DIGITAL BUILDING PERMIT

Digital Transformation in Building Permits

Module 6 - Creation of 3D city models and GeoBIM integration

9th July 2025

Funded by
the European Union

Module 6 contents

- Creation of 3D city models
- Processing of 3D city models + practical session
- GeoBIM integration and conversions between Geo and BIM

Creation of 3D city models contents

- Overview of data sources and methods
- Reconstruction requirements and types
- 3D BAG method in detail

Creation of 3D city models: data sources

Most common:

- Topographic map as planar partition
- Land use / cadastral map
- Building footprints
- Road centrelines
- Urban mesh
- Object attributes (e.g. number of building stories or number of road lanes)
- Lidar point clouds (aerial or terrestrial)
- Photogrammetric mesh or point cloud
- DSM / DTM

Creation of 3D city models: methods

Most common:

- Lifting 2D semantic data to given height
- Fusion of 2D semantic data with point cloud
- Classification of urban mesh

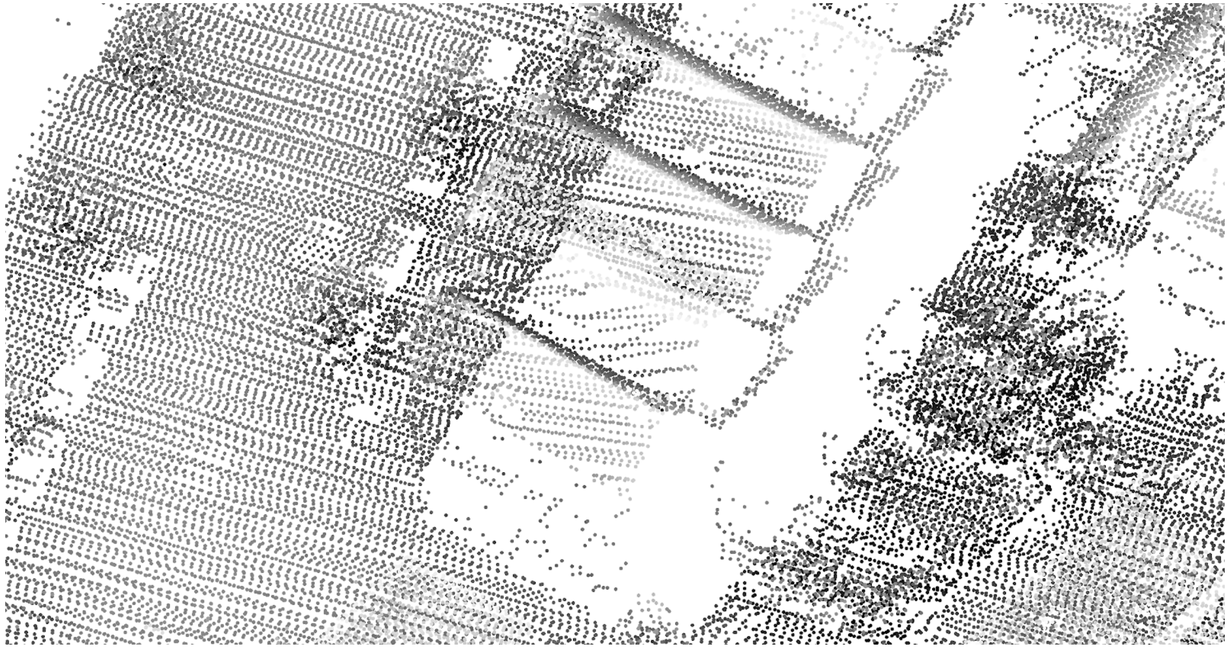
Creation of 3D city models

- **Aim**: construct semantic 3D models for different classes, e.g. buildings, roads, terrain, water bodies, etc.
- Buildings are usually the main focus
- The method and requirements can be different depending on the class and individual object, such as:
  - individual buildings and roads modelled in high detail
  - simplified terrain
  - no water bodies

Usual requirements

- **Low complexity:** the model ought to have as few vertices, edges, and faces as possible. Models with lower complexity are faster to process and take up less storage space.
- **High accuracy:** the surfaces of the model should have the lowest possible error with respect to the input.
- **Geometrically valid:** the mesh is 2-manifold, has consistent face orientation, no duplicate vertices, and no self-intersecting geometries.
- **Level of Detail (LoD):** a given degree of generalisation in the geometry of the reconstructed model compared to the actual real-world object.

Requirements: point density



Requirements: complex urban environments

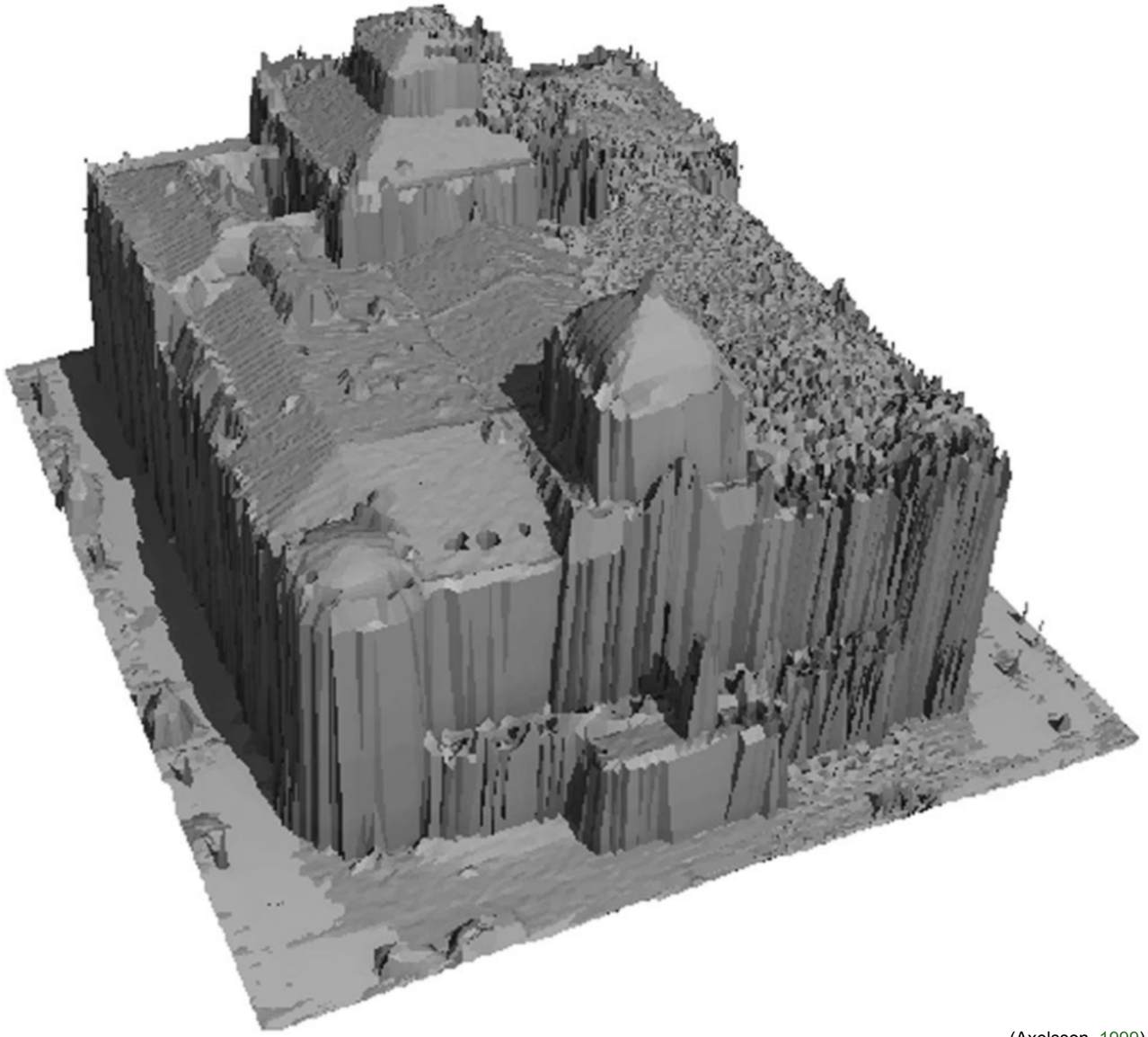
Creation of 3D city models

Two broad approaches (ends of a spectrum):

- **Data-driven**: Create a detailed model that perfectly matches the input data. Problems in the input data will cause problems in the model, e.g. holes from occluded areas and buildings that include nearby trees.

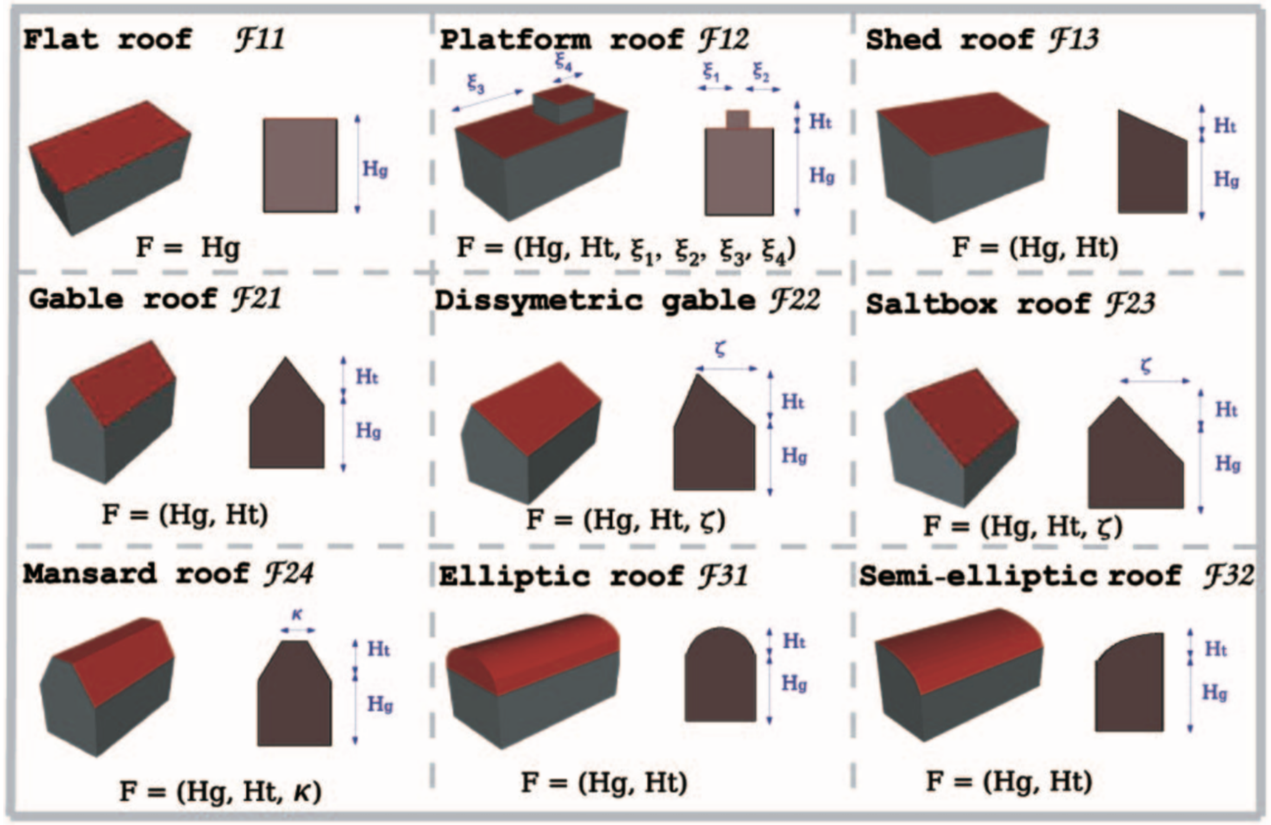
- **Model-driven**: Define rules to create models based on predefined types. Models won't fit the data as closely and will be less detailed, but problems in the input data (e.g. low point density or occlusion) can be more easily managed.

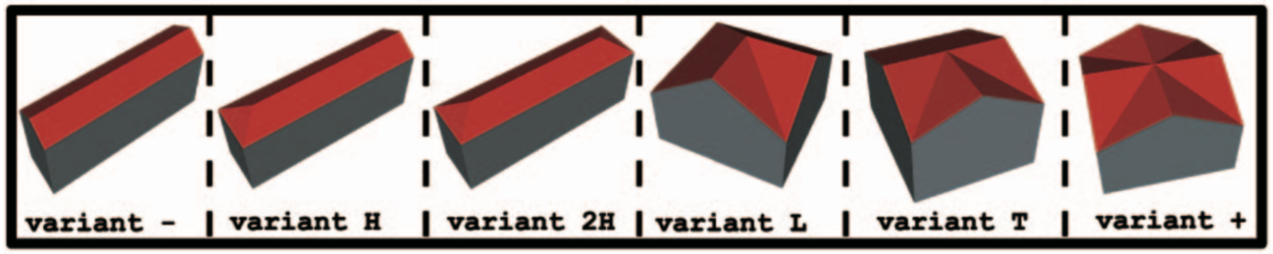
Creation of 3D city models: data-driven approach



(Axelsson, 1999)

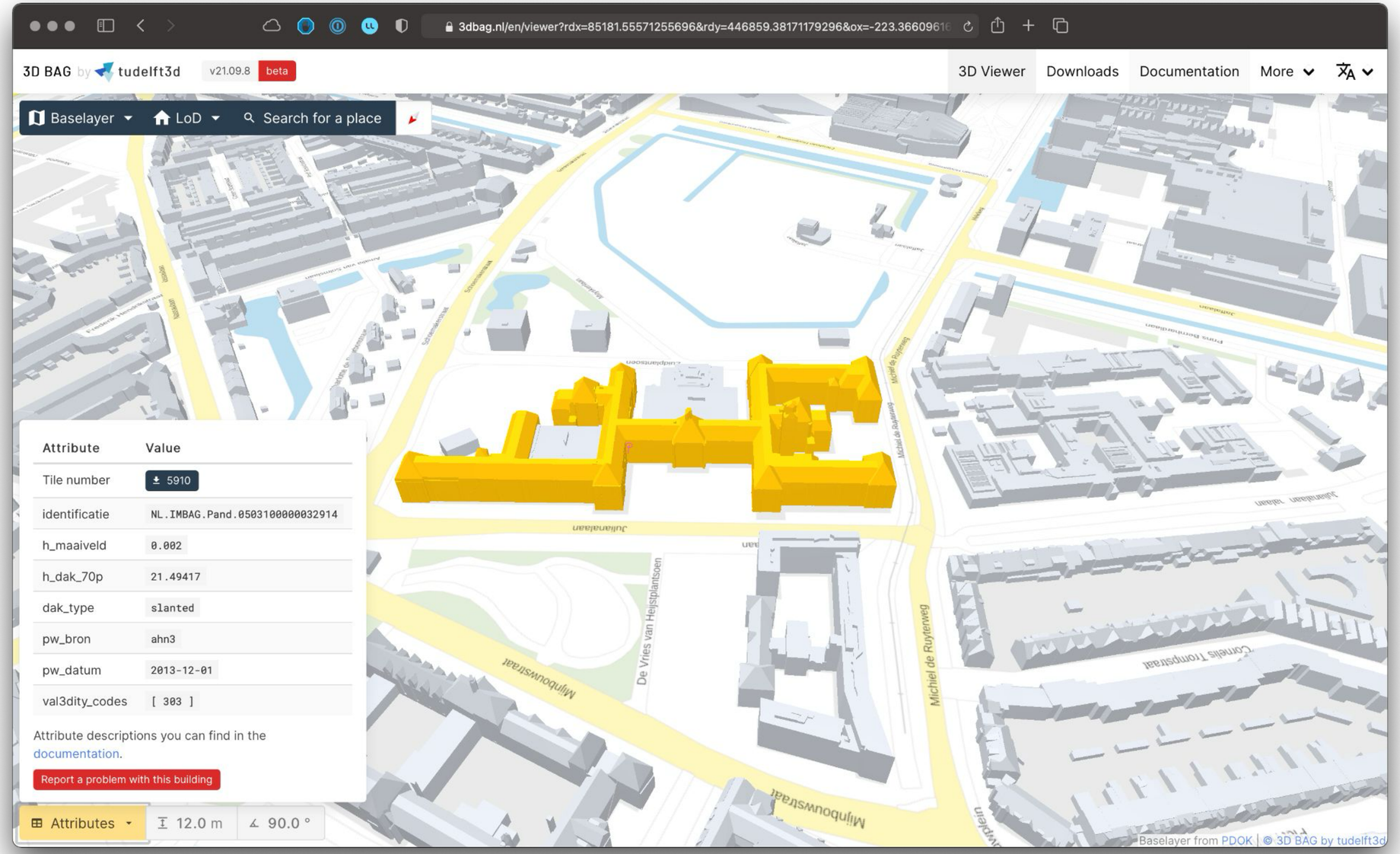
Creation of 3D city models: model-driven approach





(Lafarge et al., 2010)

Example: 3D BAG



Example: 3D BAG

- 3D city model covering all 10 million buildings in the Netherlands
- Multiple formats: CityJSON, GeoPackage, OBJ, etc.
- Open data created from other open data:
  - BAG building footprints
  - AHN point cloud
- Mixed approach: partly data-driven and partly model-driven

Example: 3D BAG

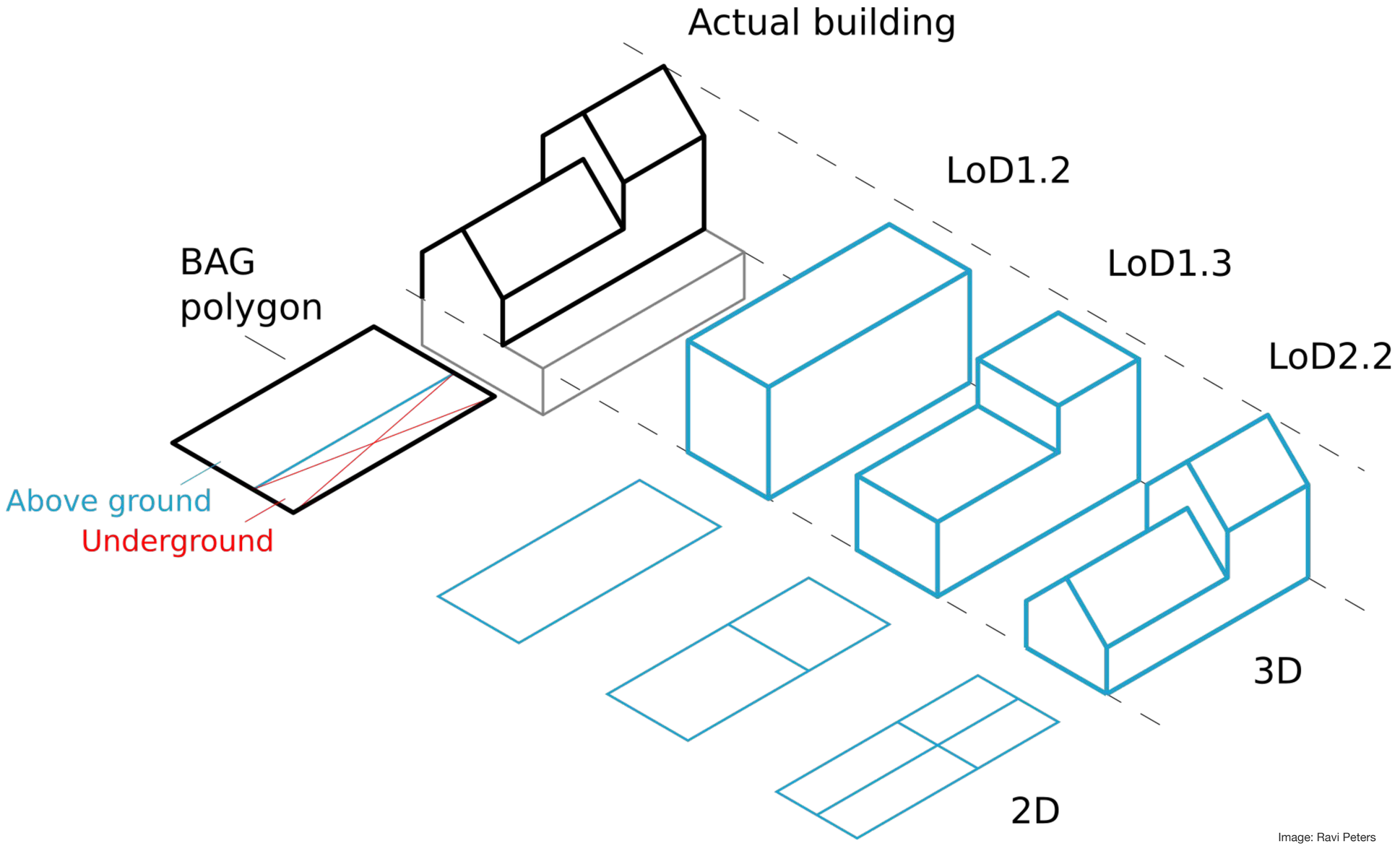
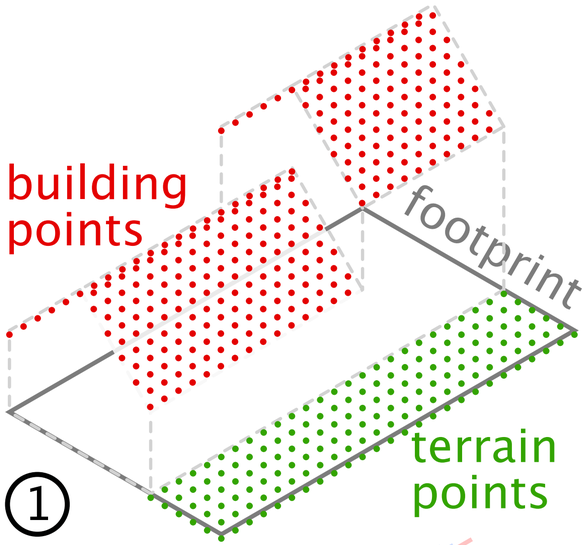
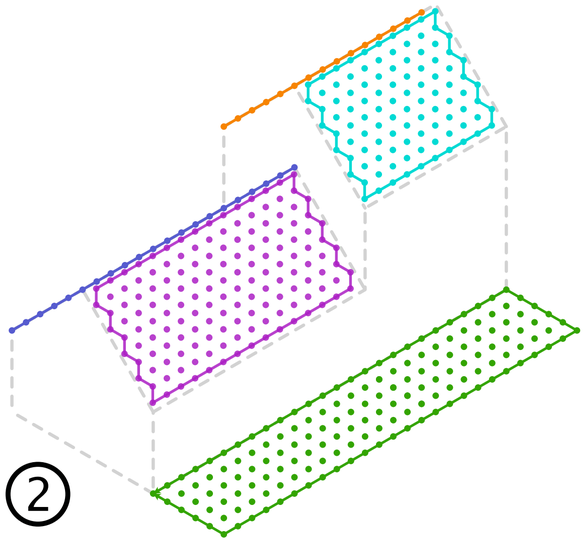


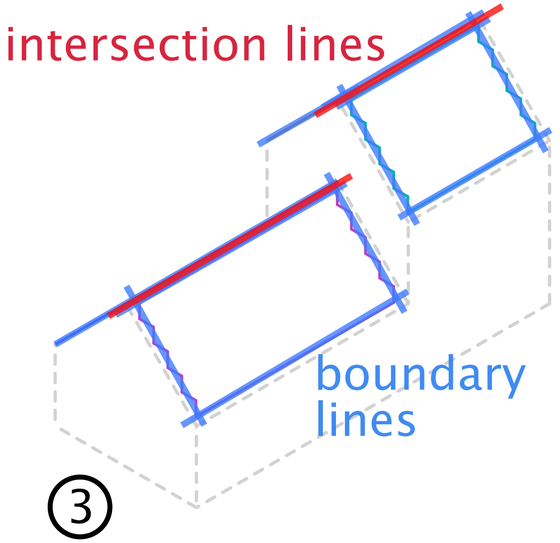
Image: Ravi Peters

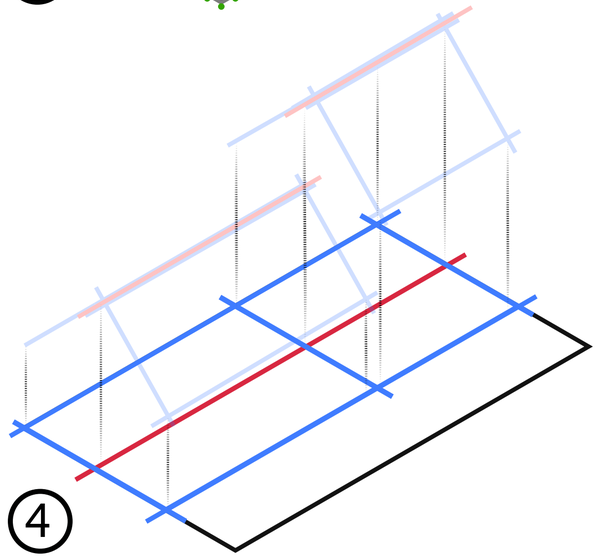
3D BAG: modelling assumptions

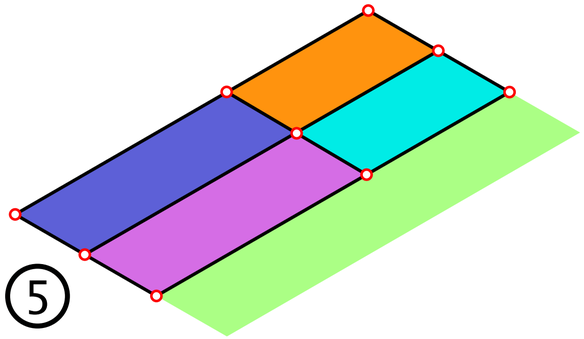
- **Piecewise planar:** The shape of a building can be adequately approximated using planar faces that are detectable from the point cloud.
- **2.5D with vertical walls:** The roof of the building is 2.5D and all walls are vertical. This implies the 3D building model can be extruded from a 2D planar partition of the roof.
- **Classified point cloud:** A reliable classification of the input point cloud is expected with least a building and a terrain (ground) class.
- **Footprints are available:** Apart from a point cloud, the method also takes 2D building footprints as input. It is assumed that the footprints are up-to-date and well aligned with the point cloud.

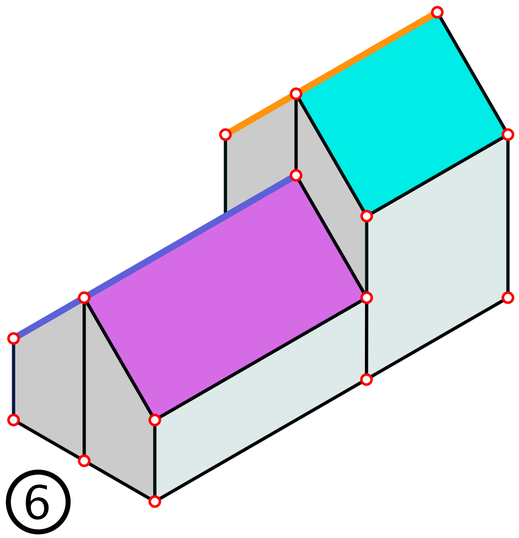

building points
footprint
terrain points
①


②


intersection lines
boundary lines
③


④


⑤


⑥

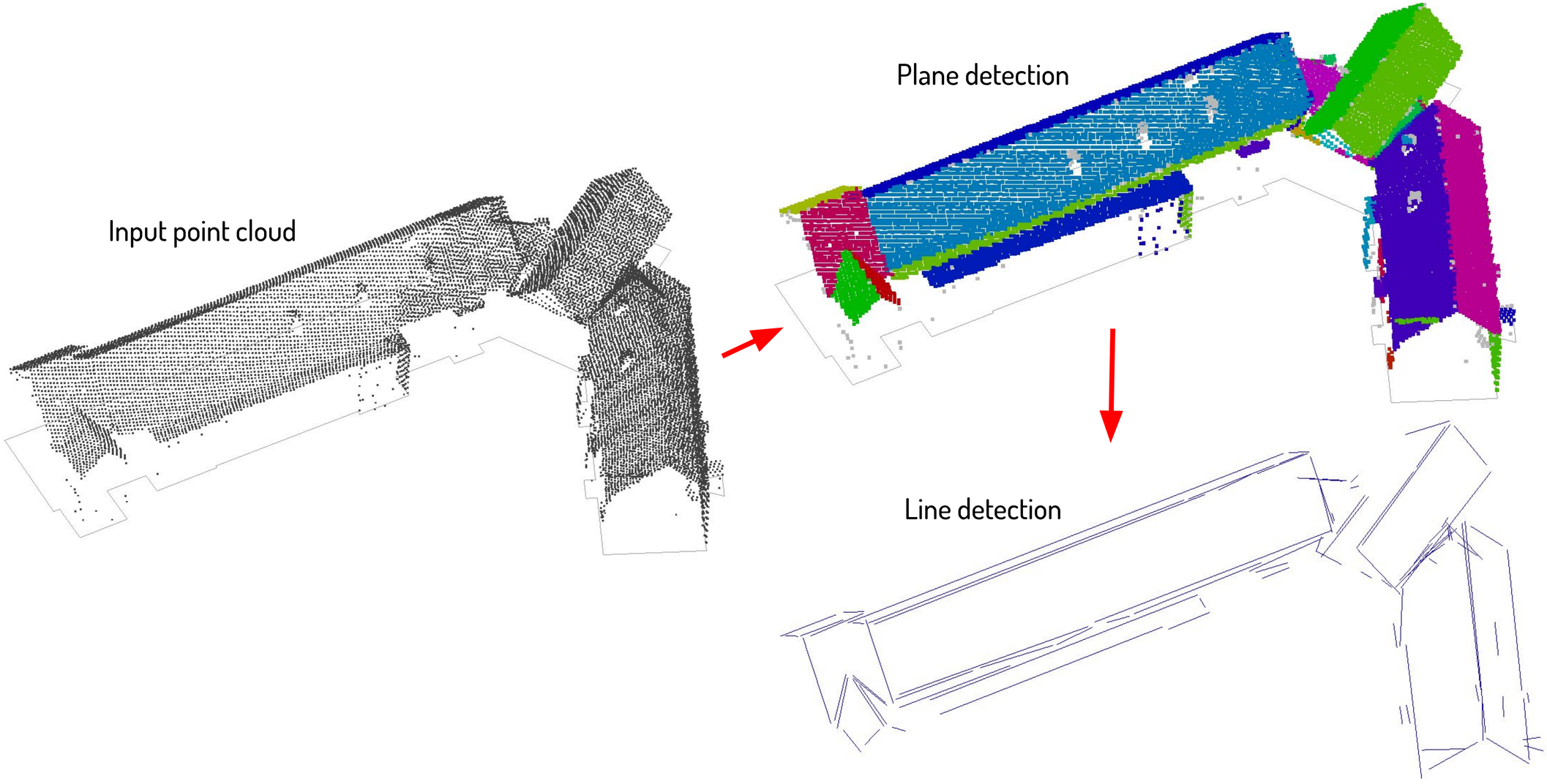
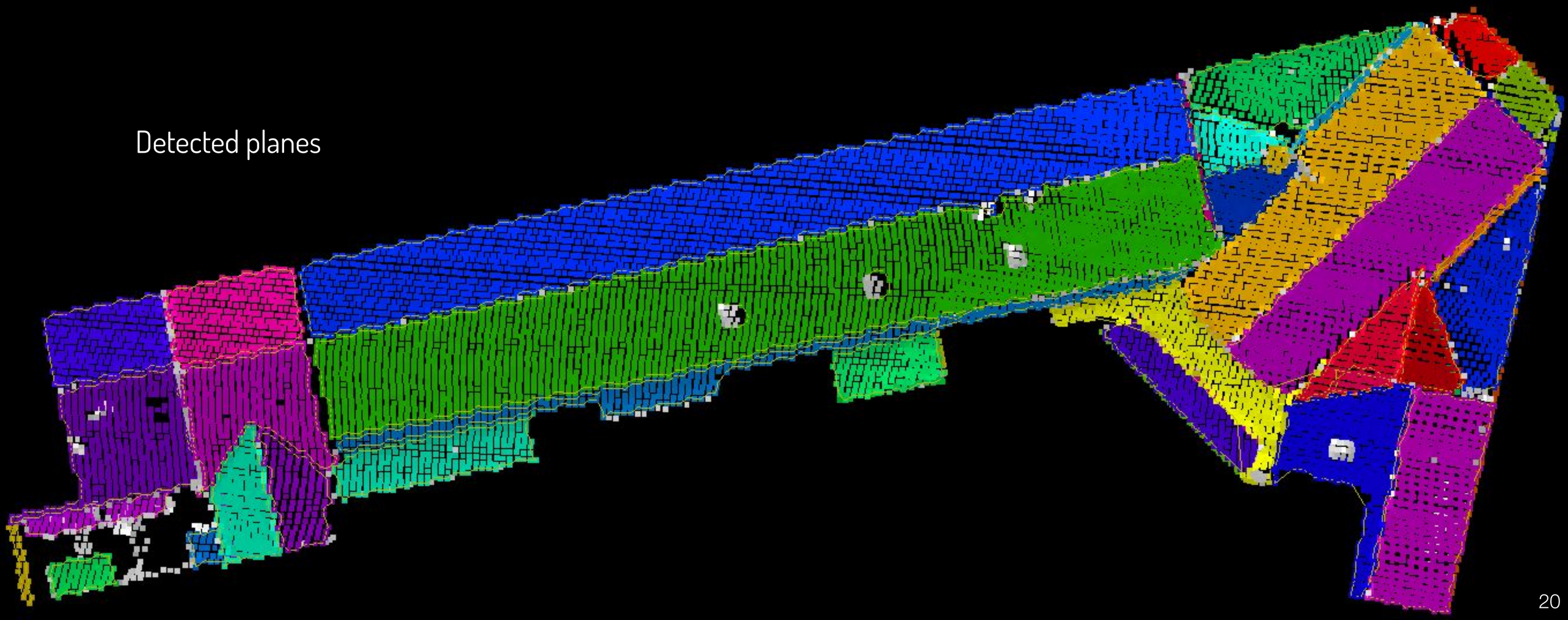


