Applications of 3D modelling of the built environment

GEO1004:
3D modelling of the built environment

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


Visibility analysis
Solar potential estimation
3D cadastre
Energy demand estimation
Infrastructure planning
Emergency response
Shadow estimation
Indoor navigation
Noise propagation
Utility management
TUDelft


dB
60
45
30


W/m2
600
500
400
300
200
100
0

Other applications

- Visualisation (eg for gaming, tourism, navigation, etc)
- Energy demand estimation (and potential for retrofitting)
- Computational fluid dynamics (eg for wind speeds, air quality, effects on buildings, etc)
- Shadow casting (eg for building permits, visibility analysis, improving energy demand/ solar potential calculations, etc)

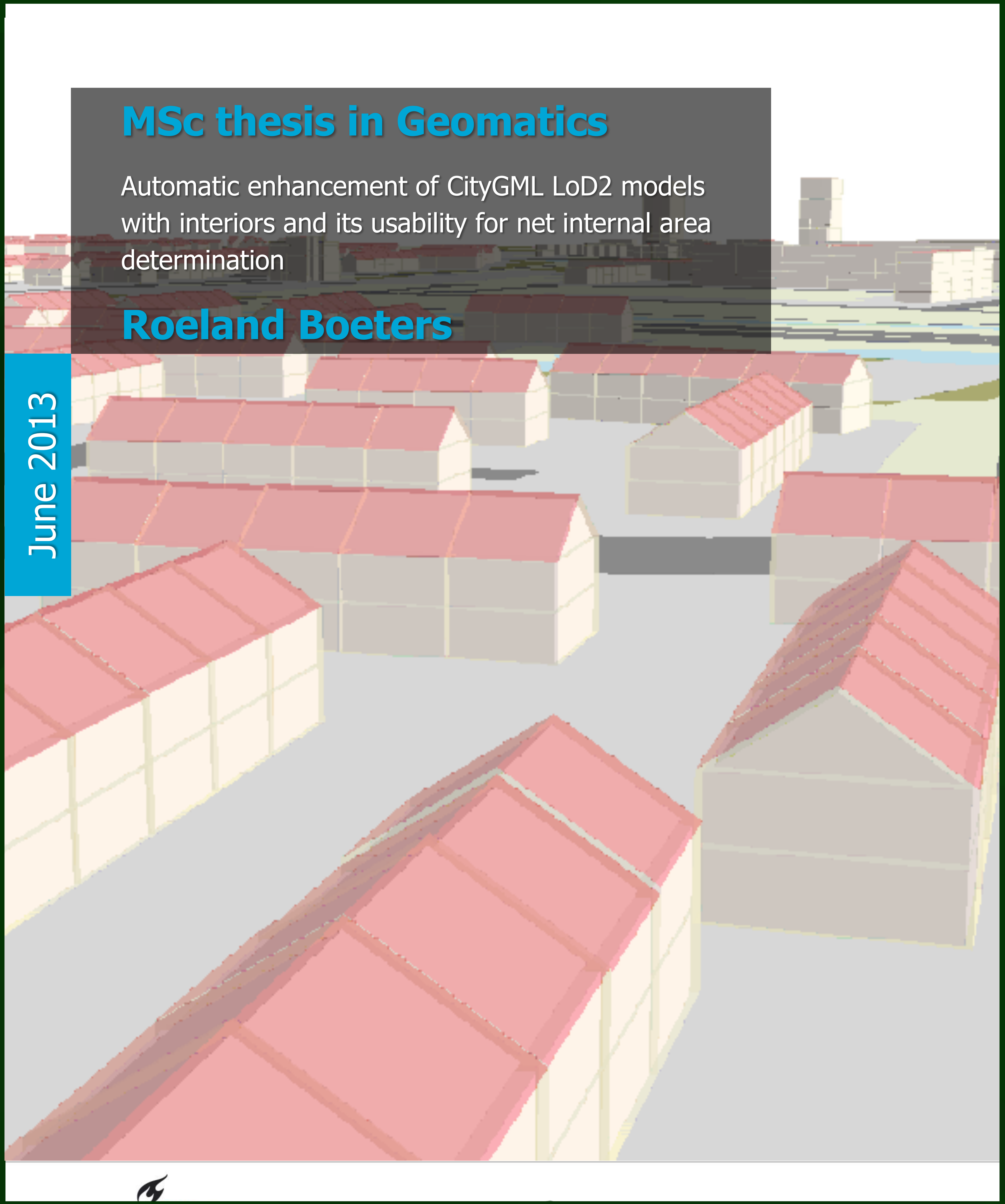
Some MSc Geomatics theses

- Motivation: create (rough) indoor geometry from widely available outdoor geometry

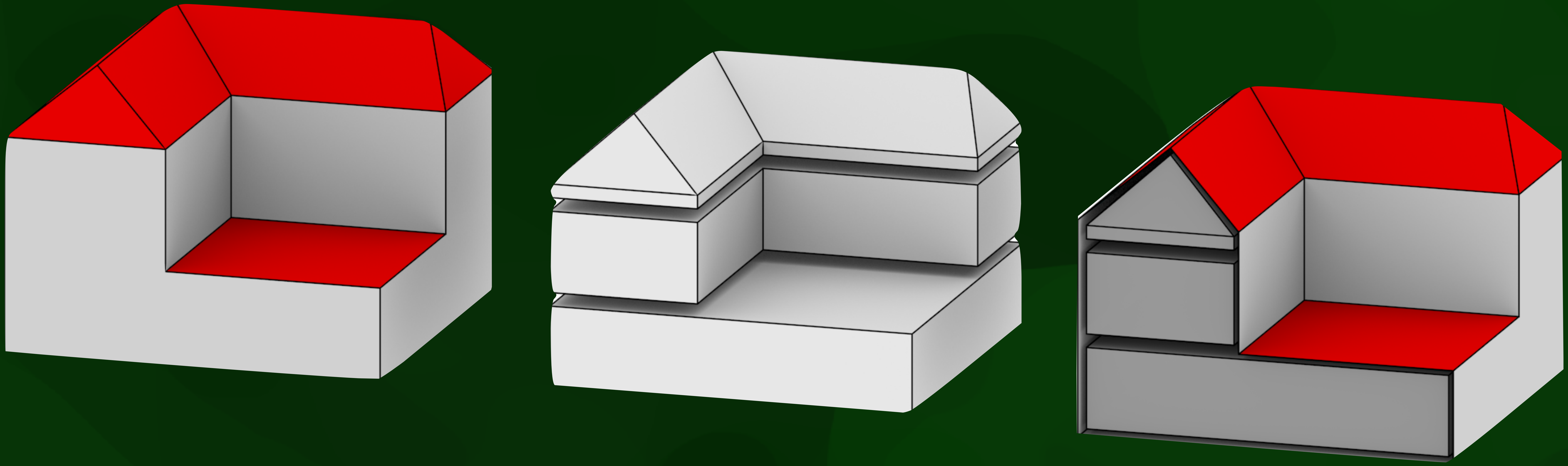
- Definition of a CityGML LOD2 with interiors (LOD2+)

- Compute interior geometry from exterior geometry + number of storeys

- Compute net internal area

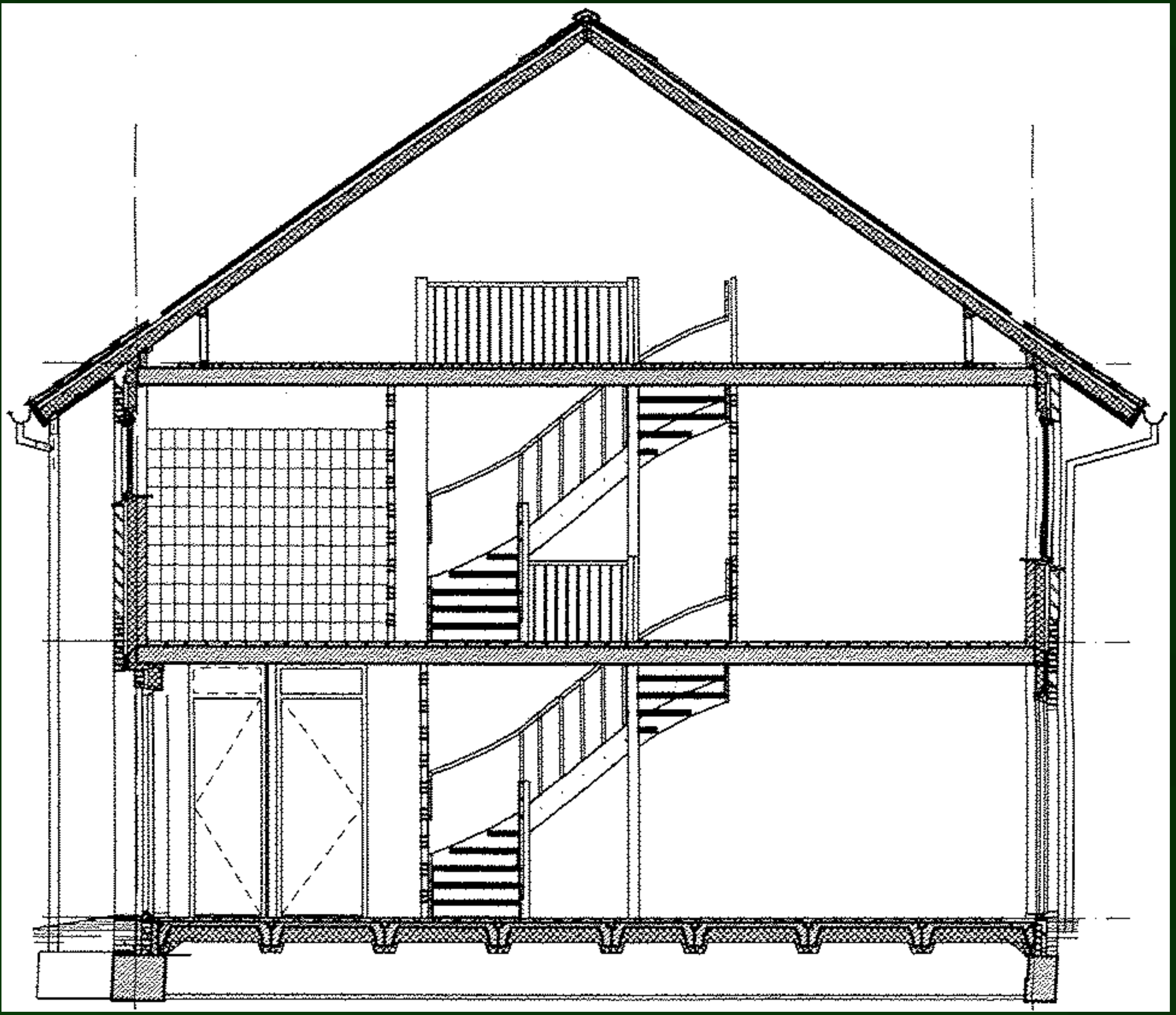


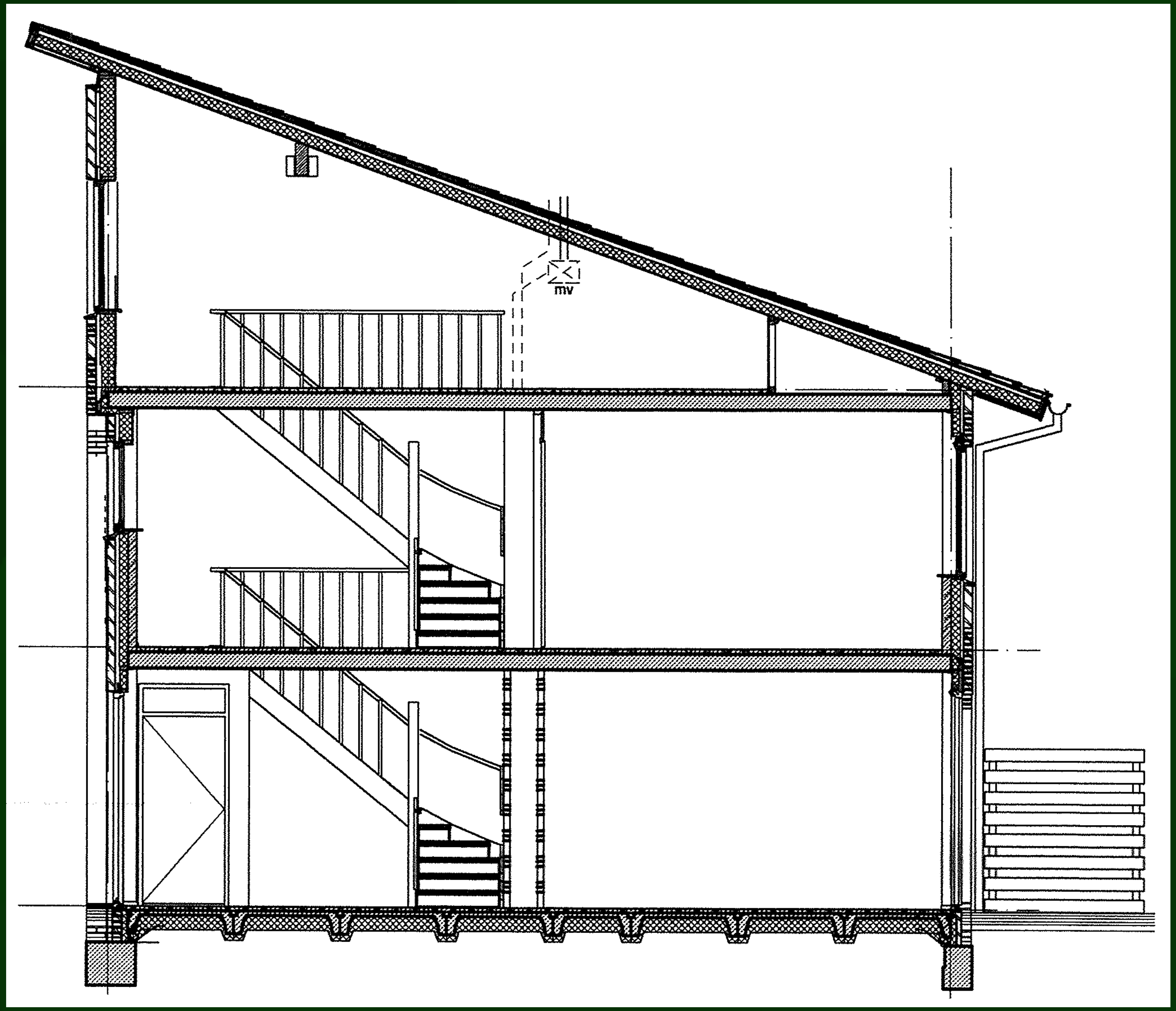
LOD2+



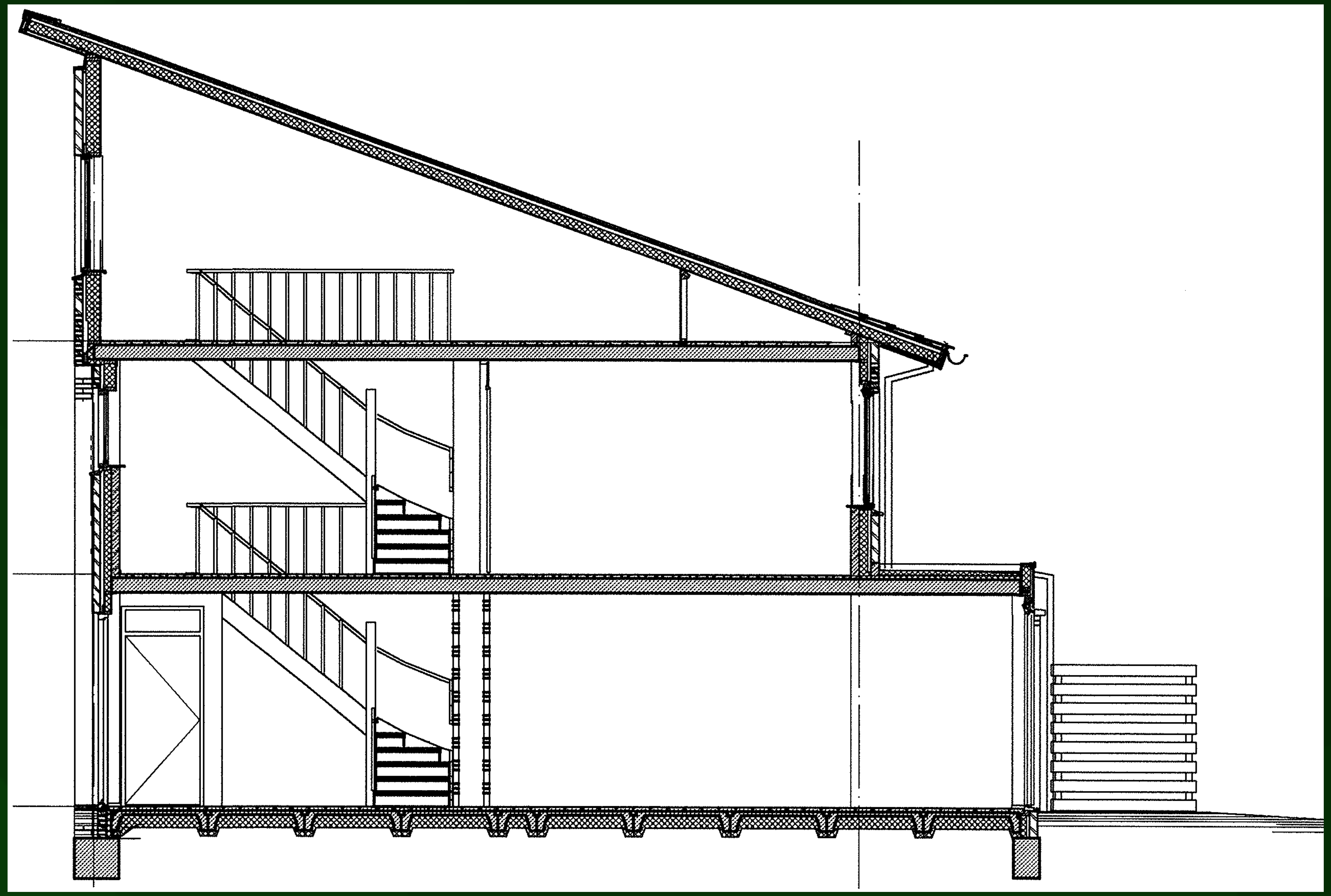
Exterior in LOD2	Interior in LOD2+
Buildings bodies are prisms Simple roof shapes Thematically classified boundary surfaces No openings in the exterior geometry	Storeys within building bodies are prisms Attic storey shapes corresponding to roof shapes Thematically classified boundary surfaces No openings in the indoor geometry

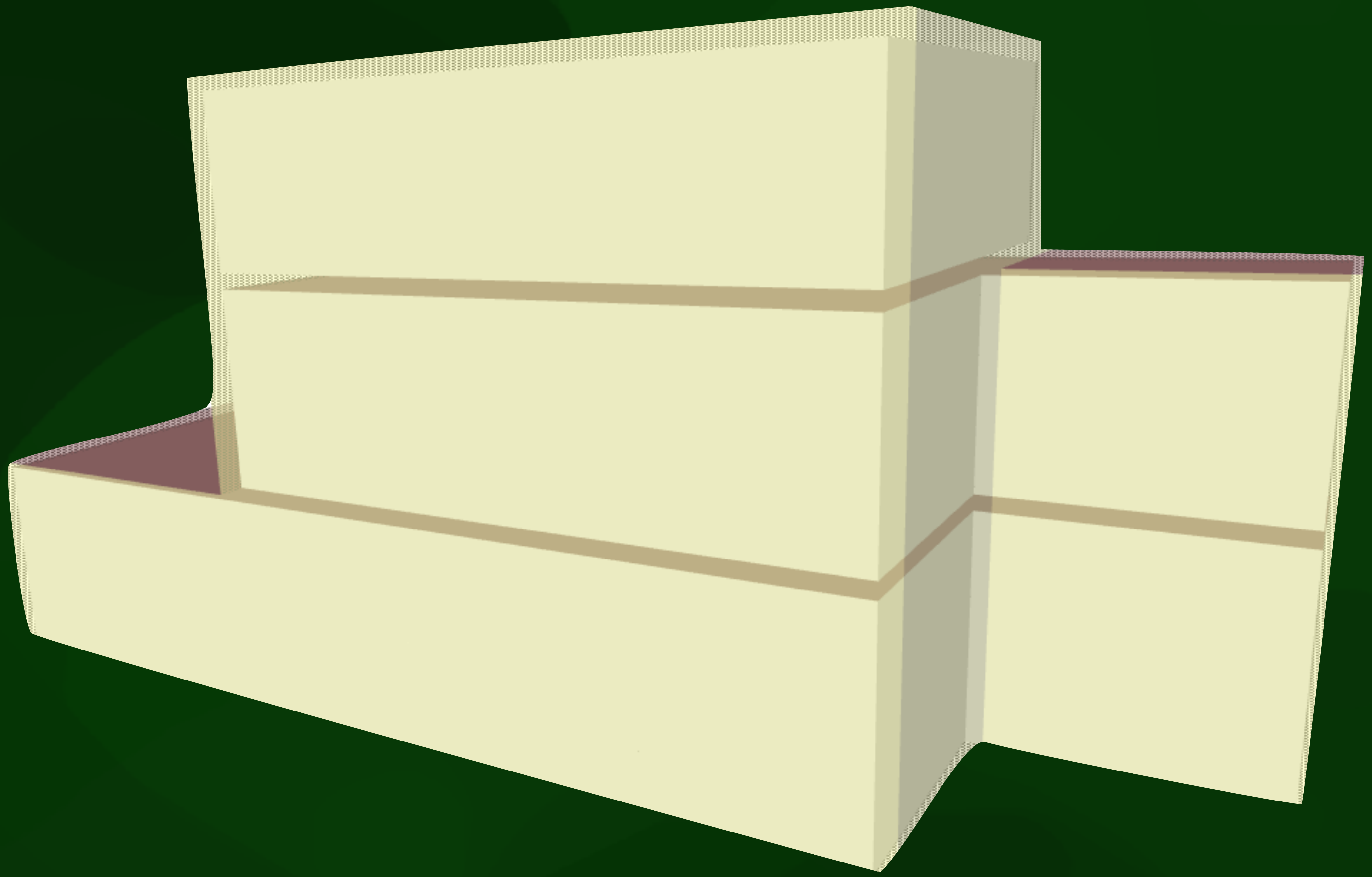
Indication of storeys





Indication of storeys

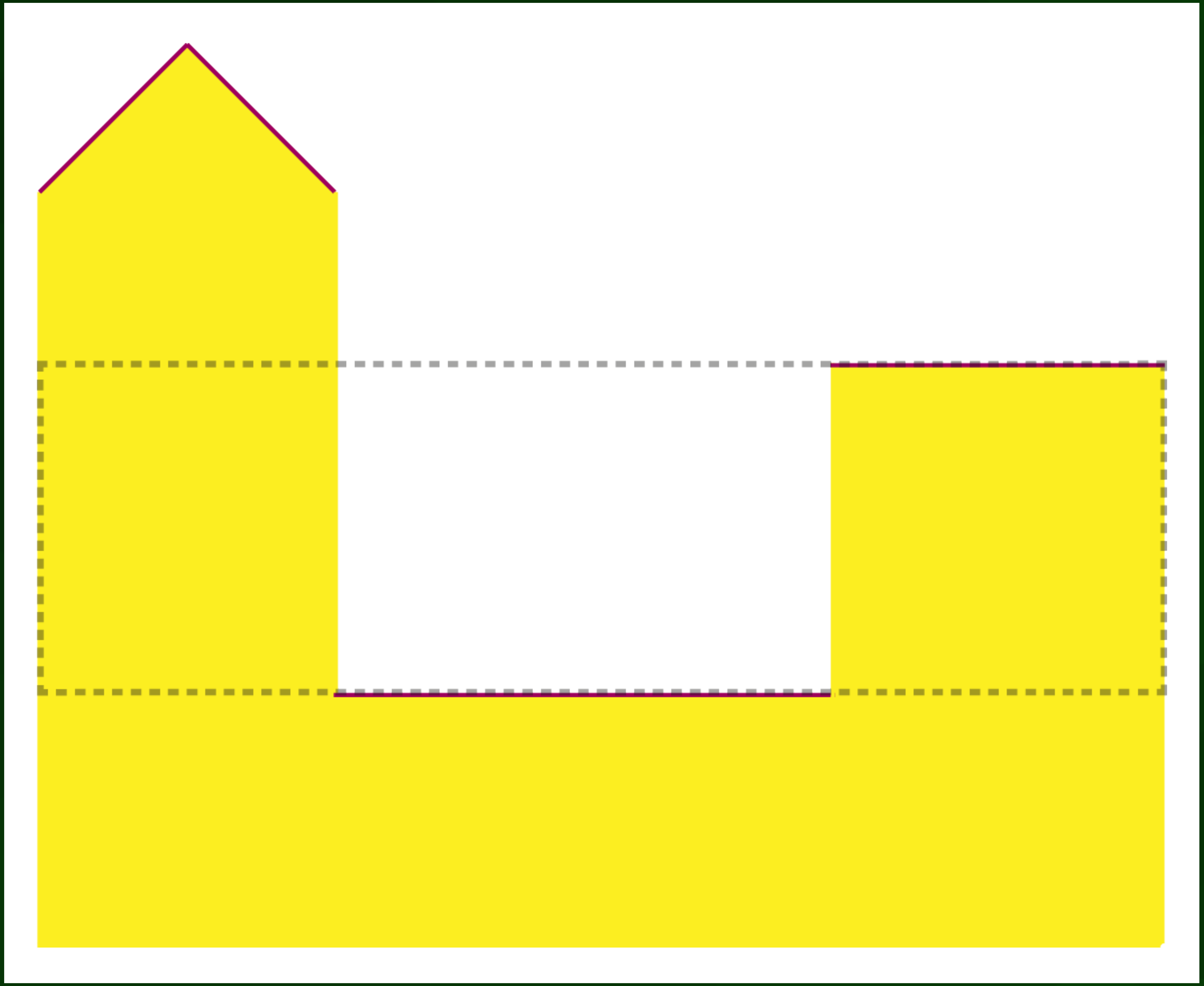




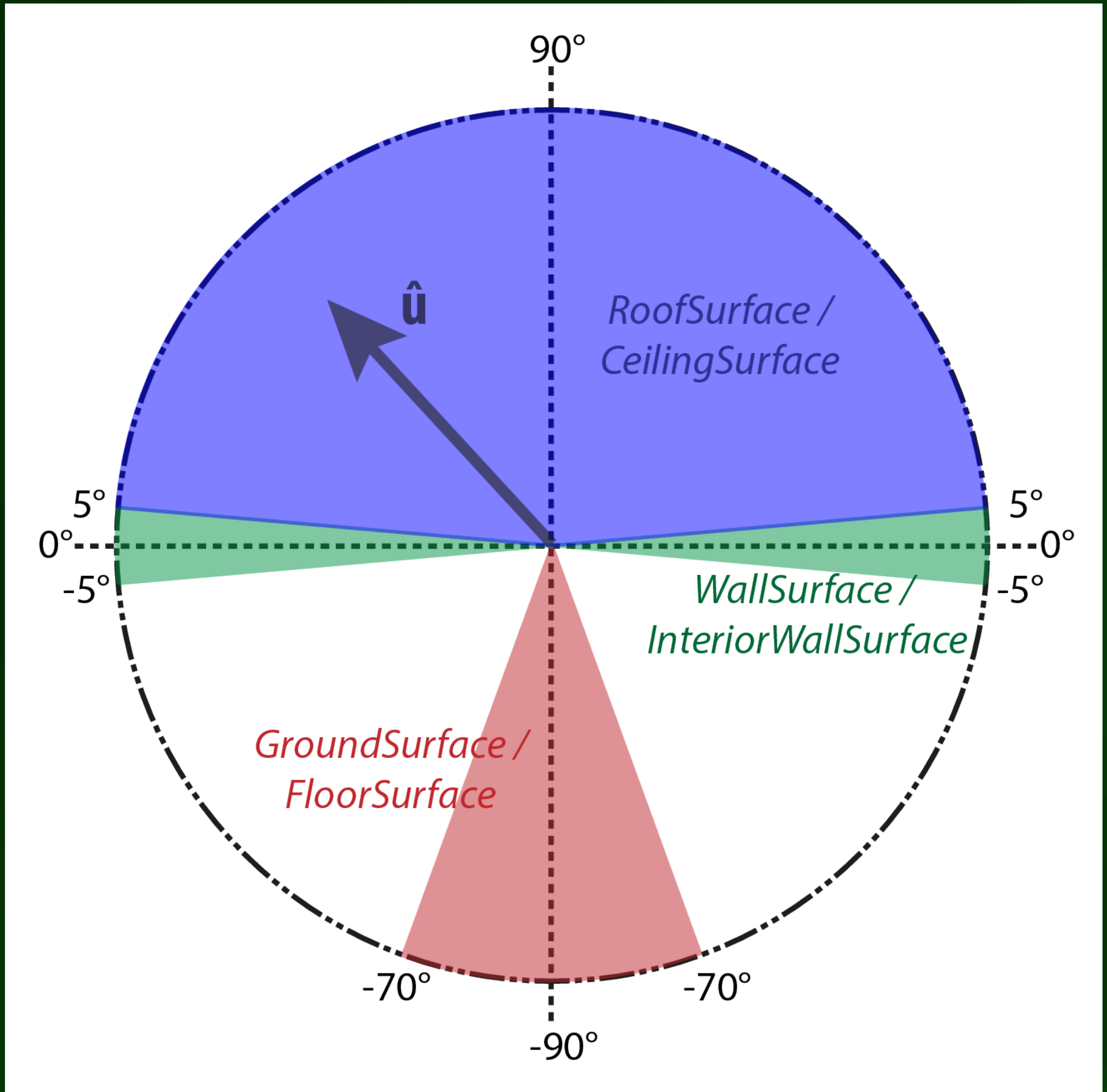
Wall thickness

Type	year y	storeys x	t_{ext} [cm]	t_{shared} [cm]
Non-stacked	$y < 1970$	$x \leq 2$	27	11
		$x \geq 3$	27	12
	$1970 \leq y \leq 1985$	$x = 2$	27	10
		$x = 3$	28	12
		$x = 4$	27	9
	$y > 1985$	$x = 2$	28	13
		$x = 3$	30	12
		$x = 4$	25	12
Stacked	$y < 1970$	$x \leq 5$	29	12
		$5 < x \leq 10$	38	11
		$x > 10$	25	9
	$1970 \leq y \leq 1985$	$x \leq 5$	28	11
		$5 < x \leq 10$	26	11
		$x > 10$	29	12
	$y > 1985$	$x \leq 5$	30	12
		$5 < x \leq 10$	38	13
		$x > 10$	35	15
Other types	$y < 1970$	$x = 1$	14	14
		$x \geq 2$	31	11
	$1970 \leq y \leq 1985$	$x = 1$	14	14
		$x \geq 2$	30	10
	$y > 1985$	$x = 1$	14	14
		$x \geq 2$	36	13

