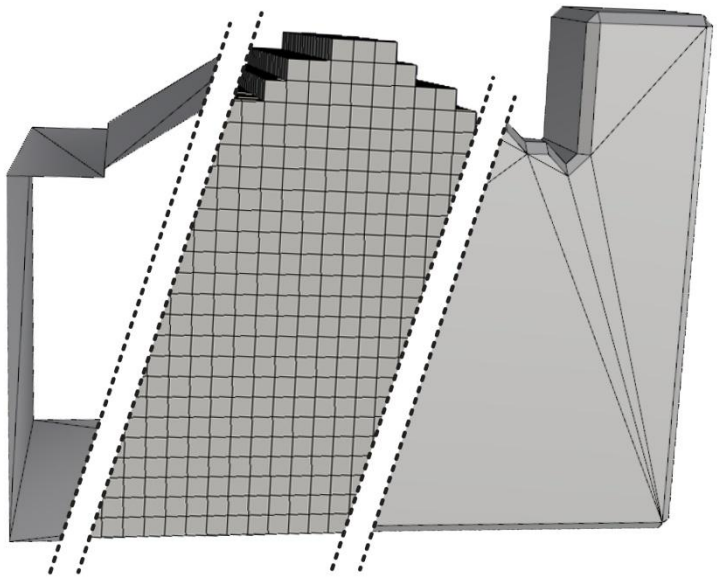


A voxel-based method for automatic repair of 3D City Building models

MSc Geomatics - P5 presentation by Damien Mulder - 22-06-2015

Content

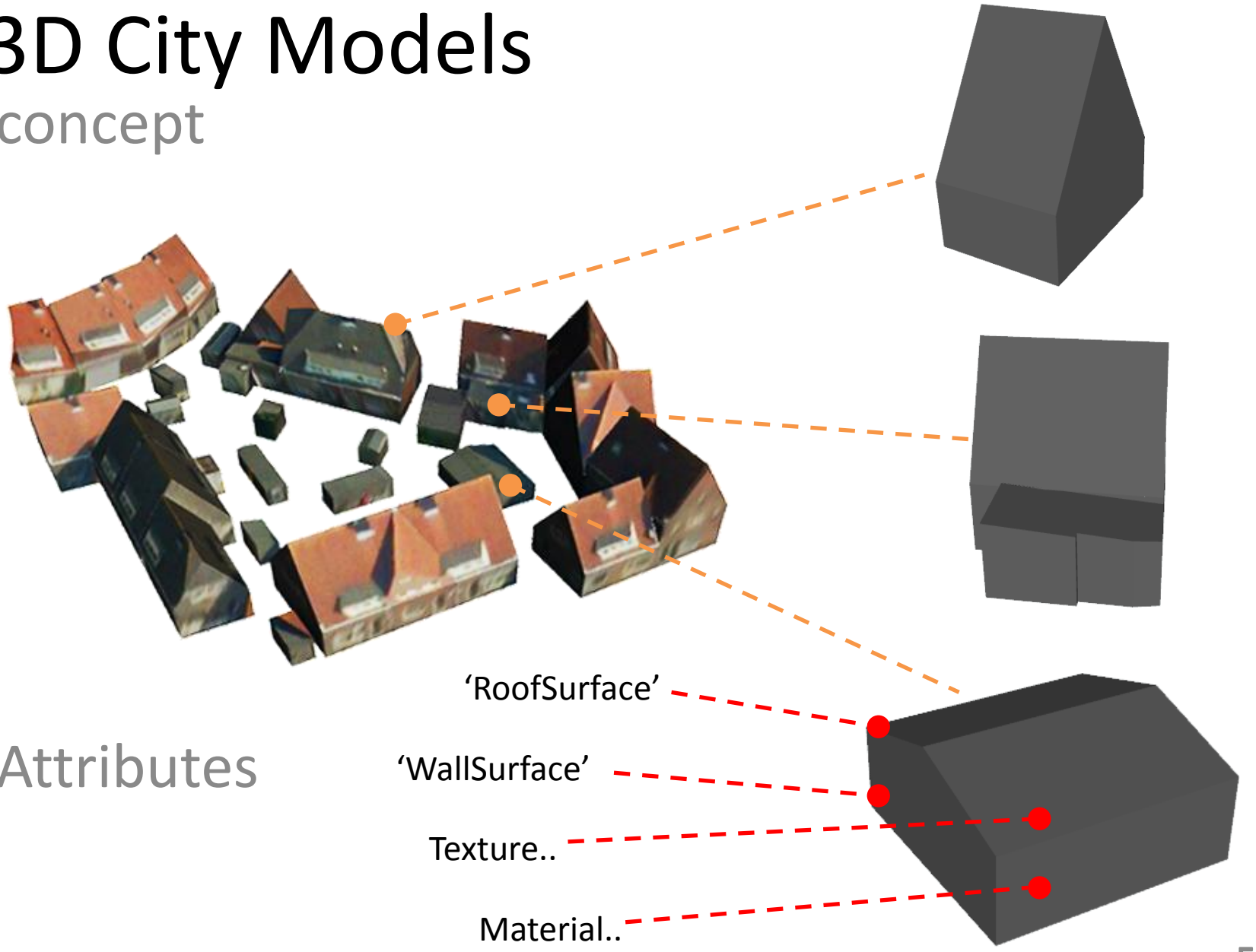
- 3D City models
- Problem statement
- Voxelization
- Approach 1
- Approach 2
- Implementation
- Conclusion & Future work


Rotterdam
aerial image


Rotterdam
aerial image
Stored as computer file

3D City Models

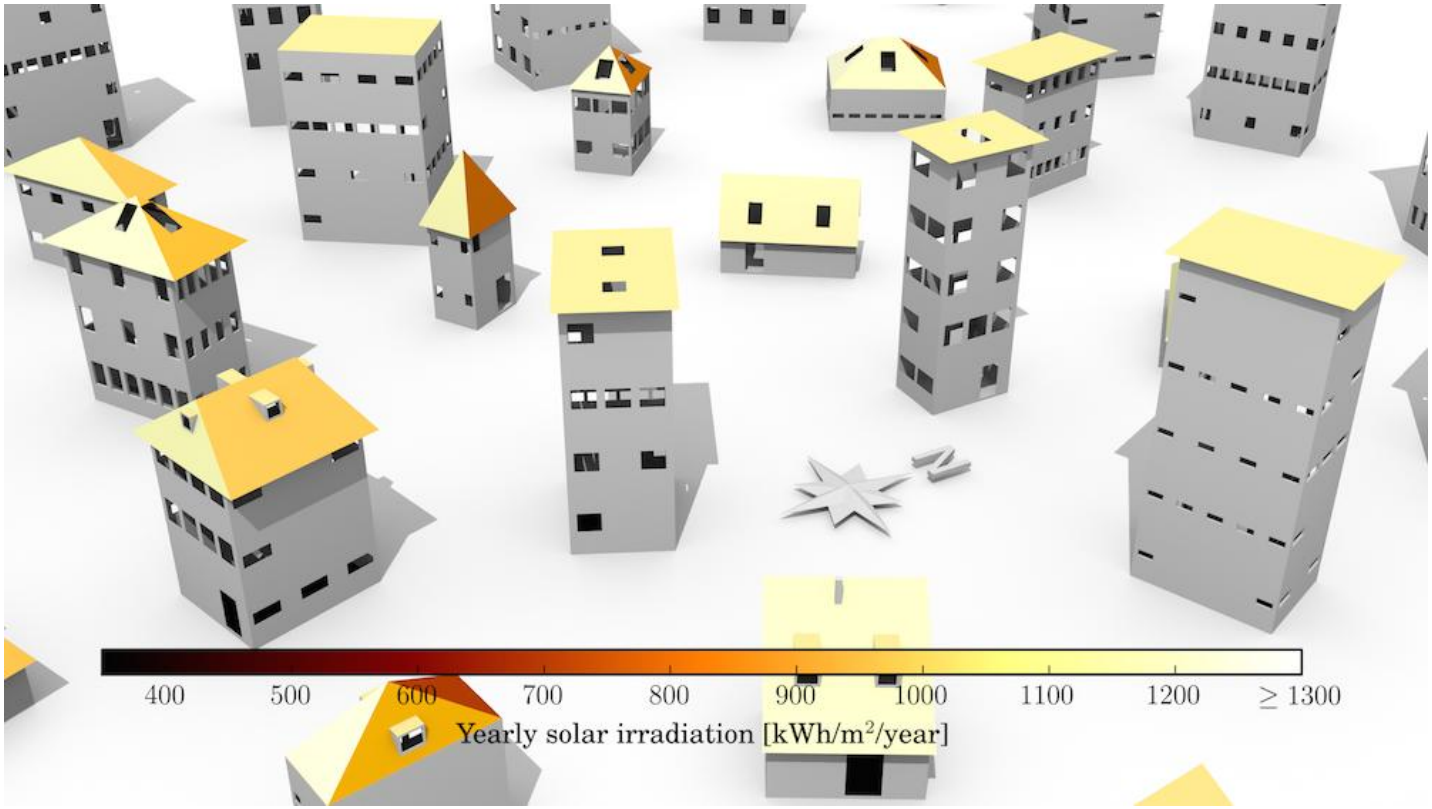
concept



Attributes

3D City Models

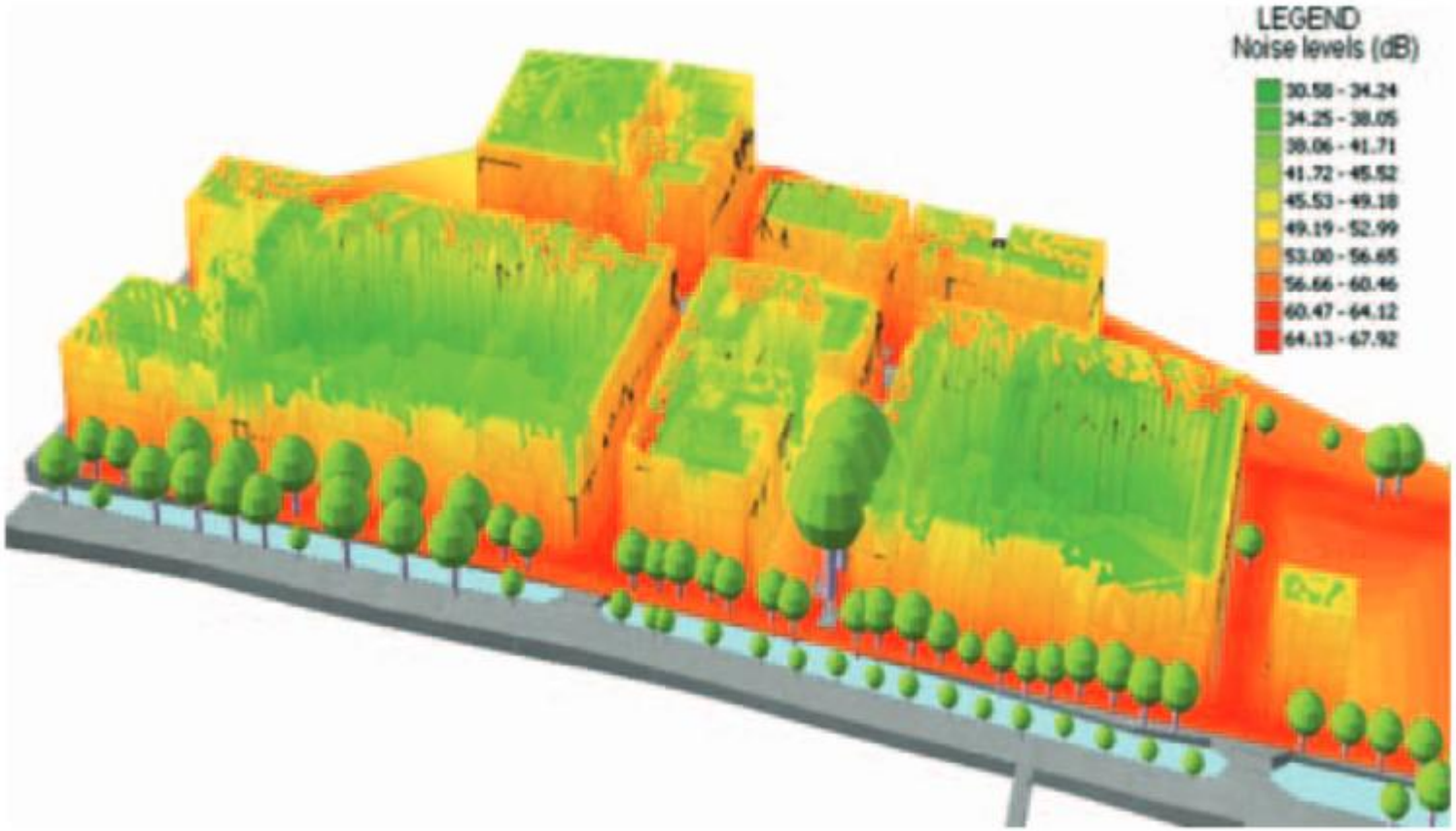
Application – Solar3DCity



(F. Biljecki, 2015)

3D City Models

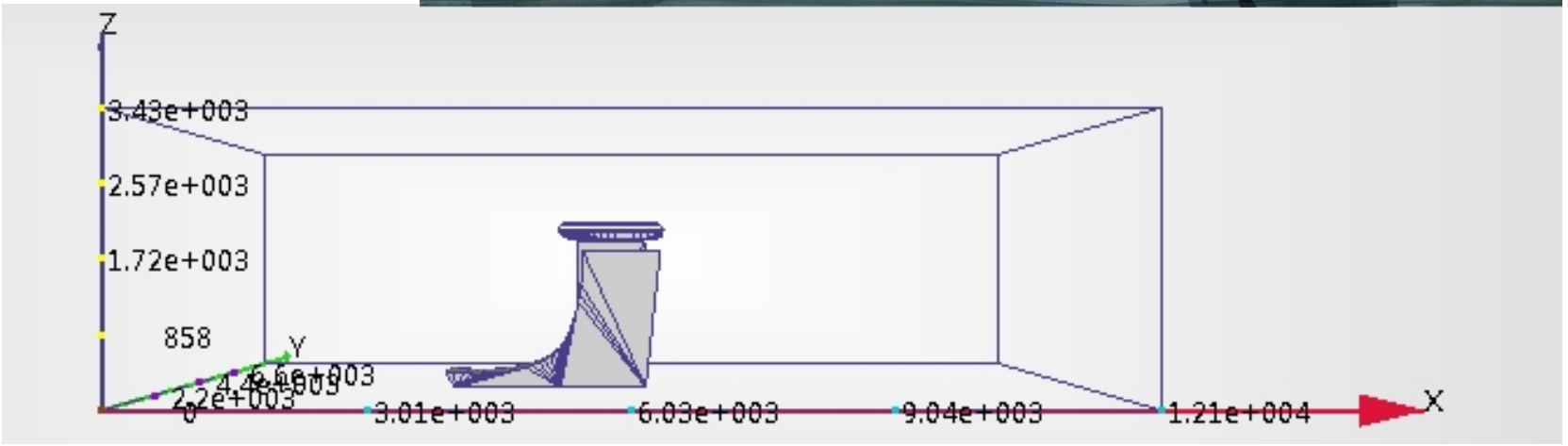
Application – Noise mapping



(J. Stoter et al., 2008)

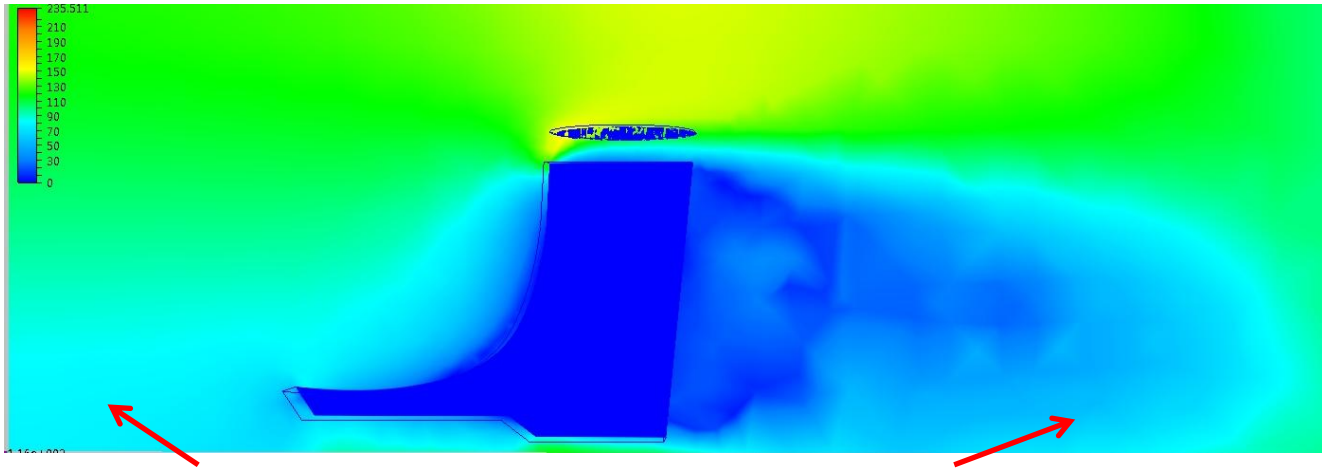
3D City Models

application

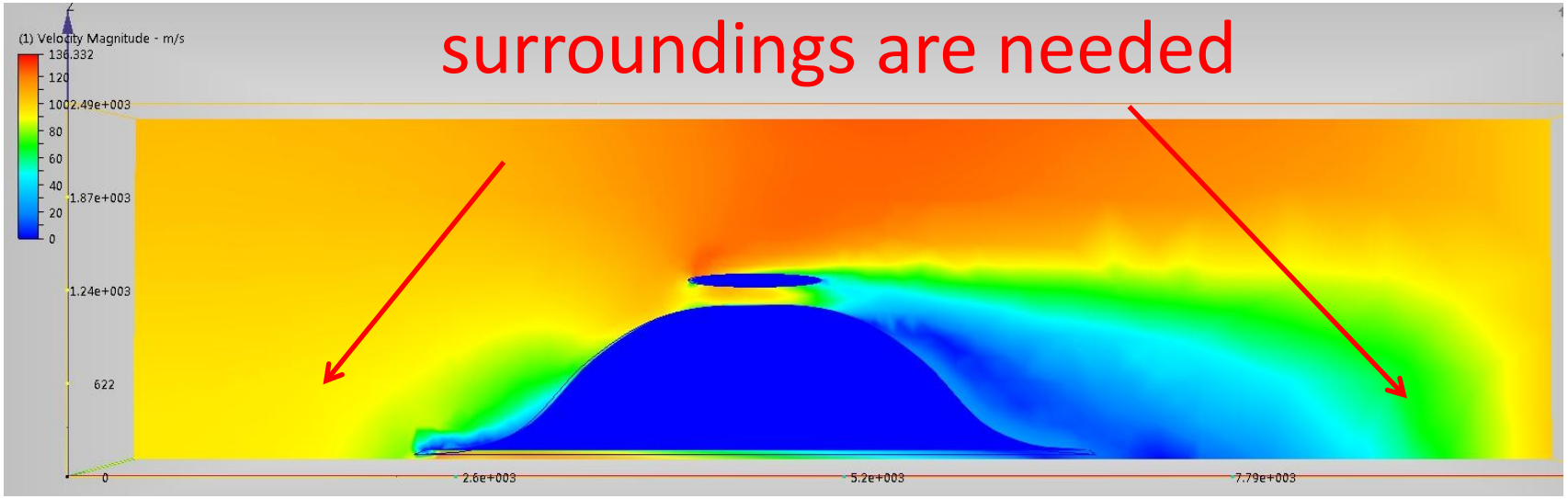


3D City Models

application



surroundings are needed



3D City Models
validation

When is the geometry valid?

