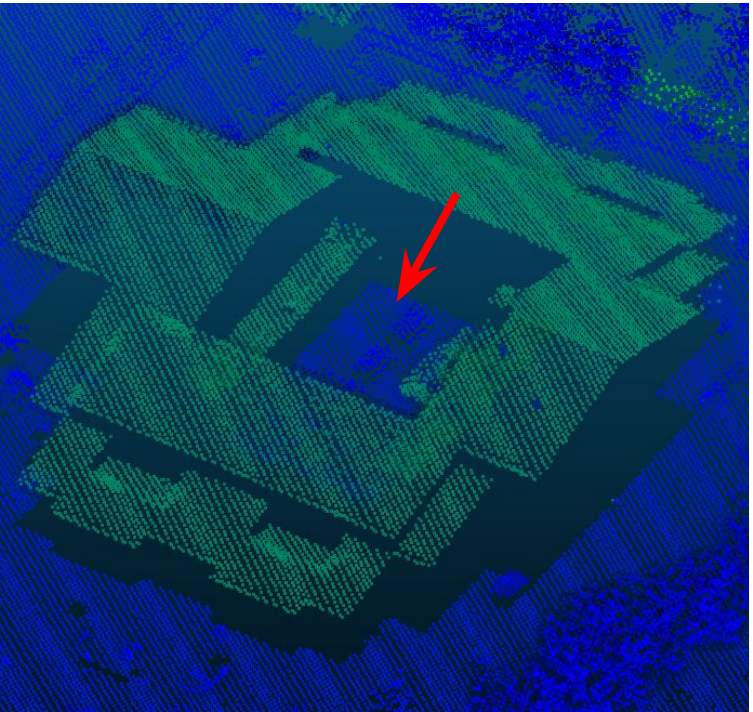
Reconstruction result:
roofplane fitted to groundpart



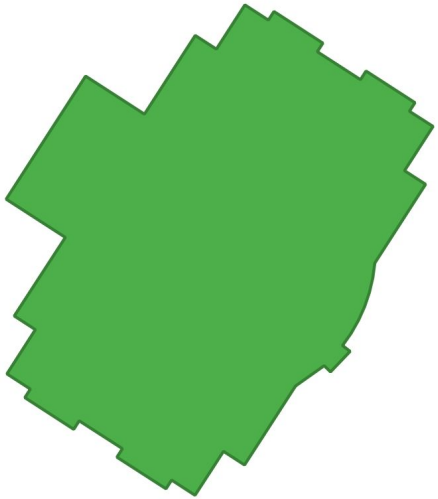
Groundparts

Reconstruction with groundpart detection

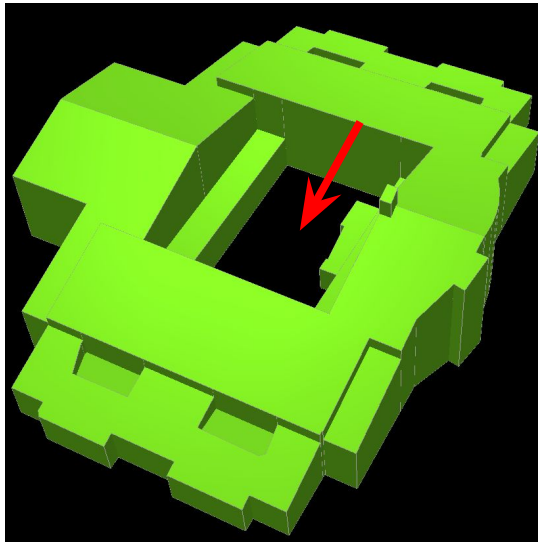
AHN3 ground and building class



BAG footprint

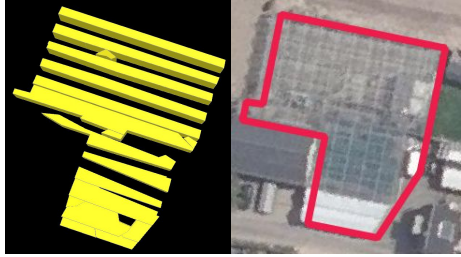


Reconstruction result:
groundpart removed from output

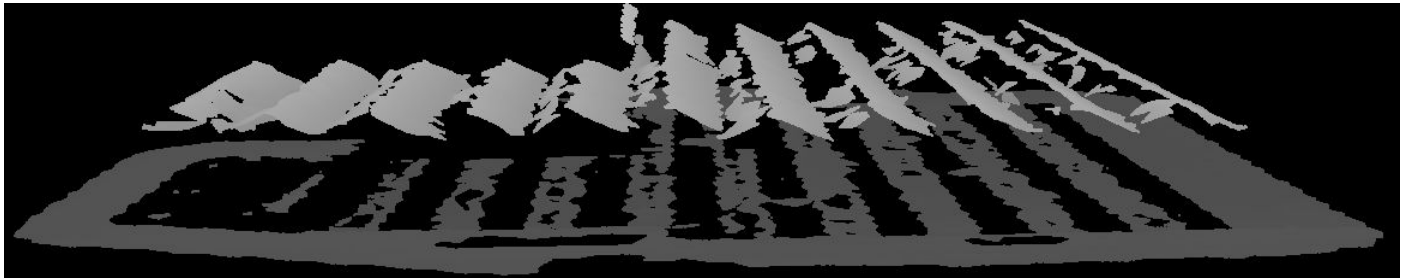


Limitation: glass roofs

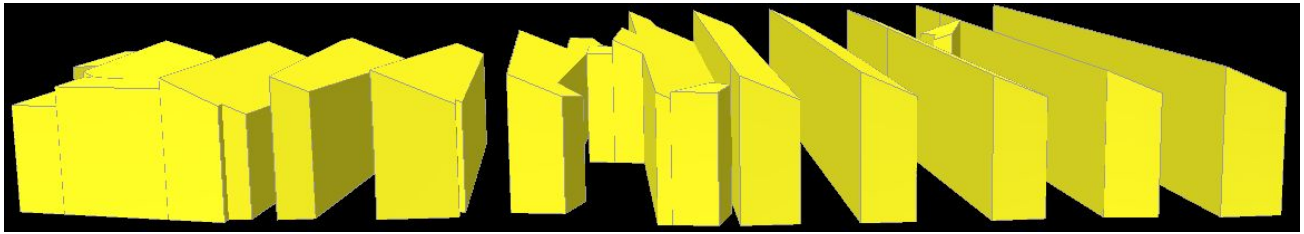
Green houses: both points on ground and on roof



