

3D BAG

GE01004: 3D modelling of the built environment

Ravi Peters

03-06-2026



3D geoinformation

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



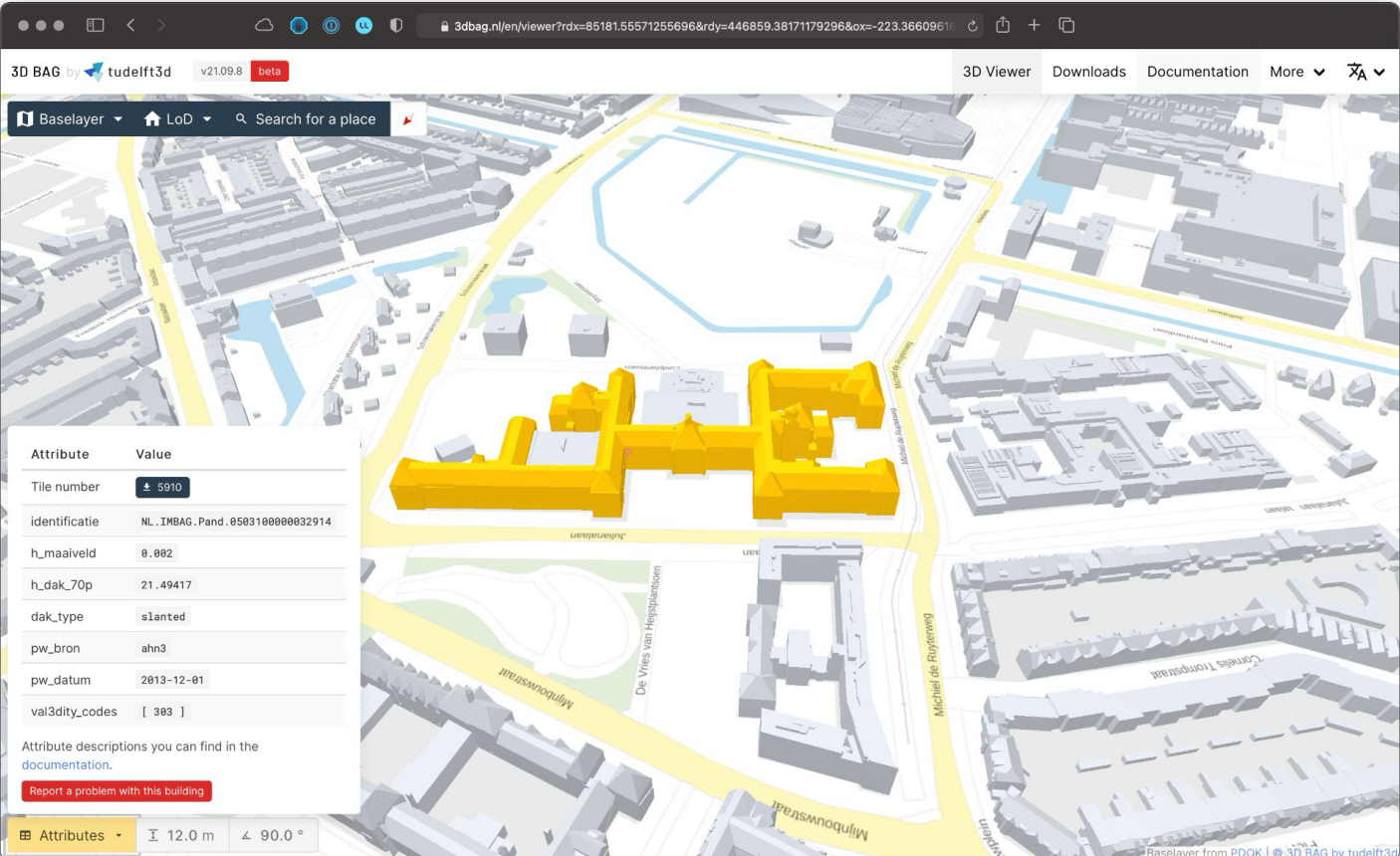
3DGI



The 3D BAG

3dbag.nl

Data:  



3D BAG by tudelft3d v21.09.8 beta

Baselayer LoD Search for a place

3D Viewer Downloads Documentation More

Attribute	Value
Tile number	± 5910
identificatie	NL.IMBAG.Pand.0503100000032914
h_maaiveld	0.002
h_dak_70p	21.49417
dak_type	slanted
pw_bron	ahn3
pw_datum	2013-12-01
val3dity_codes	[383]

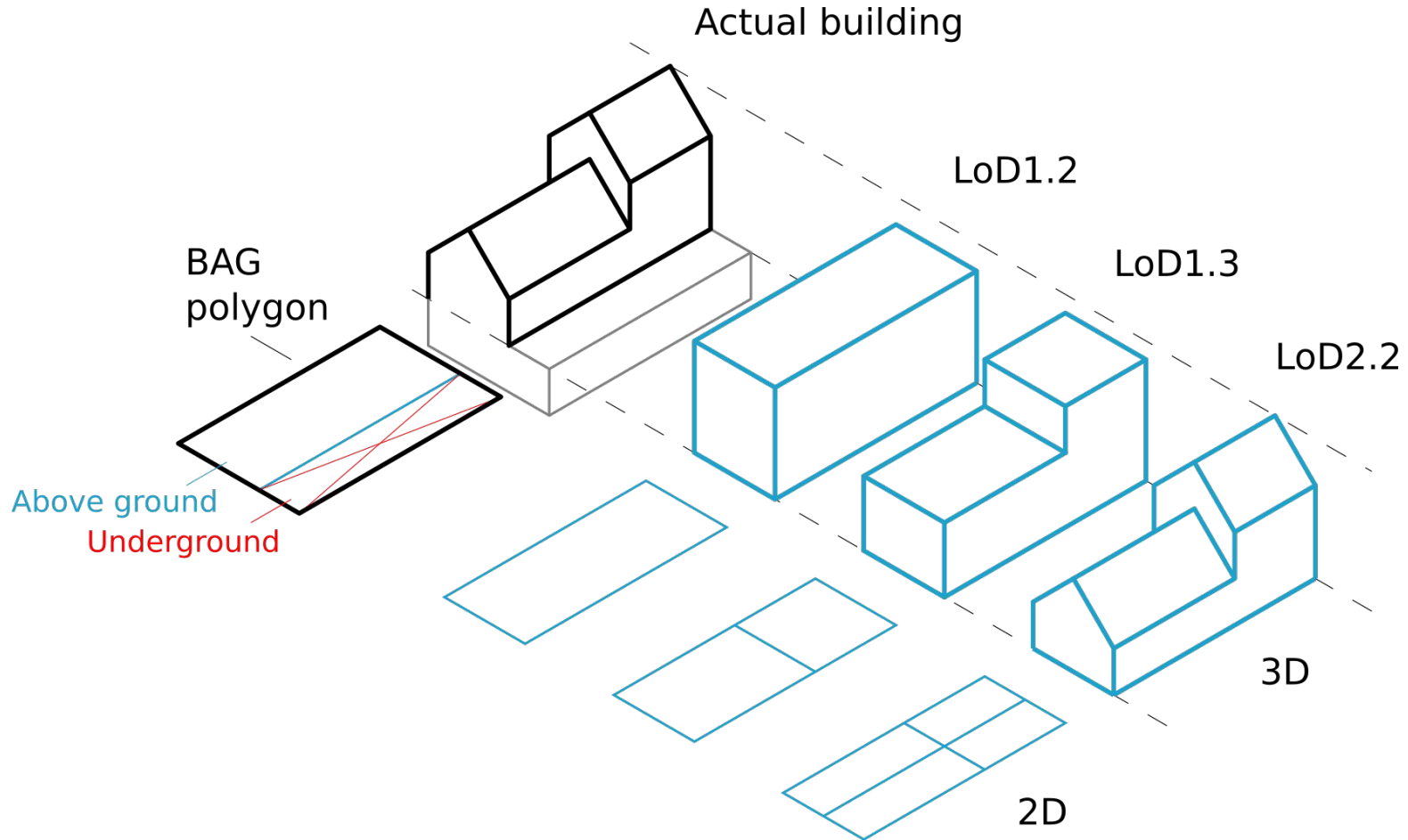
Attribute descriptions you can find in the [documentation](#).

[Report a problem with this building](#)

Attributes 12.0 m 90.0 °

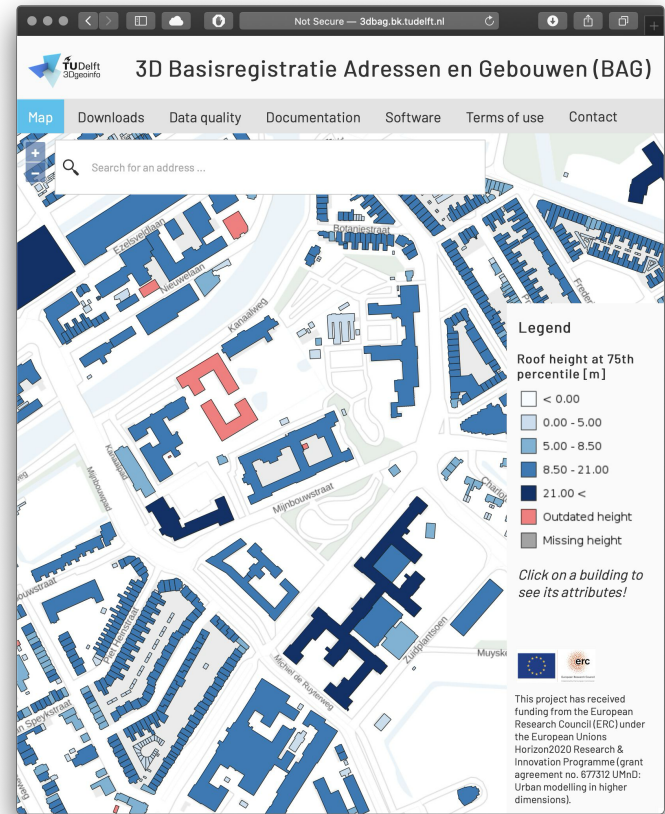
Baselayer from PDOK | 3D BAG by tudelft3d

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

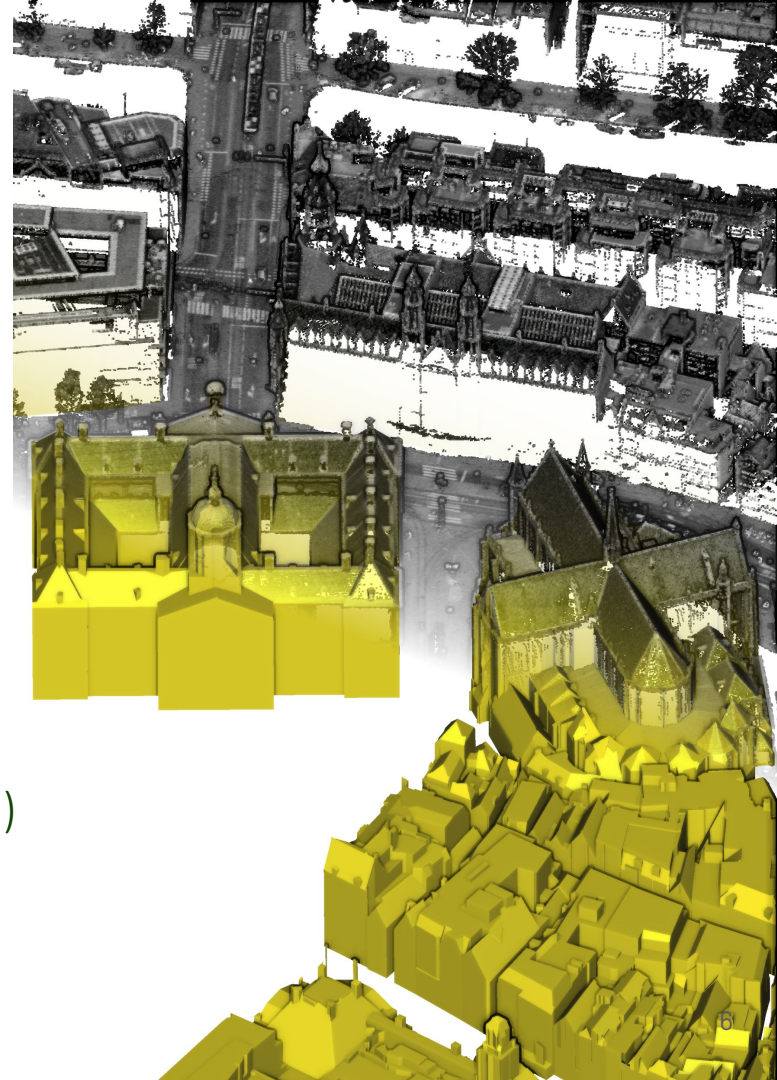
Open data in the Netherlands

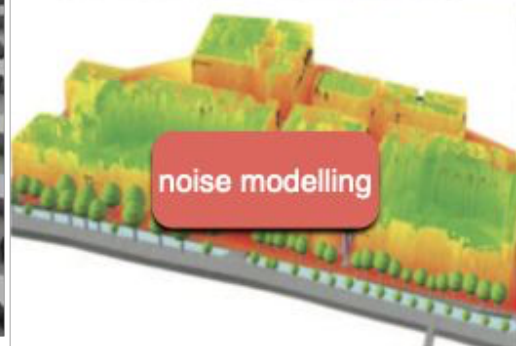
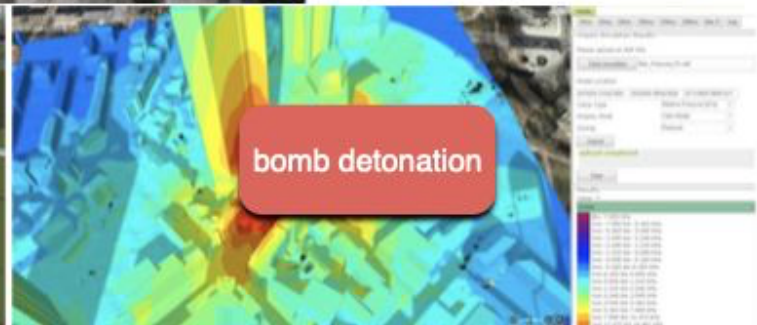
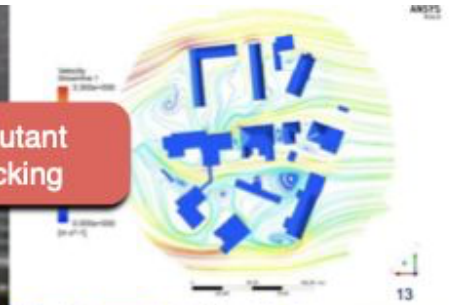
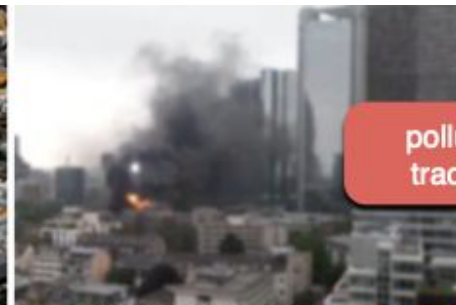
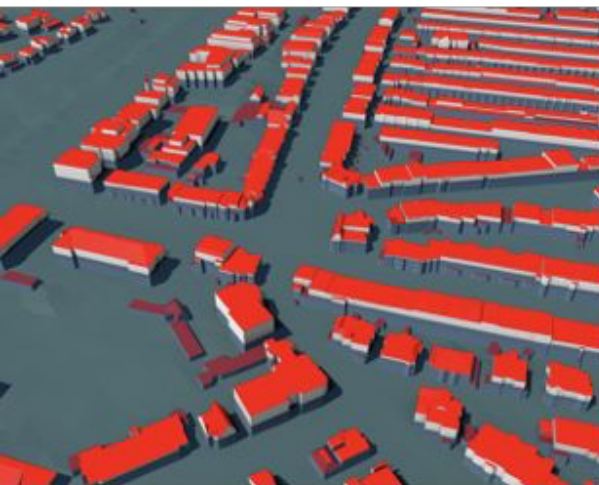
BAG <https://www.kadaster.nl/bag>

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 (AHN3)
occlusion and other no-data areas







Heating energy demand

Computational Fluid Dynamics

Solar panel potential

Heating energy demand

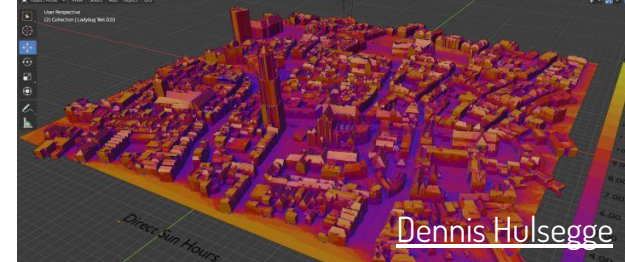
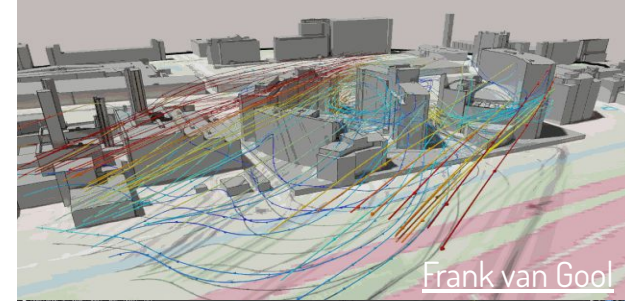
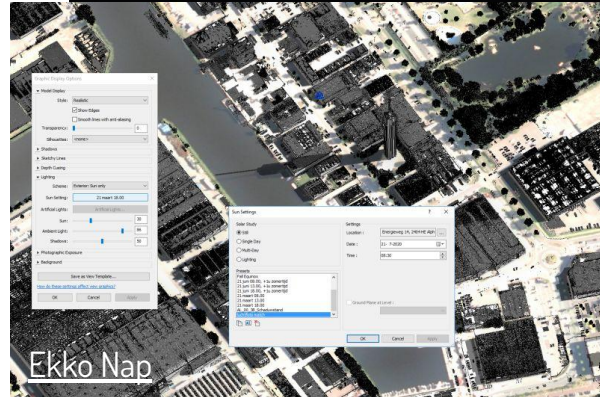
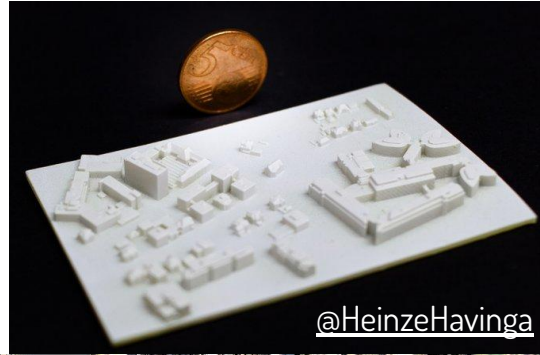
Computational Fluid Dynamics

Solar panel potential



3D BAG in practice

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



3DBAG in...