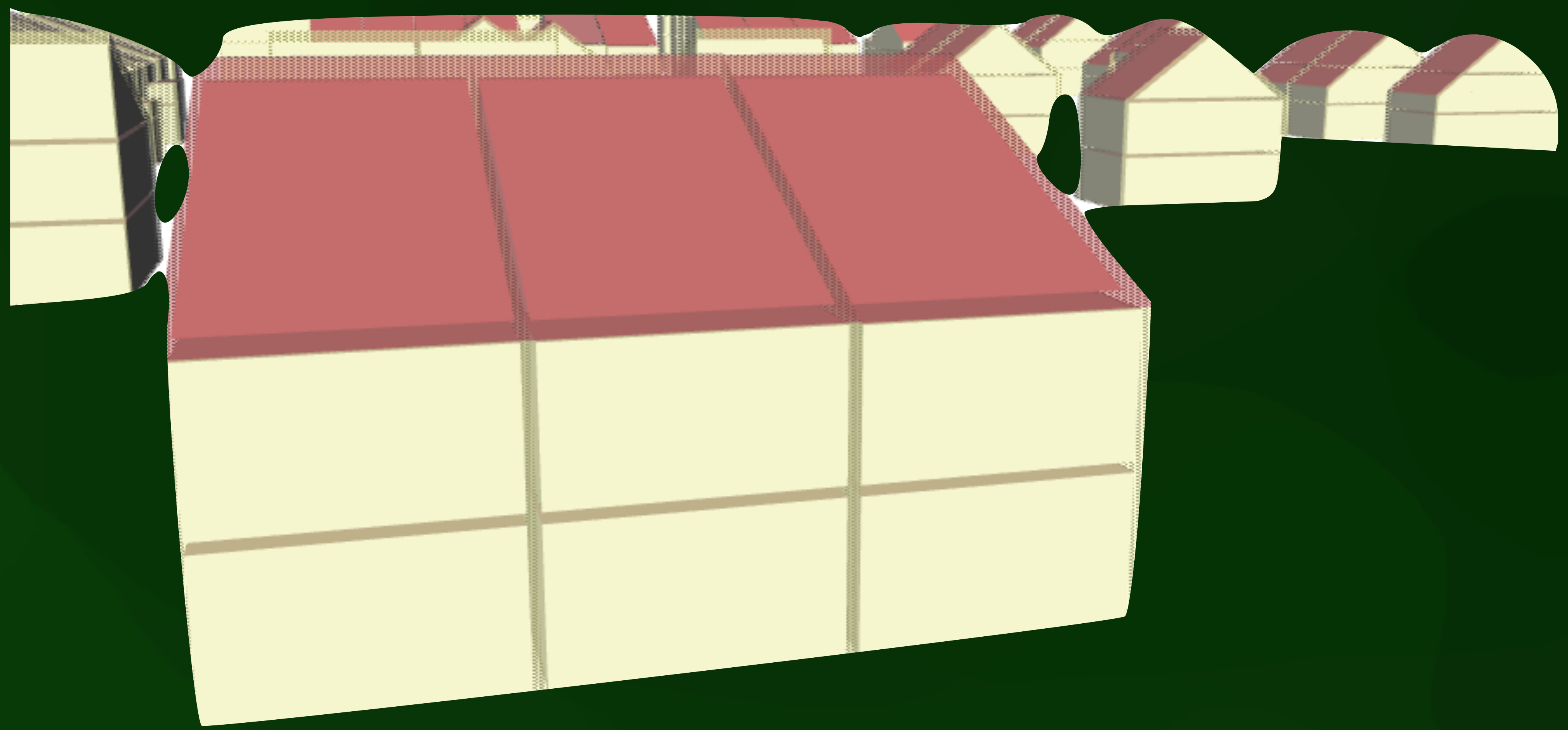
Boolean set intersection

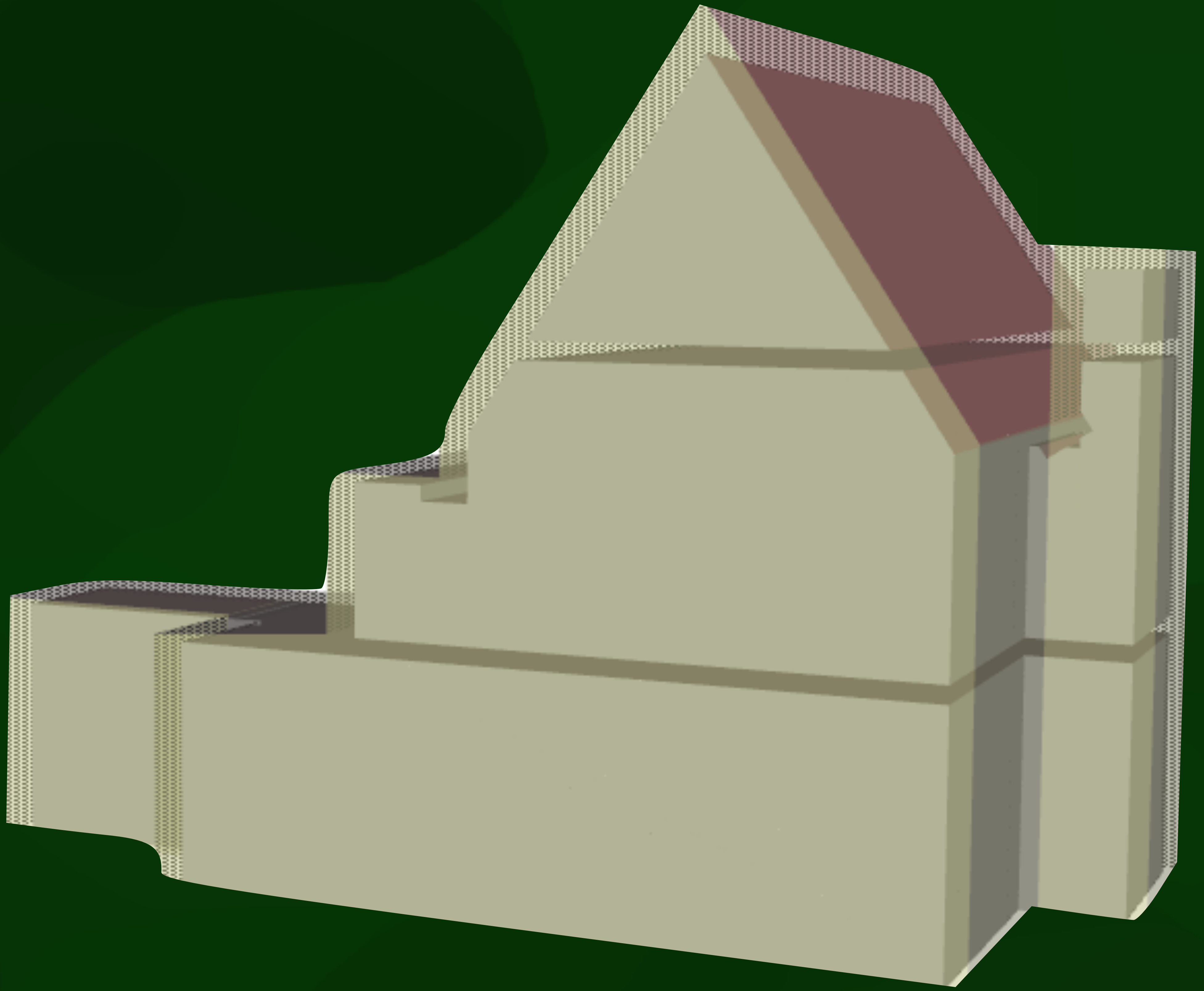


Classifying surfaces

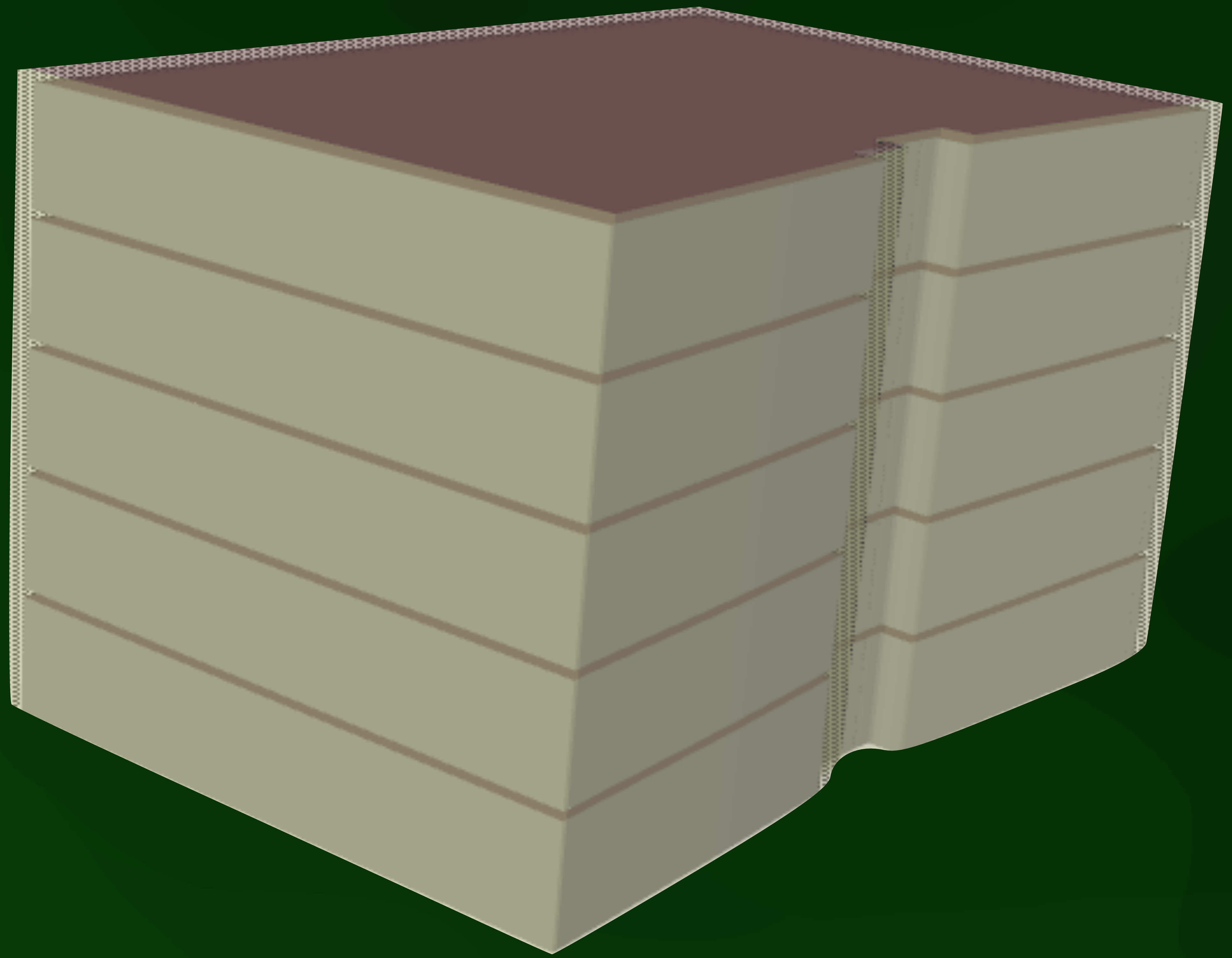


Results

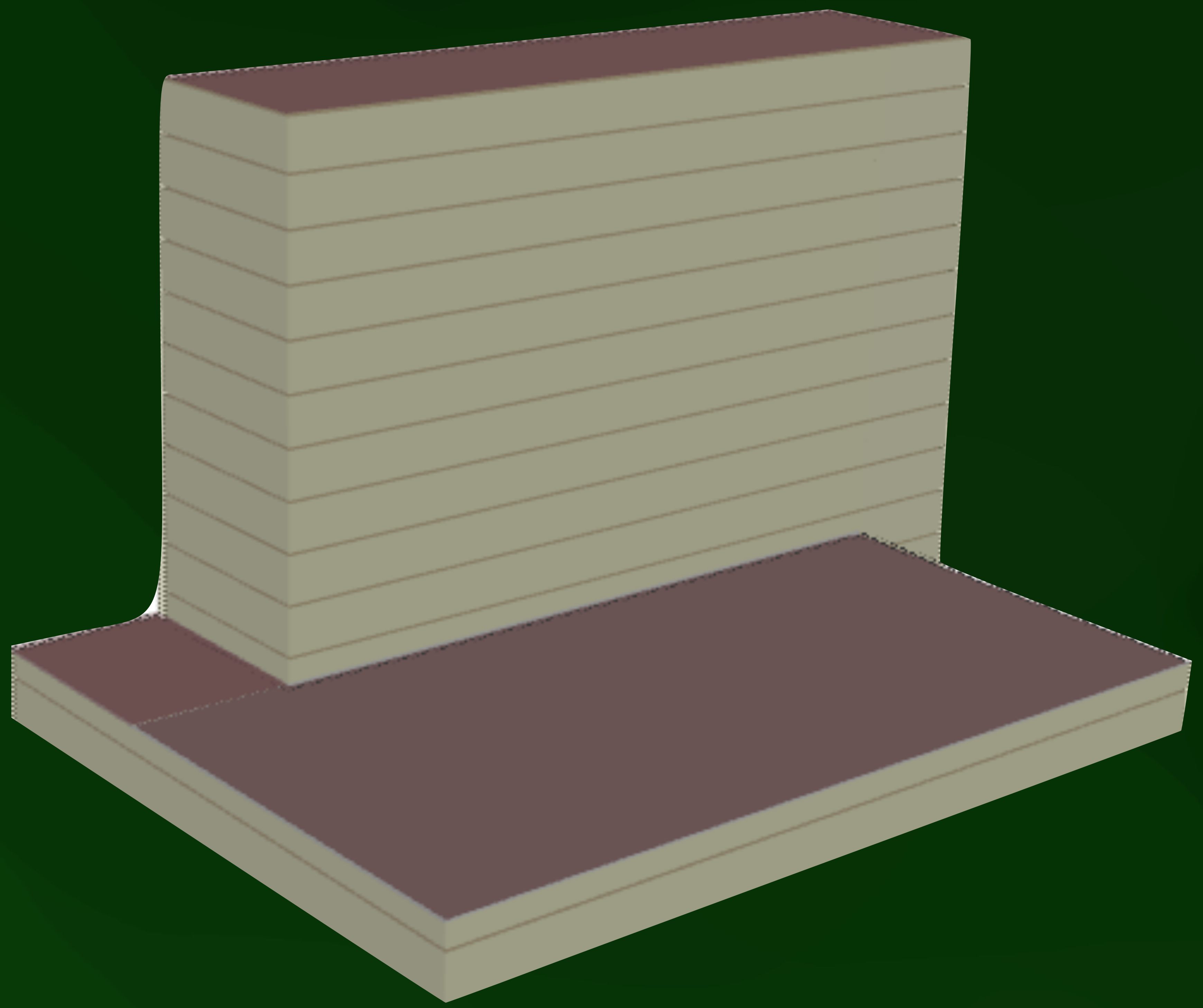




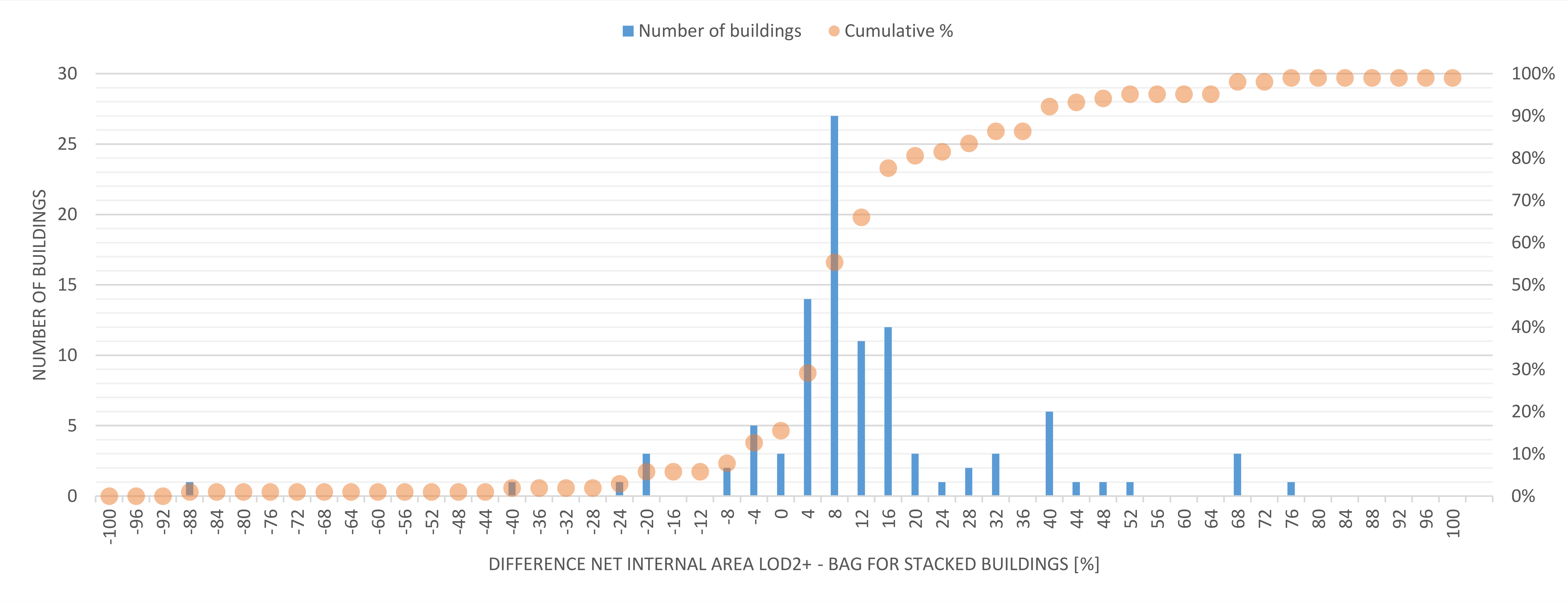
Results



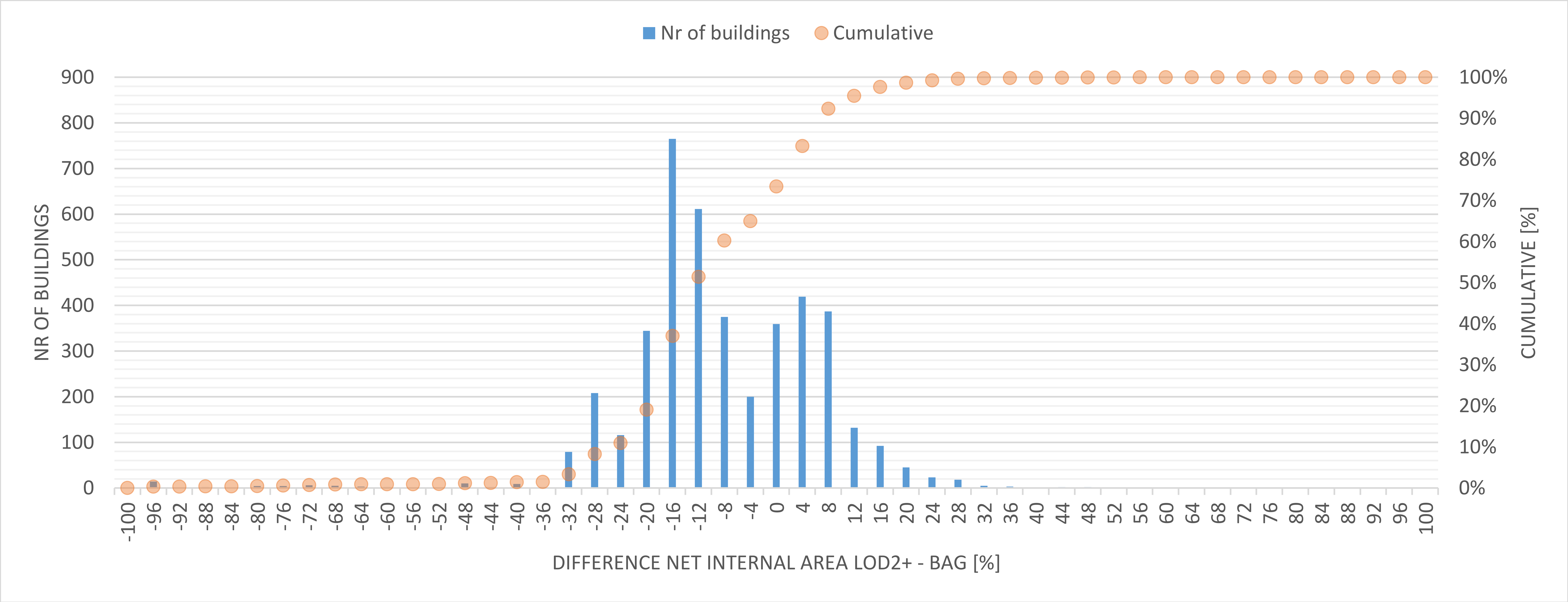
Results



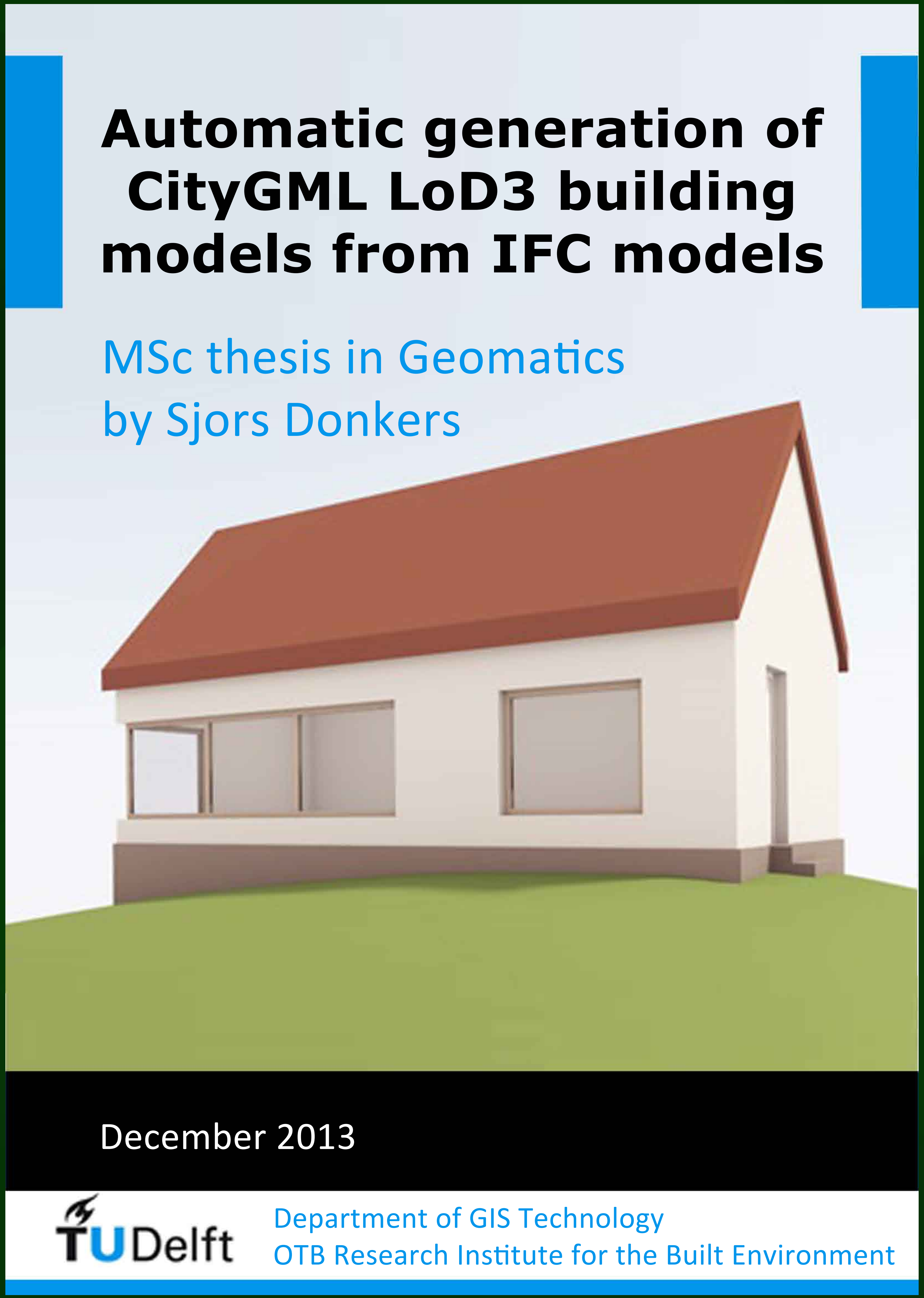
Net internal area (stacked)



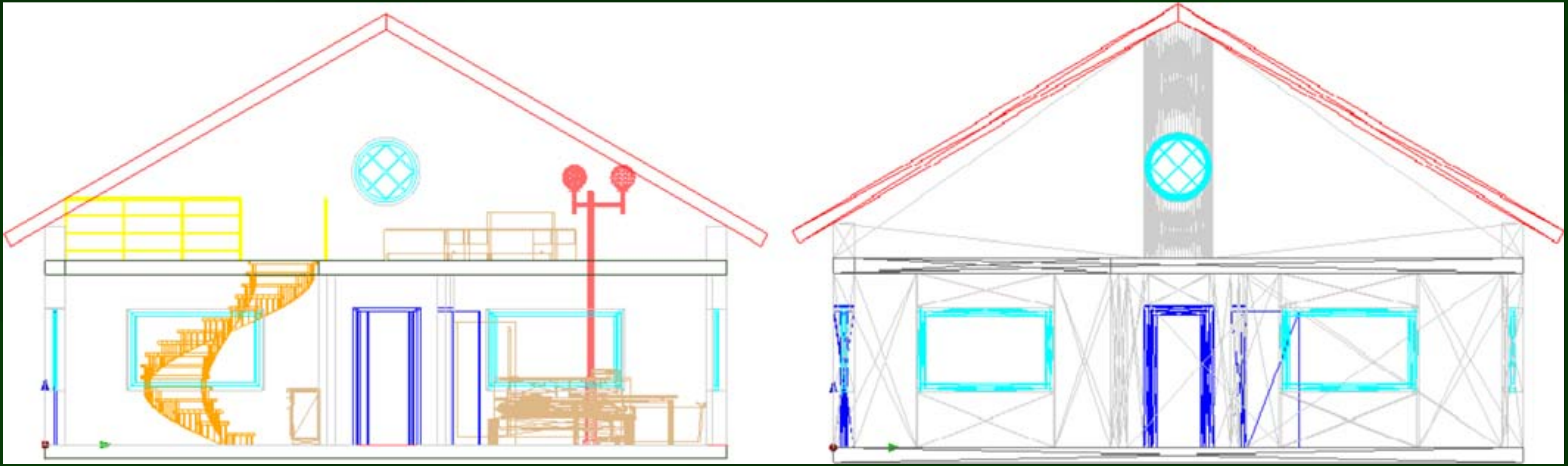
Net internal area (non-stacked)



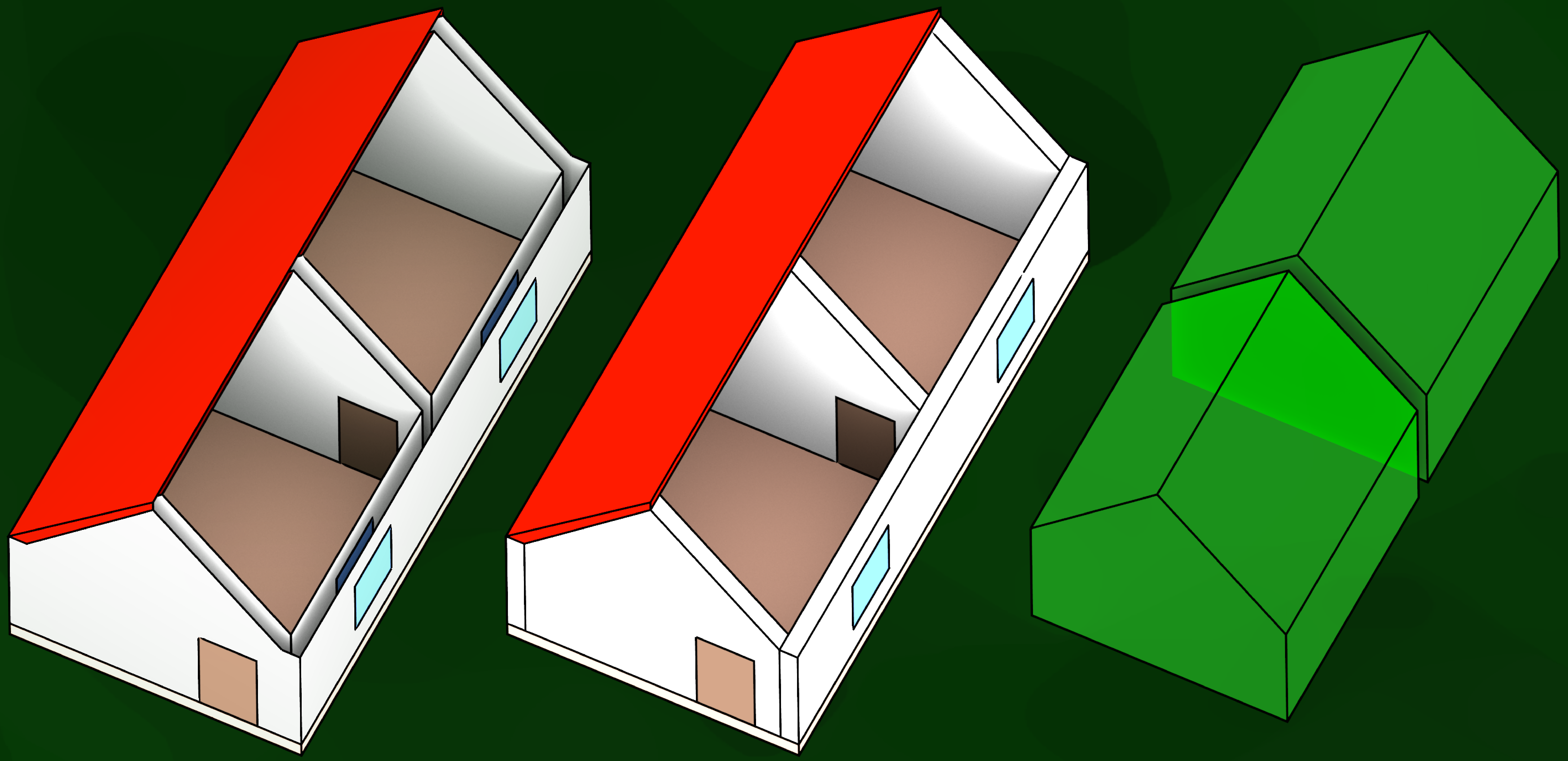
- Motivation: update 3D city models from designed BIM models (including potentially interiors)
- Fill gaps using Minkowski sum to increase size of elements
- Merge elements using Boolean set union
- Reclassify surfaces



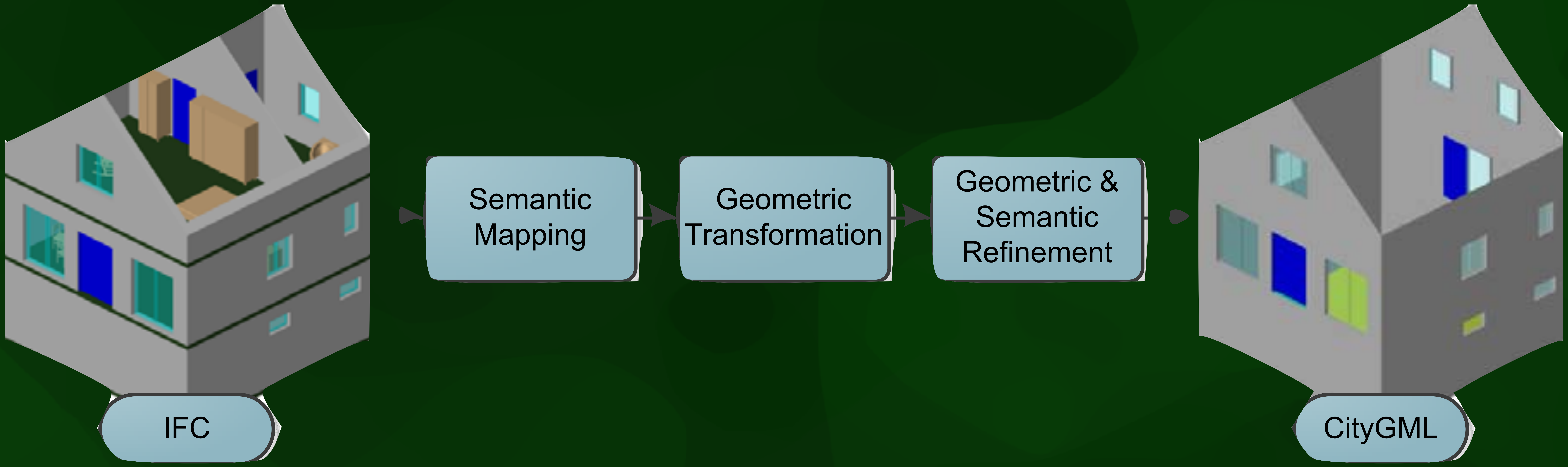
Goal



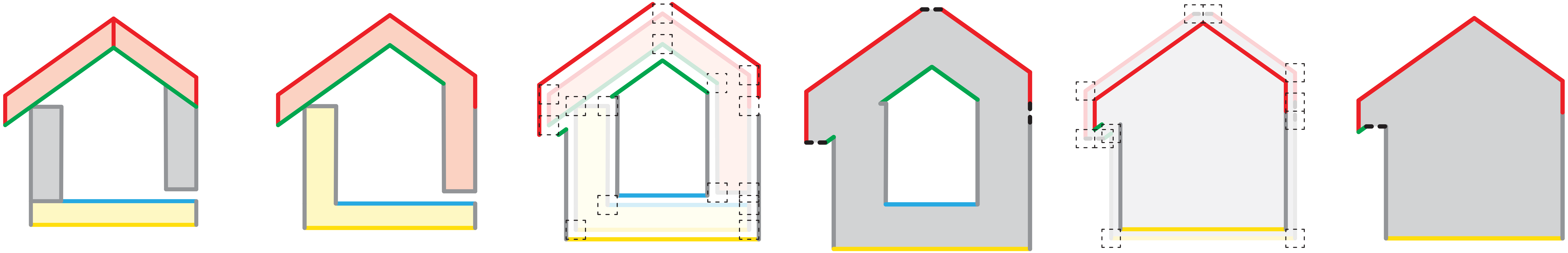
3DCM vs BIM



Methodology (semantics)

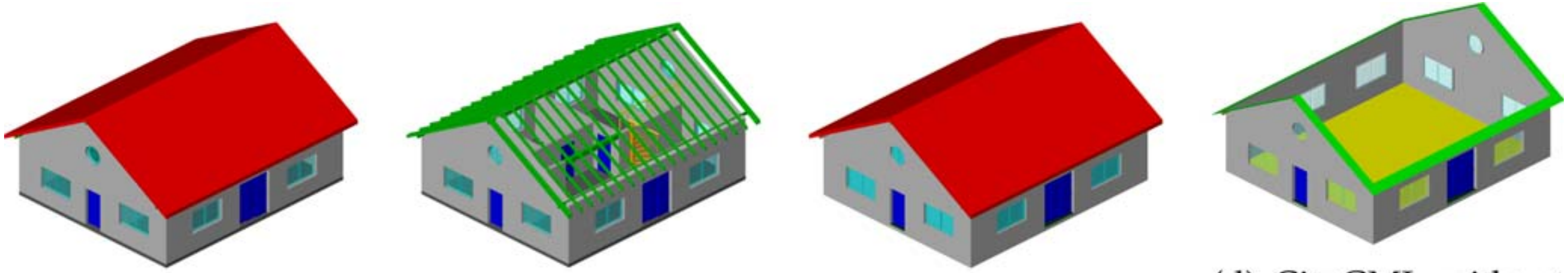


Methodology (geometry)



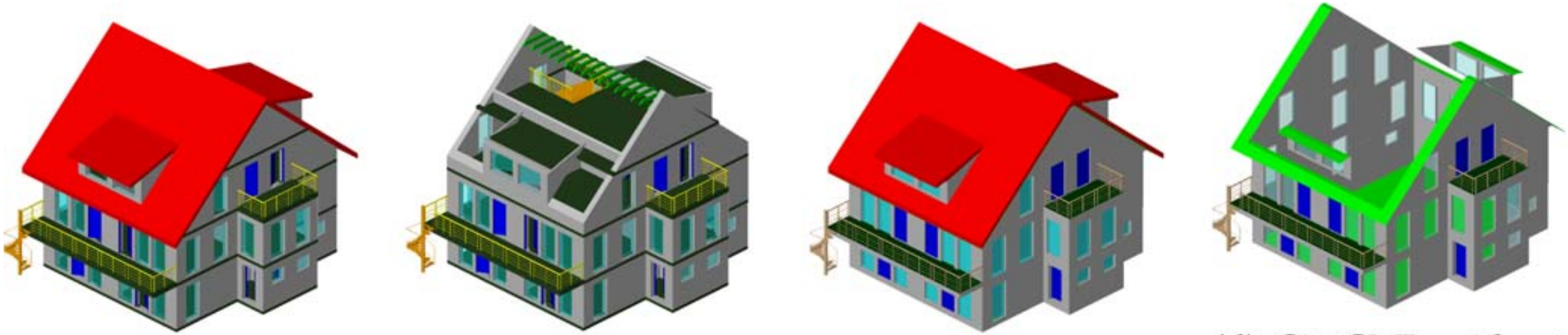
(a) input (b) union (c) dilation (d) result (e) erosion (f) final result

Results



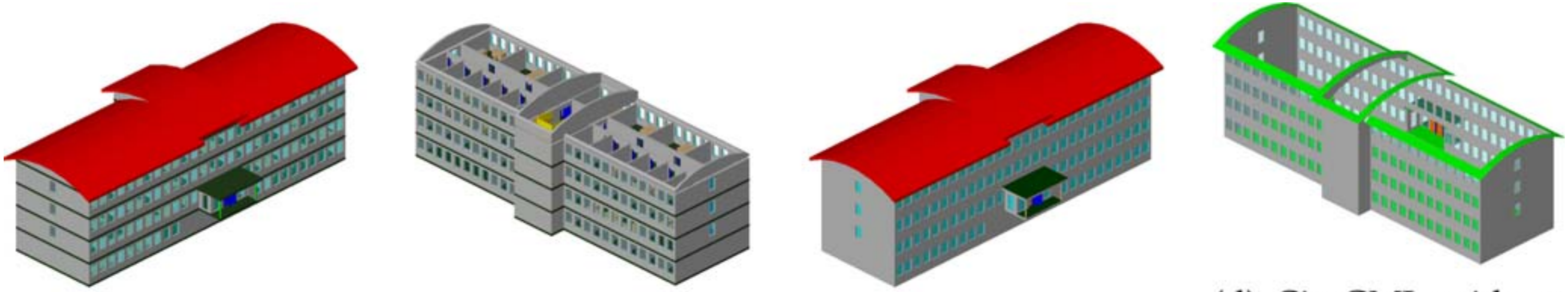
(a) input IFC (b) IFC without roof (c) output CityGML (d) CityGML without roof

Results



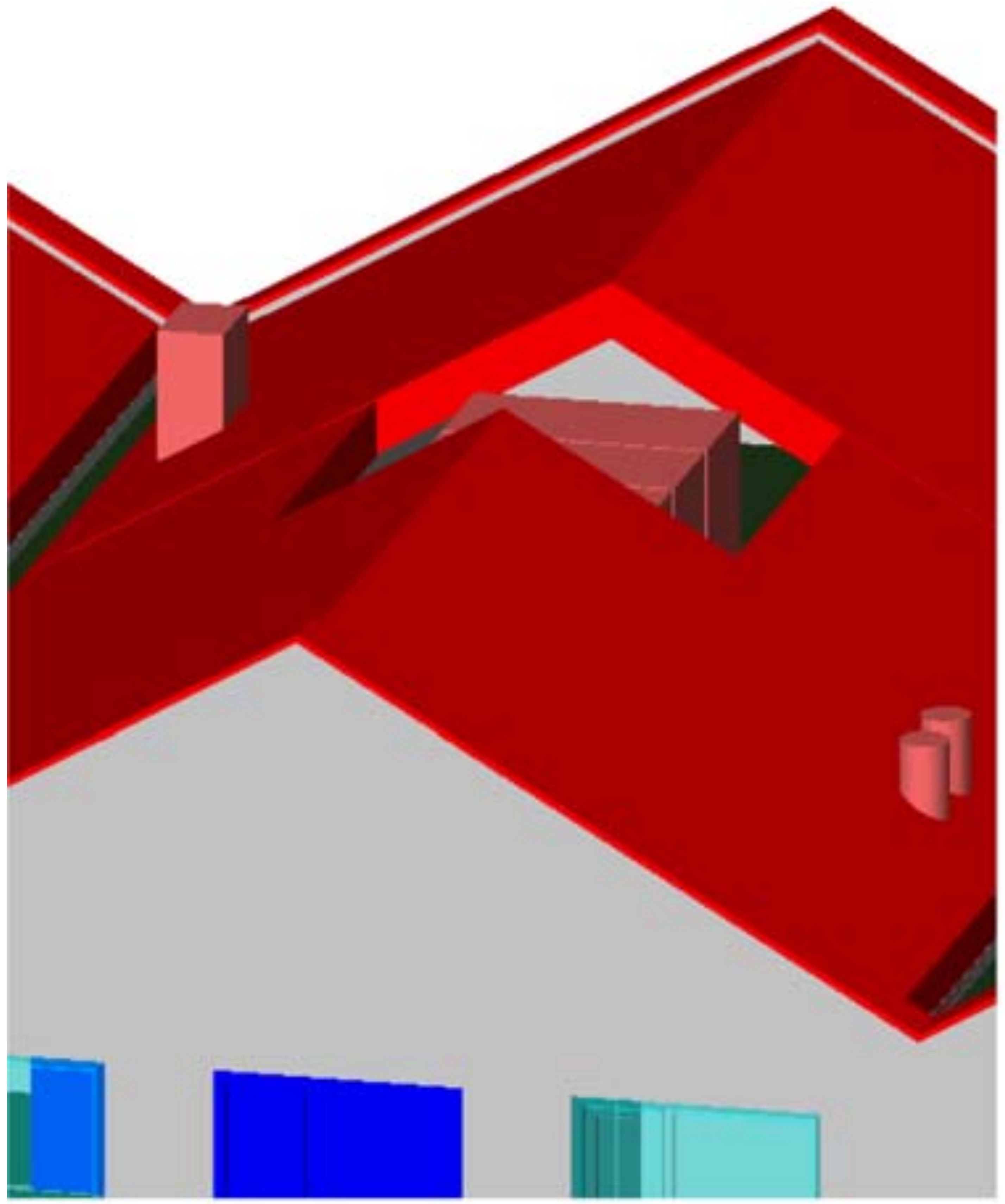
(a) input IFC (b) IFC without roof (c) output CityGML (d) CityGML without roof