AHN3
ground and building class



Heightfield



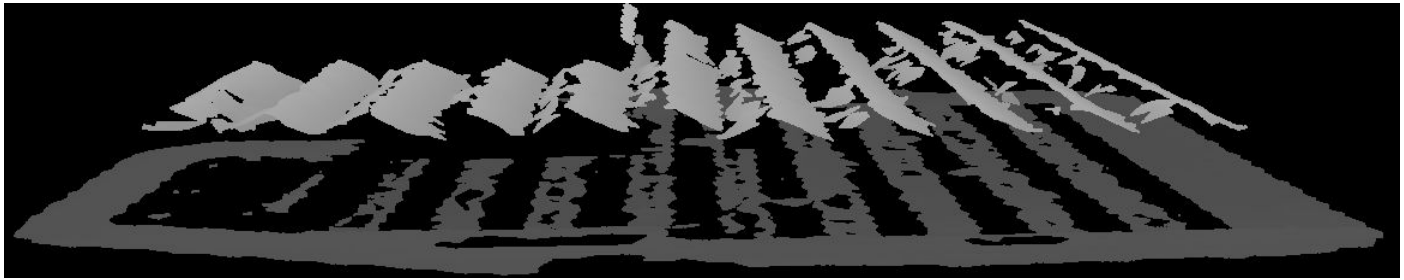
Reconstruction result

Limitation: glass roofs

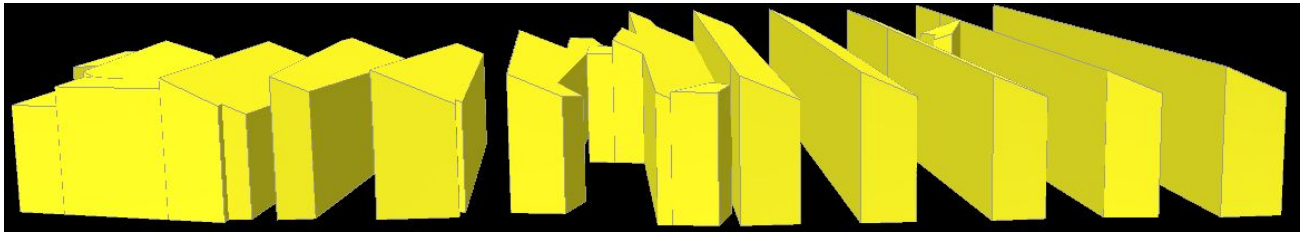
Green houses: both points on ground and on roof

Current solution:
Use greenhouse classification from TOP10NL, reconstruct as LoD 1.2