Femke Halsema • 3rd+
Burgemeester van Amsterdam
3h • 🌐

+ Follow

🌸 We staan aan de vooravond van de verjaardag van onze stad. Maandag 27 oktober vieren we dat Amsterdam, na een jaar vol festiviteiten, echt 750 jaar bestaat. En dan is het extra ...more

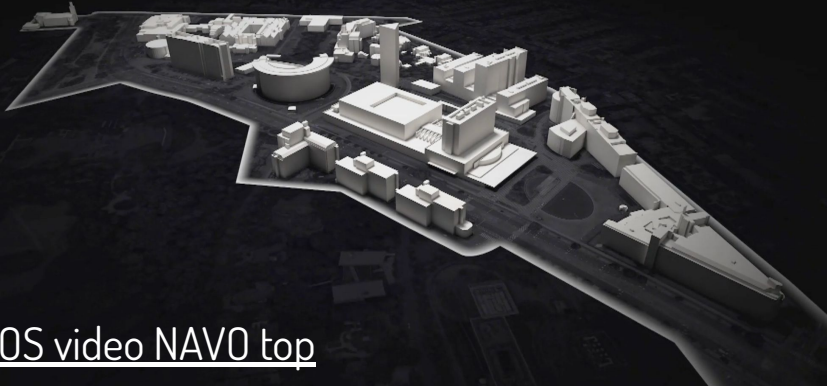
Show translation



👤 with Amsterdam 750

🌐 Wietse Balster and 210 others 9 comments • 6 reposts

NOS



NOS video NAVO top

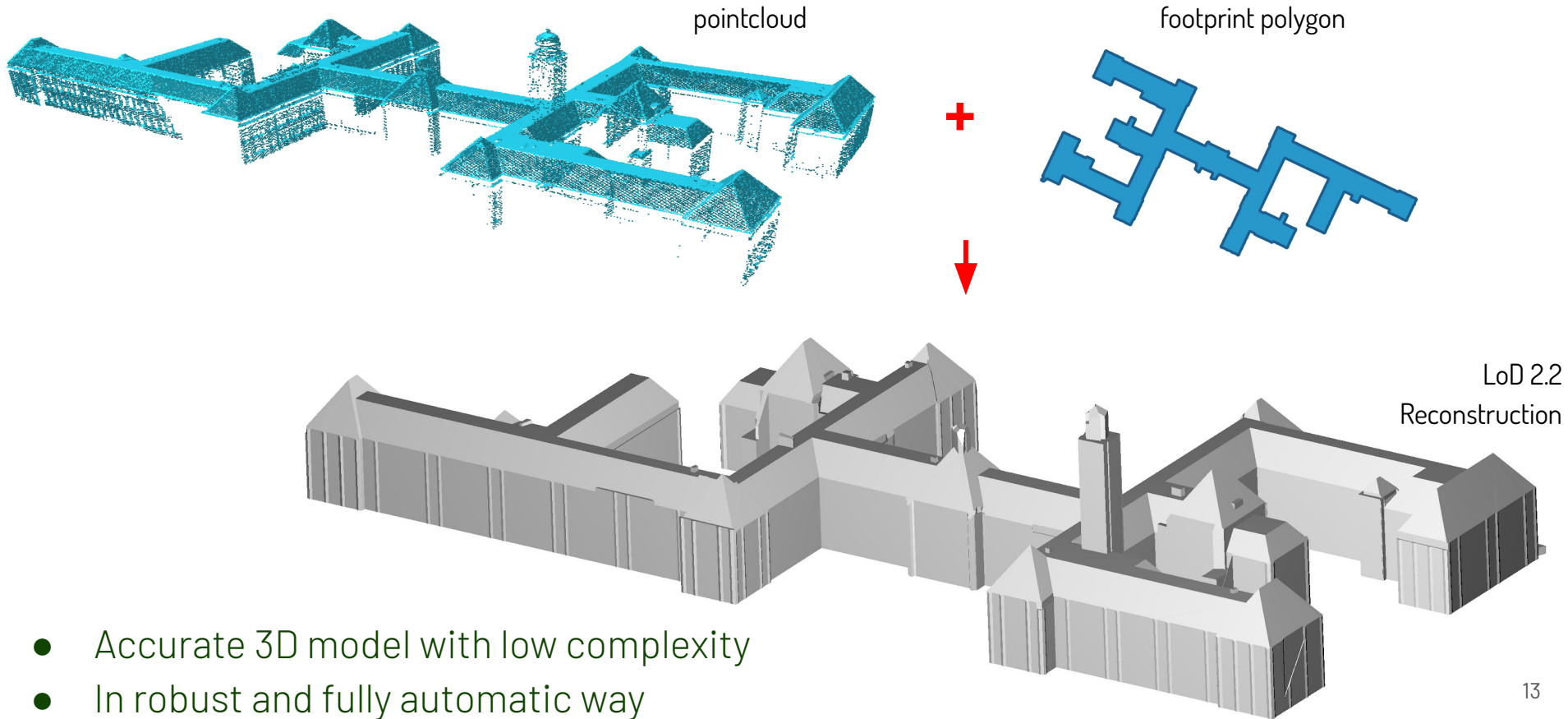


Gemeente Apeldoorn

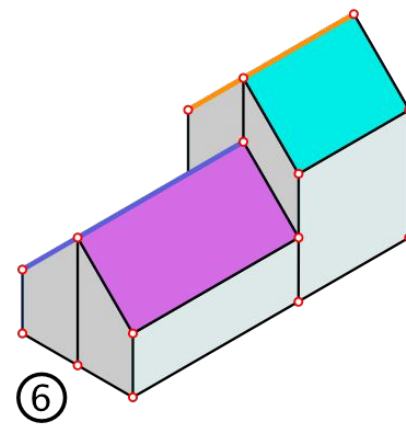
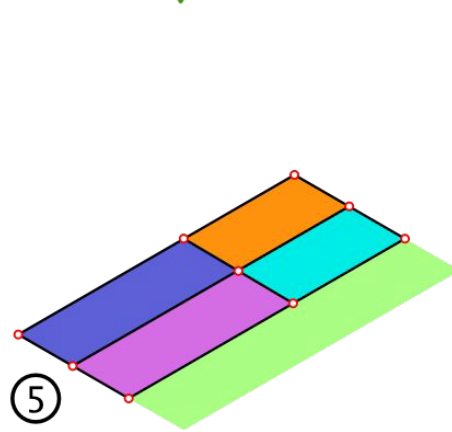
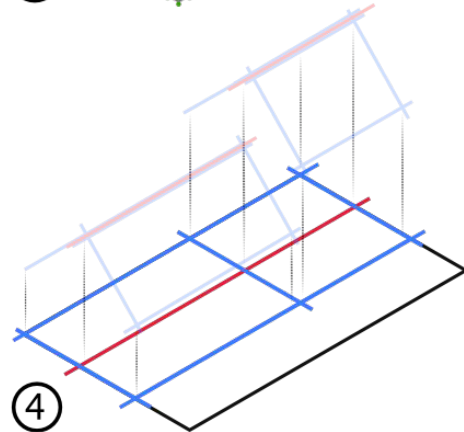
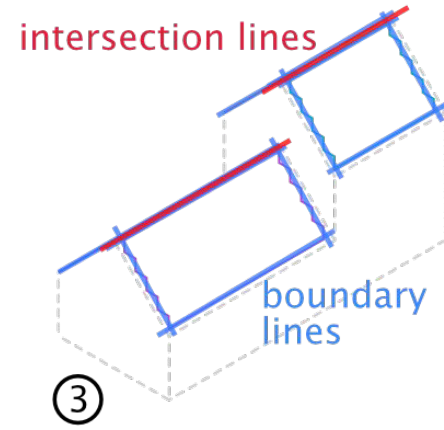
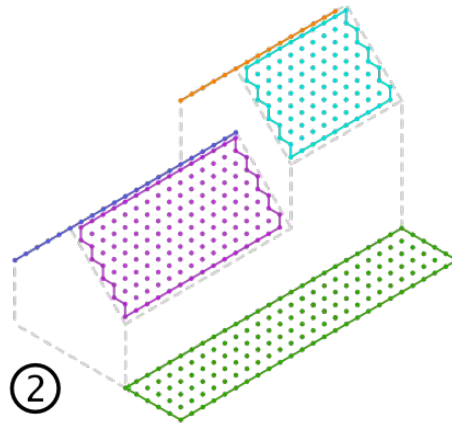
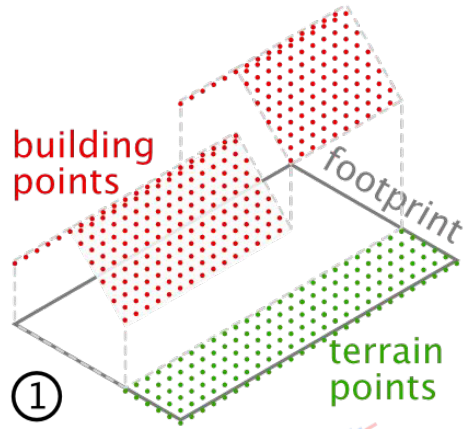
Reconstruction algorithm

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

Building reconstruction



Overview building reconstruction method

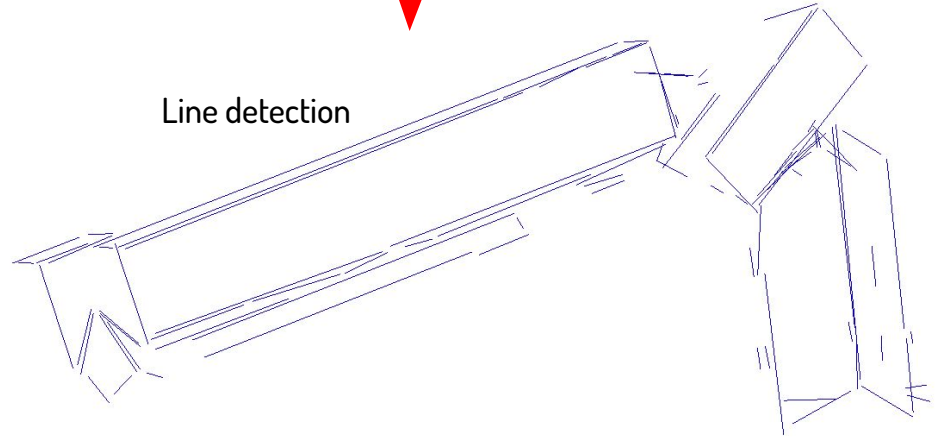
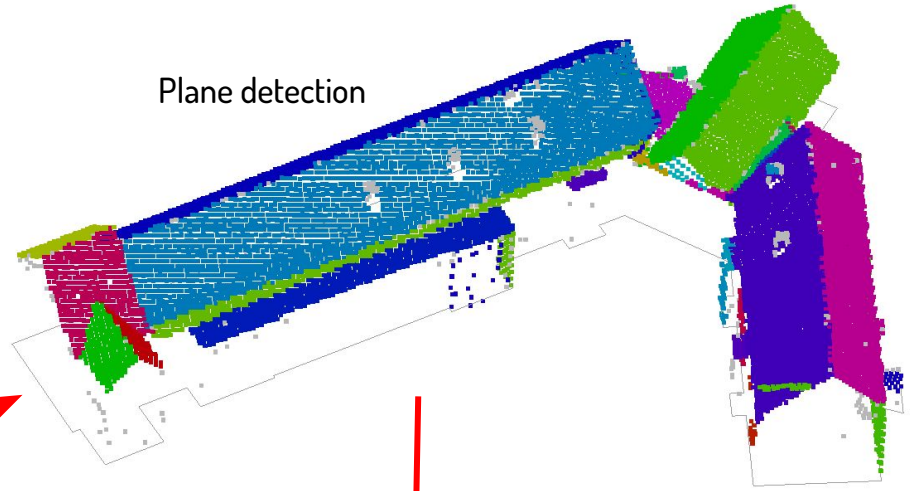
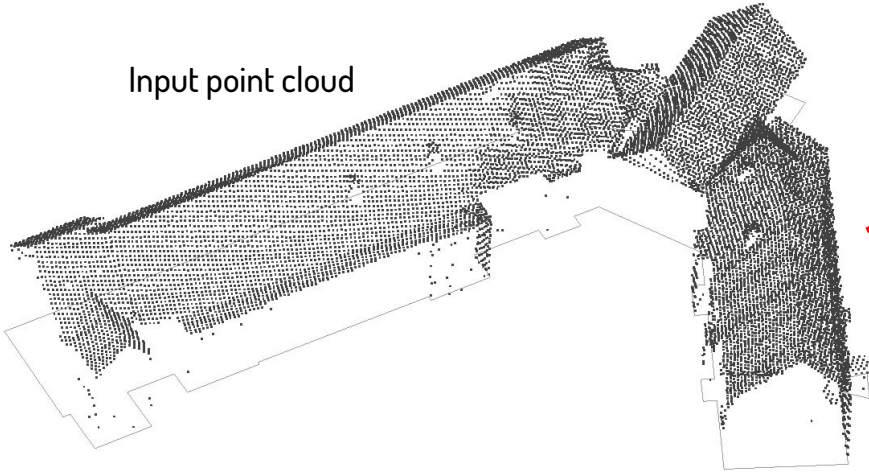