Image: Ravi Peters

3D BAG

Image: Ravi Peters

Detected planes



3D BAG

Image: Ravi Peters

α-shape

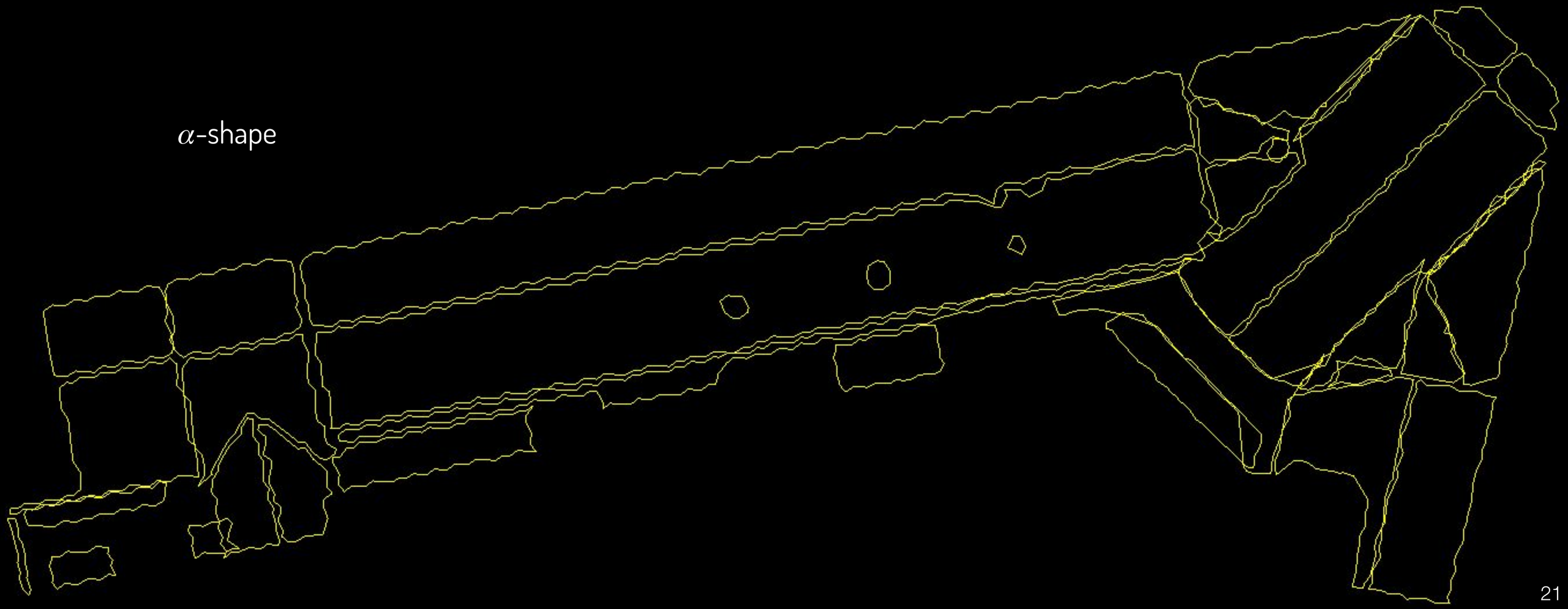
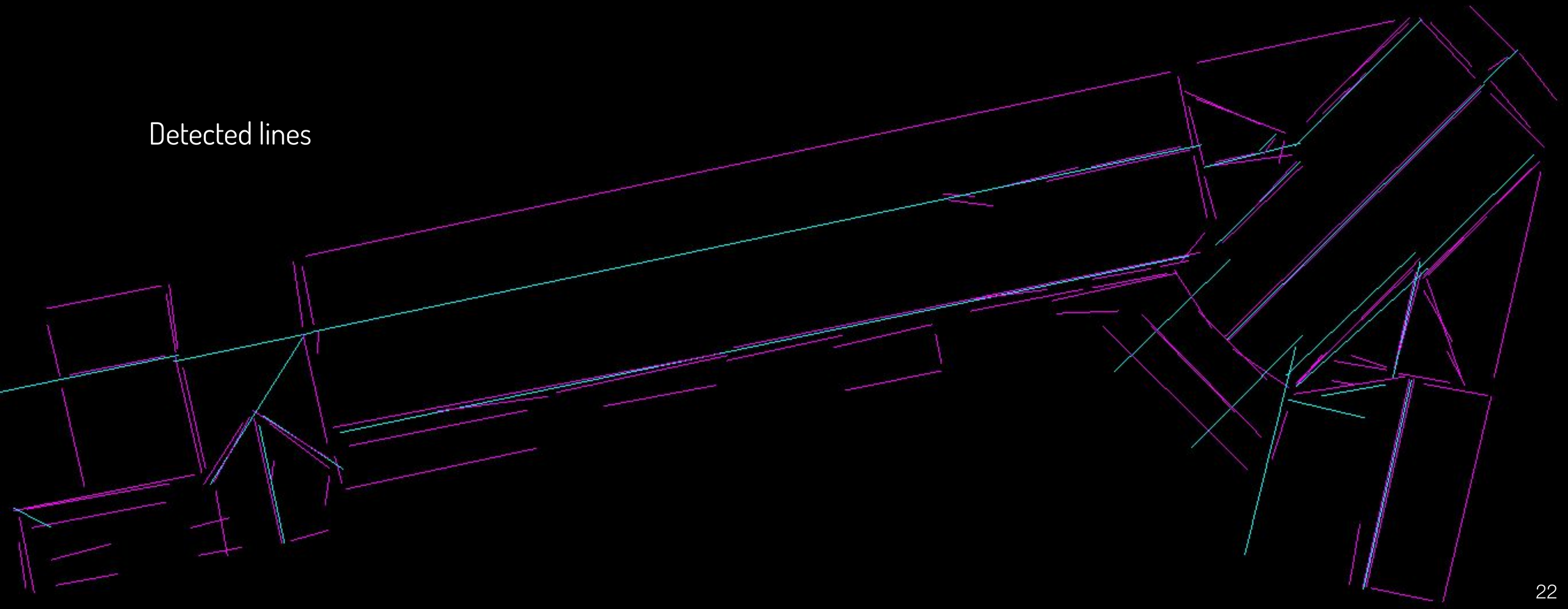
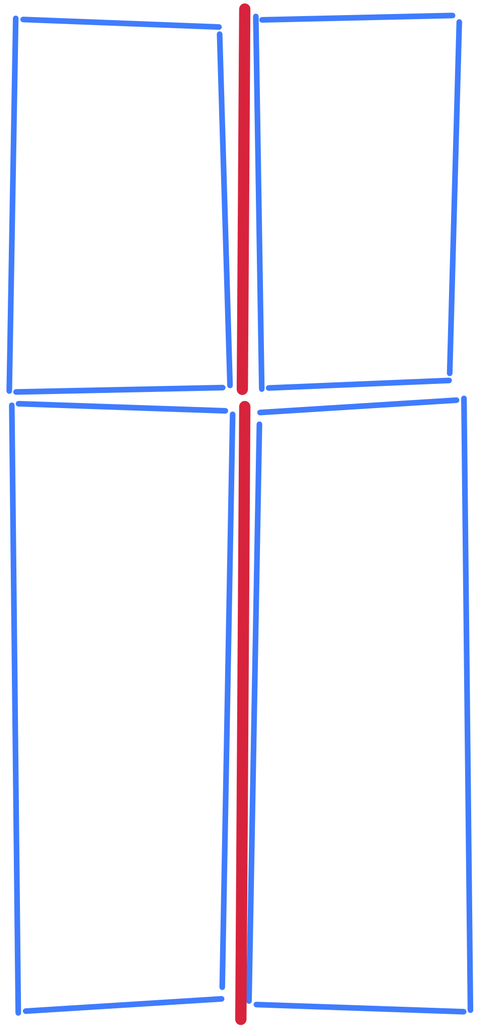


Image: Ravi Peters

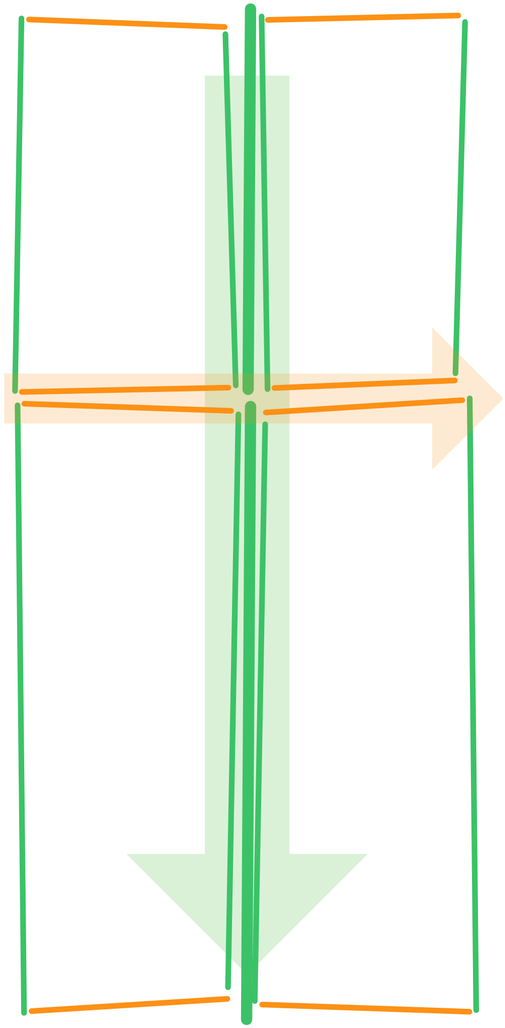
Detected lines



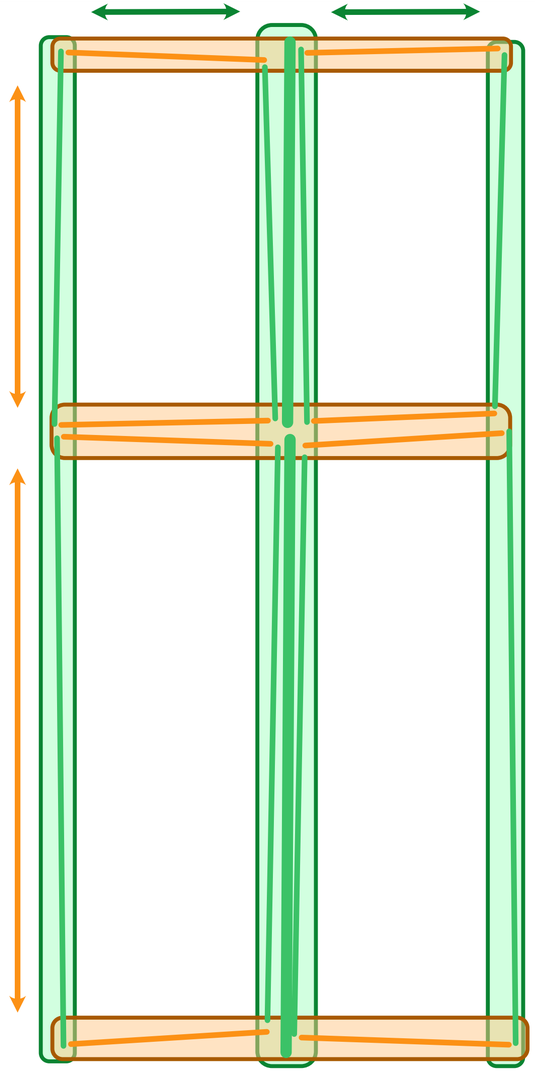
3D BAG: line regularisation



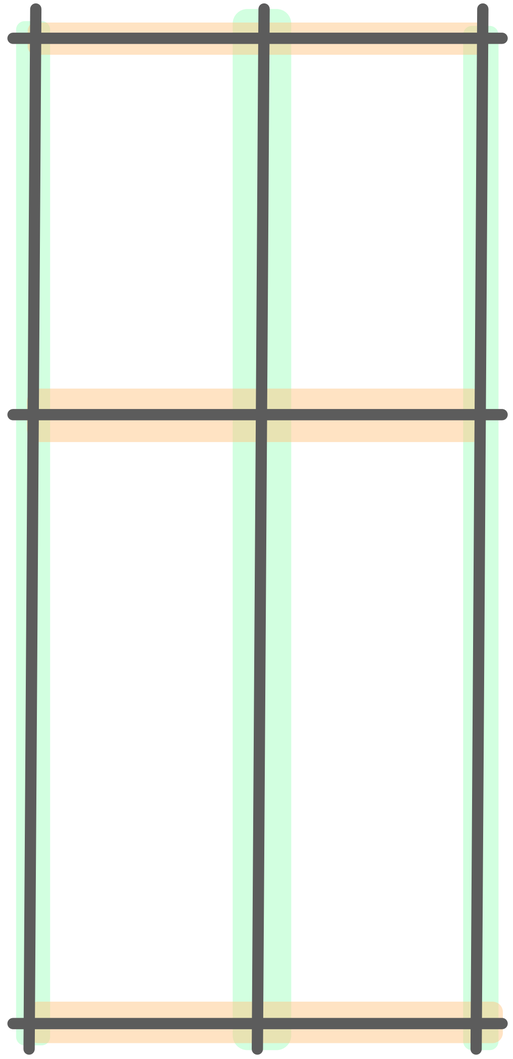
detected lines



orientation clustering



distance clustering



regularised lines

3D BAG: initial roof partition

- The regularised lines mostly form a planar partition, but there are some artefacts to fix:
  - very small faces, and
  - dangling edges.

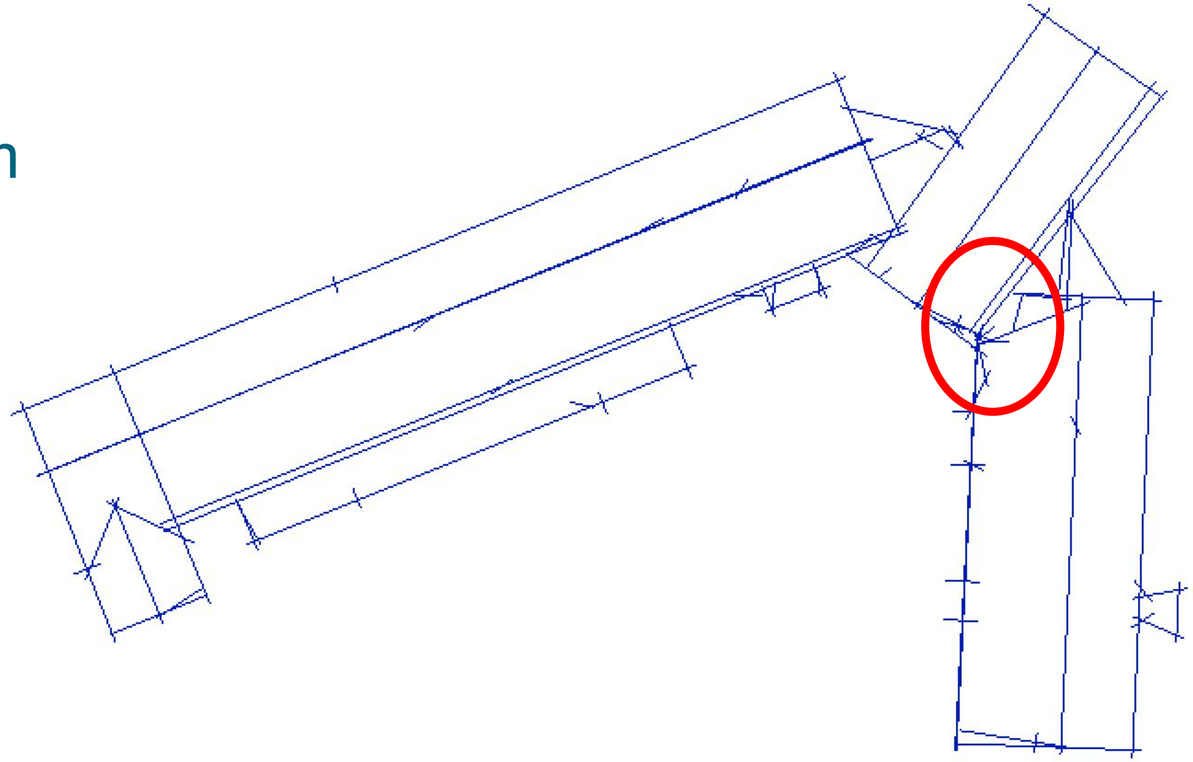
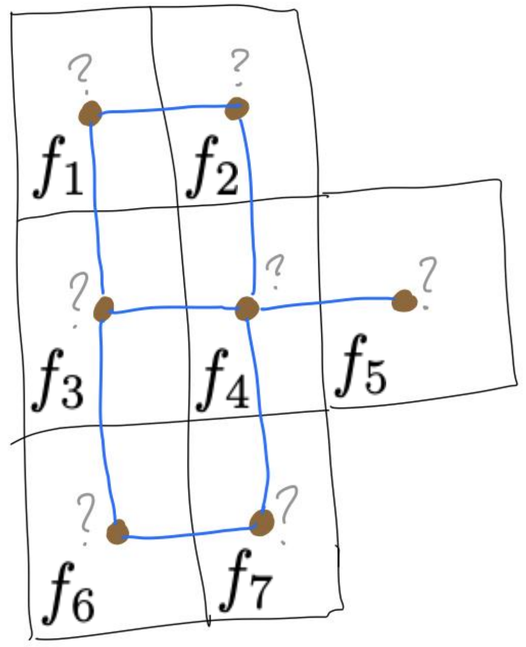


Image: Ravi Peters

3D BAG: roof partition optimisation

- In order to obtain a better roof partition, the dangling edges are removed and certain faces are merged.
- A dual graph of the original input faces is created, where faces are vertices and adjacent faces are joined by an edge.
- The edges in this graph represent potential faces that could be merged.

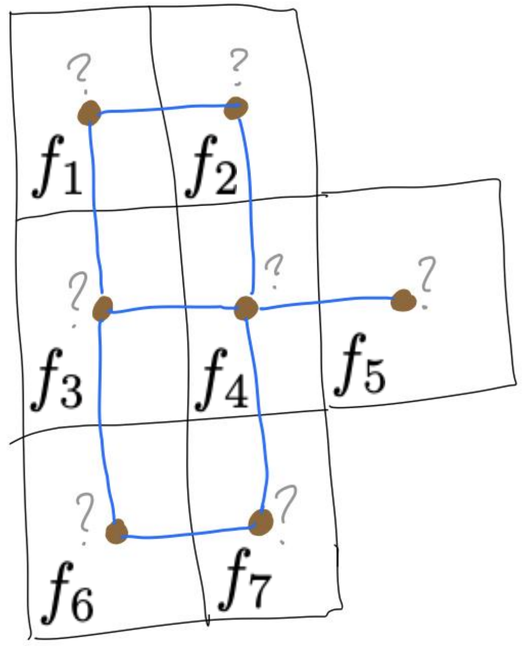
Dual graph of planar arrangement

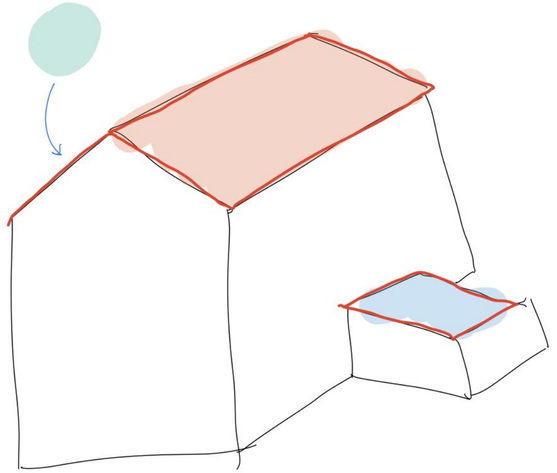


3D BAG: roof partition optimisation

- **Aim**: assign a plane label to each face, then merge adjacent faces that have the same label.
- This label is assigned by minimising a weighted sum of two opposing terms: a data term (data-driven) and a smoothness term (model-driven)

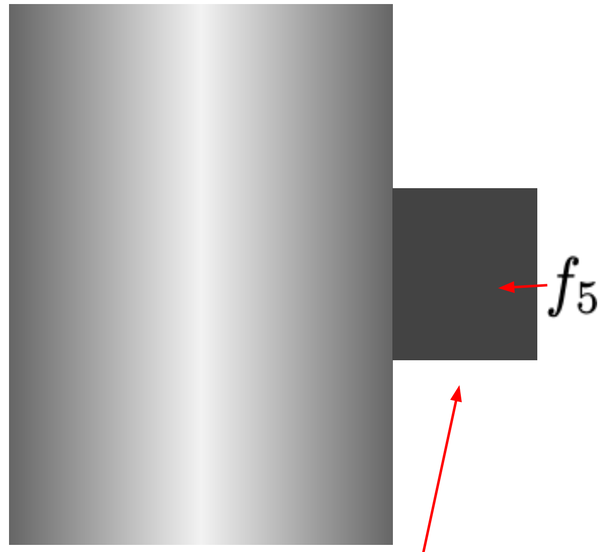
Dual graph of planar arrangement



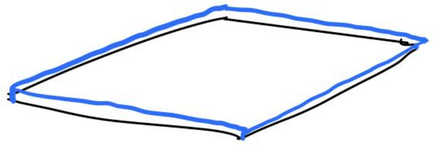


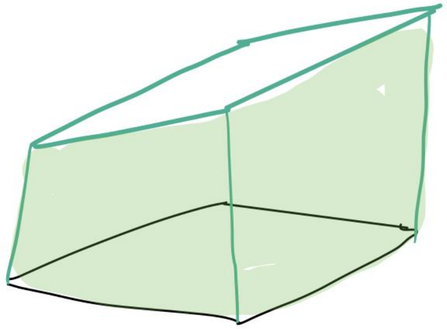
3D BAG: roof partition optimisation

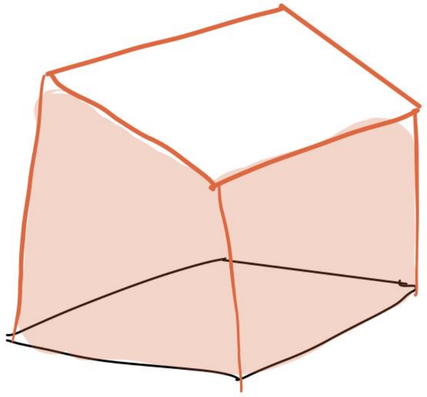
- The data term tries to pick the label that best fits the data.
- It is given by the volume between the 2.5D heightfield of the point cloud and the candidate labels' planes.

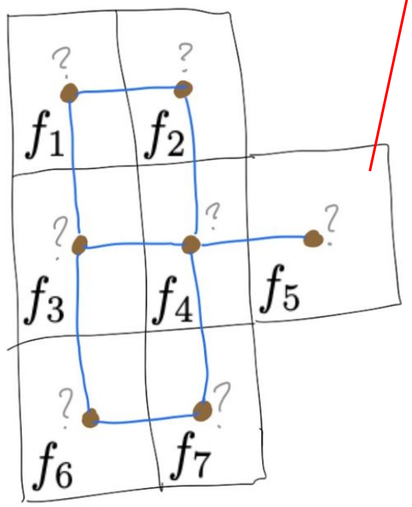


heightfield



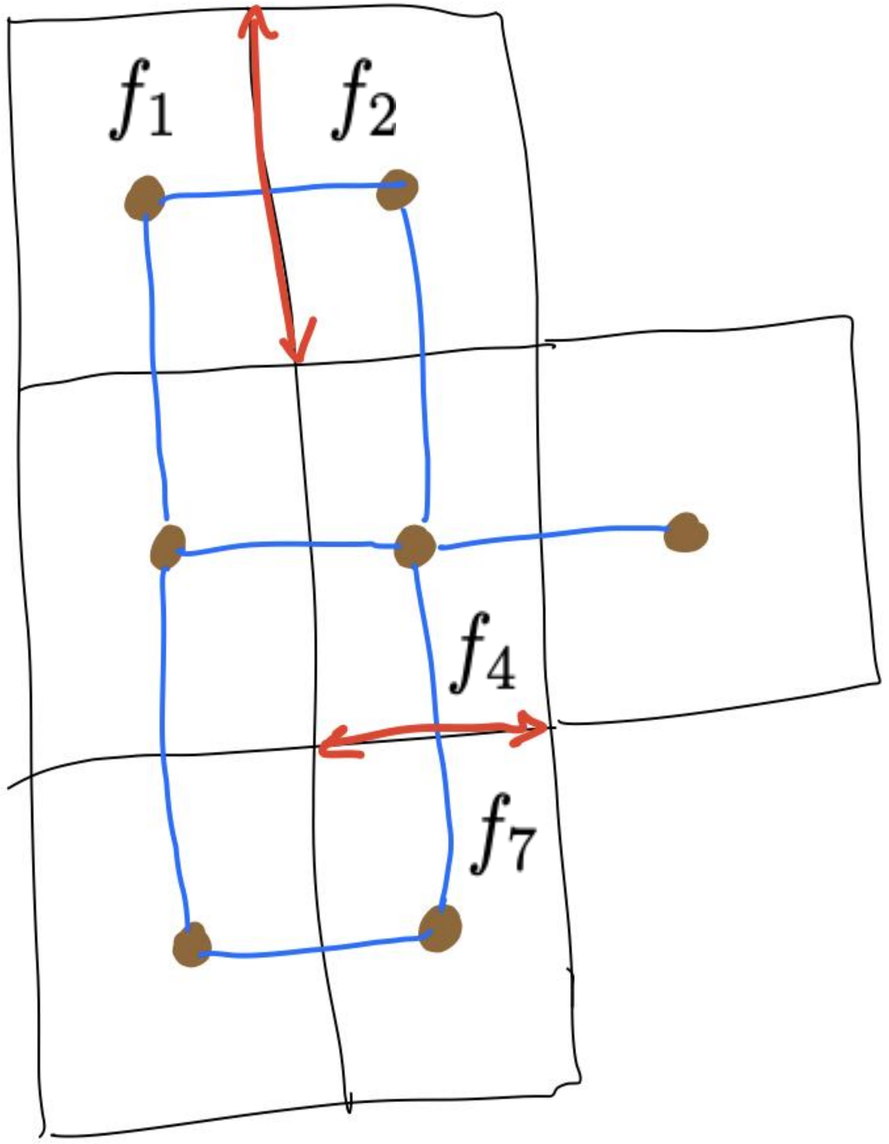




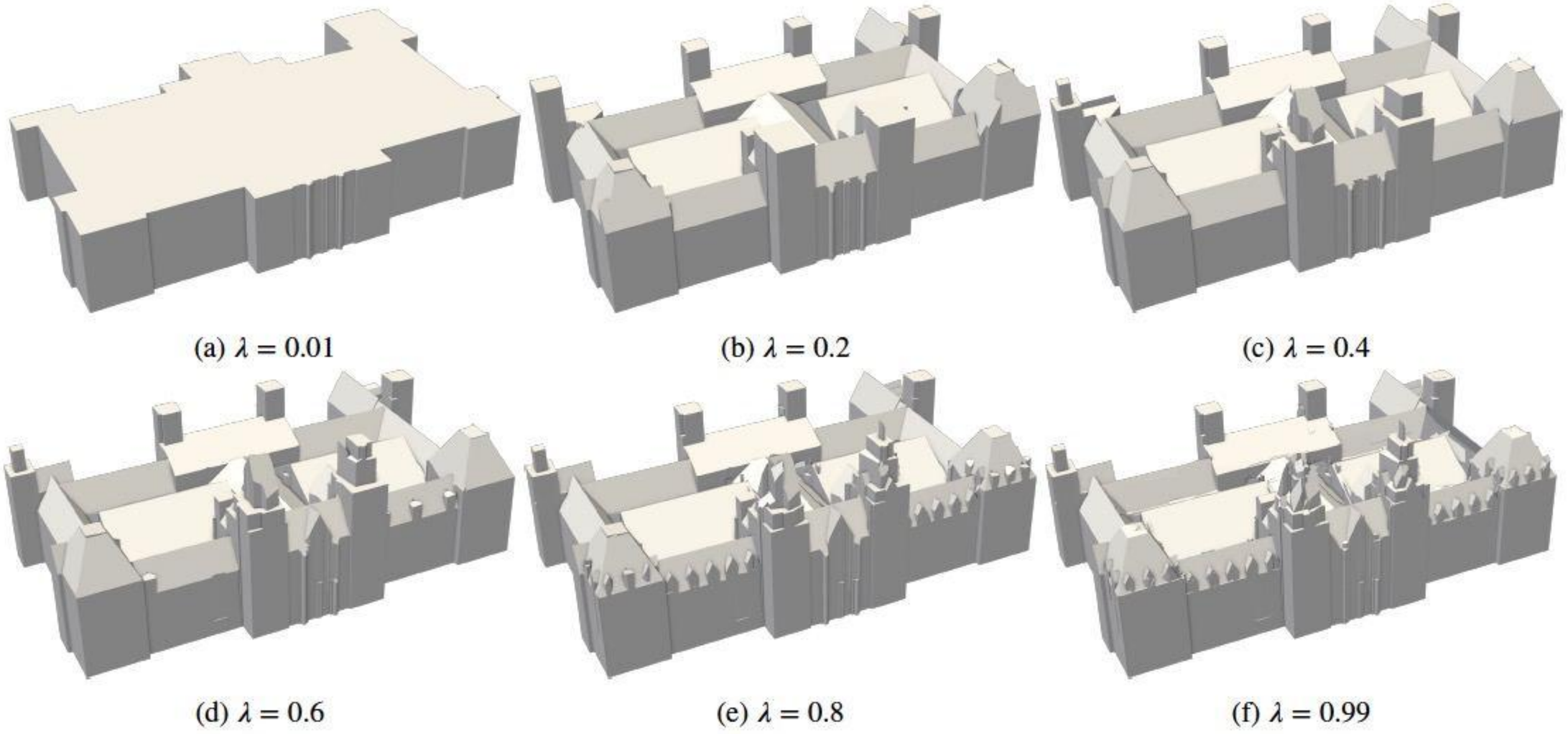


3D BAG: roof partition optimisation

- The smoothness term tries to merge adjacent faces. For a pair of adjacent faces, it is:
  - zero if the faces have the same label, and
  - the length of the common face if they do not.