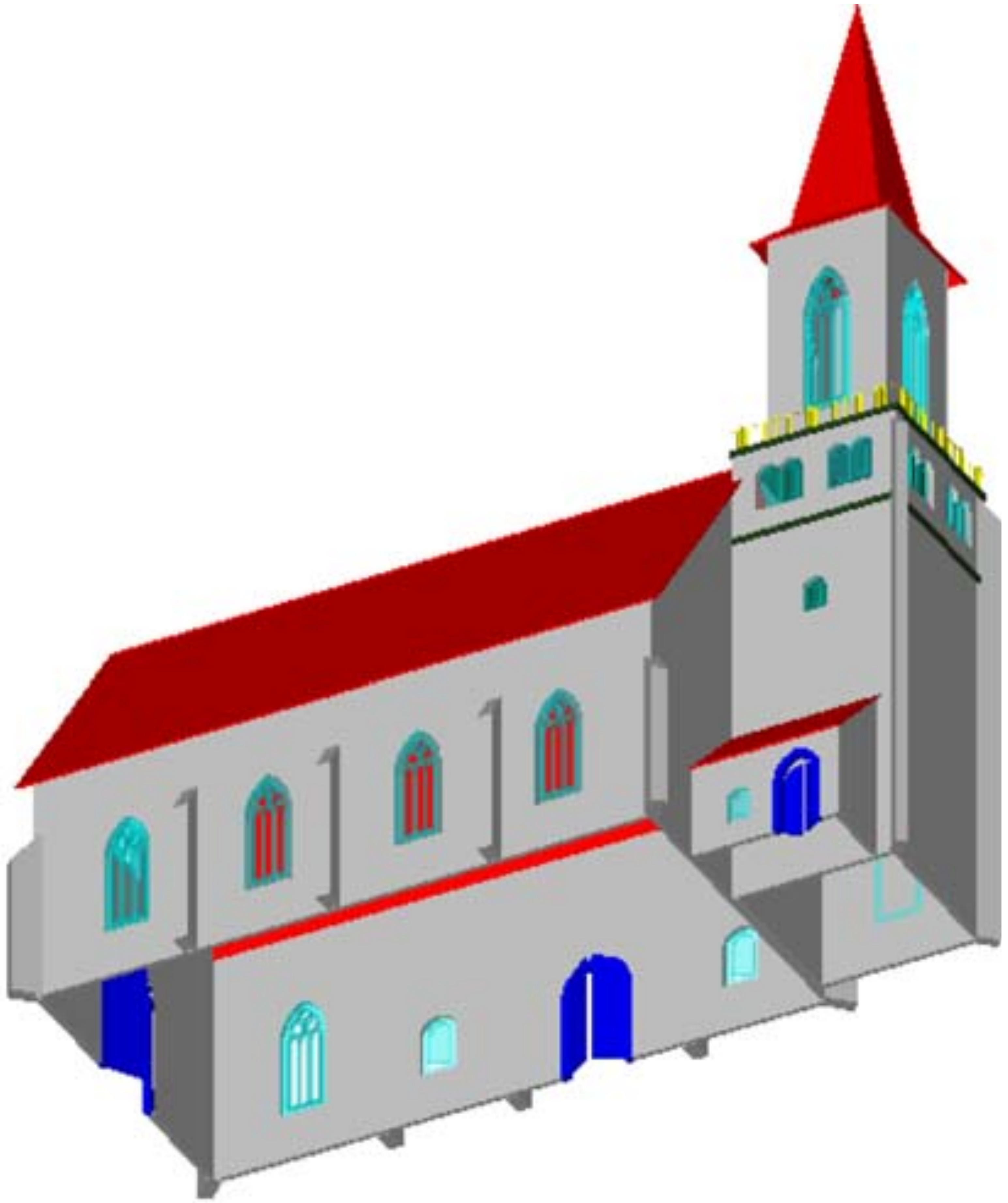
Results



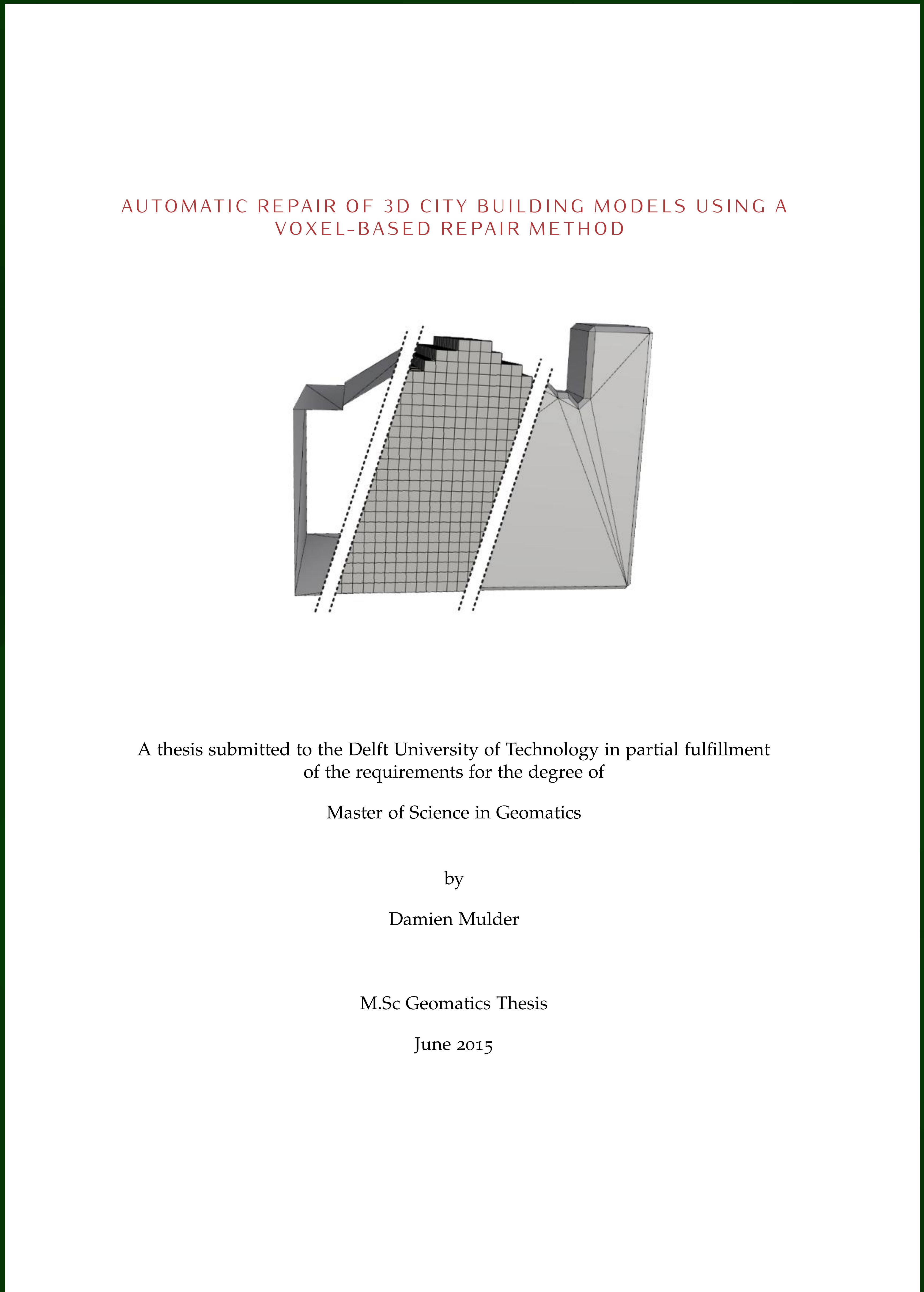
(a) input IFC (b) IFC without roof (c) output CityGML (d) CityGML without roof

Issues


(a) Building where part of the roof is missing


(b) A church missing a base slab

- Motivation: repair 3D models so that they can be used in applications
- Voxelisation
- Reconstruction of mesh
- Obtain semantics and export