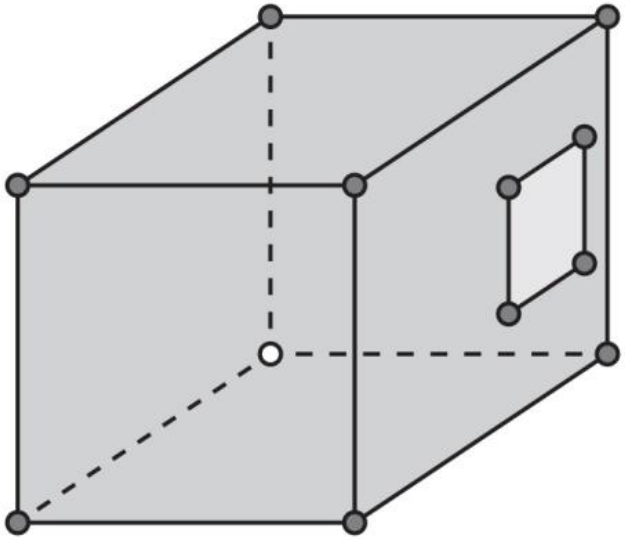
val3dity

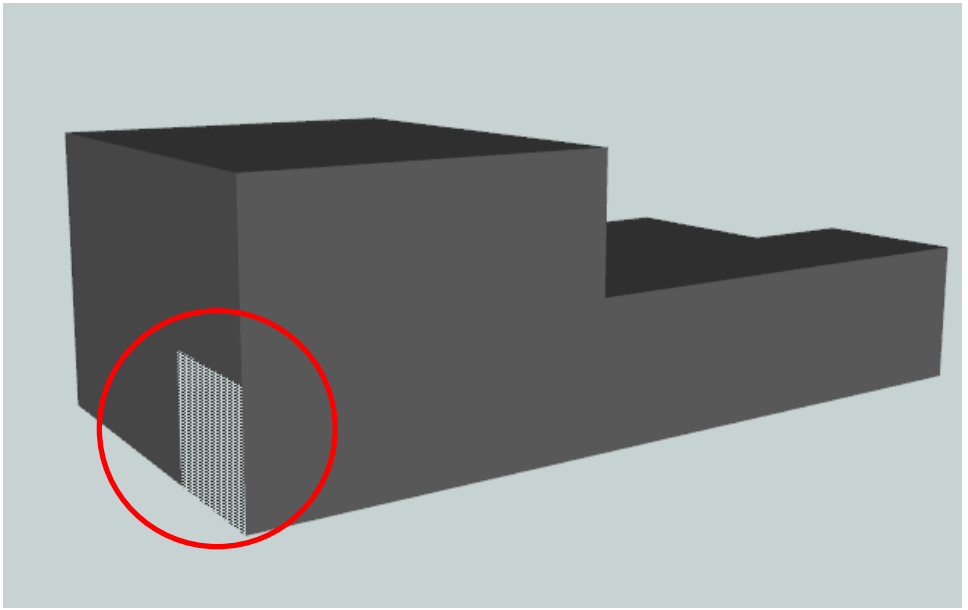
geometric validation of GML 3D primitives

(H. Ledoux, 2013)

3D City Models

validation



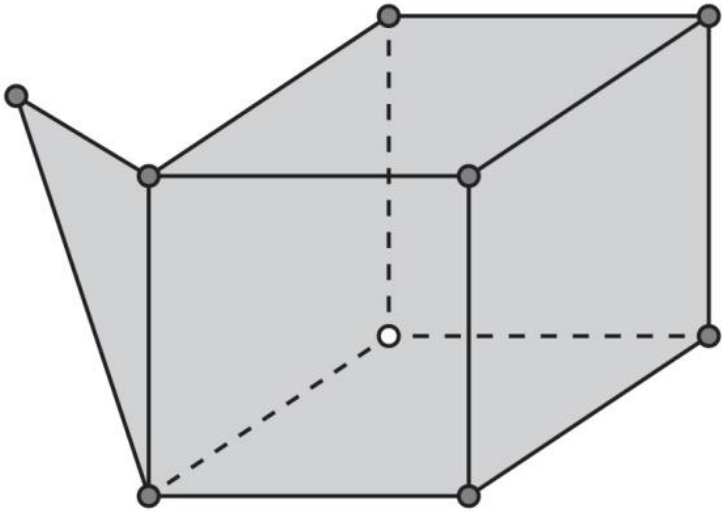


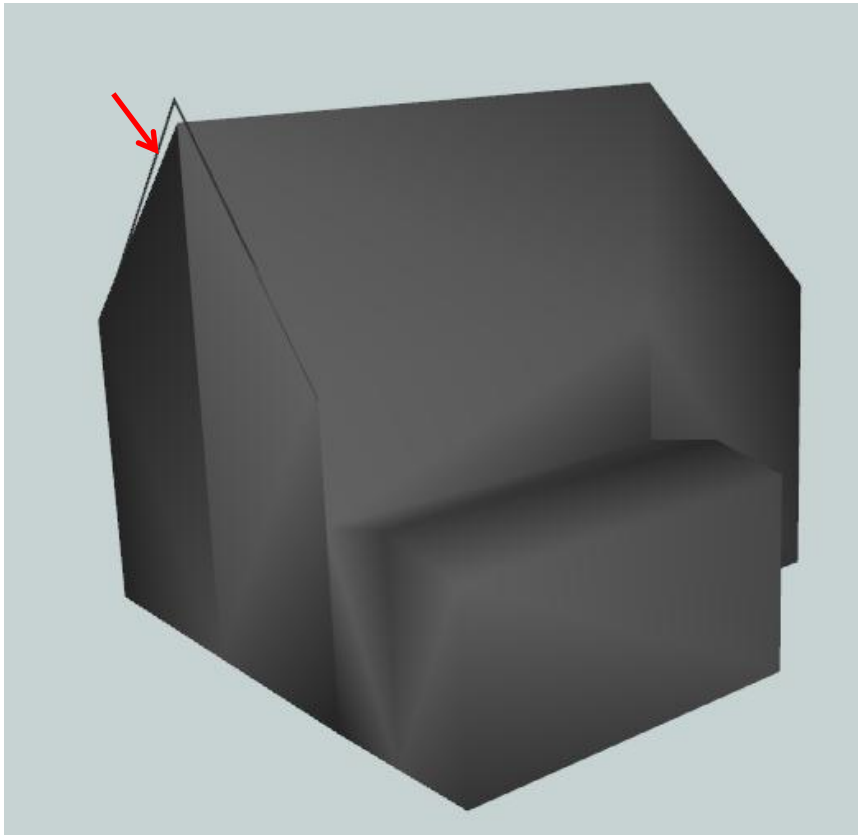
Invalid geometry:

building non closed

3D City Models

validation



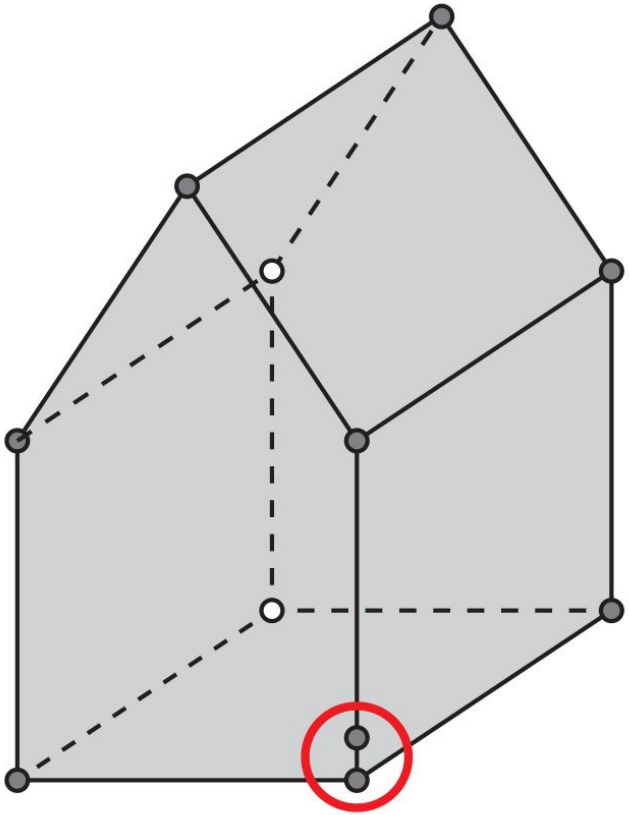


Invalid geometry:

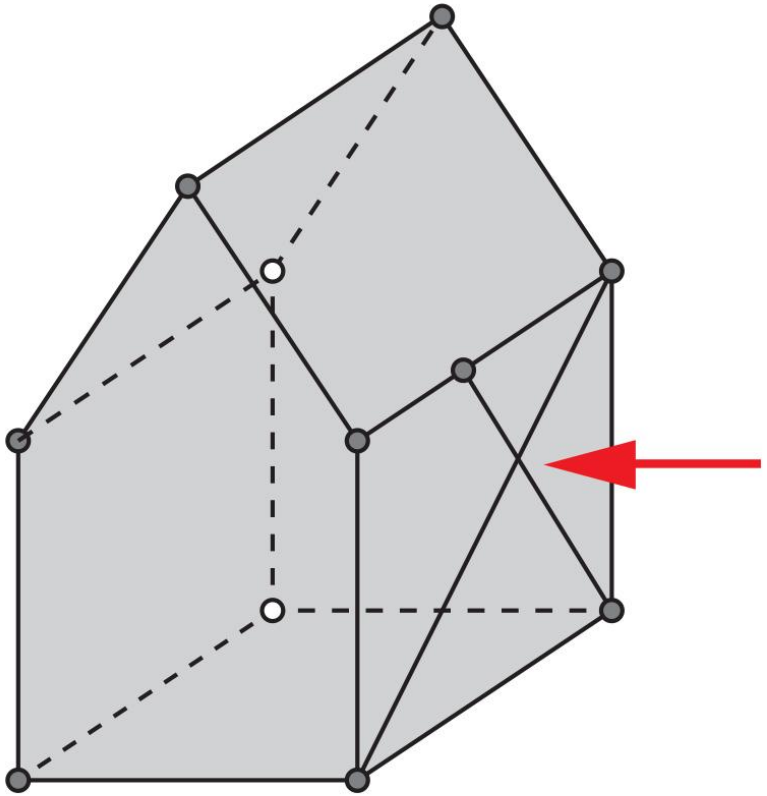
non-manifold edge

3D City Models

validation



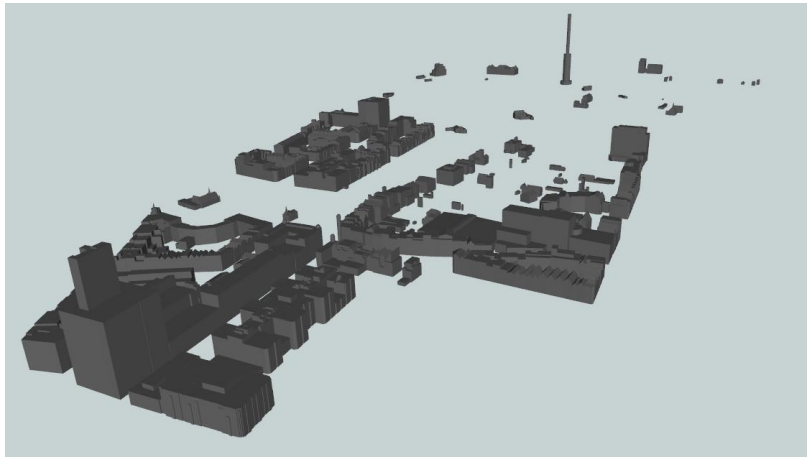
Invalid geometry:
same consective points



Invalid geometry:
self-intersections

3D City Models

Validity



Rotterdam3D **5%**

Montreal **84%**

	Buildings		Faces	
	defect	total	total	per building
Rotterdam HoogvlietZuid	10 335	10 828	100 195	9,25
Montreal VM01 2009	62	384	84 759	220,7
Montreal VM02 2009	21	209	32 973	157,7
Montreal VM03 2009	68	339	64 440	190,1

Table 1: Details concerning the test datasets

Problem statement

Repair methods

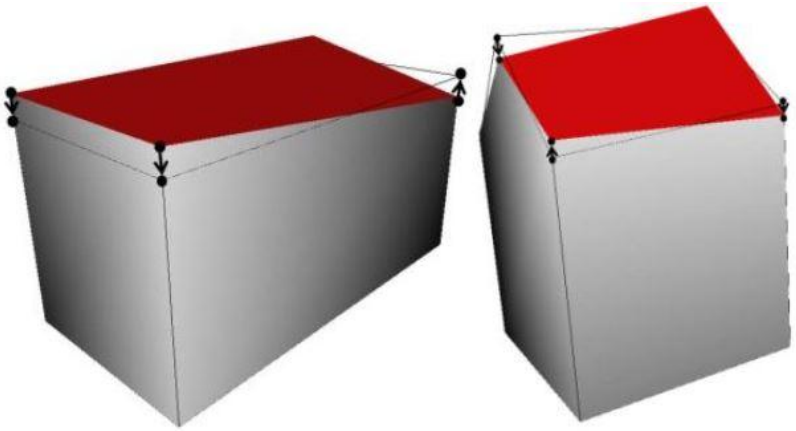
Two existing methods:

1. **Detect & Local Repair**
   (N. Alam et al., 2013)

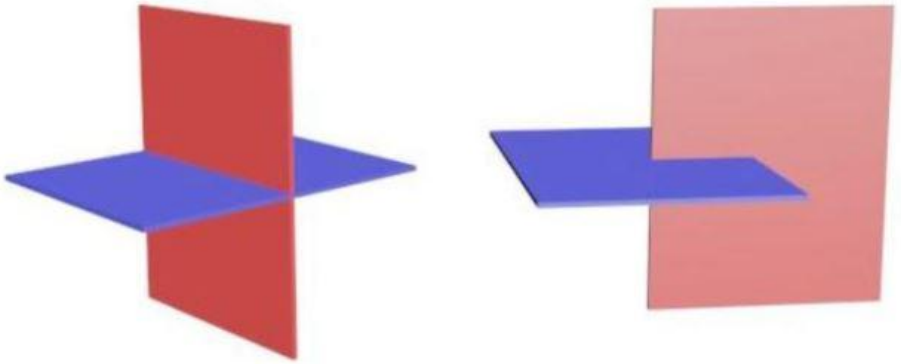
2. **Shrink Wrapping**
   (J. Zhao et al., 2013)