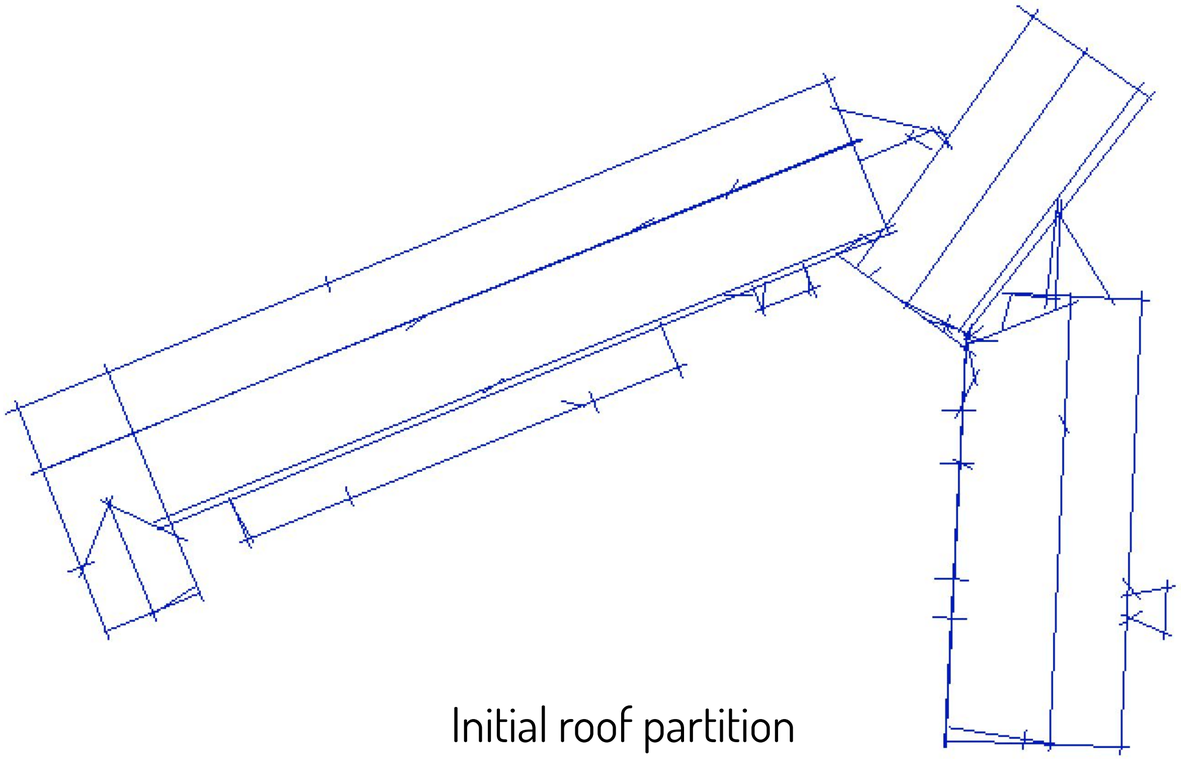
3D BAG: roof partition optimisation (weight)



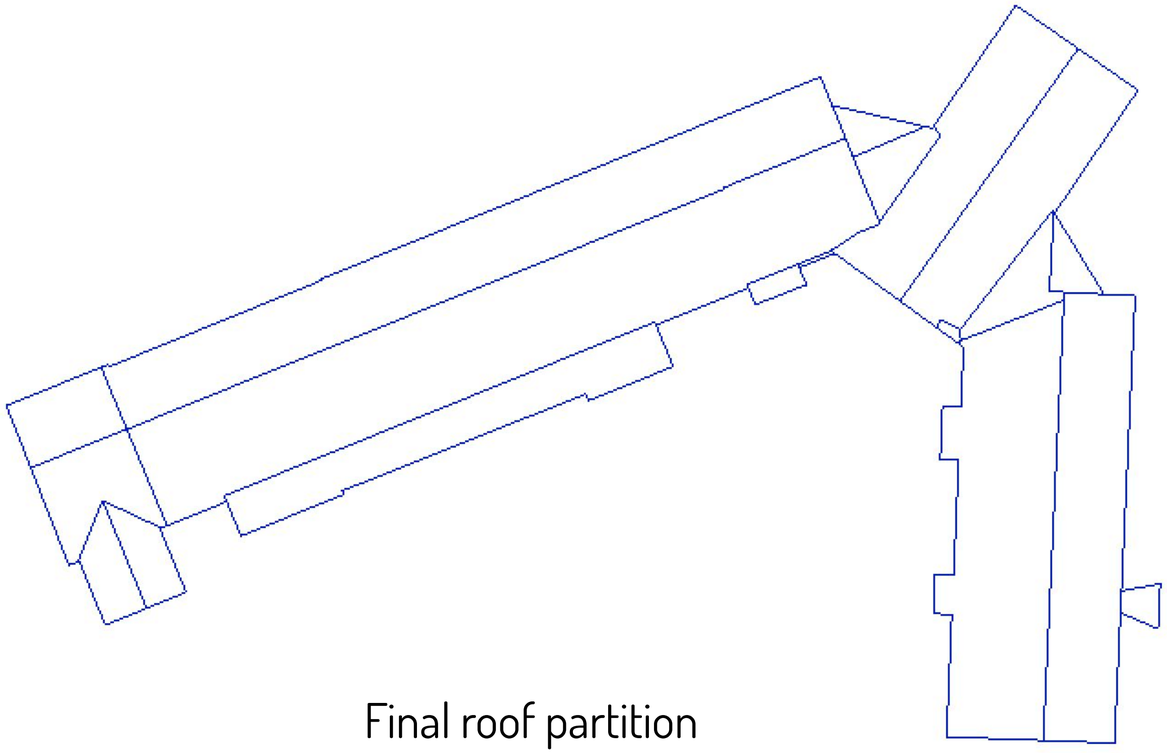
(a) $\lambda = 0.01$ (b) $\lambda = 0.2$ (c) $\lambda = 0.4$

(d) $\lambda = 0.6$ (e) $\lambda = 0.8$ (f) $\lambda = 0.99$

3D BAG: roof partition optimisation



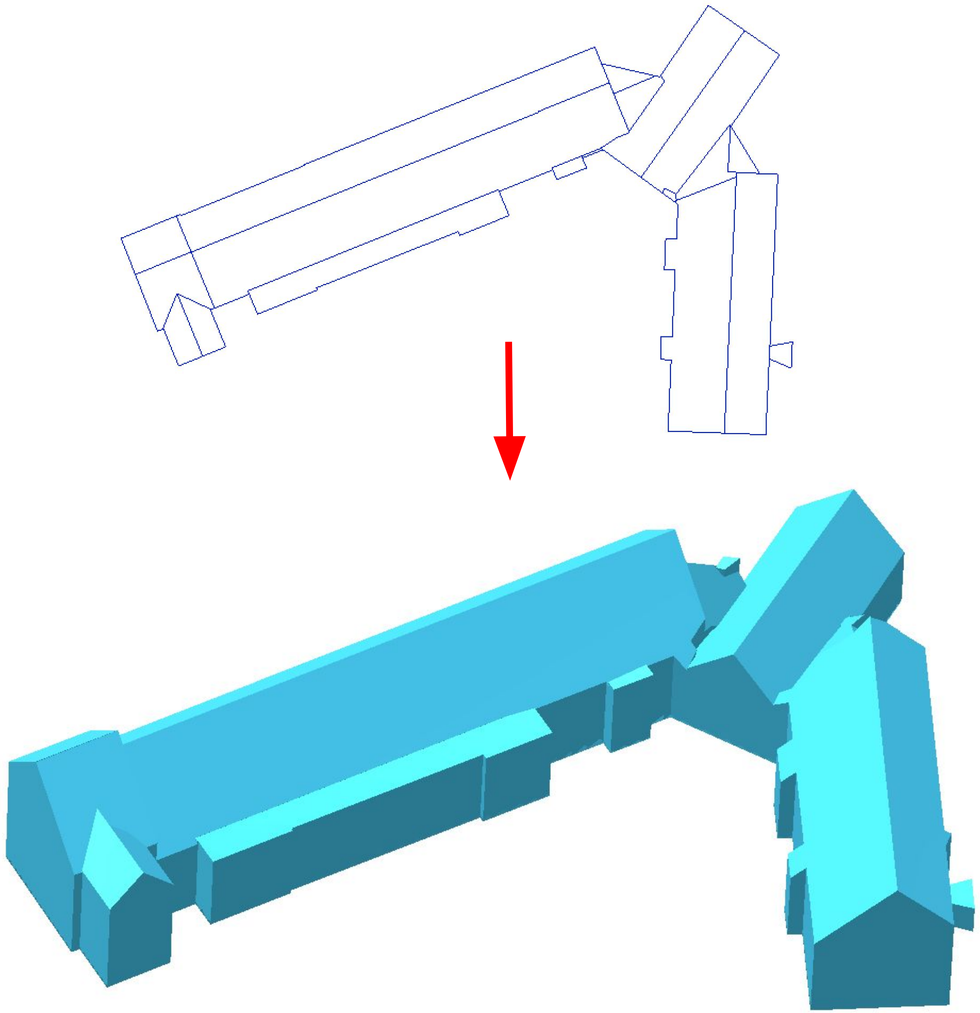
Initial roof partition



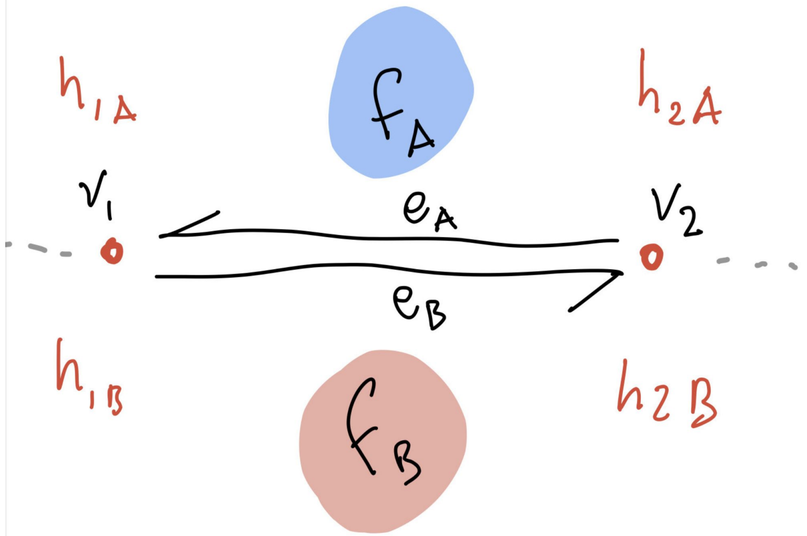
Final roof partition
(edges between equal plane labels dissolved)

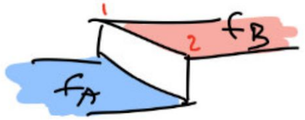
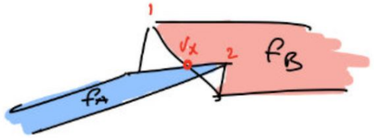
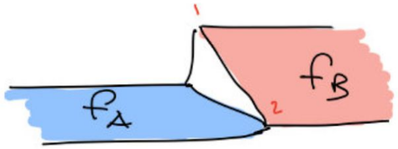
Image: Ravi Peters

3D BAG: extrusion



3D BAG: extrusion

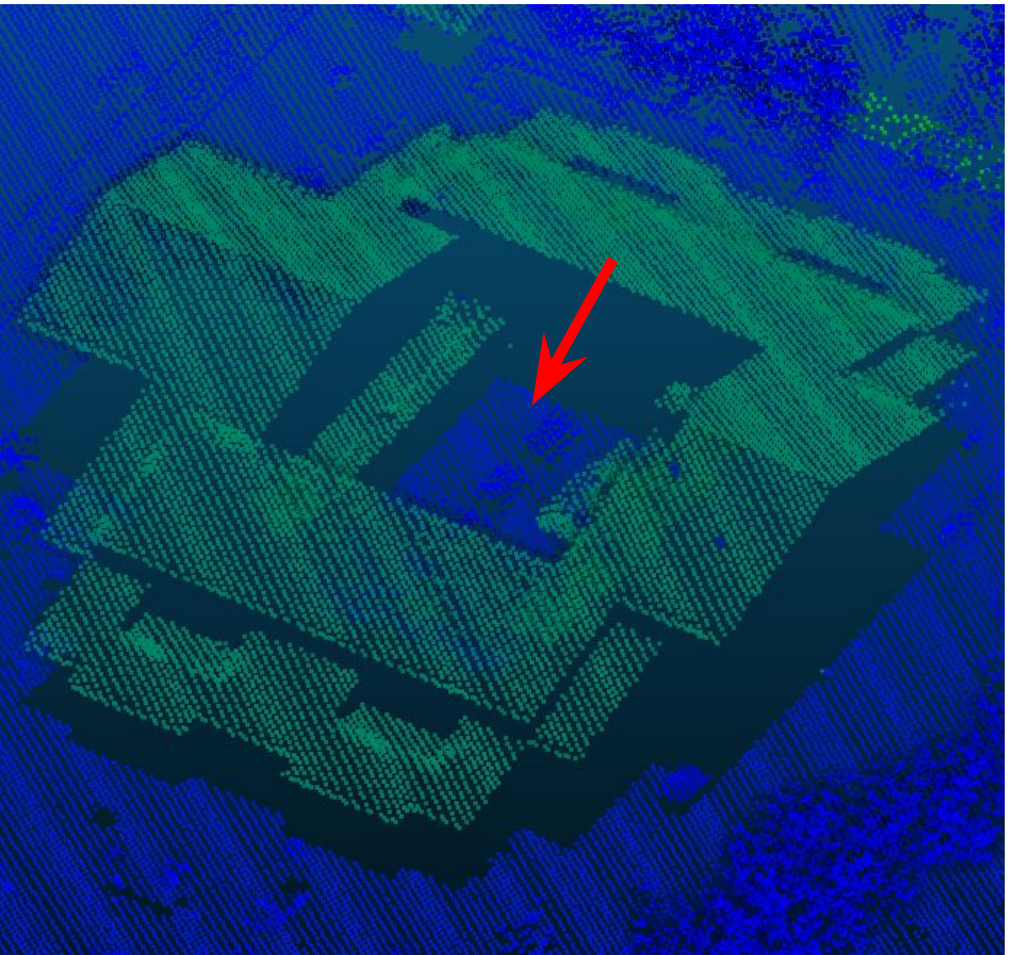


Case	condition	vertex order
	$h_{1A} < h_{1B}$ AND $h_{2A} < h_{2B}$	1. v_{1B}, v_{1A}, v_{2A}, v_{2B}
	$h_{1A} < h_{1B}$ AND $h_{2A} > h_{2B}$	1. v_{1B}, v_{1A}, v_X 2. v_{2A}, v_{2B}, v_X } 2 faces!
	$h_{1A} < h_{1B}$ AND $h_{2A} == h_{2B}$	1. v_{1B}, v_{1A}, v_{2A}

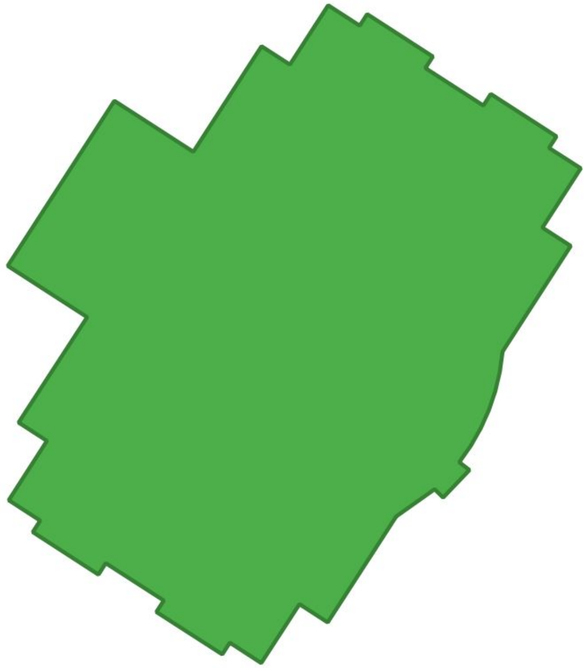
3D BAG: challenges

In some cases BAG footprint includes groundparts

AHN3 ground and building class



BAG footprint



Reconstruction result:
roofplane fitted to groundpart

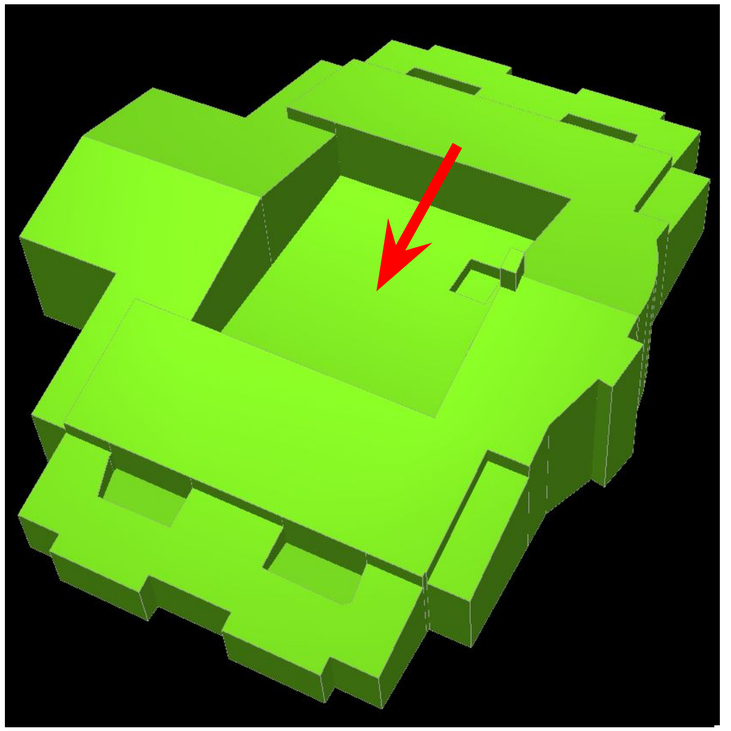
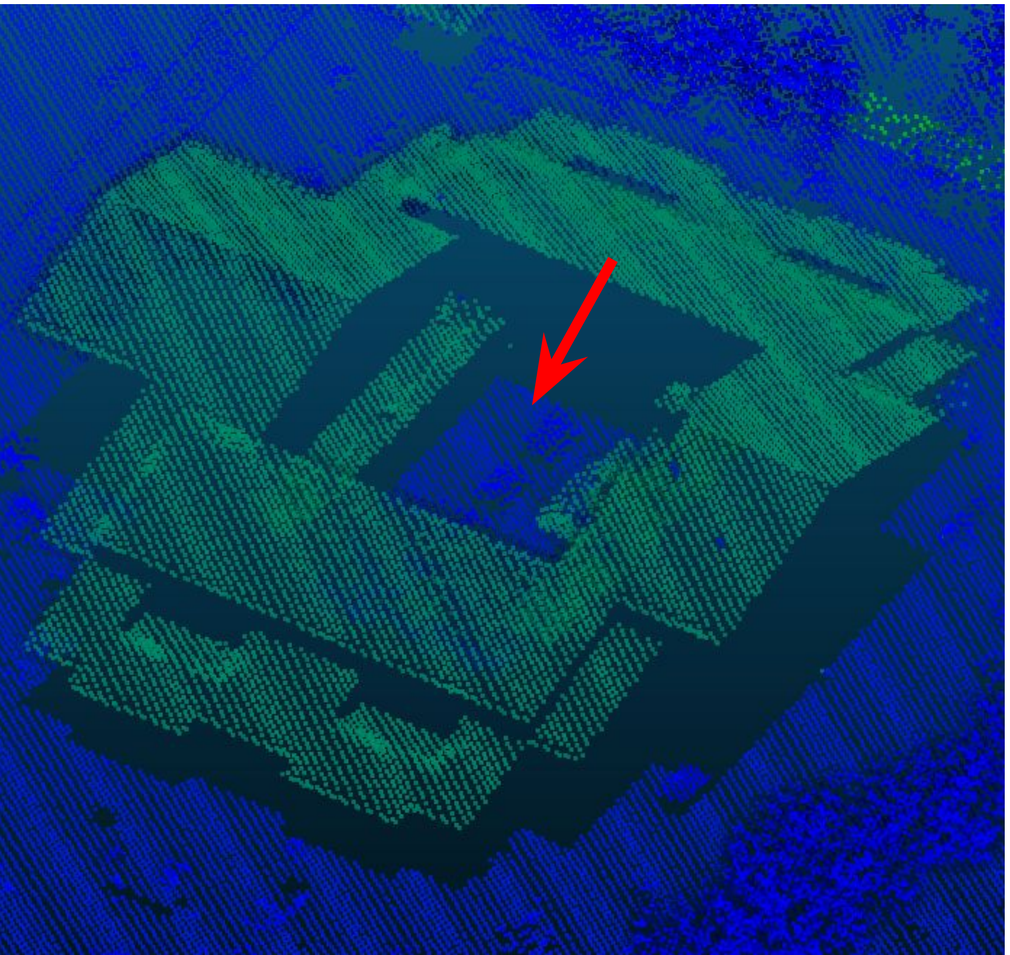


Image: Ravi Peters

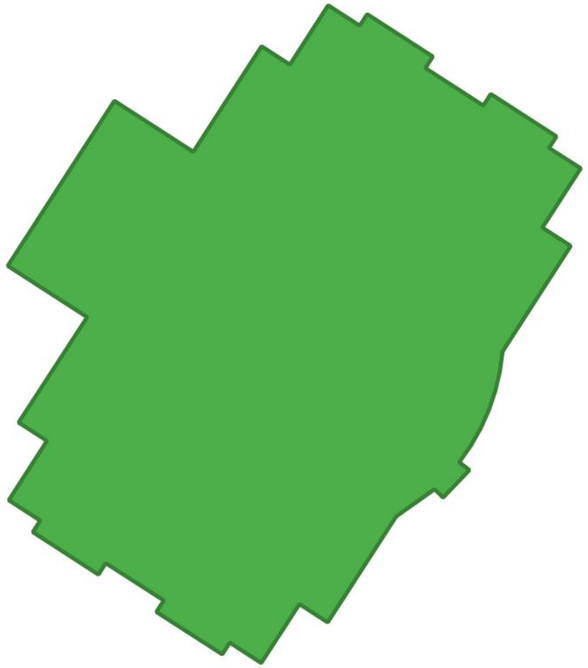
3D BAG: challenges

Reconstruction with groundpart detection

AHN3 ground and building class



BAG footprint



Reconstruction result:
groundpart removed from output

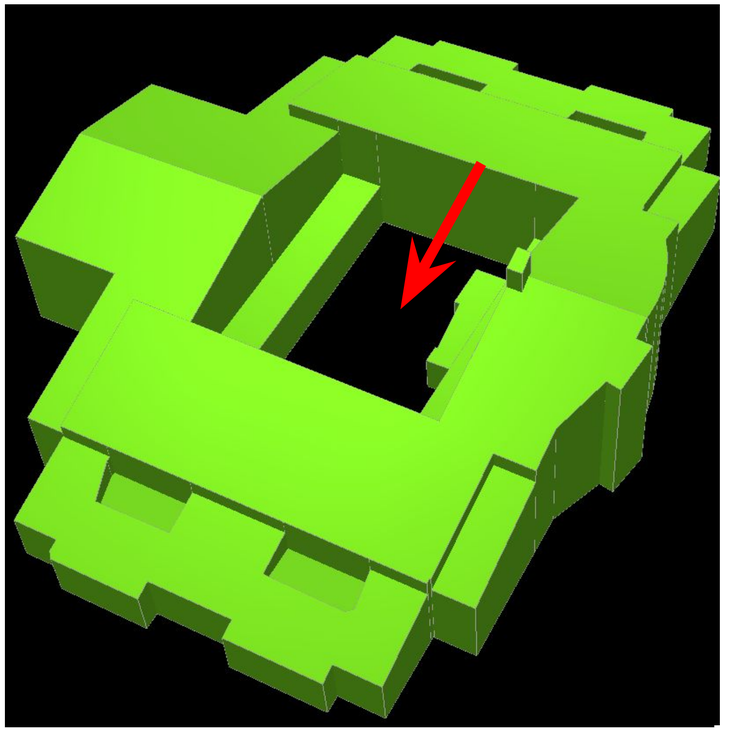
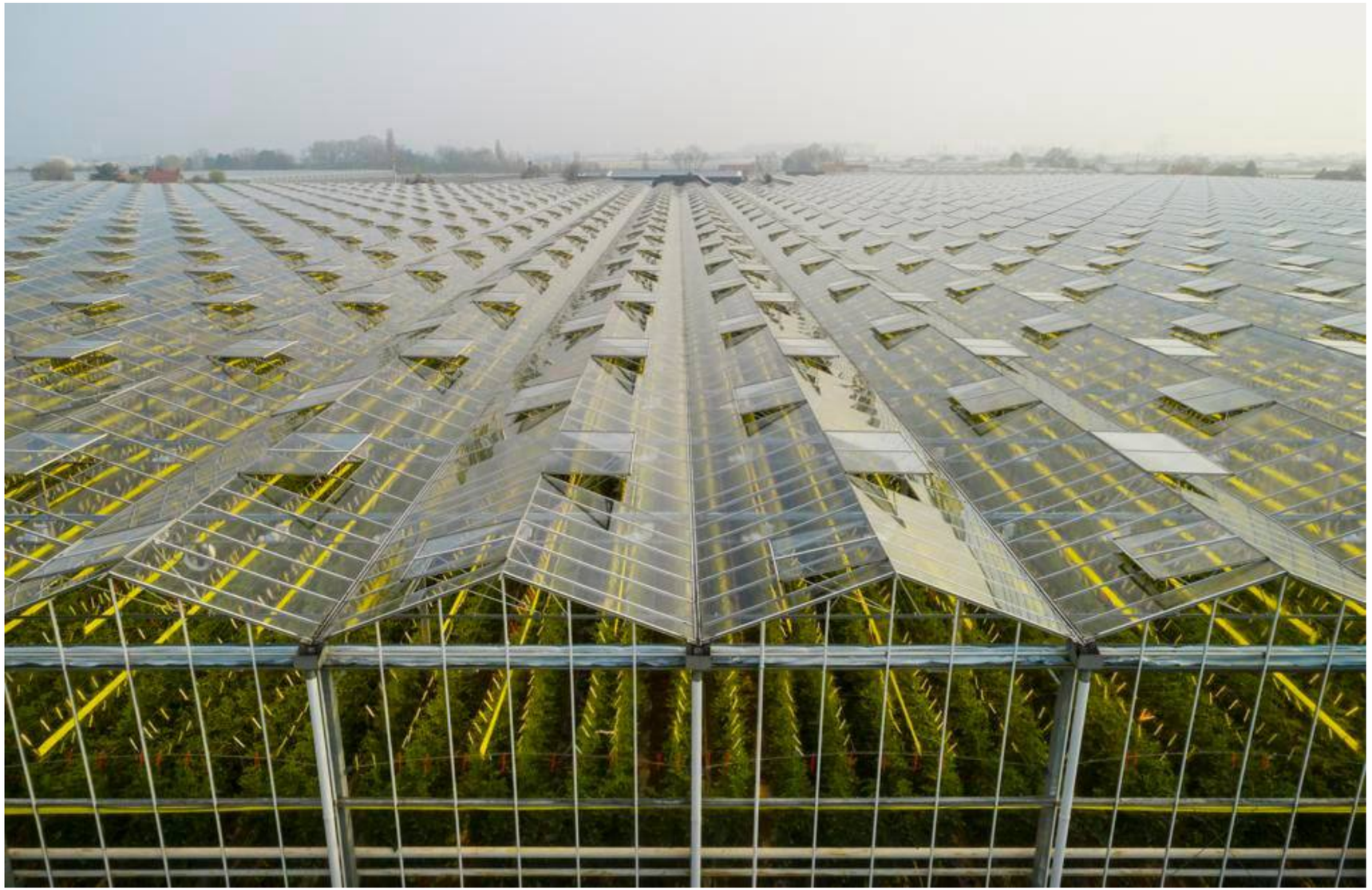