AUTOMATIC REPAIR OF 3D CITY BUILDING MODELS USING A VOXEL-BASED REPAIR METHOD

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Geomatics

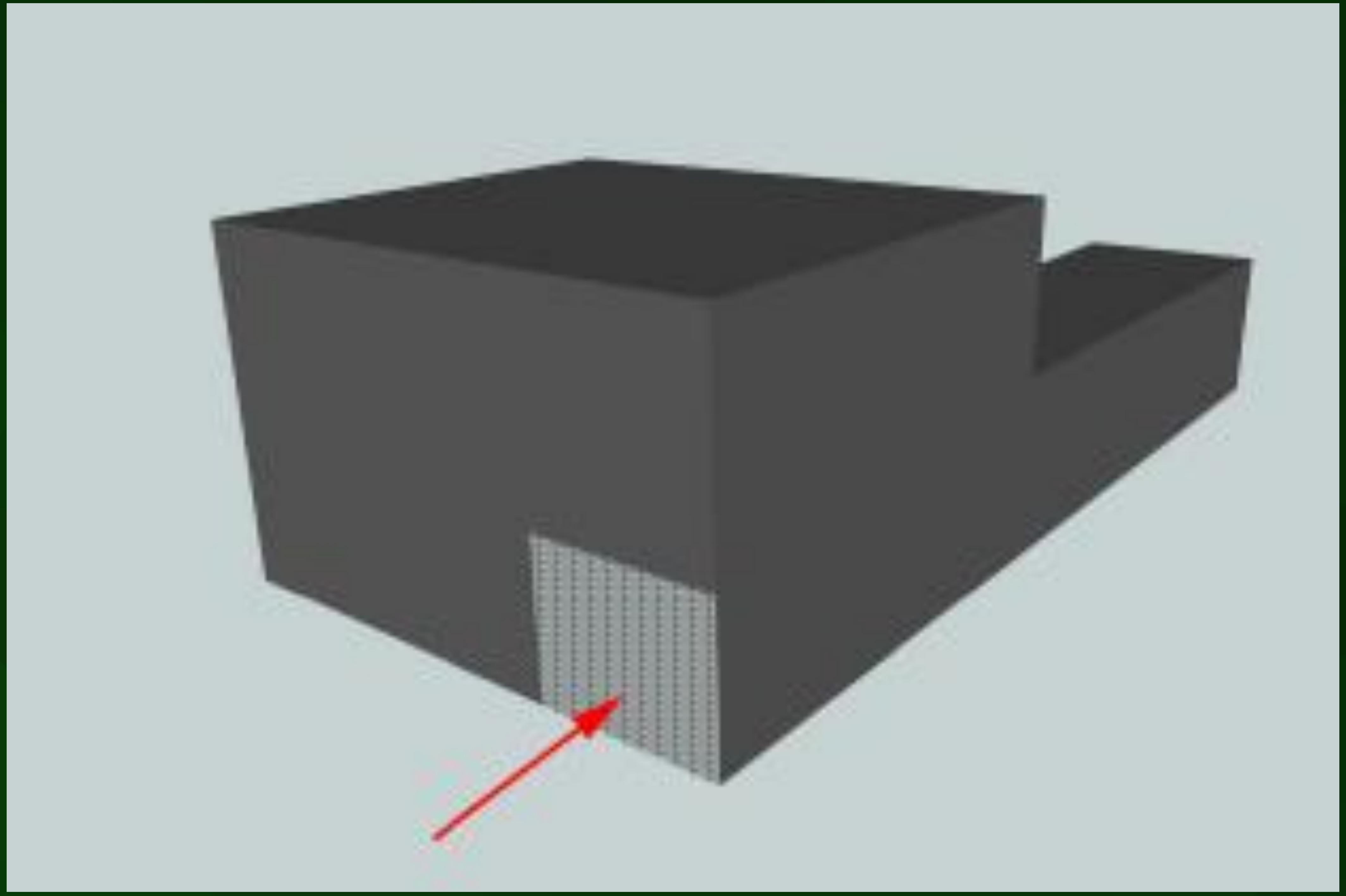
by

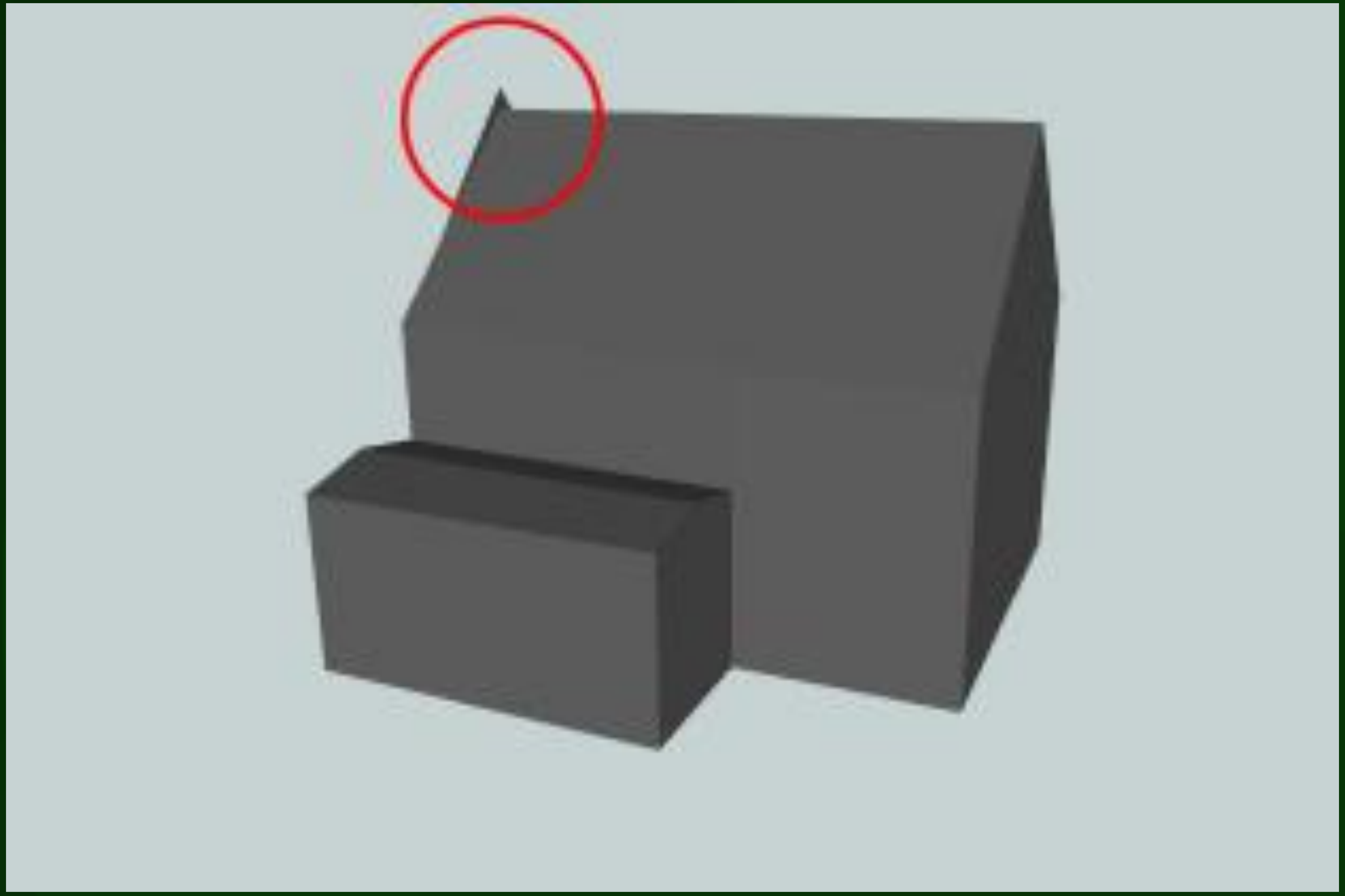
Damien Mulder

M.Sc Geomatics Thesis

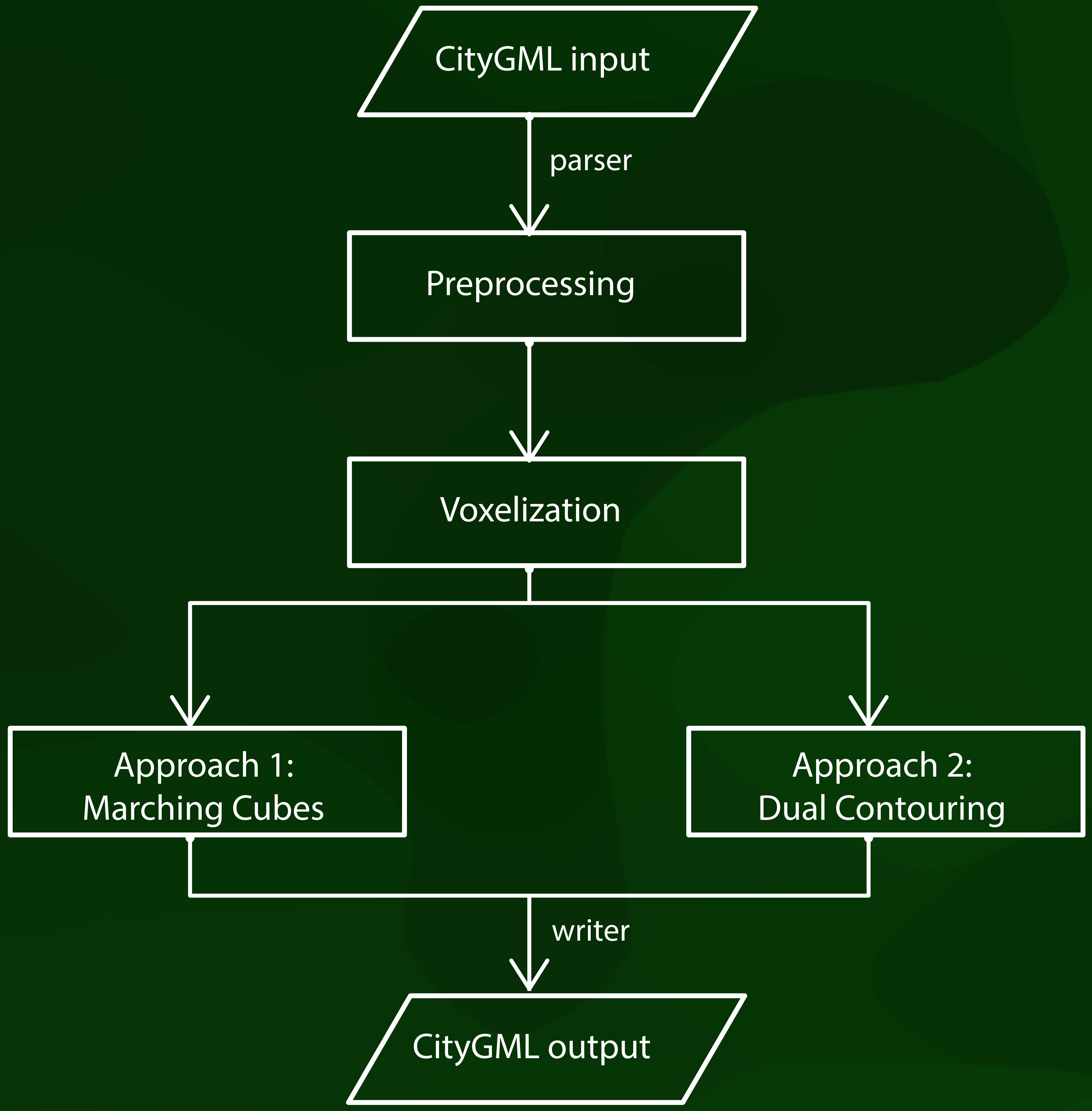
June 2015

Fixing 3D models

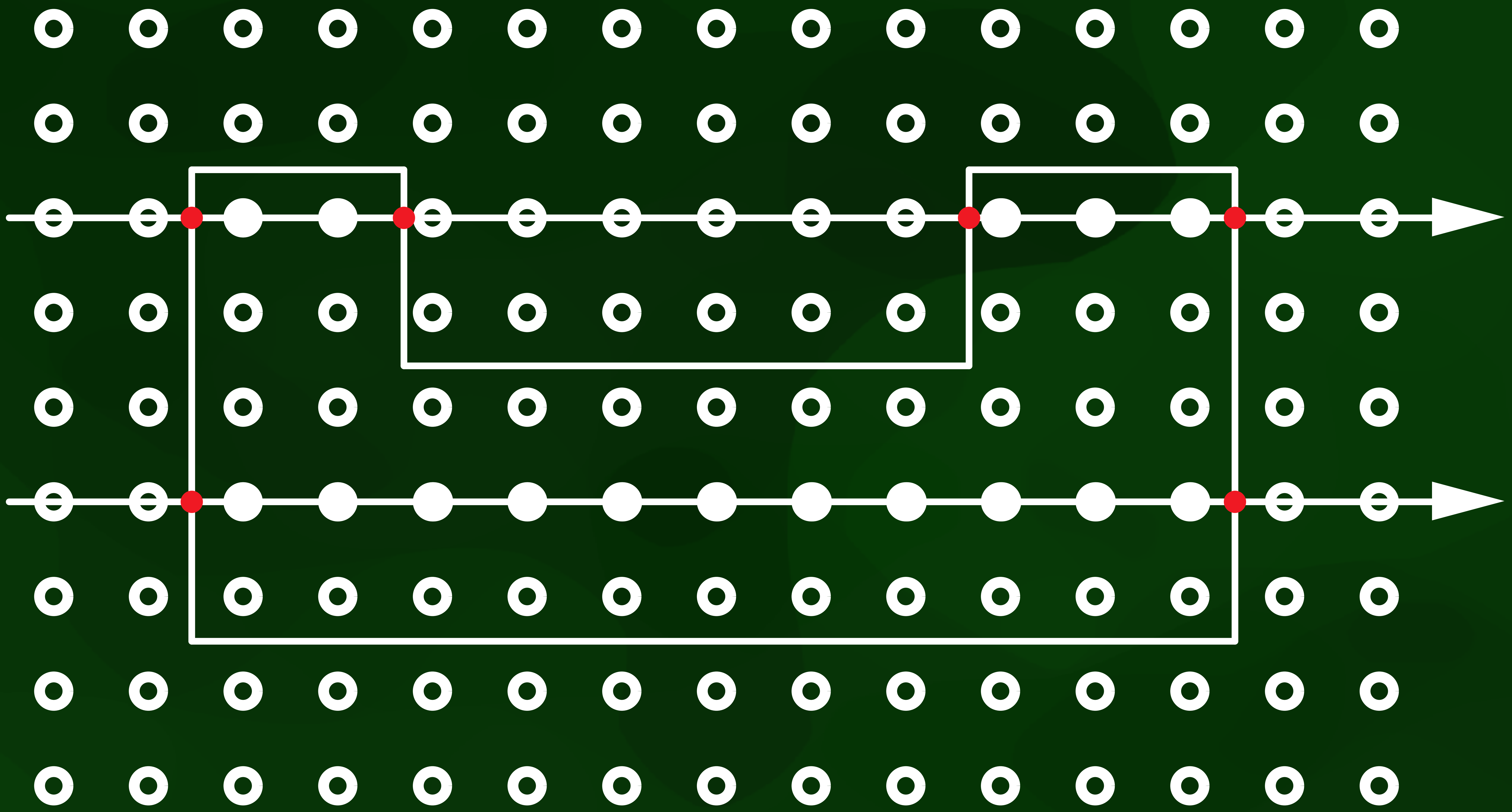




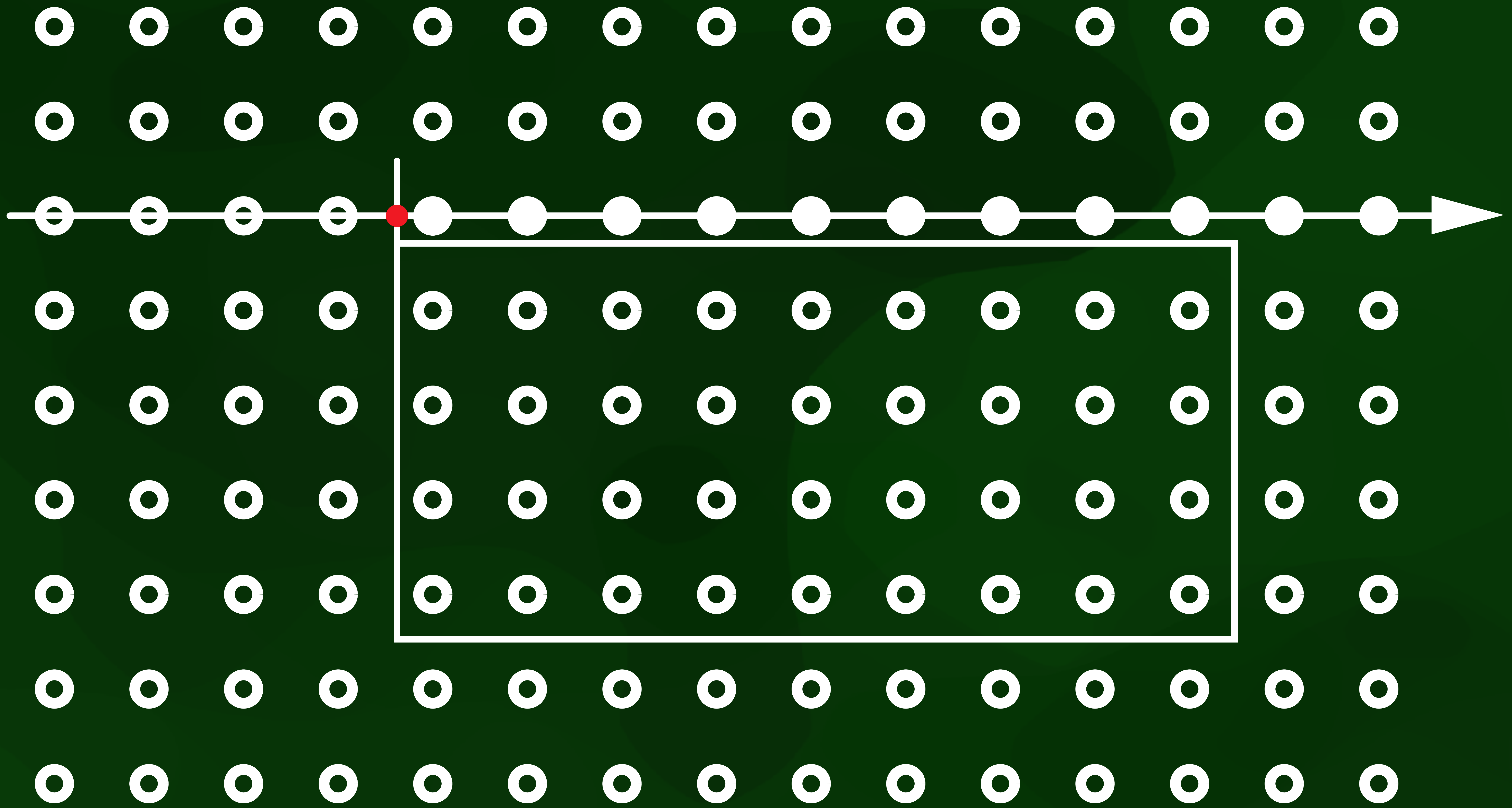
Methodology



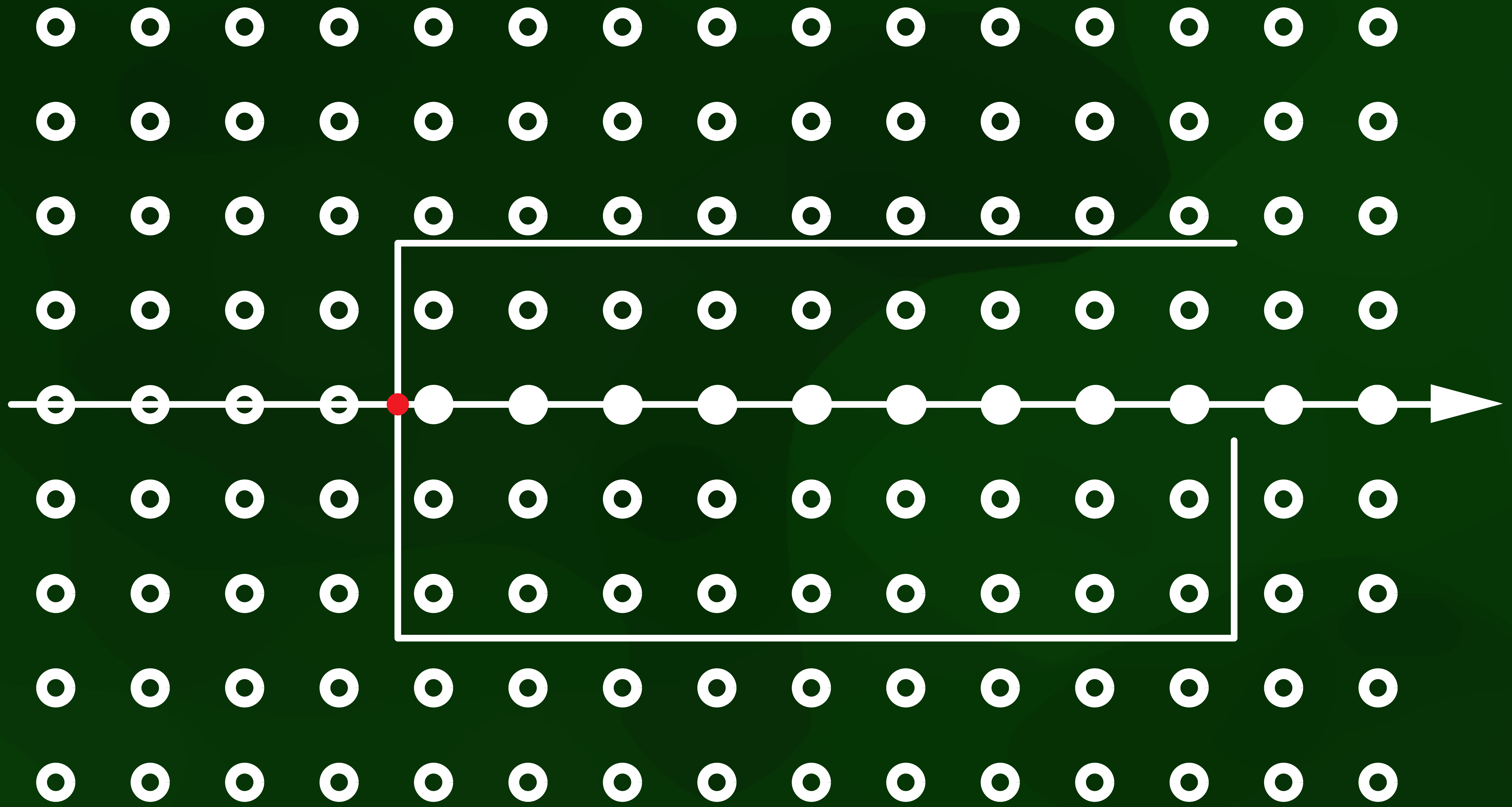
Voxelisation



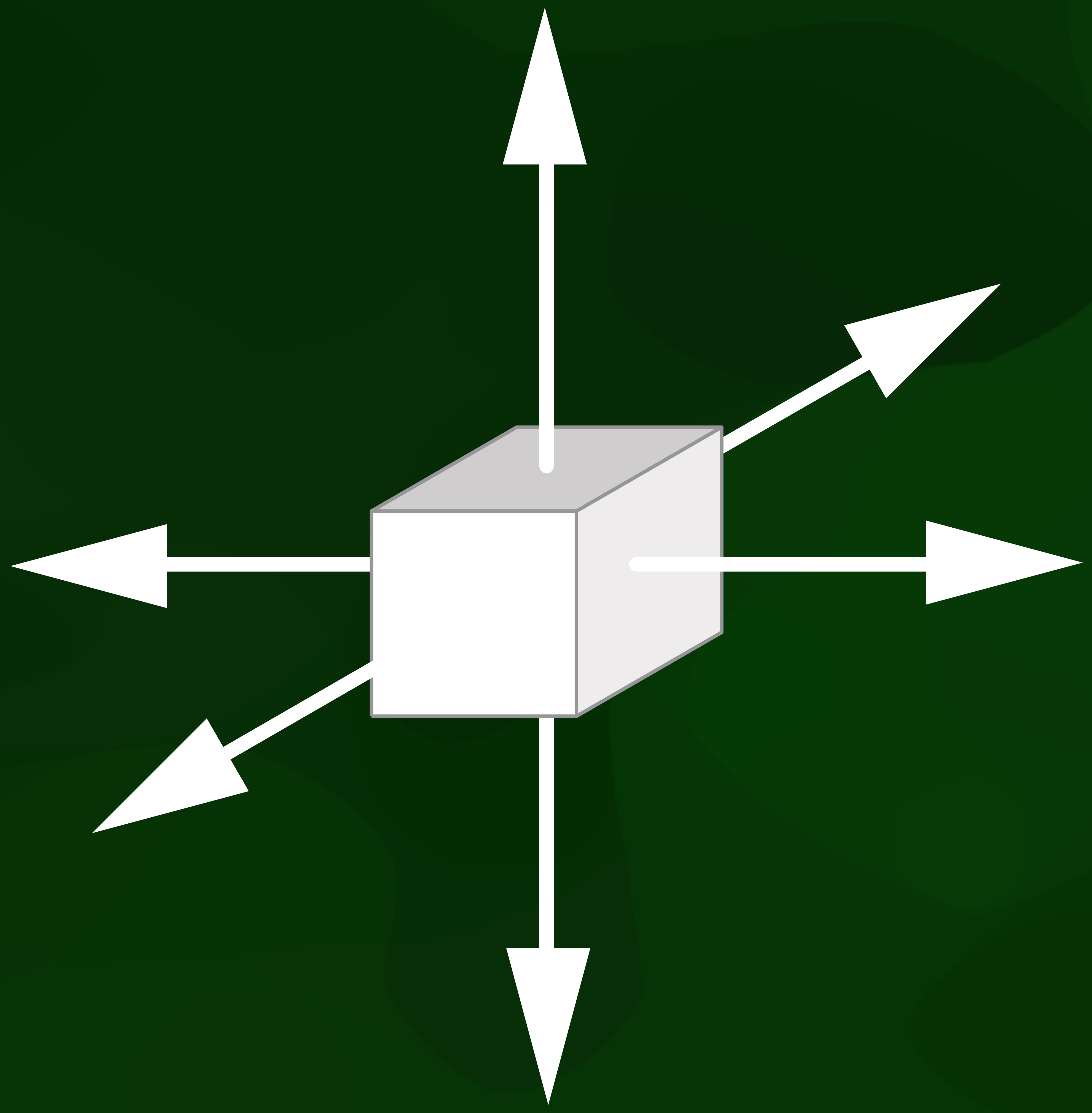
Voxelisation: overshoot



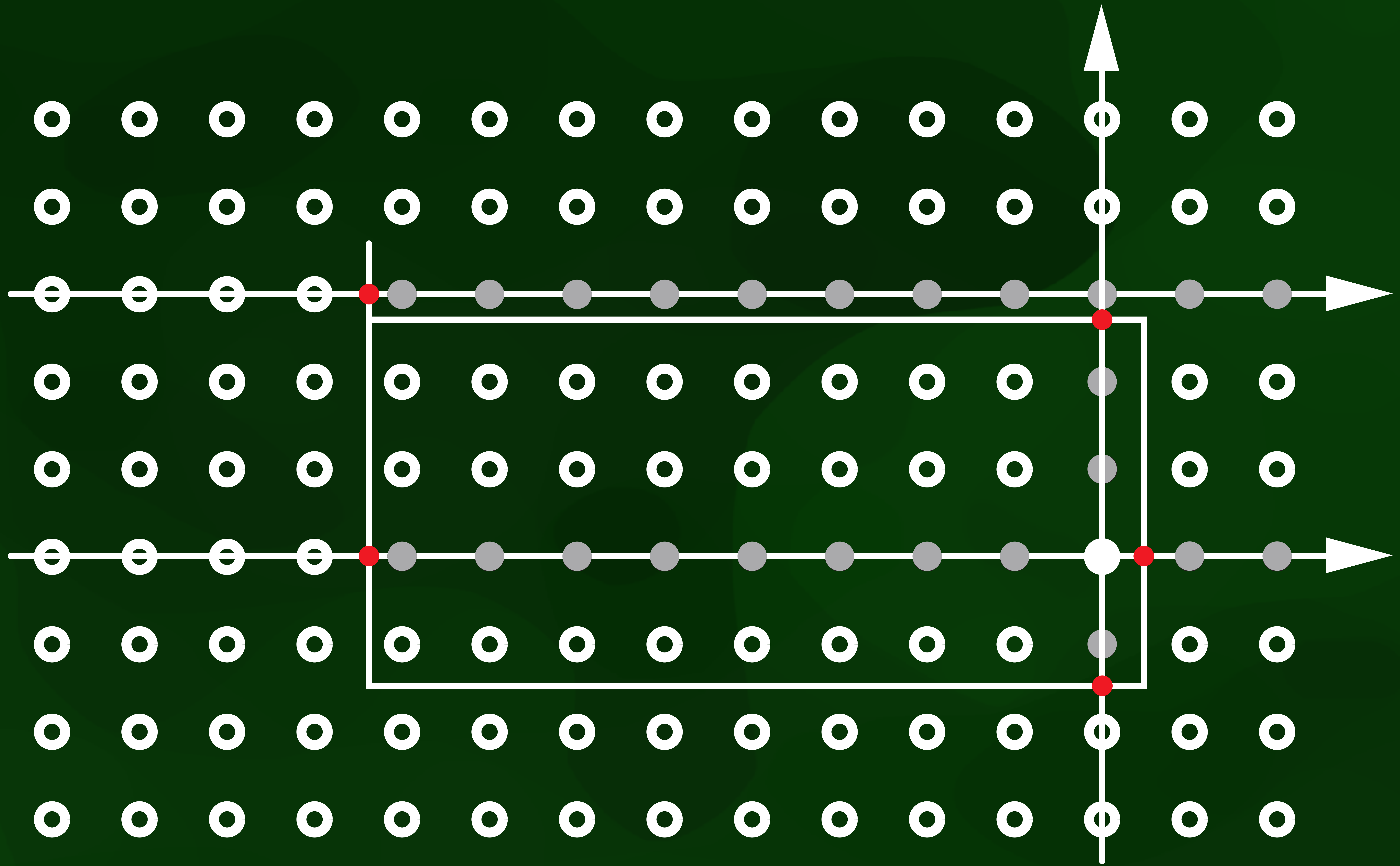
Voxelisation: gap



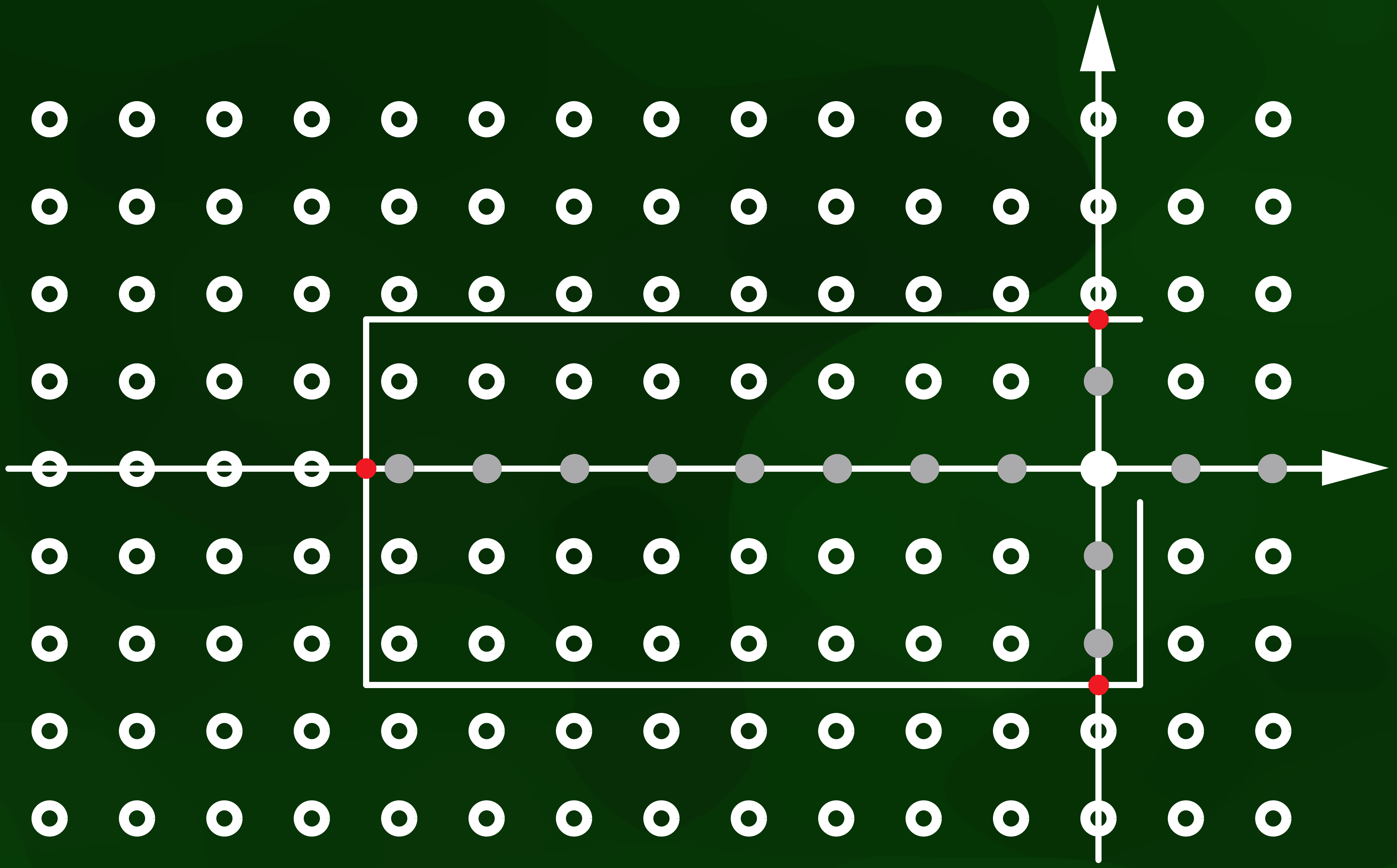
Voxelisation: shooting rays



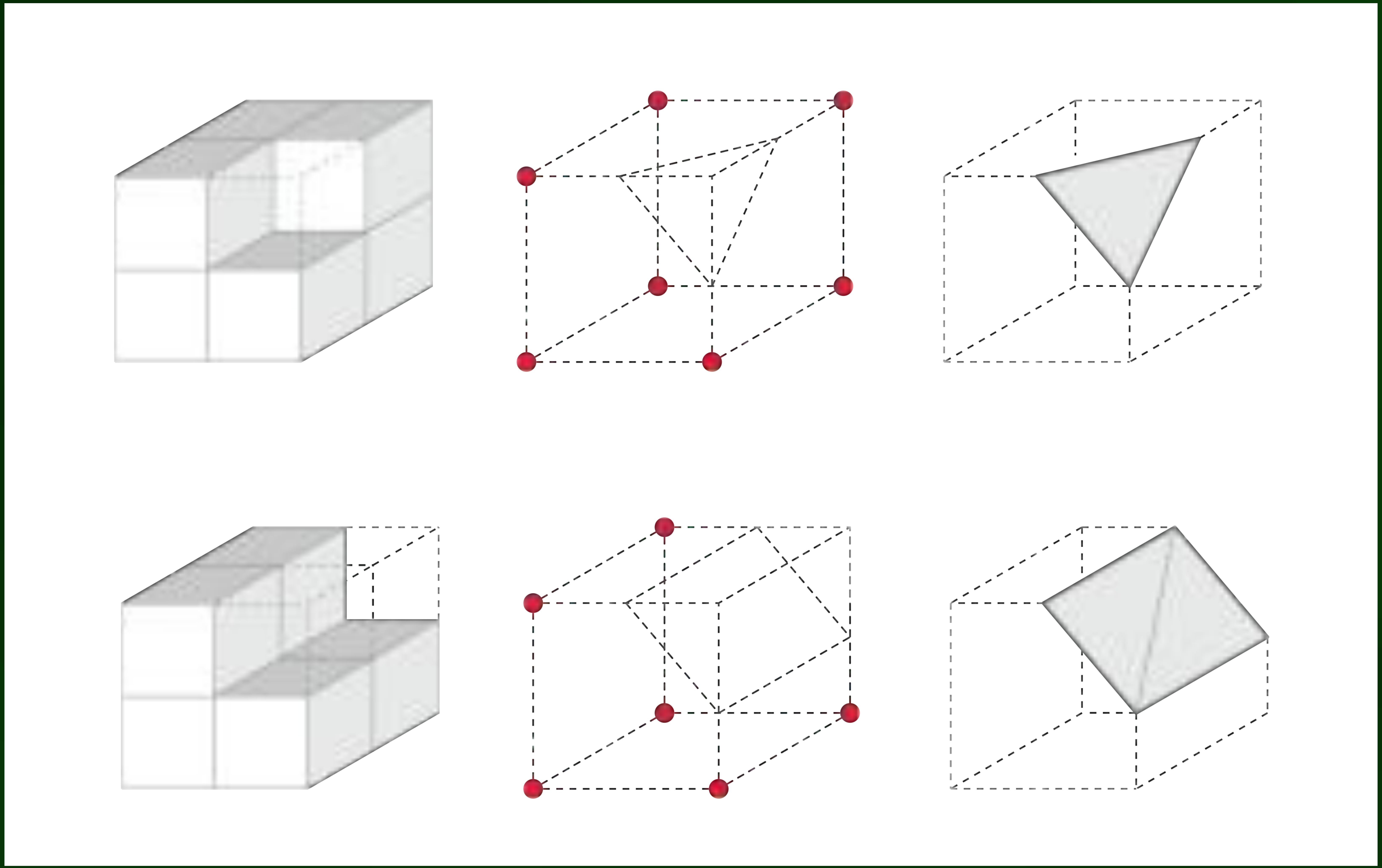
Majority counting: overshoot



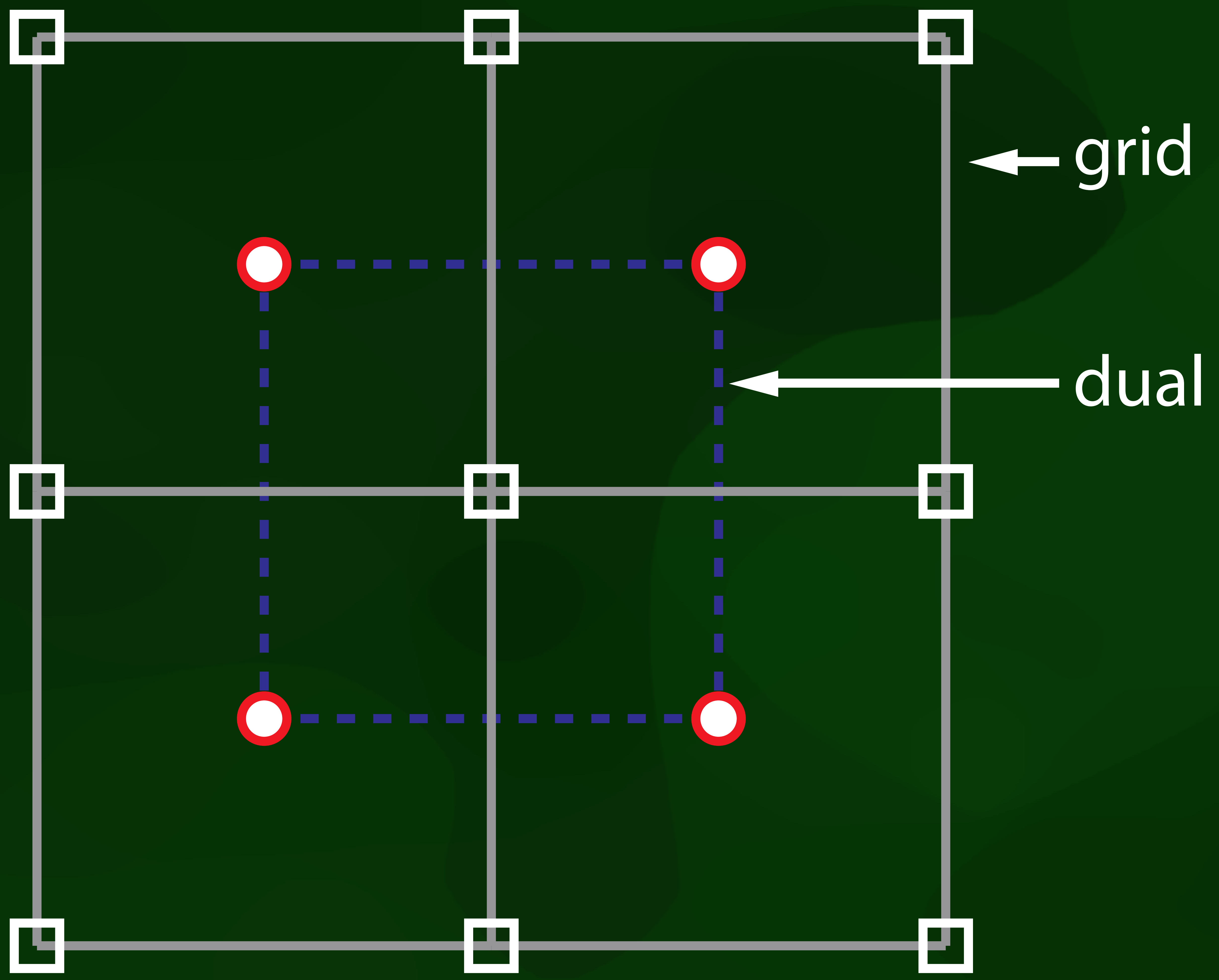
Majority counting: gap



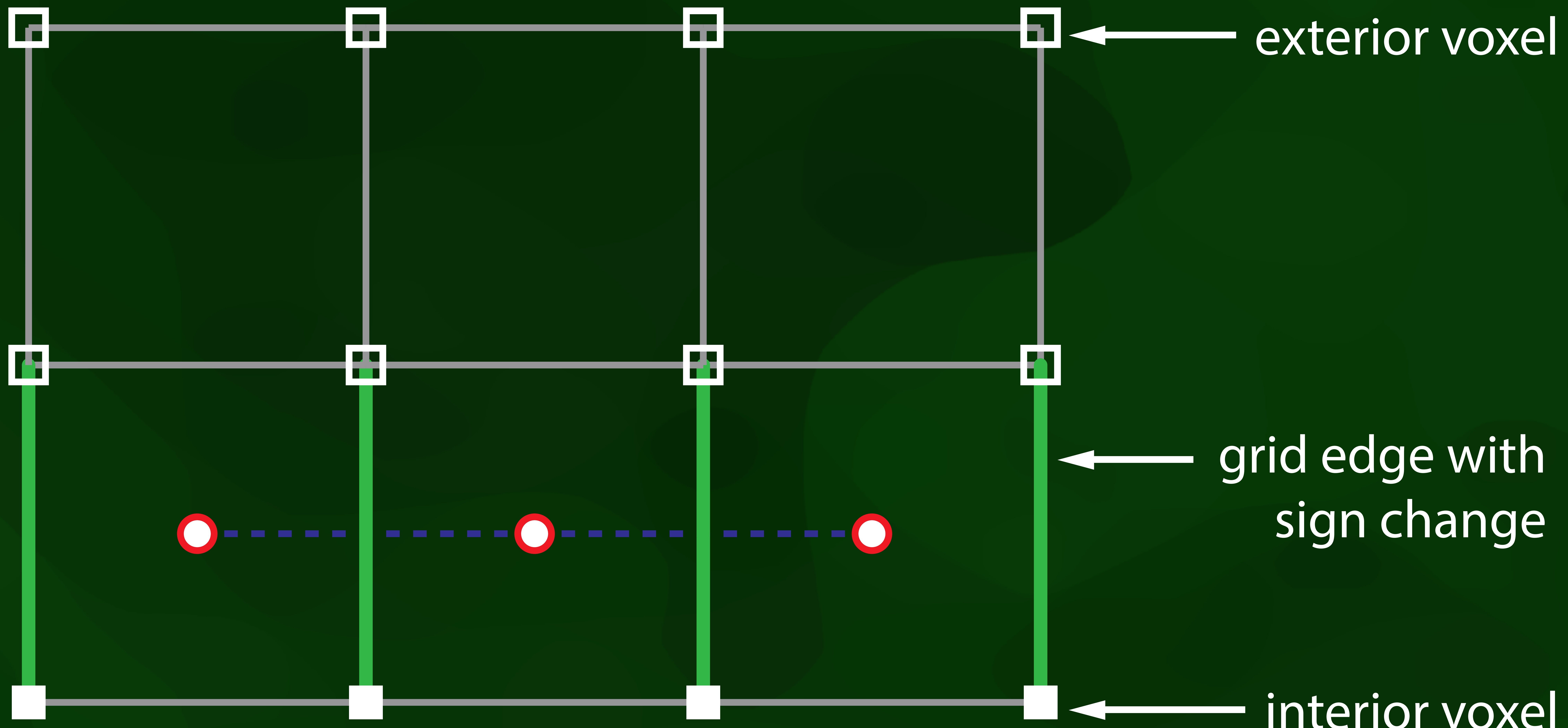
Marching cubes



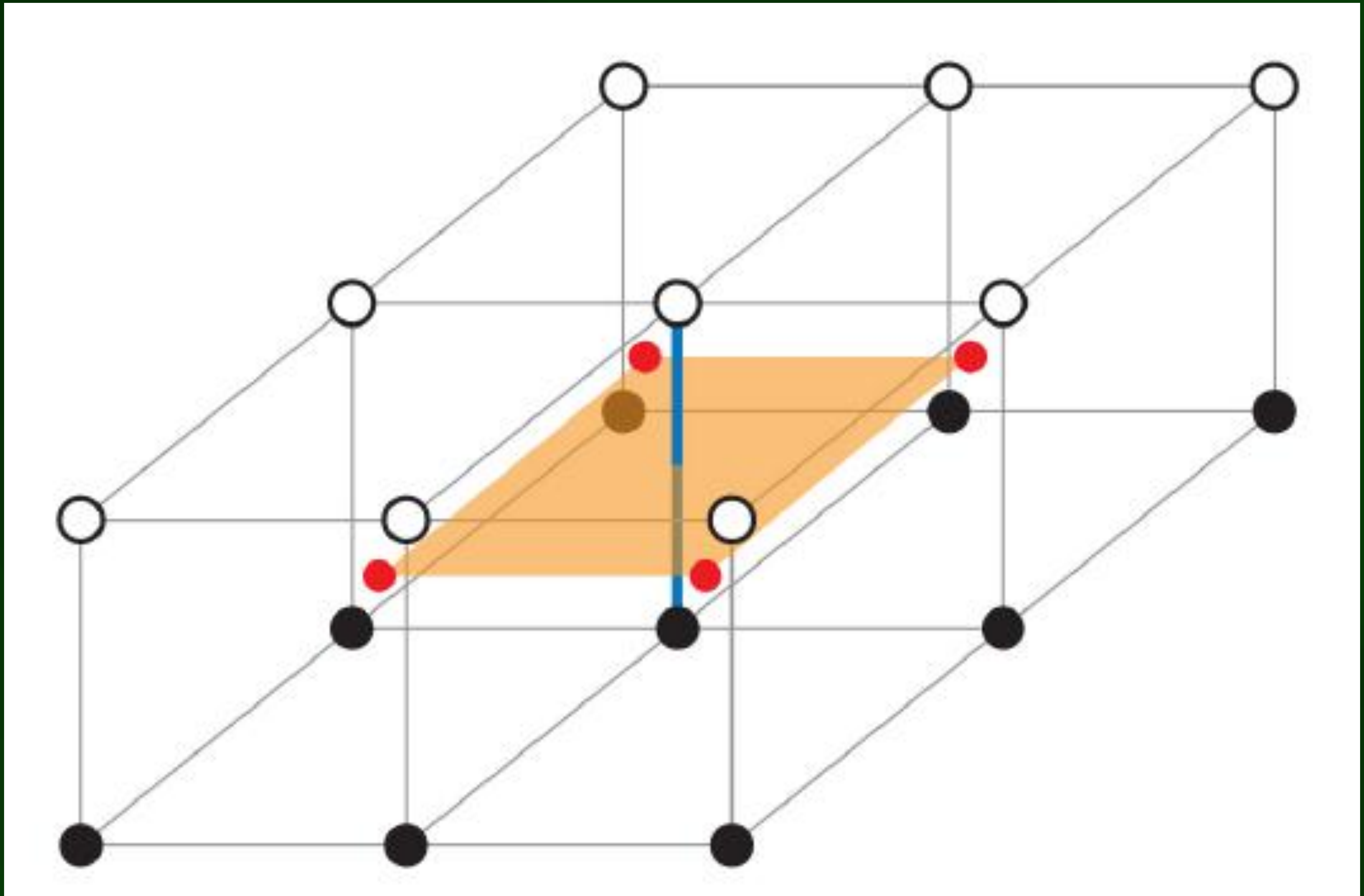
Dual contouring



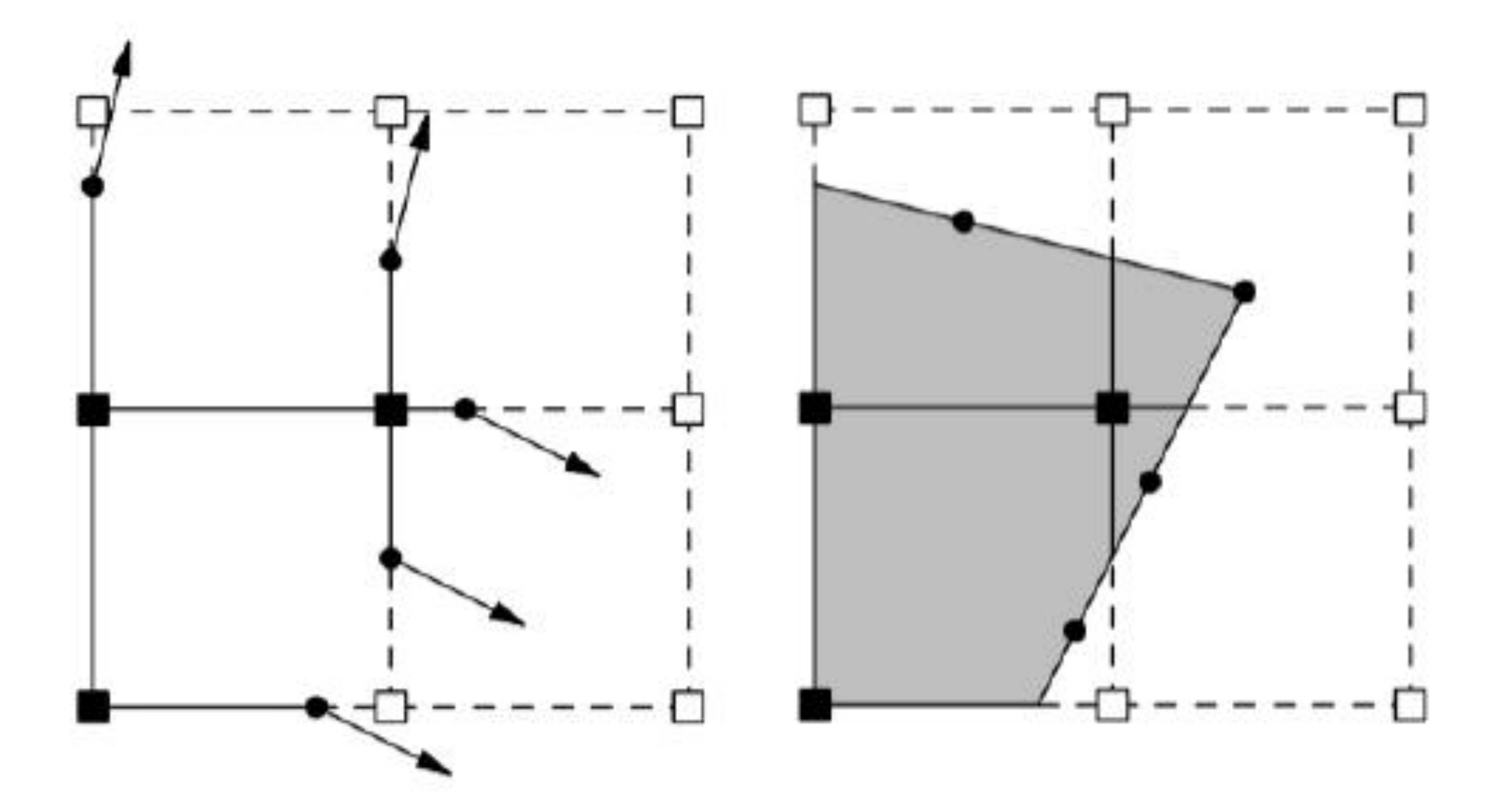
Dual contouring



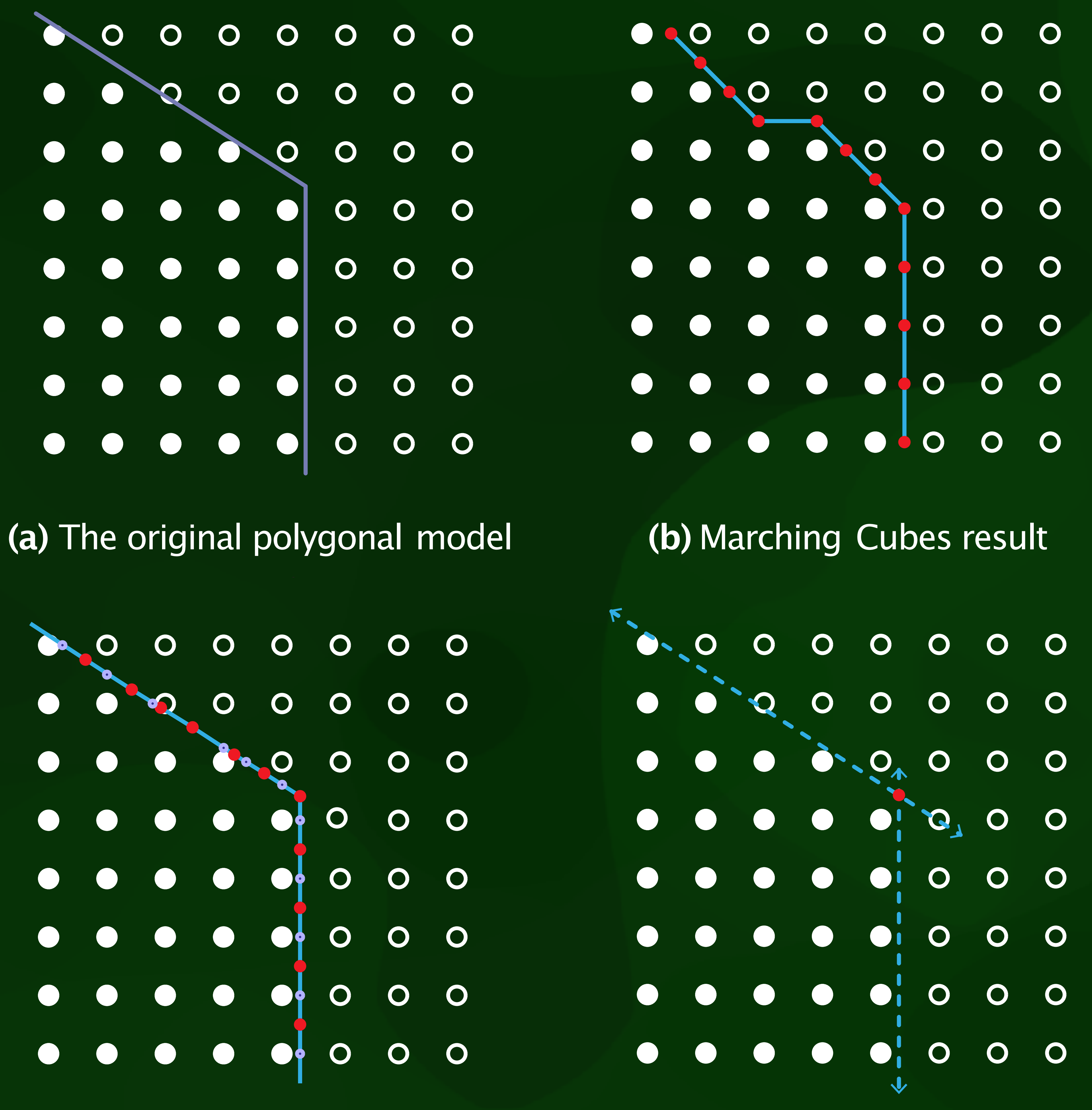
Dual contouring



Dual contouring



Full process



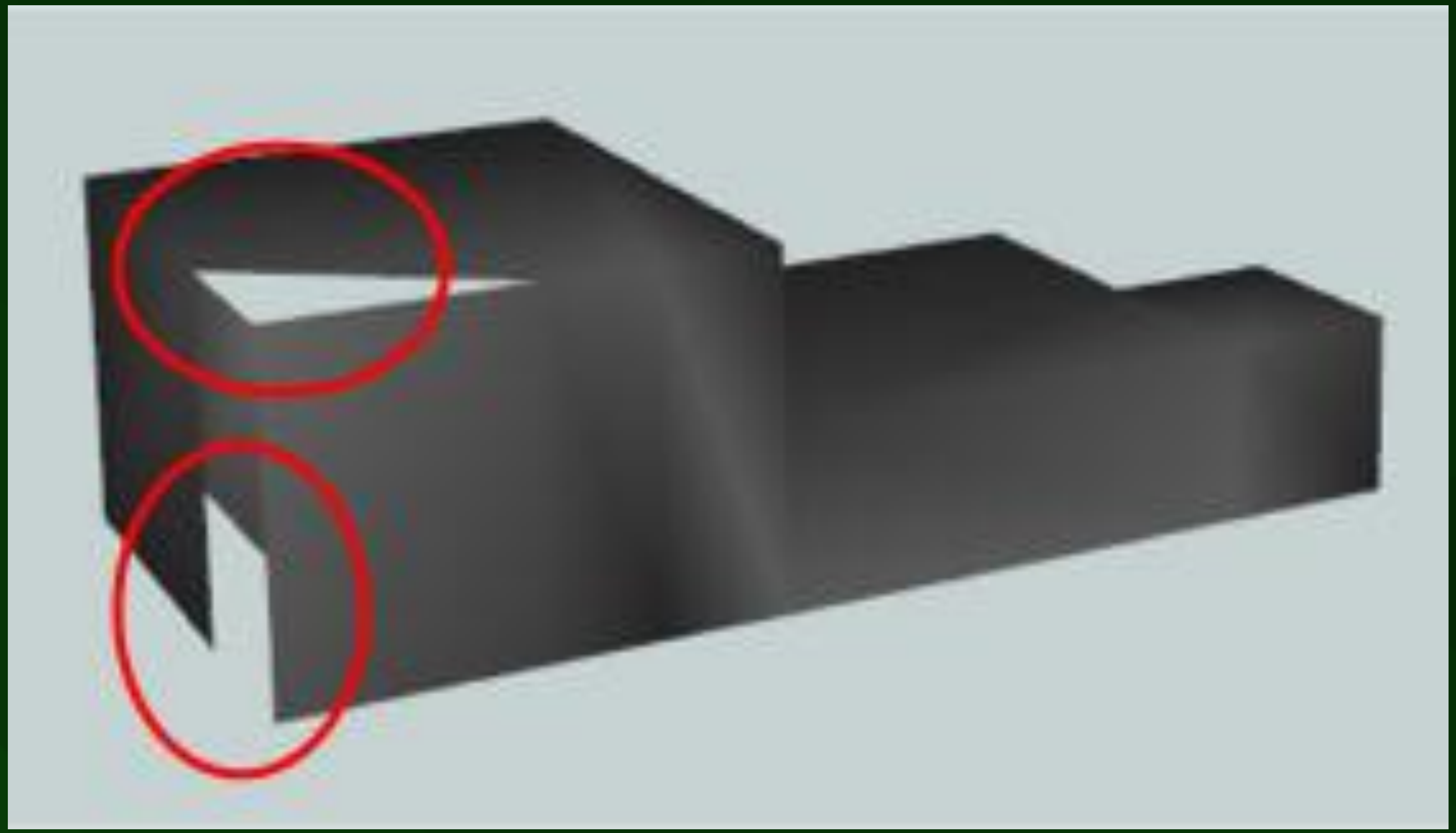