Problem statement

1. Detect & local repair



Non-planar polygon

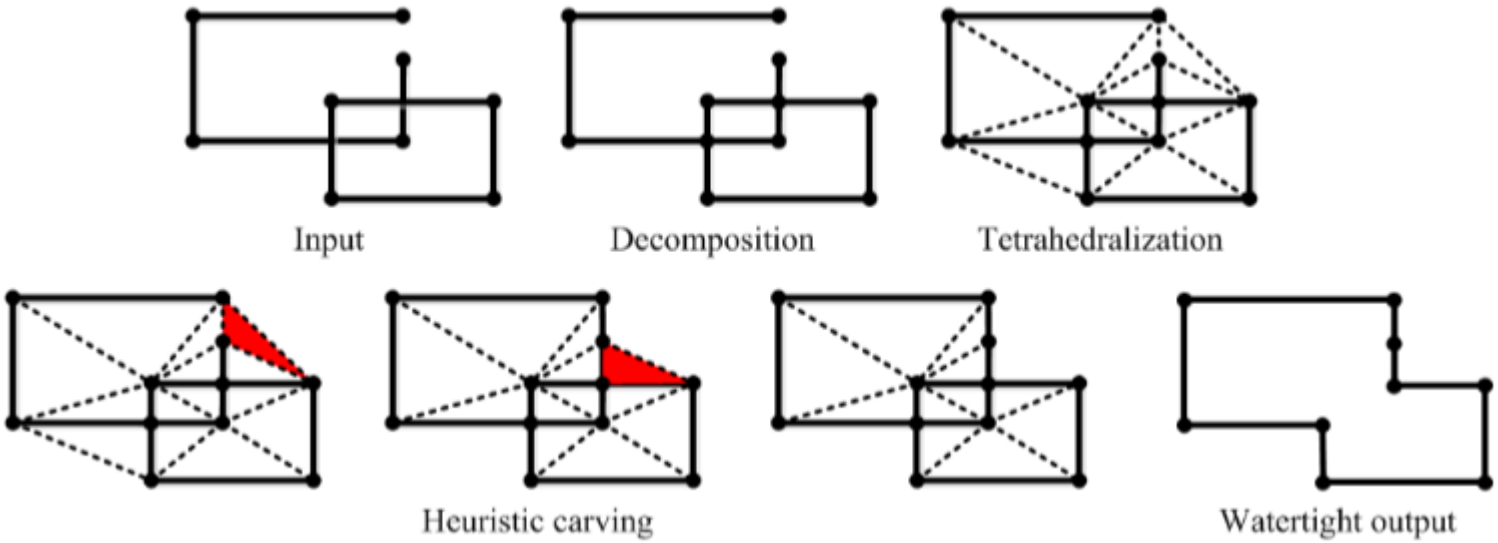


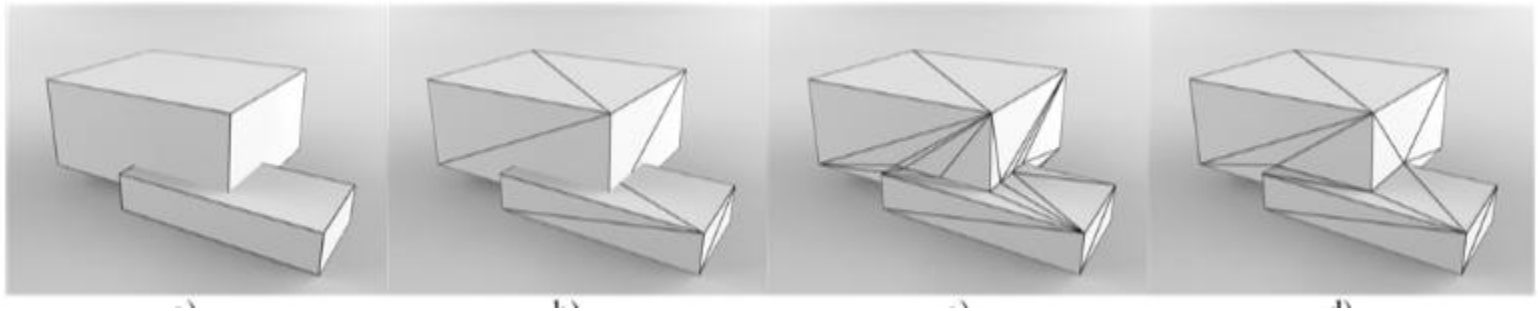
Self-intersections

(N. Alam et al., 2013)

Problem statement

2. Shrink Wrapping

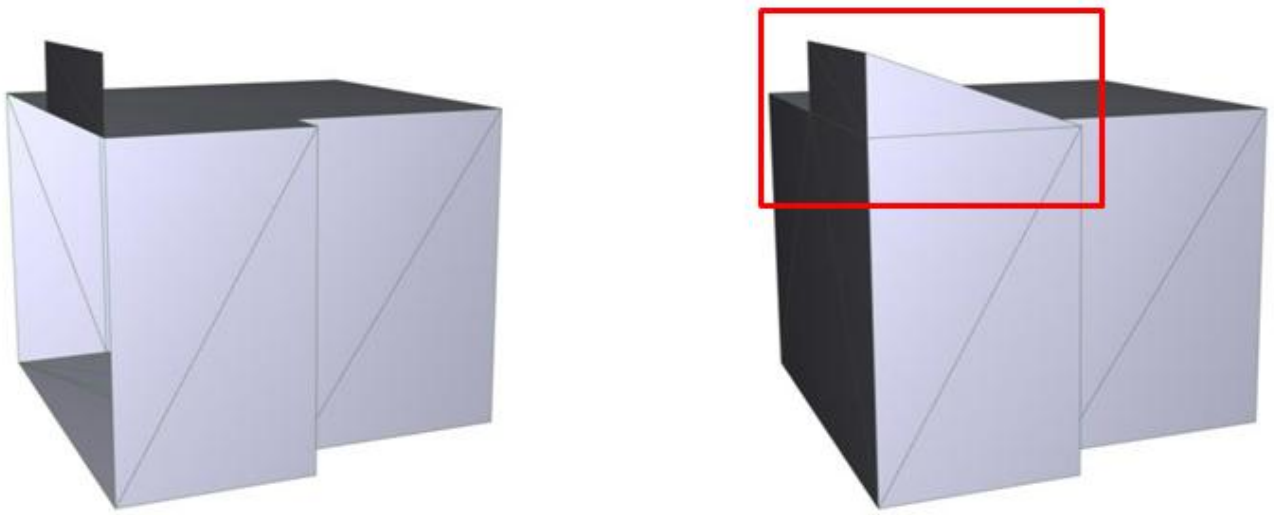




(J. Zhao et al., 2013)

Problem statement

2. Shrink Wrapping

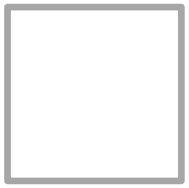


(J. Zhao et al., 2013)

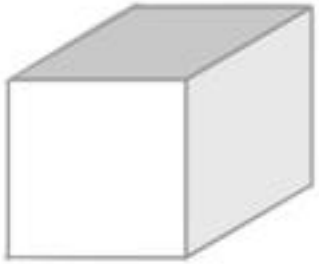
Problem statement

Alternative method

3. **Voxel-based**
   (Nooruddin & Turk, 2005)



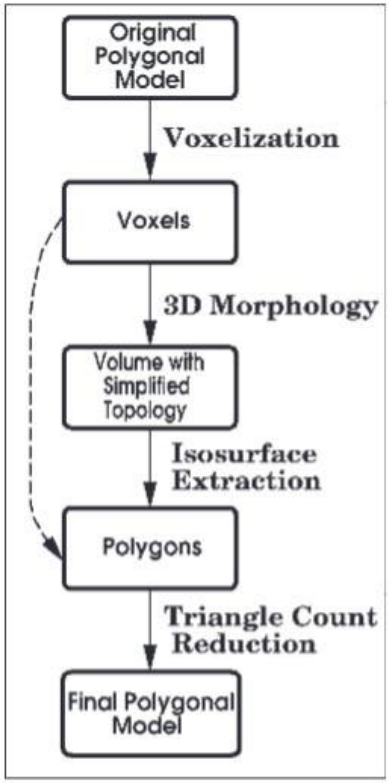
pixel

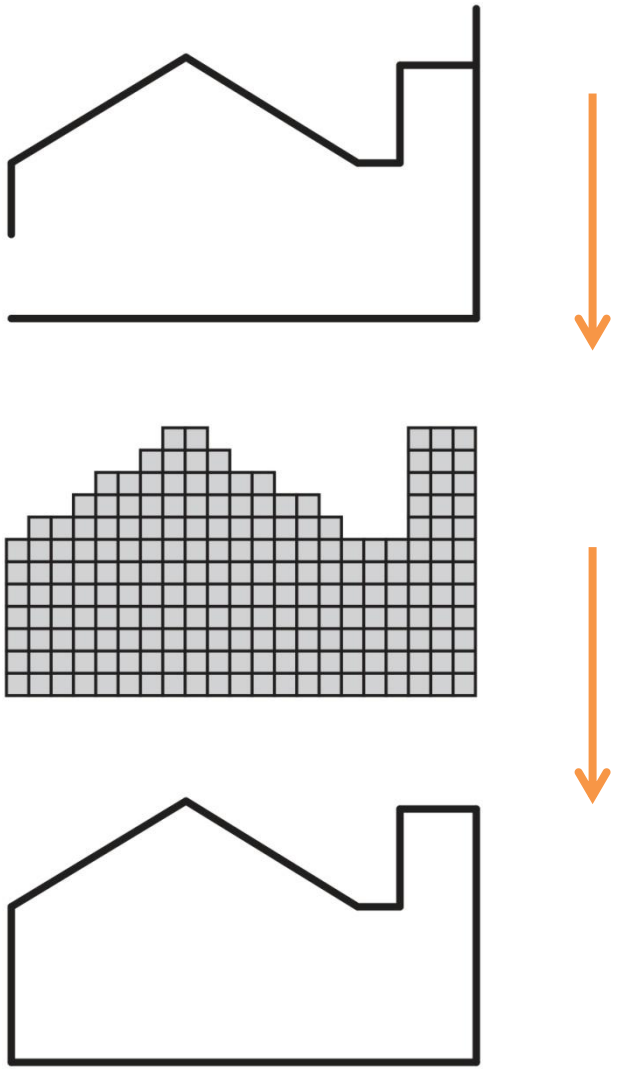


voxel

Problem statement

Alternative repair method





(Noorruddin & Turk 2005)

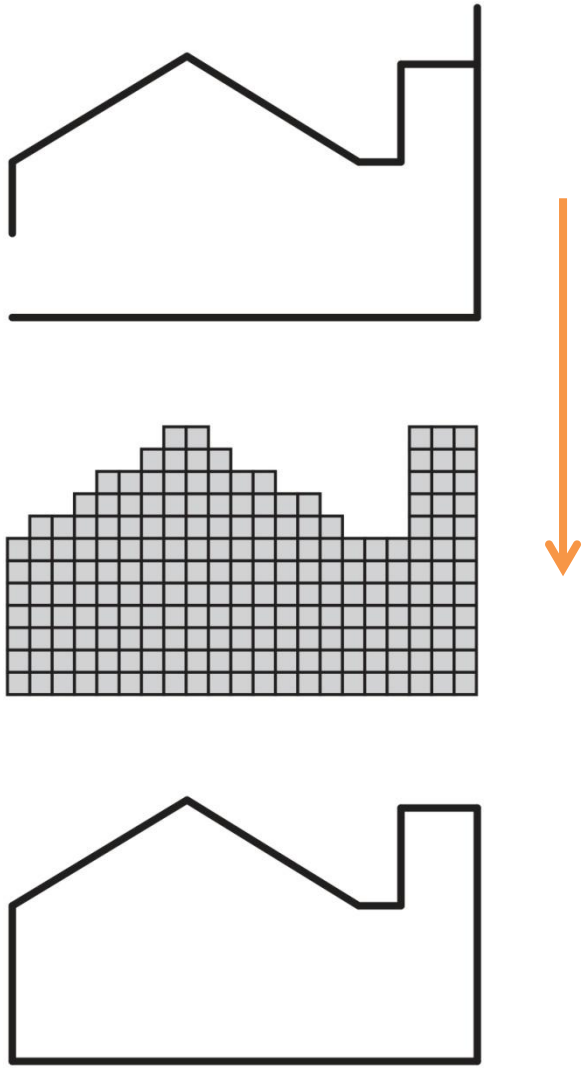
Problem statement

Current repair methods

	1. Detect & Local Repair	2. Shrink Wrapping	3. Voxel-based
Gaps	++	++	++
Consecutive points	++	++	++
Self intersections	-	+	+
Non-manifold edges	--	-	++
Non-manifold vertex	-	-	?
Attributes preserved	++	++	-
Tilted surfaces	++	++	?

Research question

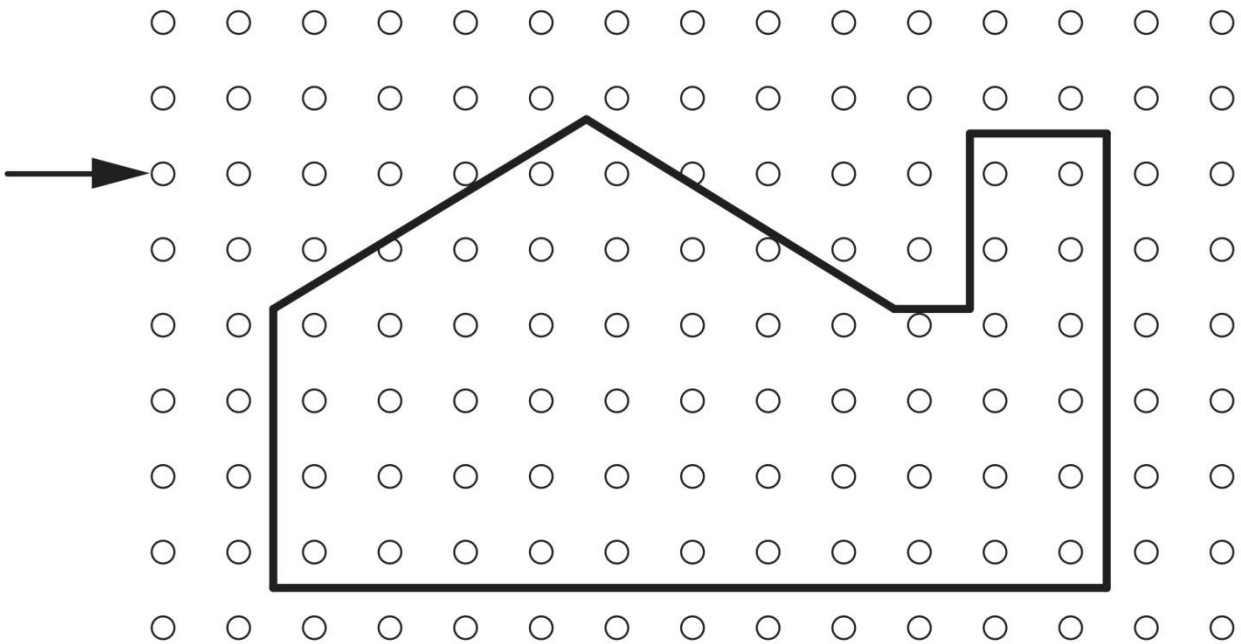
To which extent is it possible to
automatically repair a geometrically invalid
3D City Building Model using a voxel-based method?