Feature extraction

Input point cloud

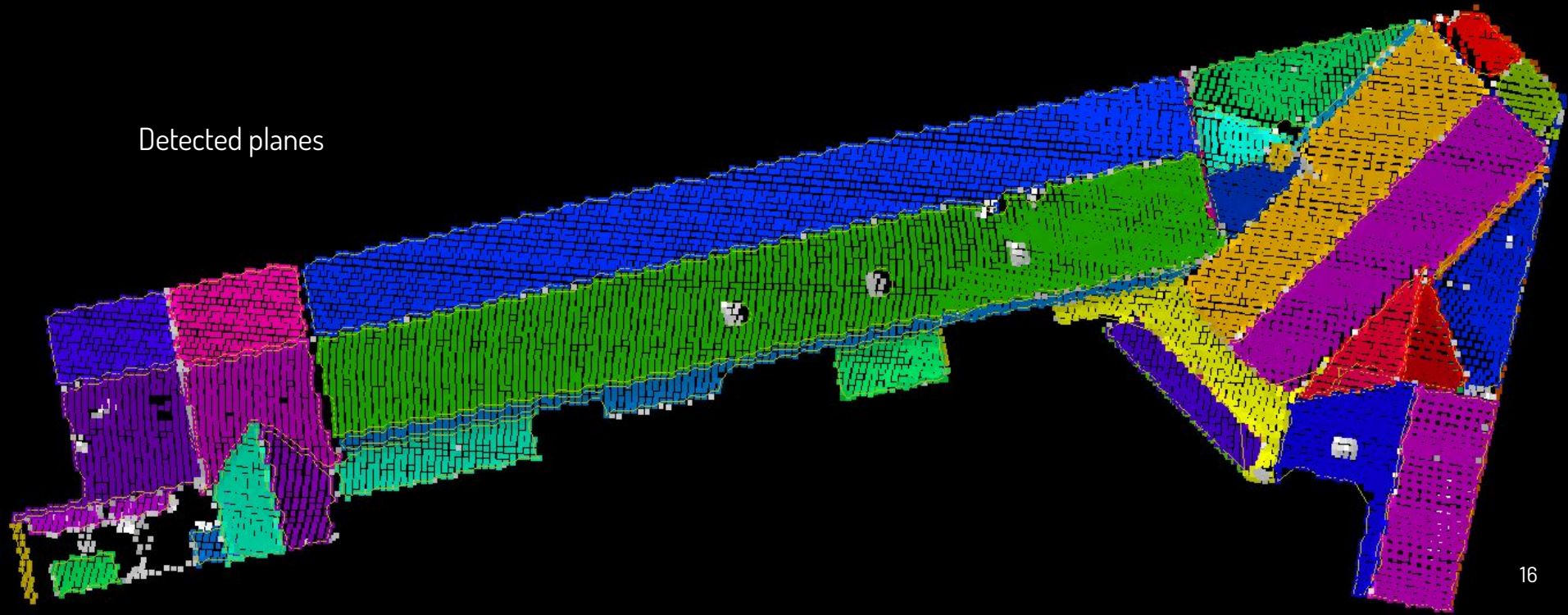
Plane detection

Line detection

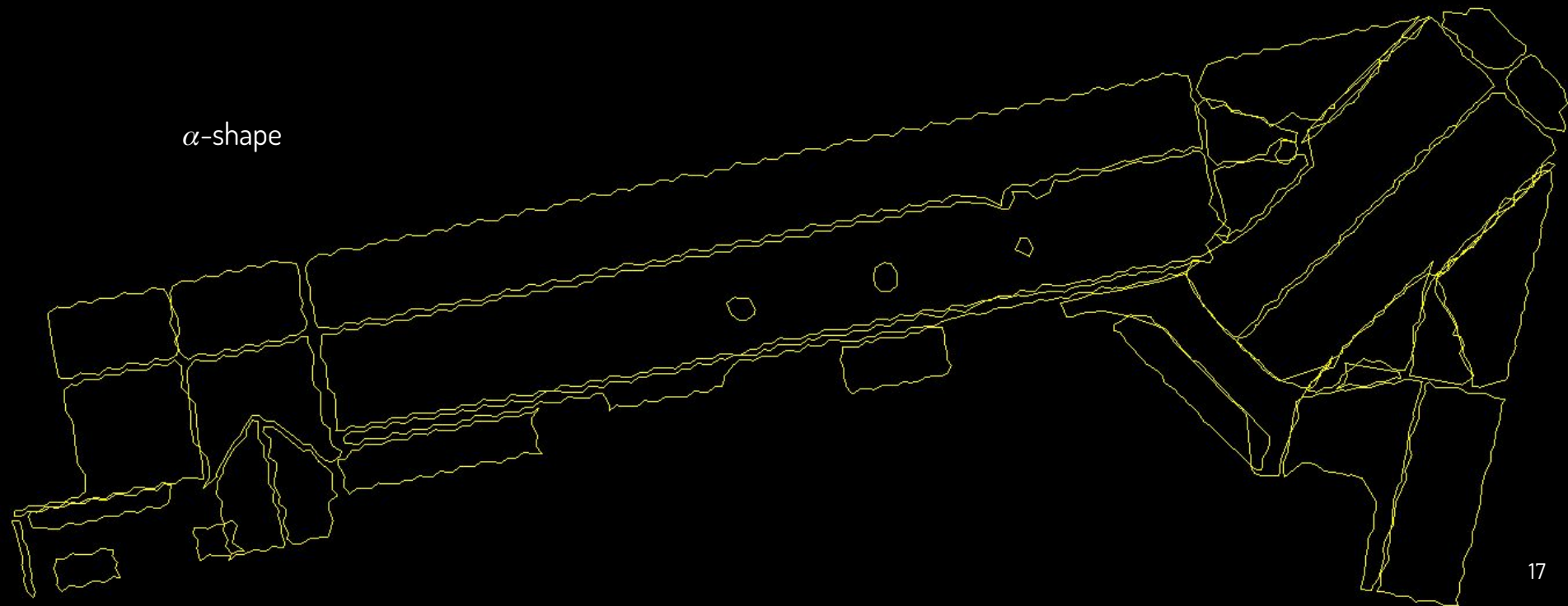


Feature extraction

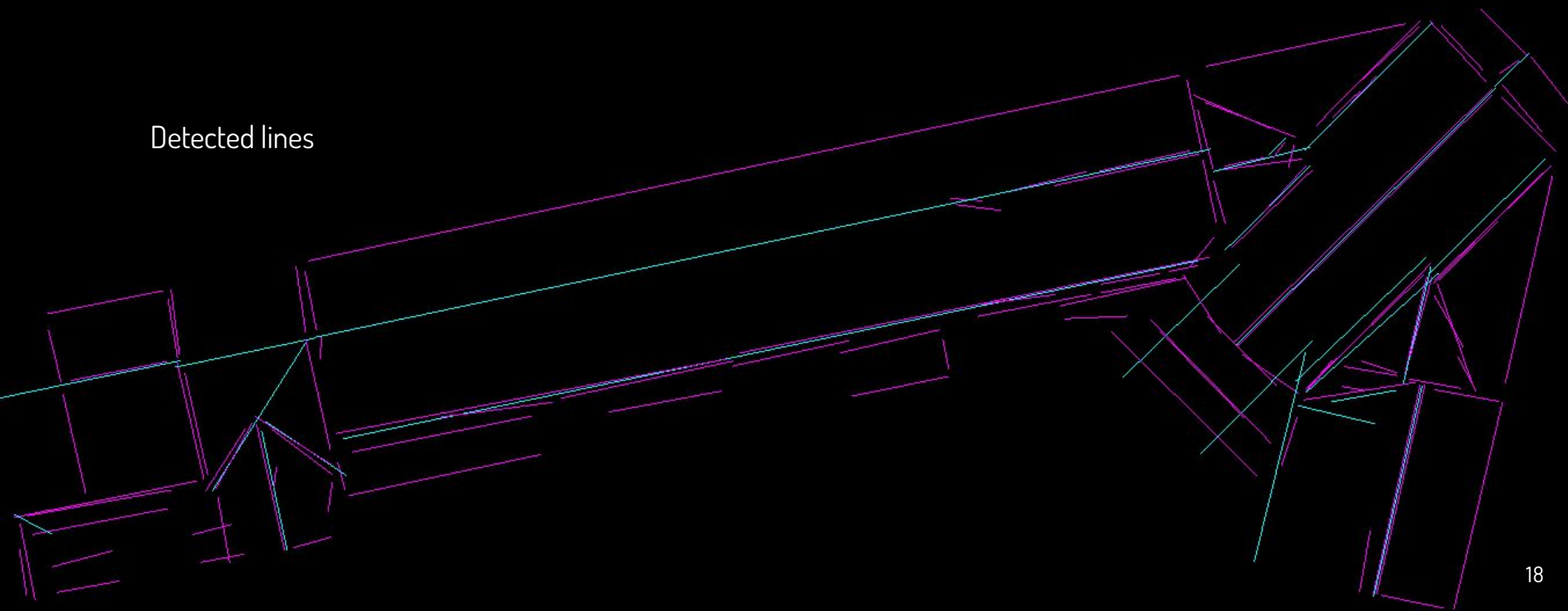
Detected planes



Feature extraction



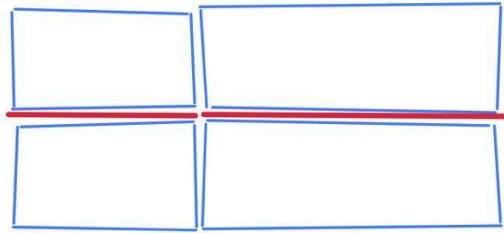
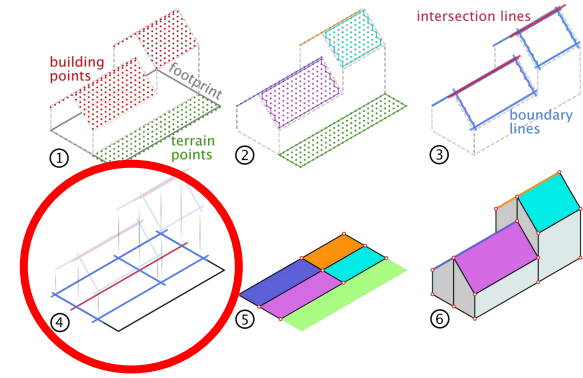
Feature extraction



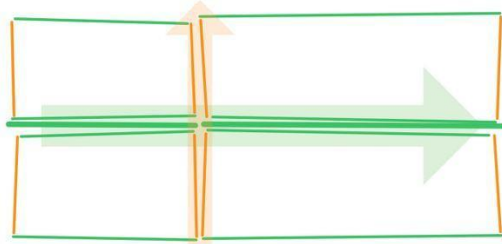
Line regularisation

Using 2-step hierarchical clustering

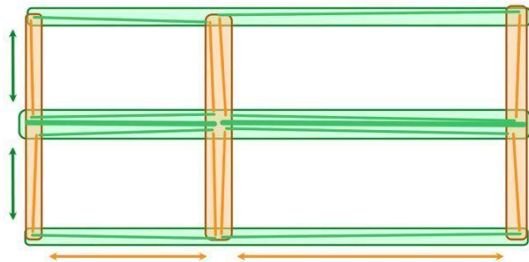
1. Based on line orientation
2. Based on euclidean distance within orientation clusters



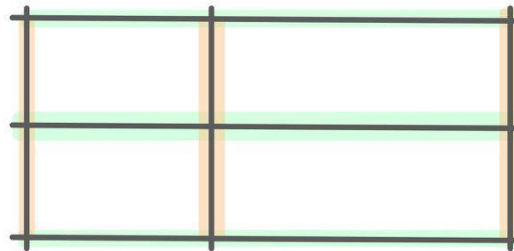
(a) Detected lines



(b) Orientation clustering



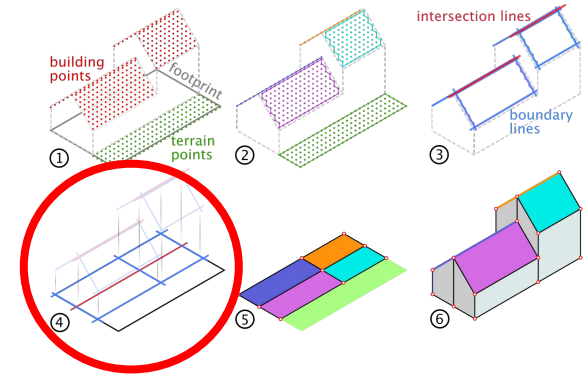
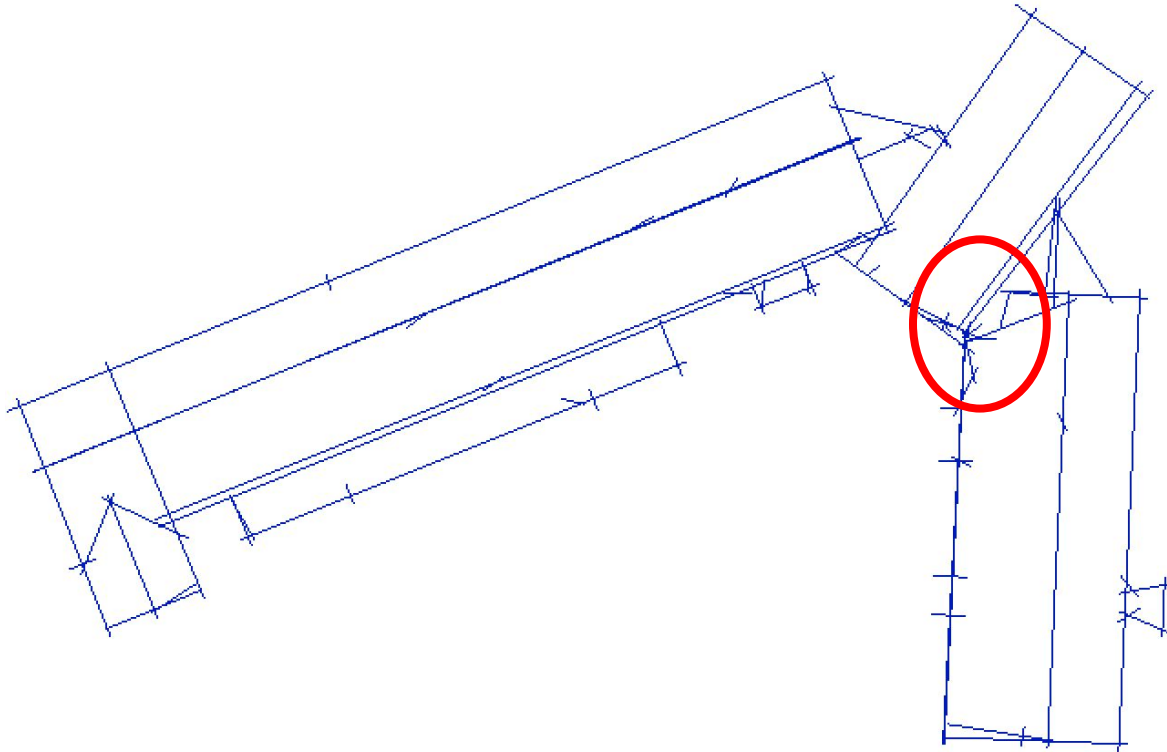
(c) Distance clustering



(d) Regularised lines

Initial roof partition

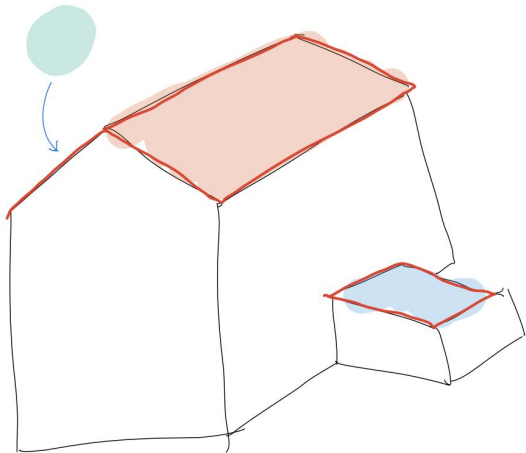
Still many small faces



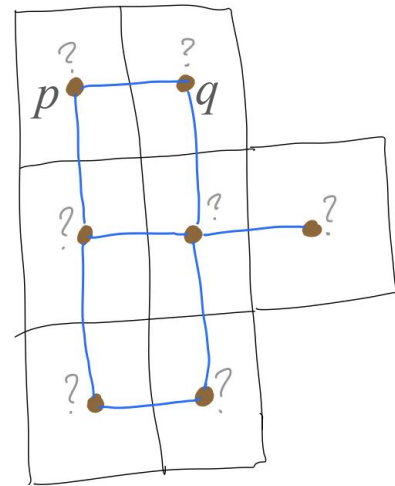
Graph-cut optimisation

[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



possible labels:



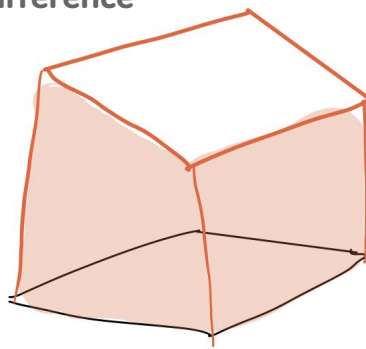
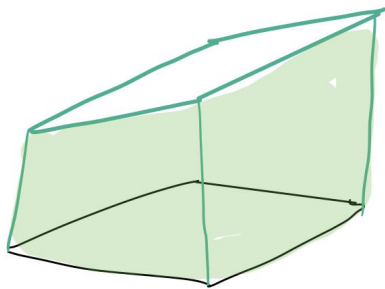
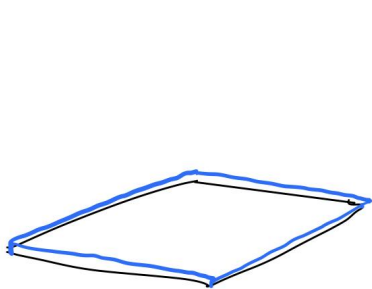
Graph-cut optimisation

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

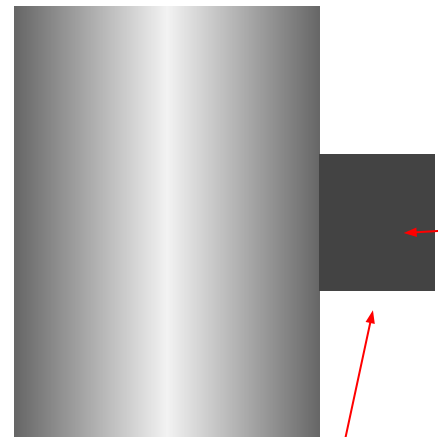
Data term:

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

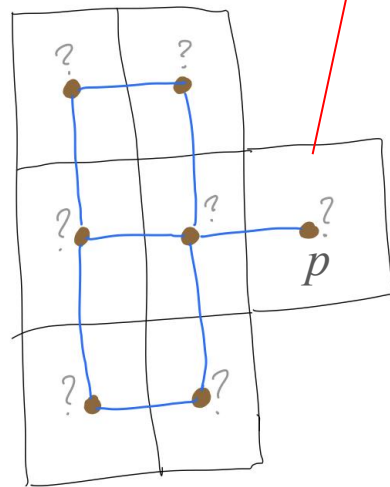
**Encourages good data fit
by selecting candidate plane with smallest volumetric difference**



Volume wrt each candidate plane



heightfield



Graph-cut optimisation

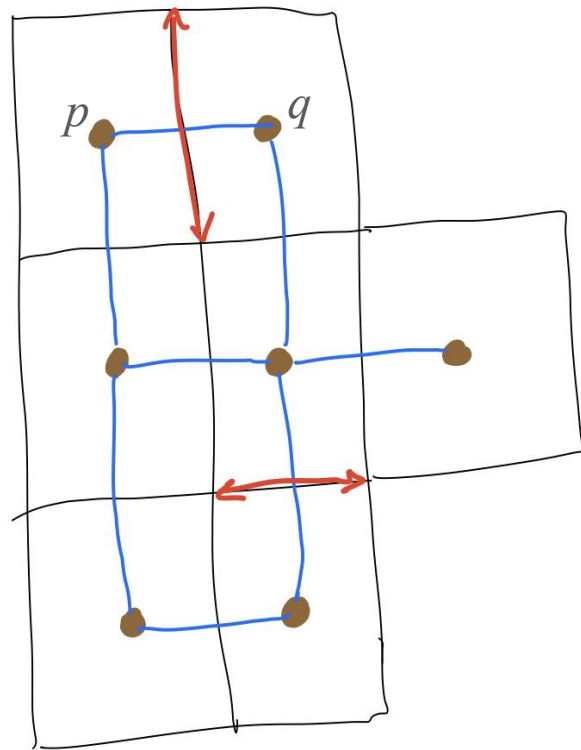
$$S_{p,q}(f_p, f_q) = \begin{cases} \text{length}(f_p, f_q) & \text{if } f_p \neq f_q \\ 0 & \text{if } f_p = f_q \end{cases}$$

Smoothness term:

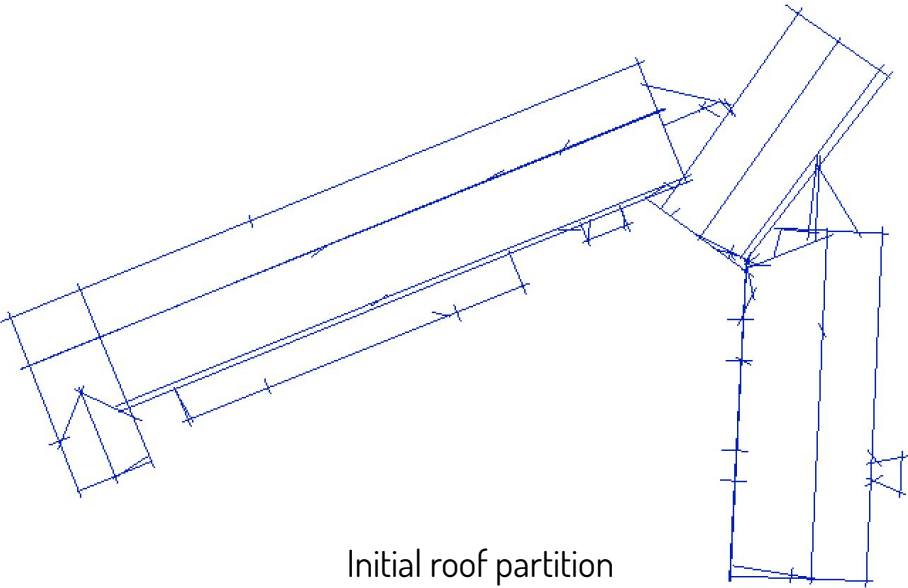
Edge length between adjacent faces

Minimisation entails:

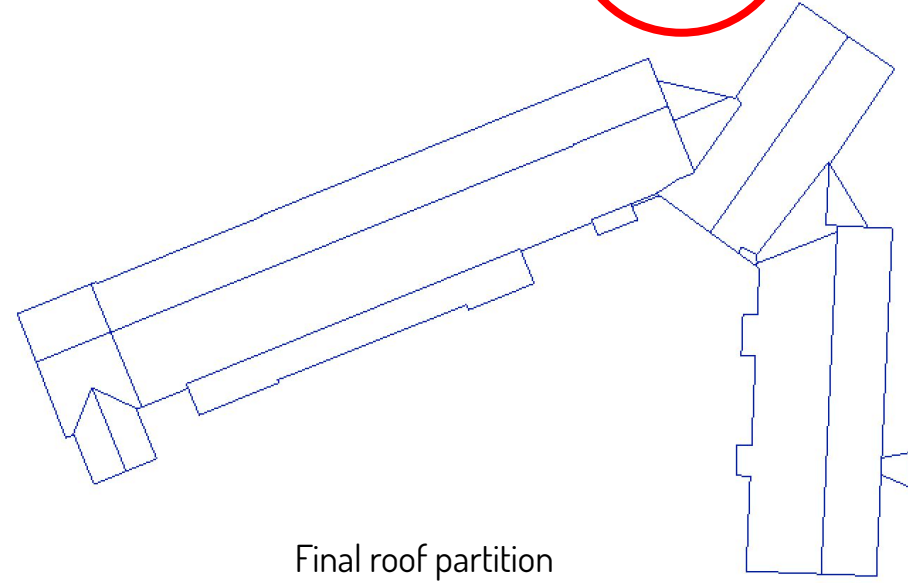
- Short and simple edges
- 'Merging' of faces,



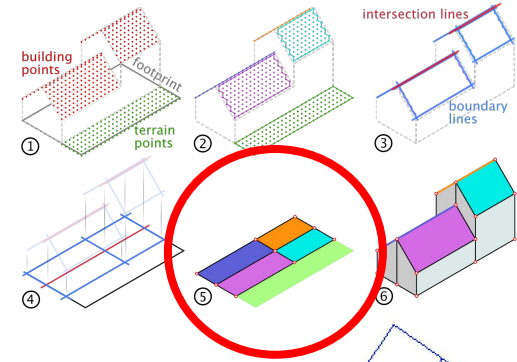
Final roof partition



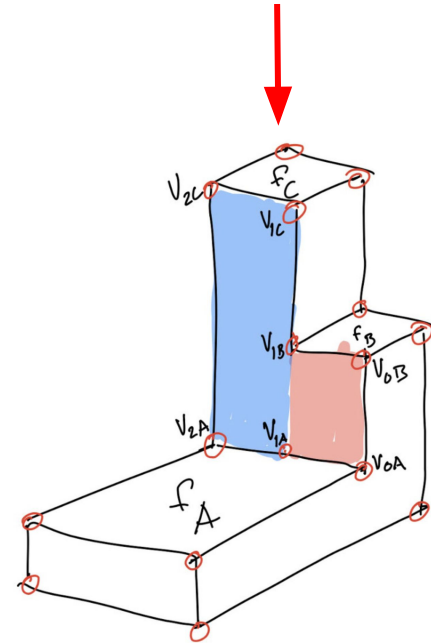
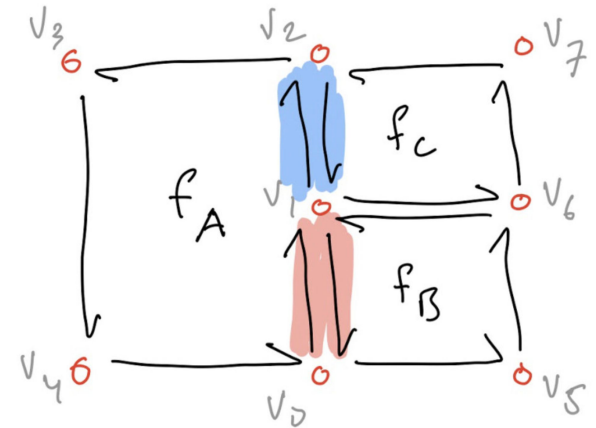
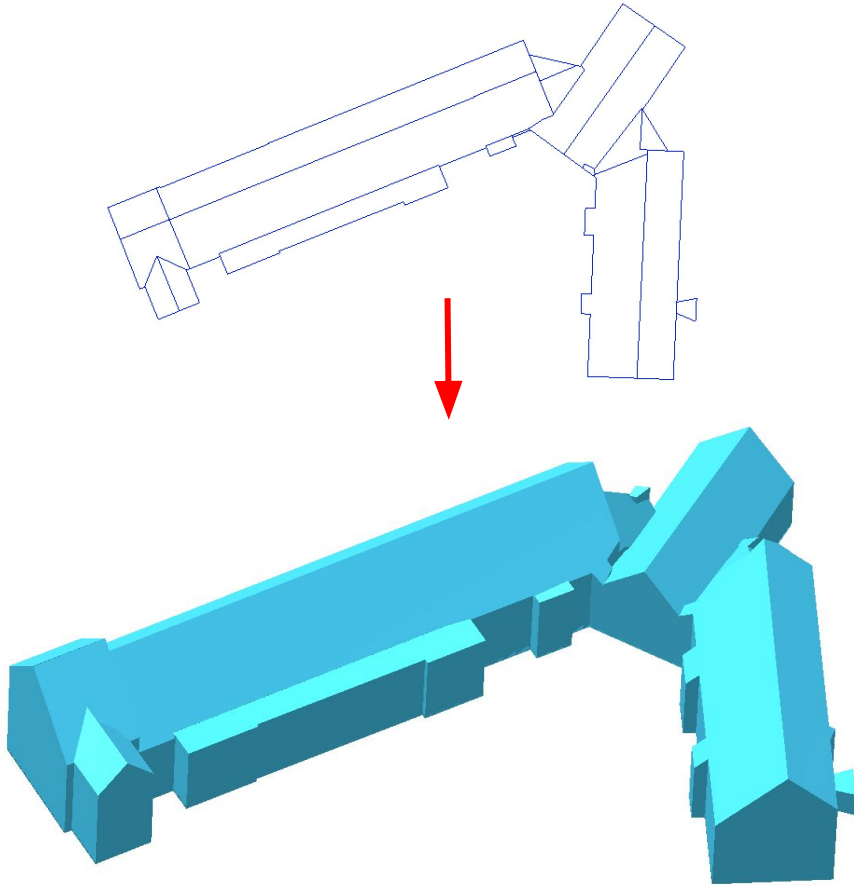
Initial roof partition



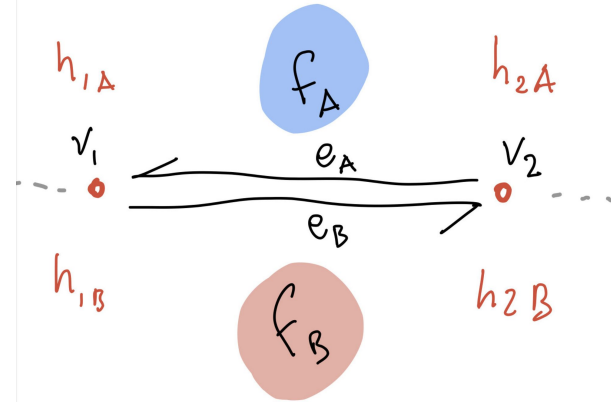
Final roof partition
(edges between equal plane labels dissolved)

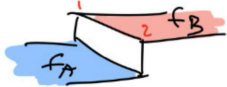
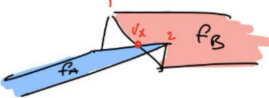
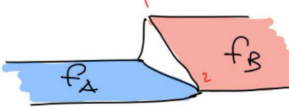


Extrusion

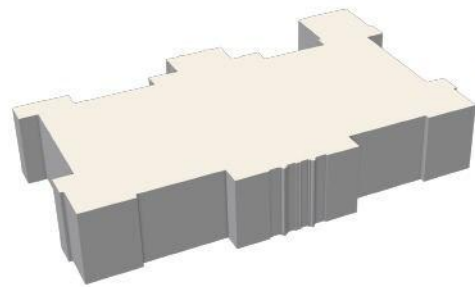


Extrusion

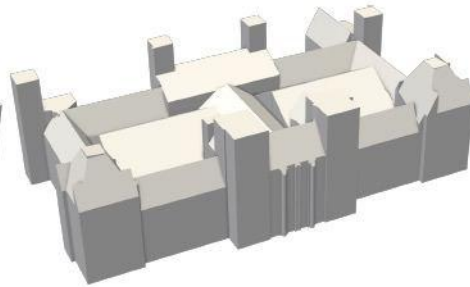


Case	condition	vertex order
	$h_{1,A} < h_{1,B}$ AND $h_{2,A} < h_{2,B}$	1. $v_{1B}, v_{1A}, v_{2A}, v_{2B}$
	$h_{1,A} < h_{1,B}$ AND $h_{2,A} > h_{2,B}$	1. v_{1B}, v_{1A}, v_x 2. v_{2A}, v_{2B}, v_x } 2 Faces!
	$h_{1,A} < h_{1,B}$ AND $h_{2,A} = h_{2,B}$	1. v_{1B}, v_{1A}, v_{2A}

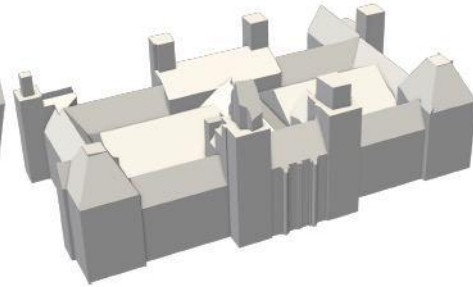
Effect of complexity term λ



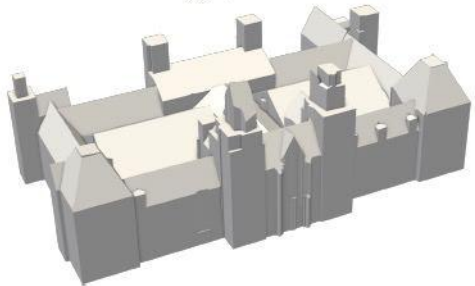
(a) $\lambda = 0.01$



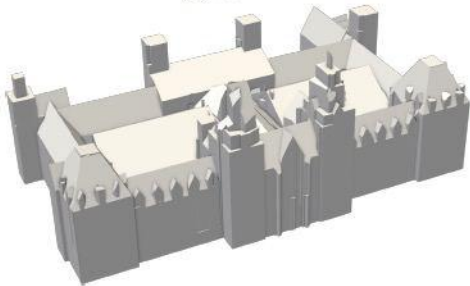
(b) $\lambda = 0.2$



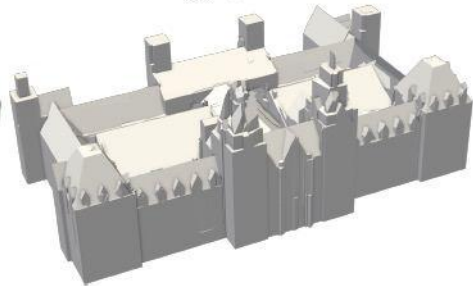
(c) $\lambda = 0.4$



(d) $\lambda = 0.6$



(e) $\lambda = 0.8$



(f) $\lambda = 0.99$

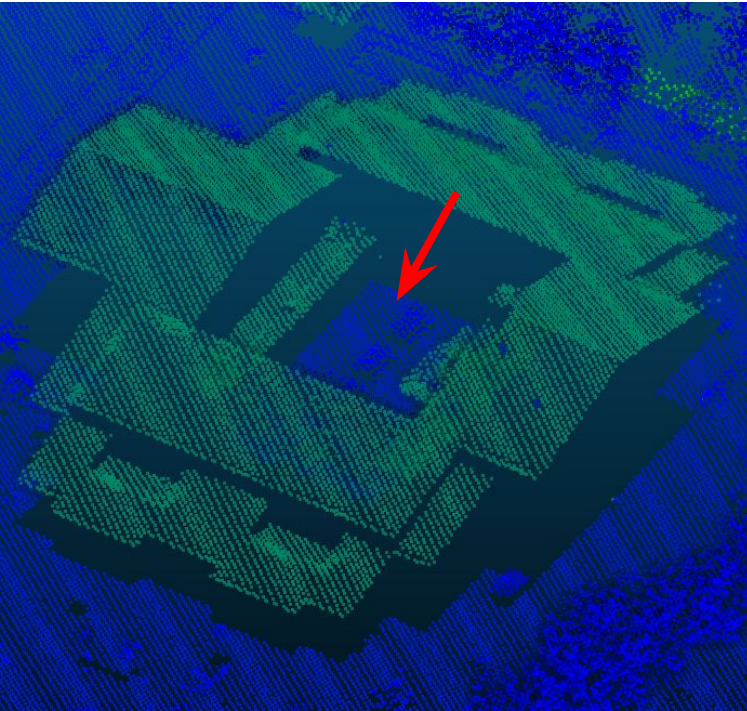
Image by Ivan Pađen

Special cases, Limitations

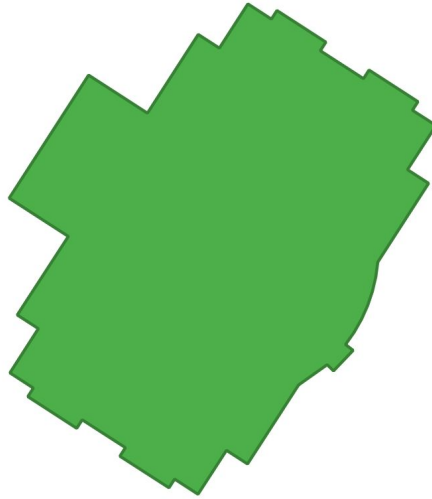
Groundparts

In some cases BAG footprint includes groundparts

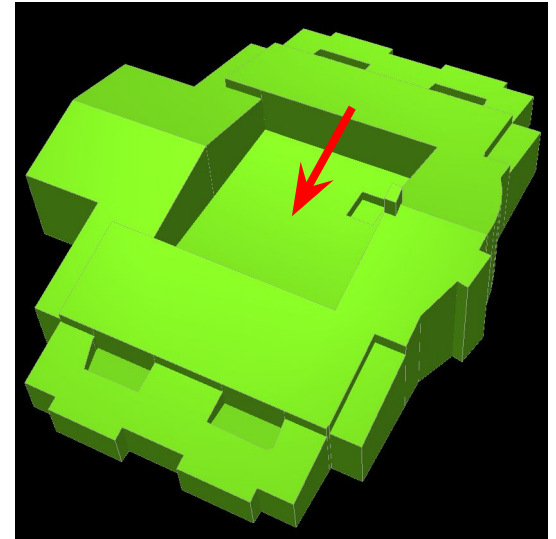
AHN3 ground and building class



BAG footprint



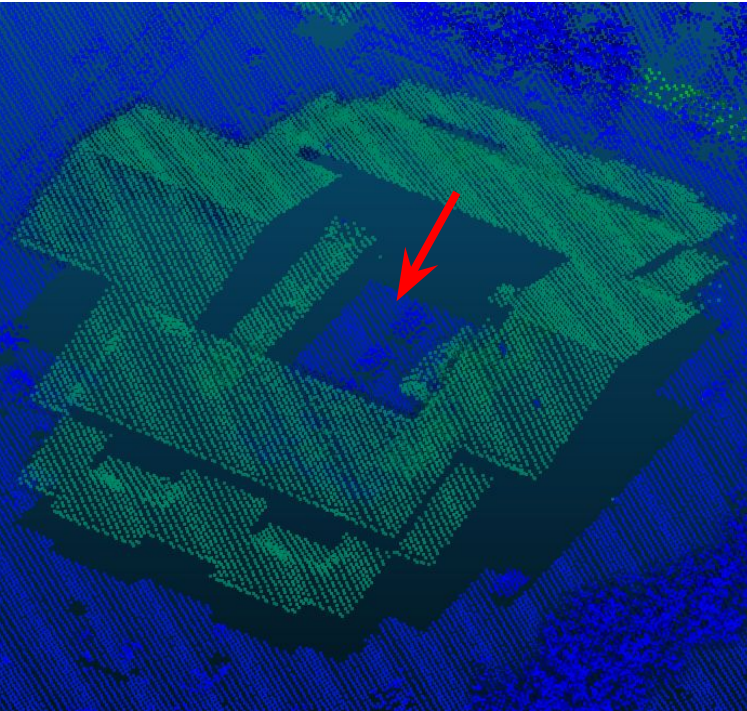
Reconstruction result:
roofplane fitted to groundpart