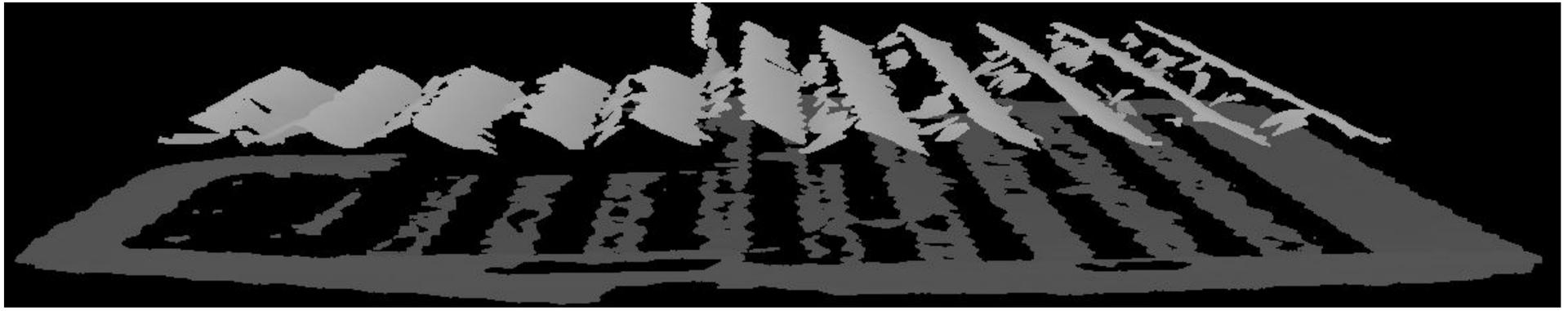
Image: Ravi Peters

3D BAG: challenges

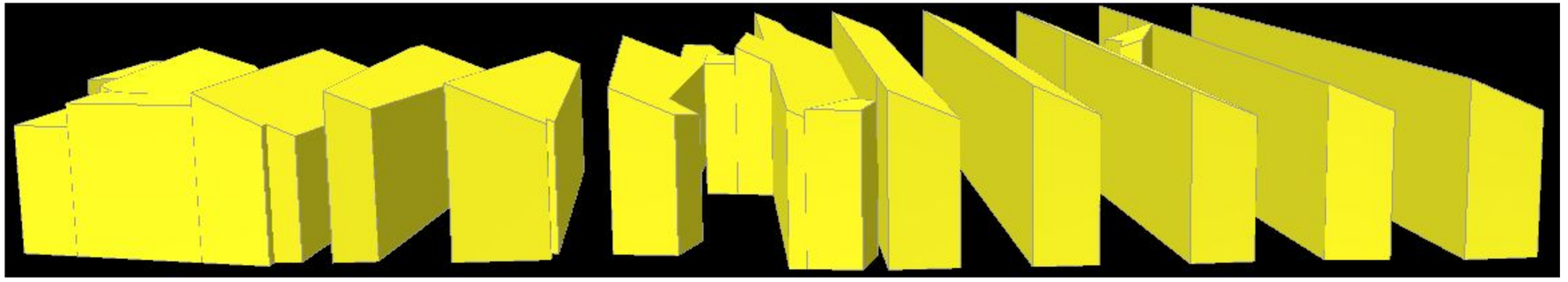


3D BAG: challenges

AHN3
ground and building class



Heightfield

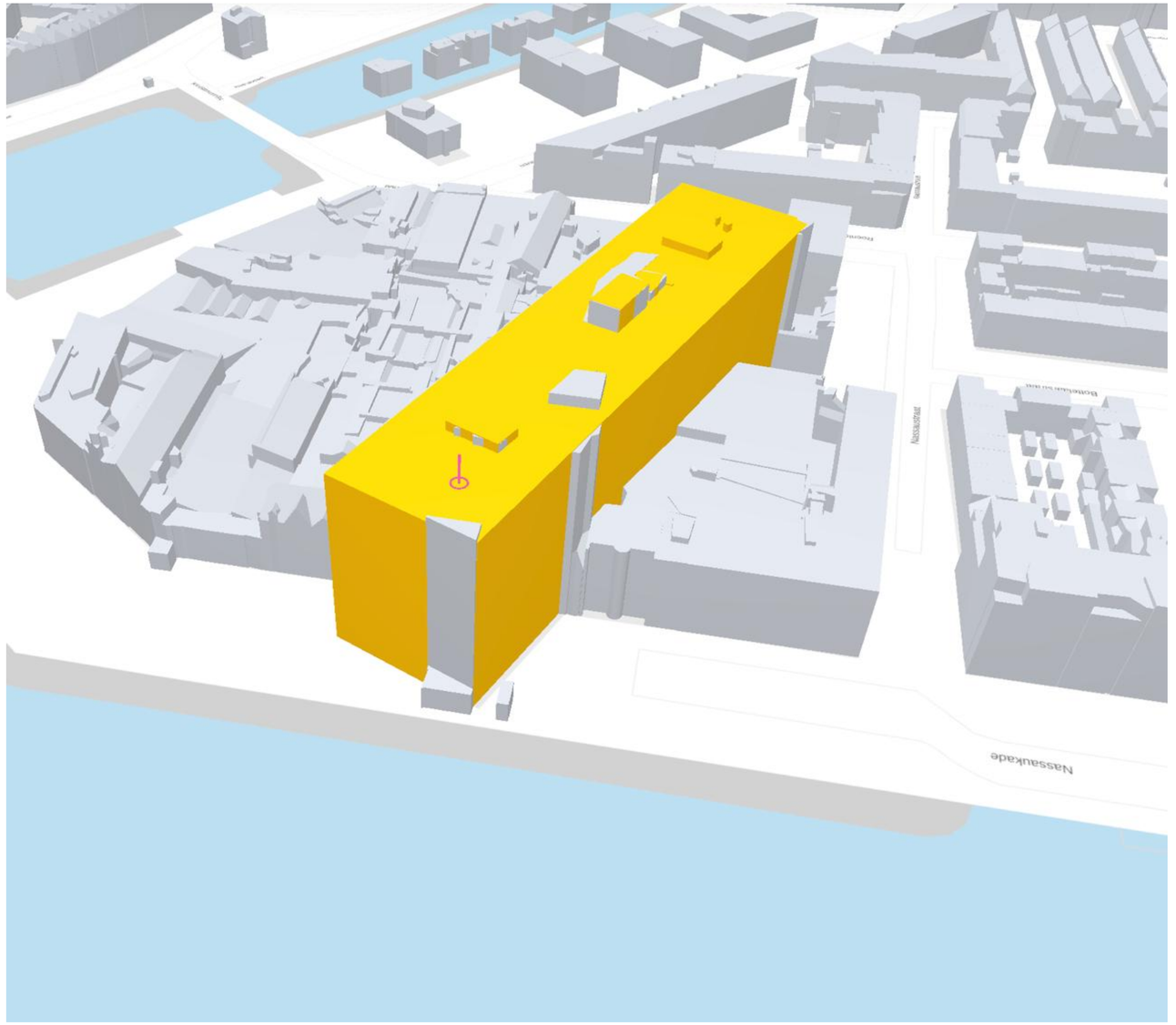


Reconstruction result

Image: Ravi Peters

3D BAG: challenges

3D BAG: challenges

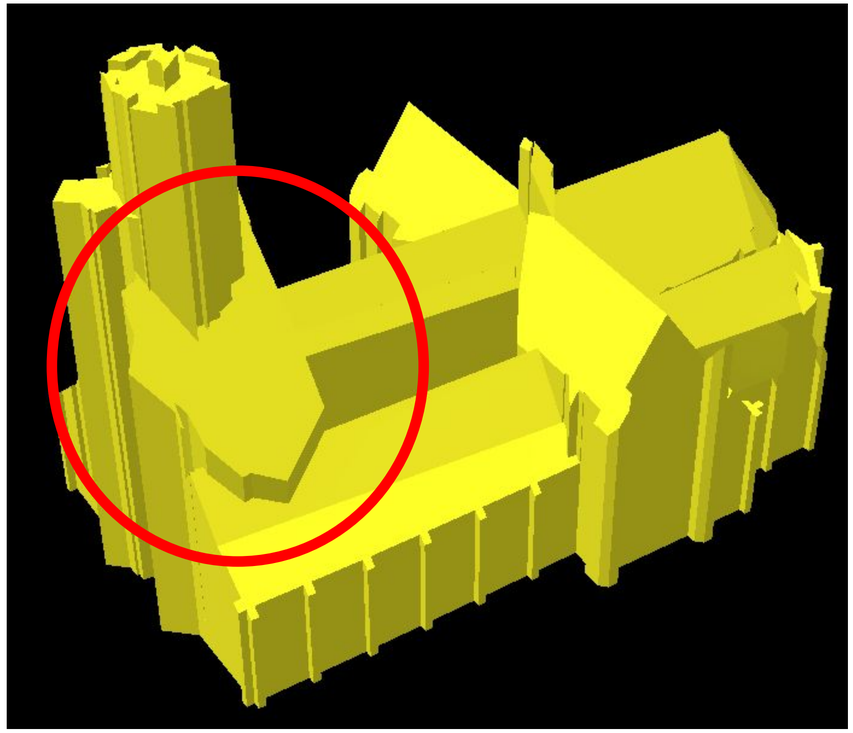


3D BAG: challenges

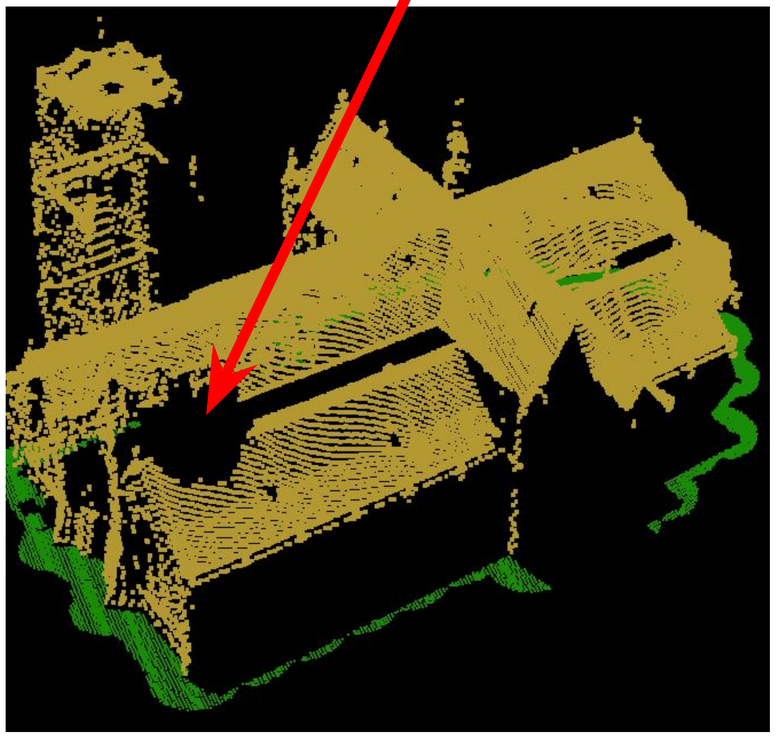
@tresdmartes

3D BAG: challenges

Occlusion/no-data



Reconstruction



AHN3 ground and building class

Heightfield

Image: Ravi Peters

3D BAG: challenges

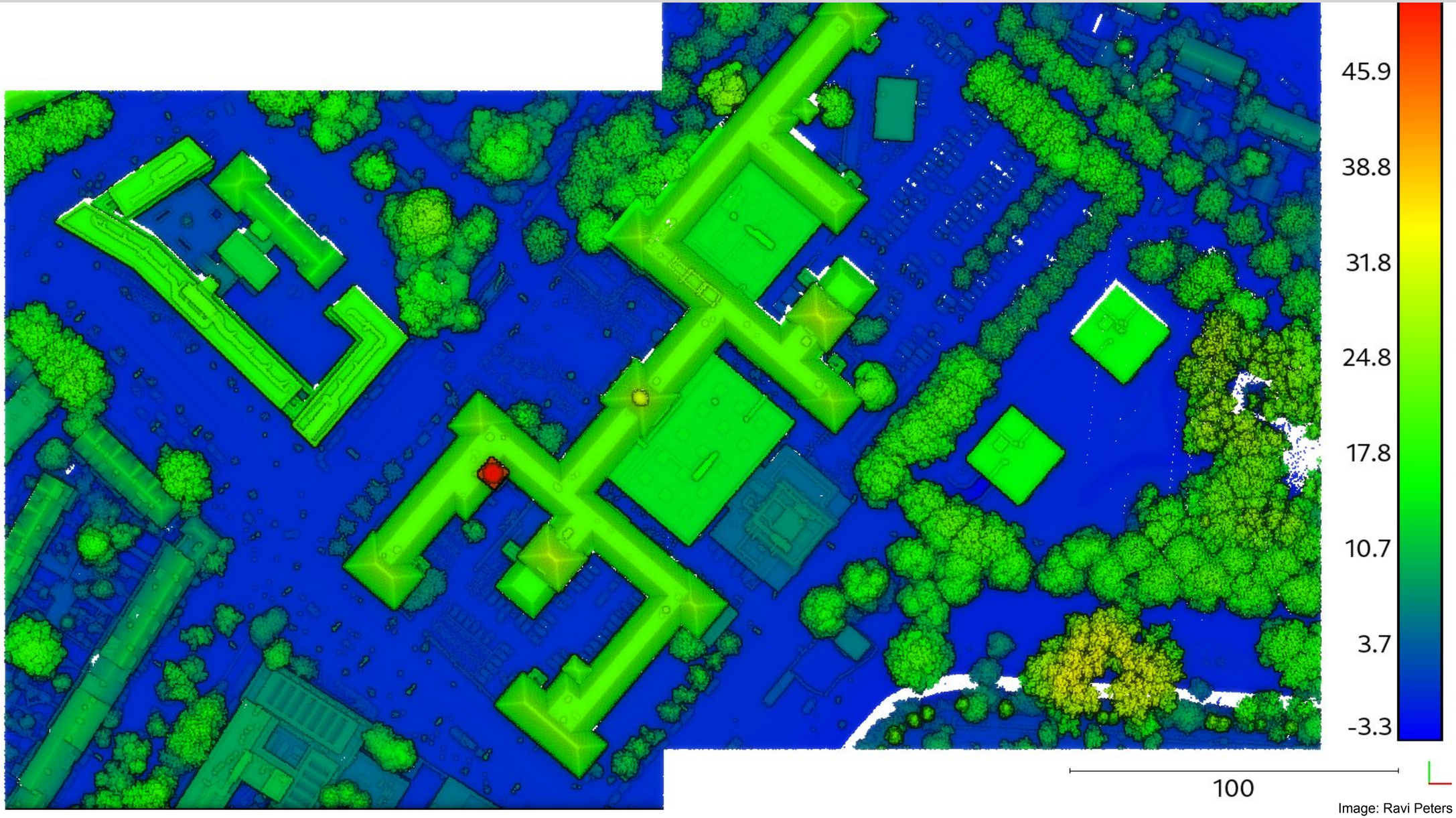


Image: Ravi Peters

3D BAG: challenges

Image: Ravi Peters

3D BAG: challenges

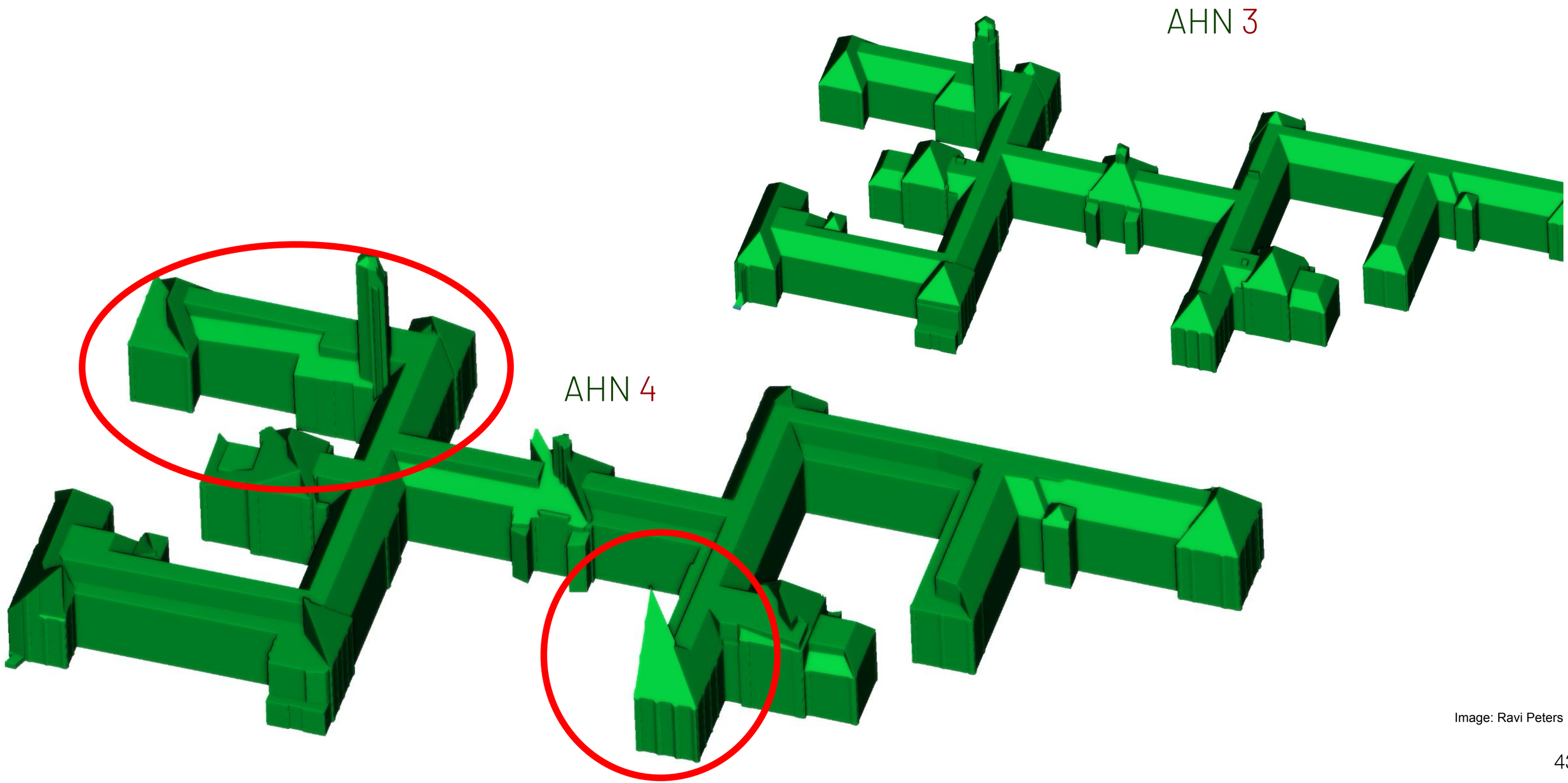
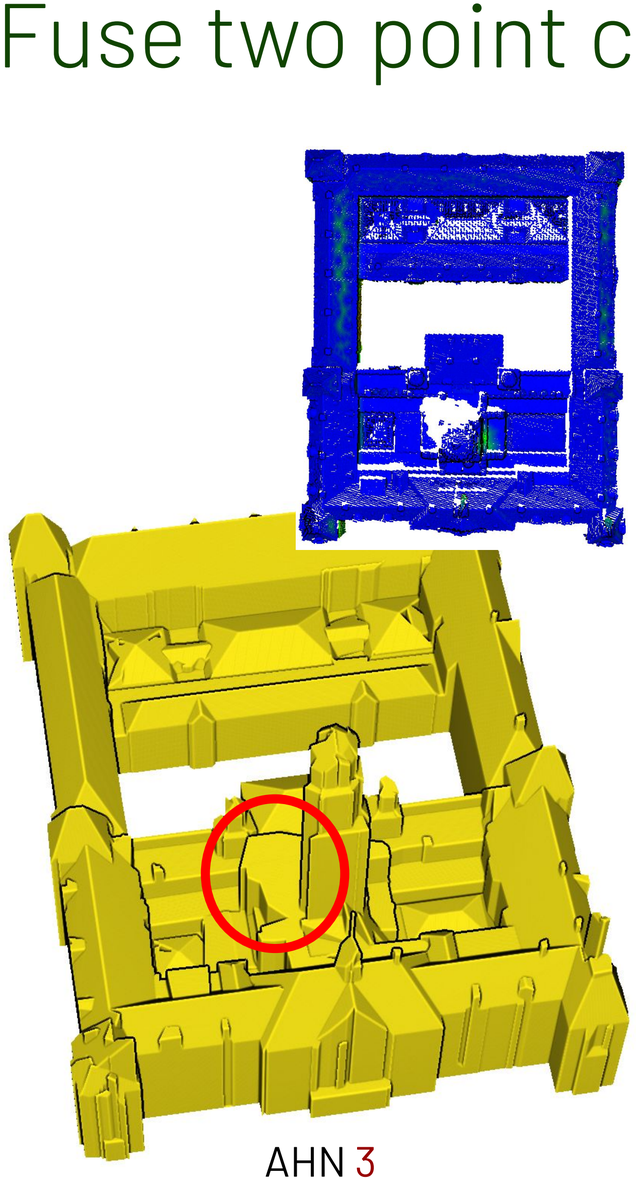
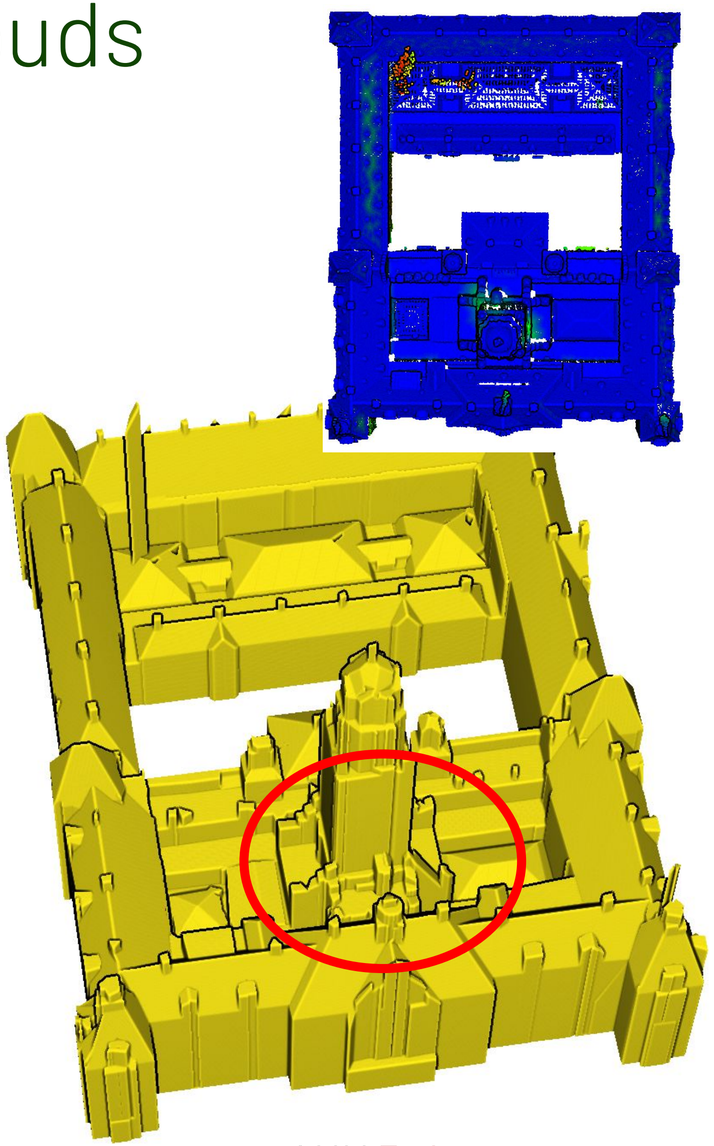


Image: Ravi Peters

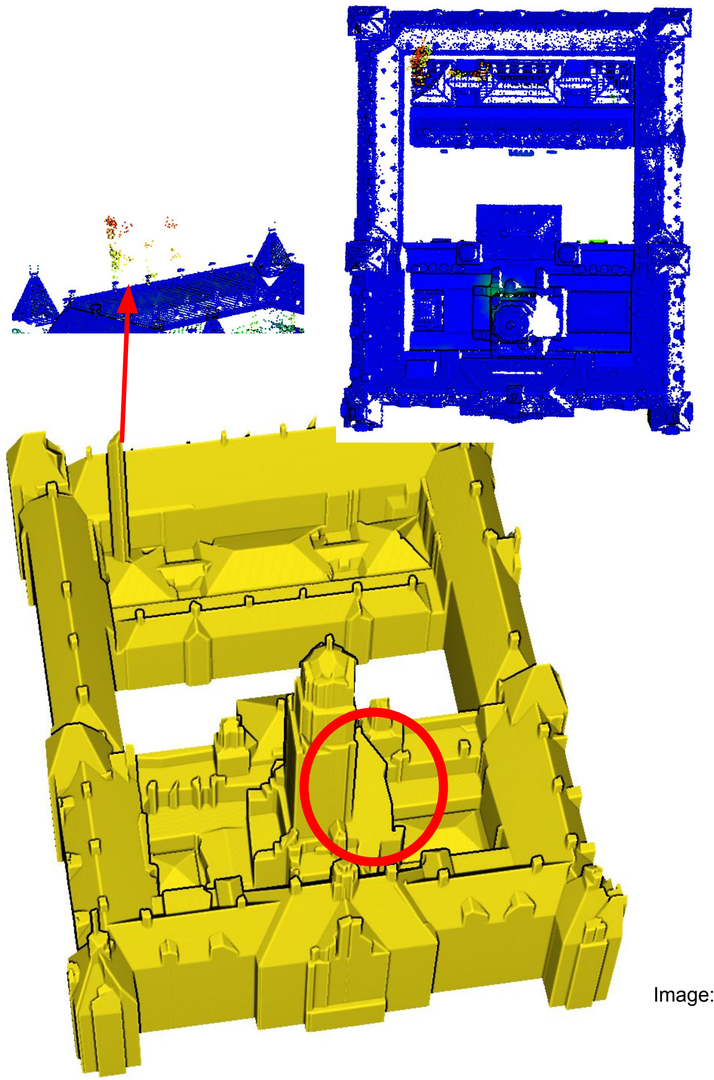
Fuse two point clouds



AHN 3



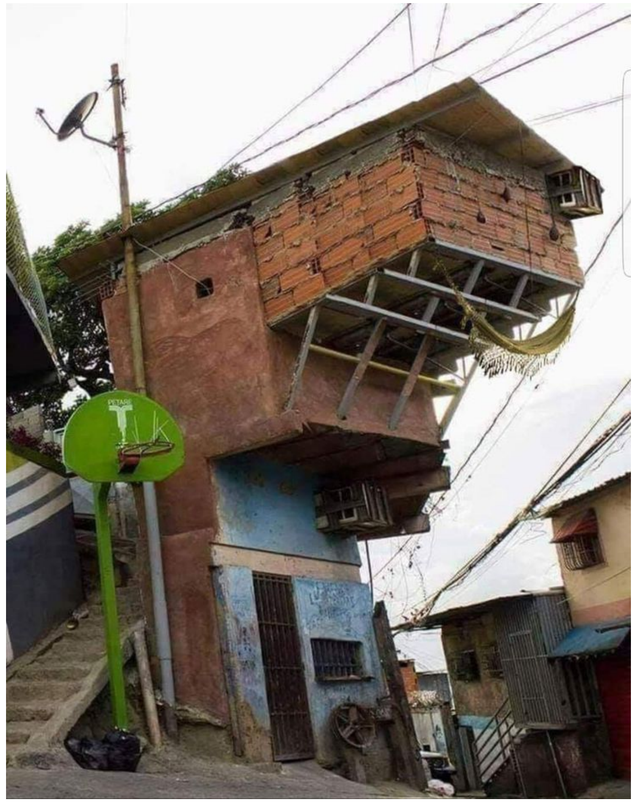
AHN 3+4



AHN 4

Image: Ravi Peters

Questions?



Module 6 contents

- Creation of 3D city models
- Processing of 3D city models + practical session
- GeoBIM integration and conversions between Geo and BIM

Processing of 3D city models contents

- Validating a 3D city model
- Operations through cjio
- Practical session

Validation of a 3D city model

Validation levels:

- syntax: is it a valid JSON or GML file?
- schema: does it conform to the CityJSON or CityGML schema?
- geometry: are the 3D primitives in the file valid?