Voxelization

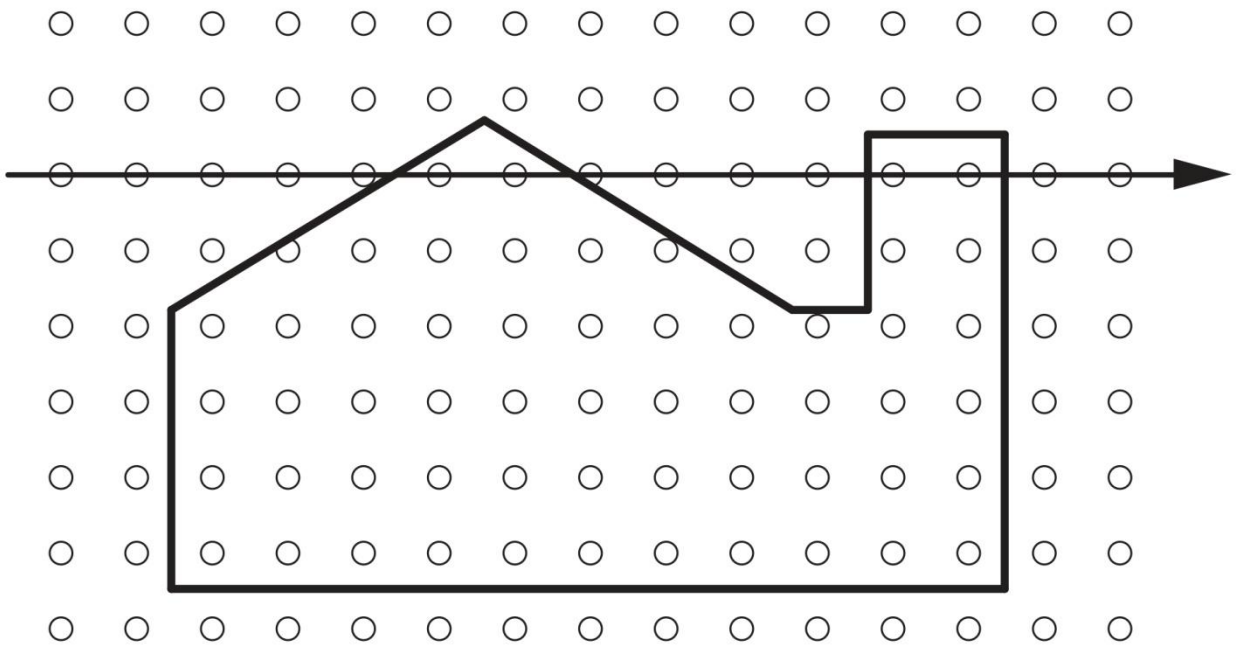
Voxelization

scan conversion



Voxelization

scan conversion

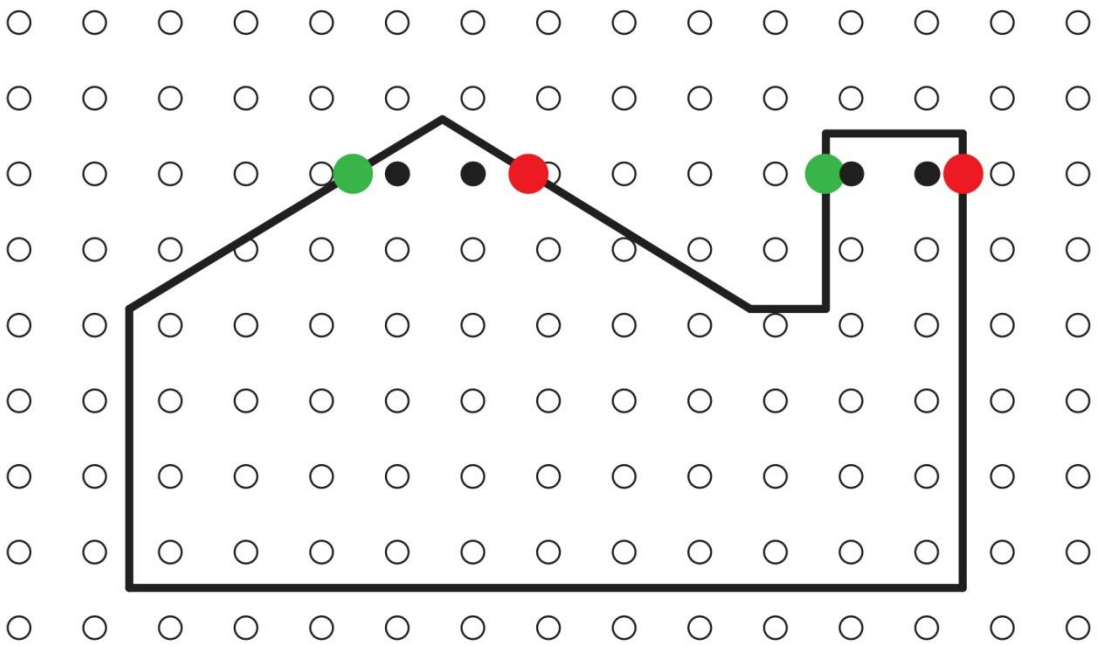


Voxelization

scan conversion

Voxelization

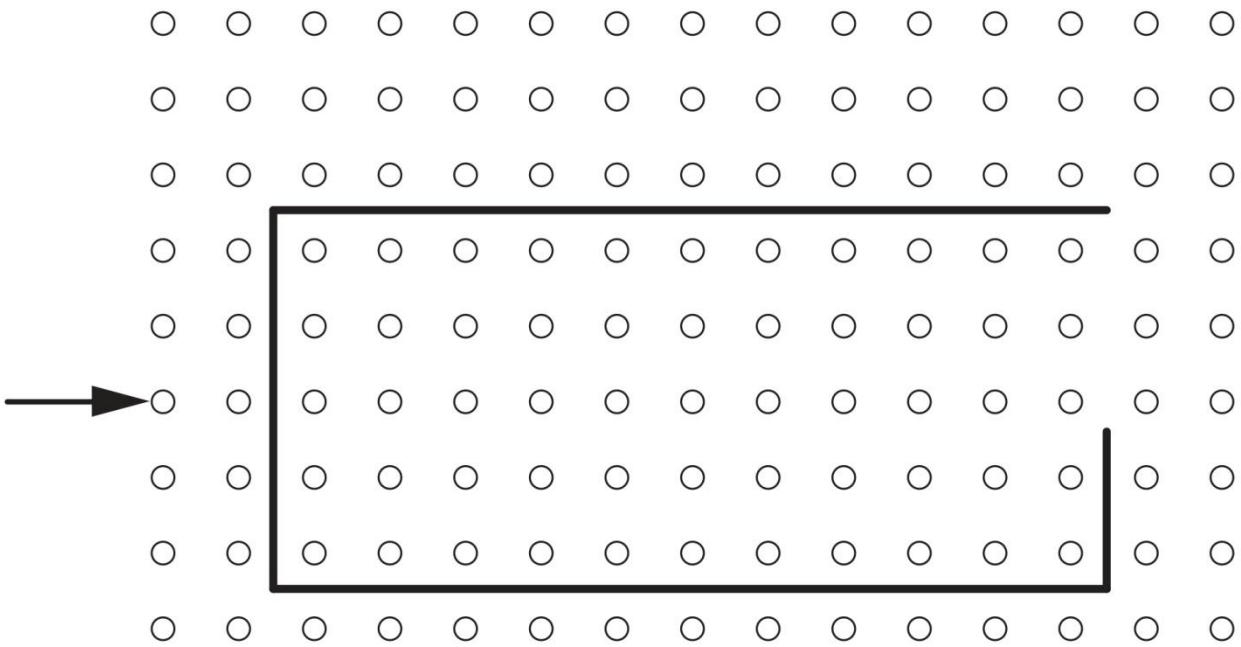
scan conversion



‘parity count’

Voxelization

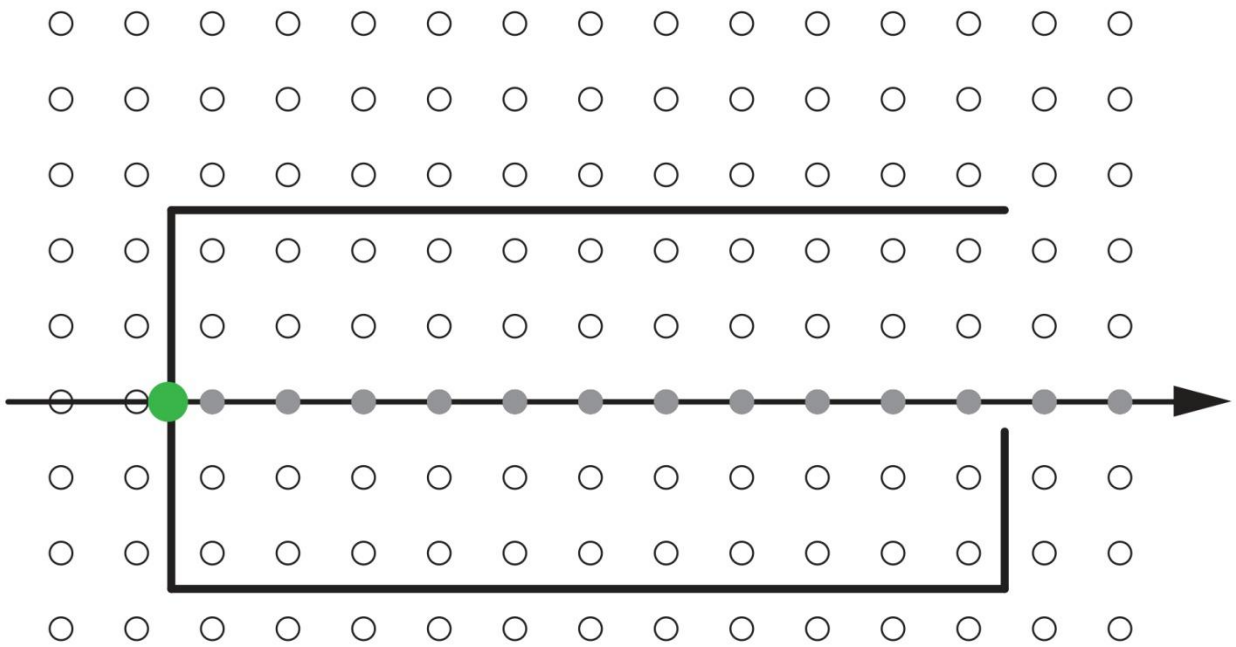
scan conversion



gap

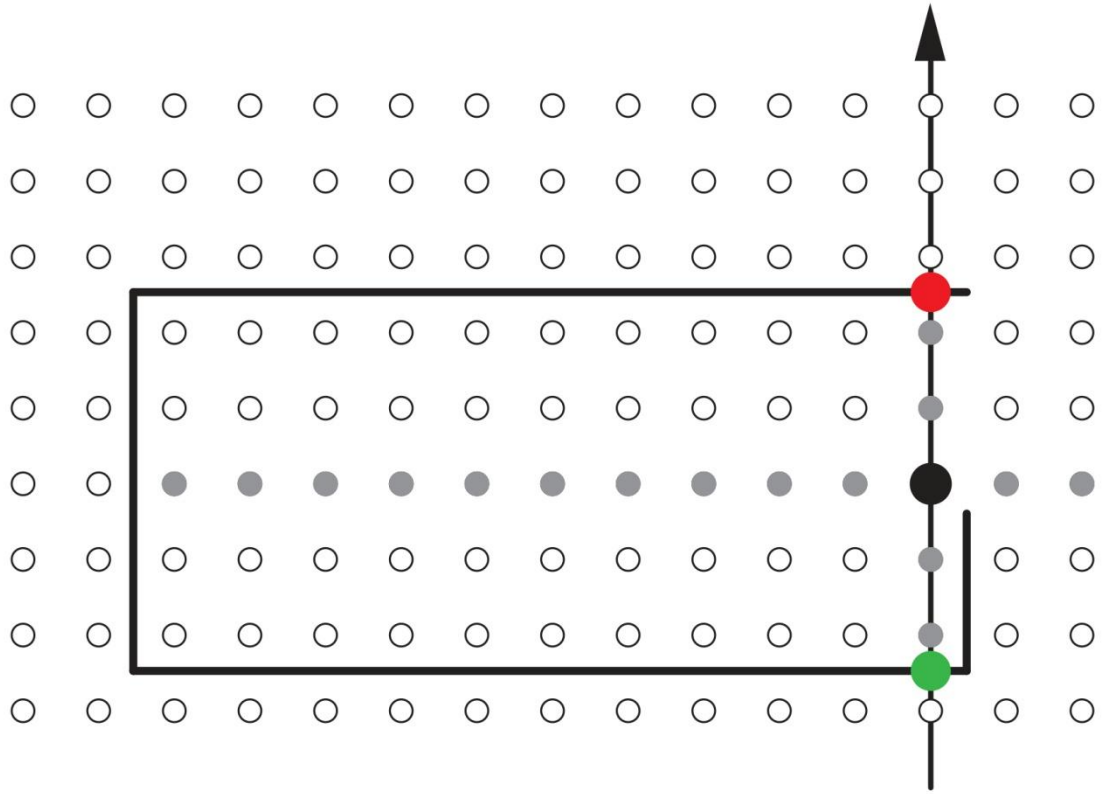
Voxelization

scan conversion



Voxelization

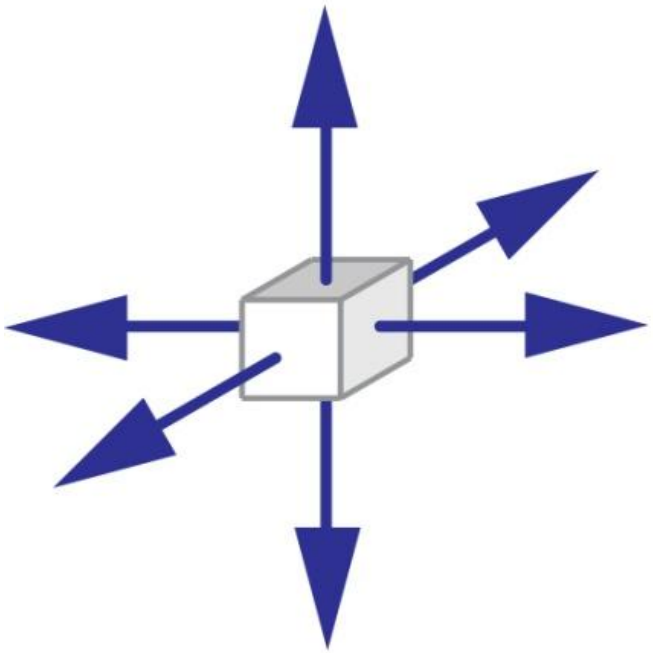
scan conversion



‘majority voting’

Voxelization

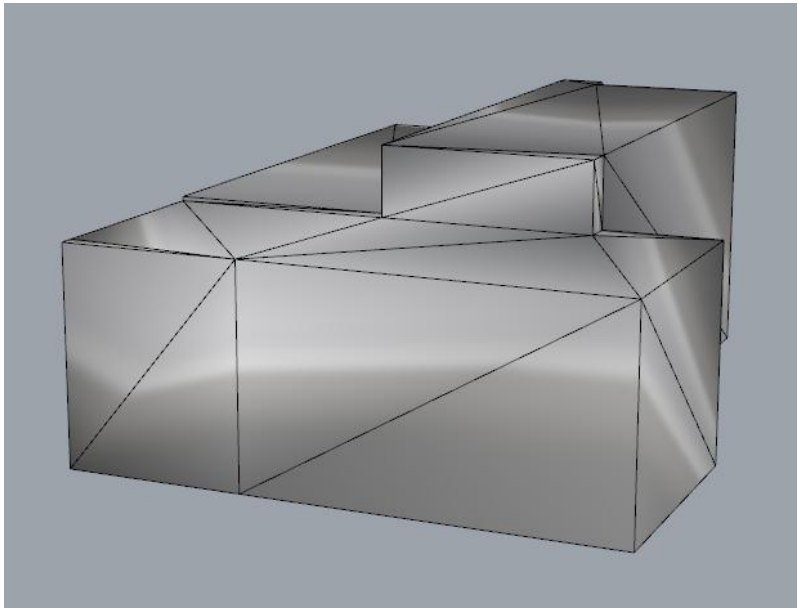
number of rays

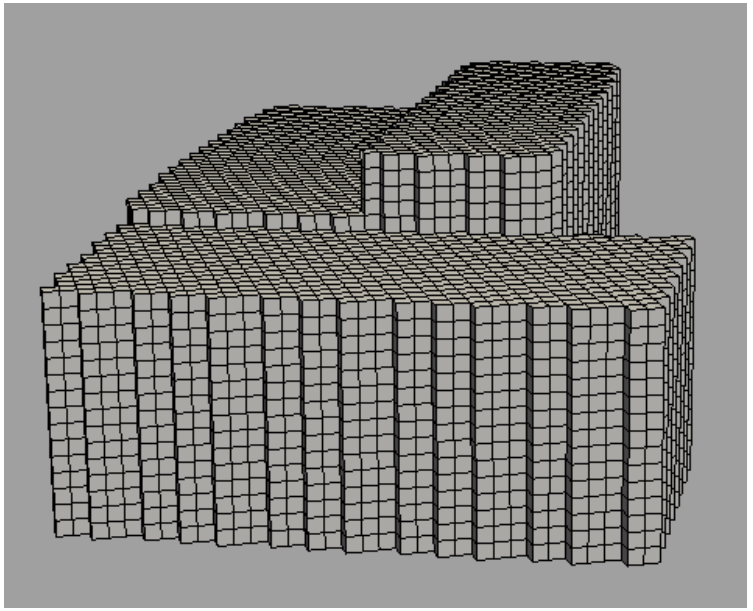


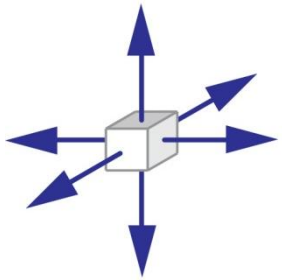
scanning in **6 directions**
majority is **4 votes**