AHN3
ground and building class



Heightfield

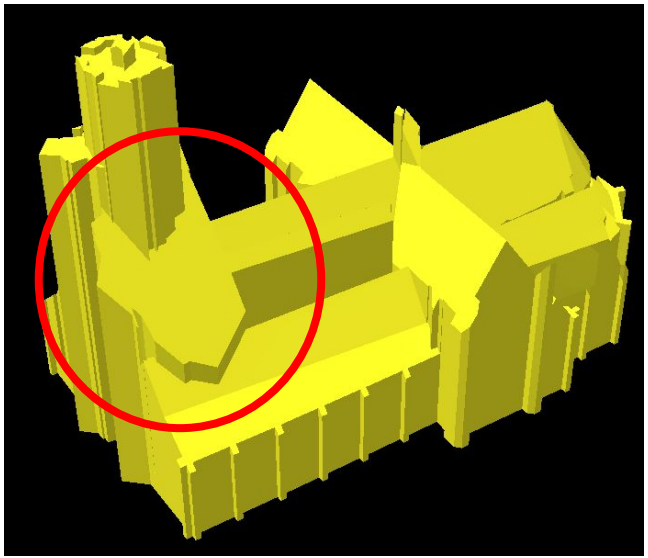


Reconstruction result

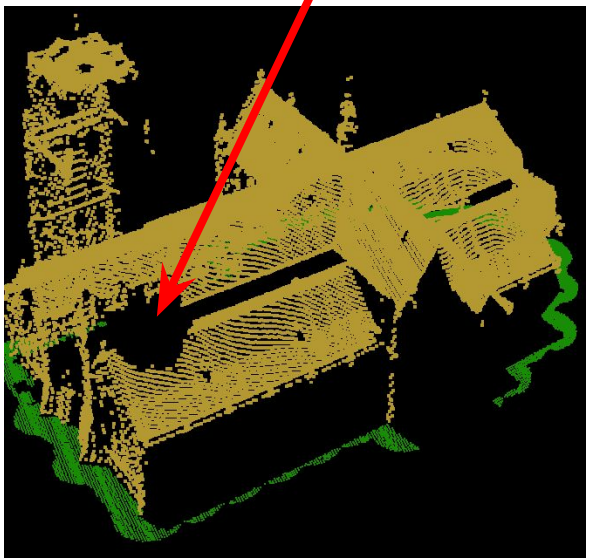
Spherical surfaces

Are approximated with planar surfaces if sufficient point density