Syntax validation (JSON)

- Types: strings, numbers, Booleans, objects, arrays and null.
- Each type has own rules, e.g. strings surrounded by double quotes.
- Plenty of tools available, e.g. JSONLint or JSON.parse() in Python.

```
{
  "first_name": "John",
  "last_name": "Smith",
  "is_alive": true,
  "age": 27,
  "address": {
    "street_address": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postal_code": "10021-3100"
  },
  "phone_numbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [
    "Catherine",
    "Thomas",
    "Trevor"
  ],
  "spouse": null
}
```

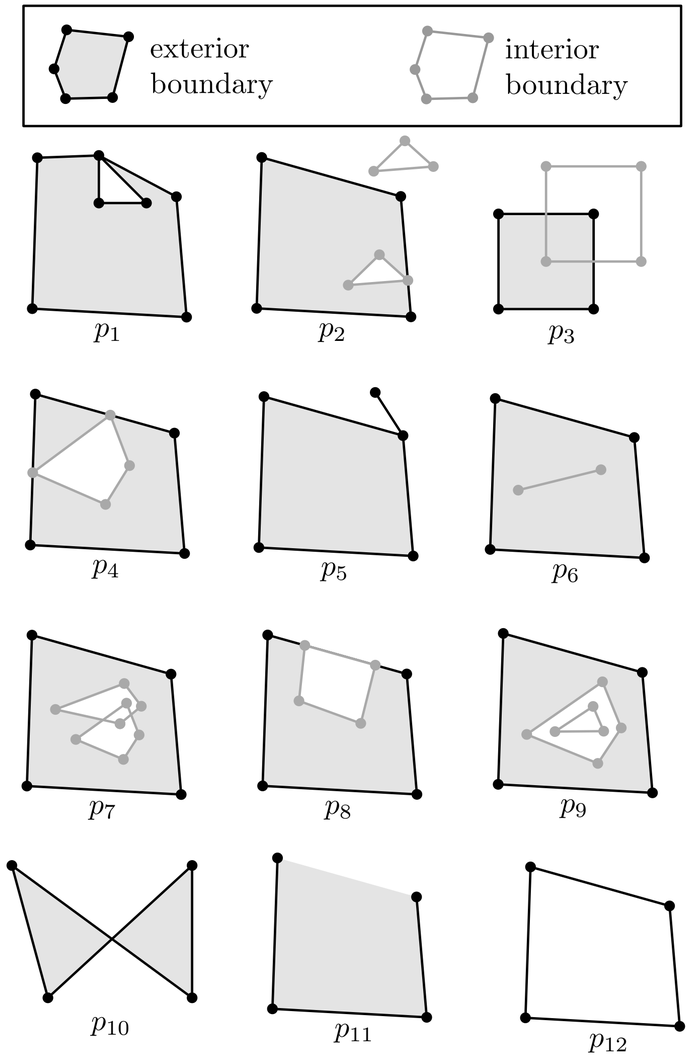
Schema validation (CityJSON)

- Checks correct structure and types using JSON Schema (https://json-schema.org/)
- Main schema for v2.0.1: https://3d.bk.tudelft.nl/schemas/cityjson/2.0.1/cityjson.schema.json
- Official validator: cjval (https://github.com/cityjson/cjval)

Geometry validation (2D)

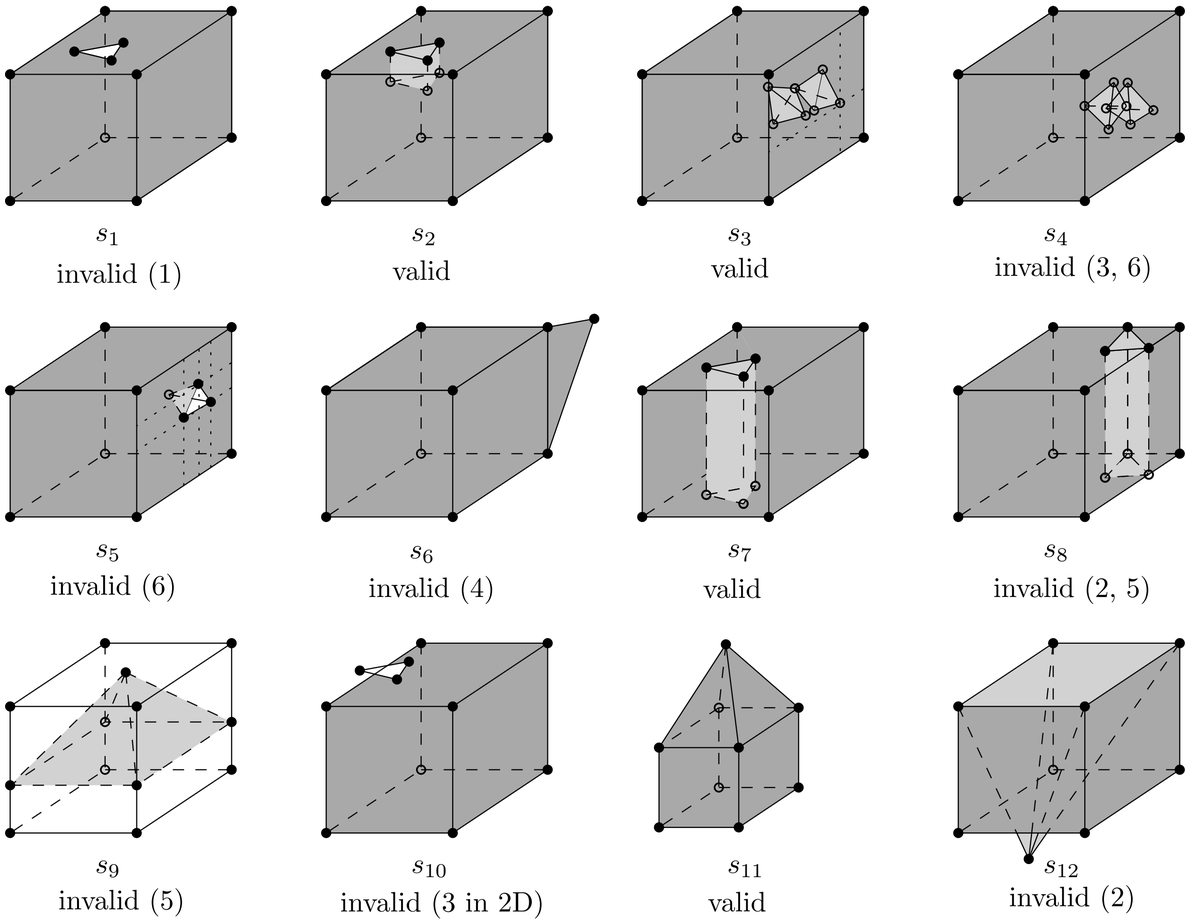
OGC Simple Features and ISO19107 rules:

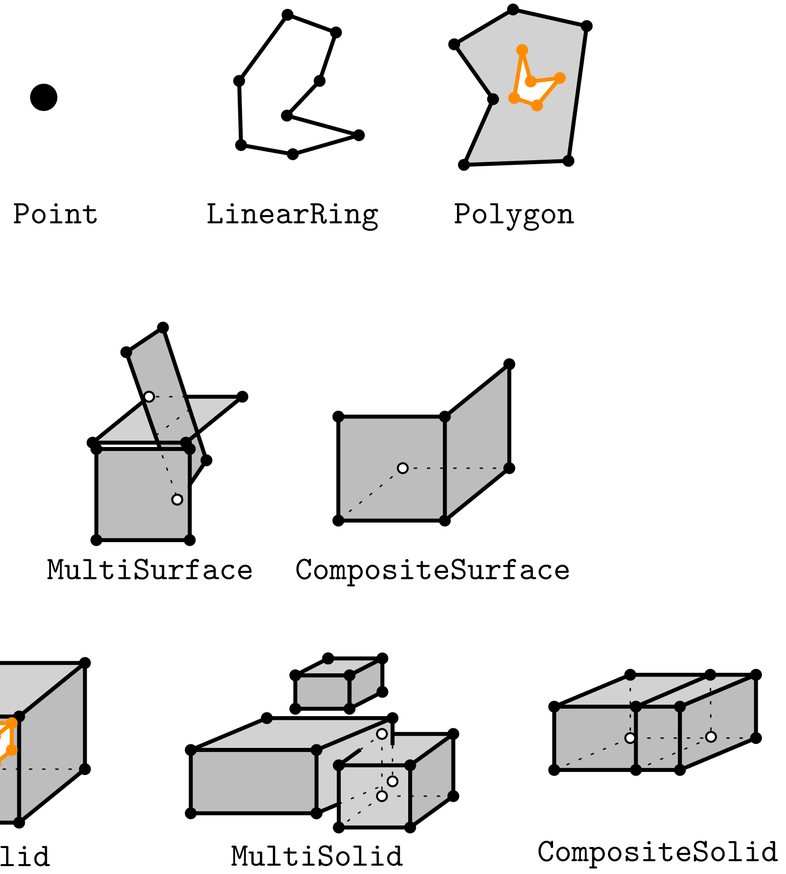
1. no self-intersection
2. closed boundaries
3. rings can touch but not overlap
4. no duplicate points
5. no dangling edges
6. connected interior
7. etc



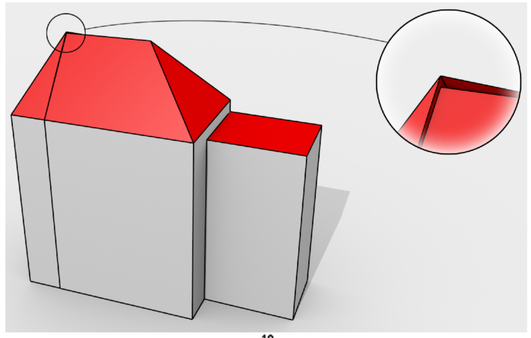
Geometry validation (2D)

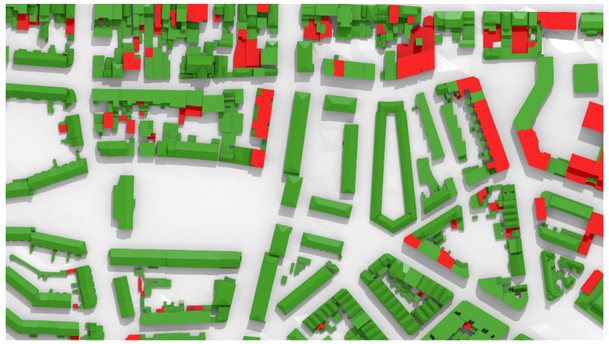
Geometry validation (3D)

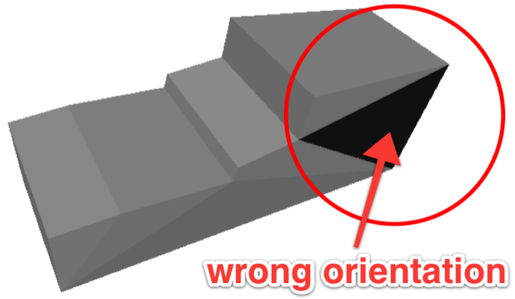


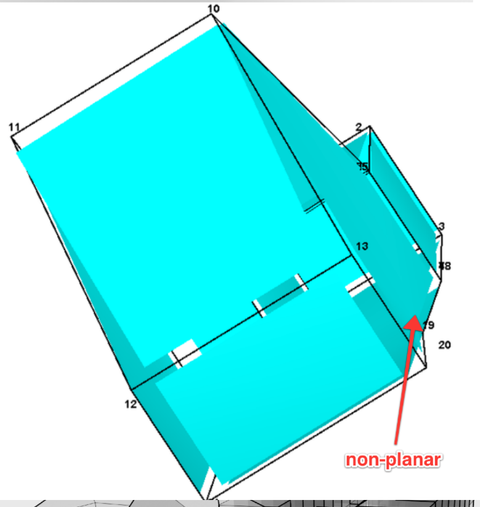


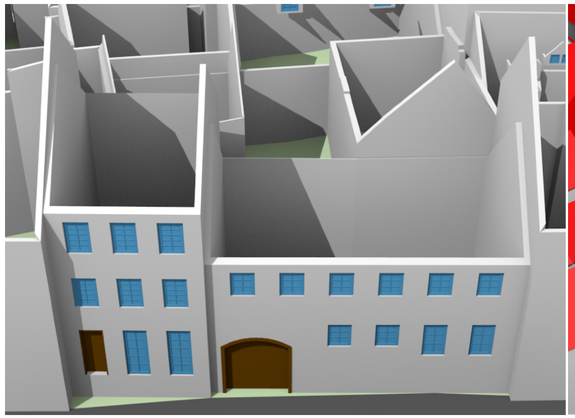
Geometry validation (3D)

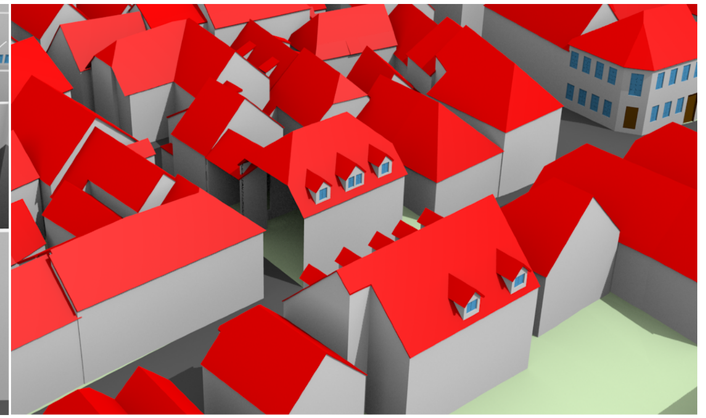


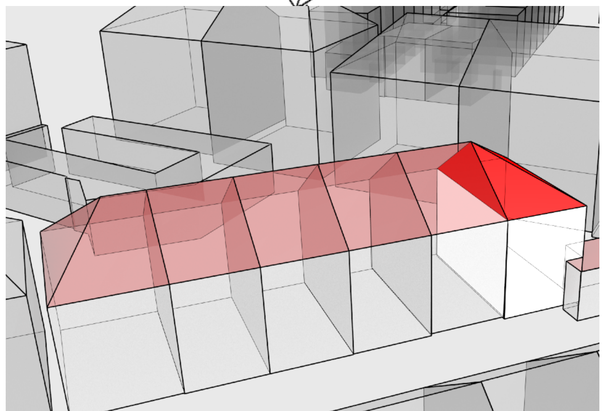


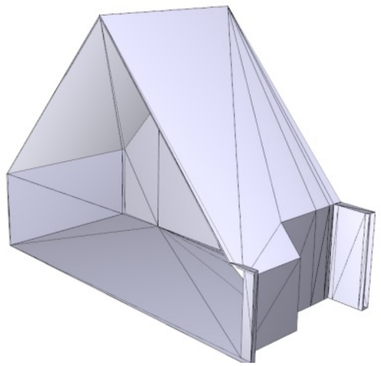


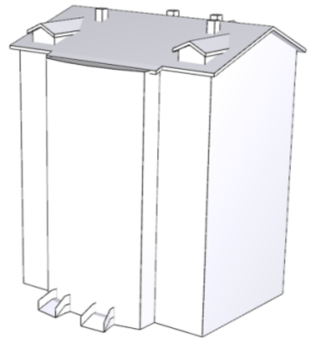


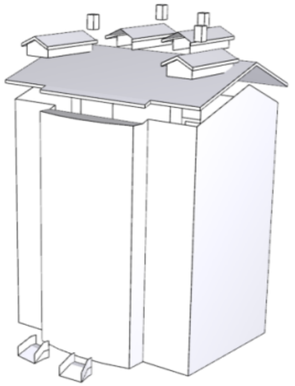




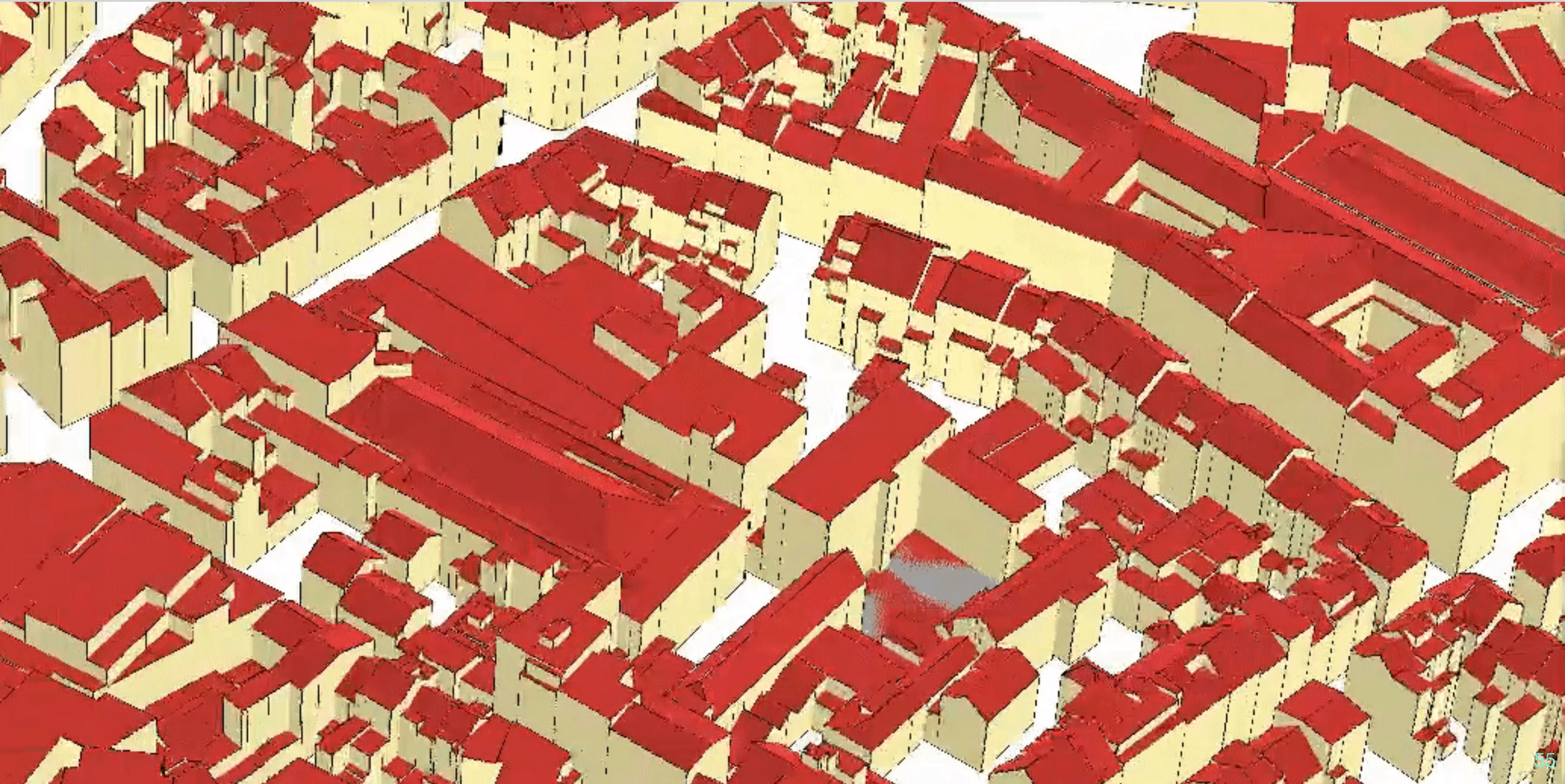




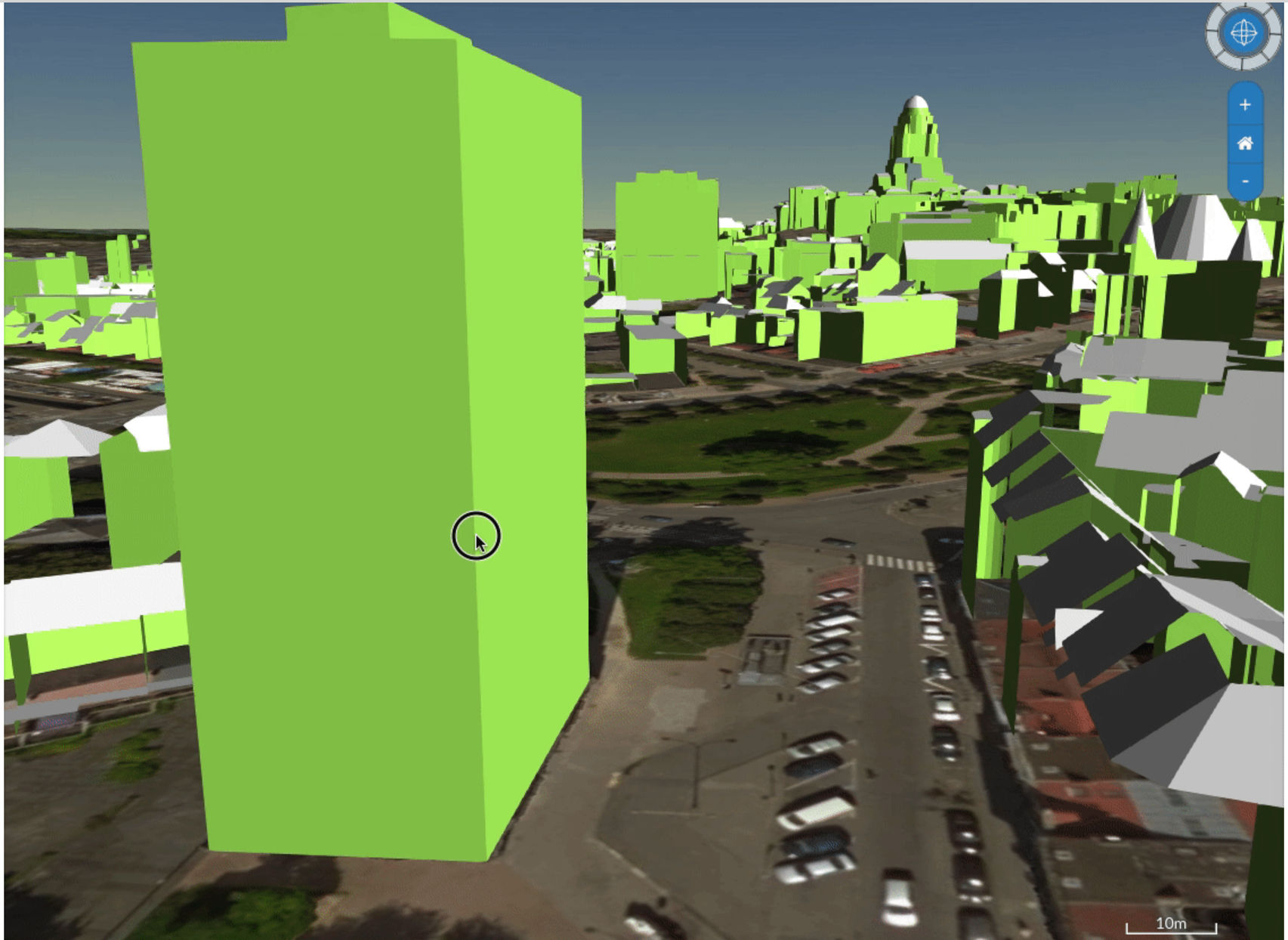




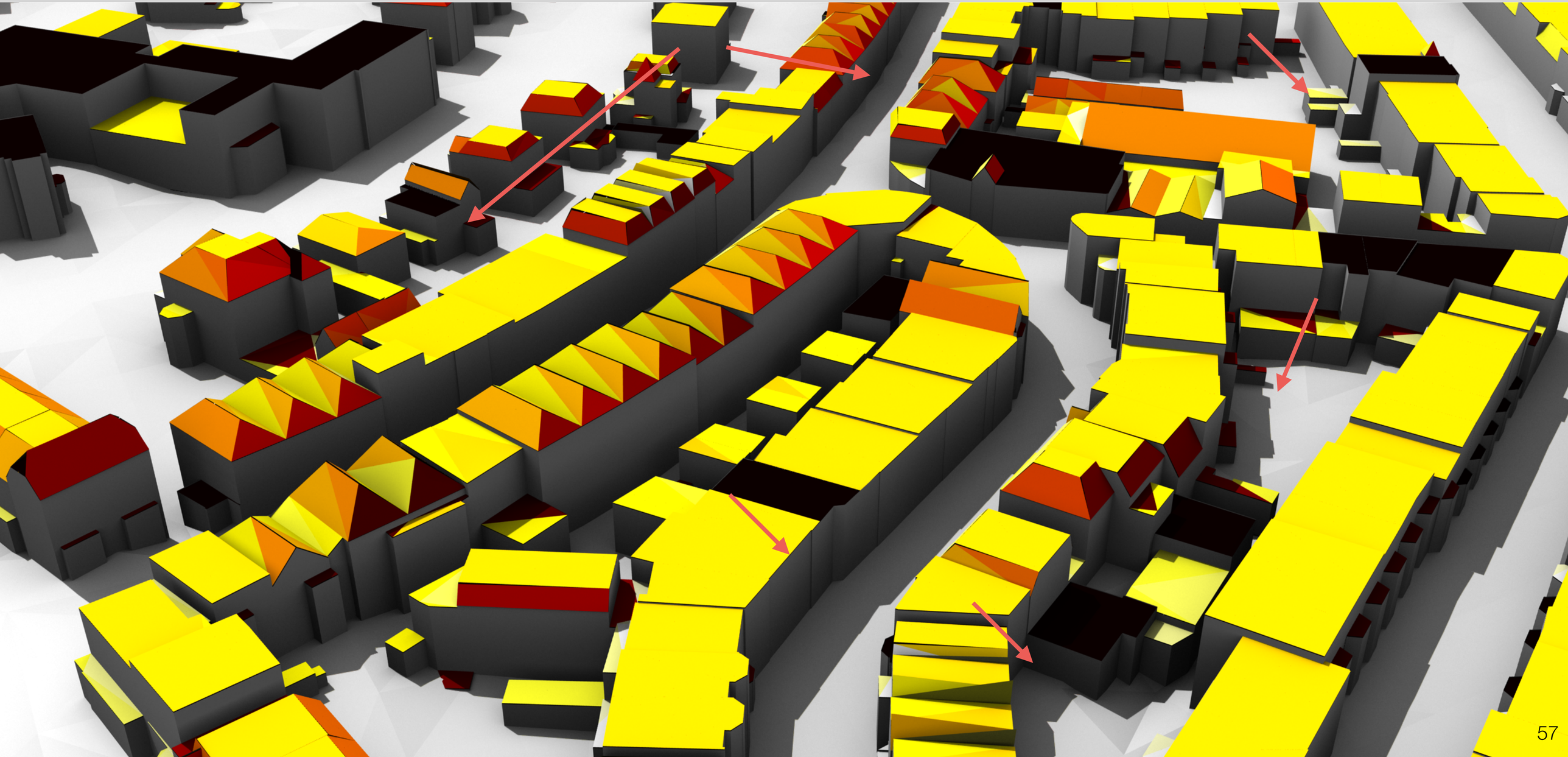
Geometry validation (3D)



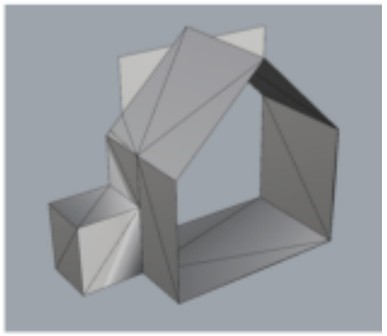
Geometry validation (3D)

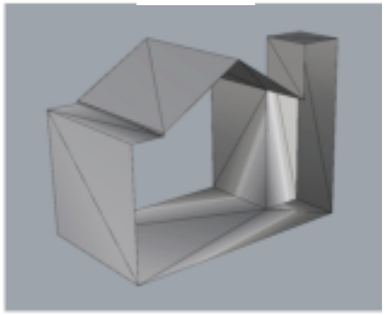


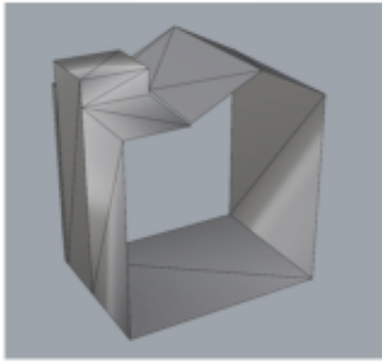
Geometry validation (3D)

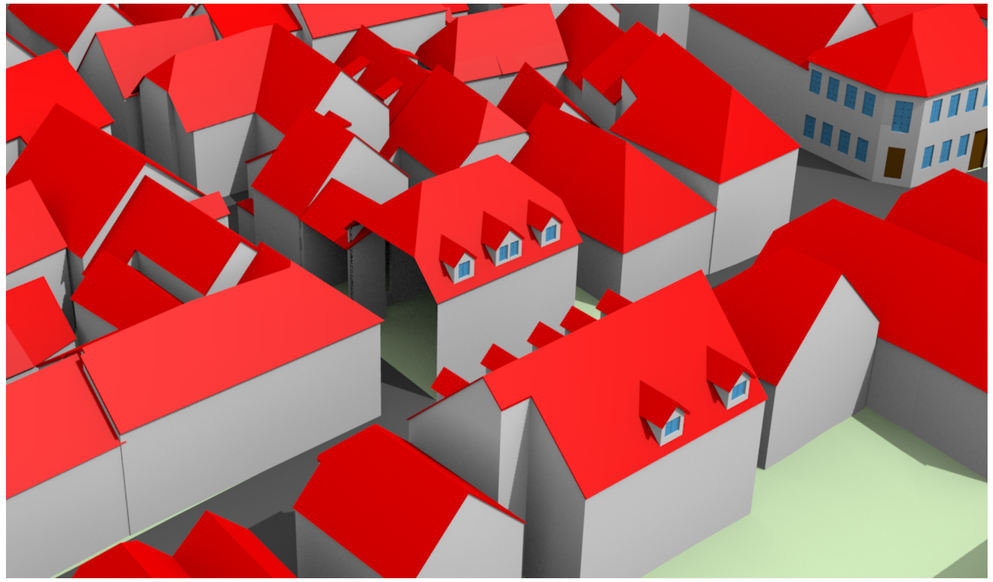


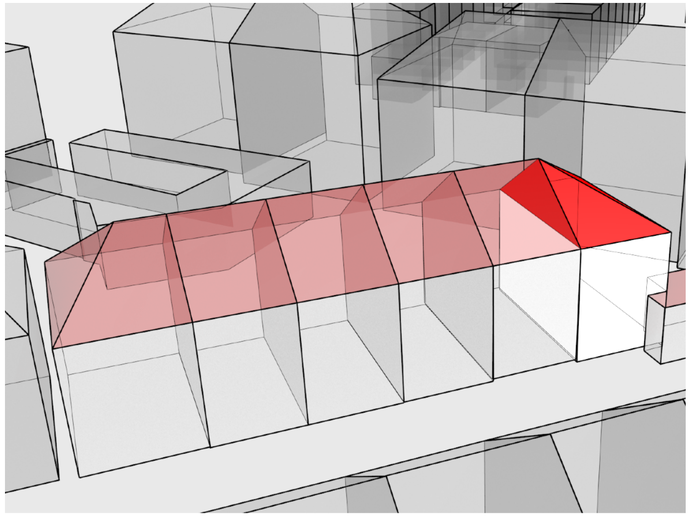
Geometry validation (3D)





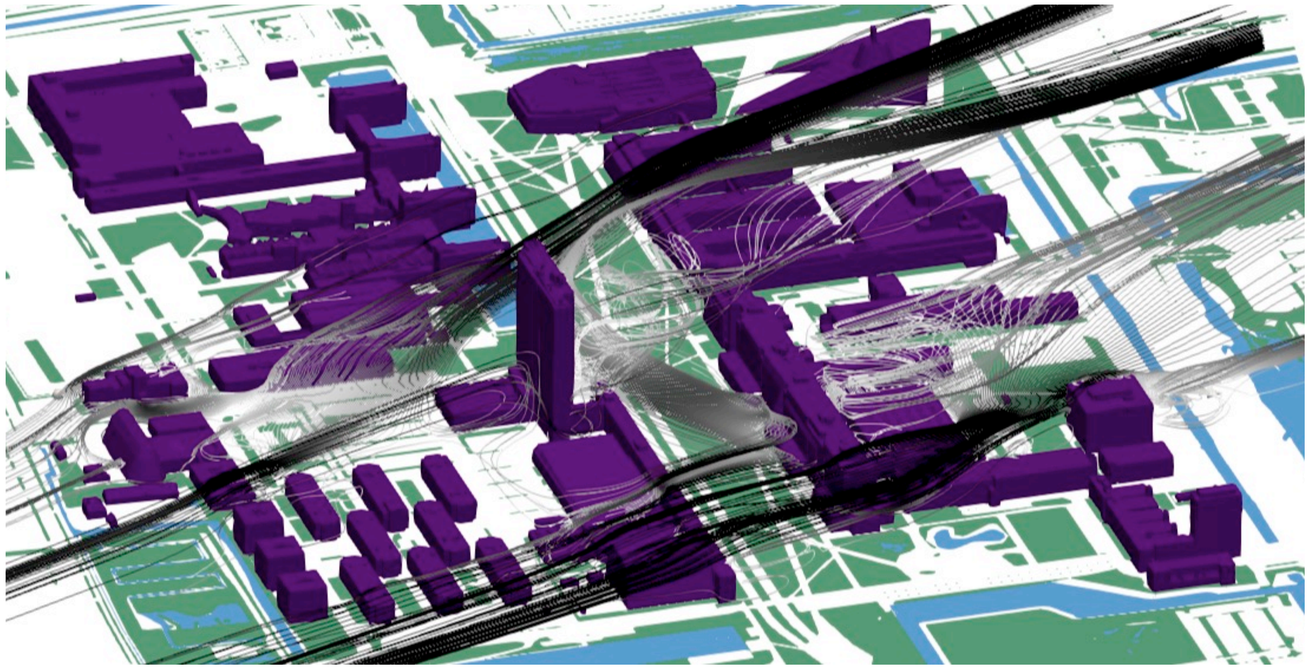






Geometry validation (3D)