Voxelization

Voxelization example



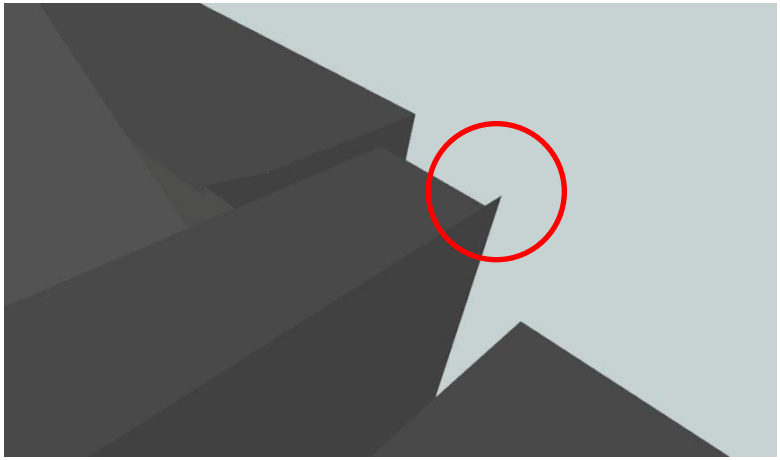


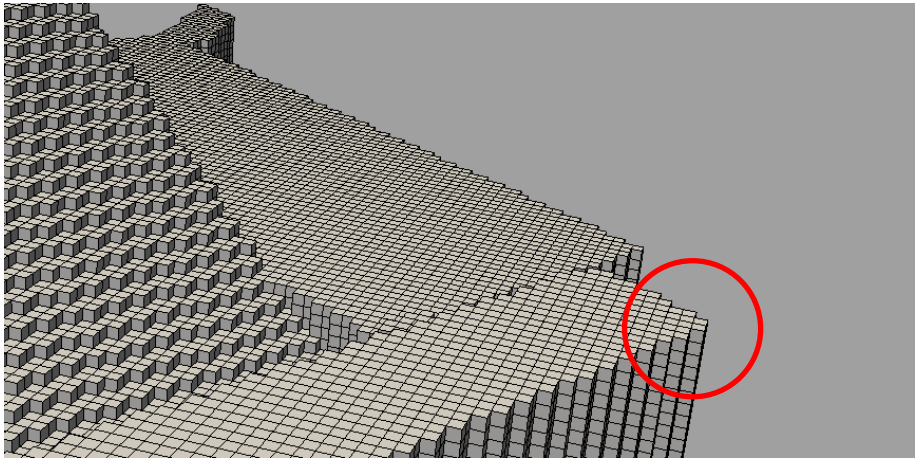


4/6

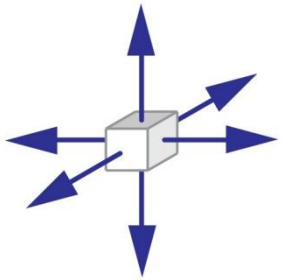
Voxelization

Repair capability



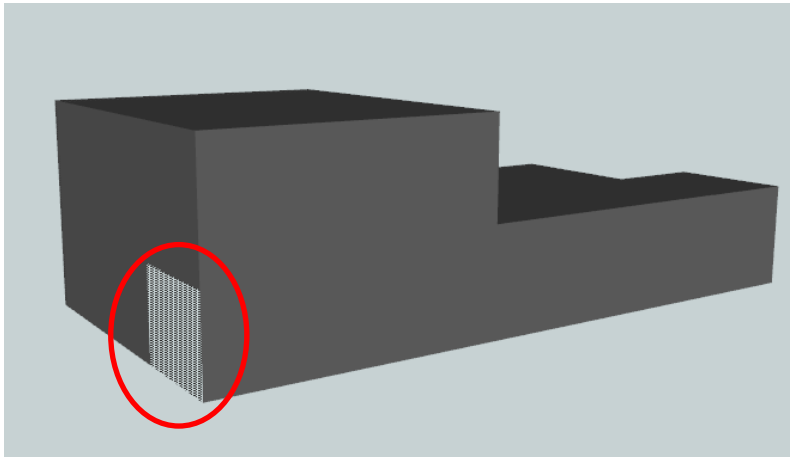


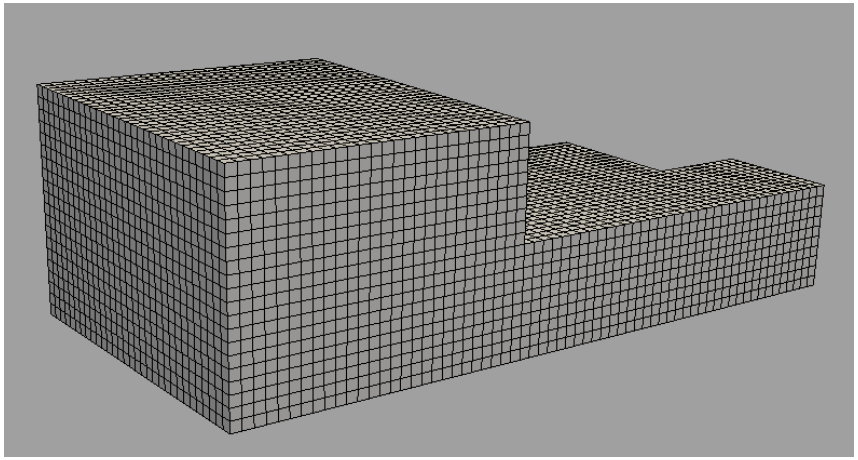
4/6



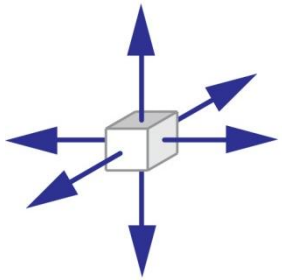
Voxelization

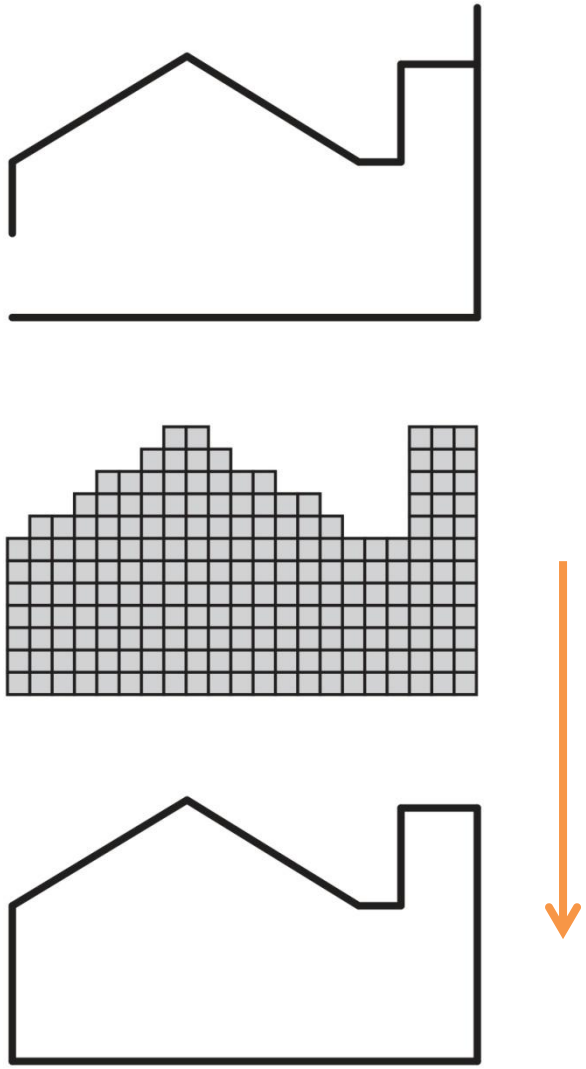
Repair capability





4/6





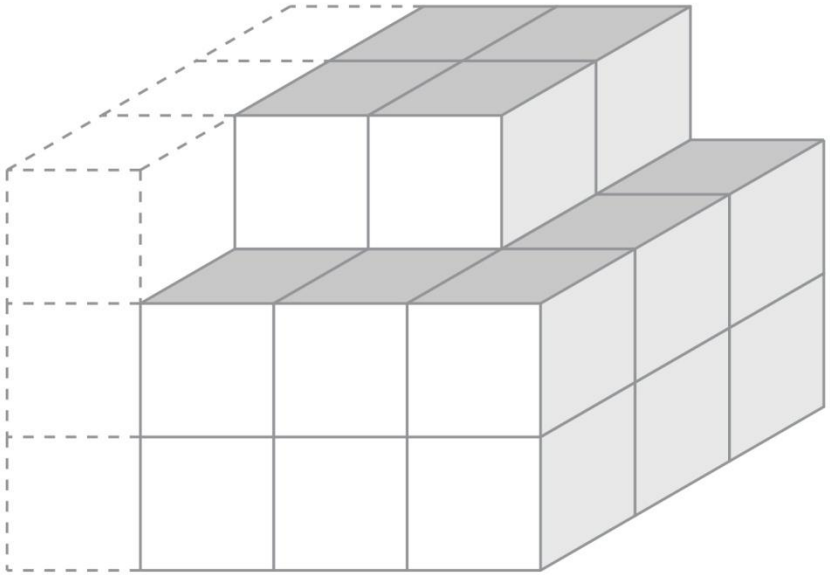
Surface reconstruction

Approach 1: Marching Cubes

Approach 2: Dual Contouring

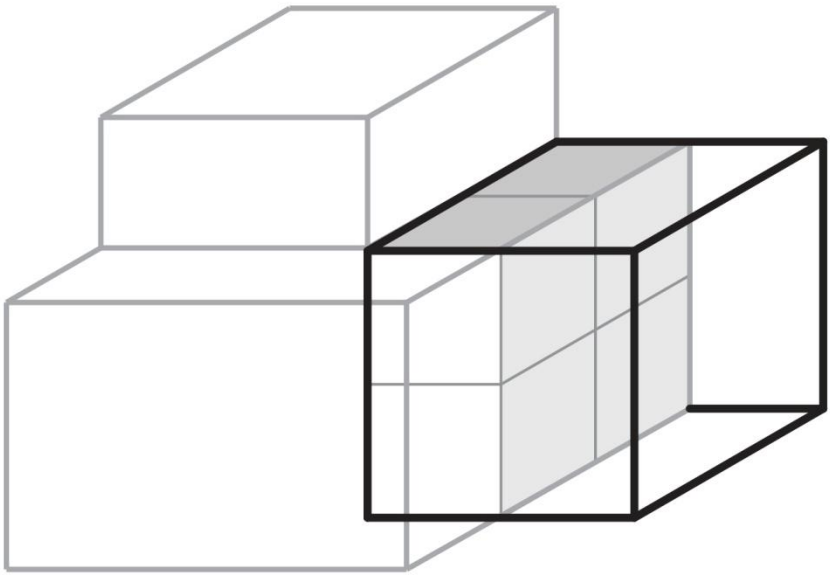
Approach 1: Marching Cubes

Iso-surface extraction



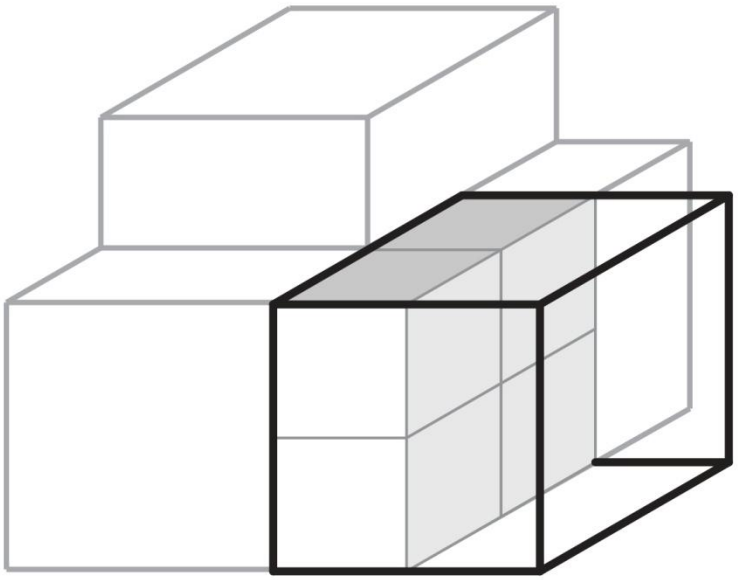
Approach 1: Marching Cubes

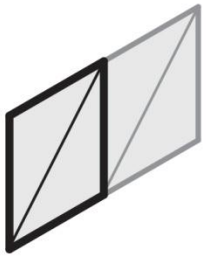
Iso-surface extraction



Approach 1: Marching Cubes

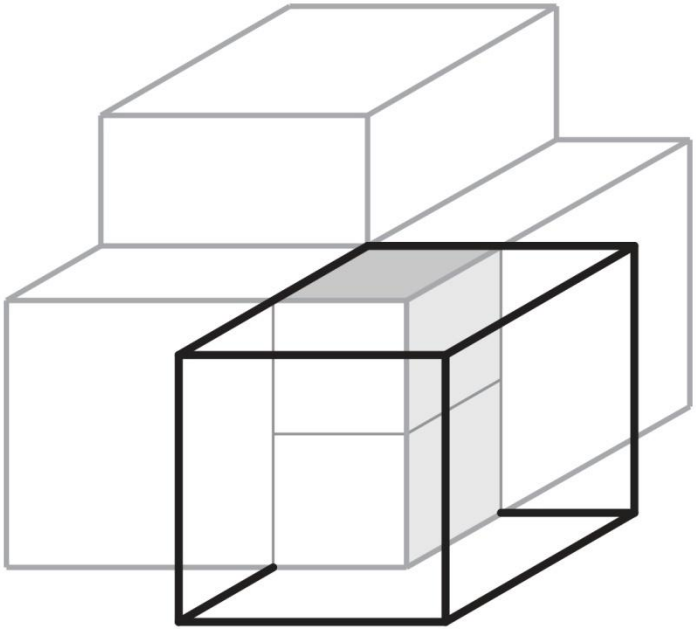
surface reconstruction

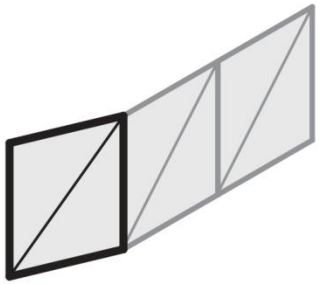




Approach 1: Marching Cubes

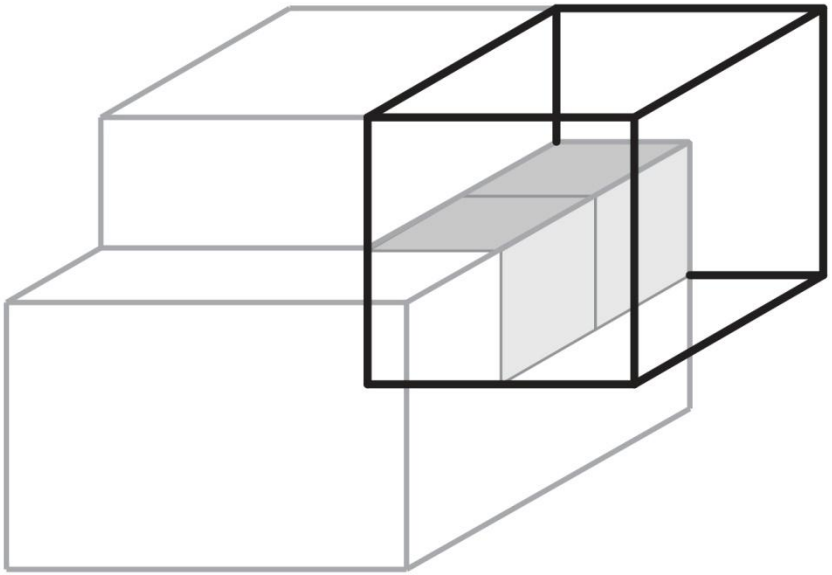
surface reconstruction

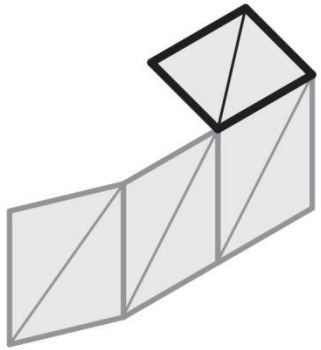




Approach 1: Marching Cubes

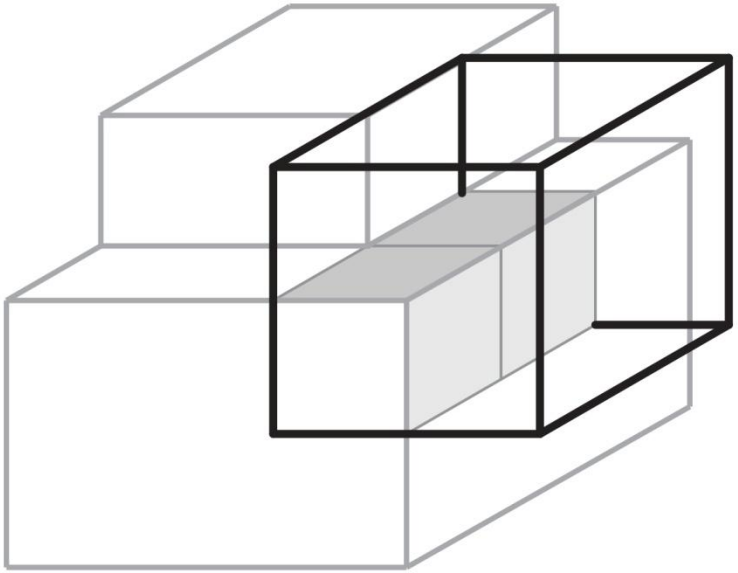
surface reconstruction

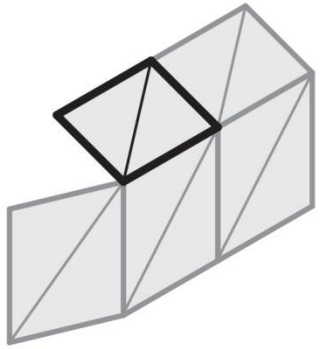