(a) The original polygonal model

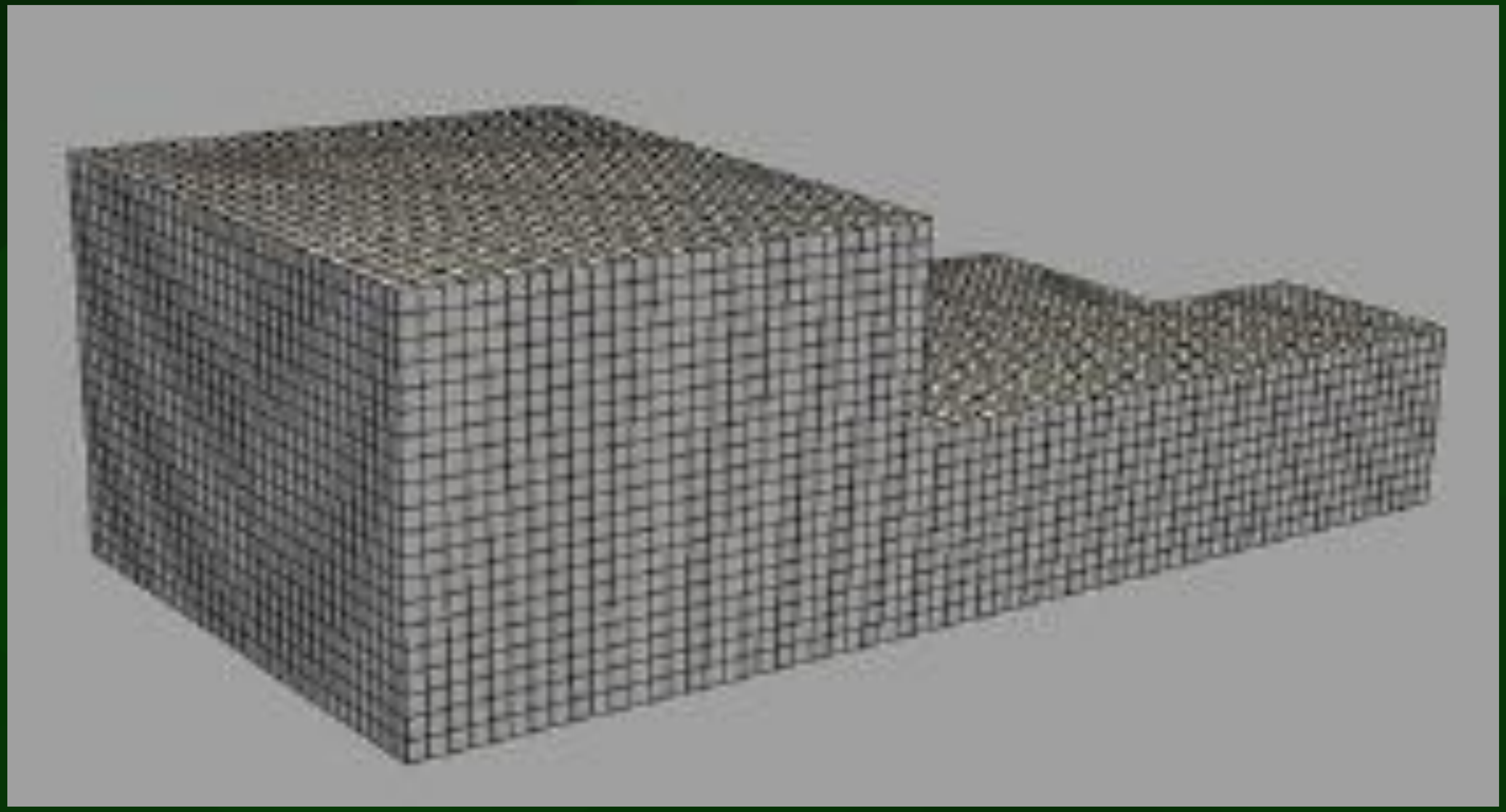
(b) Marching Cubes result

(c) Dual Contouring result

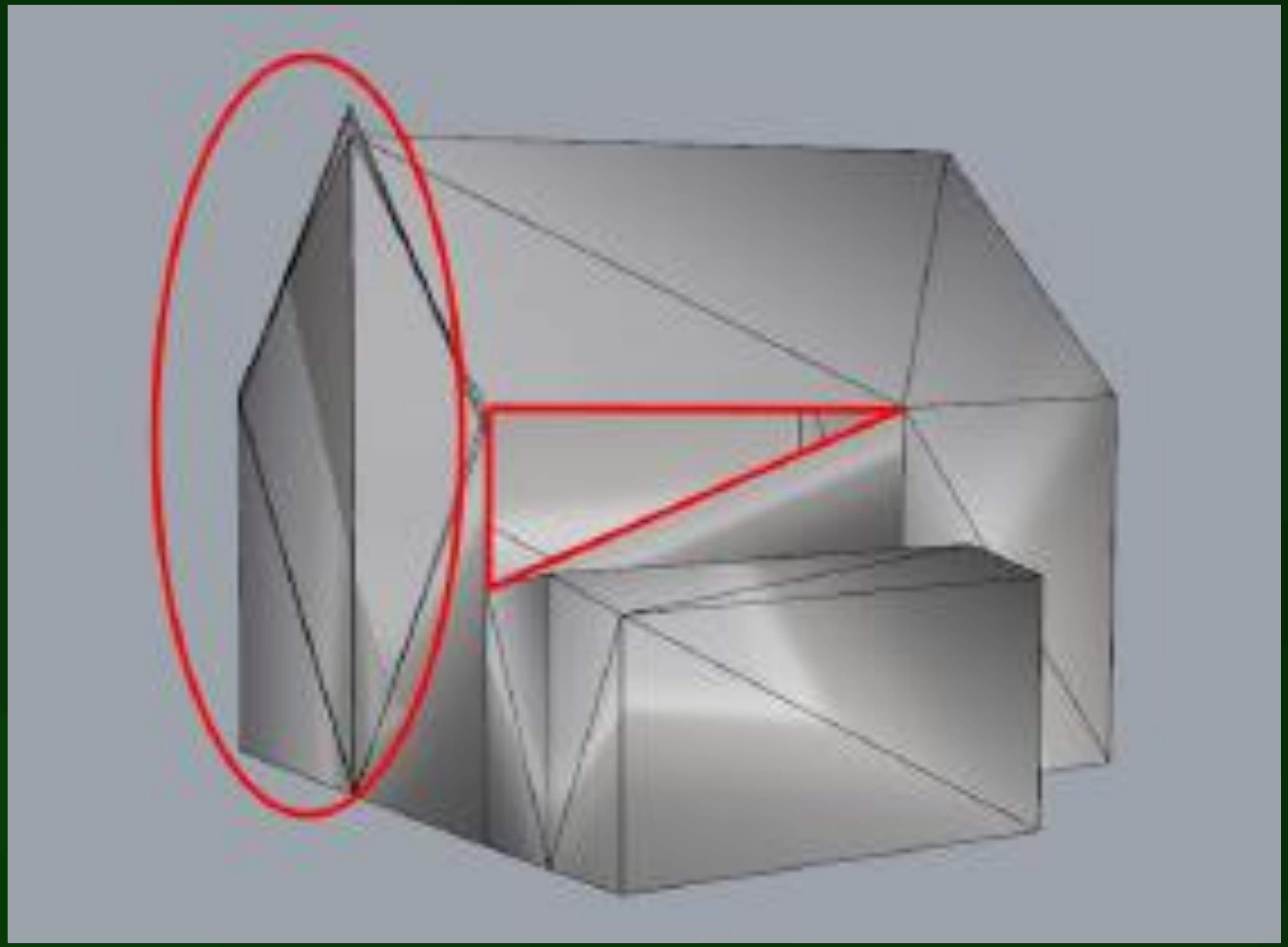
(d) Pressing result

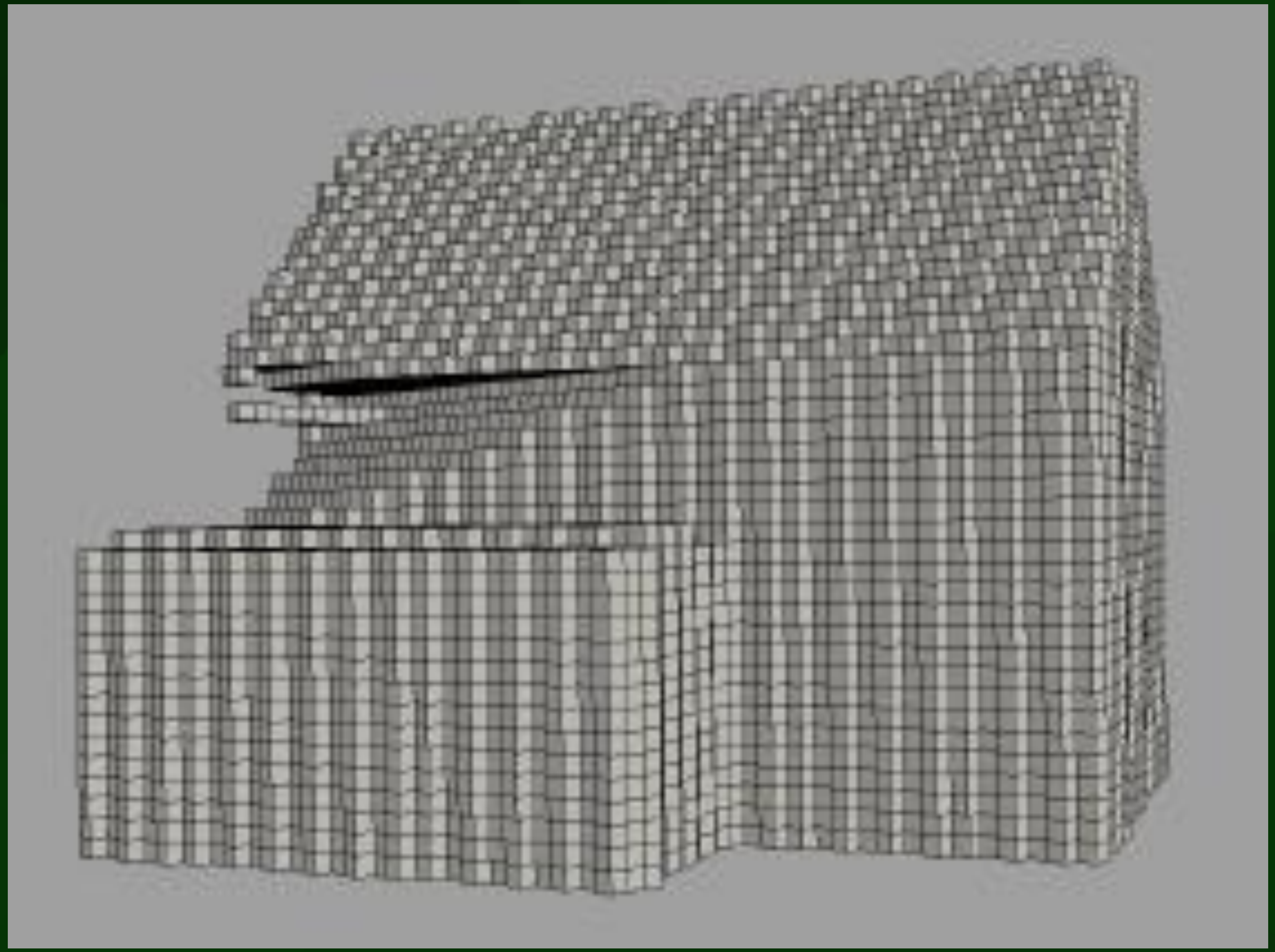
Results



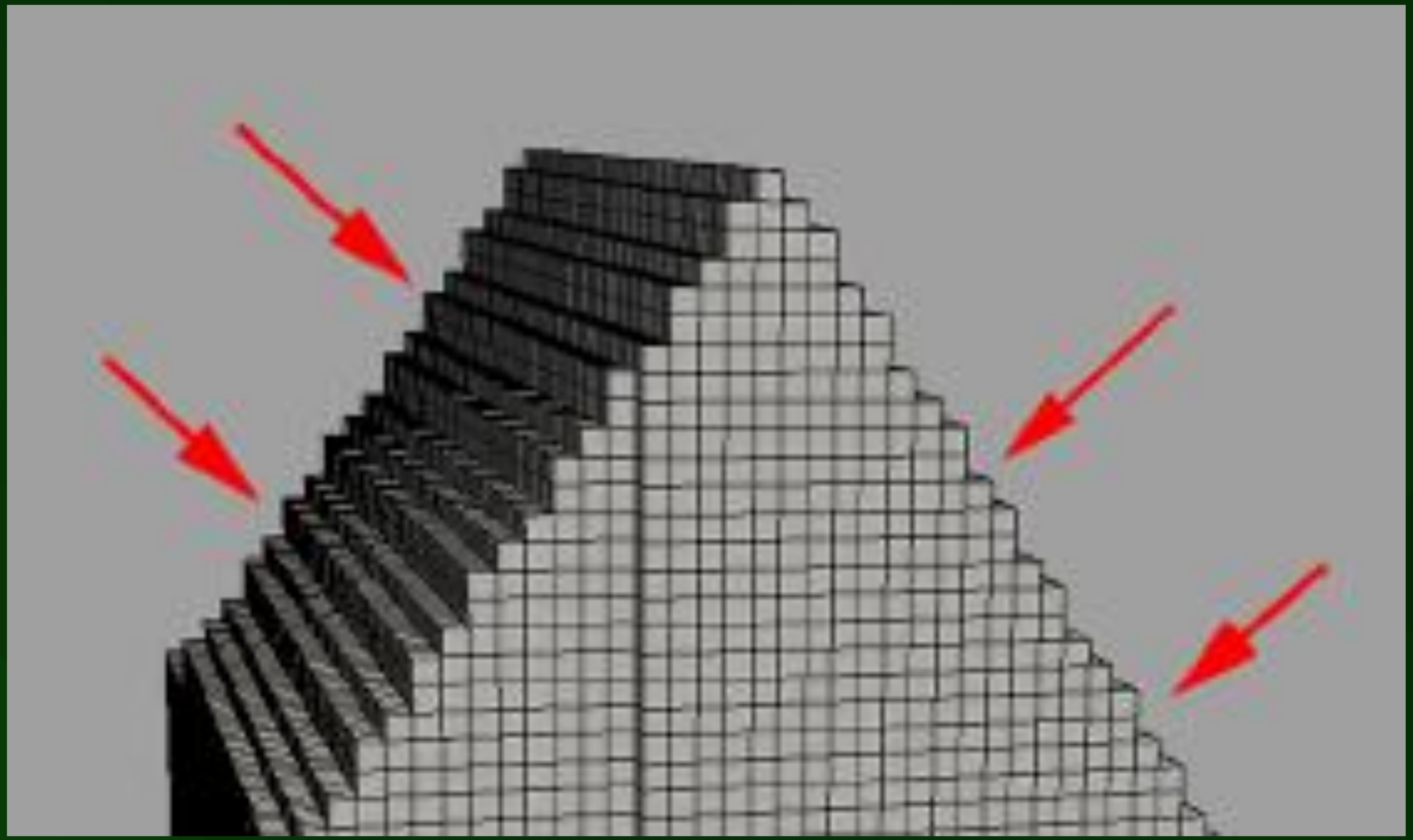


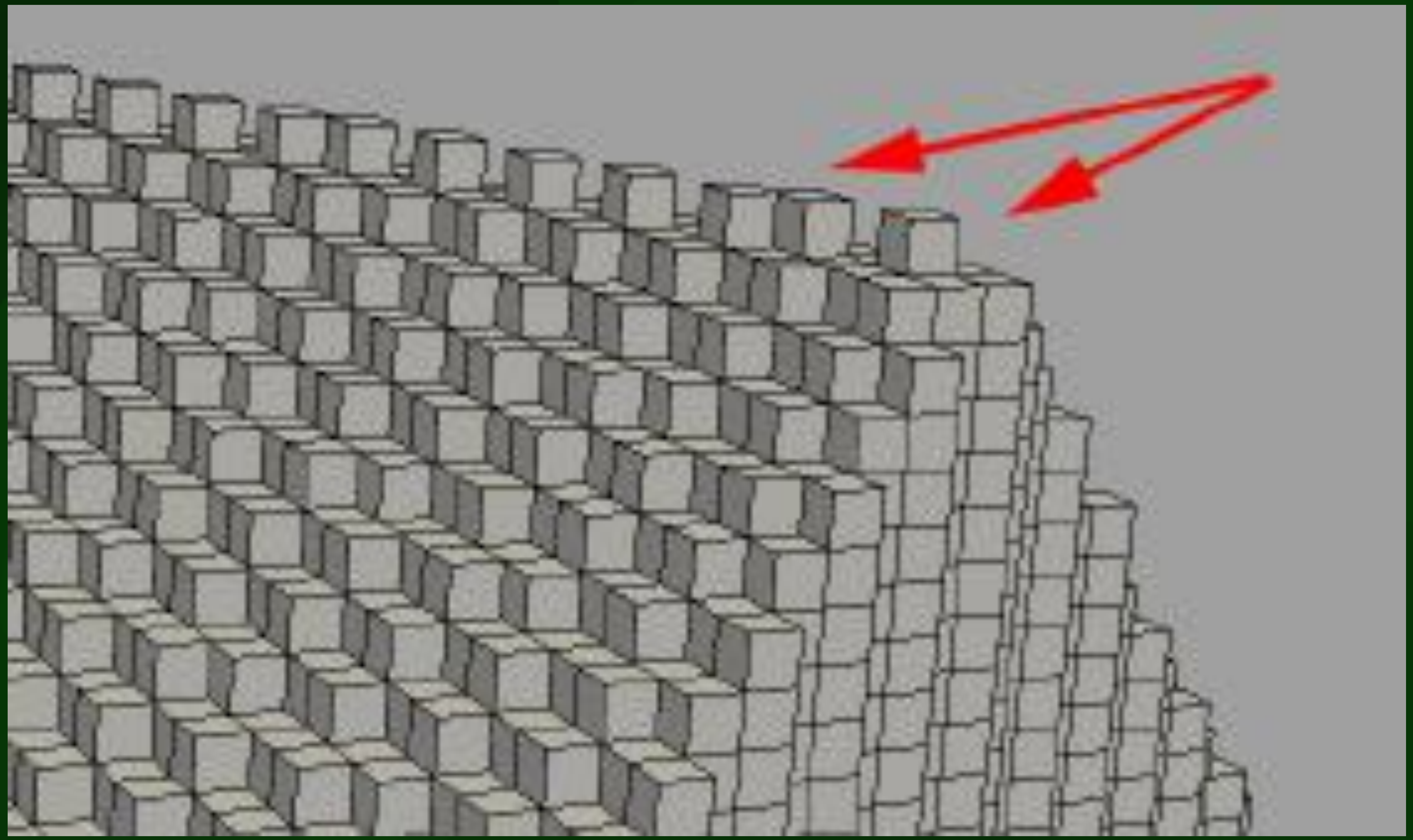
Results



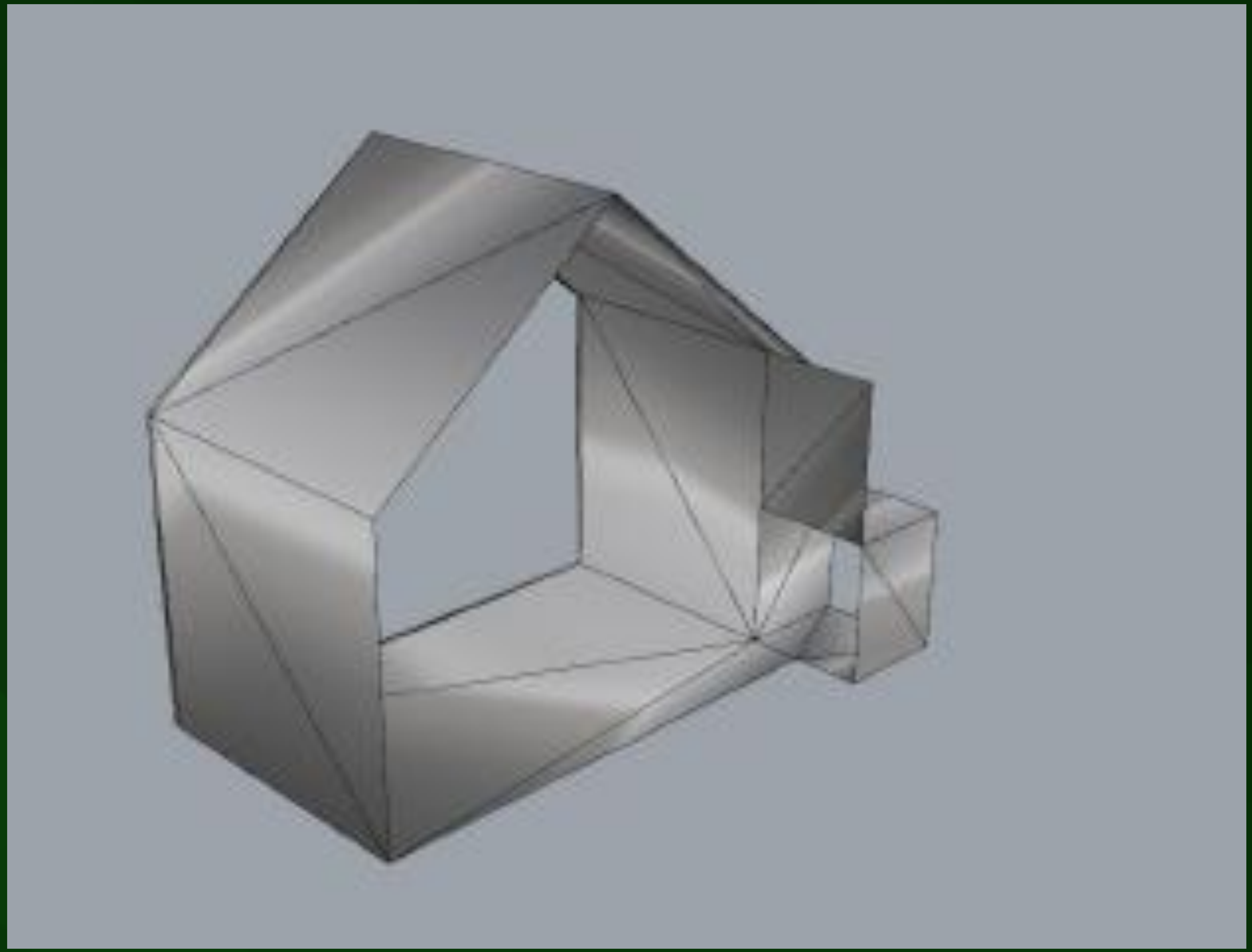


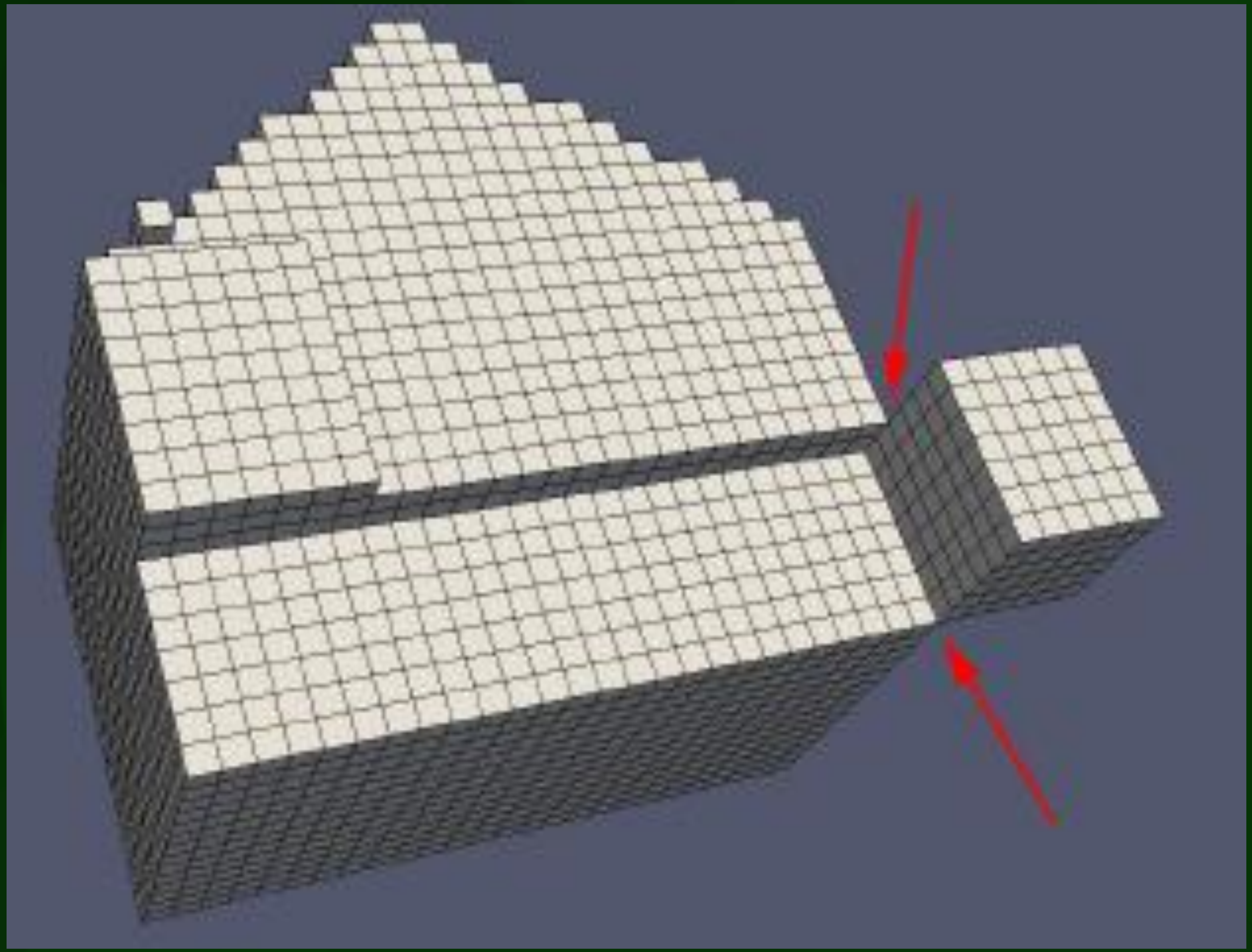
Artefacts



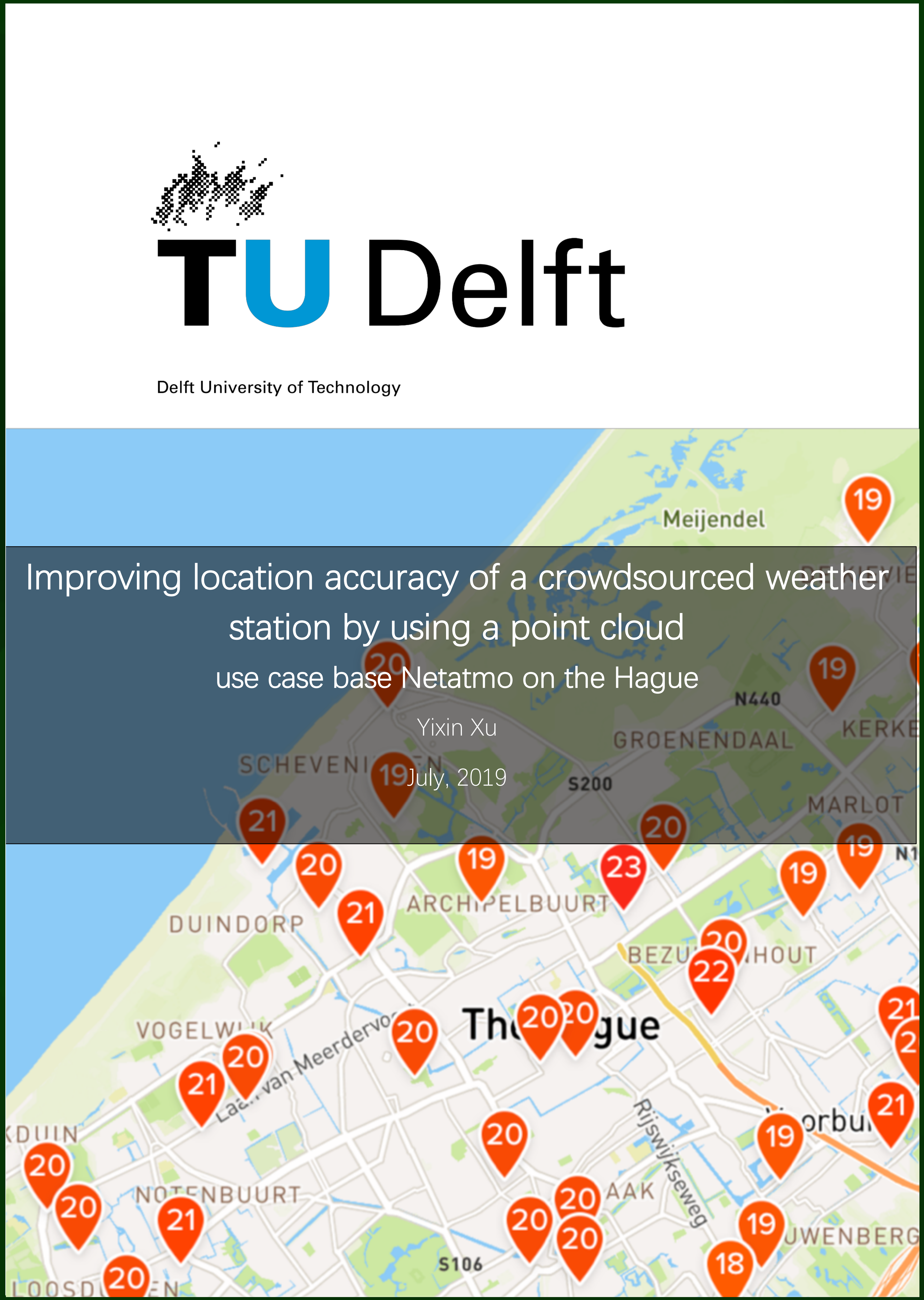


Results



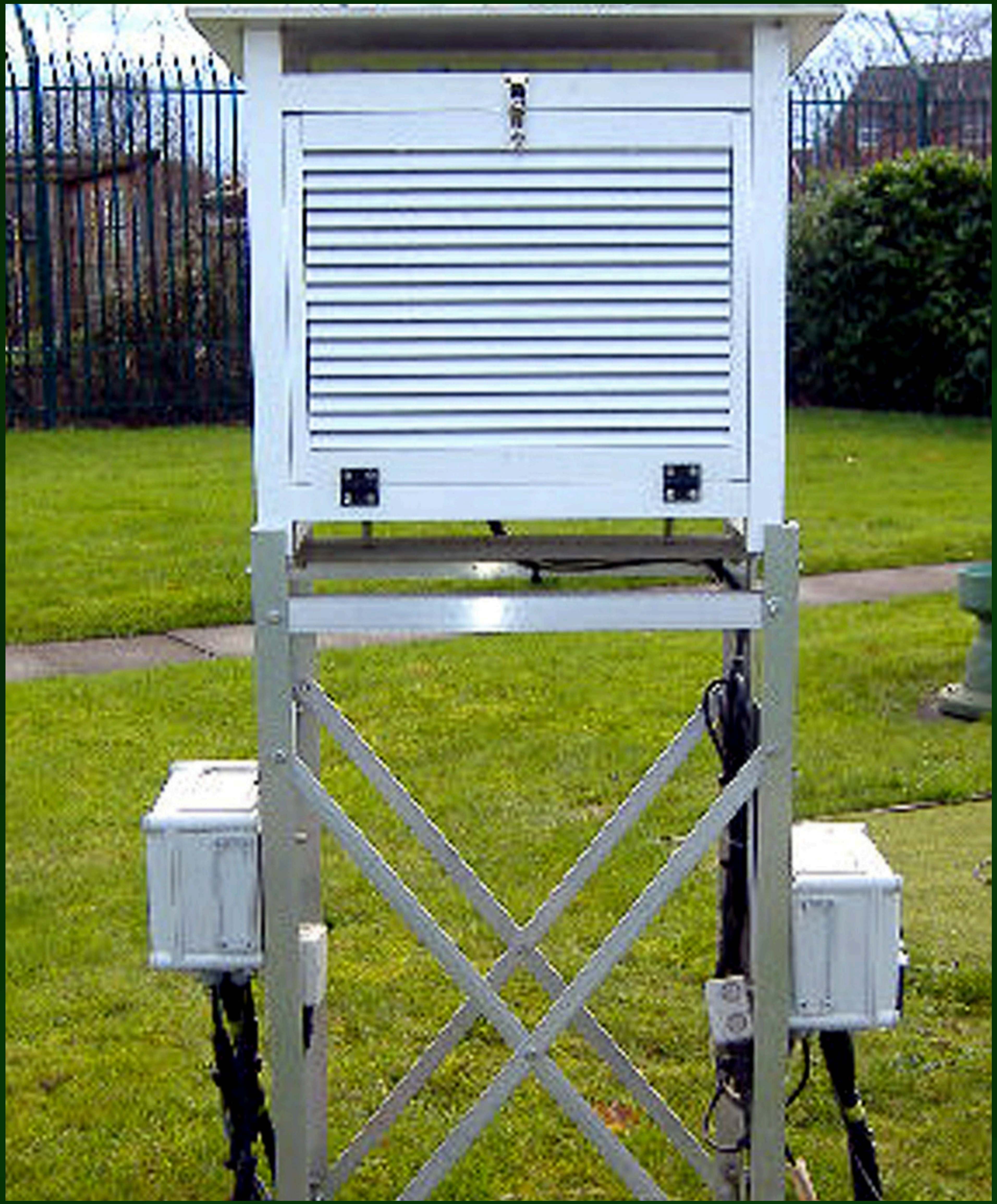


- Motivation: improving the accuracy of the location of personal weather stations for urban heat island research
- Generate potential locations
- Evaluate them through skyview + solar modelling



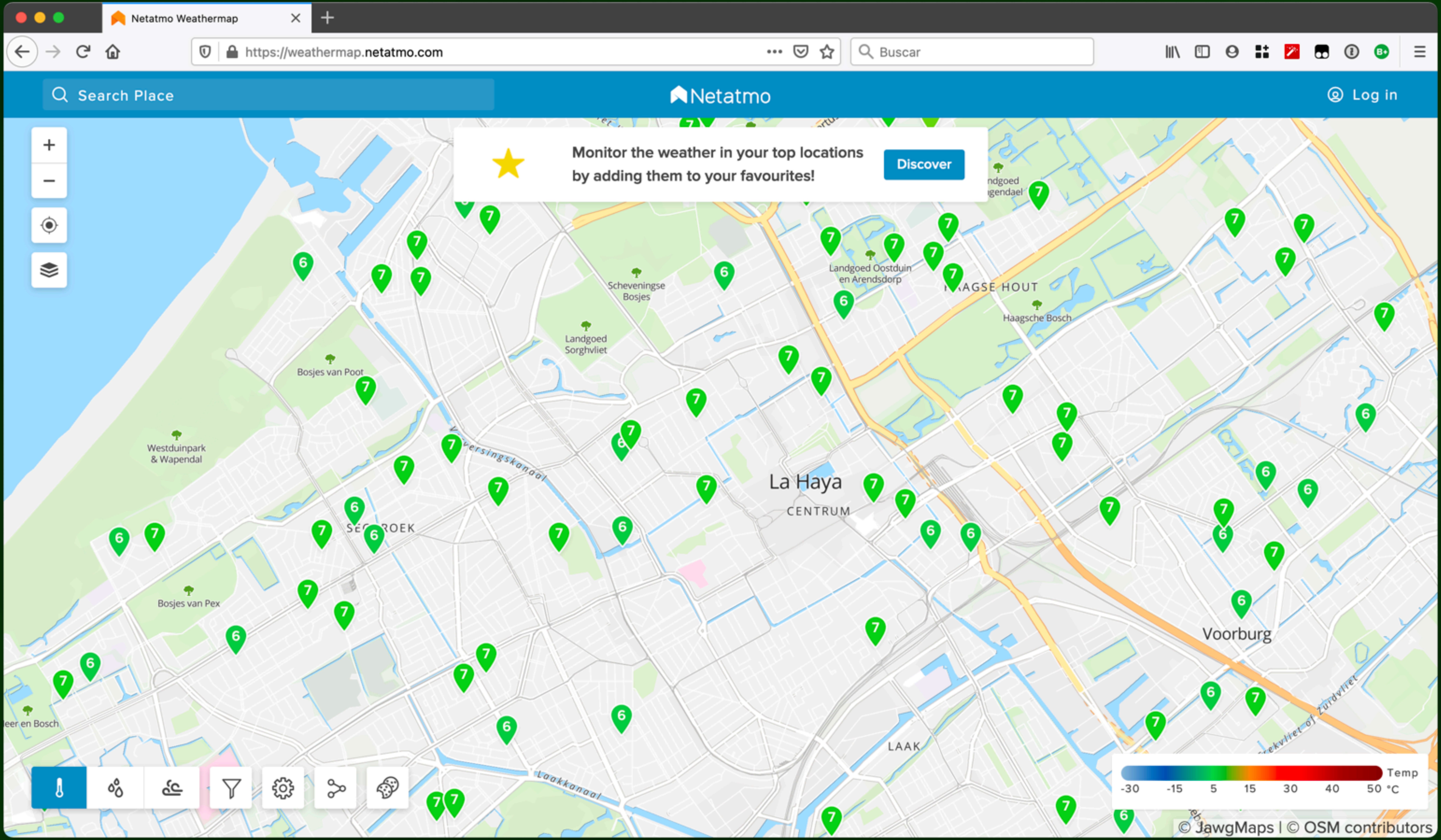
Urban heat island

Traditional weather stations

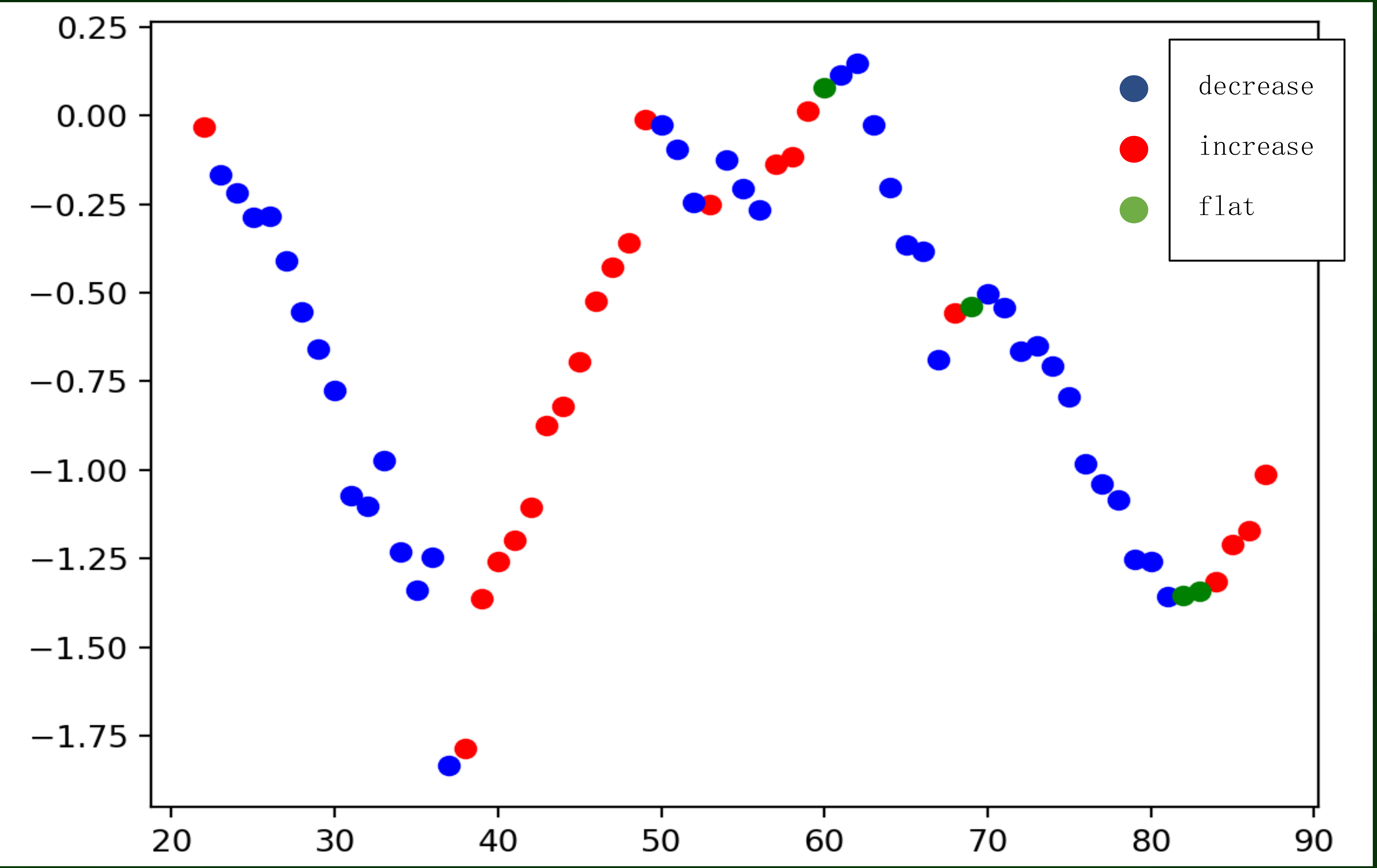


Personal weather stations

Crowdsourced weather data

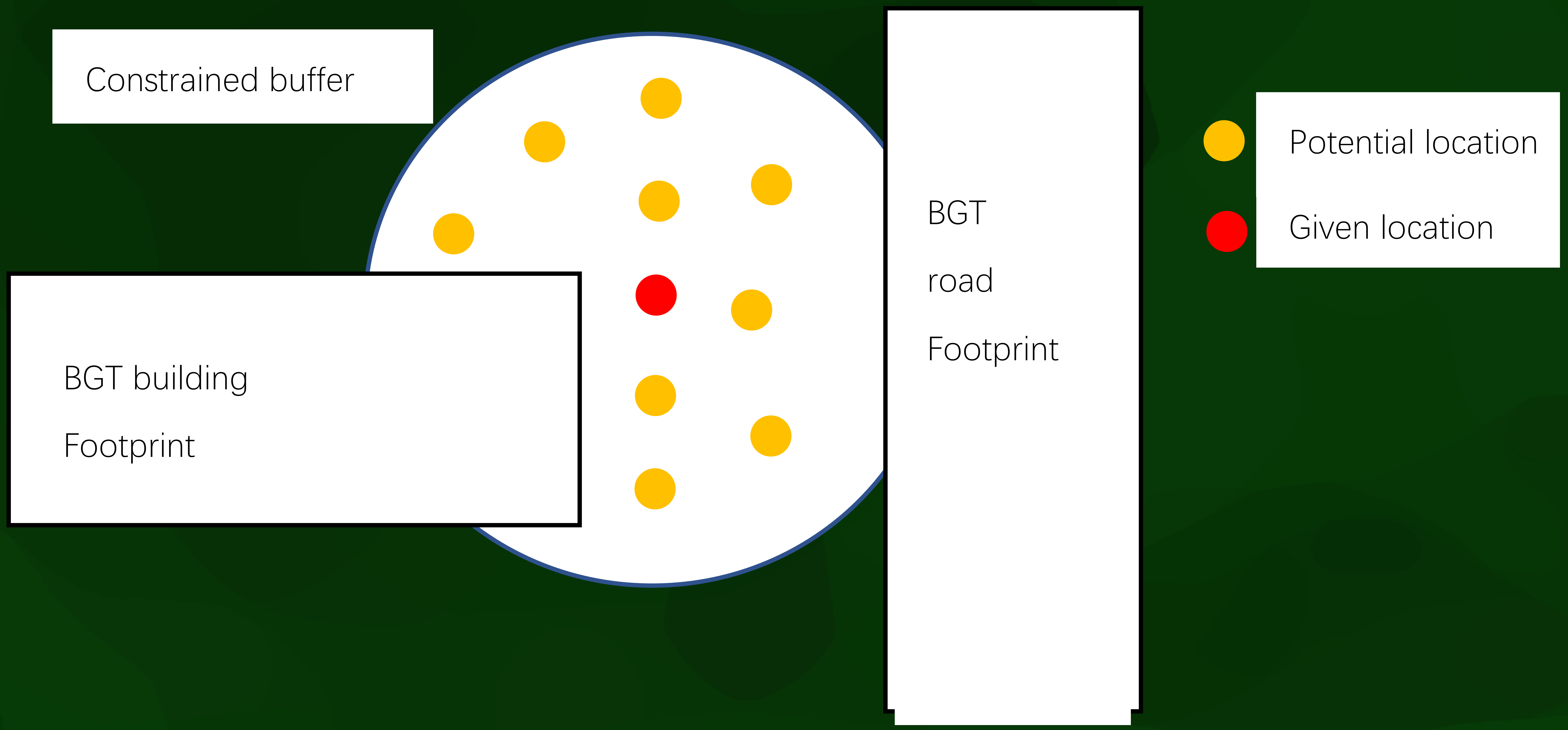


Behaviour

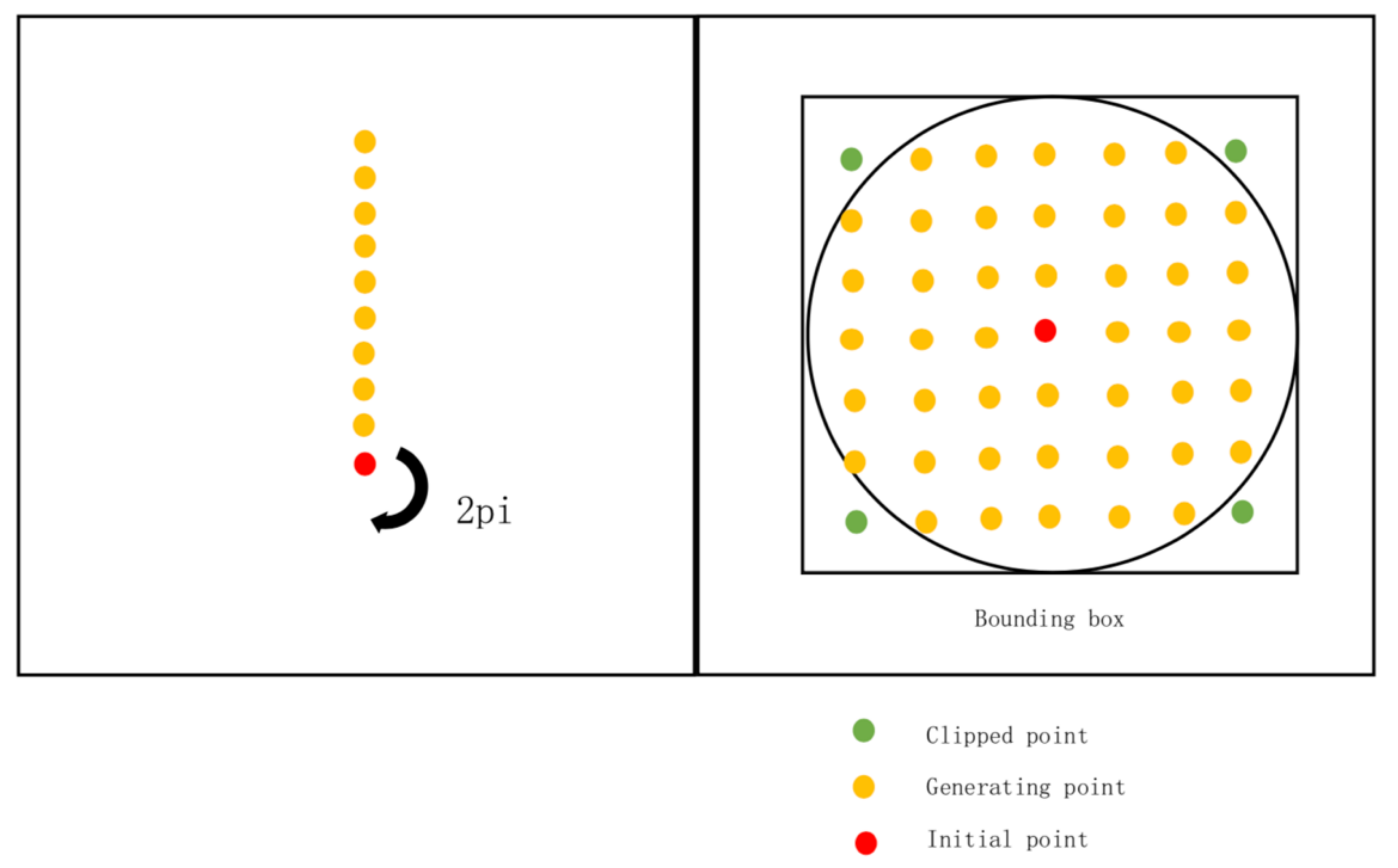


0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
20
30
40
50
60
70
80
90
decrease
increase
flat