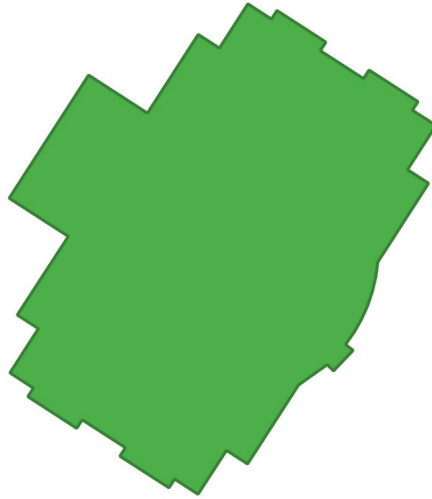
Groundparts

Reconstruction with groundpart detection

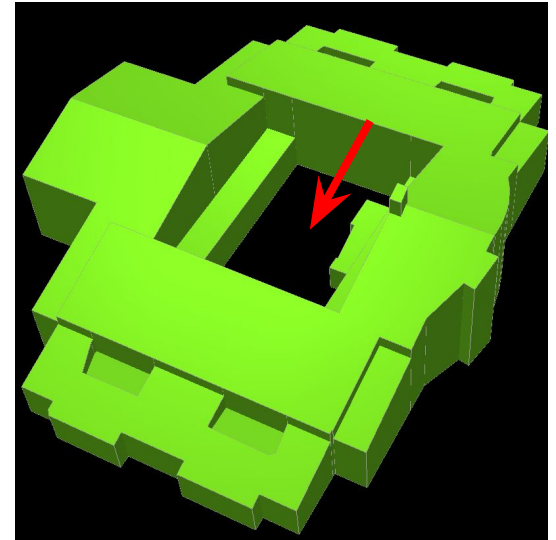
AHN3 ground and building class



BAG footprint

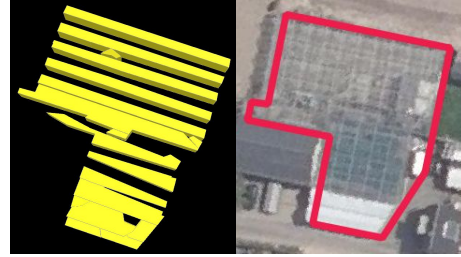


Reconstruction result:
groundpart removed from output



Limitation: glass roofs

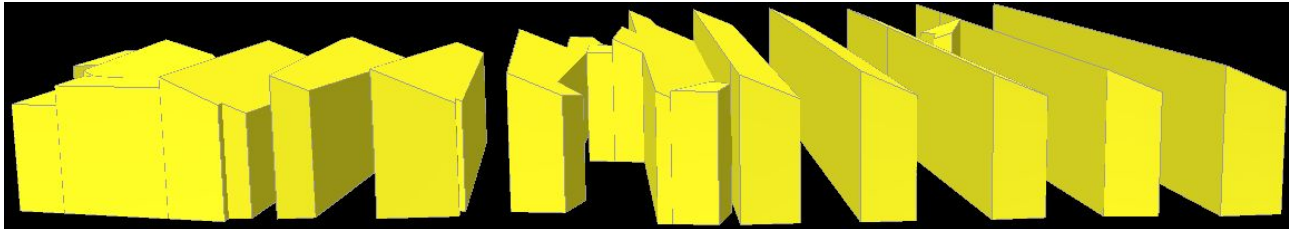
Green houses: both points on ground and on roof



AHN3
ground and building class



Heightfield



Reconstruction result

Limitation: glass roofs

Green houses: both points on ground and on roof

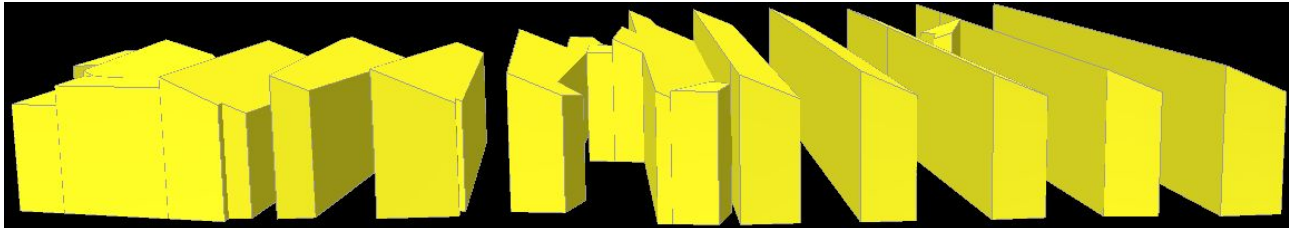
Current solution:
Use greenhouse classification from
TOP10NL, reconstruct as LoD 1.2



AHN3
ground and building class



Heightfield



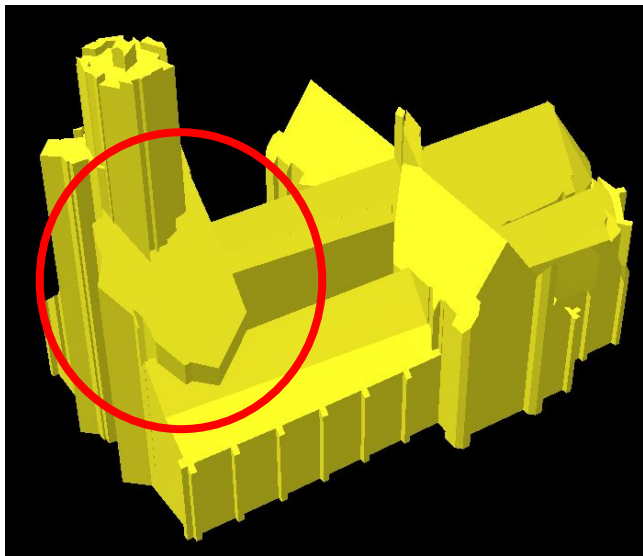
Reconstruction result

Spherical surfaces

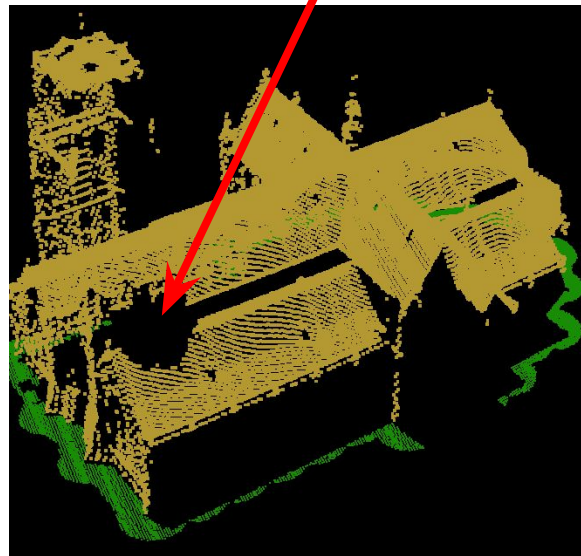
Are approximated with planar surfaces if sufficient point density



Limitations: occlusion/no-data



Reconstruction

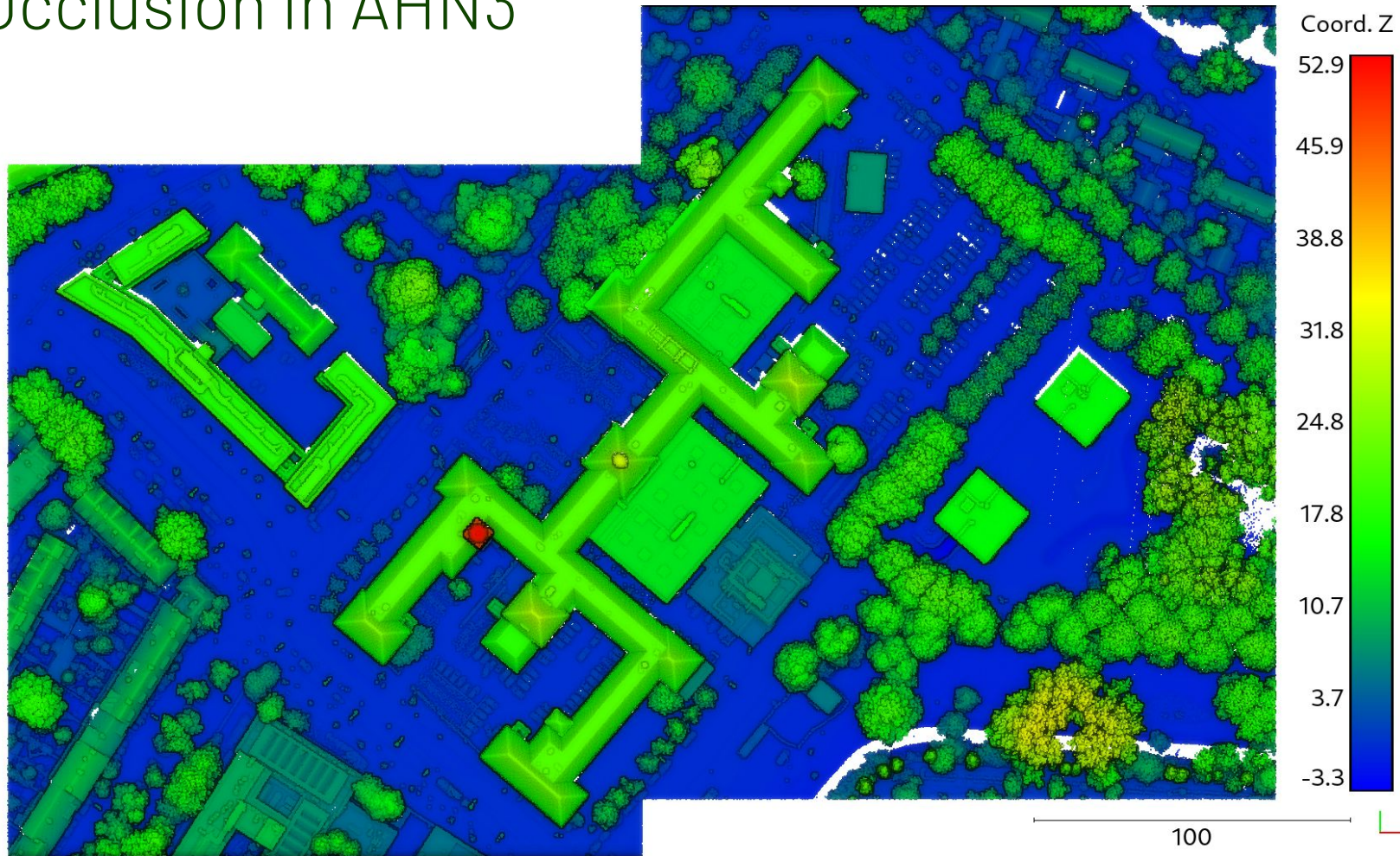


AHN3 ground and building class



Heightfield

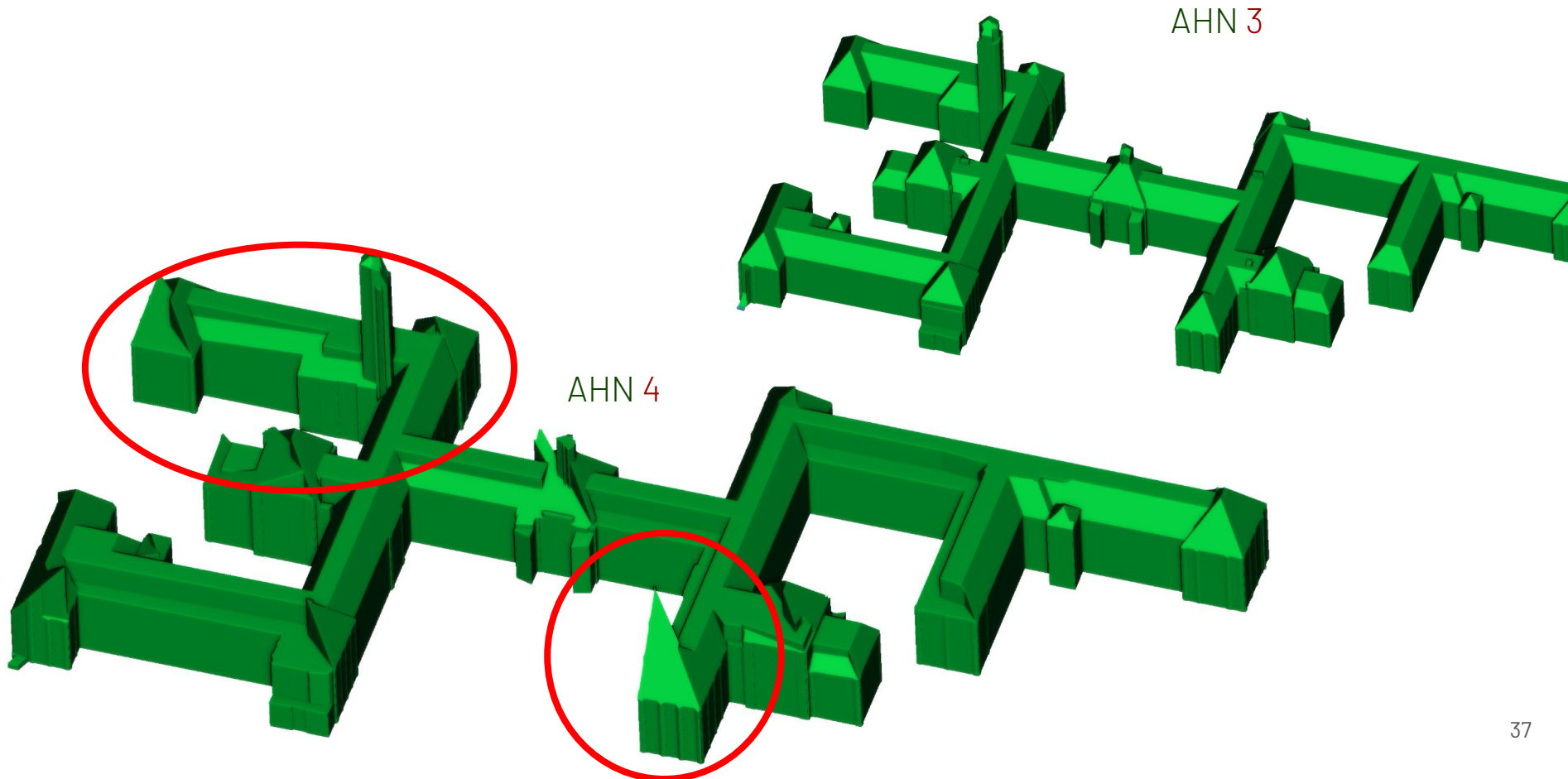
Occlusion in AHN3



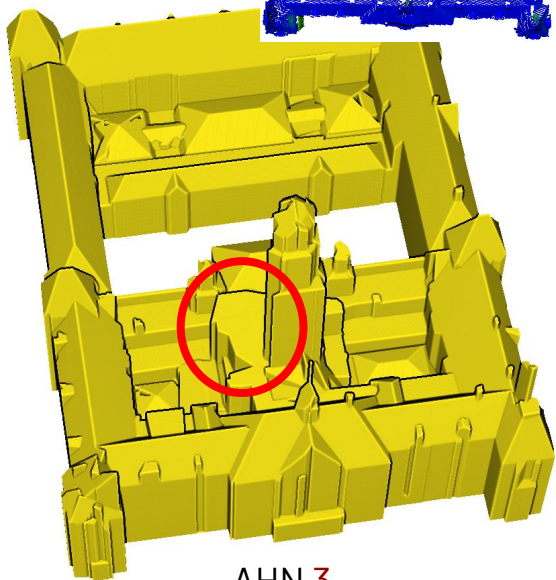
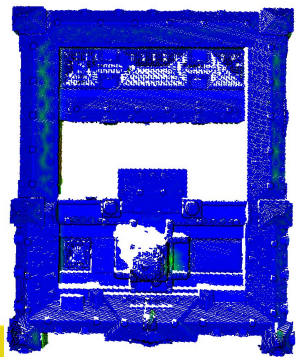
Occlusion in AHN4



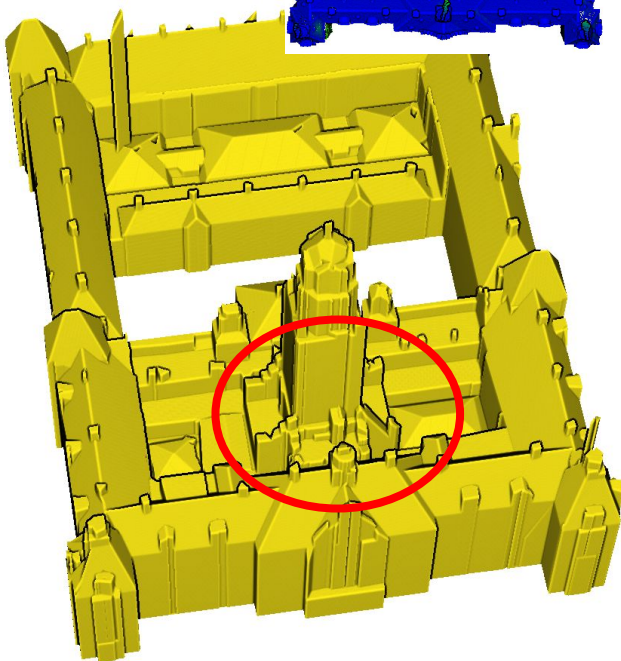
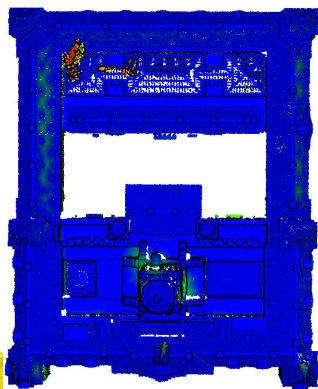
Occlusion effect on reconstruction



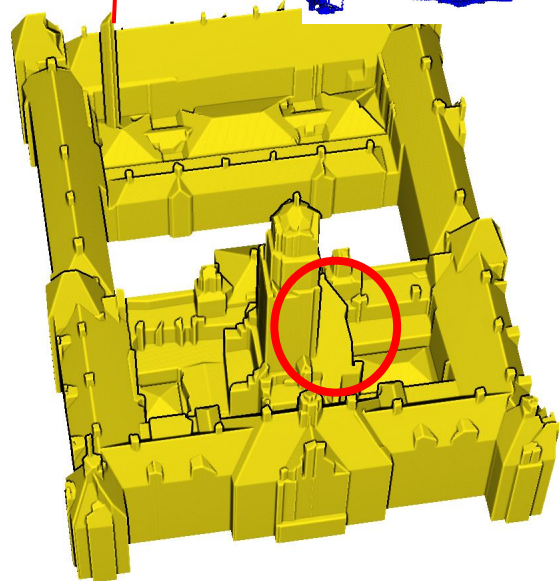
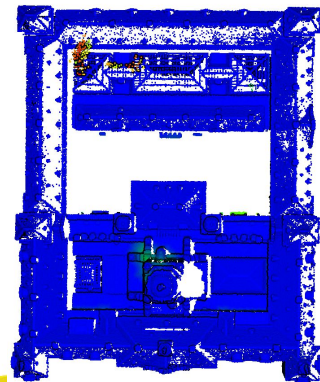
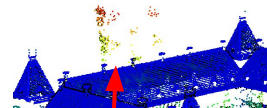
Fuse two point clouds