Limitations: occlusion/no-data



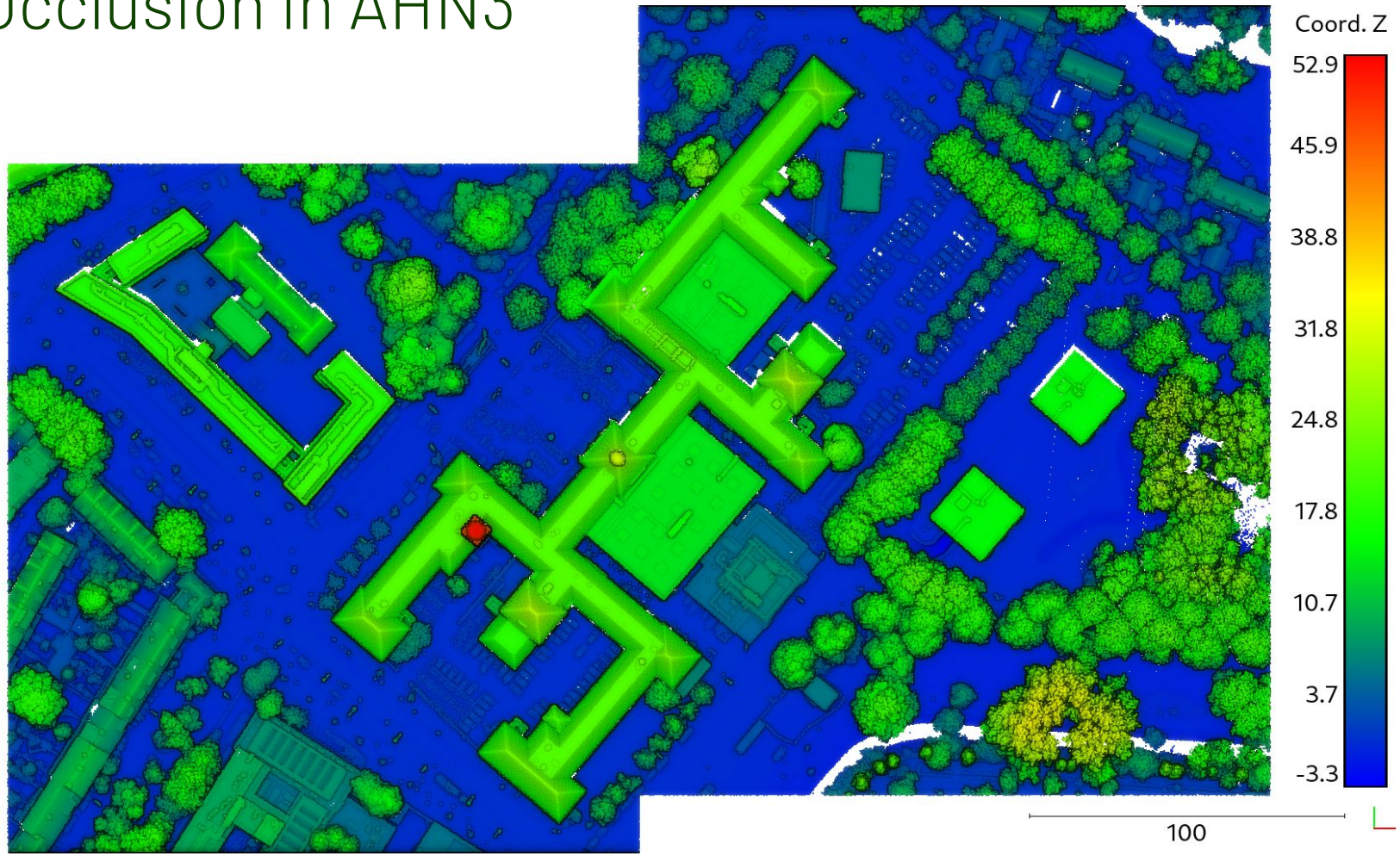
Reconstruction



AHN3 ground and building class

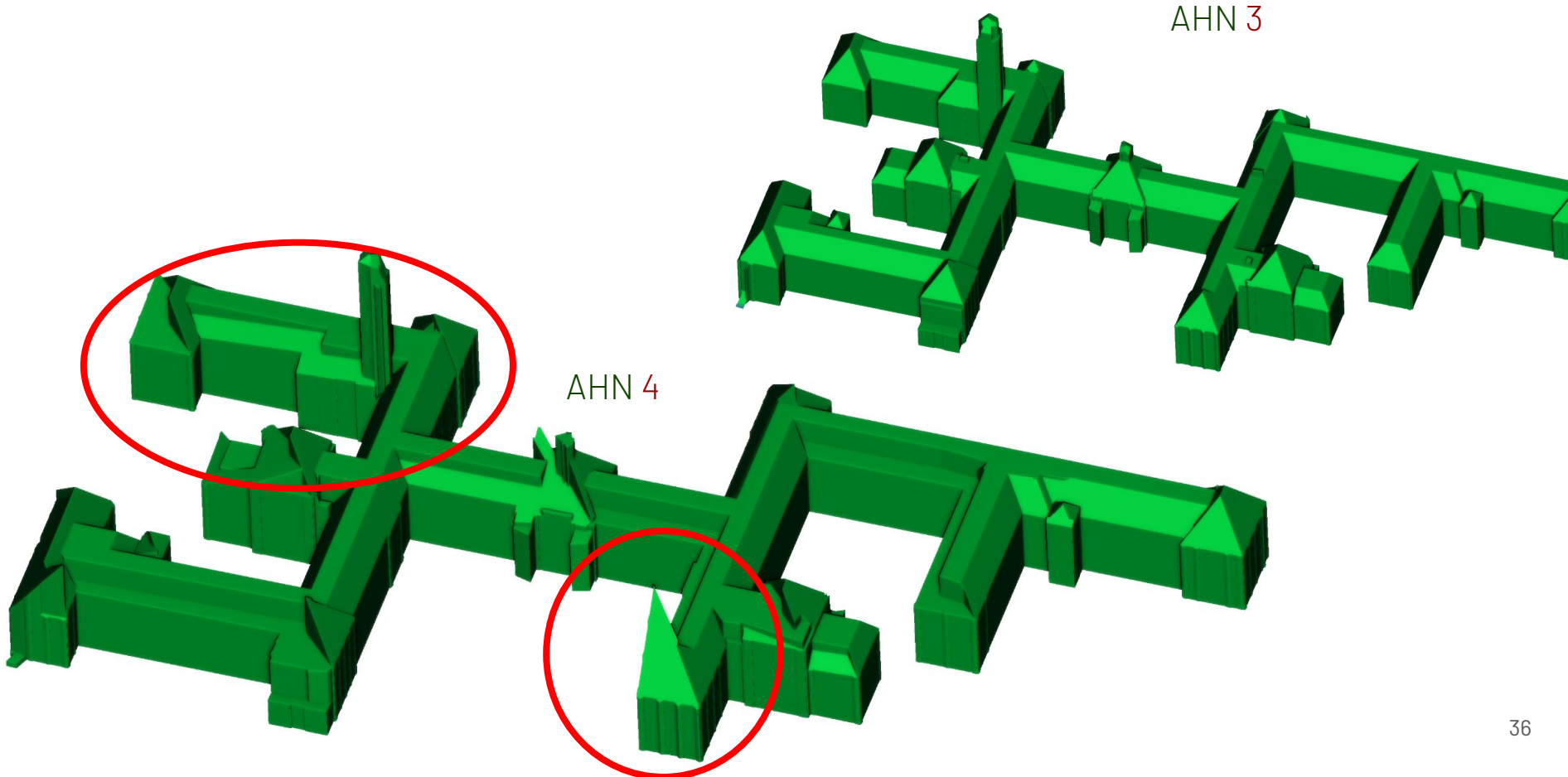
Heightfield

Occlusion in AHN3

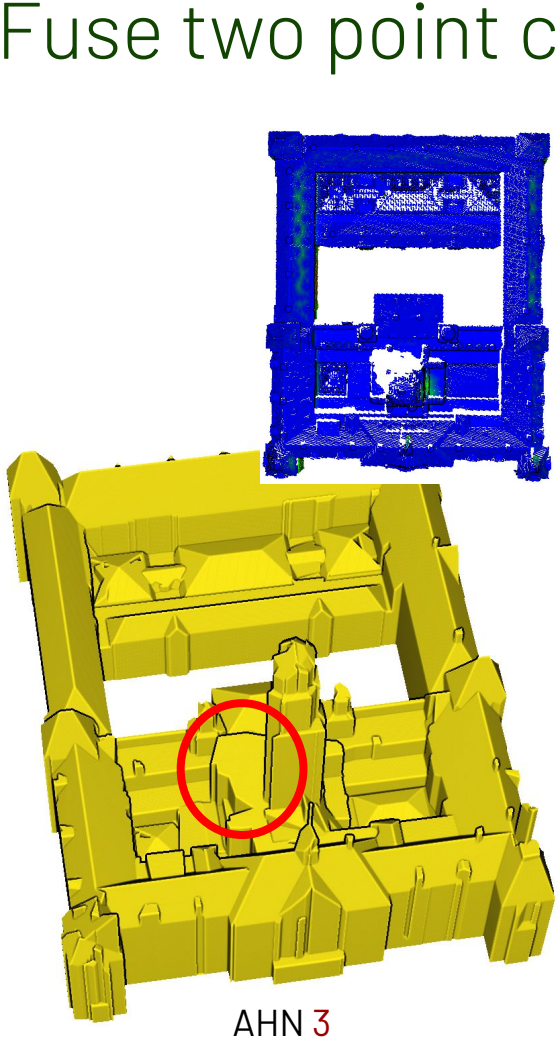


Occlusion in AHN4