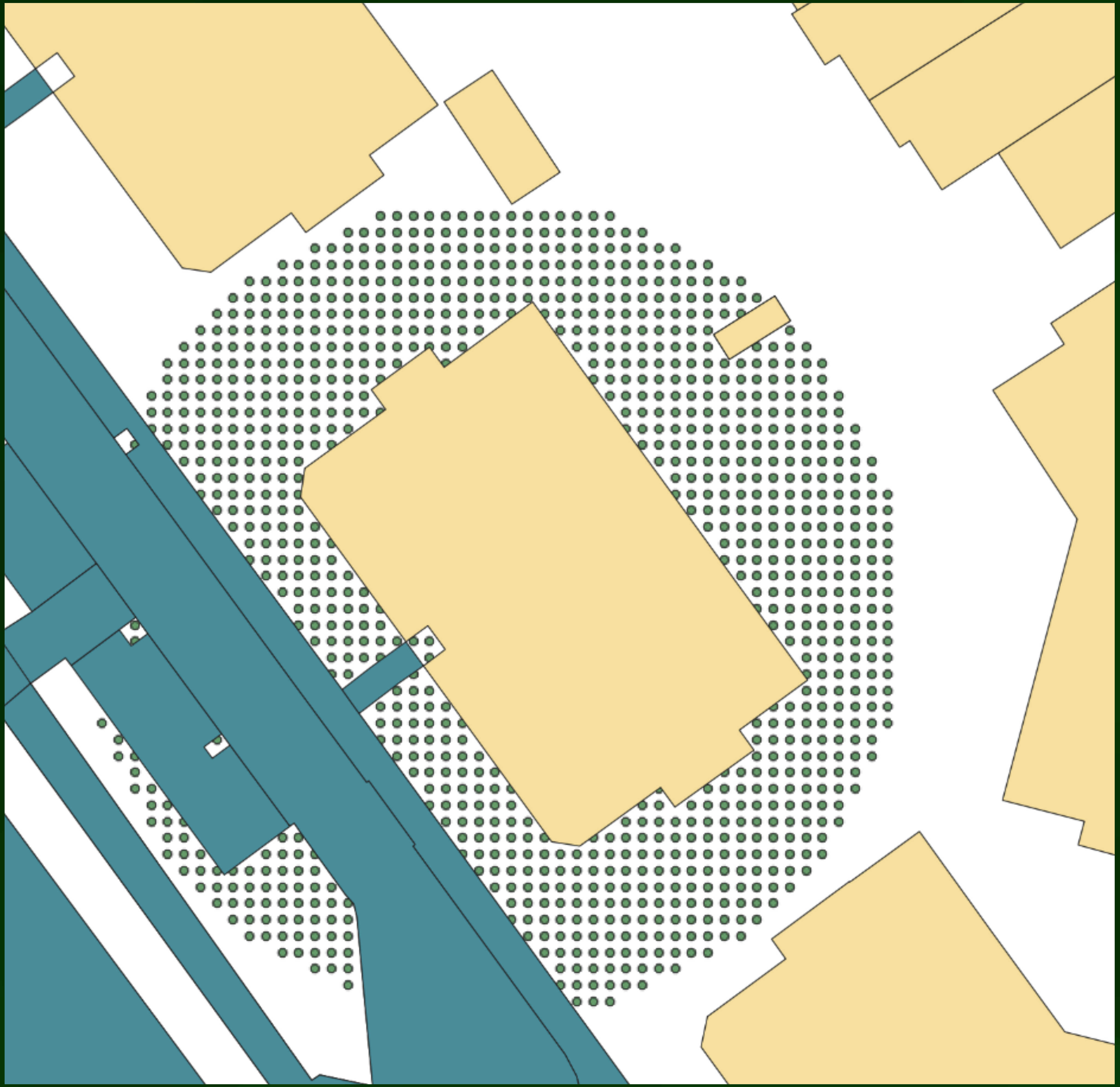
Potential locations



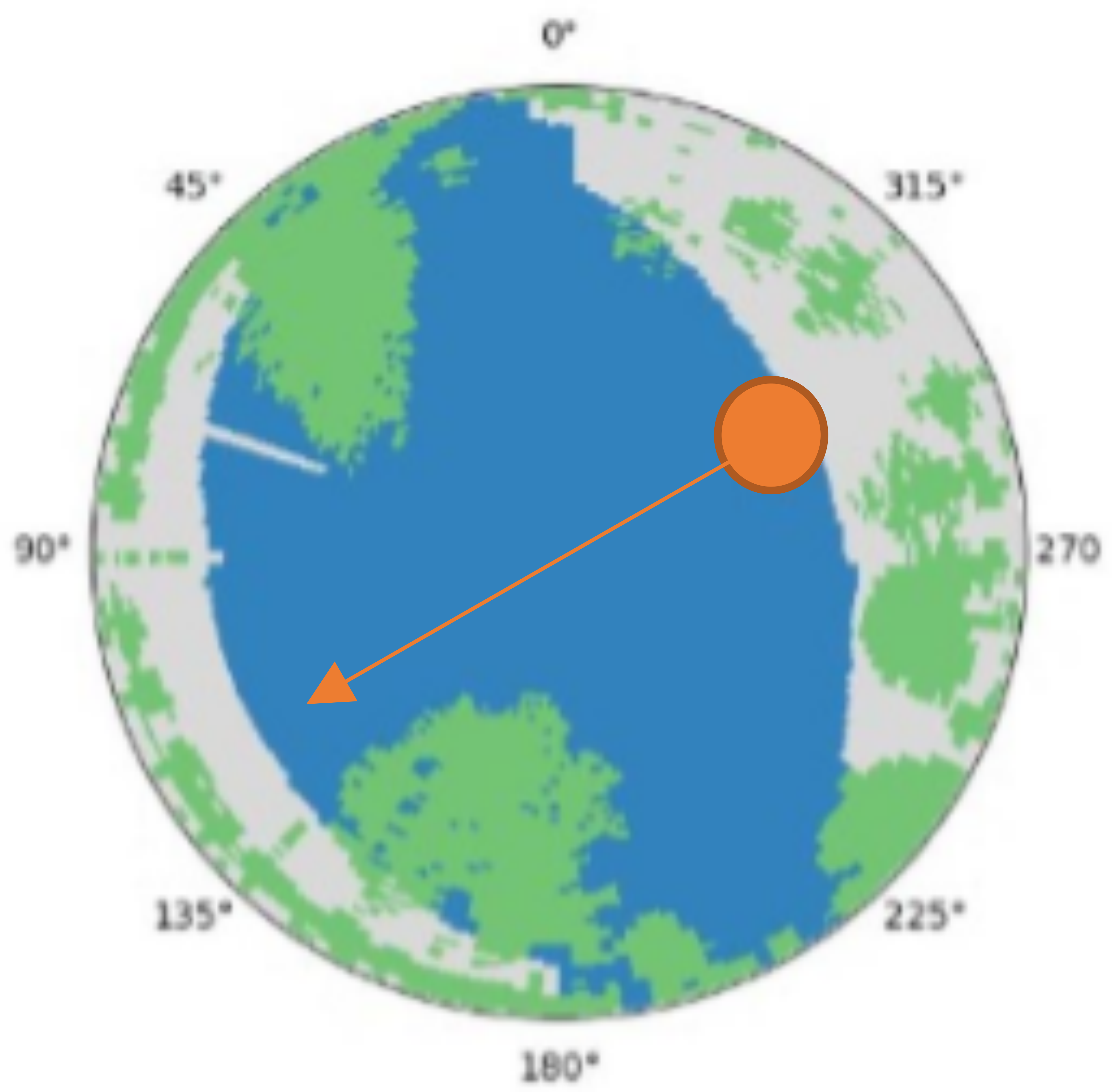
Potential locations



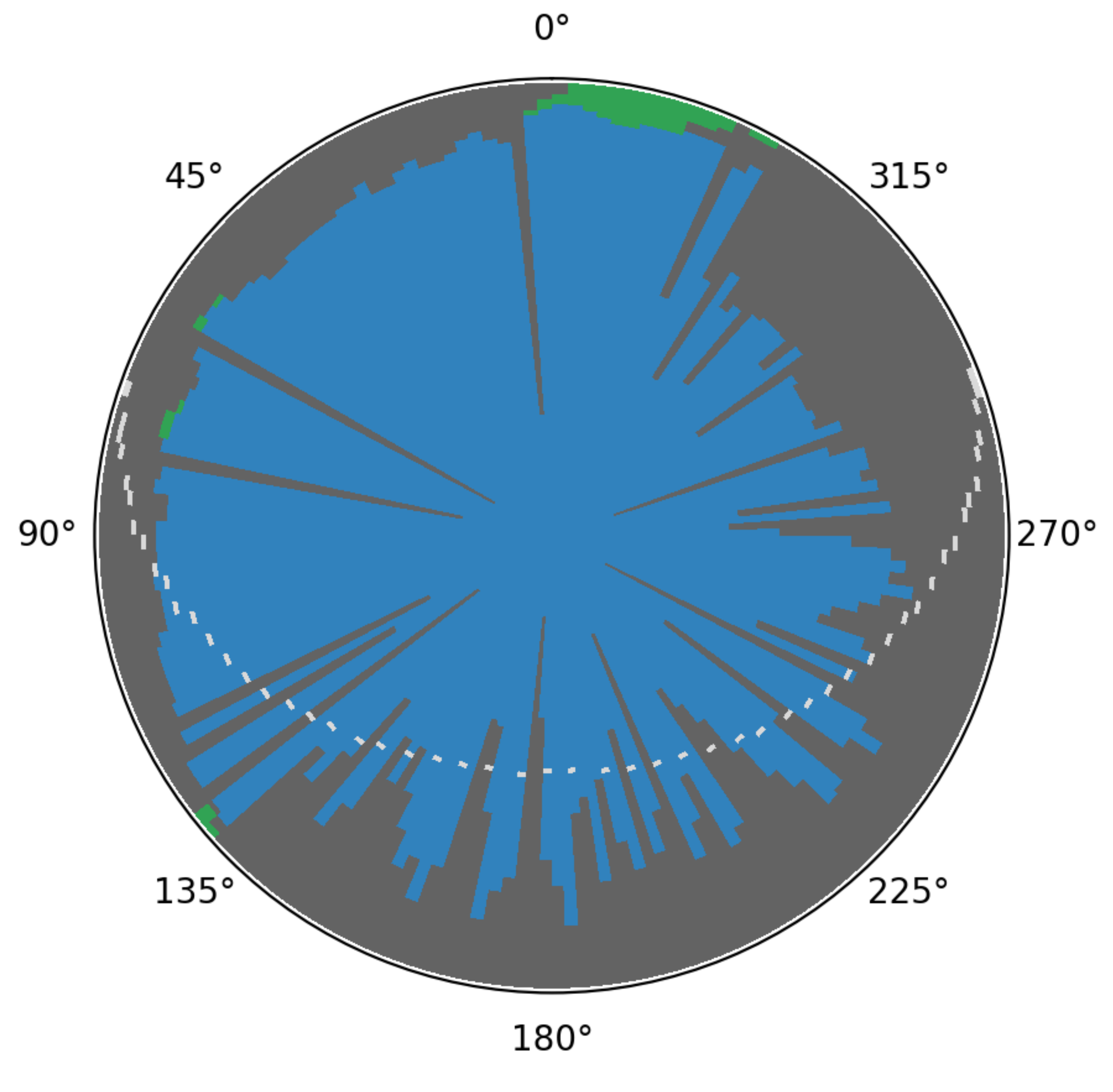
Potential locations



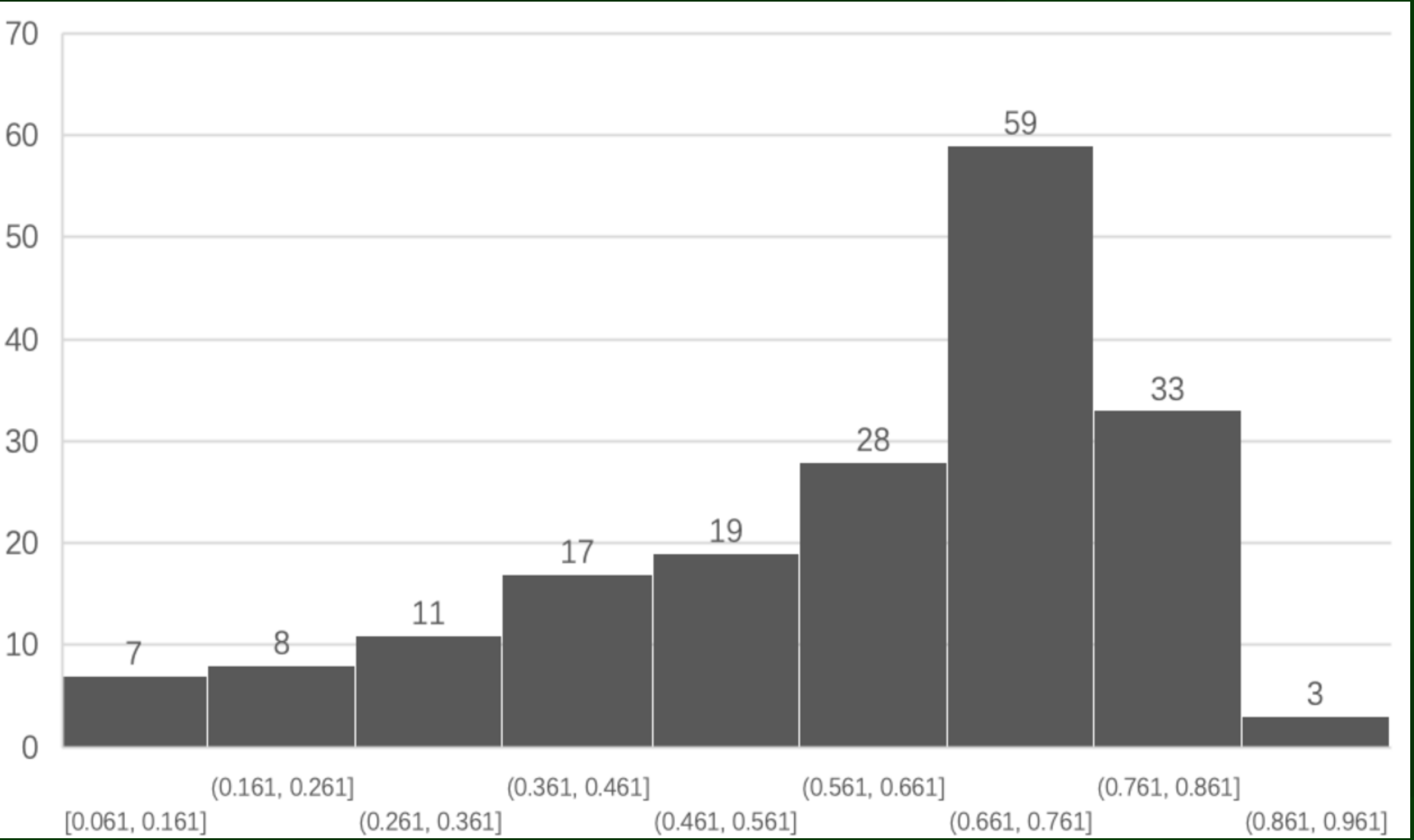
Skyview computation



Analysis



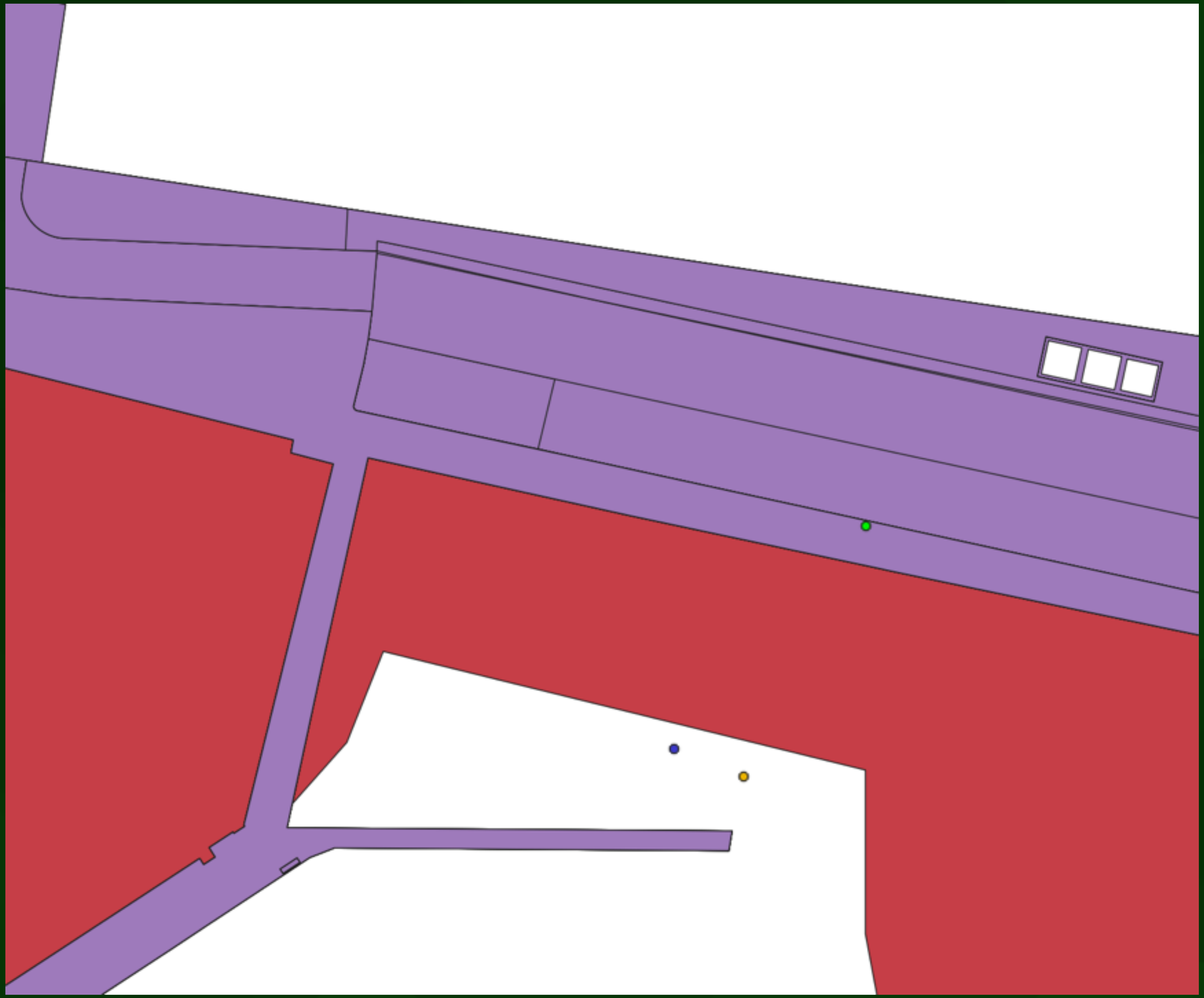
Results



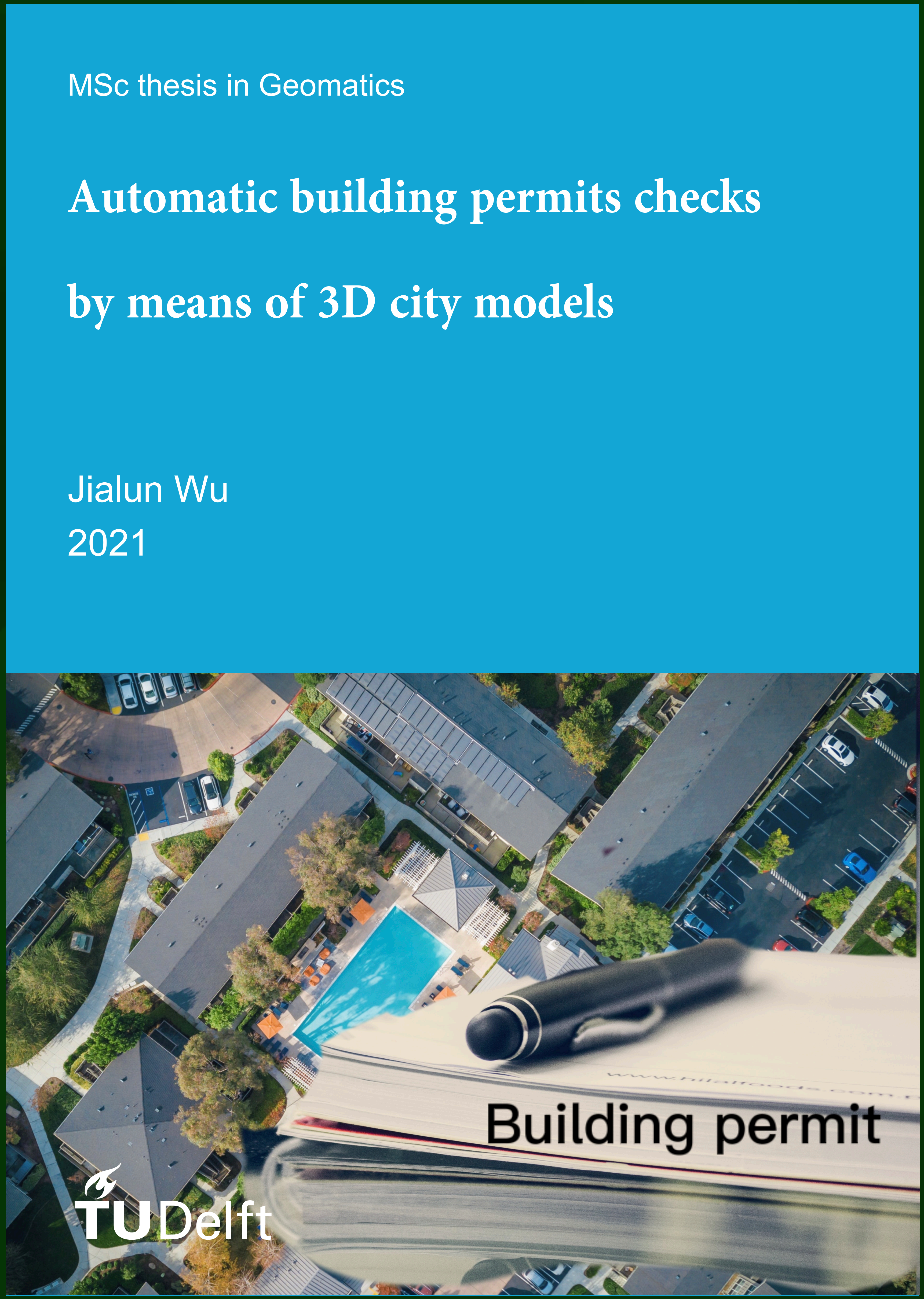
70
60
50
40
30
20
10
0
7
8
11
17
19
28
59
33
3
[0.061, 0.161]
(0.161, 0.261]
(0.261, 0.361]
(0.361, 0.461]
(0.461, 0.561]
(0.561, 0.661]
(0.661, 0.761]
(0.761, 0.861]
(0.861, 0.961]

Experiment

Experiment



- Motivation: automate some (simple) building permit checks using a 3DCM
- Formalisation of regulations
- Store necessary data in CityJSON extension
- Automate some checks (car + bicycle parking)



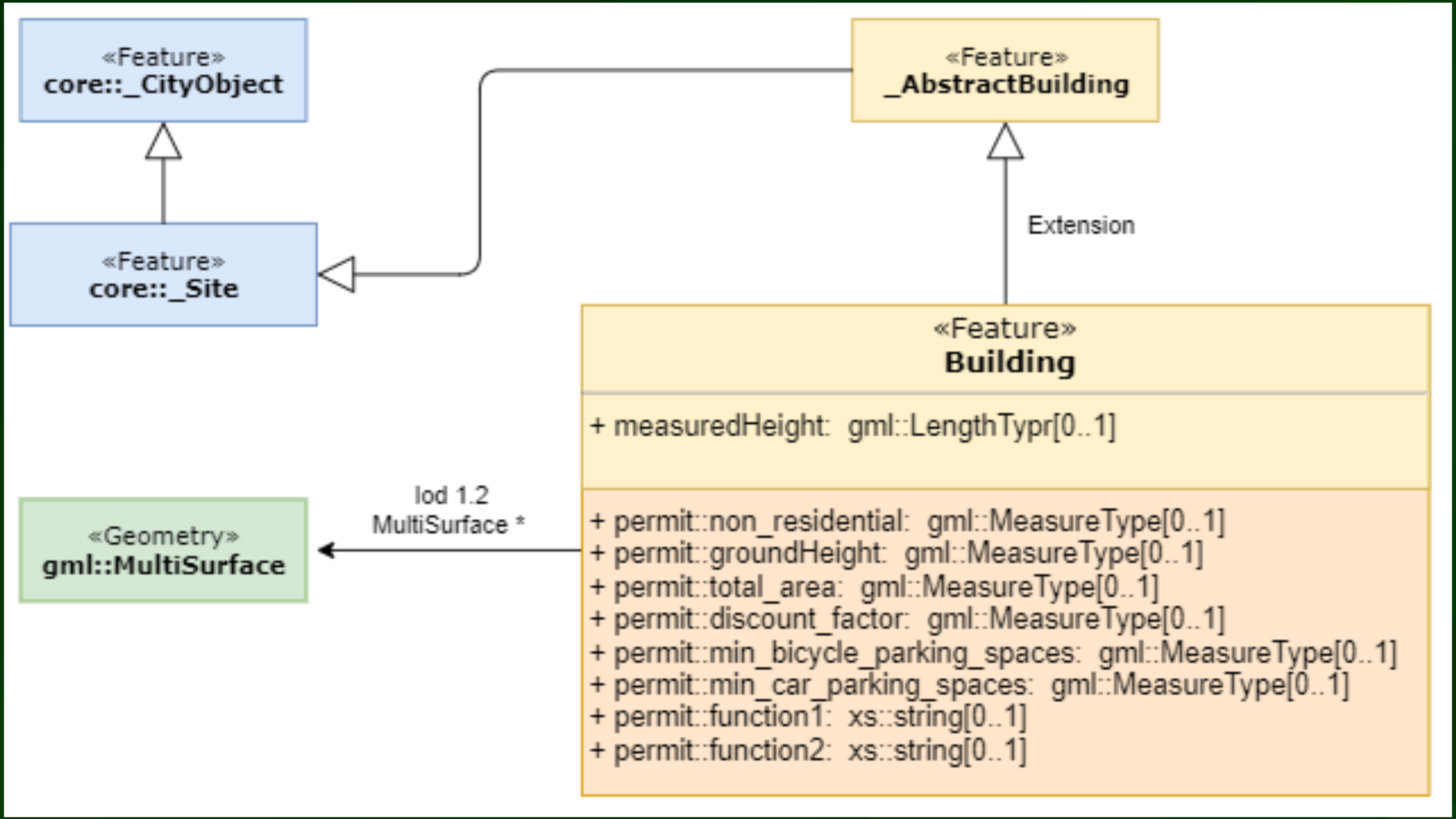
Formalisation of regulations

For residential buildings:
BUH40 = Count BU (function."home")
AND (A(BU) 40 m2)
BUH40-65 = Count BU (function."home")
AND (40 A(BU) 65 m2)
BUH65-85 = Count BU (function."home")
AND (65 A(BU) 85 m2)
BUH85 = Count BU (function."home")
AND (85 m2 A(BU))
Rules (must be true)
IF BU(function) = "home"THEN
MinNPP= (BUH40*2) + (BUH40-65*3)
+ (BUH65-85*4) + (BUH85*5)
NewParkings ≥ sum(MinNPP) + sum((MinMQPP/parkingArea))

New attributes to store

	Information	Explanation	Sources	name in sources
7*Attributes	id	Bag id of building	BAG	identificatie
	+function	Function of buildings included in codelist	BAG	NR_XXX (different functions)
	+groundHeight	Elevation above sea level at the ground level	3D BAG	ground-0.00
	measuredHeight	Elevation above sea level at rooflevel	3D BAG	roof-0.75
	+zone	Zone where the building is located	Digital map	zone
	+height_valid	Indicate the height is valid	3D BAG	height_valid
	+total_area	Gross floor area (GFA) of building	BAG	Calculation results on different attributs
2*Geometry	type	geometry type of buildings	BAG	type
	coordinates	a lists contain [x,y,z] 3D coordinates	BAG	coordinates

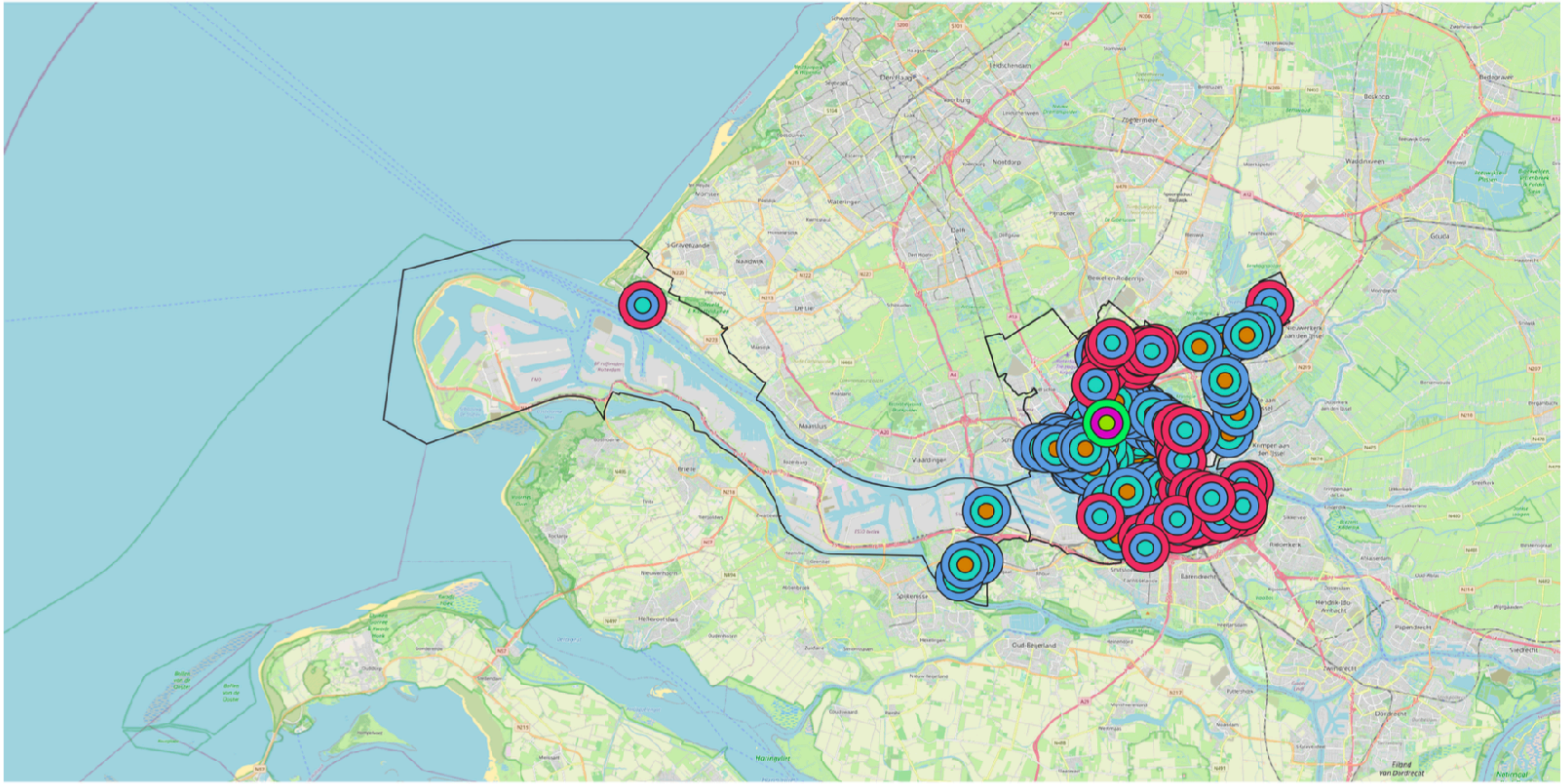
CityJSON extension



CityJSON extension

```
"68": {
  "type": "Building",
  "toplevel": true,
  "attributes": {
    "+height_valid": 1,
    "+non_residential": 1,
    "+groundHeight": 0,
    "measuredHeight": 28.0,
    "+total_area": 1371.5687999999998,
    "+discount_factor": 0.95,
    "+min_bicycle_parking_spaces": 117,
    "+min_car_parking_spaces": 78,
    "+function": "catering I"
```