AHN 3



AHN 3+4

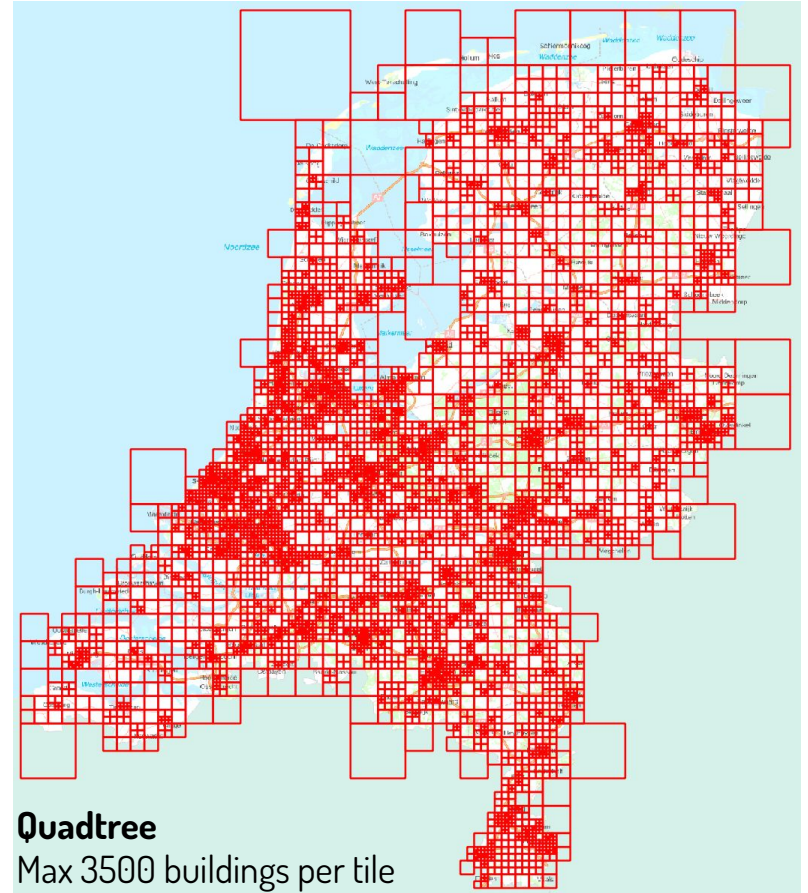
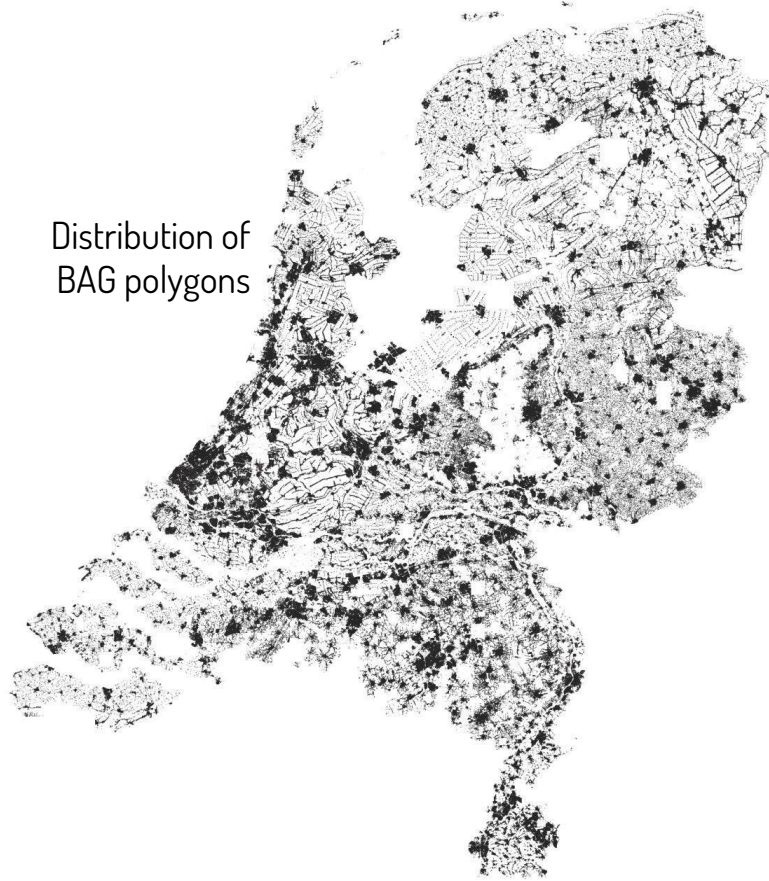


AHN 4

Data management

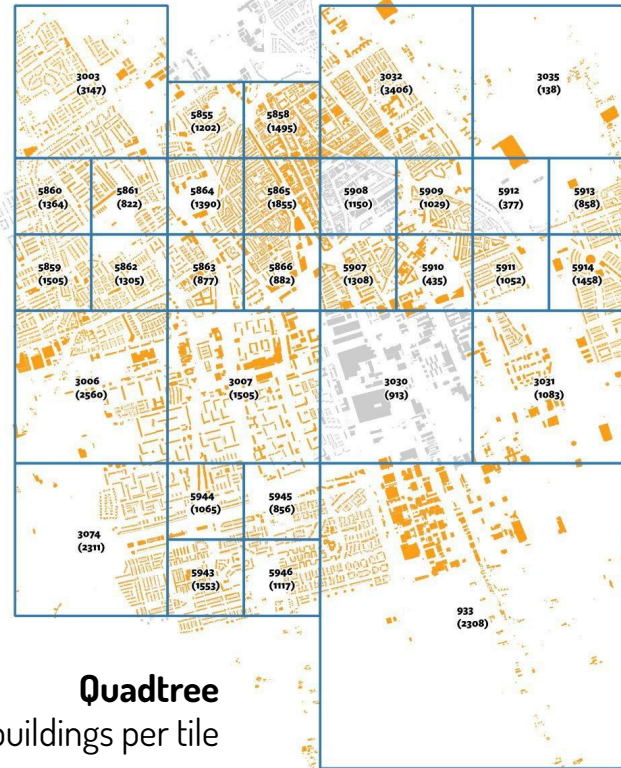
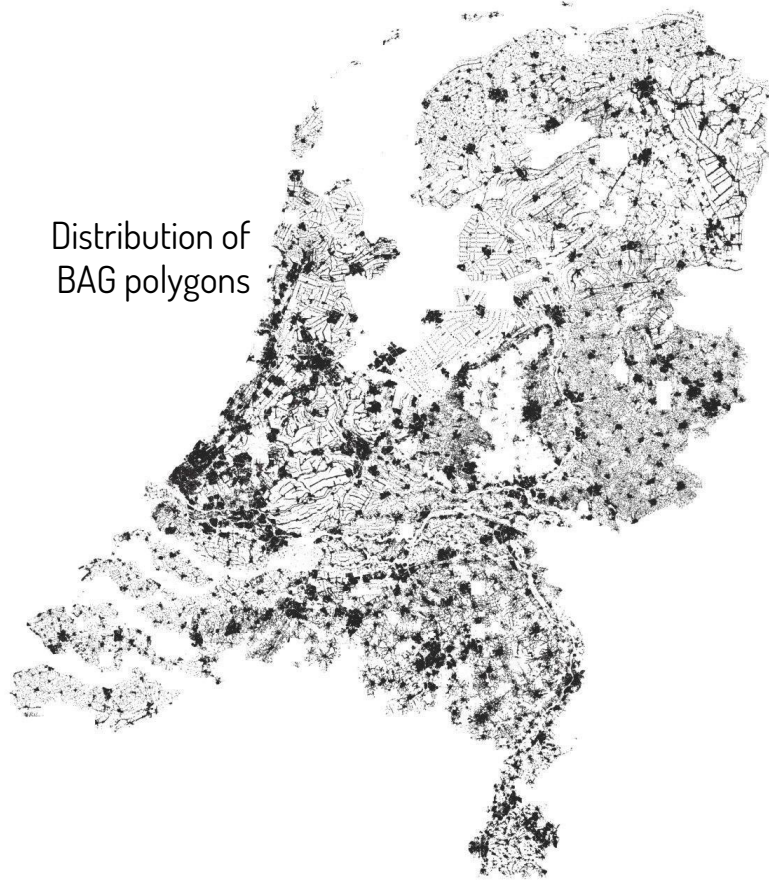
How to tile the data?

Distribution of
BAG polygons



How to tile the data?

Distribution of
BAG polygons



Quadtree

Max 3500 buildings per tile

3DBAG tech stack



3DBAG repositories

<https://github.com/3DBAG>

The screenshot shows the GitHub profile and repository page for 3DBAG. The header includes the 3DBAG logo, a search bar, and navigation links for Overview, Repositories (13), Discussions, Projects (2), Packages, Teams, People (5), Insights, and Settings. The profile section features a profile picture, the name '3DBAG', 26 followers, location 'Netherlands', website 'https://3dbag.nl/en/viewer', and email 'info@3dbag.nl'. An 'Unfollow' button is present. The main content area displays the 'README.md' file, which describes the 3DBAG GitHub organization and lists various tools and APIs. The right sidebar shows 'View as: Public', a note about the README, 'Get started with tasks', 'Top discussions this past month', 'People' (with avatars and an 'Invite someone' button), and 'Top languages' (Python, HTML, Rust, CMake, Jupyter Notebook).

3DBAG

26 followers Netherlands <https://3dbag.nl/en/viewer> info@3dbag.nl Unfollow

README.md

3DBAG

This is the GitHub organisation of the 3DBAG Innovation Platform. This place is used for collaborating on the software that used for producing the 3DBAG, for organising the 3DBAG development and communicating with the 3DBAG users and contributors.

The main platform for communication are the [Discussions](#). If you have any questions or comments about the 3DBAG in general, about a specific software or you spotted an error, please review first if you question has been answered already. If not, don't hesitate to open a new discussion.

The software that you will find in this organisation:

- [3dbag-api](#): OGC Features API that serves the 3DBAG CityJSON data at <https://api.3dbag.nl/>
- [3dbag-dashboard](#): Quality dashboard for the 3DBAG at <https://3dbag.nl/en/dashboard>
- [3dbag-docs](#): The 3DBAG documentation at <https://docs.3dbag.nl/en/>
- [3dbag-pipeline](#): Production pipeline that orchestrates source data ingestion, preprocessing, building reconstruction, postprocessing and deployment
- [3dbag-viewer](#): The 3D viewer at <https://3dbag.nl/en/viewer>
- [CityBuf](#): Experimental CityJSON encoding as FlatBuffers
- [geodepot](#): Test data management system for geodata
- [geodepot-api](#): API to a geodepot repository
- [roofer](#): Automatic LoD2.2 building reconstruction that generates the 3D building models of the 3DBAG

View as: Public

You are viewing the README and pinned repositories as a public user.

[Get started with tasks](#) that most successful organizations complete.

Top discussions this past month

Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

[Start a new discussion](#)

People

Invite someone

Top languages

Python HTML Rust CMake Jupyter Notebook

Pinned

[3dbag-viewer](#) Public

[roofer](#) Public

Customize pins

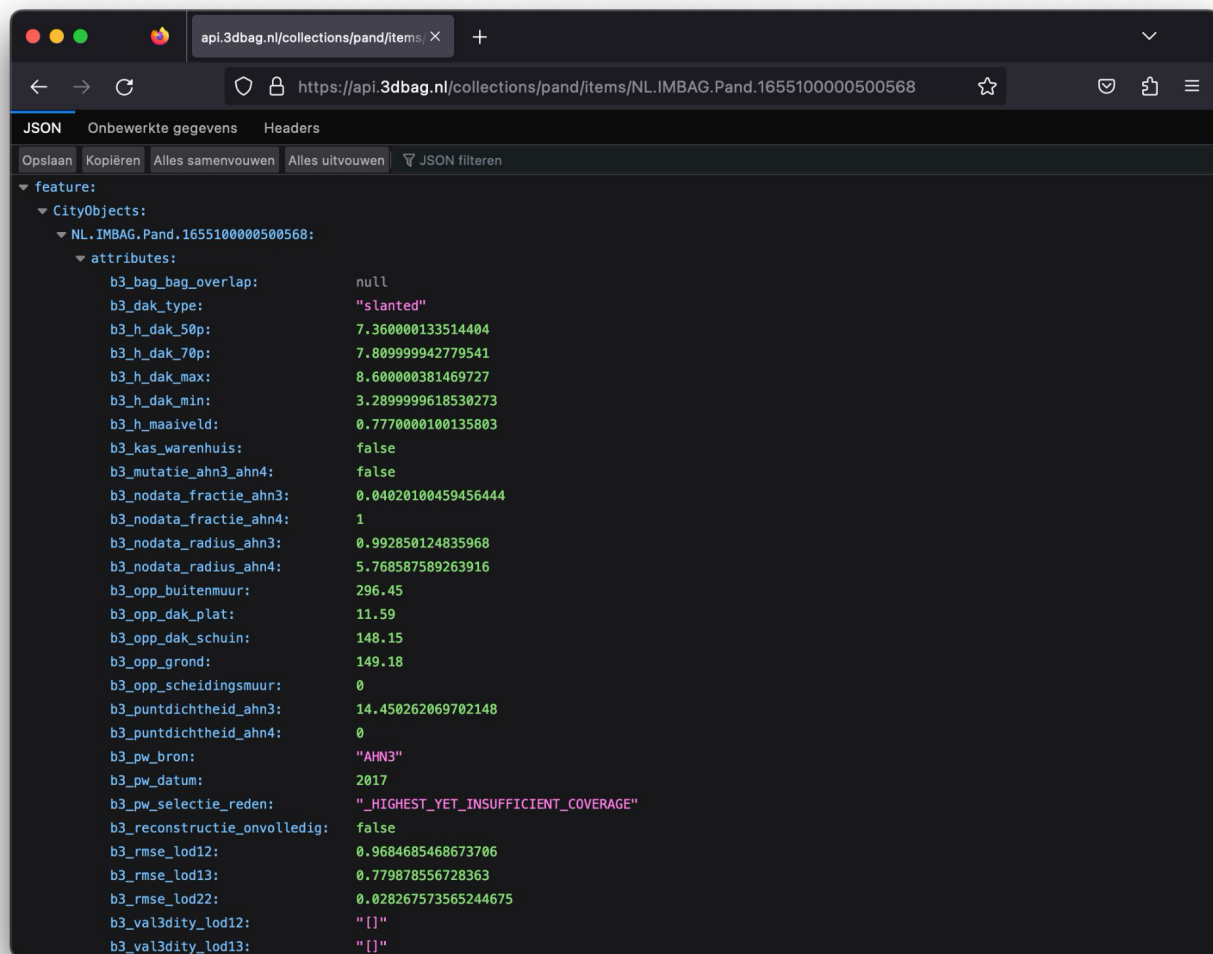
3DBAG additions

3DBAG API

Request directly CityJSON
Features by

- Building ID
- Bounding box

Based on [CJDB](#)
(2022 geomatics synthesis project)



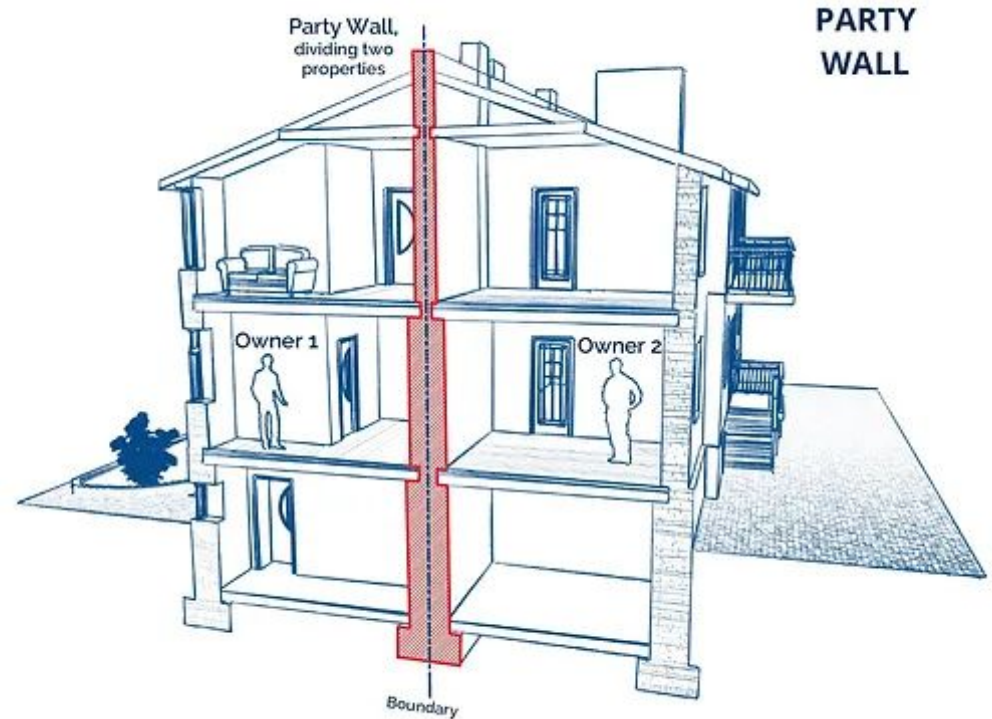
Party wall areas

We calculate and include:

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

Needed for eg. energy label estimation

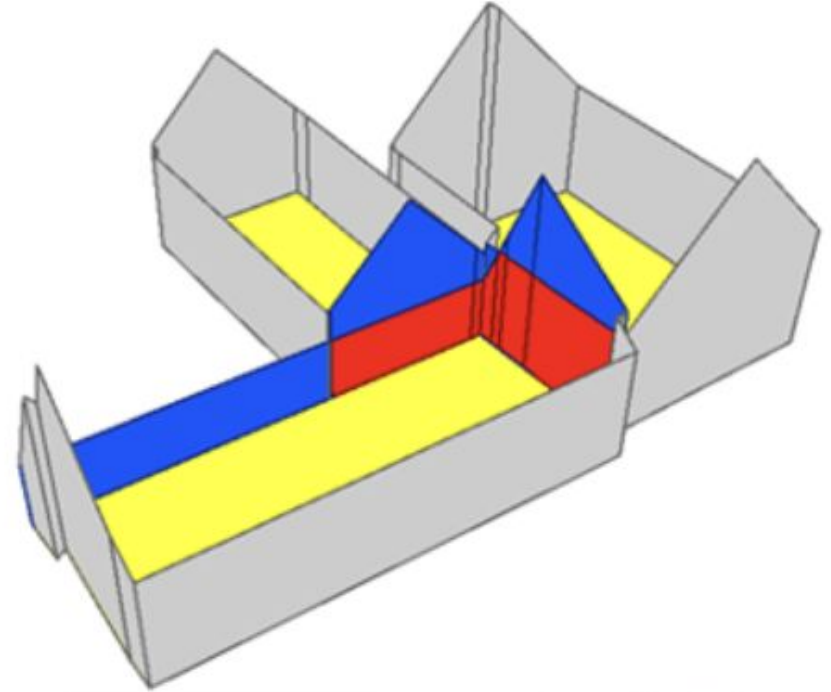
Sponsored by RVO



Calculation of party walls

For each building

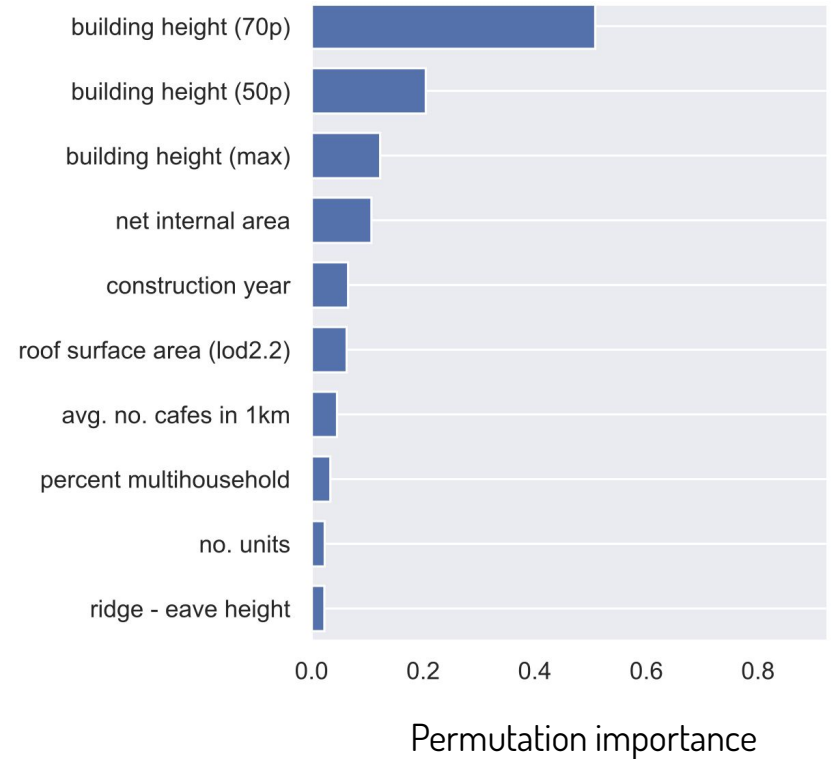
1. Find neighbouring buildings
2. Extract all 3D wall polygons
3. Find co-planar polygons through clustering by plane parameters
4. Intersect the co-planar wall polygons from different buildings.
5. Calculate area of intersection



Estimation nr of floors

Based on MSc thesis of [Ellie Roy \(Geomatics 2022\)](#)

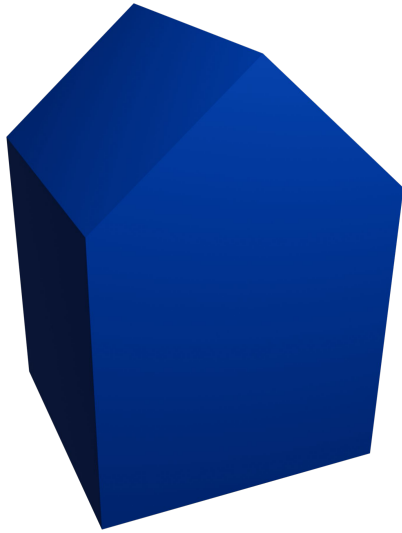
- Machine learning method based on Gradient Boosting Regression
- Model trained on groundtruth data from a couple of Dutch municipalities
- Used features collected from various datasets (3DBAG, CBS, ...)
- Accuracy drops for >5 floors
- Available in 3DBAG release v2024.02.28



Underpasses

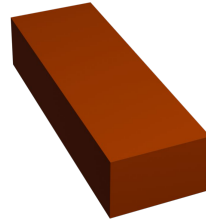


Add underpasses through boolean difference



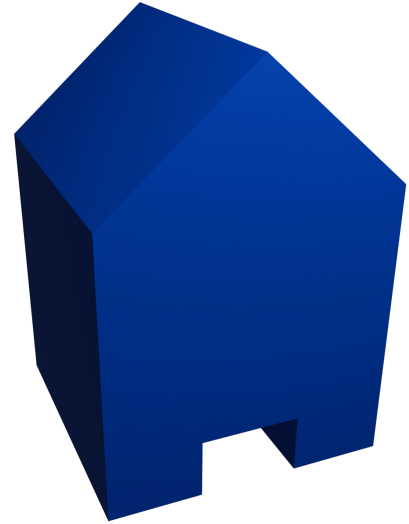
Existing 3DBAG model

-



Underpass volume

=



3DBAG model with underpass

Underpasses: 2D + height

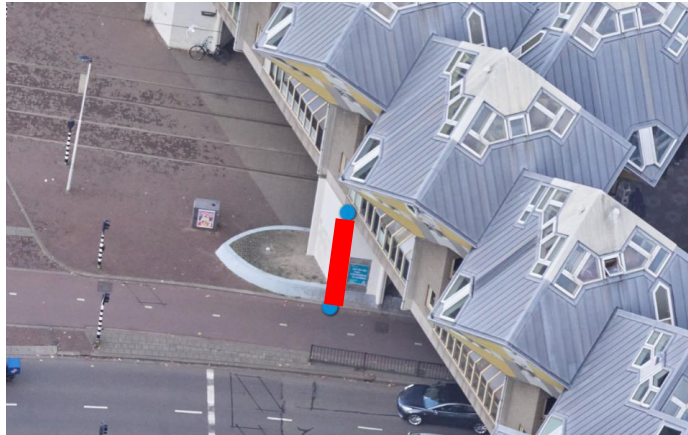
2D geometry

1. From existing 2D mapdata (BAG+BGT)

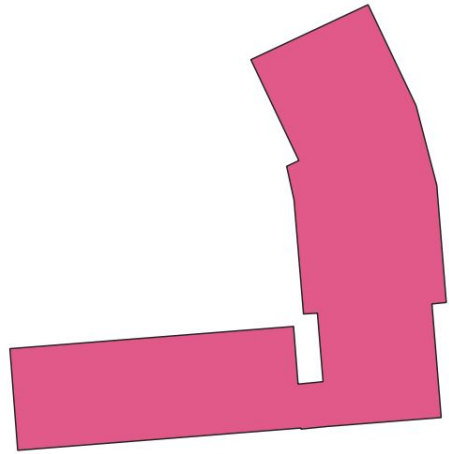


Height - 3 possibilities:

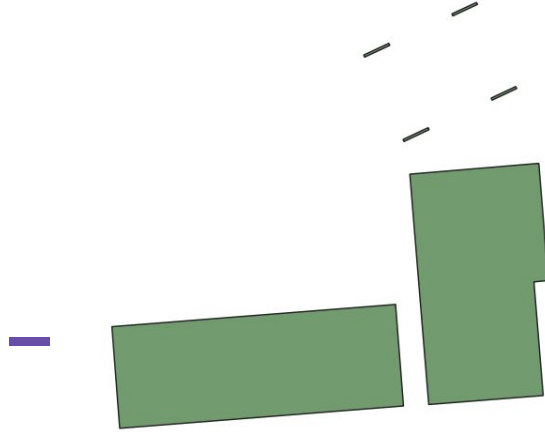
1. Measurements: streetlidar, oblique images
2. Assumptions: eg. "underpass is minimum 2.5 m"
3. Crowdsourcing via 3DBAG viewer



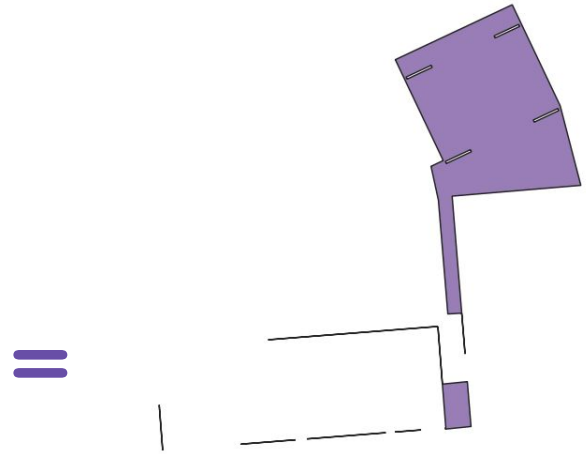
2D underpass detection



BAG dataset

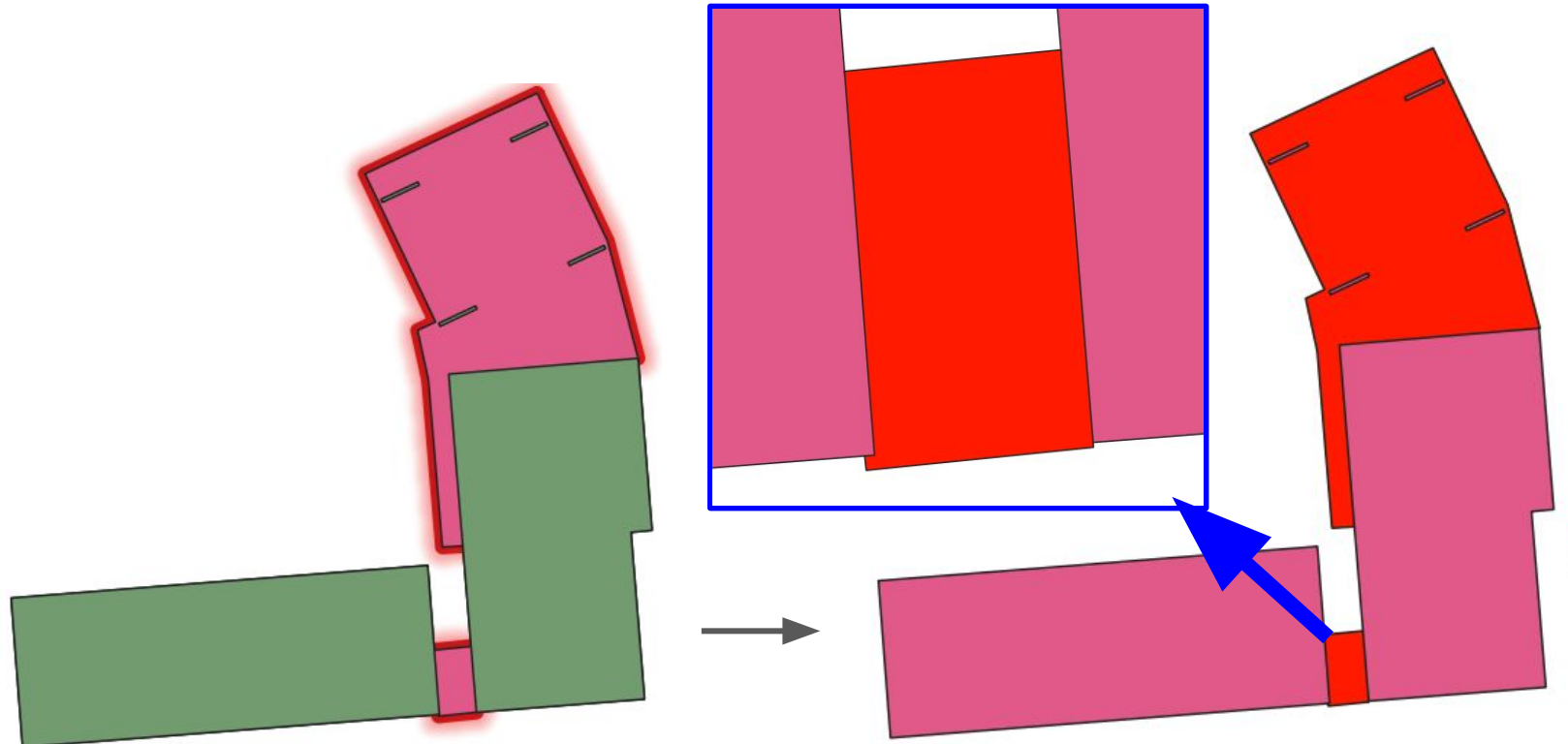


BGT dataset



Underpass (before cleaning)

2D underpass processing



Classification interior/exterior edges

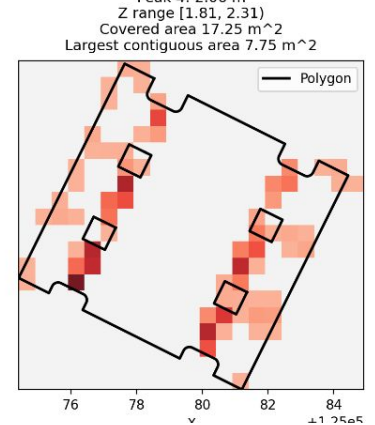
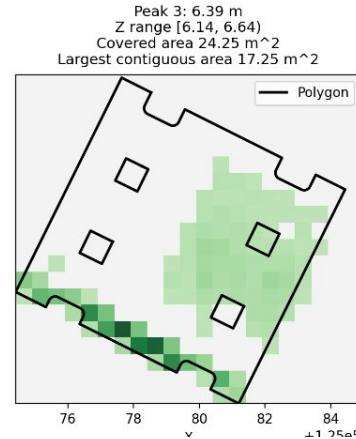
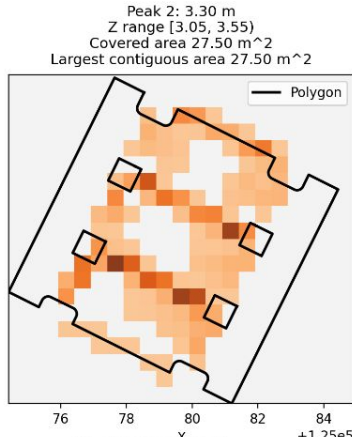
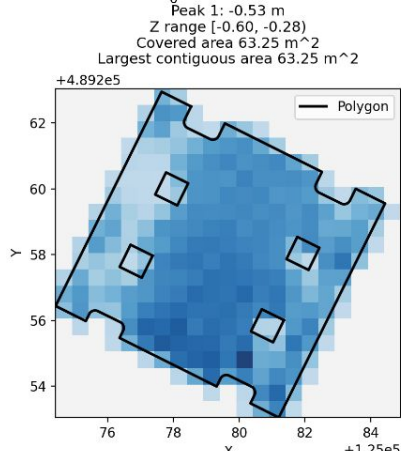
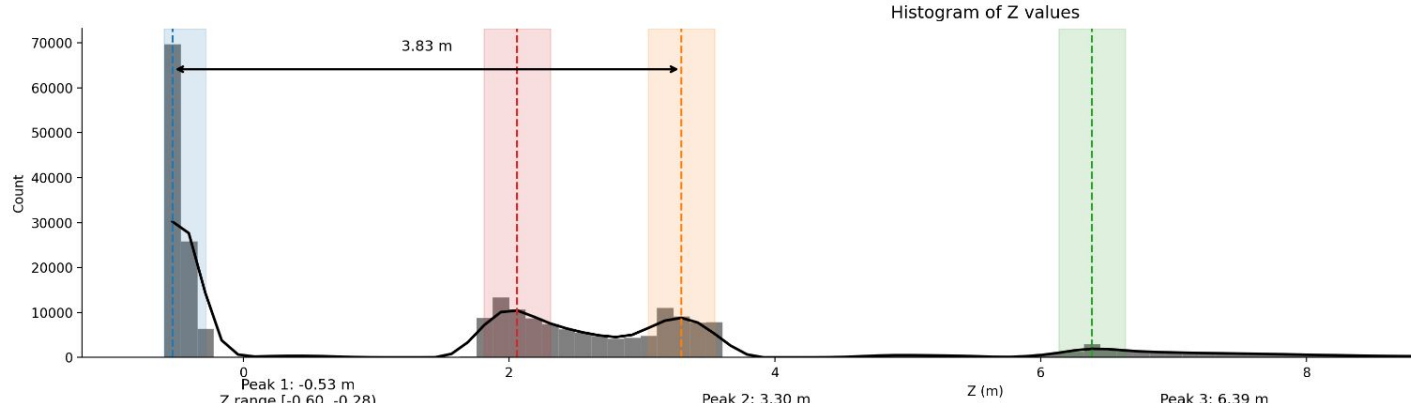
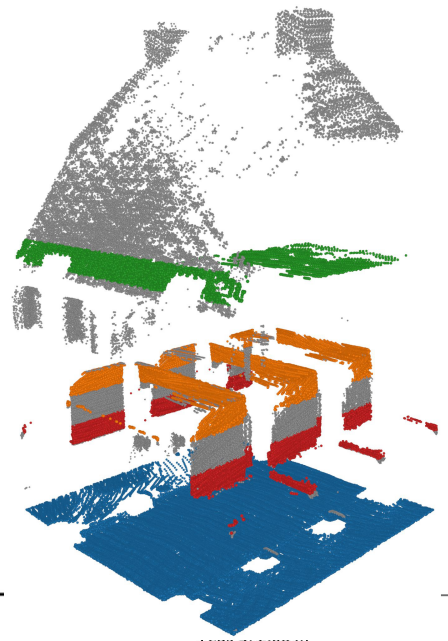
Exterior edge of setting