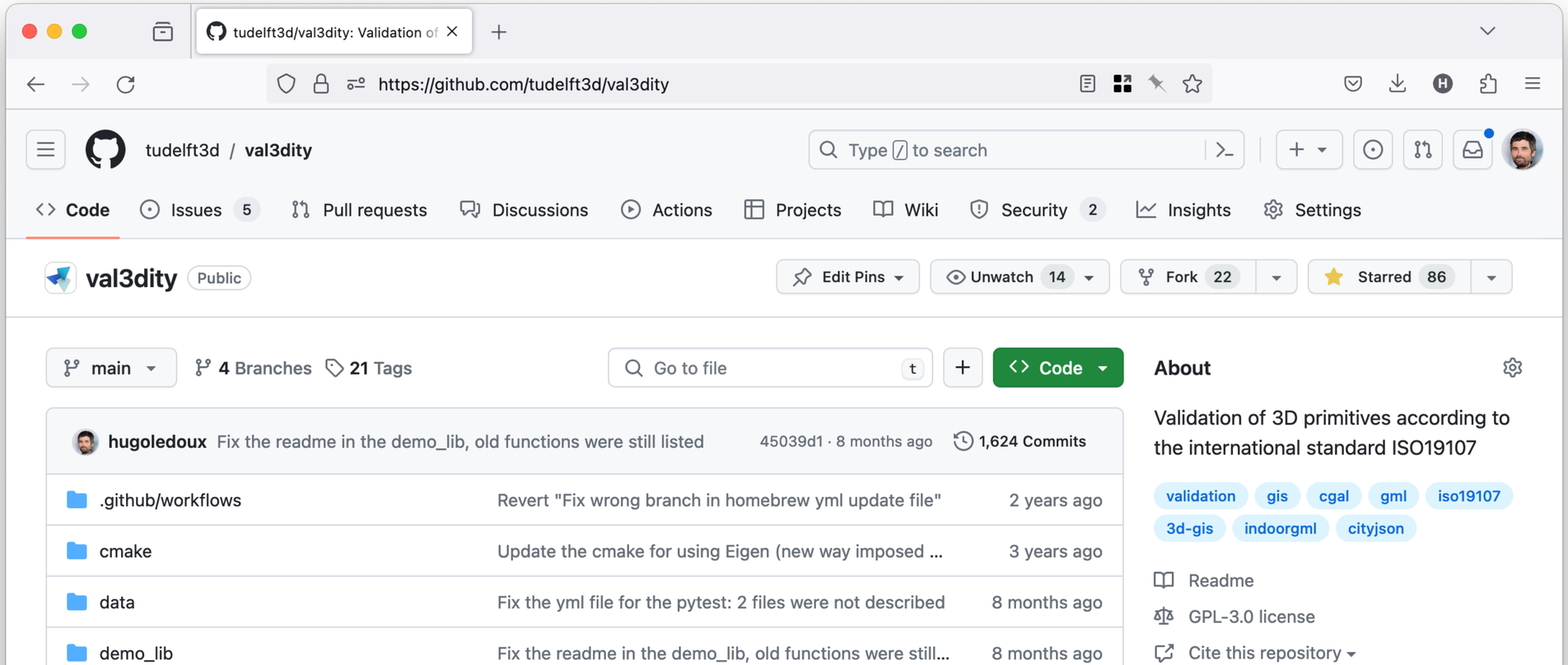
- Advanced simulations: very strict validity requirements
- In practice, up to 90% of time is spent fixing large models



Geometry validation (3D)

https://github.com/tudelft3d/val3dity (download) or

http://geovalidation.bk.tudelft.nl/val3dity/ (online for small models)



Geometry validation (3D)

CompositeSolid

Solid & MultiSolid

CompositeSurface

MultiSurface

LinearRing level

- 101 TOO_FEW_POINTS
- 102 CONSECUTIVE_POINTS_SAME
- 103 RING_NOT_CLOSED
- 104 RING_SELF_INTERSECTION

Polygon level

- 201 INTERSECTION_RINGS
- 202 DUPLICATED_RINGS
- 203 NON_PLANAR_POLYGON_DISTANCE_PLANE
- 204 NON_PLANAR_POLYGON_NORMALS_DEVIATION
- 205 POLYGON_INTERIOR_DISCONNECTED
- 206 INNER_RING_OUTSIDE
- 207 INNER_RINGS_NESTED
- 208 ORIENTATION_RINGS_SAME

Shell level

- 300 NOT_VALID_2-MANIFOLD
- 301 TOO_FEW_POLYGONS | not possible for
- 302 SHELL_NOT_CLOSED | CompositeSurface
- 303 NON_MANIFOLD_CASE
- 305 MULTIPLE_CONNECTED_COMPONENTS
- 306 SHELL_SELF_INTERSECTION
- 307 POLYGON_WRONG_ORIENTATION

Solid level

- 401 INTERSECTION_SHELLS
- 402 DUPLICATED_SHELLS
- 403 INNER_SHELL_OUTSIDE
- 404 SOLID_INTERIOR_DISCONNECTED
- 405 WRONG_ORIENTATION_SHELL

Solid interactions level

- 501 INTERSECTION_SOLIDS
- 502 DUPLICATED_SOLIDS
- 503 DISCONNECTED_SOLIDS

CityGML Objects

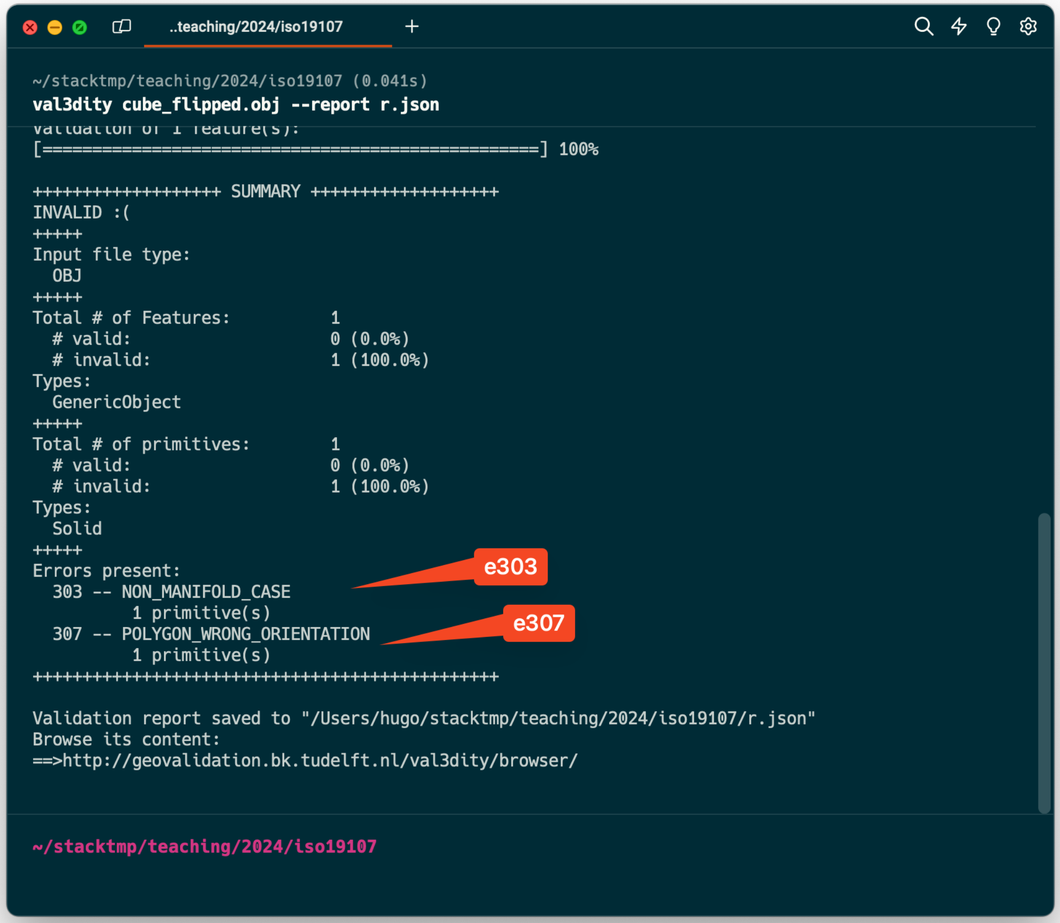
- 601 BUILDINGPARTS_OVERLAP

IndoorGML Objects

- 701 PRIMAL_CELLS_OVERLAP
- 702 DUAL_VERTEX_OUTSIDE_PRIMAL_CELL
- 703 PRIMAL_DUAL_XLINKS_ERROR
- 704 PRIMAL_DUAL_ADJACENCIES_INCONSISTENT

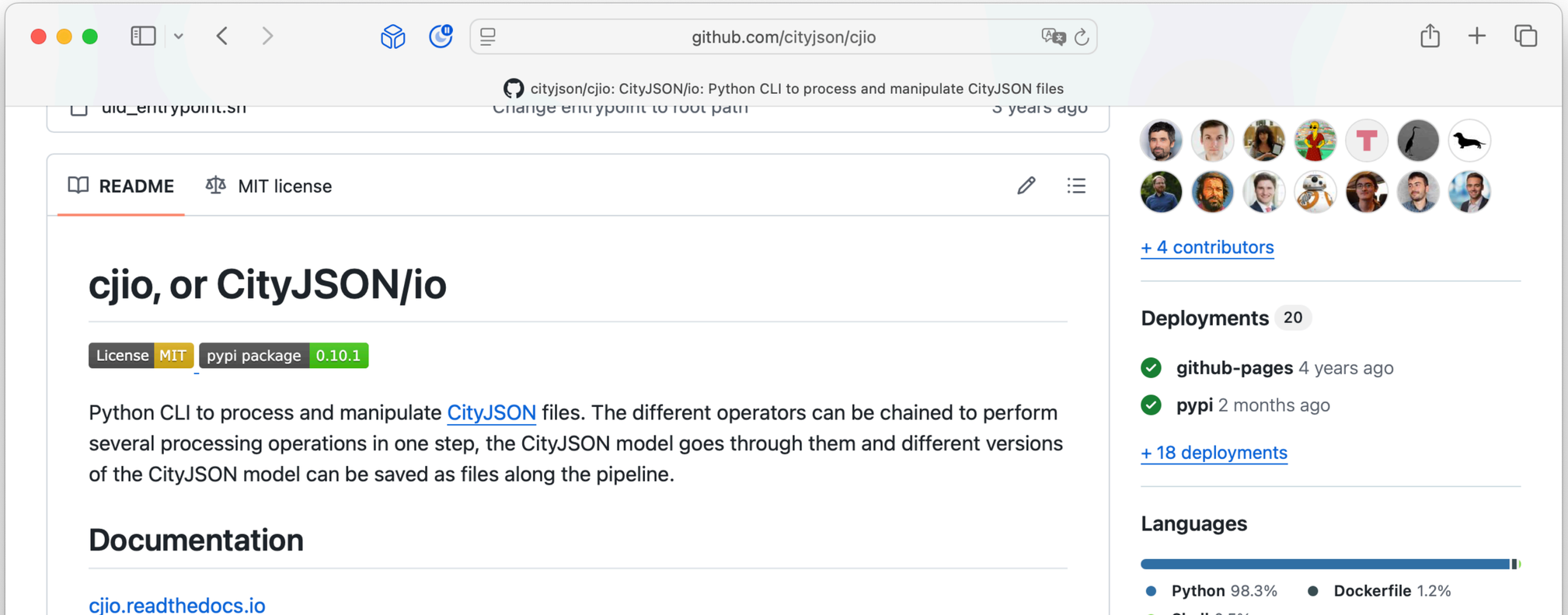
Others

- 901 INVALID_INPUT_FILE
- 902 EMPTY_PRIMITIVE
- 903 WRONG_INPUT_PARAMETERS
- 904 FORMAT_NOT_SUPPORTED
- 906 PRIMITIVE_NO_GEOMETRY
- 999 UNKNOWN_ERROR



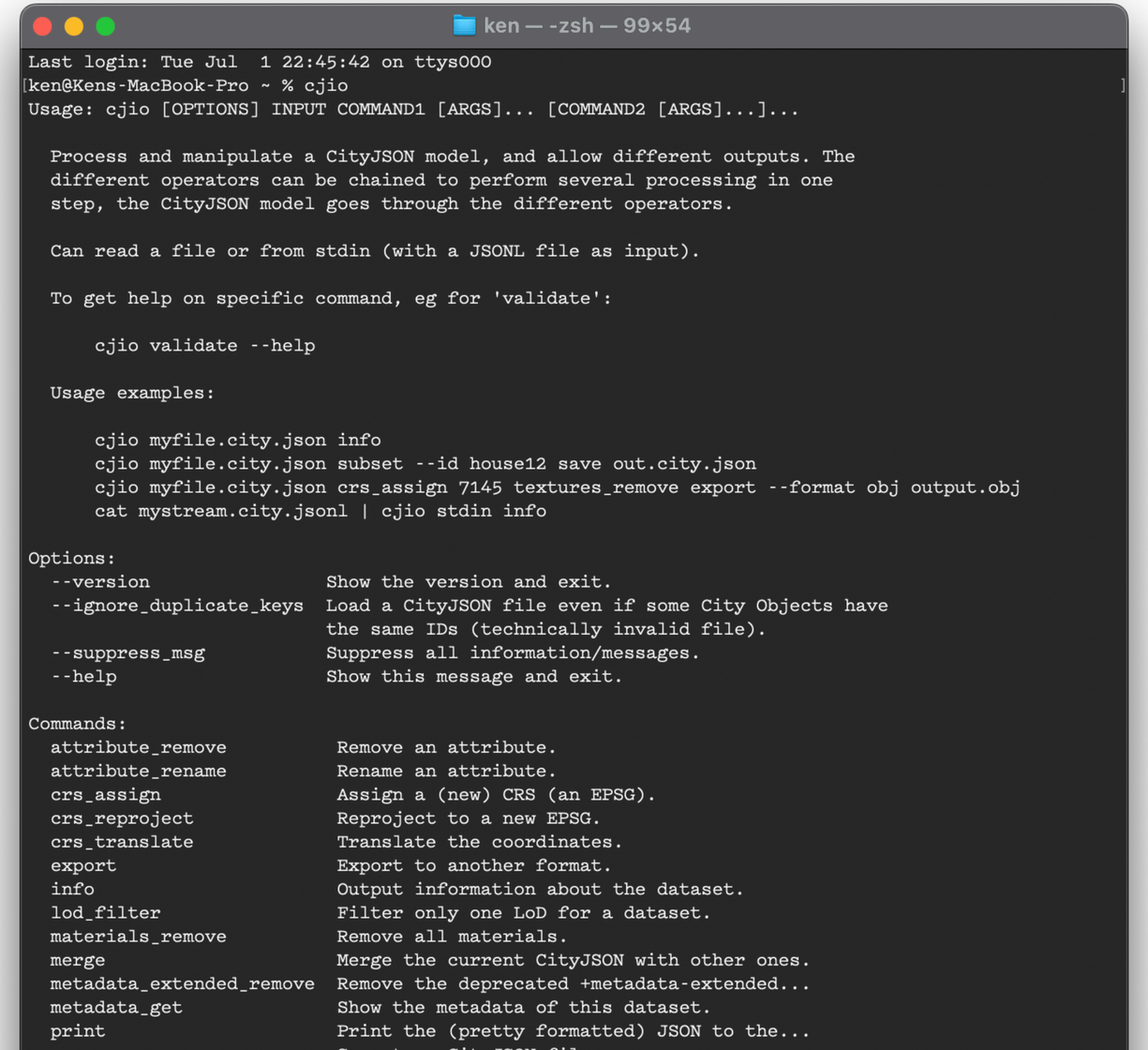
Operations through cjio

- cjio or CityJSON/io
- https://cjio.readthedocs.io/en/latest/

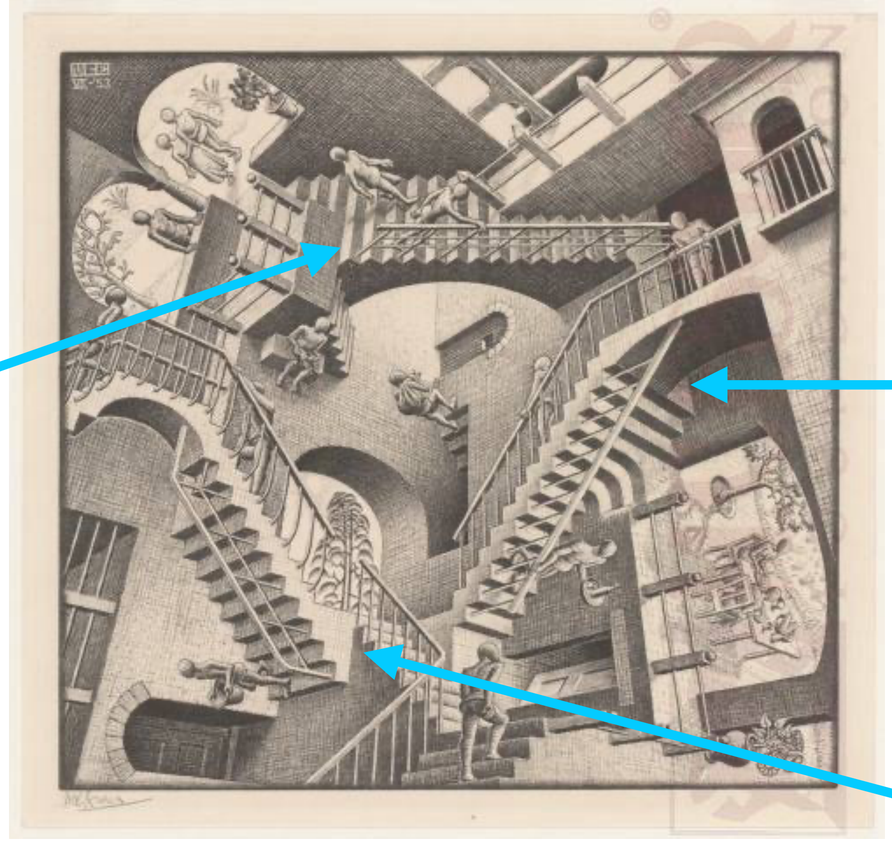


Operations through cjio

- attribute removal or renaming
- CRS assignment, reprojection or translation
- export CityJSONSeq, OBJ, glTF, b3dm or STL
- subset using id, bounding box, type, radius around point, etc.
- triangulate faces
- schema validation (using cjval)



Questions?



Practical session 4

- Create your own 3D city model using 3dfier: http://tudelft3d.github.io/3dfier/
- Find two datasets for a region of your interest:
  - A set of non-overlapping polygons that include building footprints (plus potentially other types)
  - A point cloud with elevation data, either Lidar or photogrammetric
- See tutorial: http://tudelft3d.github.io/3dfier/generate_lod1.html

Module 6 contents

- Creation of 3D city models
- Processing of 3D city models + practical session
- GeoBIM integration and conversions between Geo and BIM

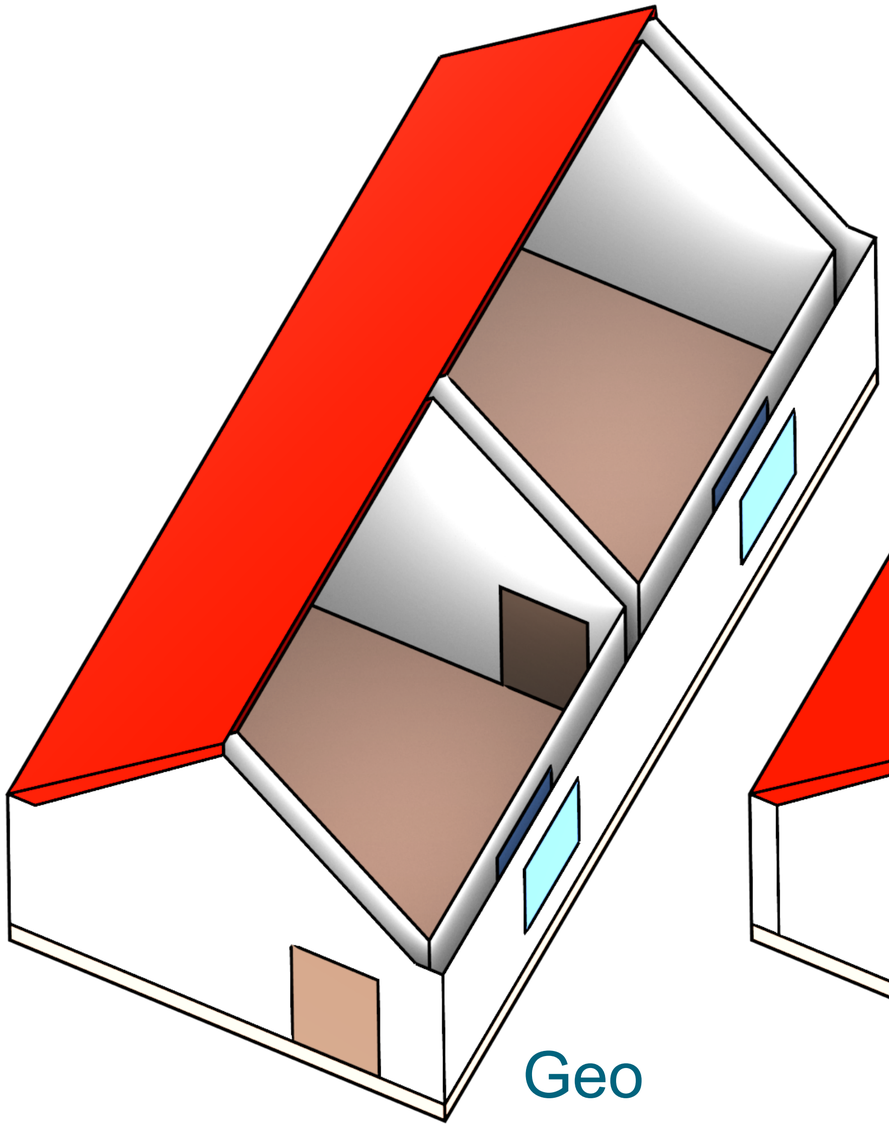
GeoBIM integration contents

- Comparison of Geo and BIM
- GeoBIM integration: 3 approaches

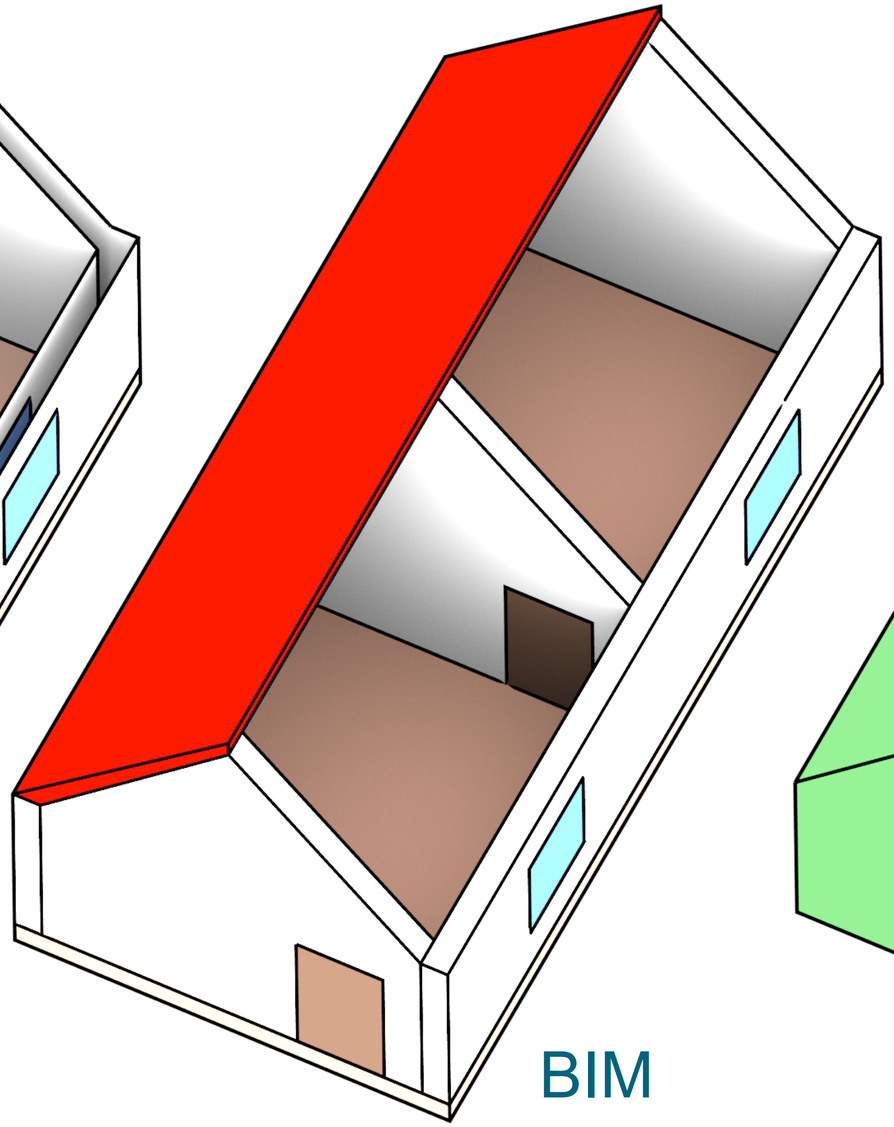
Differences between Geo and BIM

	Geo (3D city models)	BIM
origins	cartography, maps	architecture, CAD
scale	large regions	one construction site
made using	processed sensed data	usually manual as a design
detail (geom/semantics)	less detailed	very detailed
models	visible semantic surfaces	volumetric built elements
up to date	based on input data	as designed, maybe as built
focus	everything but mainly cities	buildings and infrastructure
strengths	spatial analyses	design and construction

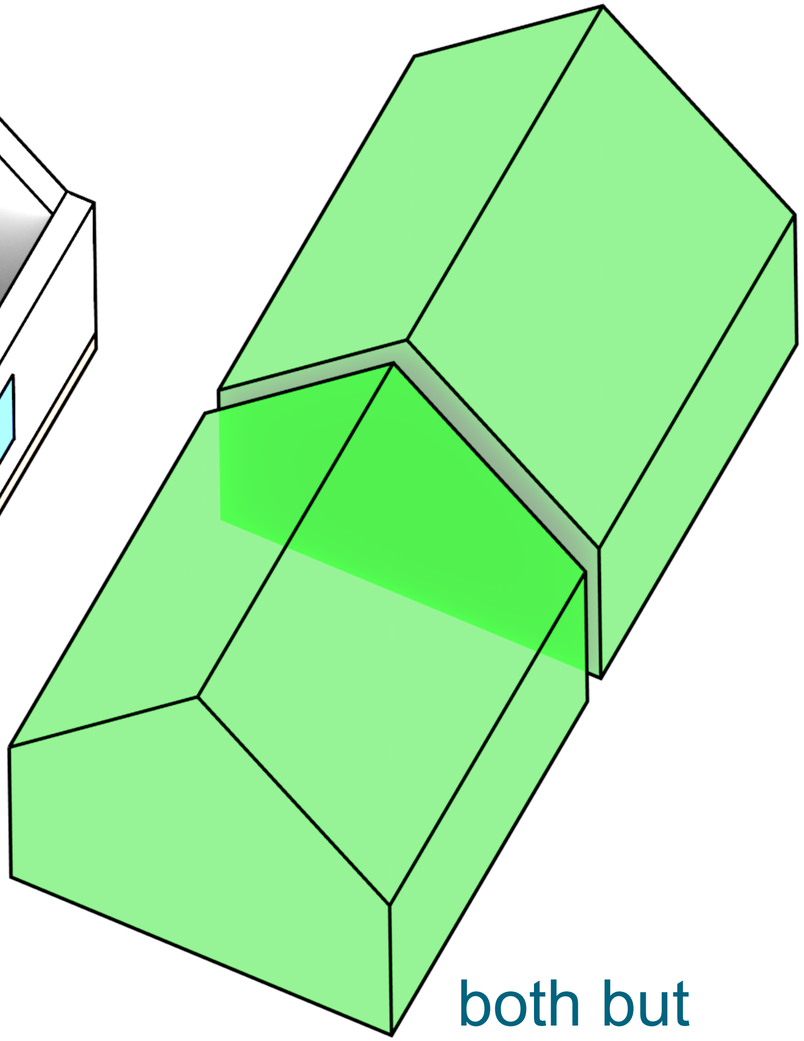
Differences between Geo and BIM



Geo



BIM



both but
with issues

GeoBIM integration

- Geo and BIM are mutually complementary, which leads us to…
- GeoBIM: joint usage of GIS (Geo) and BIM data, usually through:
  - Geo to BIM conversion (for use in BIM software)
  - BIM to Geo conversion (for use in Geo software)
  - Conversion / mapping to another model (e.g. joint model using OWL/ RDF/SPARQL or an application-specific model)

Why GeoBIM integration?

For Geo:

- High LoD cadastre including small features e.g. balconies
- Avoid (some) data acquisition
- Information about building interiors
- Continuous data updates
- Data exchange with other fields, e.g. AEC or asset management

Why GeoBIM integration?

For BIM:

- Reference context for design
- Conduct checks that require surroundings, e.g. daylight simulations
- Multi-scale vision for applications, e.g. asset management
- Data exchange with other fields, e.g. planners and environmental engineers
- Techniques for scan to BIM come from Geo side

Why GeoBIM integration?

Shared benefits:

- Data interoperability
- Cost and time savings
- Digital twin development
- Sustainability and resource planning

Integration vs. Interoperability

- **Interoperability** is the ability of systems or products to operate effectively and efficiently in conjunction, on the exchange and reuse of available resources, services, procedures, and information, in order to fulfil the requirements of a specific task (Kavouras and Kokla, 2007).

- **Integration** is the combination or conflation of information from different data sets (Worboys, Duckham, 2004).