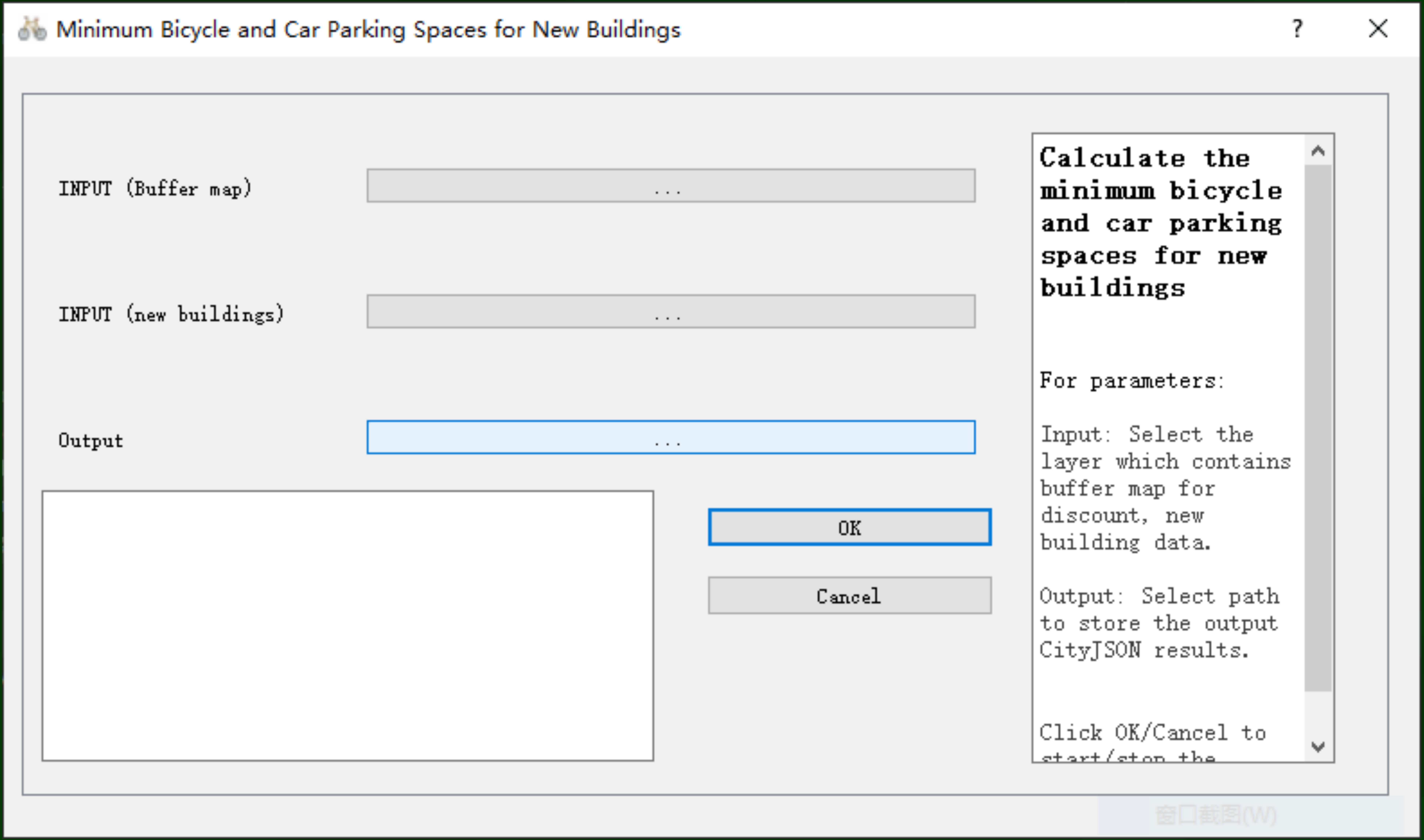
Generating required info



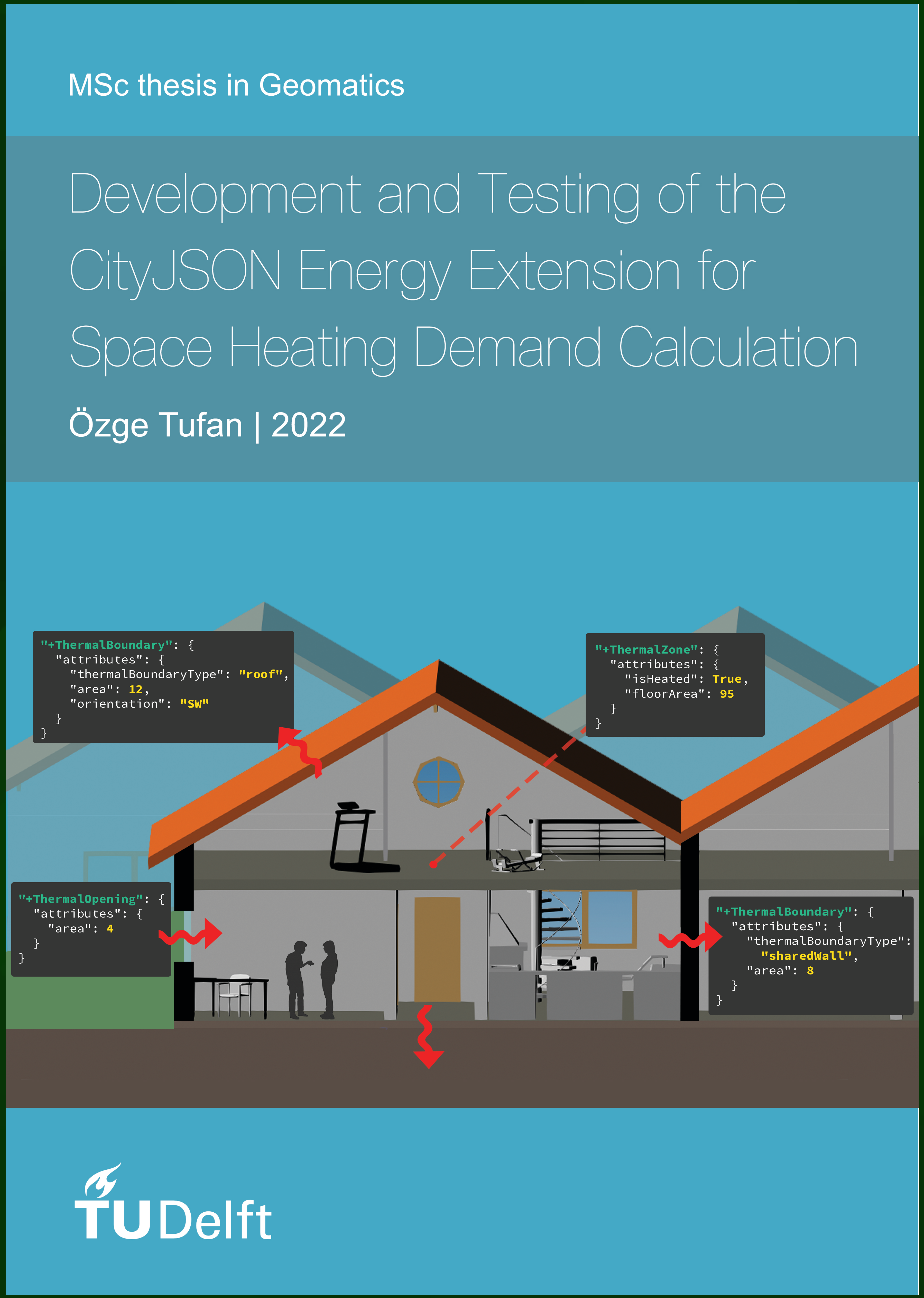
Programming checks

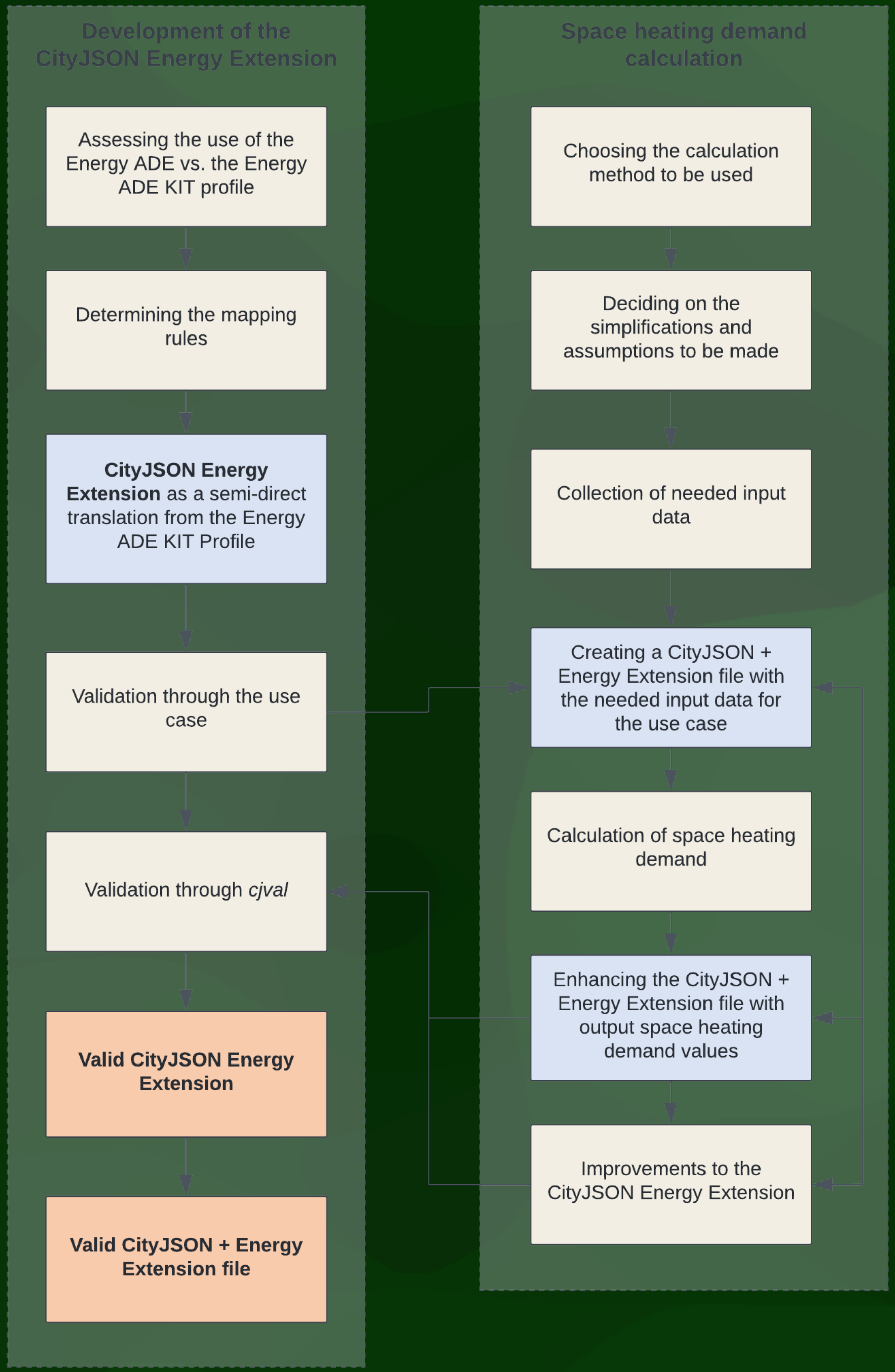
```
N_40 = int(f['properties']['N_40'])
N_40_65 = int(f['properties']['N_40_65'])
N_65_85 = int(f['properties']['N_65_85'])
N_85_120 = int(f['properties']['N_85_120'])
N_120 = int(f['properties']['N_120'])
if f['properties']['zone'] == 'A':
    oneb['attributes']['+min_car_parking_spaces'] = int(
    N_40 * 0.1 + N_40_65 * 0.4 + N_65_85 * 0.6 + N_85_120 * 1 +
    N_120 * 1.2)
if f['properties']['zone'] == 'B':
    oneb['attributes']['+min_car_parking_spaces'] = int(
    N_40 * 0.1 + N_40_65 * 0.5 + N_65_85 * 0.8 + N_85_120 * 1 +
    N_120 * 1.2)
if f['properties']['zone'] == 'C':
    oneb['attributes']['+min_car_parking_spaces'] = int(
    N_40 * 0.1 + N_40_65 * 0.6 + N_65_85 * 1.4 + N_85_120 * 1.6
    + N_120 * 1.8)
```

Results: tool



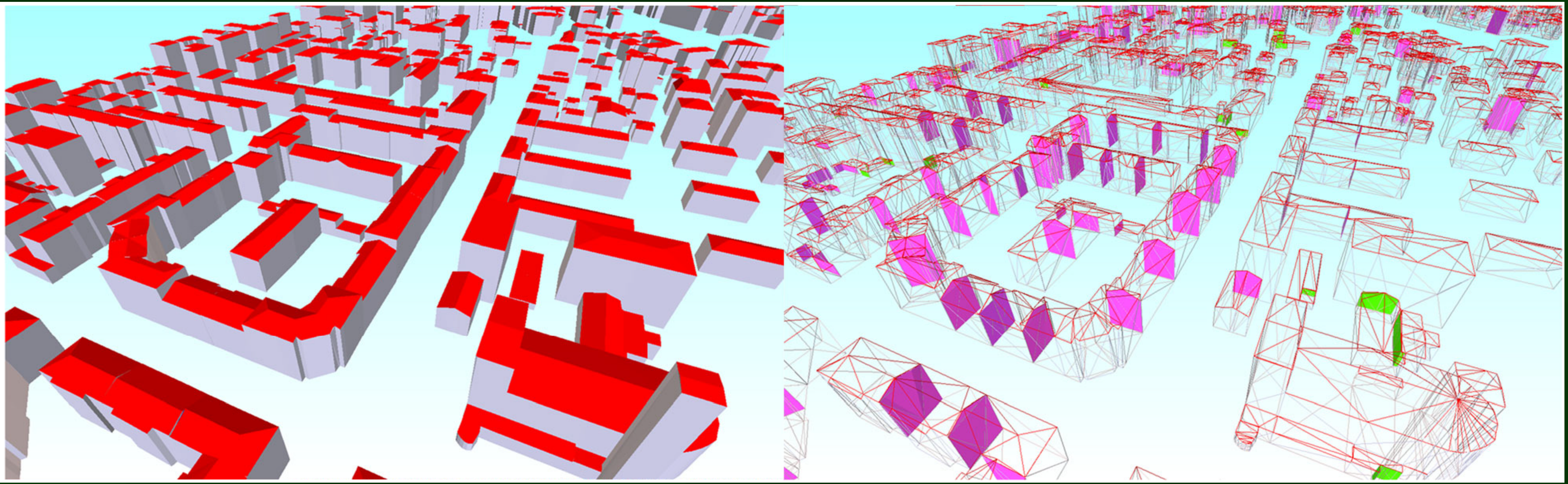
- Motivation: use 3DCM for space heating demand calculations
- Develop CityJSON extension with all required information
- Implement space heating models
- Use implementation to improve extension design





Development of the CityJSON Energy Extension
Assessing the use of the Energy ADE vs. the Energy ADE KIT profile
Determining the mapping rules
CityJSON Energy Extension as a semi-direct translation from the Energy ADE KIT Profile
Validation through the use case
Validation through cjval
Valid CityJSON Energy Extension
Valid CityJSON + Energy Extension file
Space heating demand calculation
Choosing the calculation method to be used
Deciding on the simplifications and assumptions to be made
Collection of needed input data
Creating a CityJSON + Energy Extension file with the needed input data for the use case
Calculation of space heating demand
Enhancing the CityJSON + Energy Extension file with output space heating demand values
Improvements to the CityJSON Energy Extension

Storing new (complex) geometries



New attributes

Net internal area	Exludes internal structural elements
Class	Type of use of the building, e.g. residential, mixed-use
Function*	Further description of the class, e.g., health, business
Usage	Whether the building is still in use
Measured height	Height of the building, in *m*
Relative to terrain	Whether the building is (entirely) above or below the terrain
Roof type	E.g. slanted, single/multiple horizontal
Year of construction	Construction year of the building
Footprint area*	Footprint area, calculated from the LoD0 geometry, in m^2
Storeys above ground*	Number of storeys situated above ground level
Storeys below ground*	Number of storeys situated below ground level
Building name*	Unique name of the building
Is single part	Boolean value to show whether the building has *BuildingParts*
# of adjacent buildings	Number of topologically adjacent buildings
LoD2 volume*	Building volume, calculated from the LoD2 geometry, in m^3
LoD max	Maximum LoD present for the building
Building (pand) ID	Unique ID of the building
List adjacent buildings	Building (pand) ID of topologically adjacent buildings
Surface ID	Unique ID of the *BoundarySurface*
Parent building ID	Building (pand) ID of the building that the surface belongs to
Surface name	Unique name of the *BoundarySurface*
Azimuth	Azimuth of the surface, in *degrees*
Inclination	Inclination of the surface, in *degrees*
Direction	Direction of the surface
LoD2 area	Surface area, calculated from the LoD2 geometry, in m^2
Surface normal	Normal vector of the surface

New attributes

```
"extraAttributes": {
  "Building": {
    "+buildingType": {...},
    "+constructionWeight": {...},
    "+volume": {...},
    "+floorArea": {...},
    "+heightAboveGround": {...}
  }
}
```

```
"Build1": {
  "type": "Building",
  "geometry": [...],
  "attributes": {
    "+buildingType": "singleFamily",
    "+constructionWeight": "heavy",
  }
}
```

New City Objects

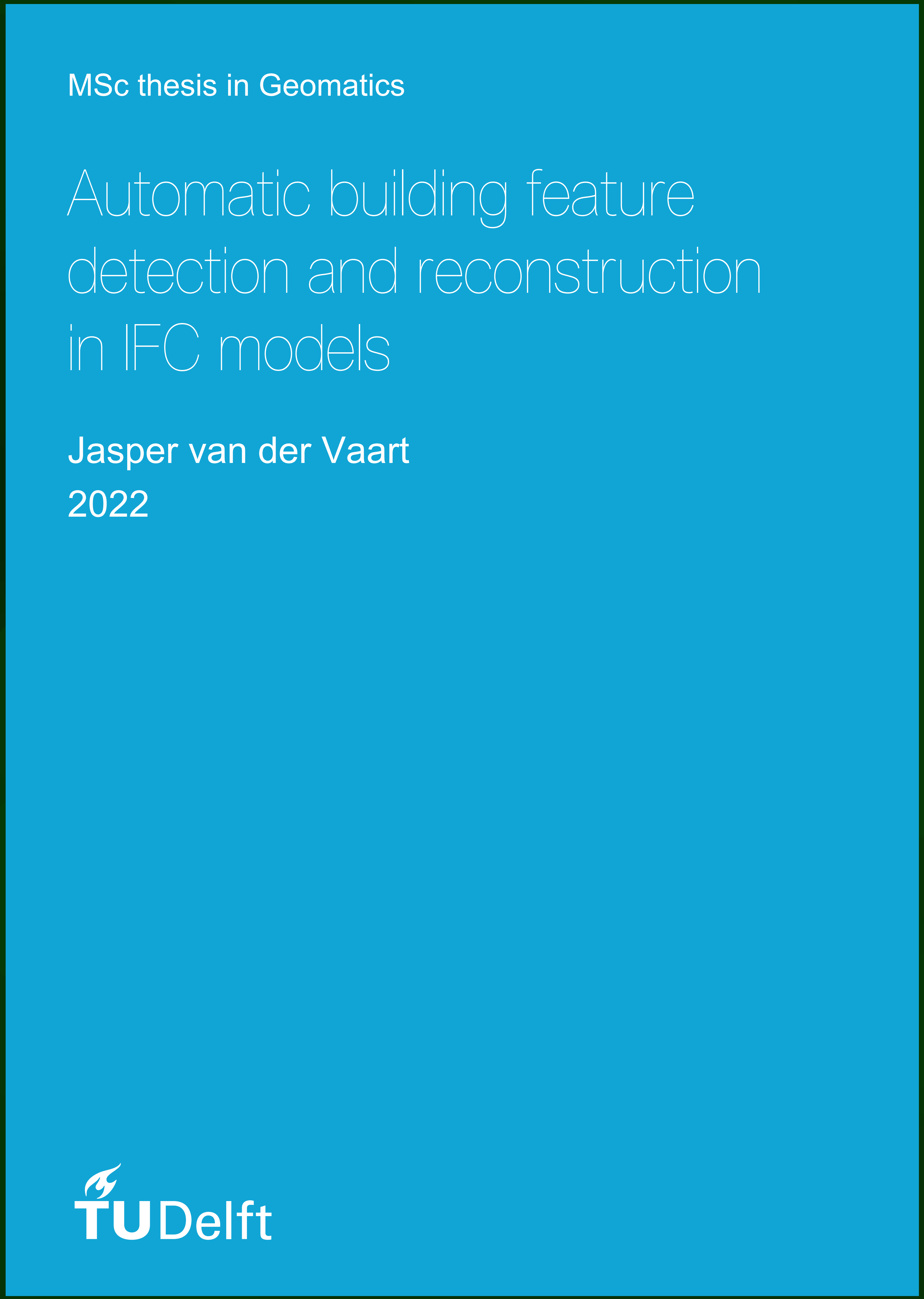
```
"extraCityObjects": {
  "+WeatherData": {
    "type": "object",
    "properties": {
      "type": {...},
      "attributes": {
        "type": "object",
        "properties": {
          "weatherDataType": {...},
          "values": {...},
          "position": {...}
        }
    ...
}
```

```
"OutdoorTemperature": {
  "type": "+WeatherData",
  "attributes": {
    "weatherDataType": "airTemperature",
    "values": "RegularTimeSeries1",
              //ID of TimeSeries object
  }
},
"RegularTimeSeries1": {
  "type": "+RegularTimeSeries",
  "attributes": {
    "values": [2.61, 4.82, 5.91, 9.32,
       14.73, 16.12],
    ...
  }
}
```

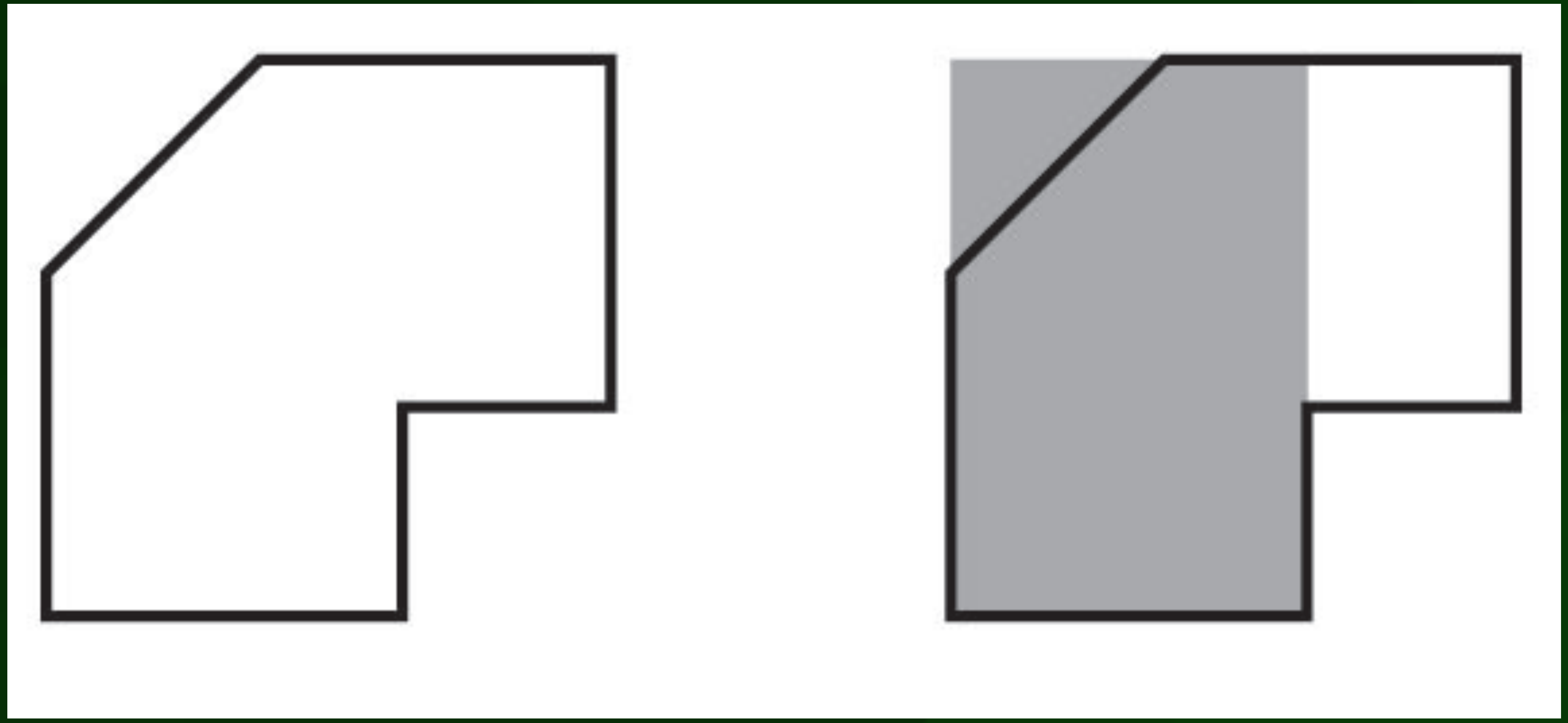
Test data

Results: heating energy demand

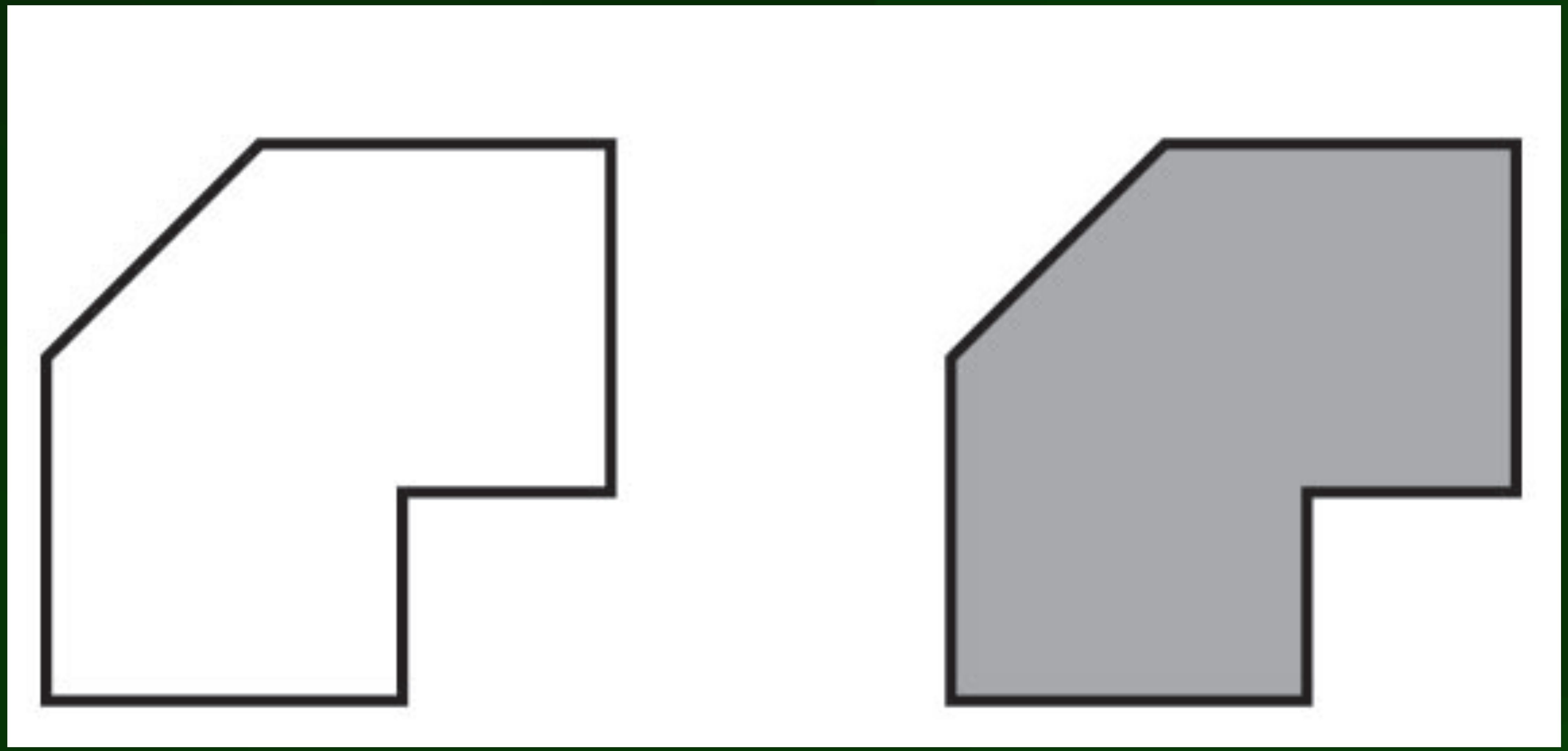
- Motivation: unreliable or non-existent information in IFC models
- Automatically create shapes of rooms, storeys and apartments
- Built on IfcOpenShell



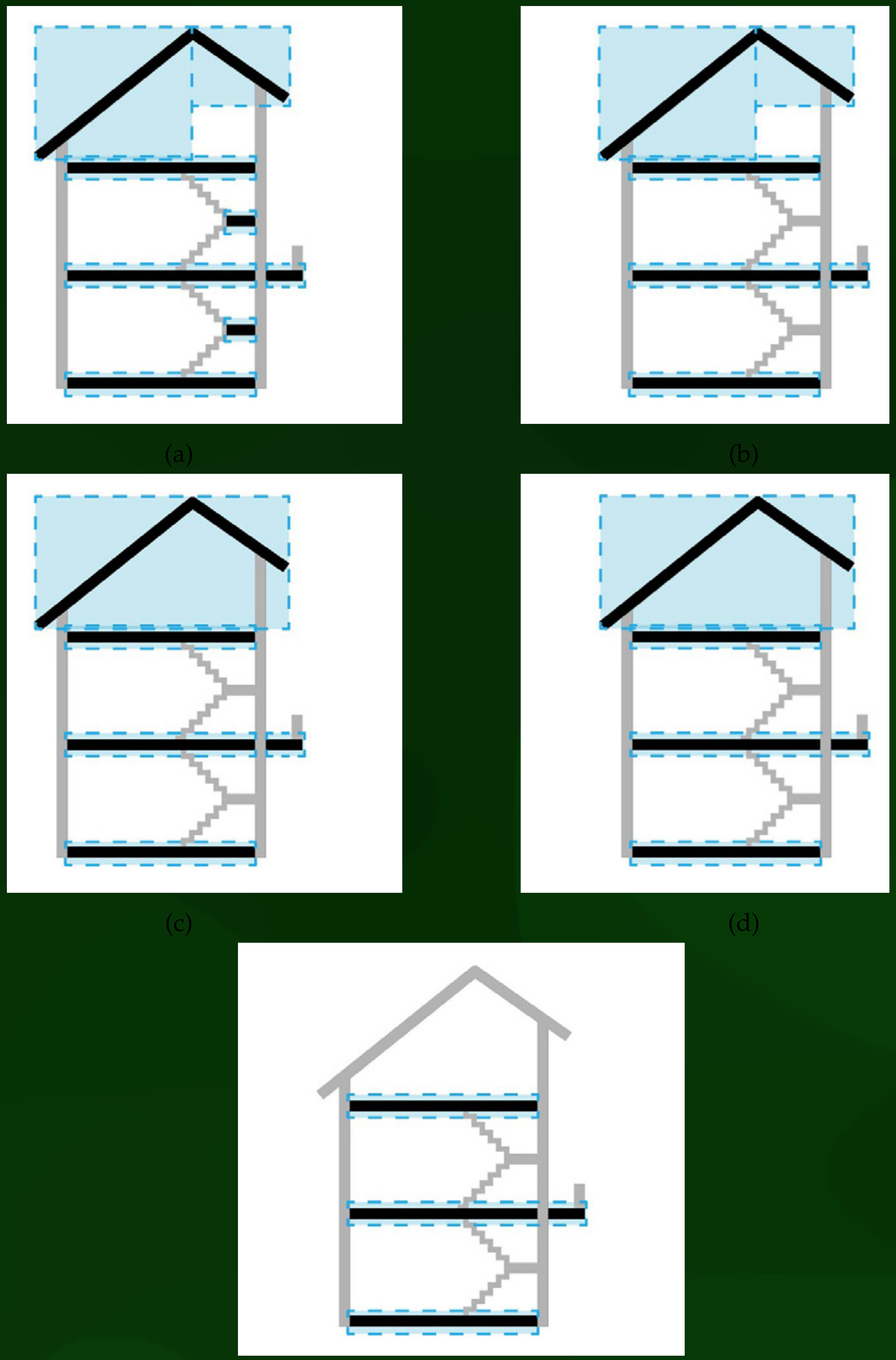
Current IfcSpaces



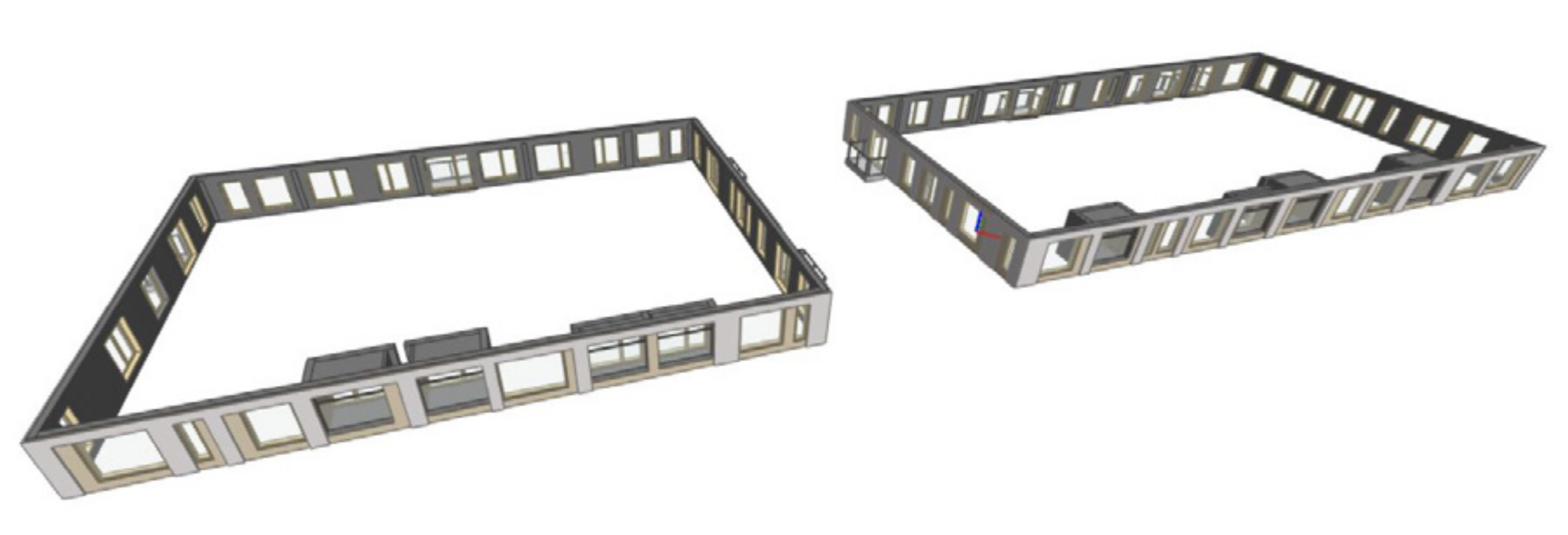
(a)