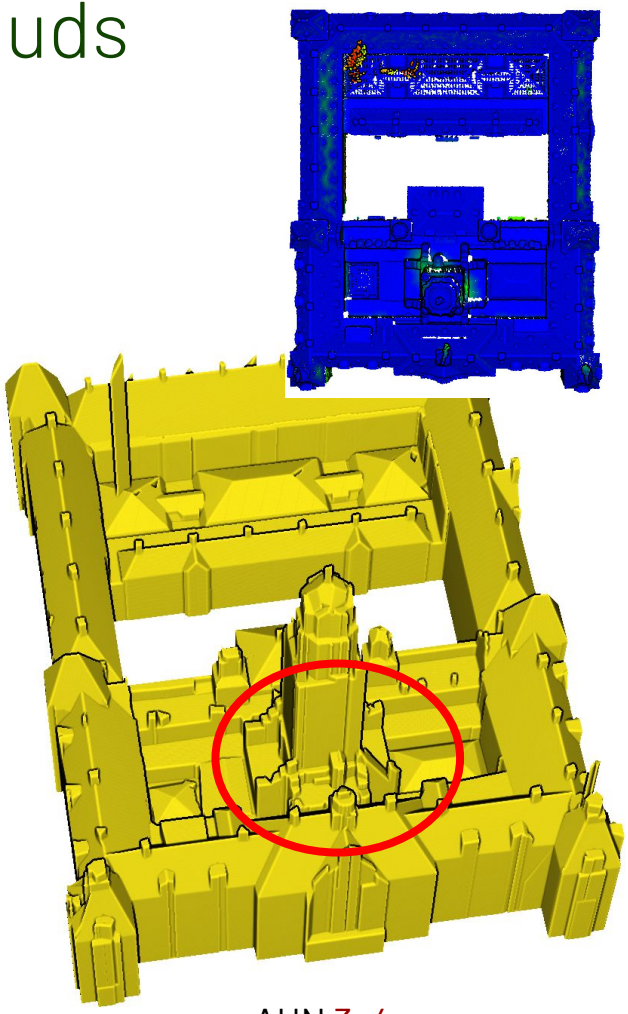
Occlusion effect on reconstruction



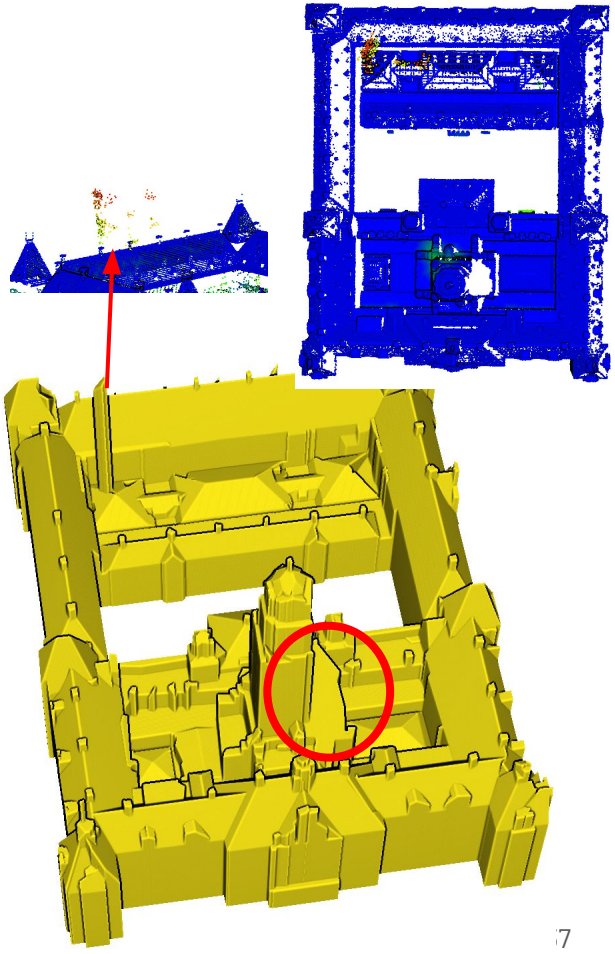
Fuse two point clouds



AHN 3



AHN 3+4

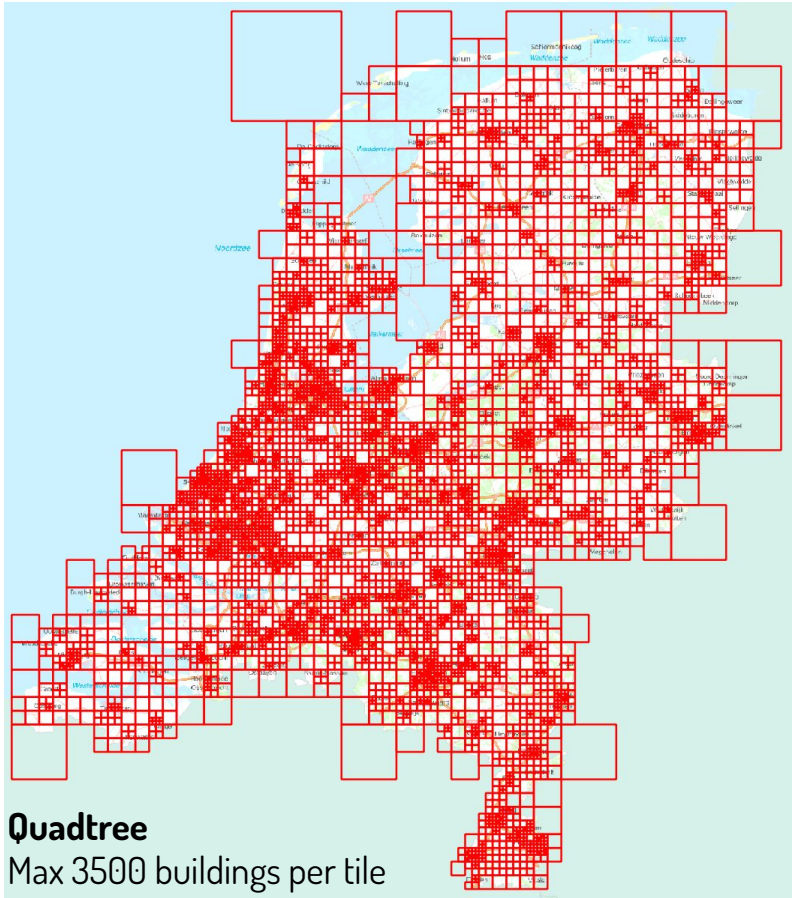


AHN 4

Data management

How to tile the data?