Approach 1: Marching Cubes

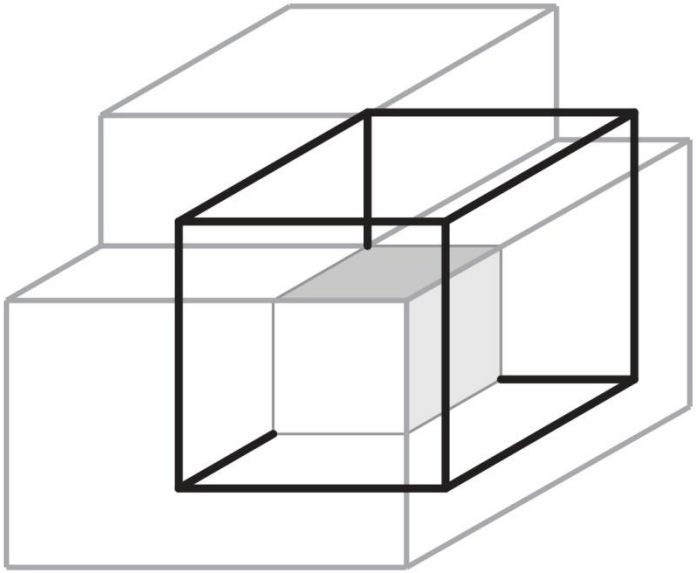
surface reconstruction

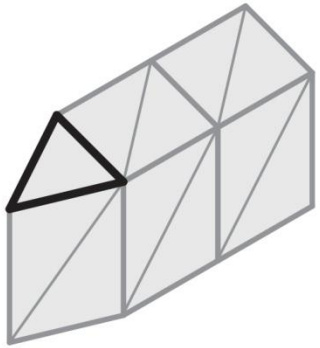




Approach 1: Marching Cubes

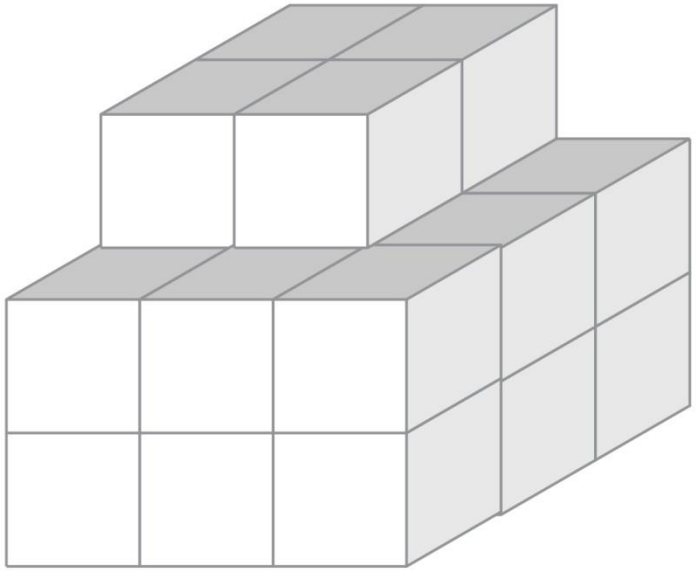
surface reconstruction

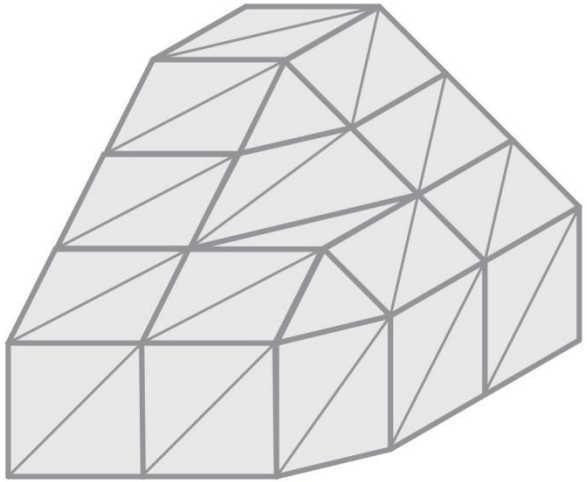




Approach 1: Marching Cubes

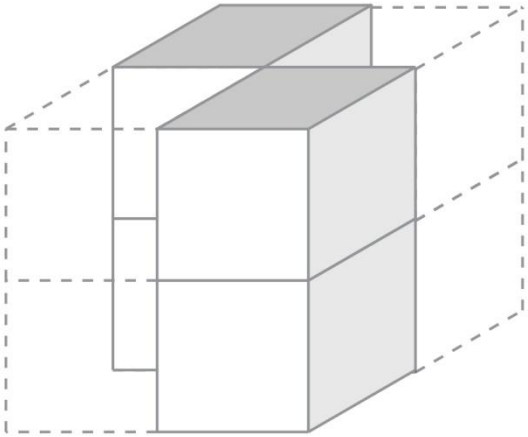
surface reconstruction

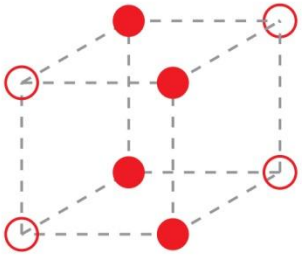


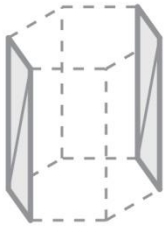


Approach 1: Marching Cubes

Iso-surface extraction



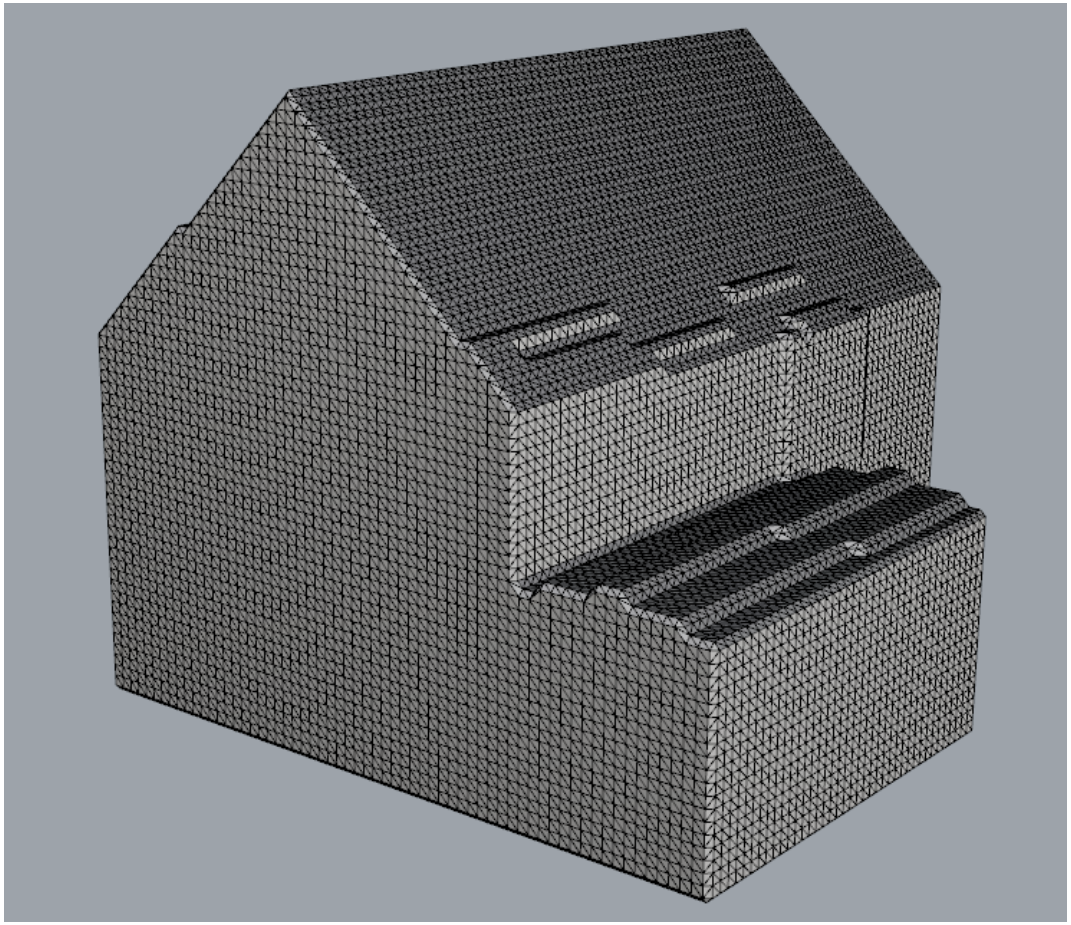


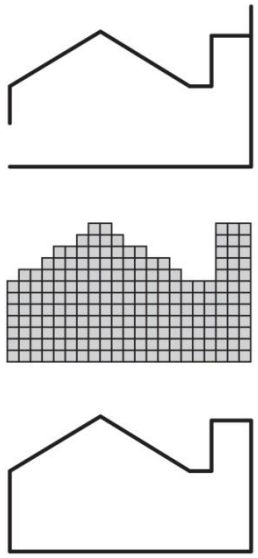


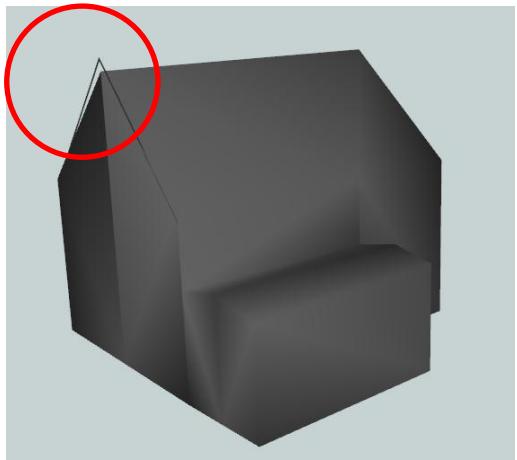
ambiguities possible

Approach 1: Marching Cubes

surface reconstruction

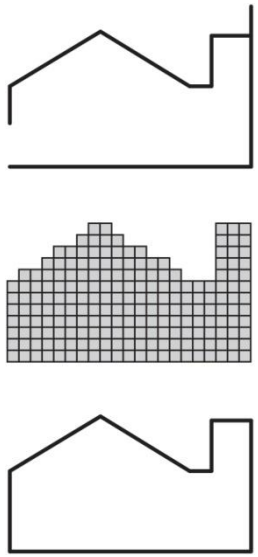


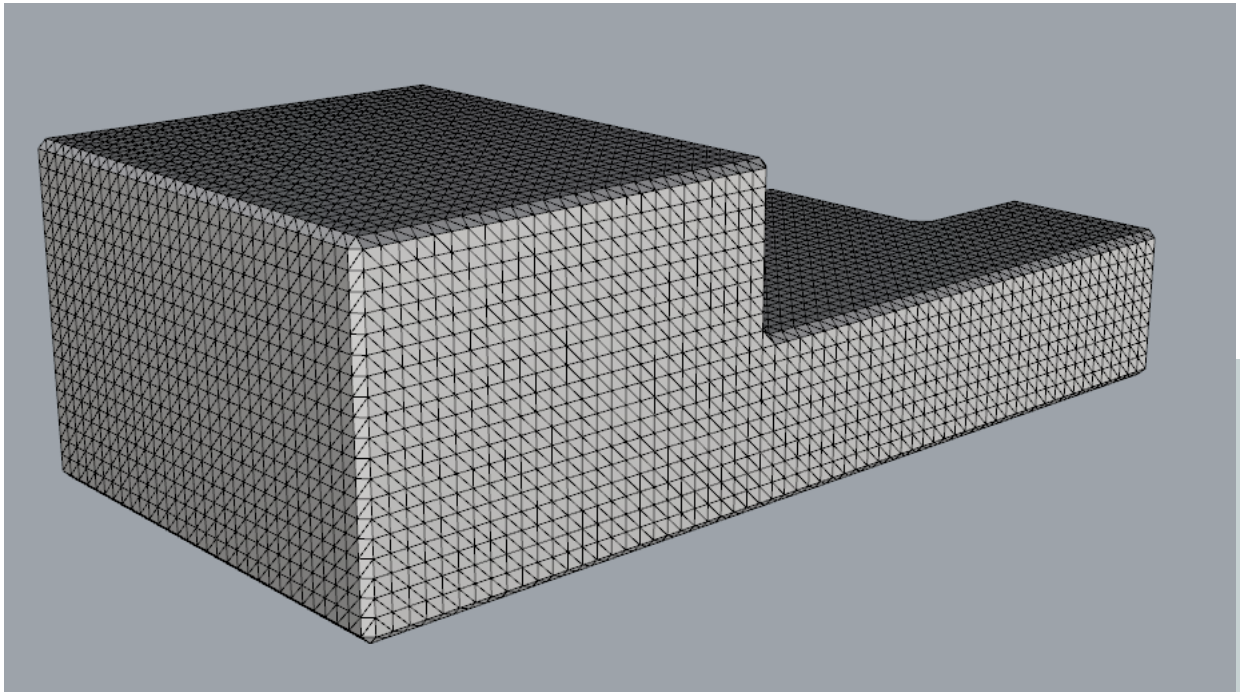


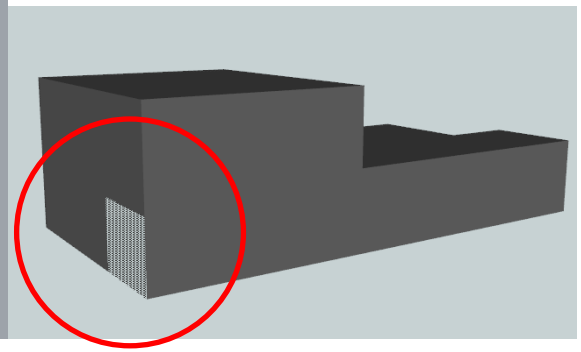


Approach 1: Marching Cubes

surface reconstruction

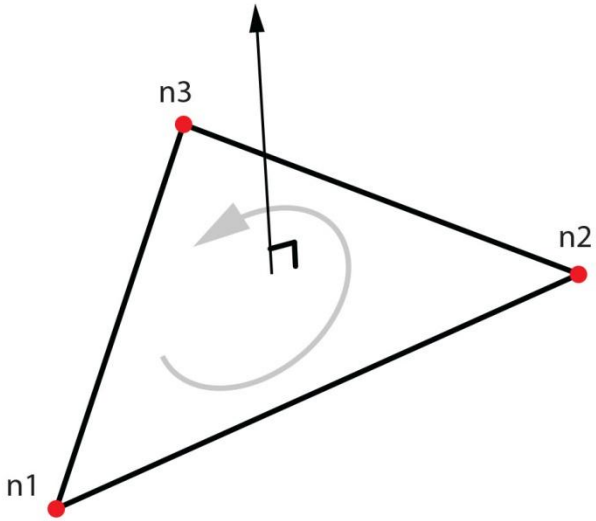


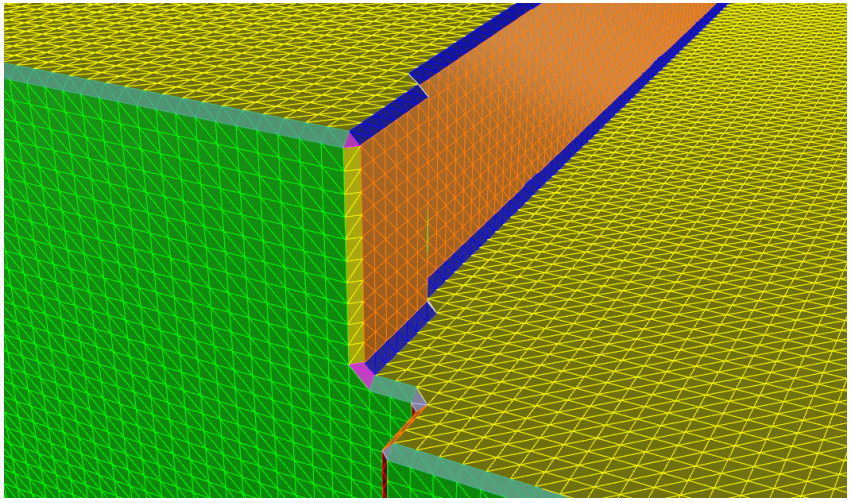




Approach 1: Marching Cubes

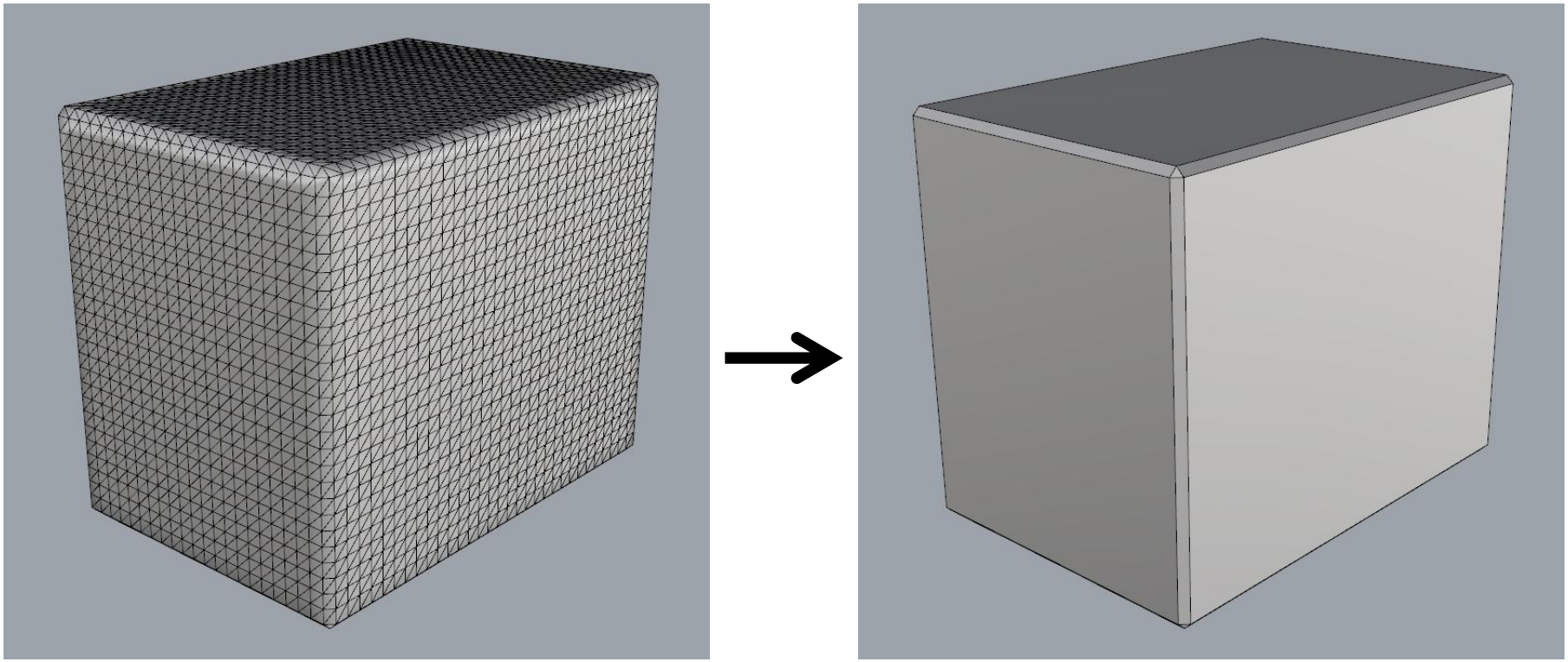
detriangulation





Approach 1: Marching Cubes

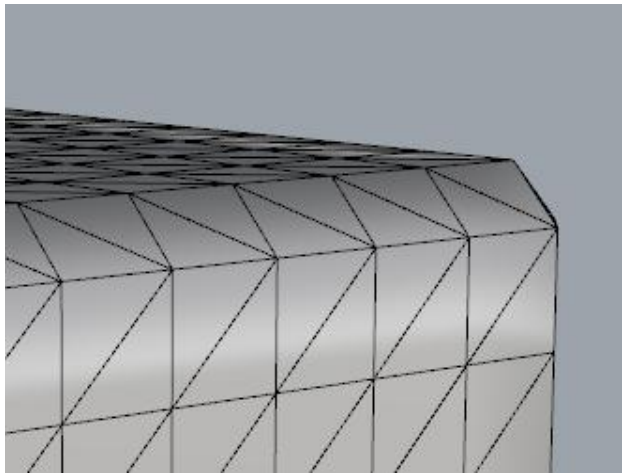
detriangulation