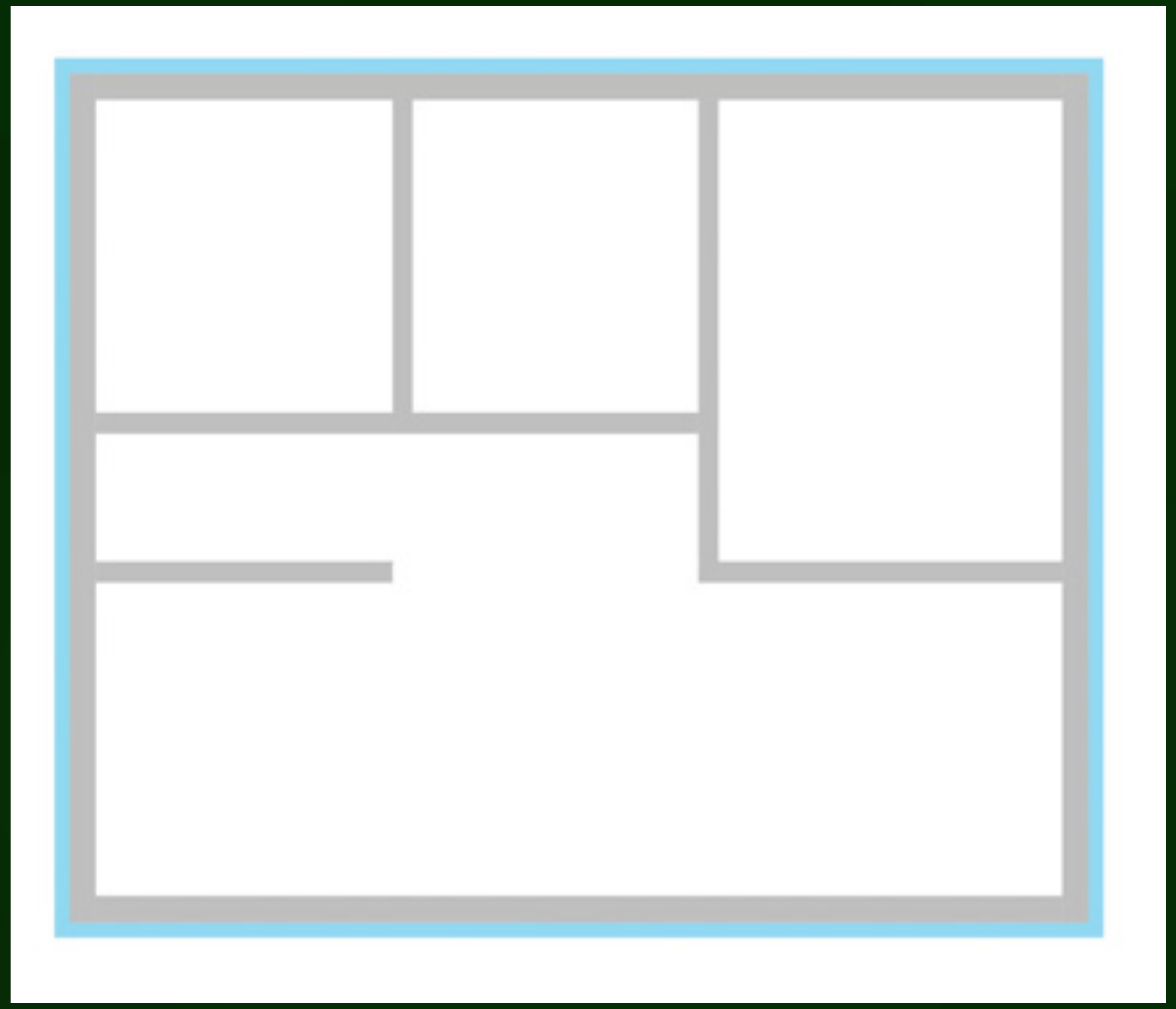
Computing storeys



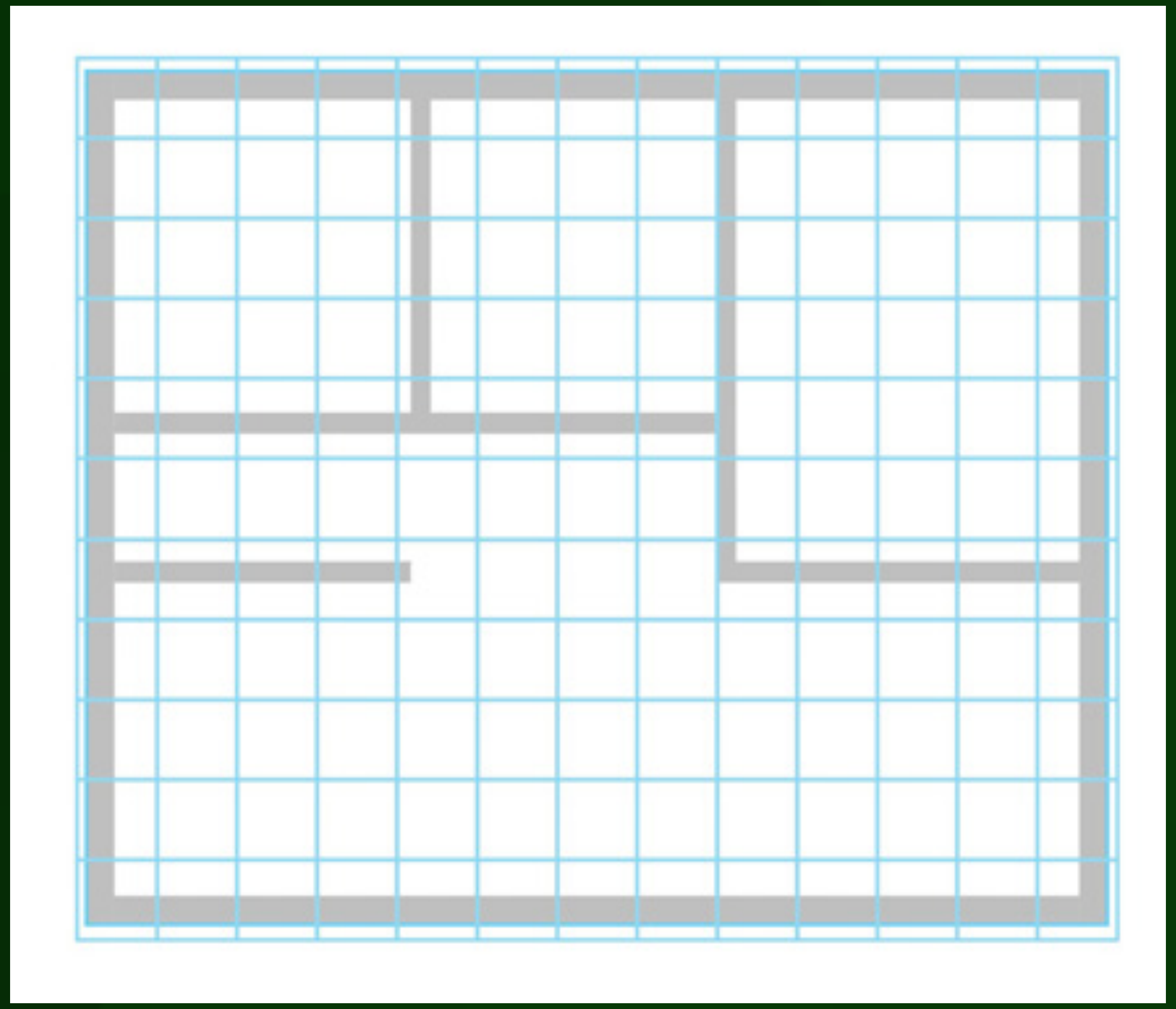
Computing storeys



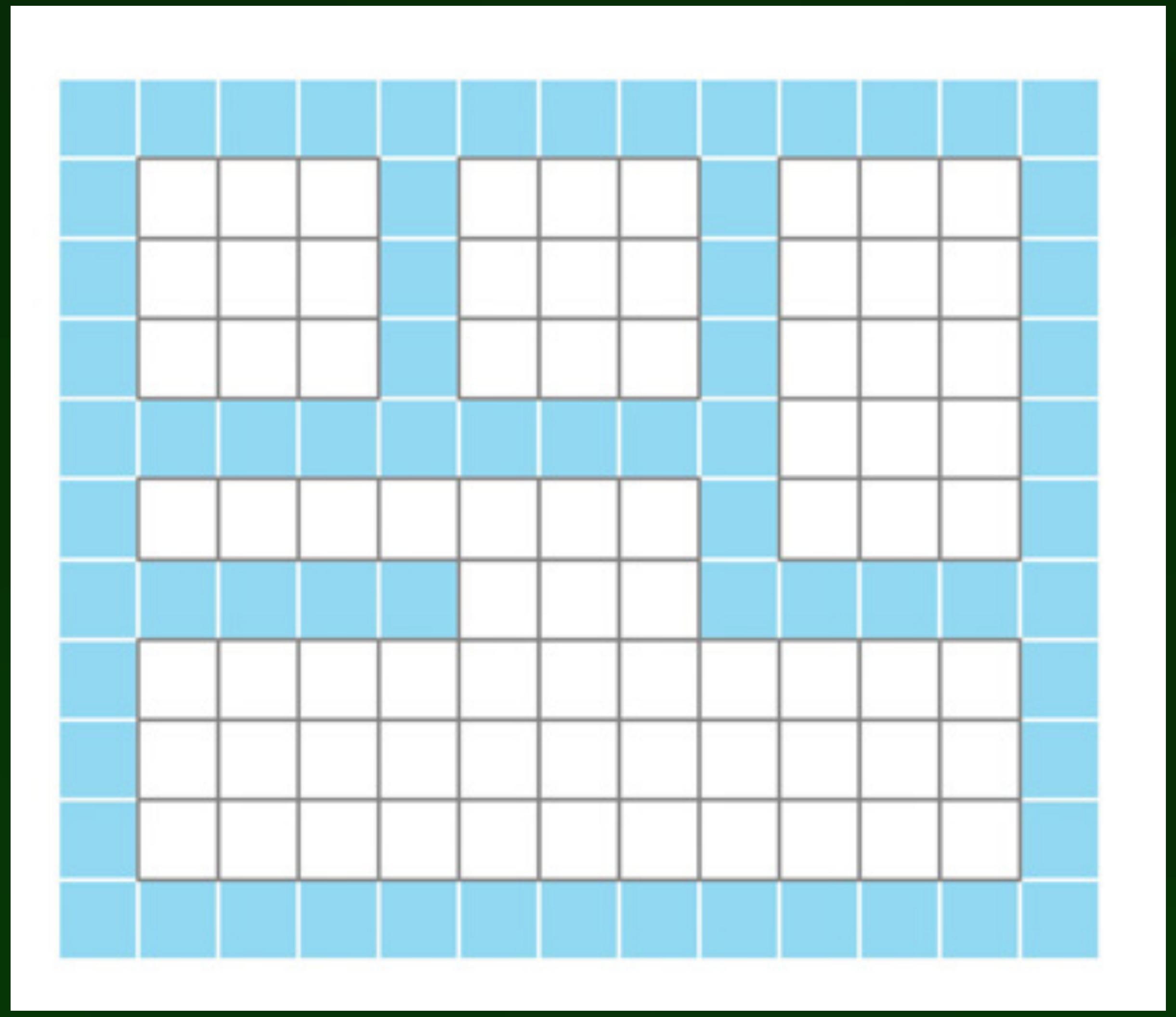
Computing rooms

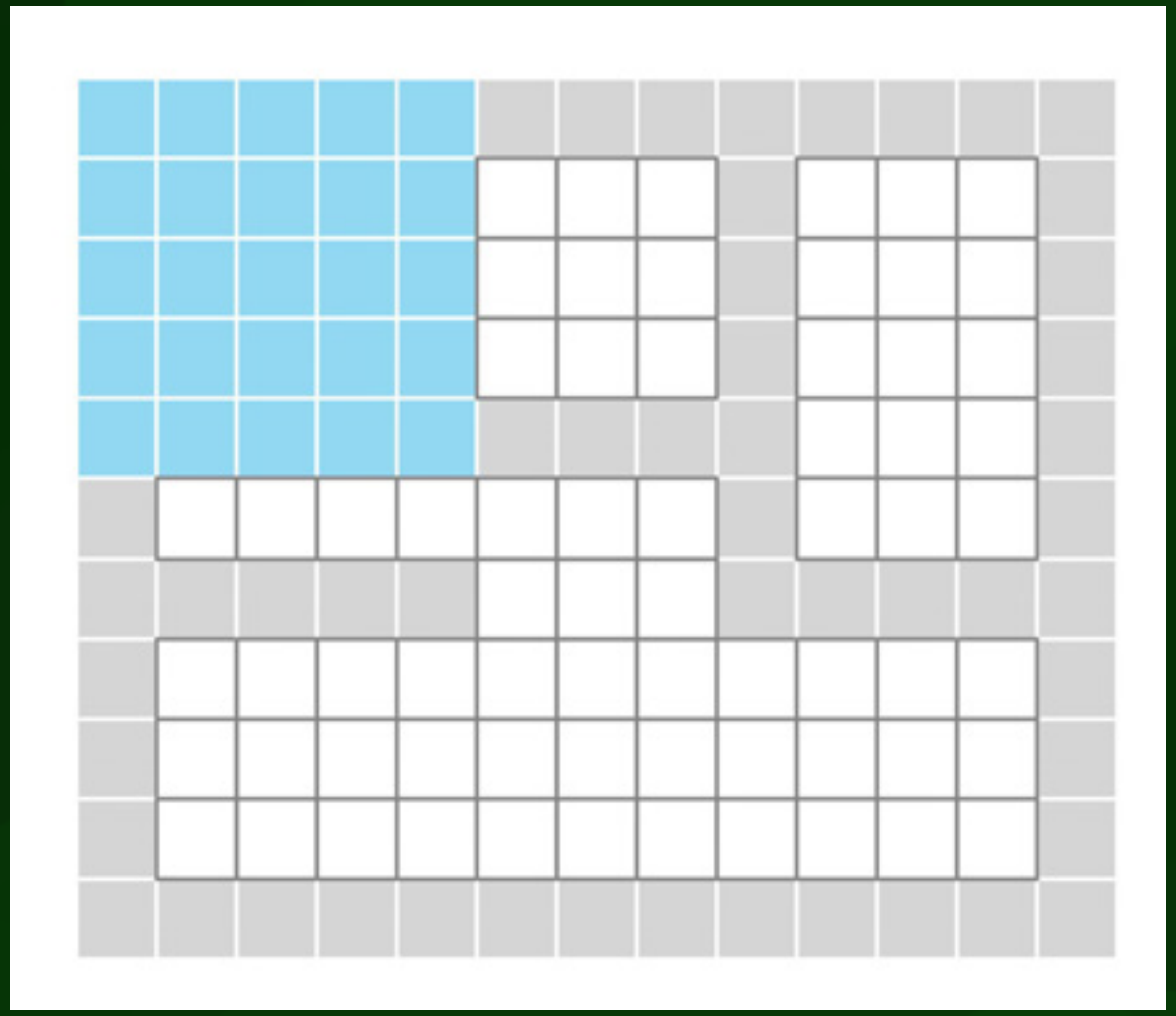


(c)

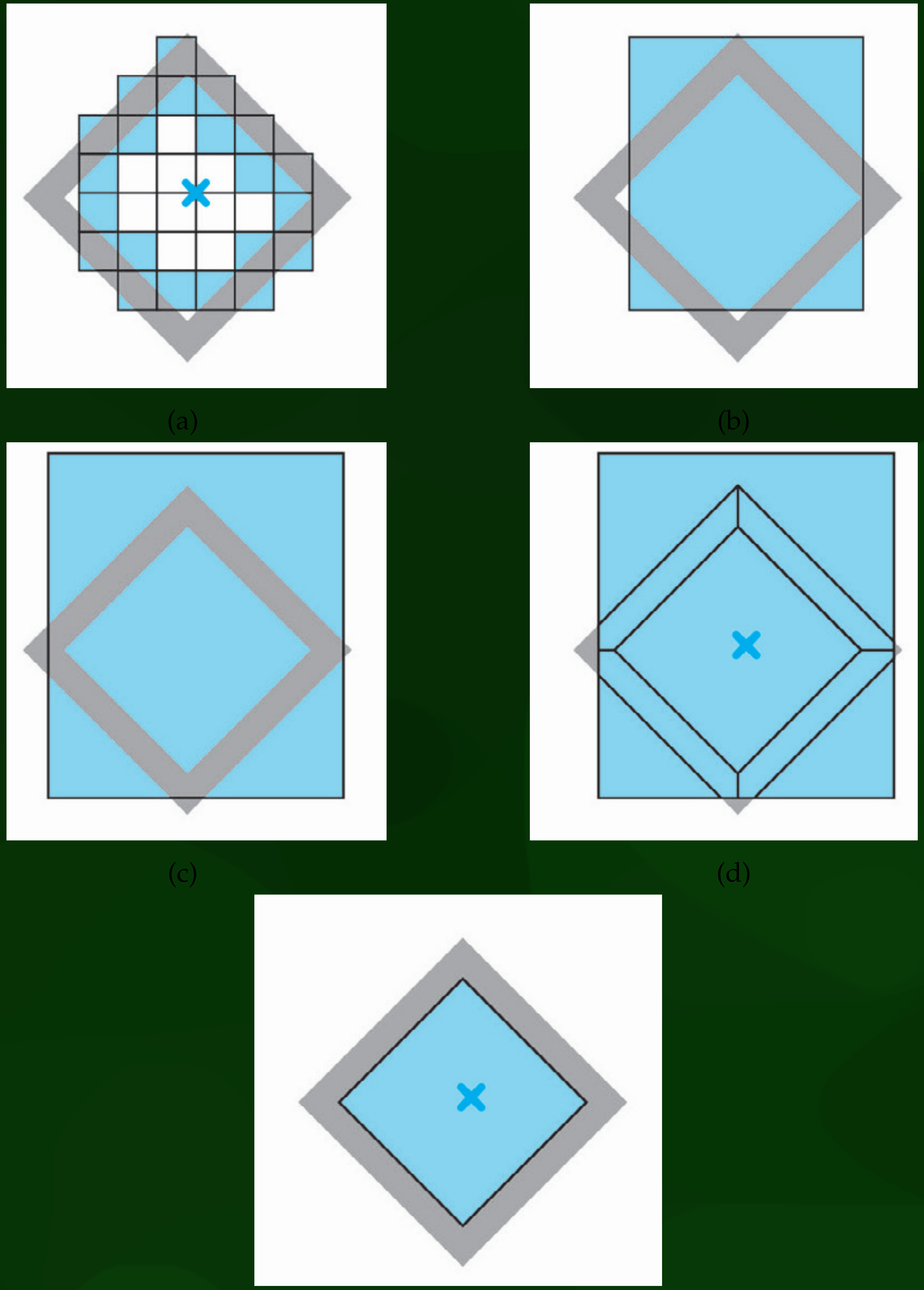


(d)

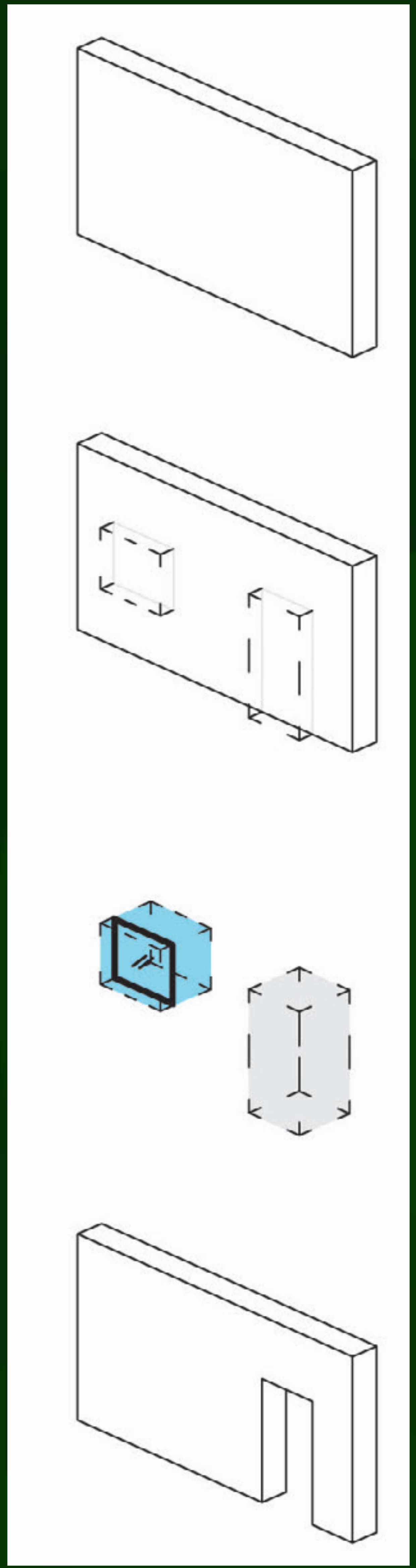




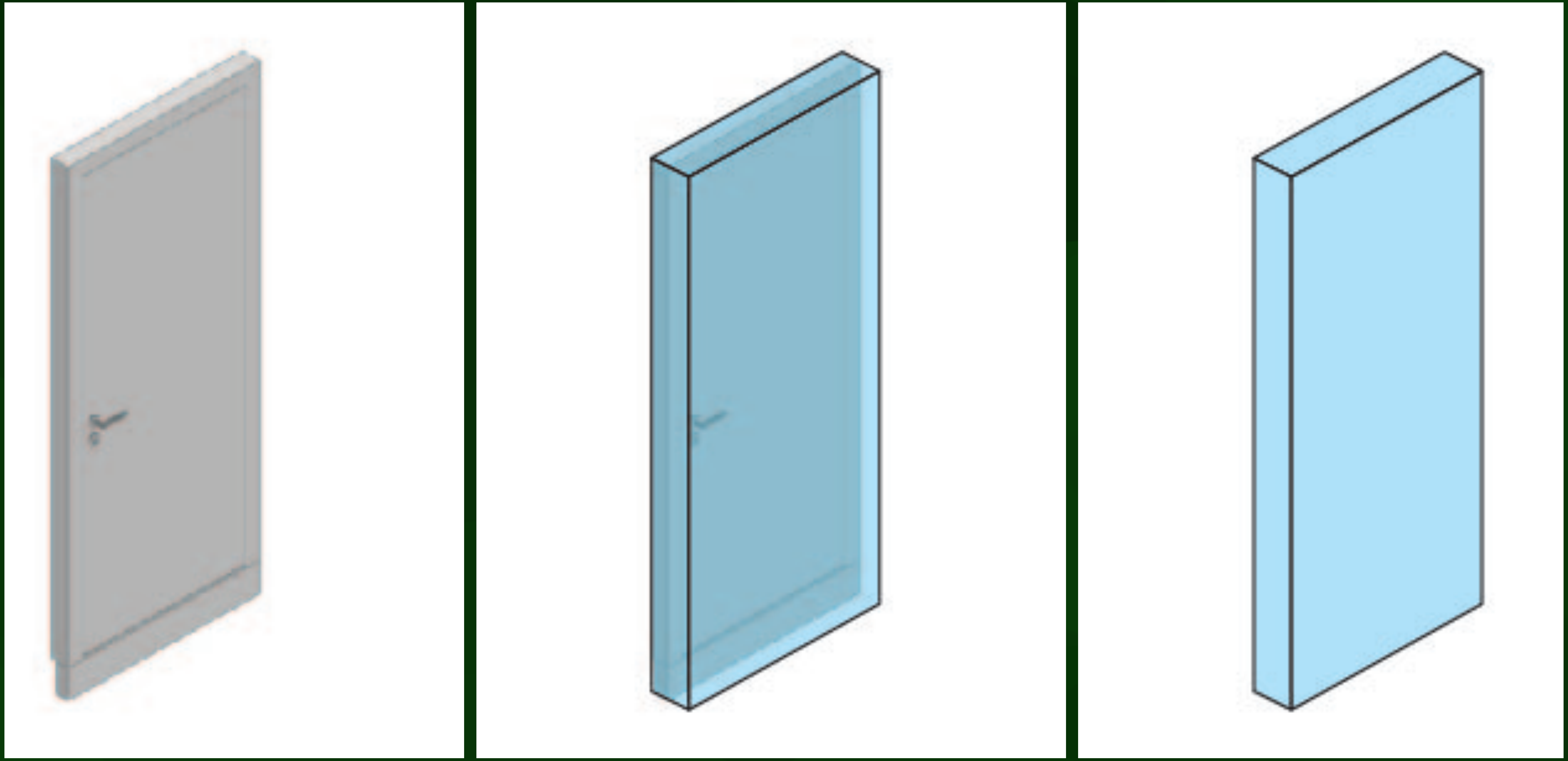
Computing rooms



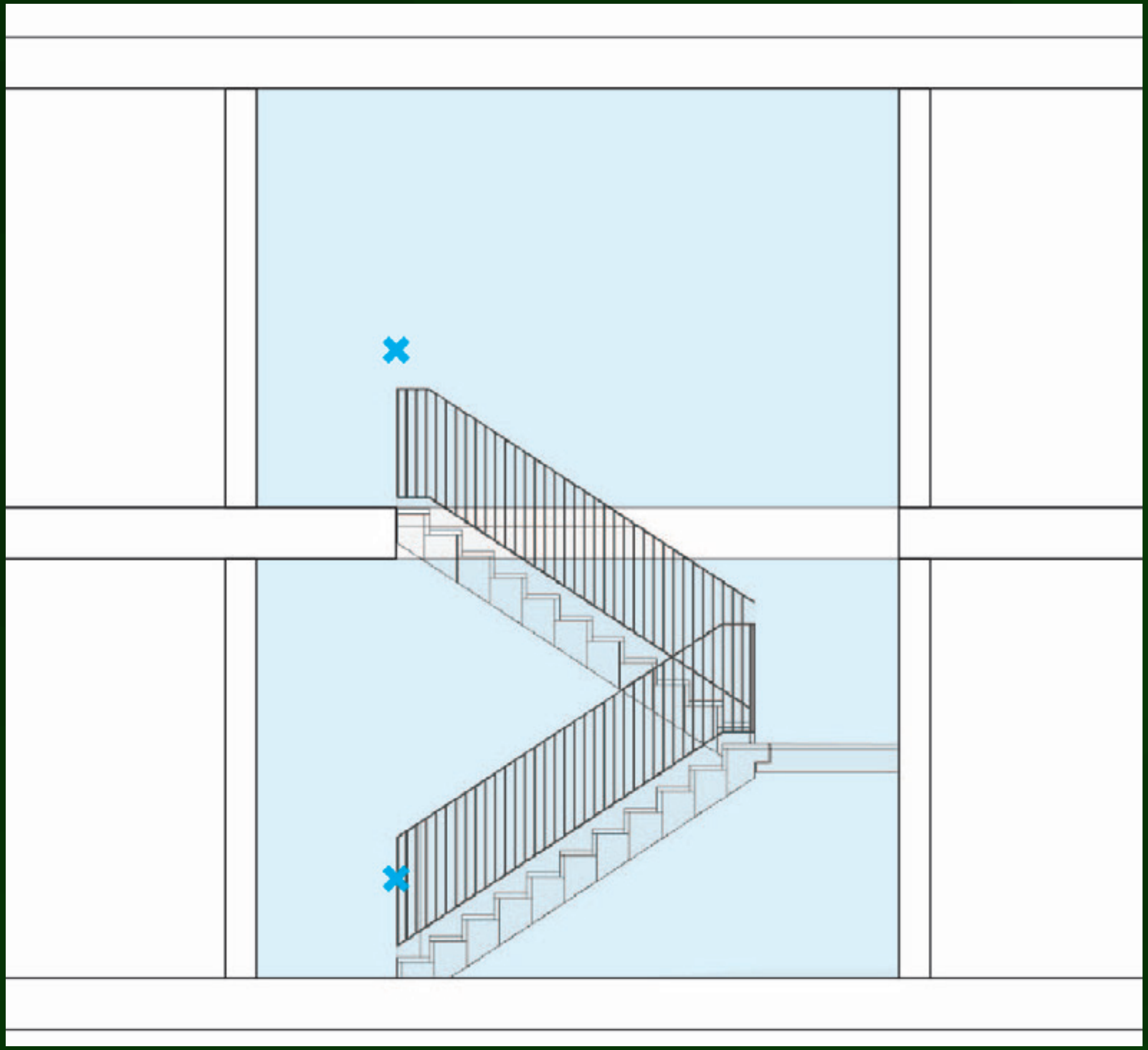
Computing rooms



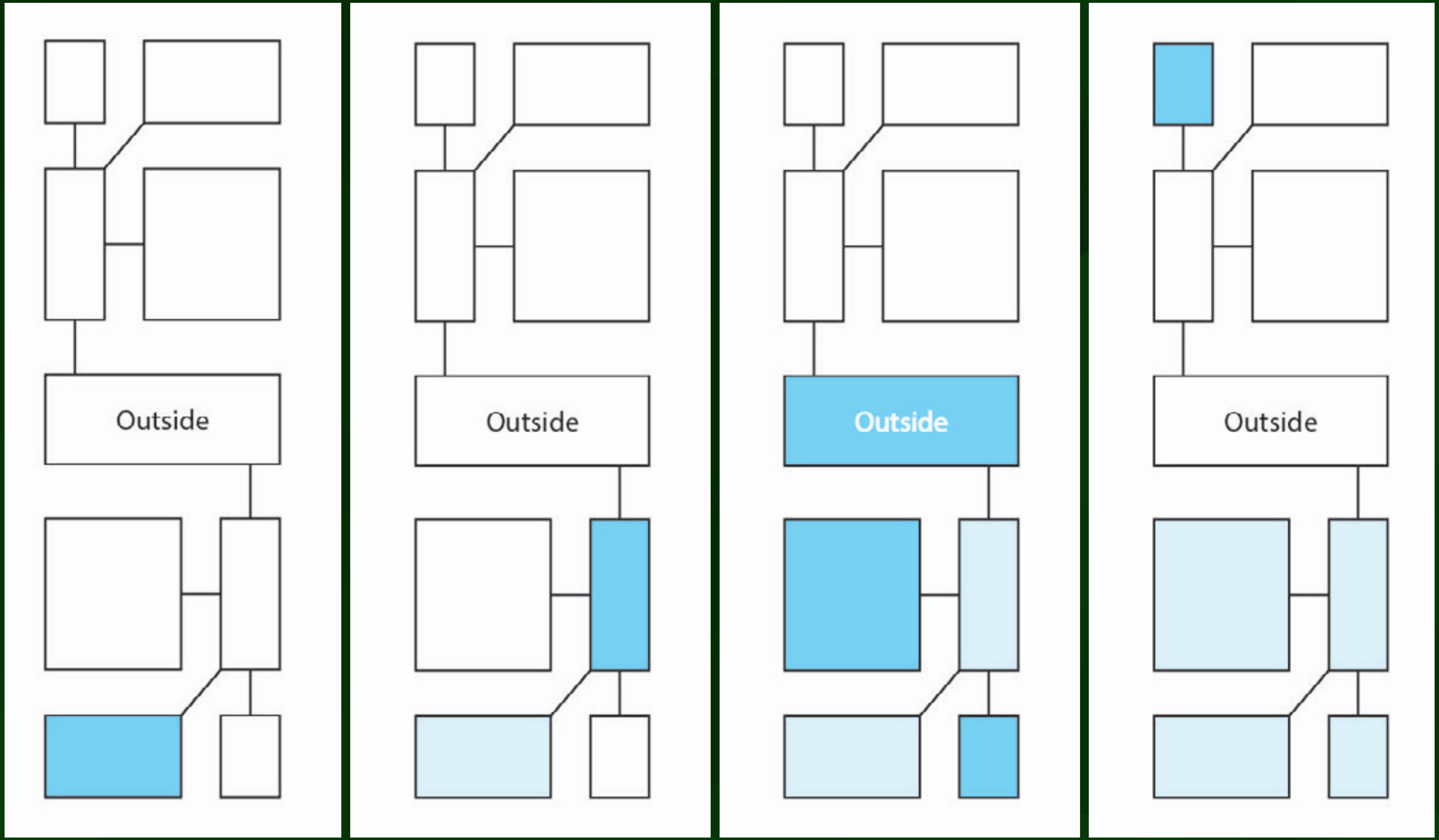
Computing rooms



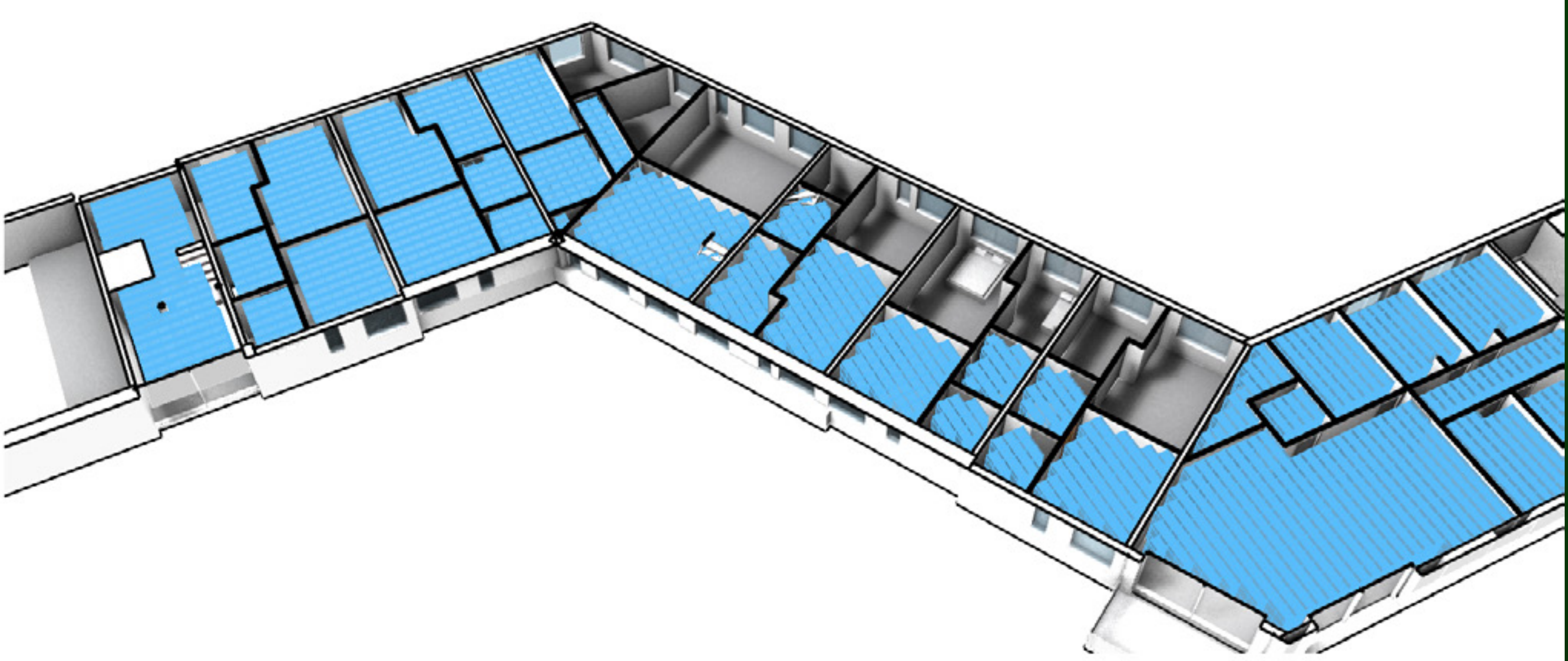
Computing rooms



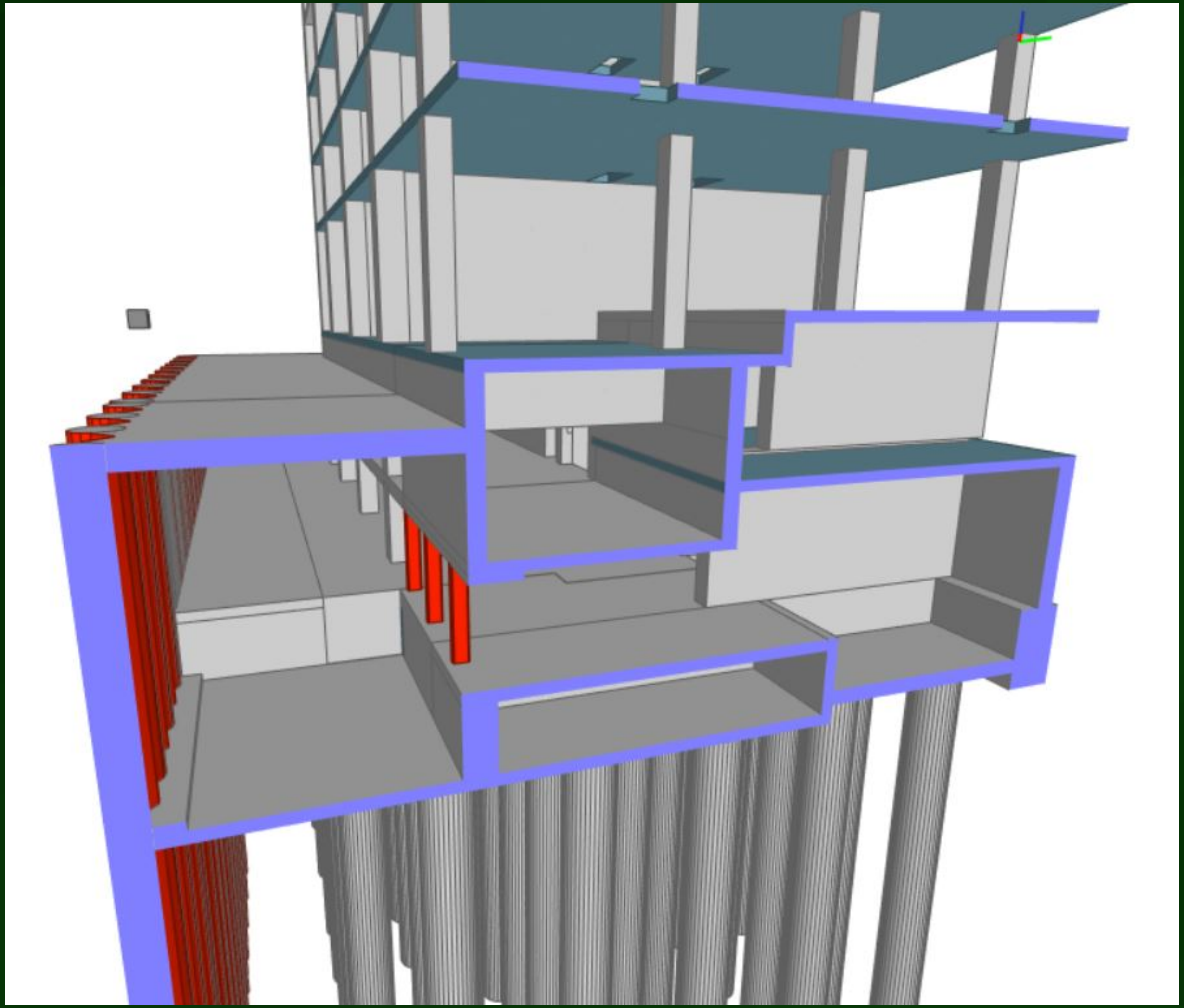
Computing apartments



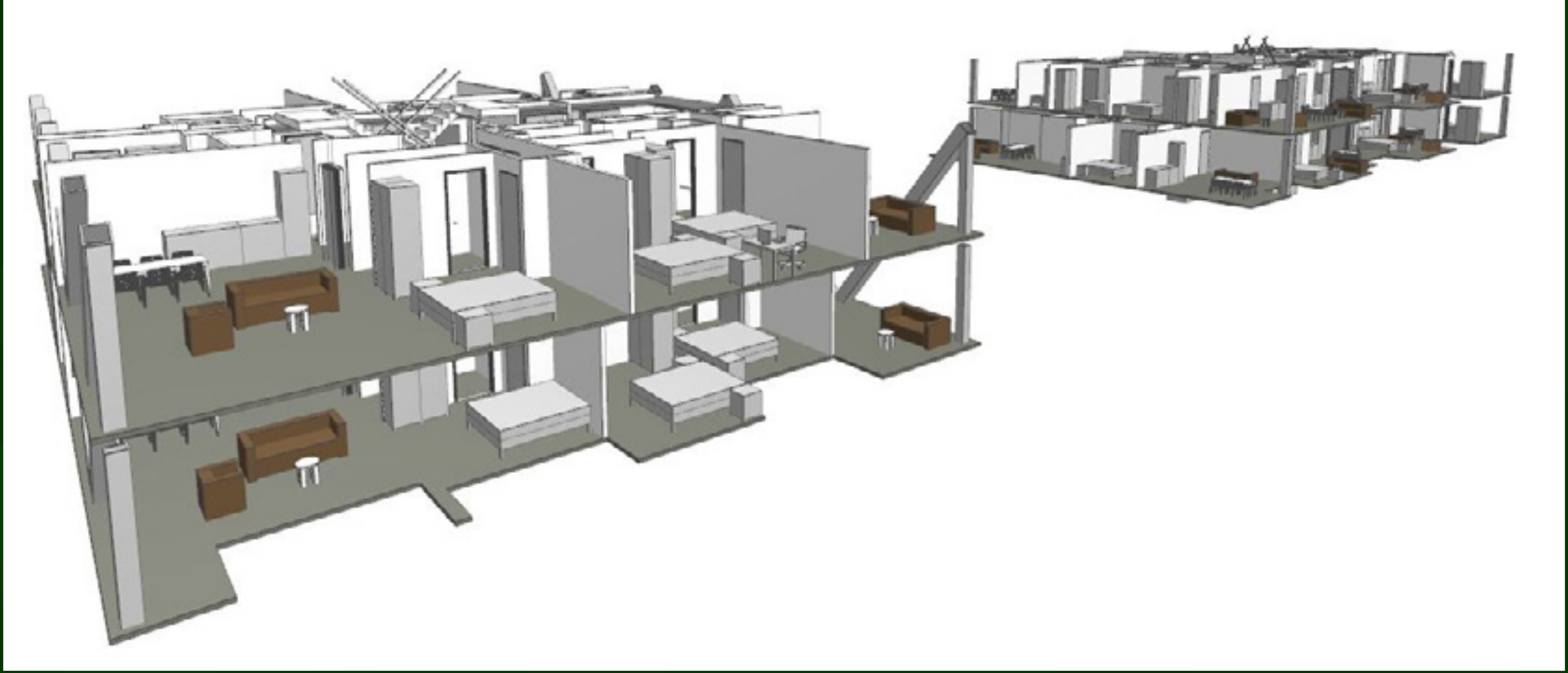
Results



Results



Results



Recommendations

- GEO5014: Geomatics as support for energy applications
- GEO5015: Modelling wind and dispersion in urban environments
- Your own MSc thesis

Sources of images

- [2-6]: Filip Biljecki (paper on application of 3D city models and PhD thesis)
- [9-20]: Roeland Boeters (MSc thesis and related paper)
- [21-29]: Sjors Donker (MSc thesis)
- [30-48]: Damien Mulder (MSc thesis)
- [49, 51-62]: Yixin Xu (MSc thesis)

Sources of images

- [50]: Anna-Maria Ntarladima (MSc thesis)
- [63-70]: Jialun Wu (MSc thesis)
- [71-78]: Özge Tufan (MSc thesis)
- [79-91]: Jasper van der Vaart (MSc thesis)