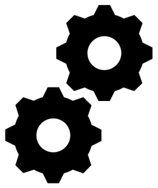
Integration vs. Interoperability

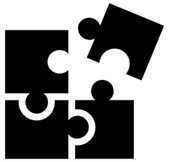
Data **Interoperability** (of a single dataset)
=
Data format
+
Info enabling data exchange and reuse (metadata)
+
Alignment with use case requirements
+
Alignment with software/application requirements

Including

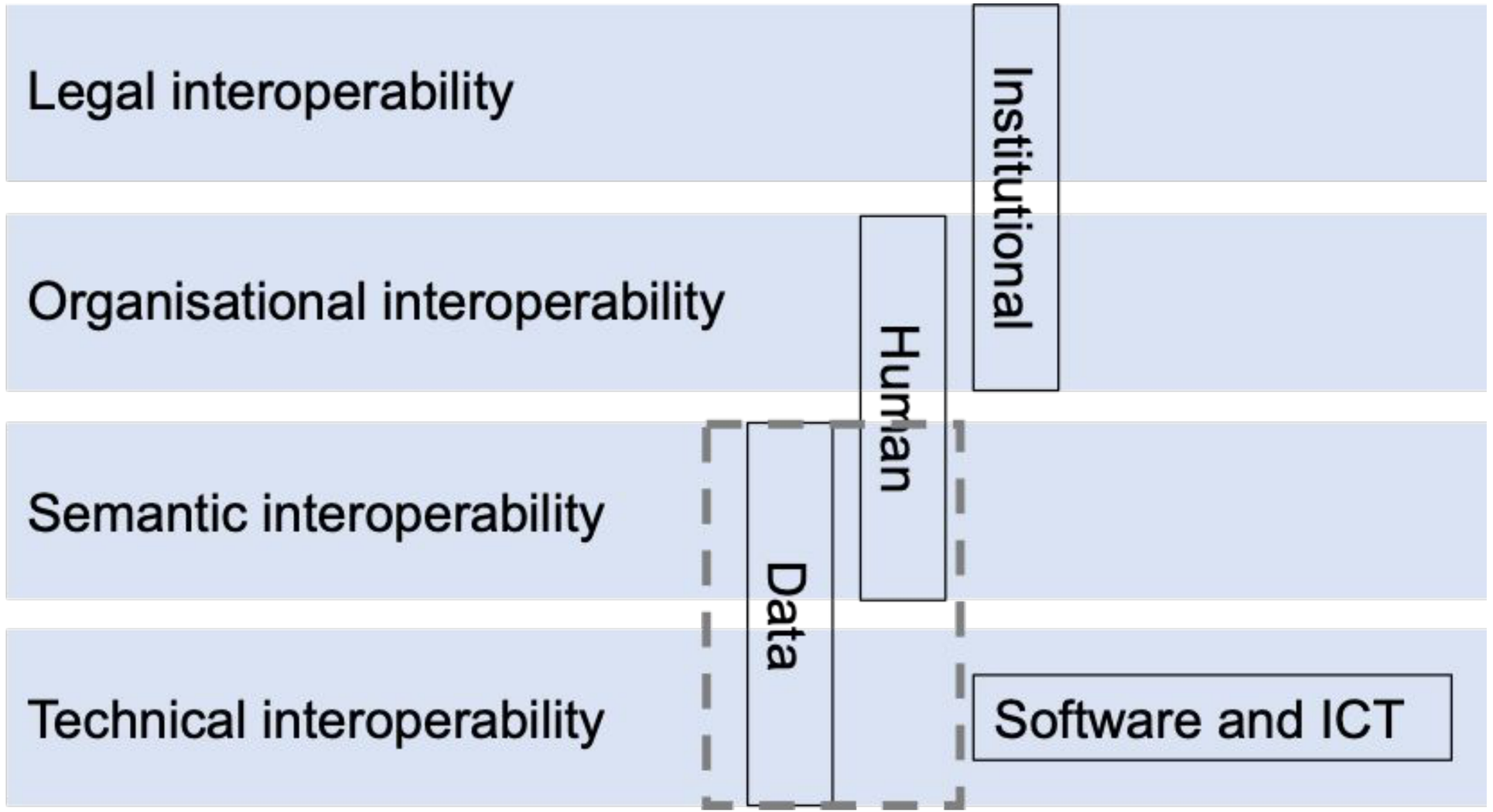
Data **Integration** (of two or more datasets)
=
Data harmonization
+
Data Merge/fusion/combine

Guided by





Integration vs. Interoperability



Integration vs. Interoperability

-

Open data models

Open protocols definitions and software components

Best practices and users agreements

CityGML CityJSON

LandInfra/InfraGML

IndoorGML

INSPIRE: Infrastructure for Spatial Information in Europe

gbXML

buildingSMART International

Home | Standards | Services | Resources

Data standards	Workflow standards	Interface standards (APIs)
Industry Foundation Classes (IFC)	Information Delivery Specification (IDS)	Shared/Common API (Foundations API)
› IFC Specifications database	BIM Collaboration Format (BCF)	BCF API
› IFC Formats	› bcfXML	Documents API
› Model View Definitions (MVD)	Information Delivery Manual (IDM)	Property Exchange API (coming soon)
› MVD Database	› IDM Database	

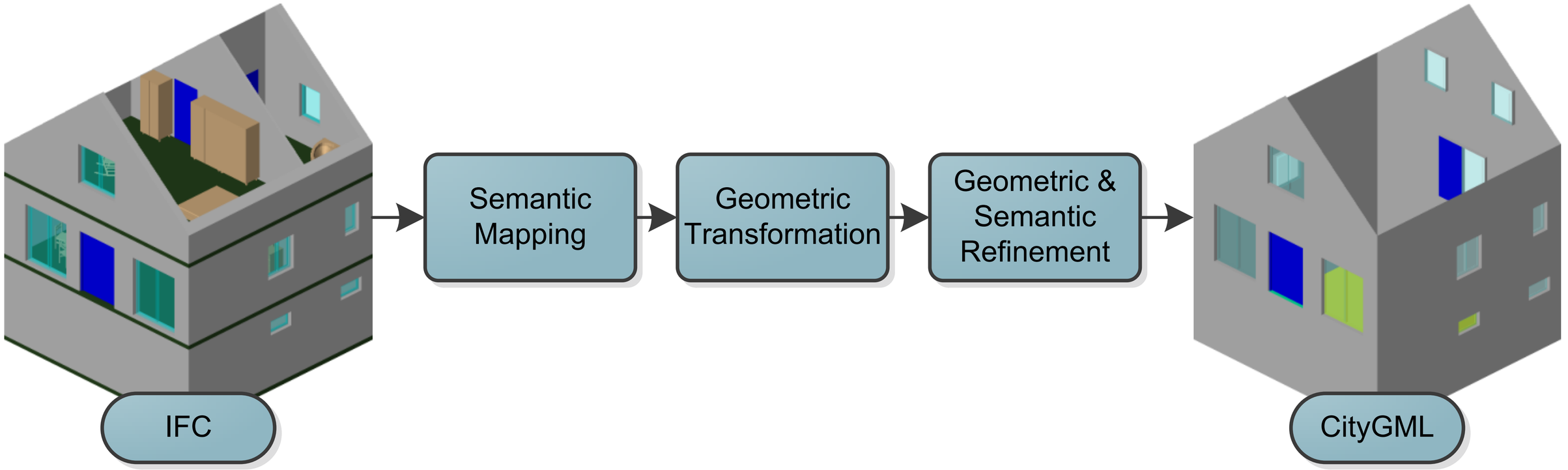
OGC APIs | Building Blocks for Location

GeoBIM integration contents

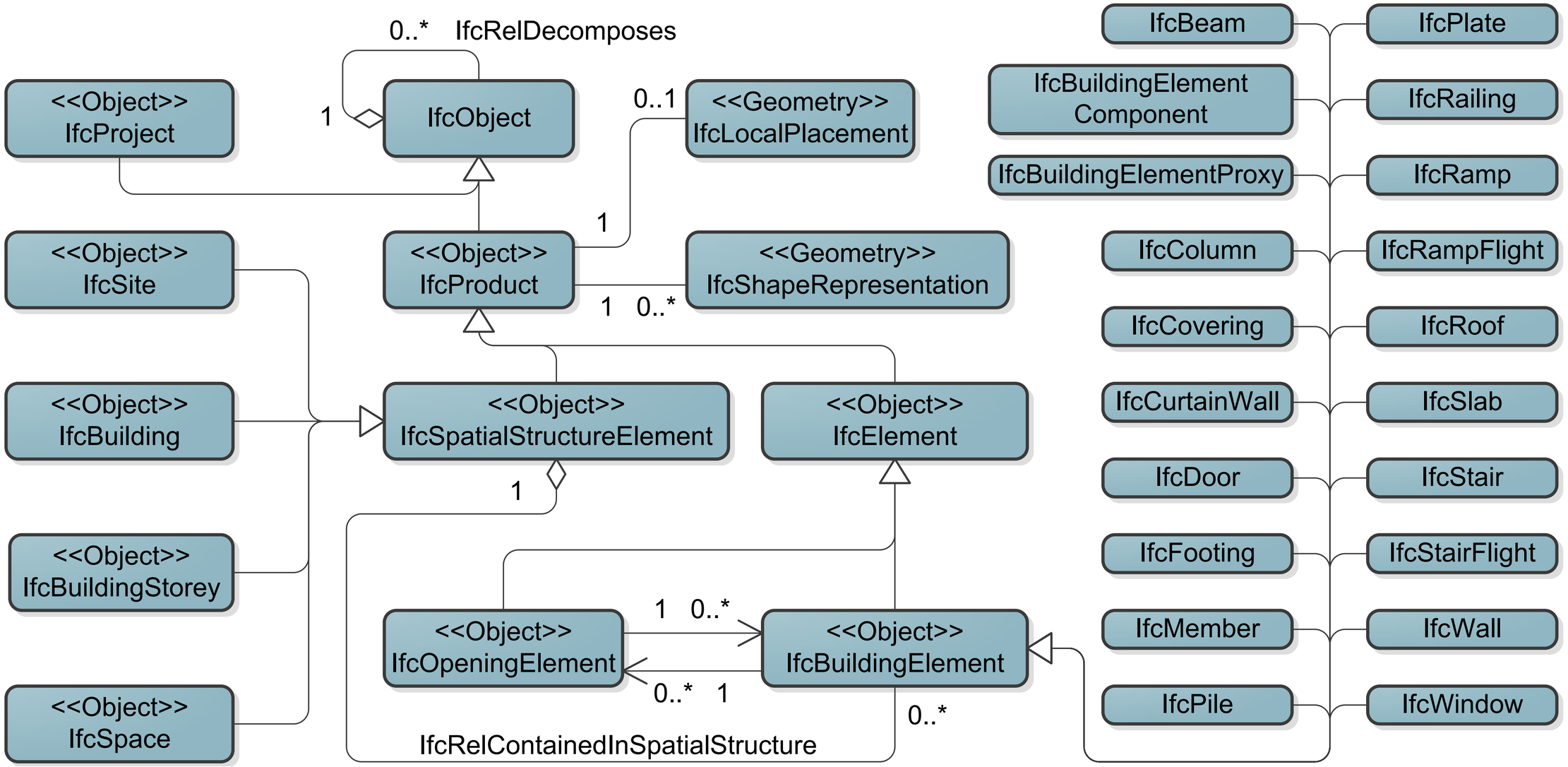
- Comparison of Geo and BIM
- GeoBIM integration: 3 approaches