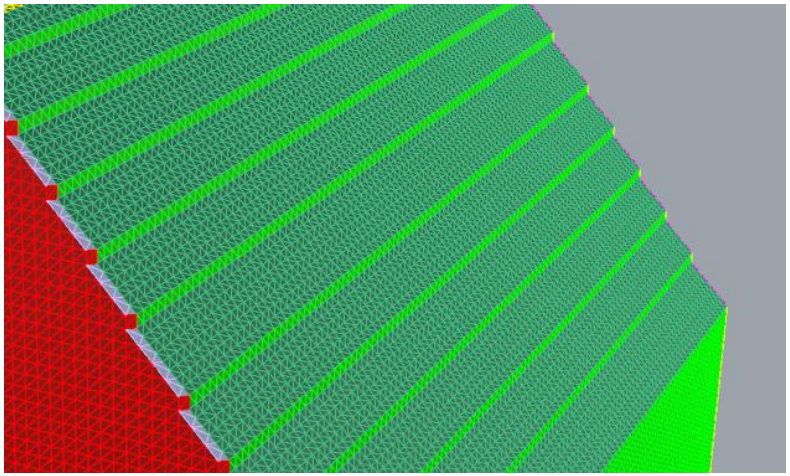


From many triangles to a few polygons

Approach 1: Marching Cubes

two drawbacks

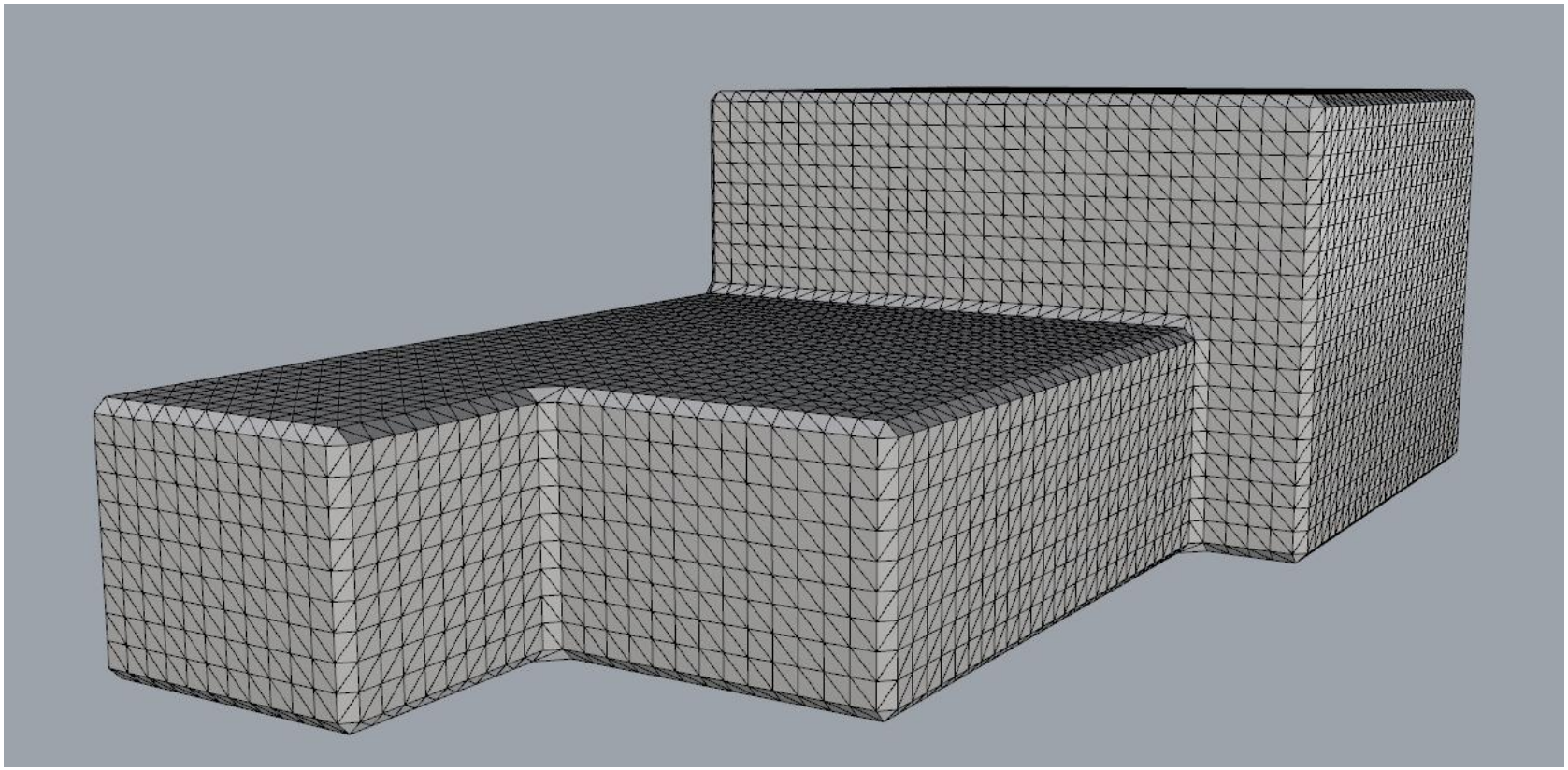




corners rounded off & stair stepping effect

Approach 1: Marching Cubes

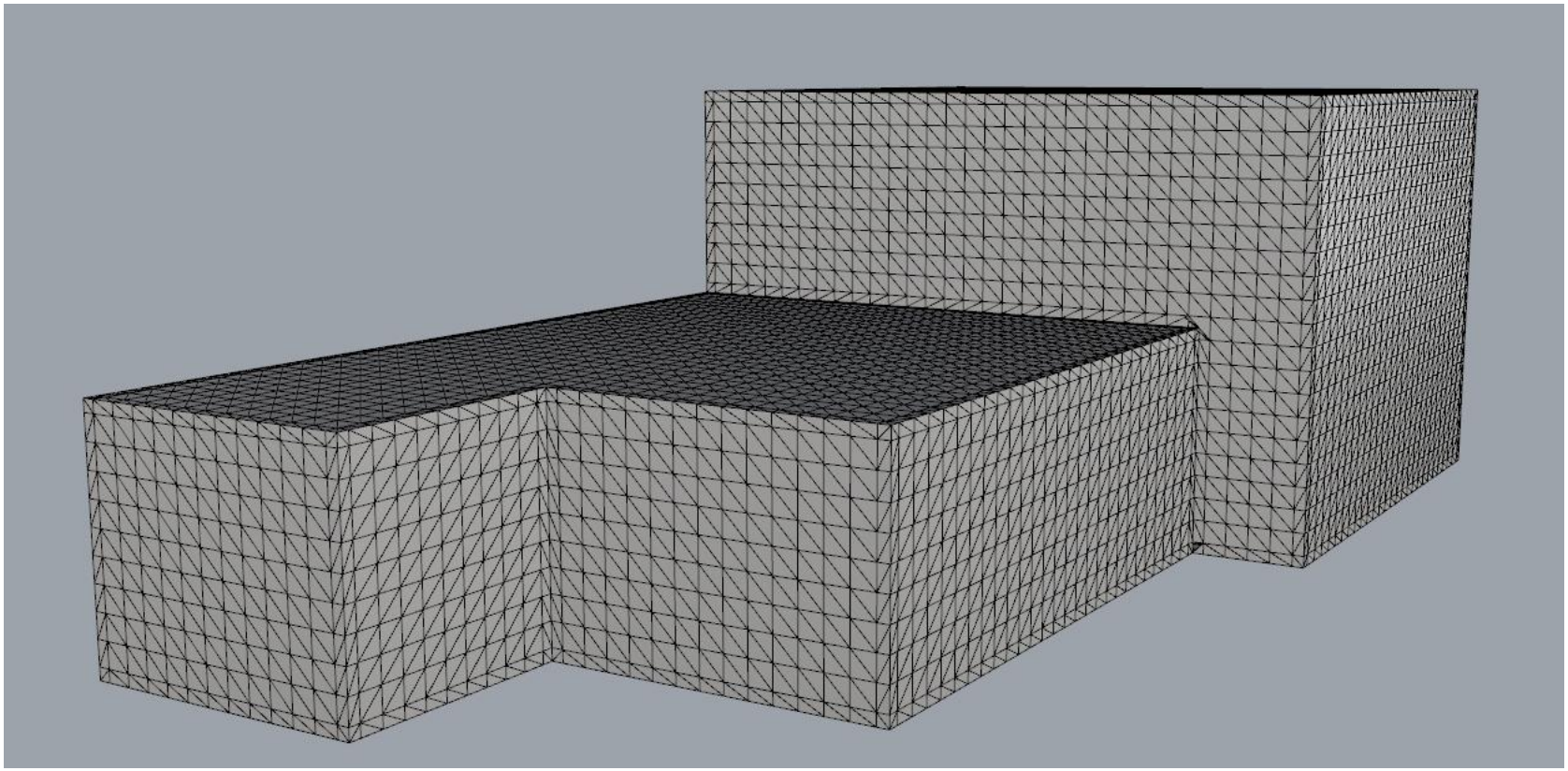
Edge sharpening



Edge sharpening algorithm by Attene (2003)

Approach 1: Marching Cubes

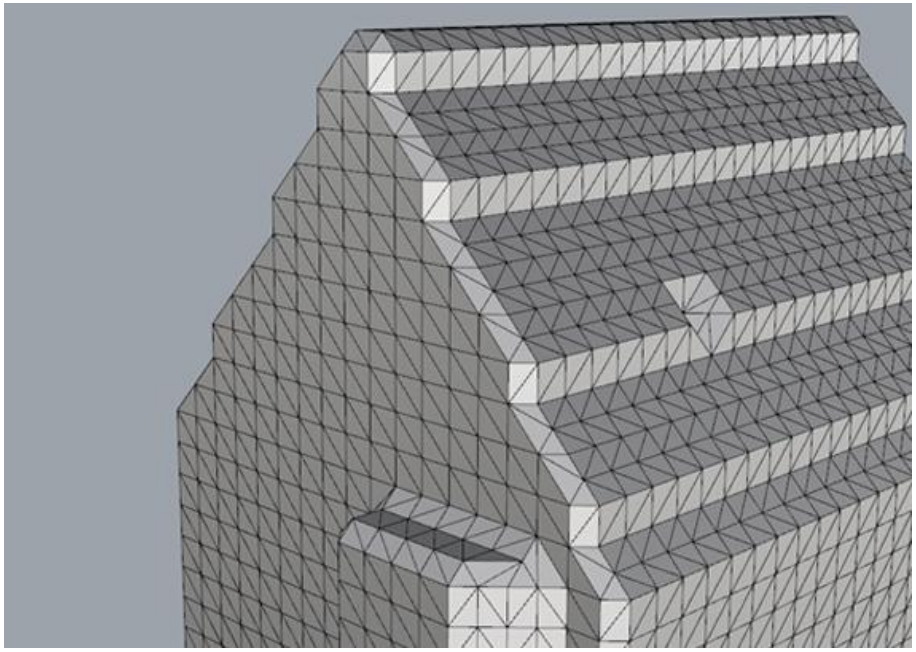
Edge sharpening

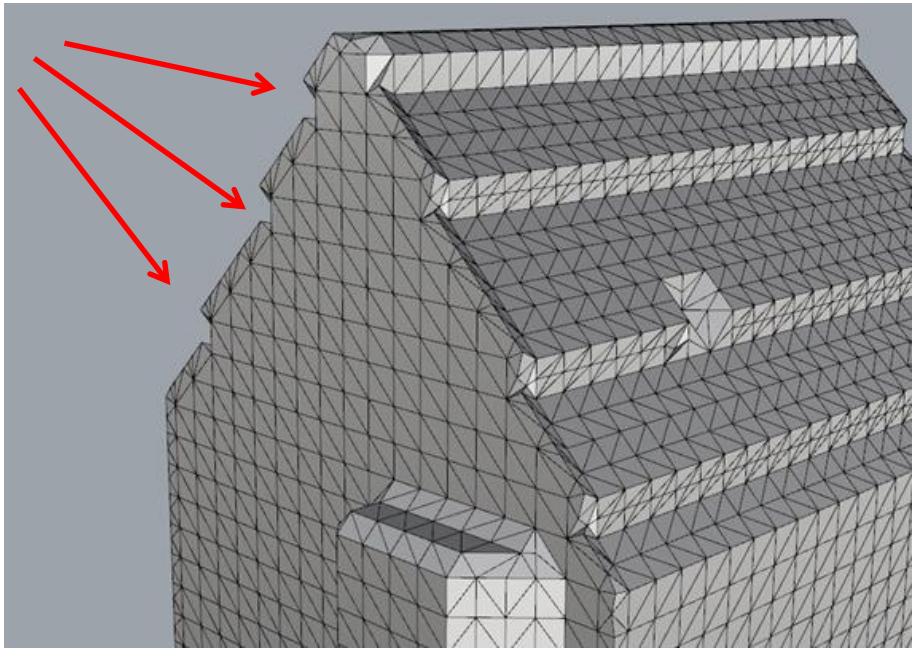


Edge sharpening algorithm by Attene (2003)

Approach 1: Marching Cubes

Edge sharpening



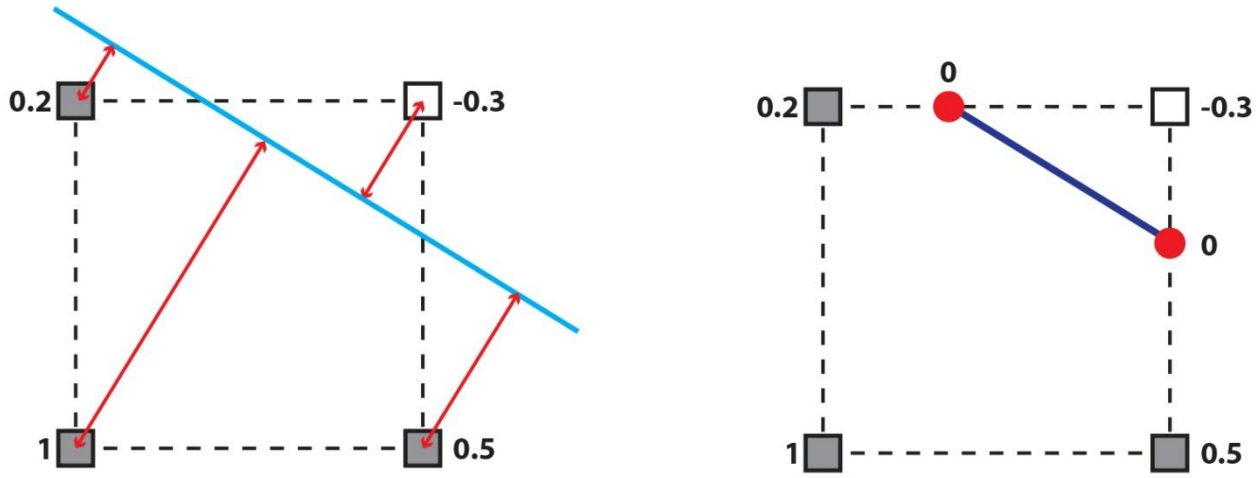


Stair-stepping effect is sharpened... which is not desirable

Edge sharpening algorithm by Attene (2003)

Approach 1: Marching Cubes

Distance field



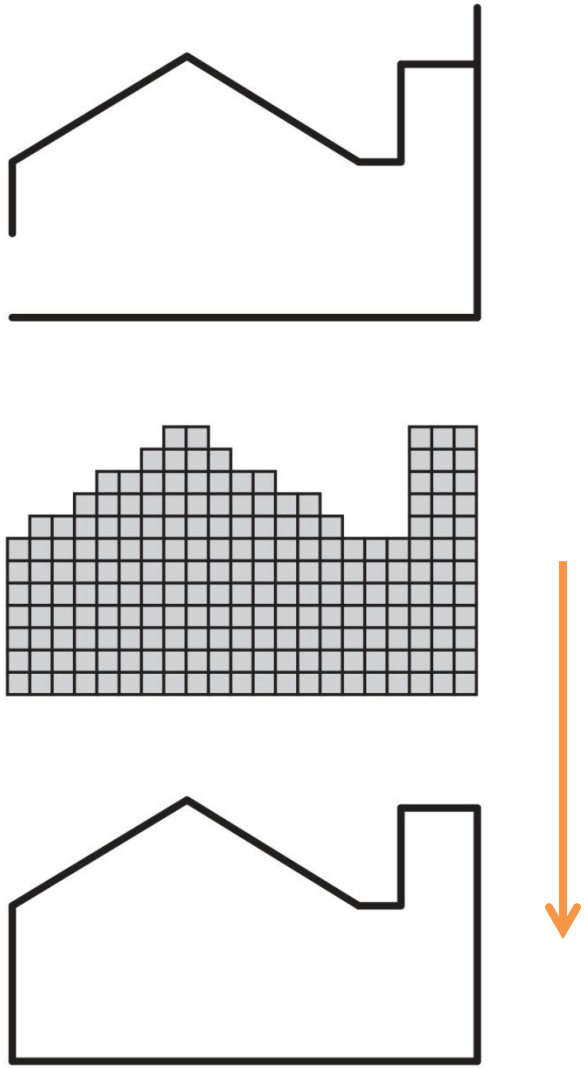
Stair stepping is avoided when using a signed distance field

Approach 1: Marching Cubes

evaluation

- ✓ Decent repair capability
- ✓ Almost no exceptions in resulting mesh
- ✓ Overshoots, gaps & self-intersections can be repaired

- Shift in geometry
- Corners rounded (or added risk in edge sharpening)
- Tilted surfaces approximated with stair stepping