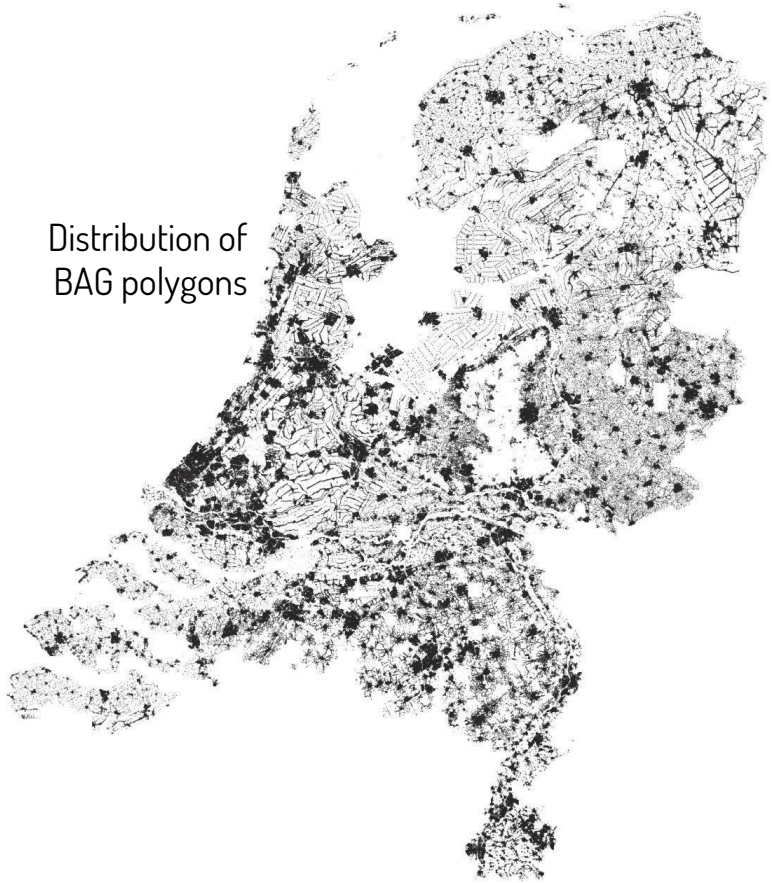
Distribution of
BAG polygons

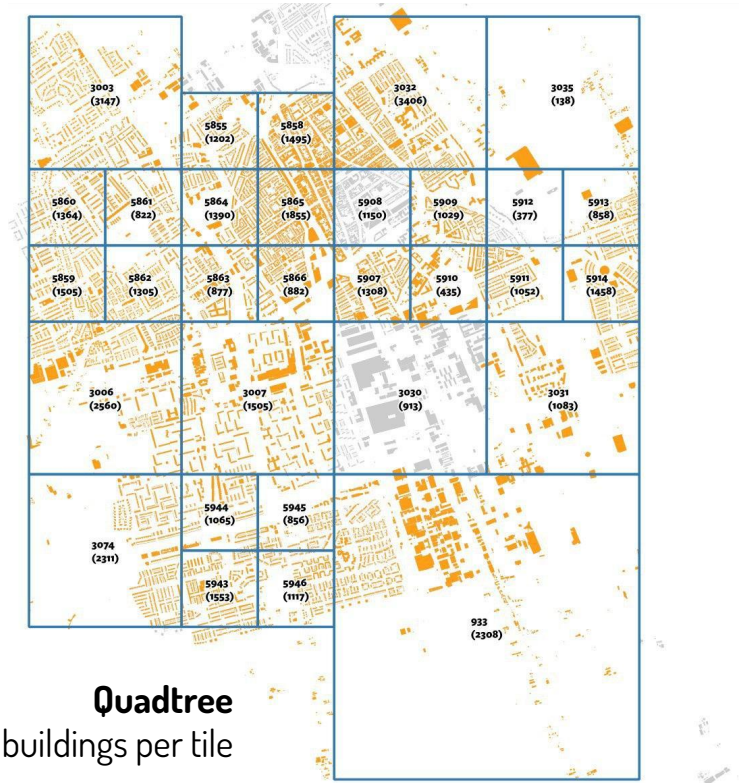


Quadtree
Max 3500 buildings per tile

How to tile the data?

Distribution of
BAG polygons





Quadtree
Max 3500 buildings per tile

Data distribution

Tiles

- GeoPackage
- CityJSON
- OBJ (triangulated)

Webservices

- WFS/WMS
- 3DBAG API
  (OGC API CityJSON Features)

Downloads for tile number 9-280-556

To keep filesizes manageable the 3DBAG dataset is subdived in tiles. For each tile we offer the data in a number of different file formats. Use the button below to select the tile of interest to see the download options.

Tile number	Format	File	SHA-256	Version
9-280-556	CityJSON	9-280-556.city.json	See tiles layer in WFS	v2024.02.28
9-280-556	OBJ	9-280-556-obj.zip	See tiles layer in WFS	v2024.02.28
9-280-556	GPKG	9-280-556.gpkg	See tiles layer in WFS	v2024.02.28

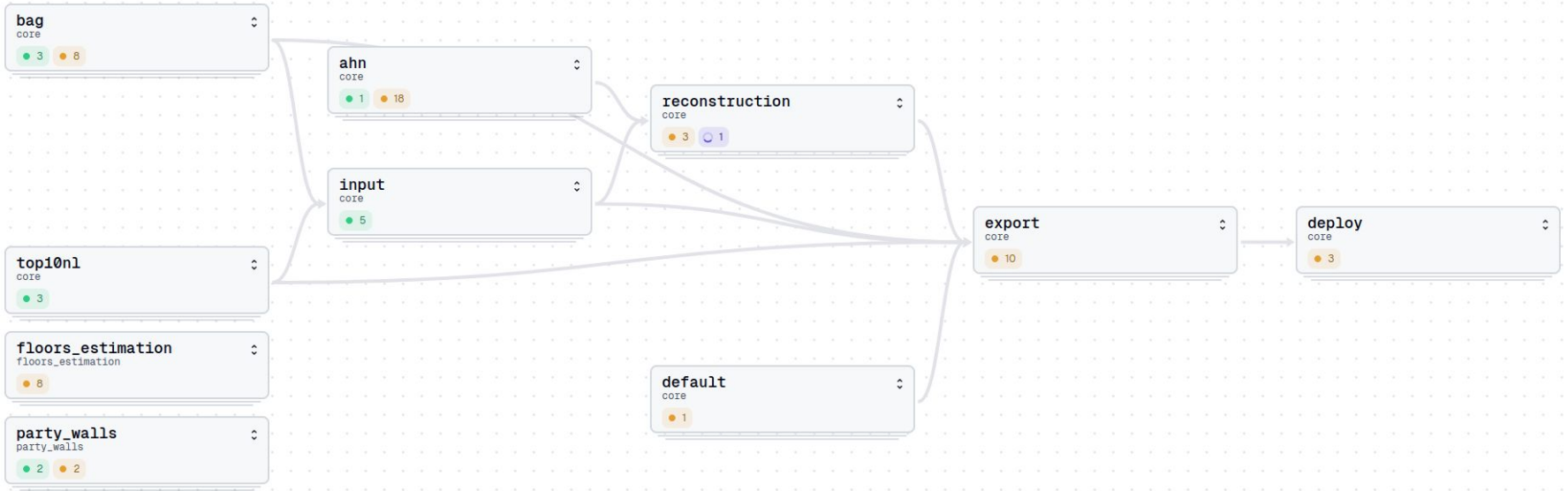
Webservices

These allow you to explore the entire dataset in another software (eg. QGIS) without having to download anything beforehand. Note that only the 2D projection of the models is served via WMS/WFS.