Image: GIM International, 2021

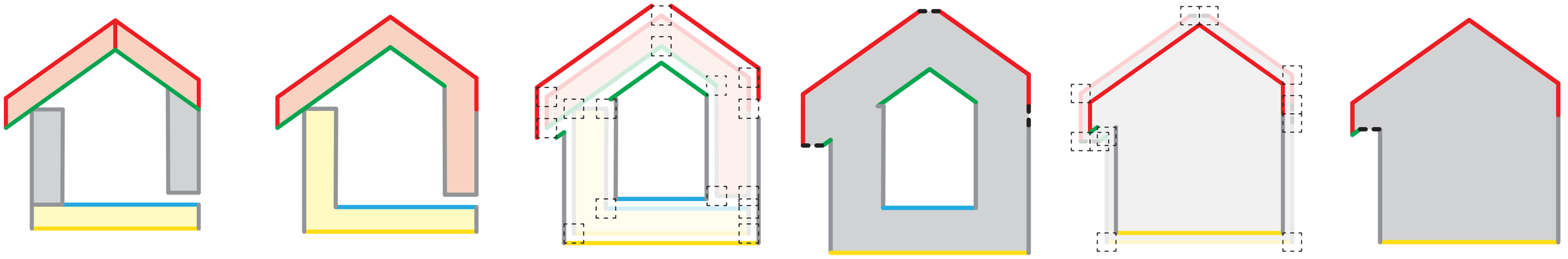
BIM to Geo conversion



BIM to Geo conversion

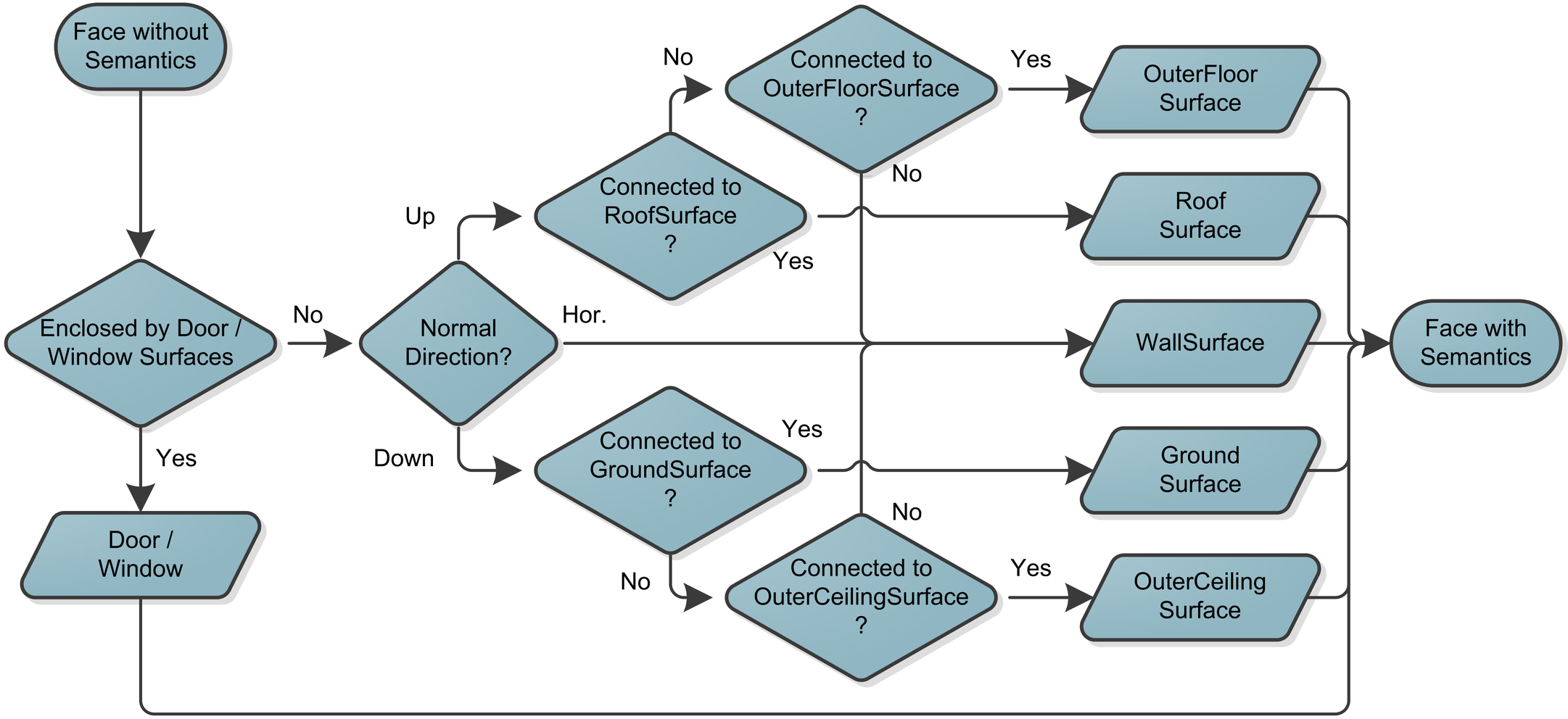


BIM to Geo conversion

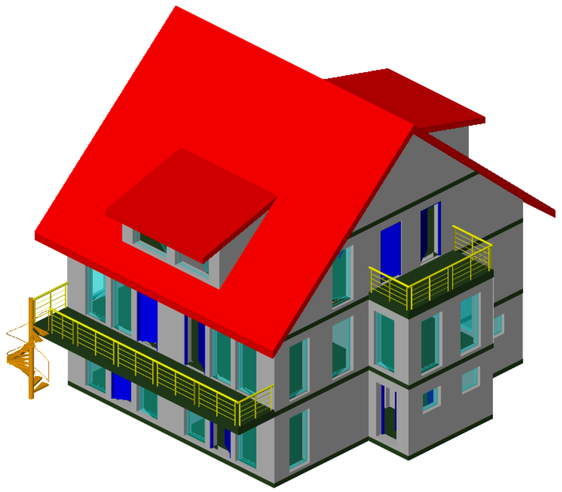


(a) input (b) union (c) dilation (d) result (e) erosion (f) final result

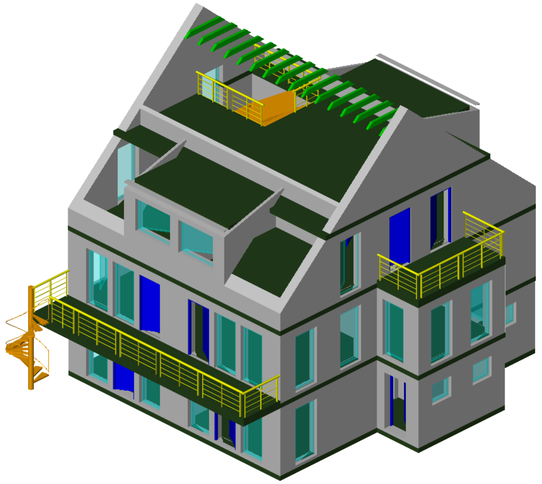
BIM to Geo conversion



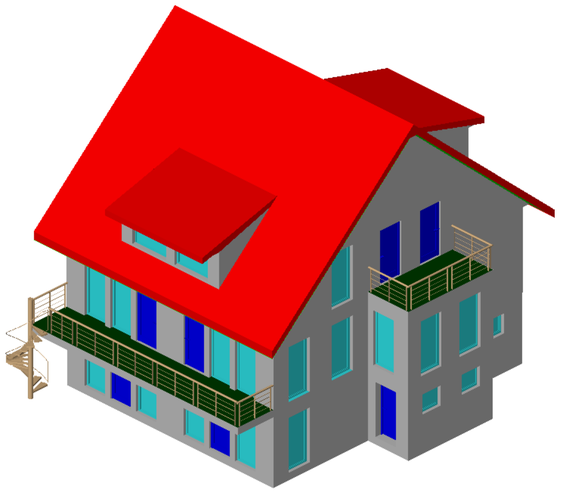
BIM to Geo conversion



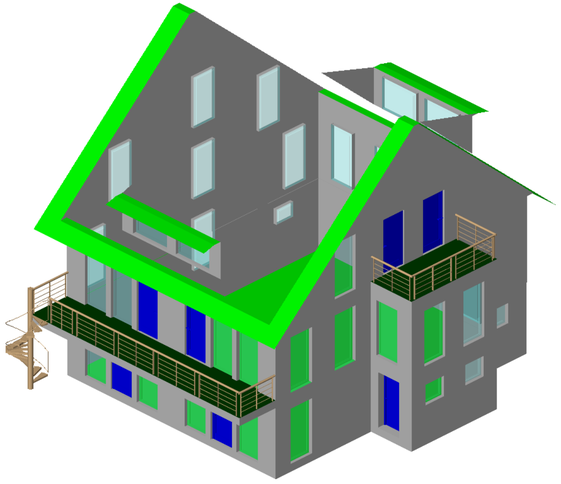
(a) input IFC



(b) IFC without roof

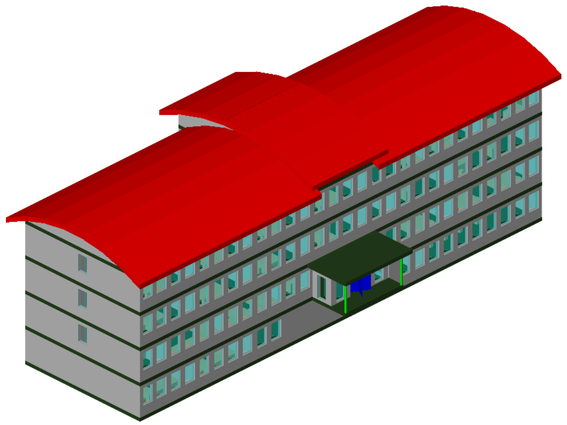


(c) output CityGML

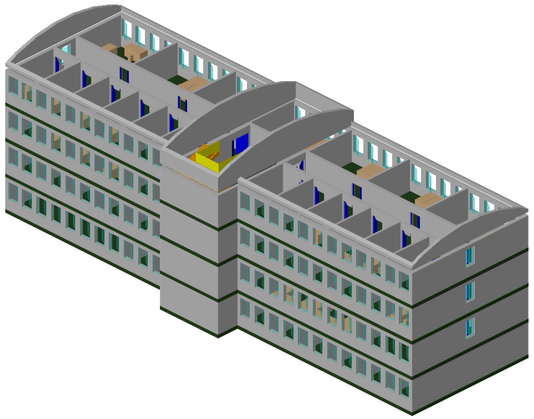


(d) CityGML without roof

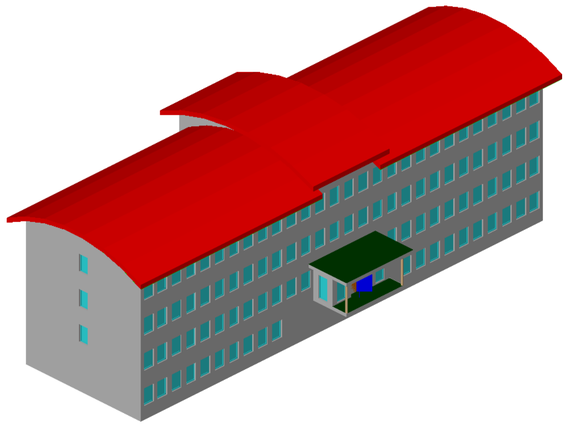
BIM to Geo conversion



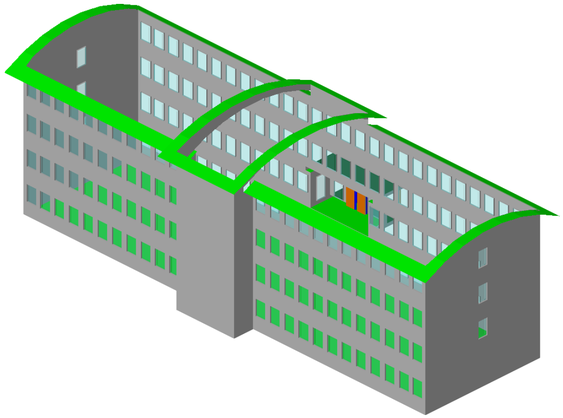
(a) input IFC



(b) IFC without roof

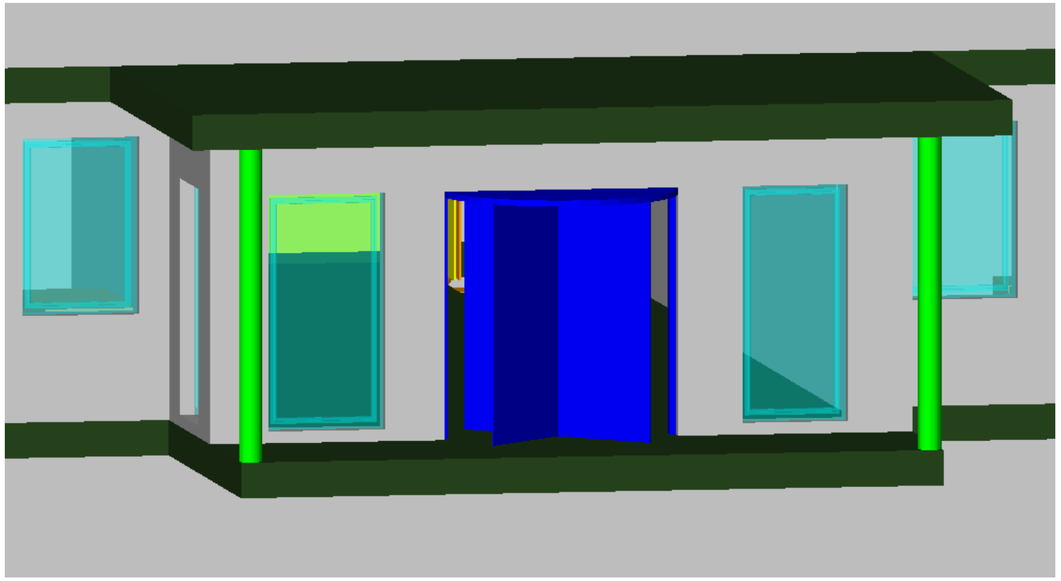


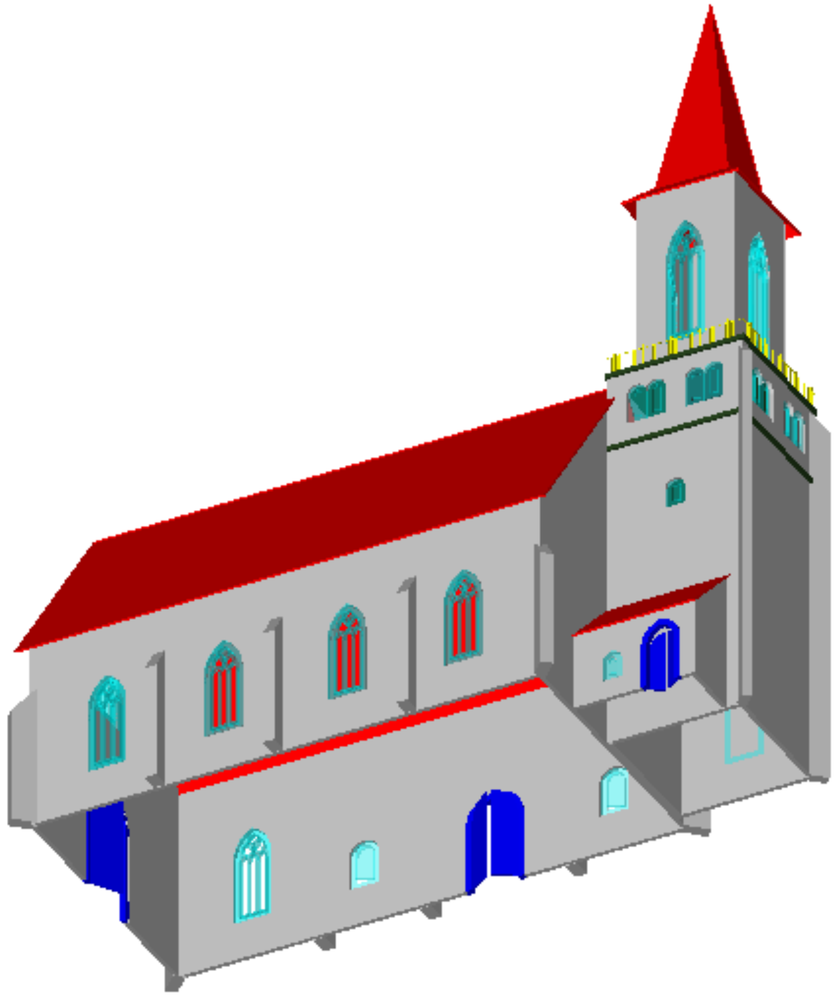
(c) output CityGML



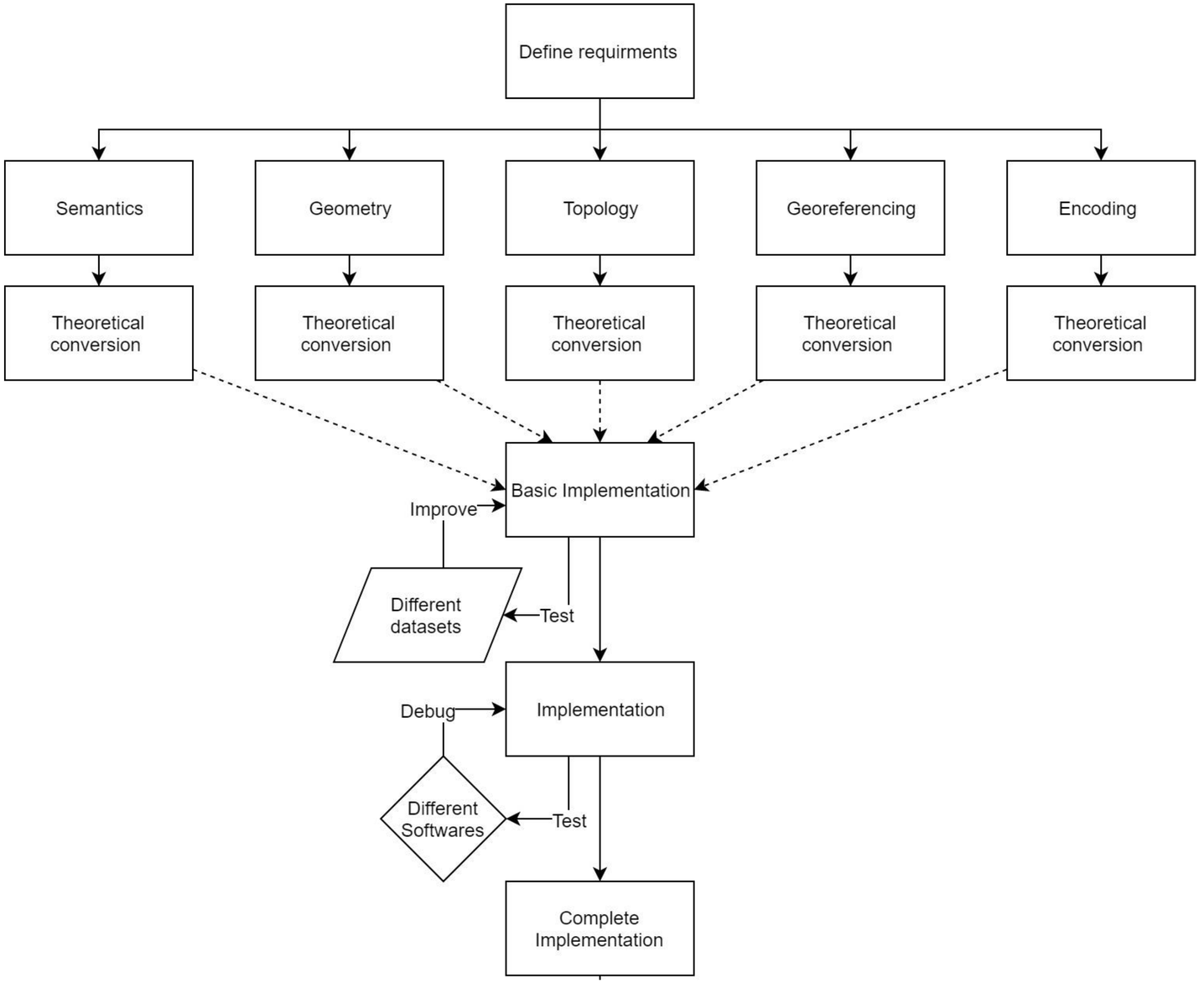
(d) CityGML without roof

BIM to Geo conversion

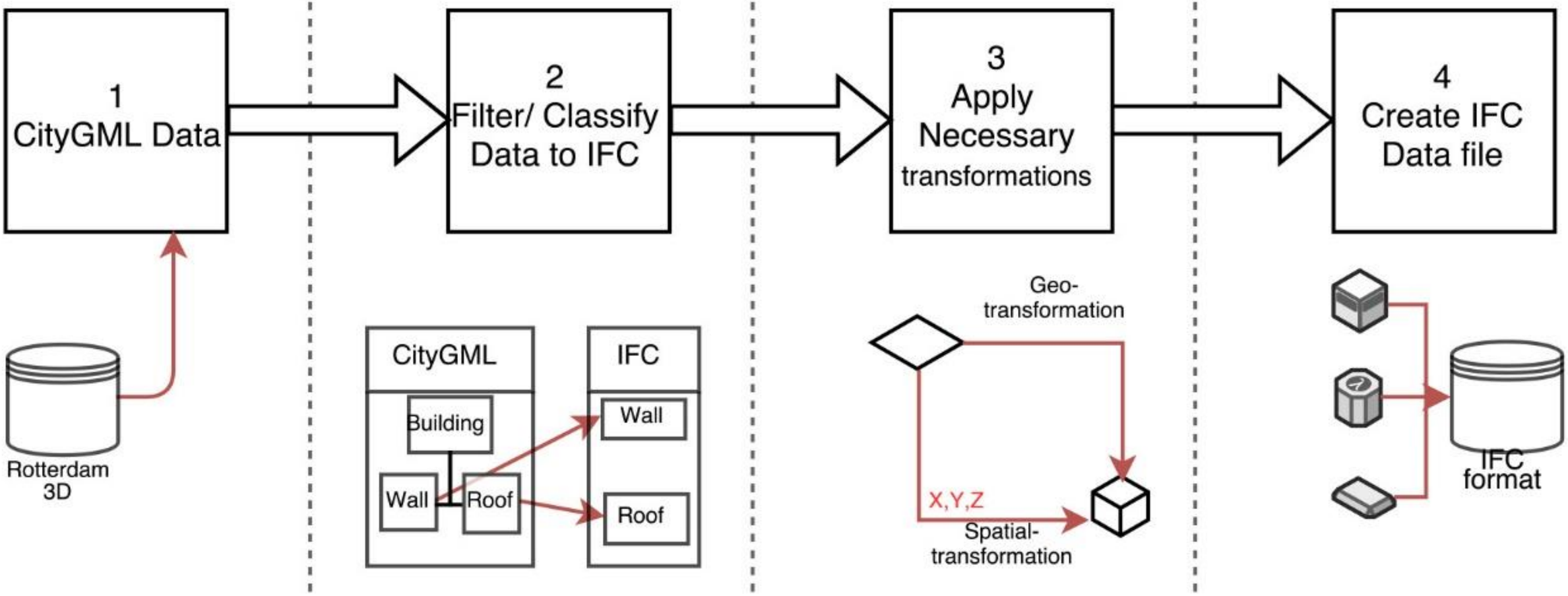




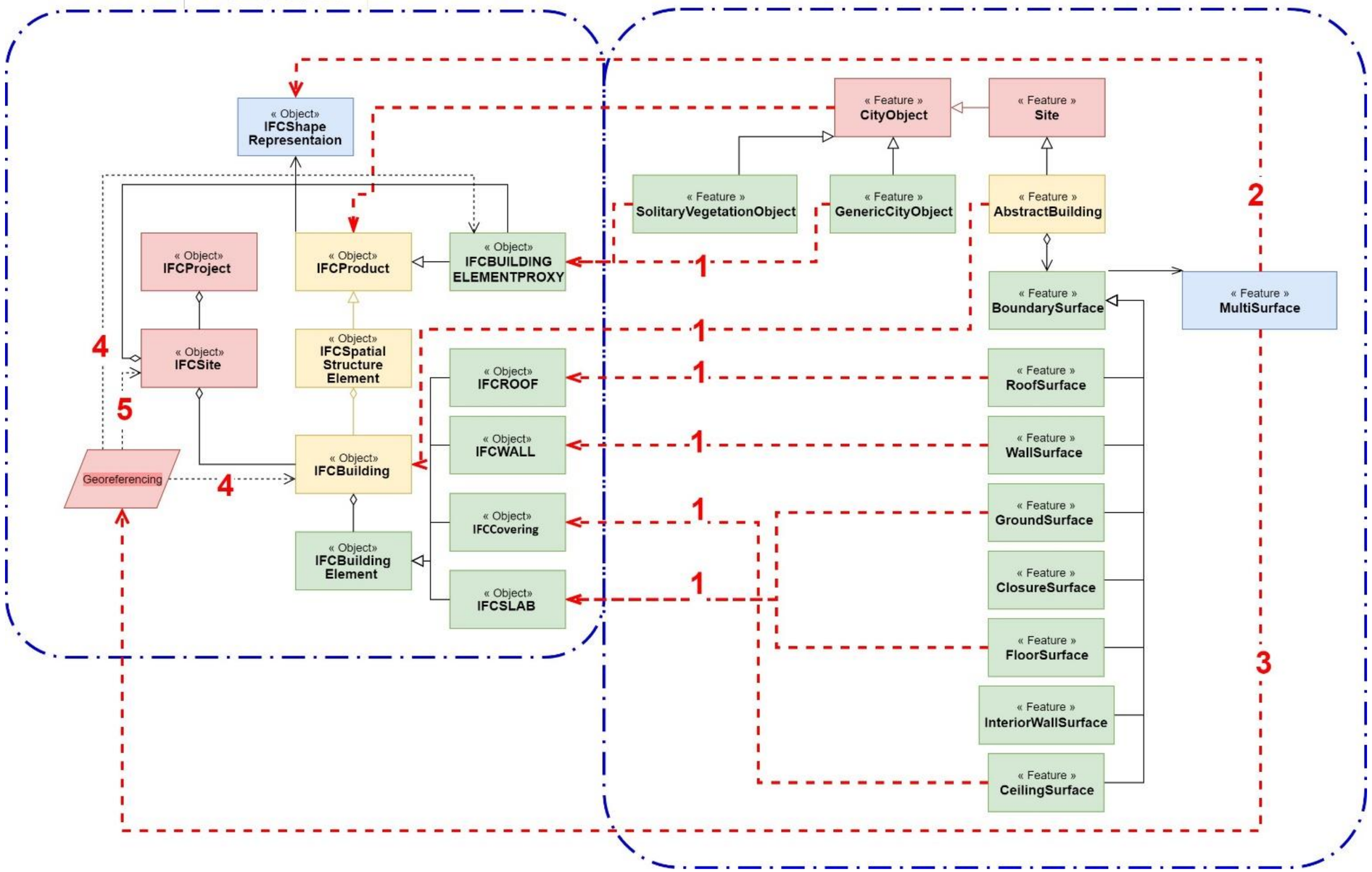
BIM to Geo conversion



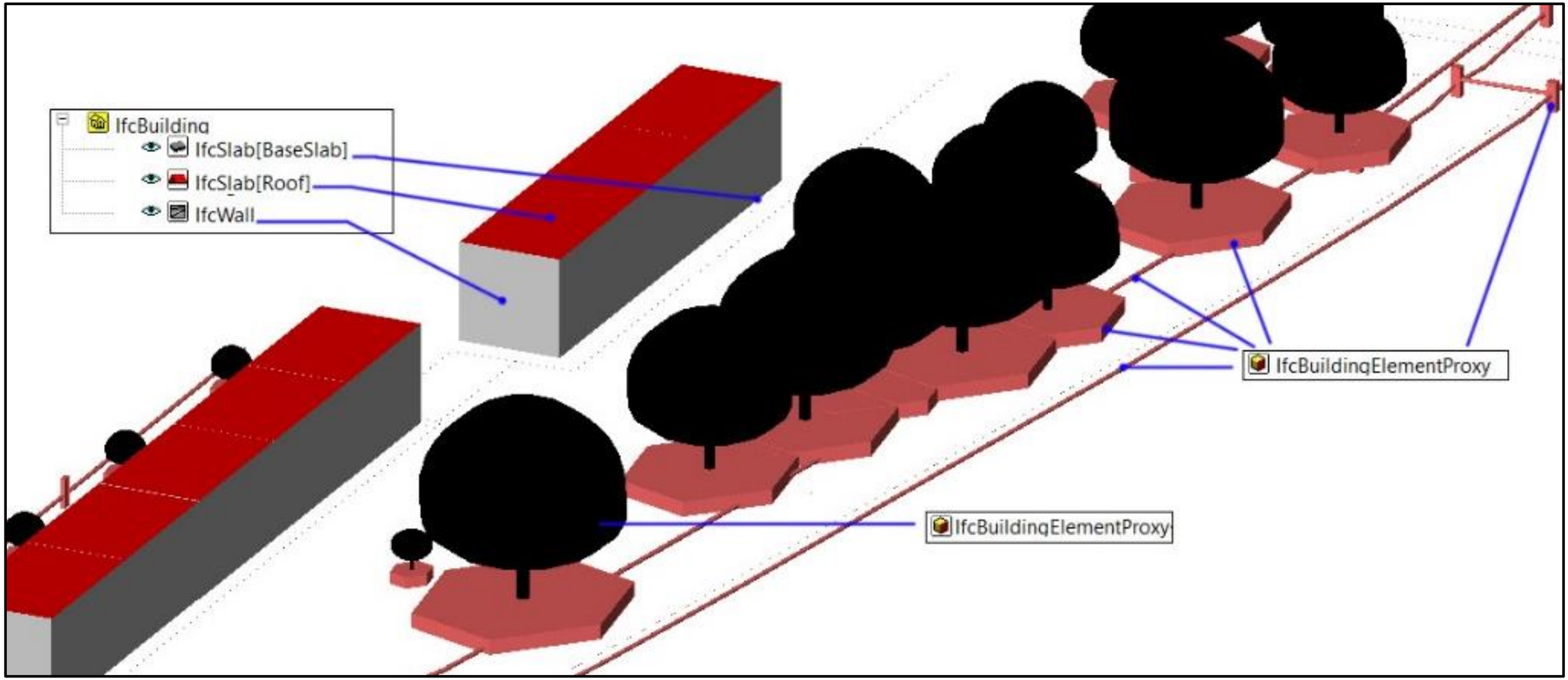
BIM to Geo conversion



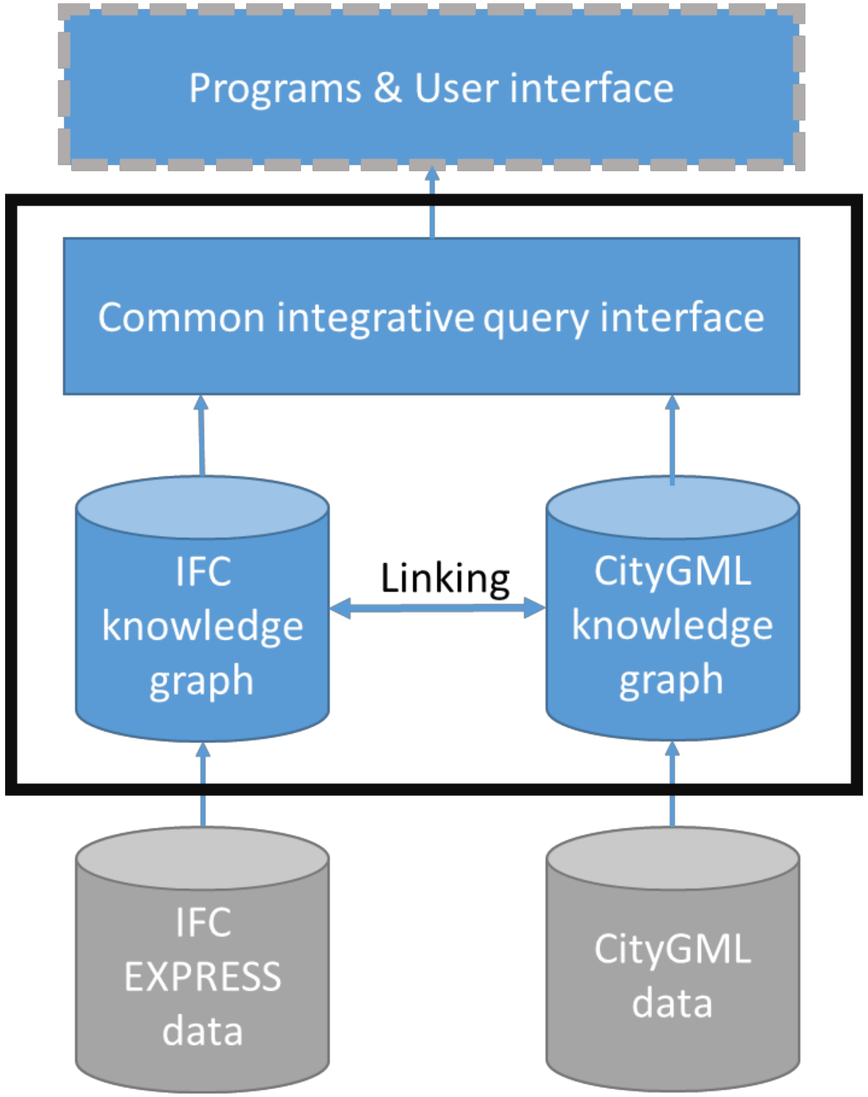
BIM to Geo conversion



BIM to Geo conversion

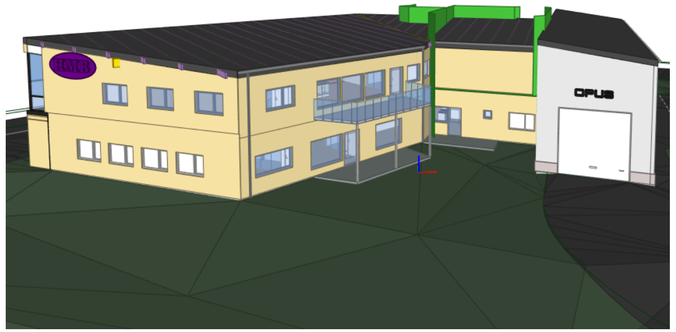


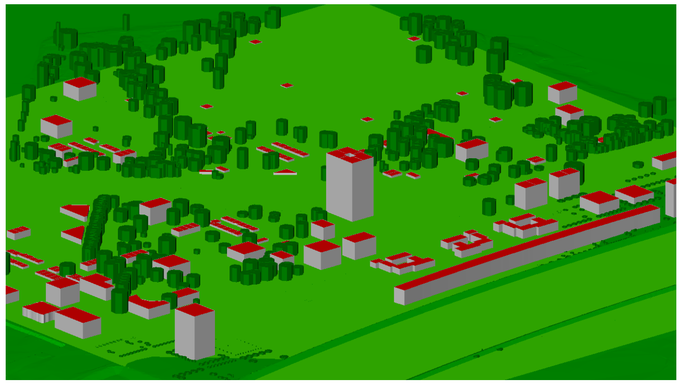
Geo and BIM to another model

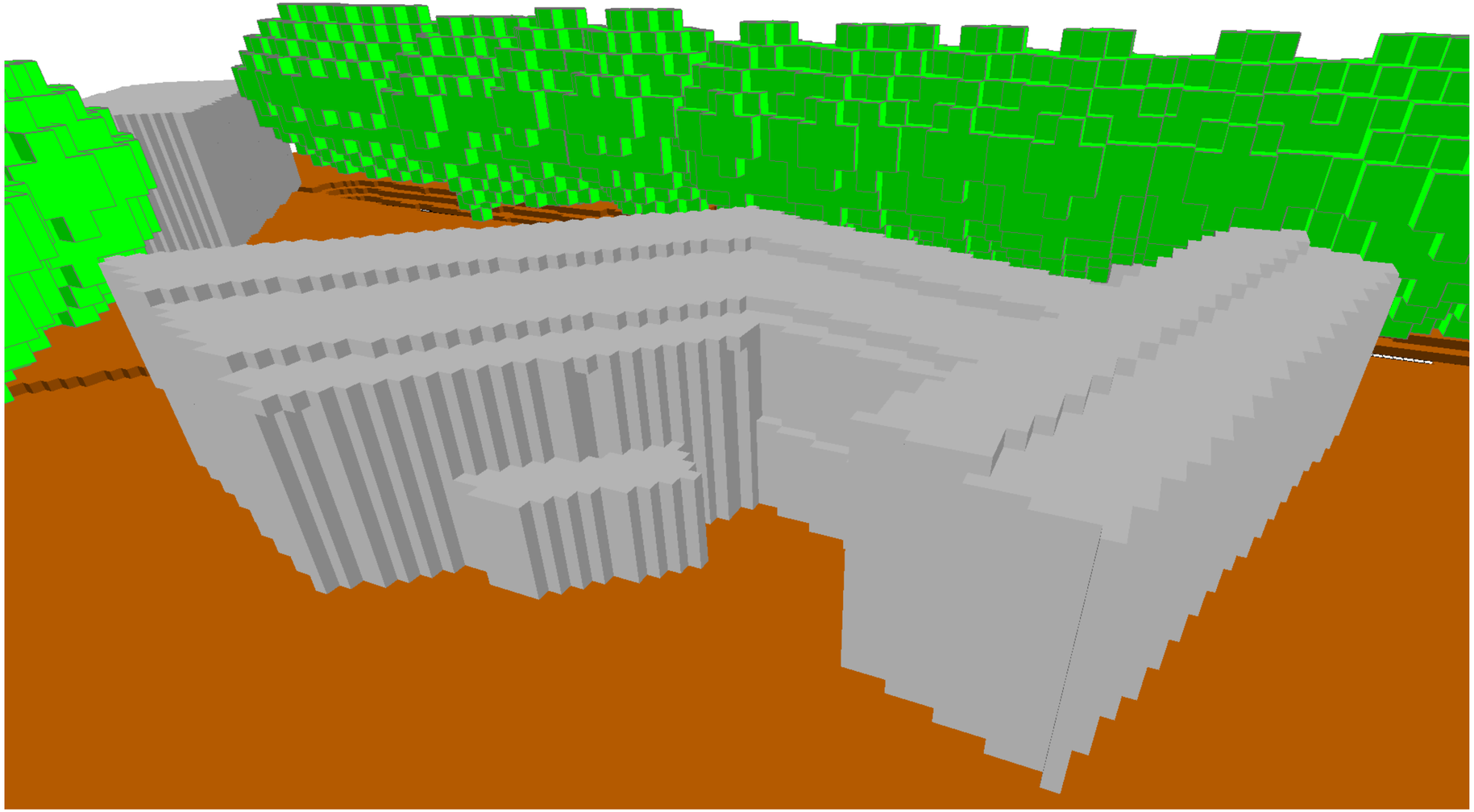


(Huang et al, 2020)

Geo and BIM to another model

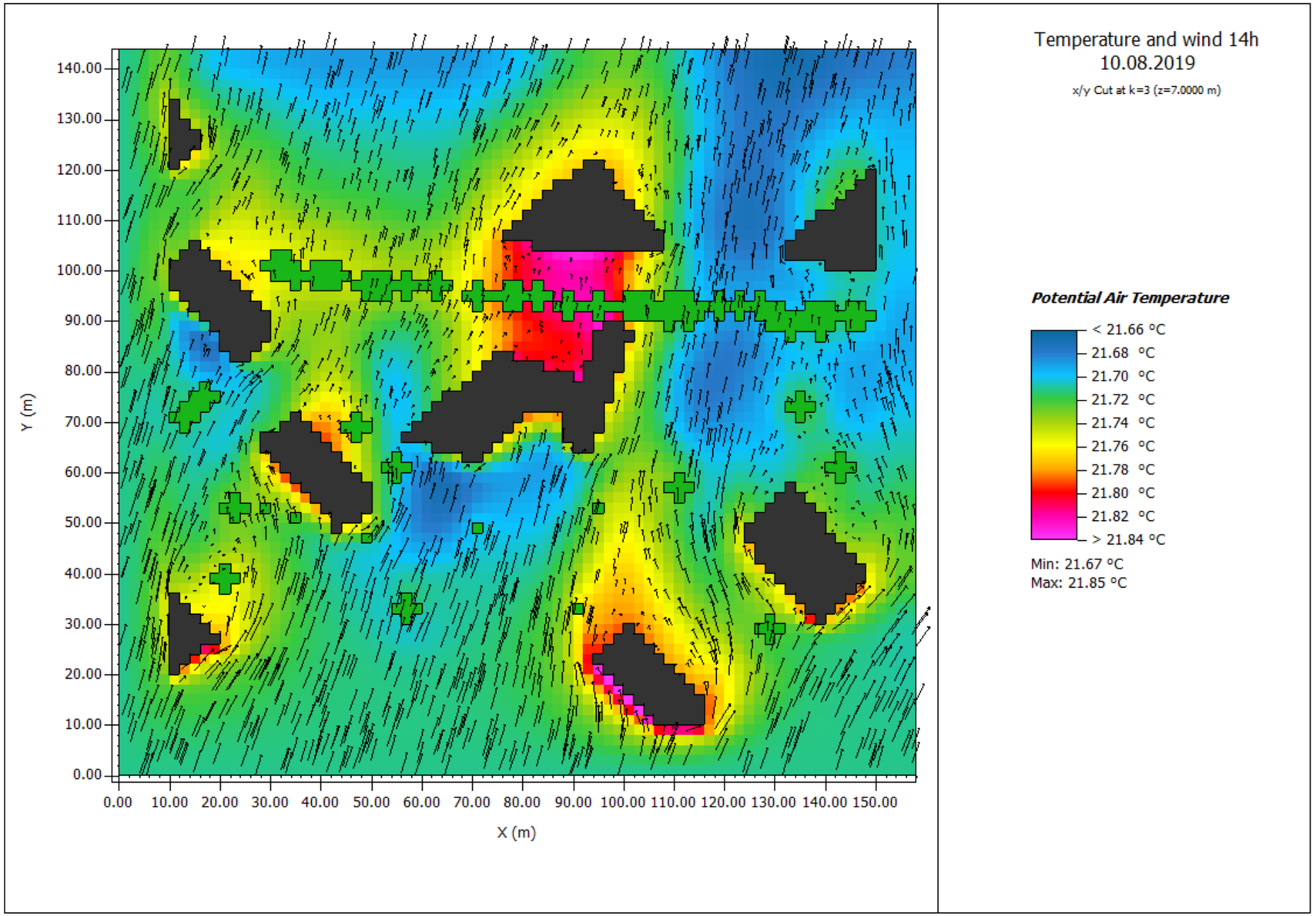






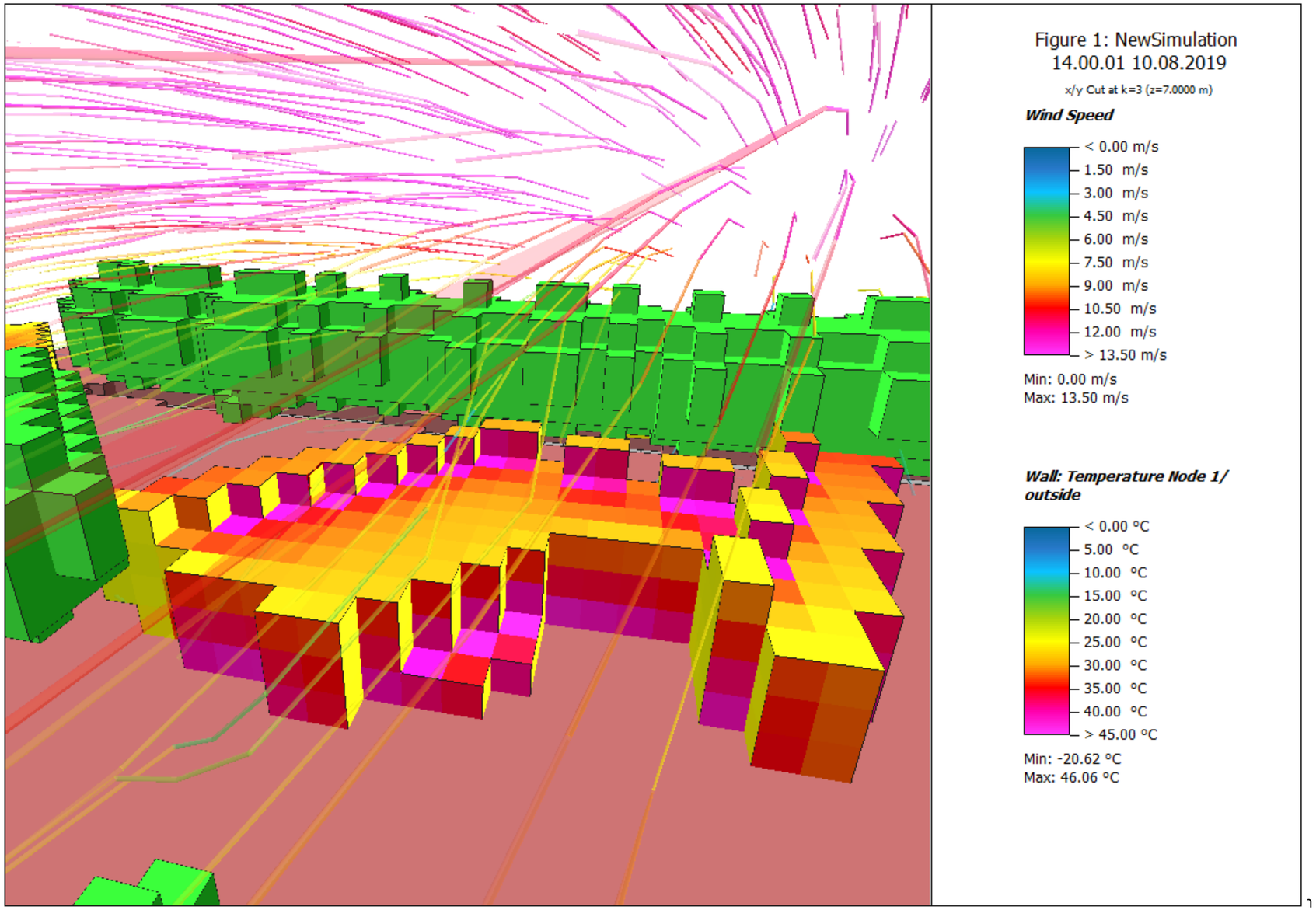
(van Heerden, 2021)

Geo and BIM to another model



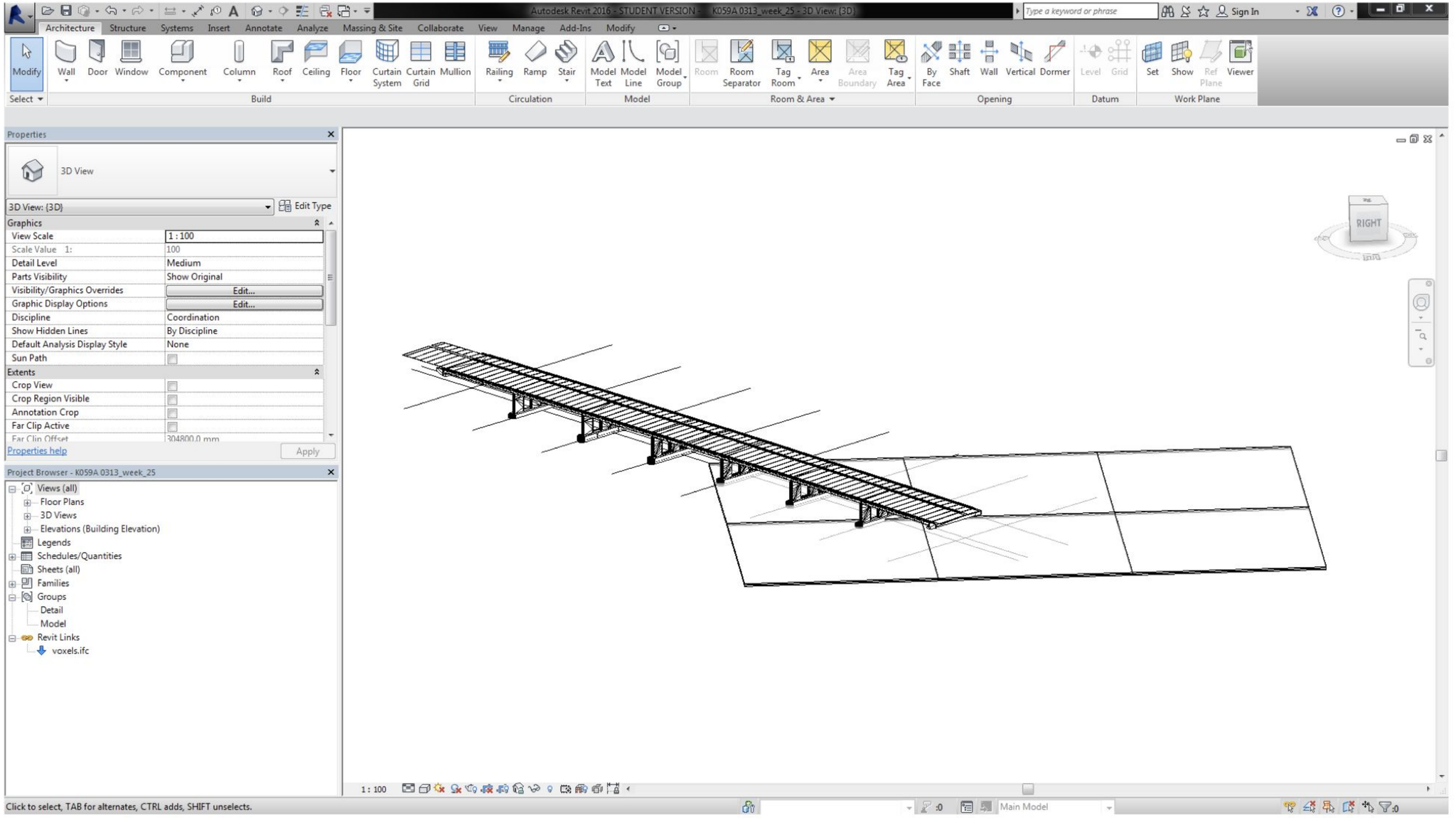
(van Heerden, 2021)

Geo and BIM to another model



ı Heerden, 2021)

Geo and BIM to another model



One last question?....

References

- Axelsson, P. (1999). Processing of laser scanner data—algorithms and applications. ISPRS Journal of Photogrammetry and Remote Sensing 54.2, pp. 138–147.

- Lafarge, F., X. Descombes, J. Zerubia, and M. Pierrot Deseilligny (2010). Structural approach for building reconstruction from a single DSM. IEEE Trans. on Pattern Analysis and Machine Intelligence 32.1, pp. 135–147.

- Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. Sjors Donkers, Hugo Ledoux, Junqiao Zhao and Jantien Stoter. Transactions in GIS 20 (4), 2016, pp. 547–569.

- Automatic conversion of CityGML to IFC. Nebras Salheb. Master's thesis, Delft University of Technology, October 2019.

References

- Huang, W., Olsson, P.-O., Kanters, J., and Harrie, L.: RECONCILING CITY MODELS WITH BIM IN KNOWLEDGE GRAPHS: A FEASIBILITY STUDY OF DATA INTEGRATION FOR SOLAR ENERGY SIMULATION, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., VI-4/W1-2020, 93–99, https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-93-2020, 2020.

- N.D. van Heerden. BIM and 3D City Models as Input for Microclimate Simulation, Master Thesis, Delft University of Technology, 2021.