Detectie in Rotterdam



Height estimation from streetlidar

NL.IMBAG.Pand.0363100012146576

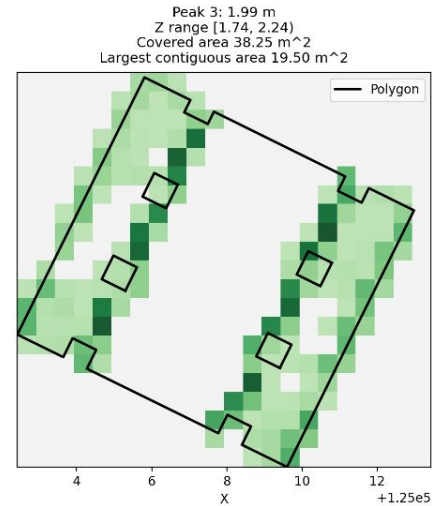
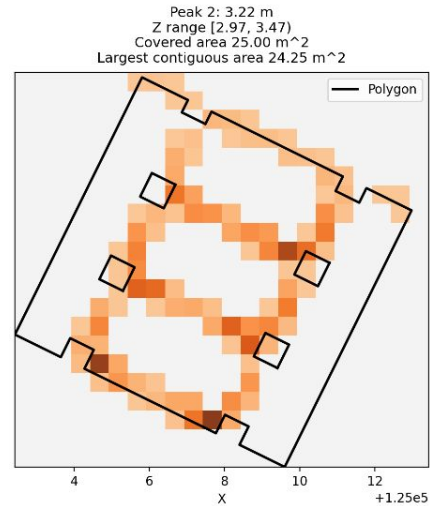
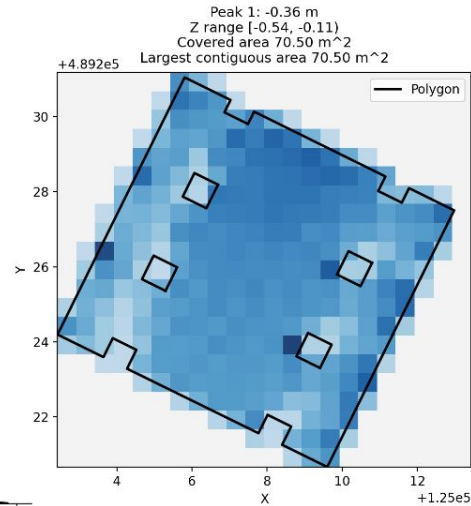
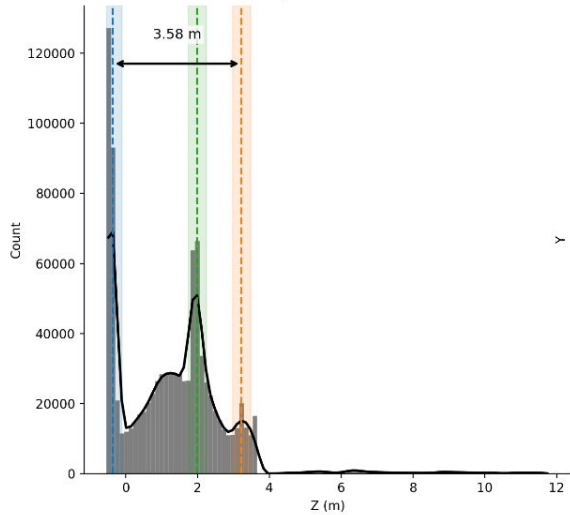


Height estimation



NL.IMBAG.Pand.0363100012095711

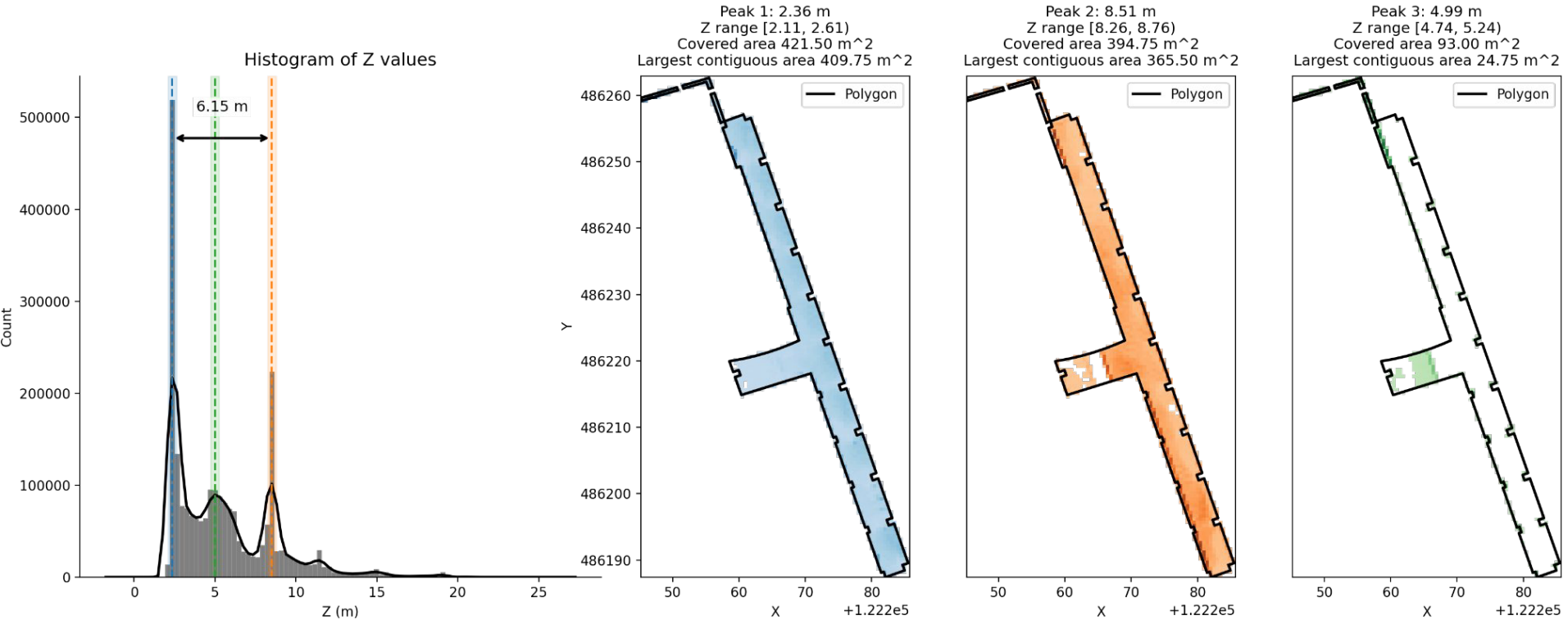
Histogram of Z values



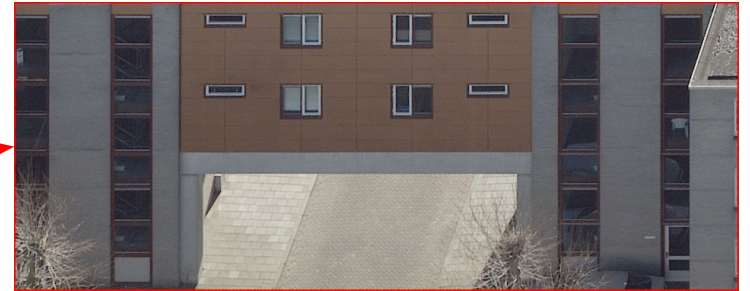
Height estimation



NL.IMBAG.Pand.0363100012170850

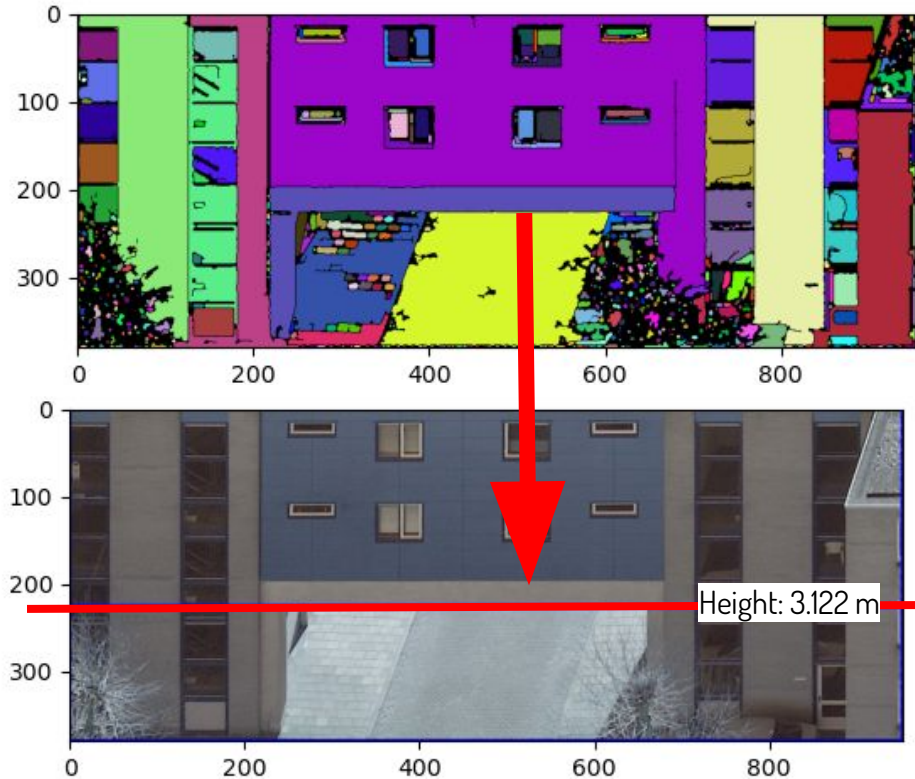


Height estimation from oblique images

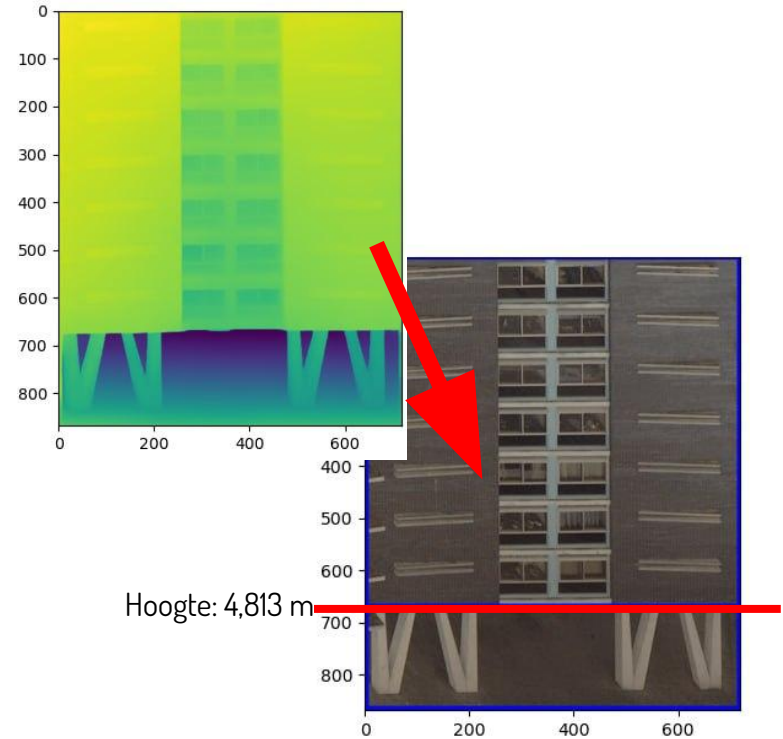


Perspective projection & image rectification

Height estimation from oblique images

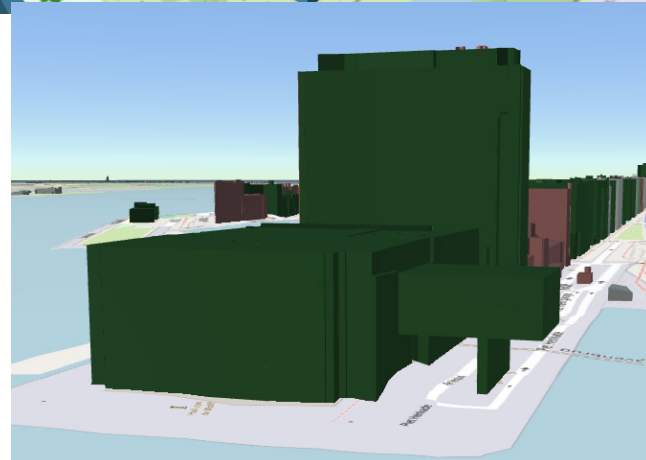
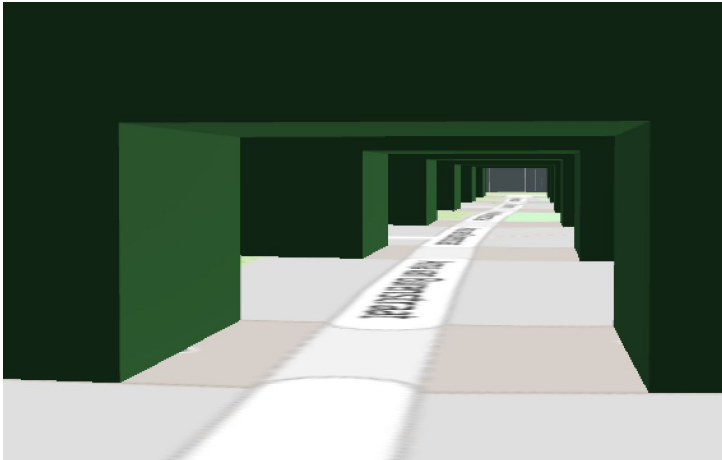
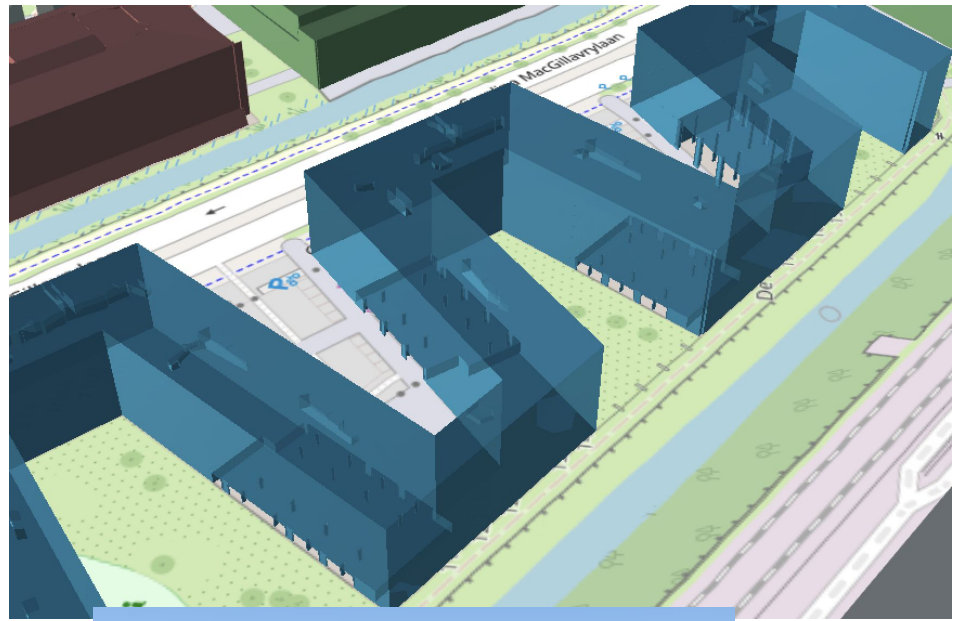
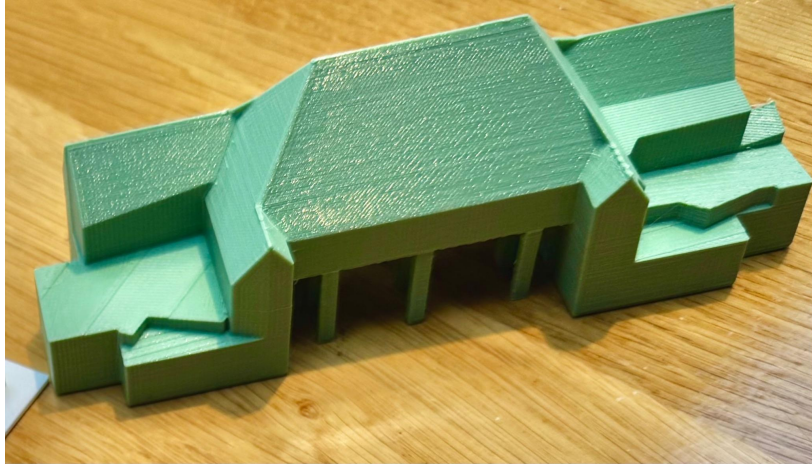


A) edge detection and image segmentation



B) depth estimation with AI (DepthAnything v2)

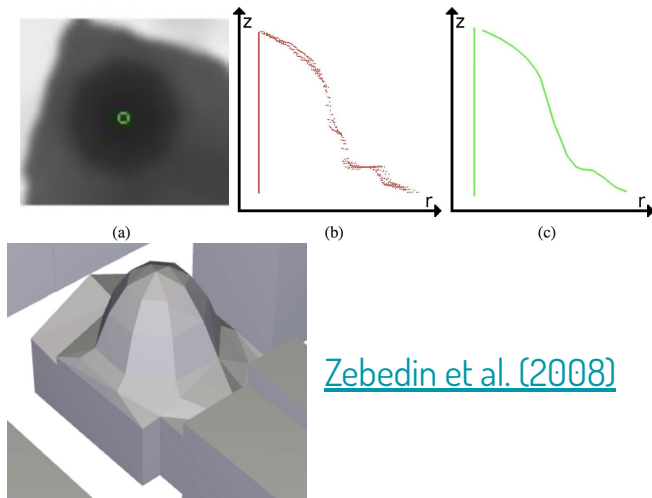
Results



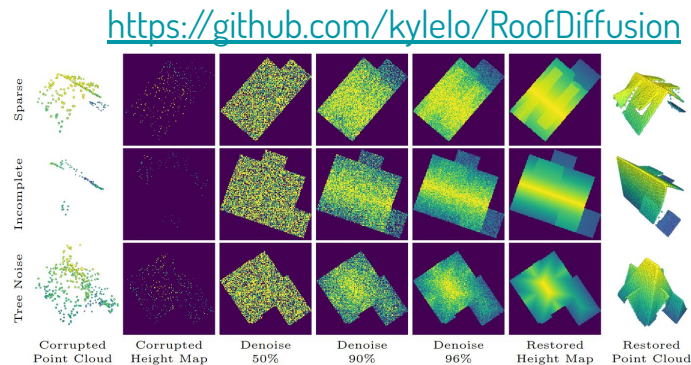
3DBAG future work

Future work

- Incorporating local pointclouds (municipalities)
 - Improve data quality and timeliness
- Detect simple primitives/surfaces of revolution
- 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?



[Zebedin et al. \(2008\)](#)



Thank you!

Ravi Peters

ravi.peters@3dgi.nl

Want to try the reconstruction algorithm yourself?

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