Type	URL
WMS	https://data.3dbag.nl/api/BAG3D/wms?request=getcapabilities
WFS	https://data.3dbag.nl/api/BAG3D/wfs?request=getcapabilities
3D API (experimental)	https://api.3dbag.nl/

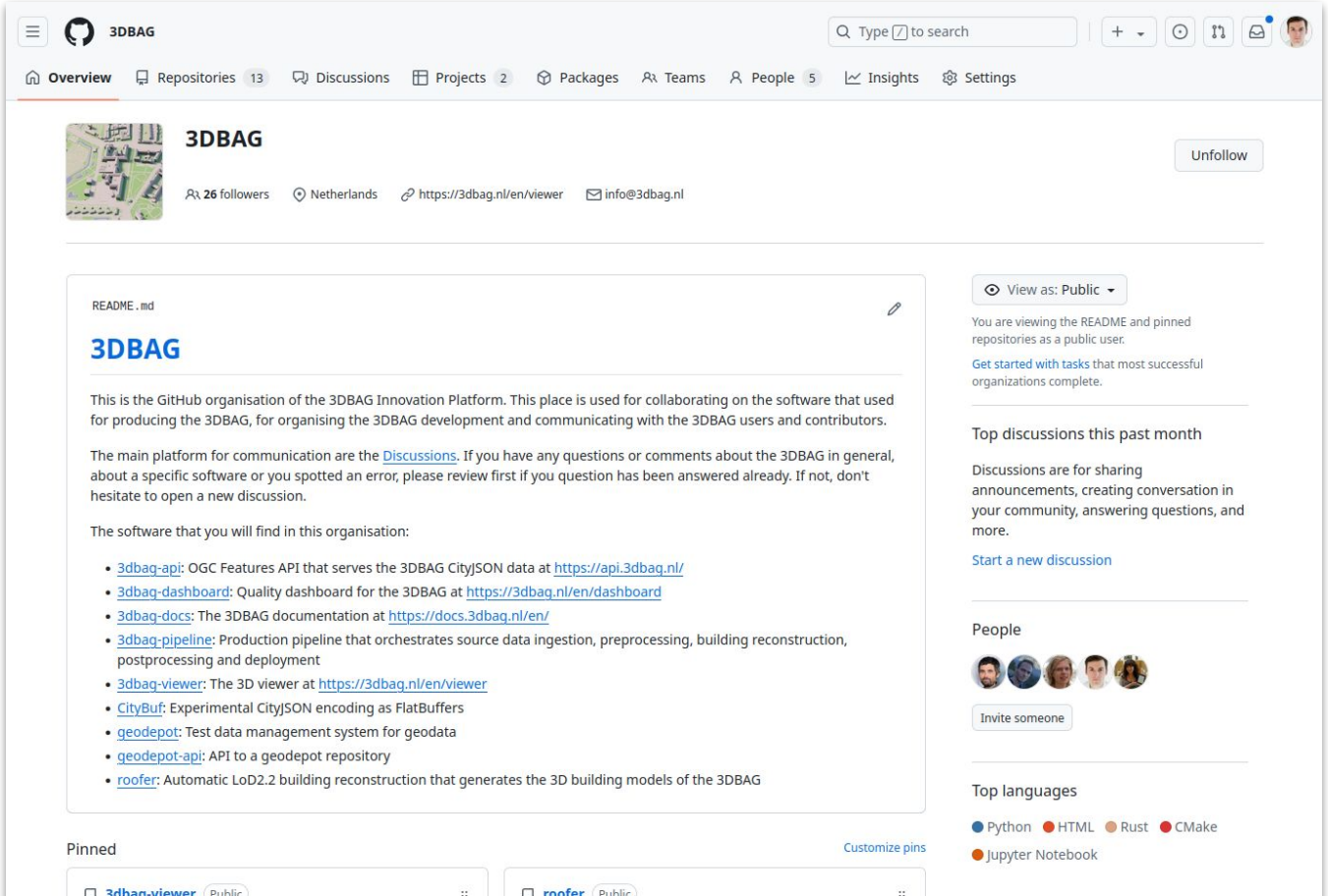
3DBAG tech stack

Data orchestration pipeline



3DBAG repositories

https://github.com/3DBAG



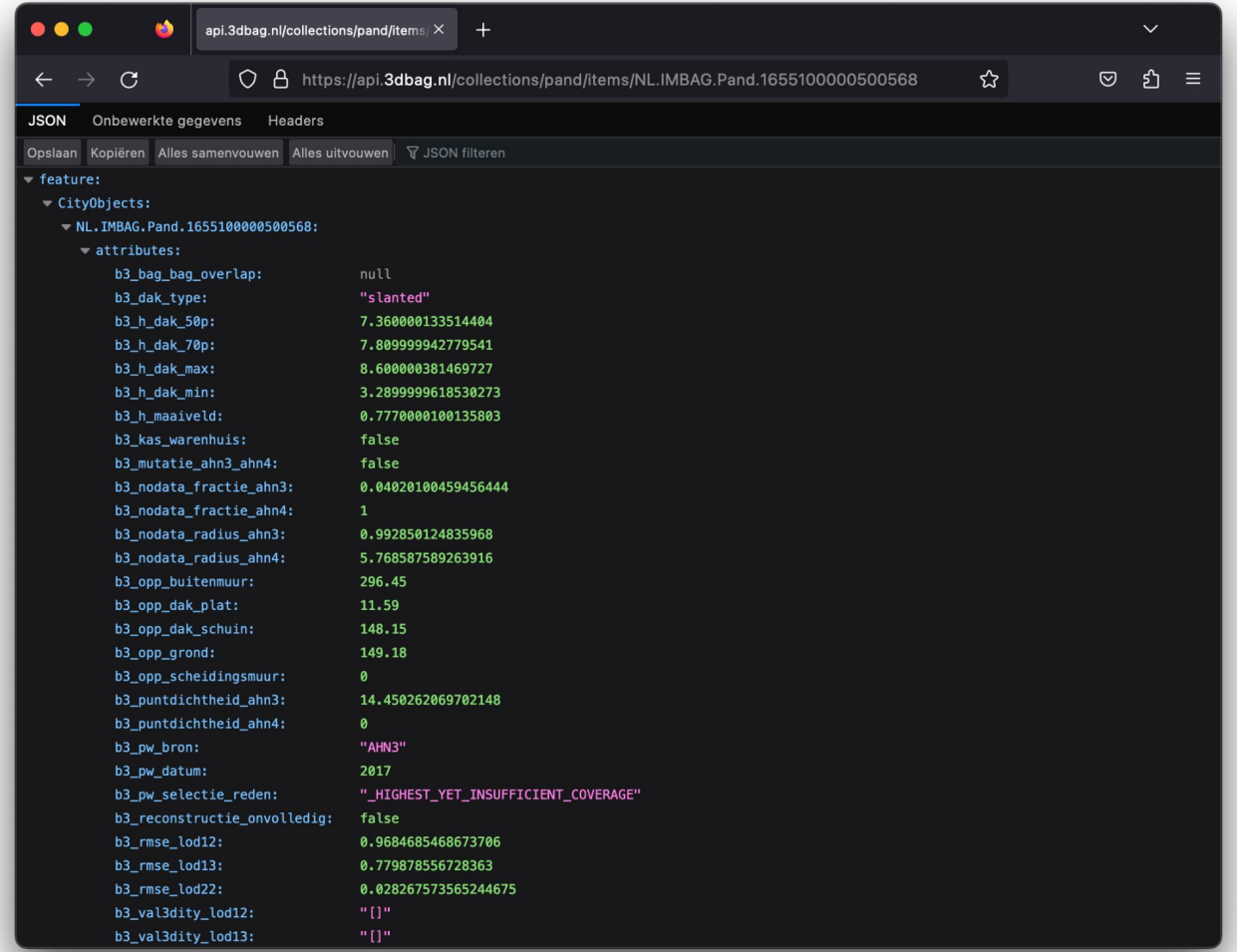
3DBAG additions

3DBAG API

Request directly CityJSON Features by

- Building ID
- Bounding box

Based on CJDB
(2022 geomatics synthesis project)



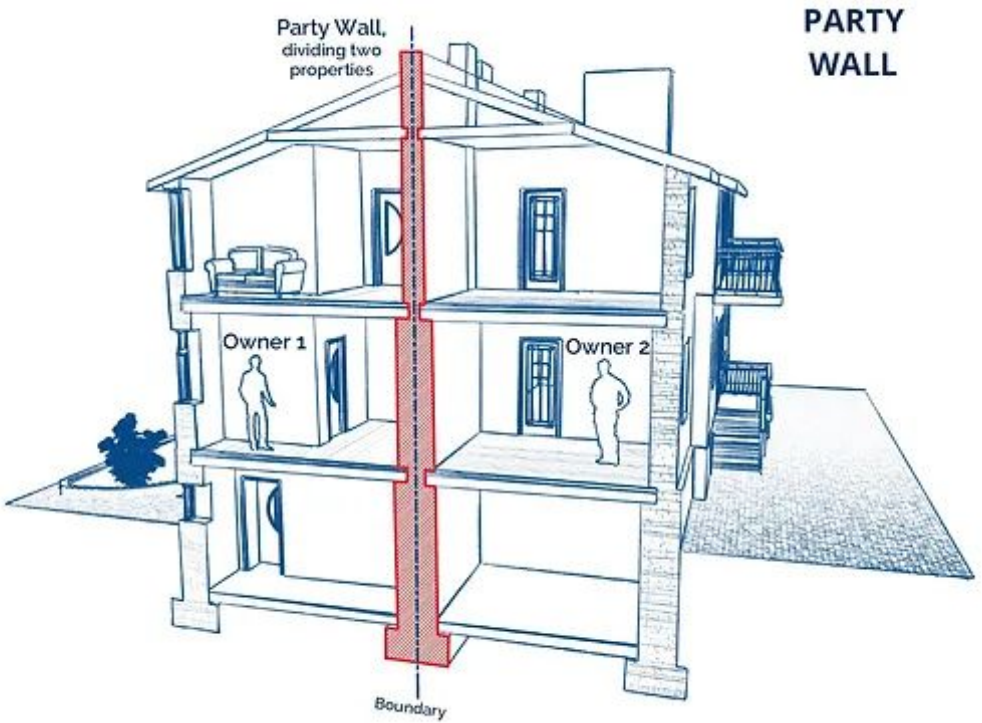
Party wall areas

We calculate and include:

- area party walls
- area exterior walls
- area floor surfaces
- area roof surfaces
- building volume

Needed for eg. energy label estimation

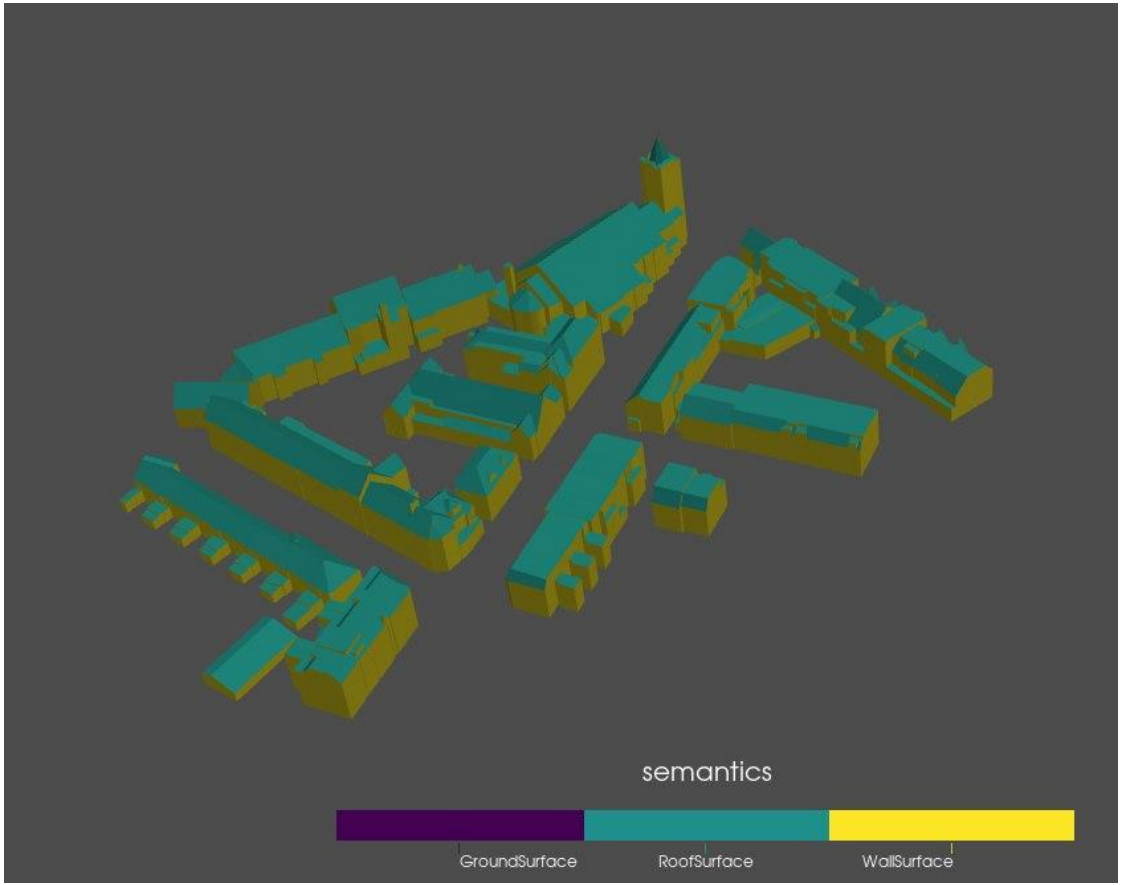
Sponsored by RVO



Calculation of party walls

Most semantic surfaces already assigned during reconstruction

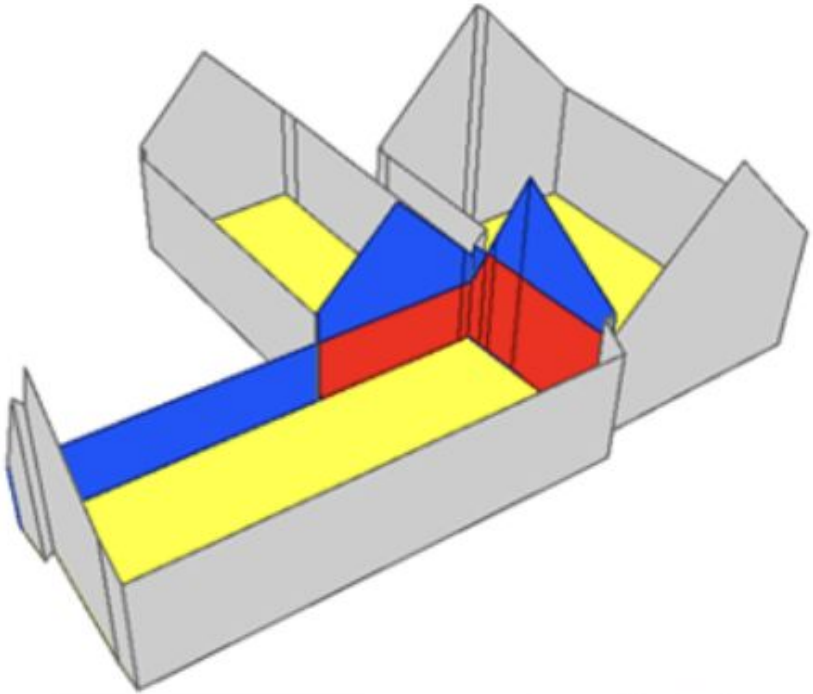
We just need to split *WallSurfaces* into party walls and exterior walls.



Calculation of party walls

For each building

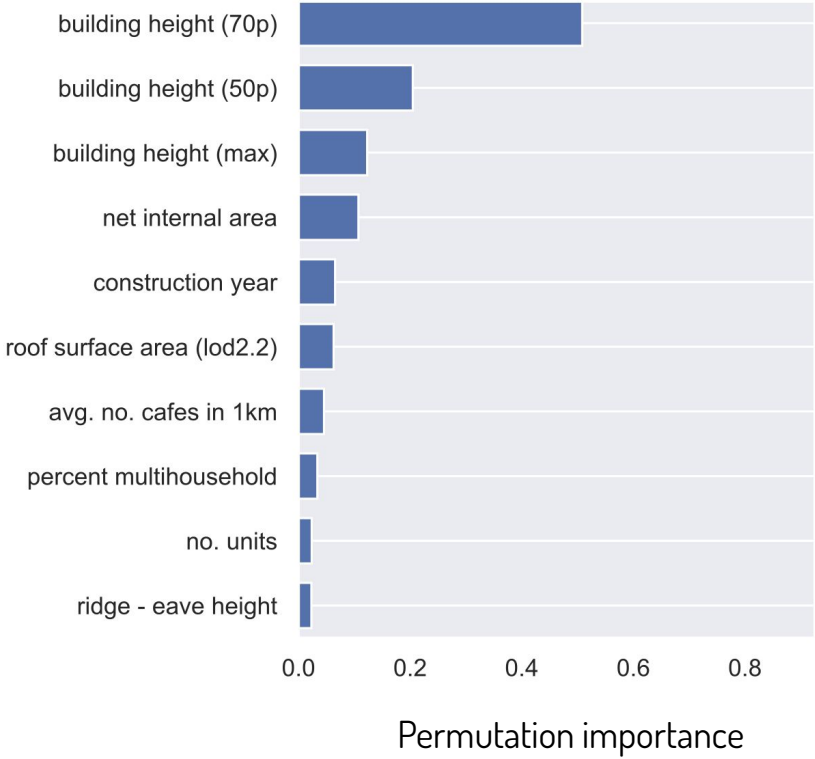
1. Find neighbouring buildings
2. Extract all 3D wall polygons
3. Find co-planar polygons through clustering by plane parameters
4. Intersect the co-planar wall polygons from different buildings.
5. Calculate area of intersection



Estimation nr of floors

Based on MSc thesis of Ellie Roy (Geomatics 2022)

- Machine learning method based on Gradient Boosting Regression
- Model trained on groundtruth data from a couple of Dutch municipalities
- Used features collected from various datasets (3DBAG, CBS, ...)
- Accuracy drops for >5 floors
- Available in 3DBAG release v2024.02.28

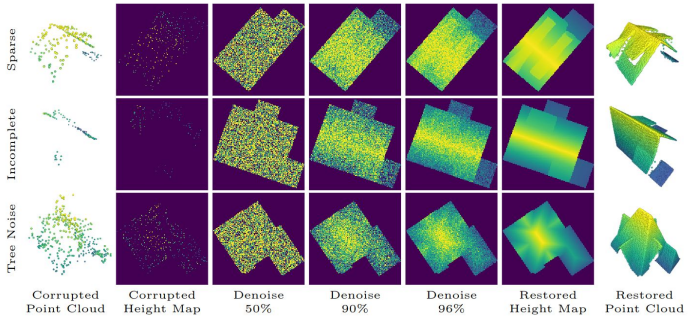


3DBAG future work

Future work

- Data updates
  - Eg. when new pointcloud is available
- Further improve geometric validity
  - Already > 99% is valid, but there are still some issues
- More efficient data exchange format
  - CityJSON has some limitations (text based, requires parsing)
  - "FlatCityBuf" a binary flatbuffer encoding of CityJSONSequences
    - No parsing needed, thus faster reading and more memory efficient
    - Also suitable for 'cloud native' applications
- Address defects in point cloud caused by eg. occlusion
  - Using 'RoofDiffusion' approach?

https://github.com/kylelo/RoofDiffusion



Thank you!

Ravi Peters
ravi.peters@3dgi.nl

Want to try the reconstruction algorithm yourself?

https://github.com/3dbag/roofer