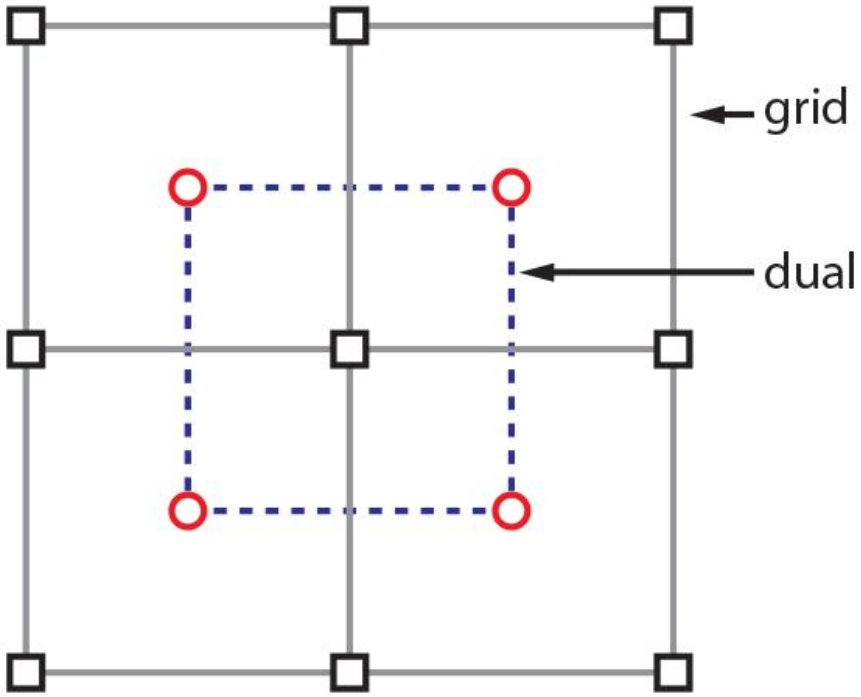


Surface reconstruction:

Approach 2: Dual Contouring

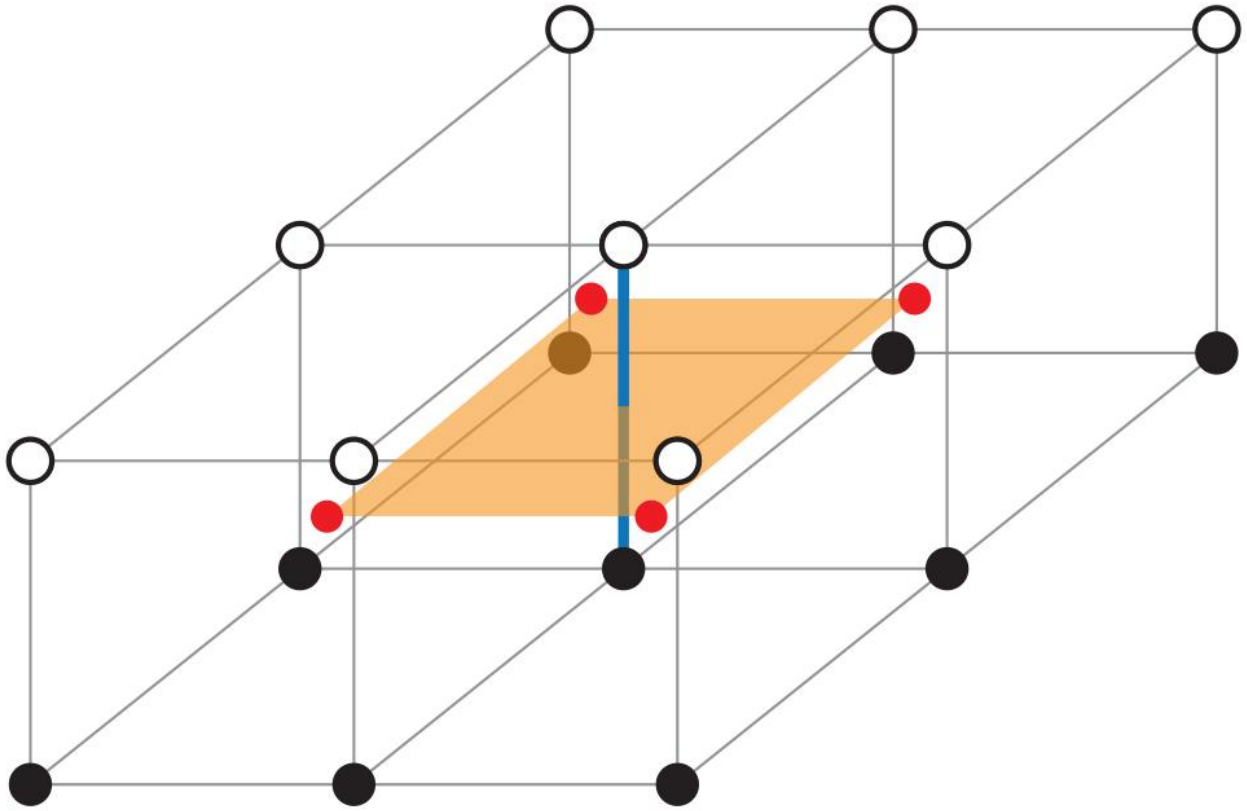
Approach 2: Dual Contouring

Principle



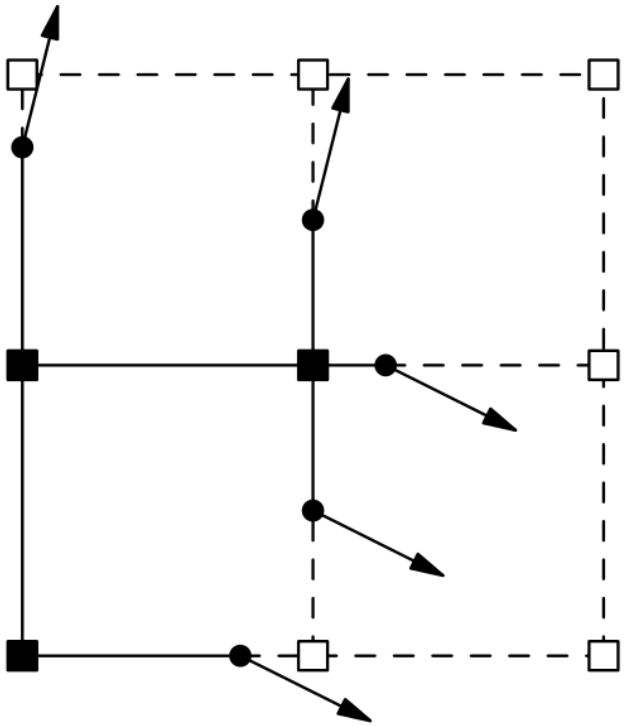
Approach 2: Dual Contouring

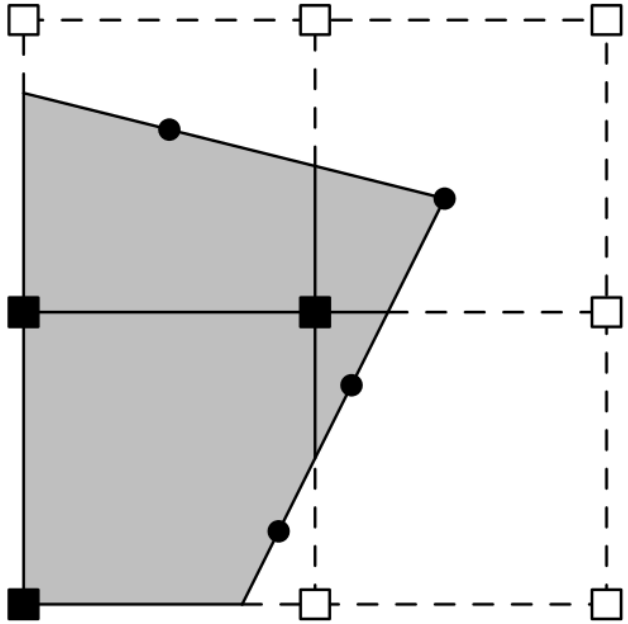
Principle



Approach 2: Dual Contouring

Principle





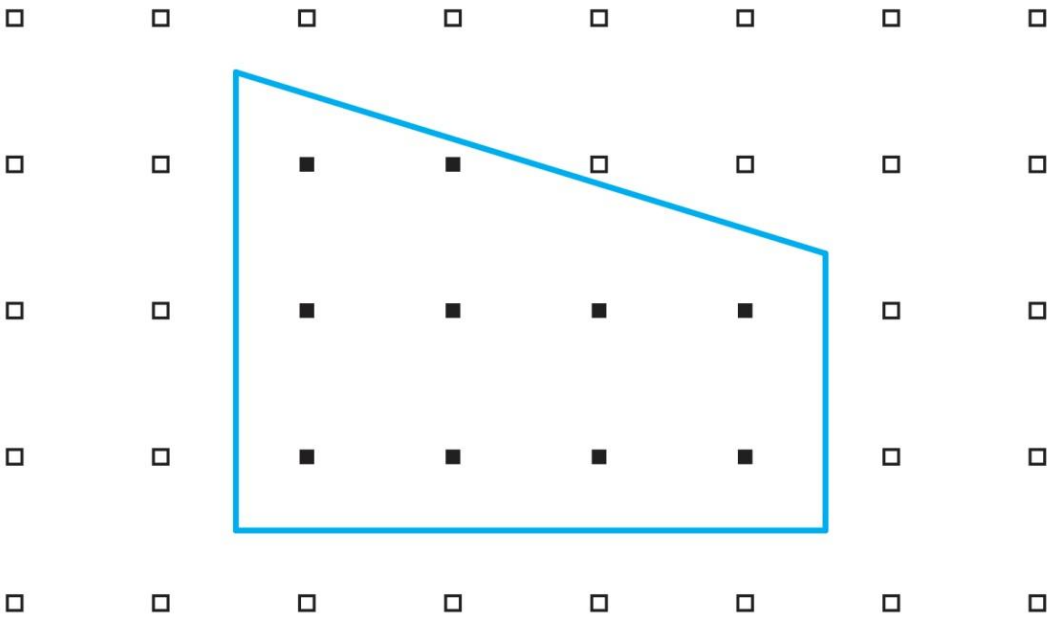
input = intersection + normal vector
minimizing the Quadratic Error Function (QEF):

$$E|x| = \sum (n_i \cdot (x - p_i)^2)$$

(Ju et al., 2002)

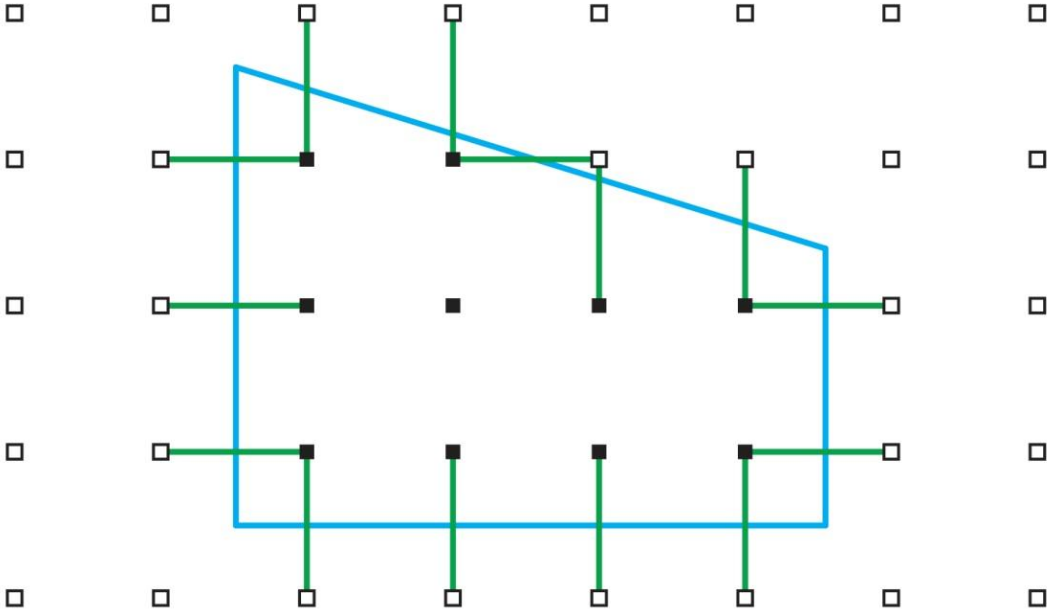
Approach 2: Dual Contouring

Principle



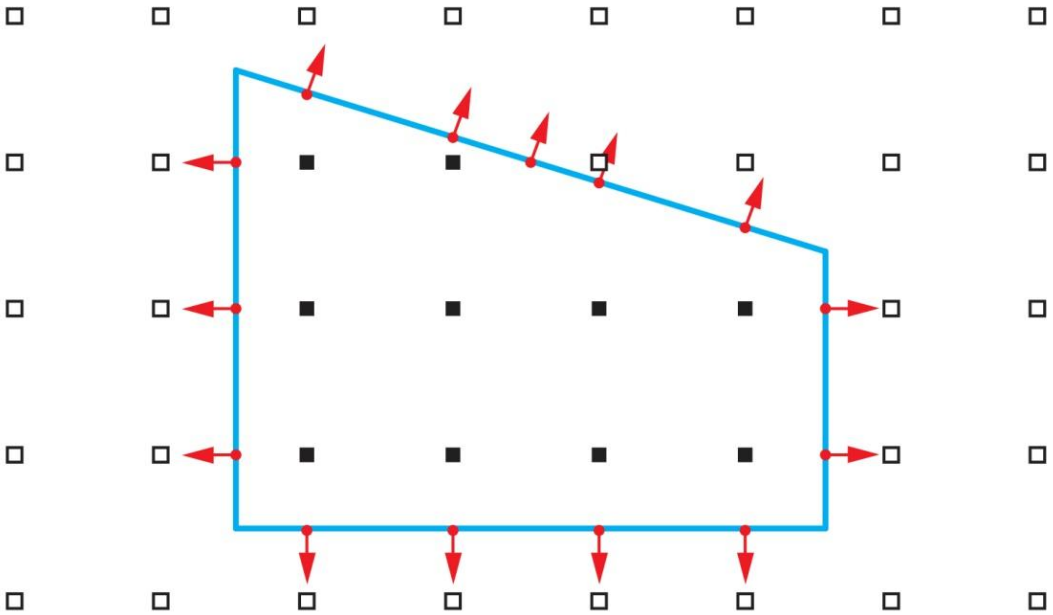
Approach 2: Dual Contouring

Principle



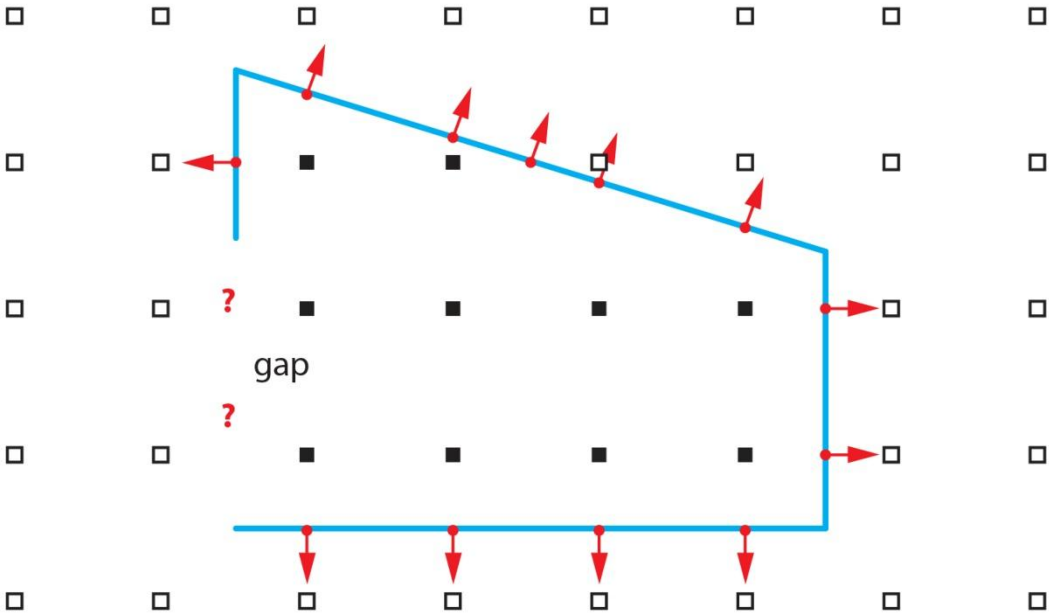
Approach 2: Dual Contouring

Principle



Approach 2: Dual Contouring

Principle

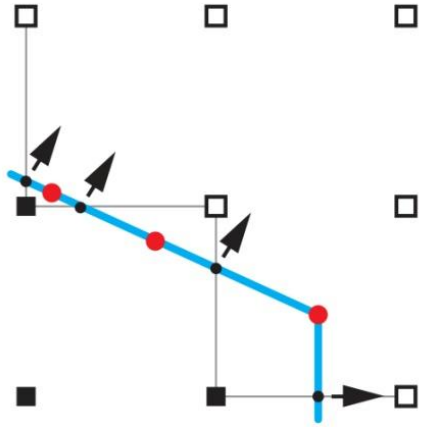


In case of gap **missing intersections**

Approach 2: Dual Contouring

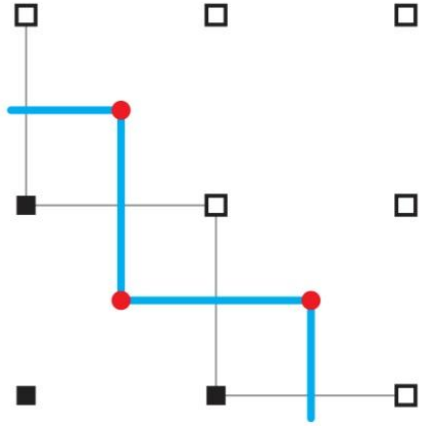
Two ways of computing dual vertex

All intersections:



1. QEF

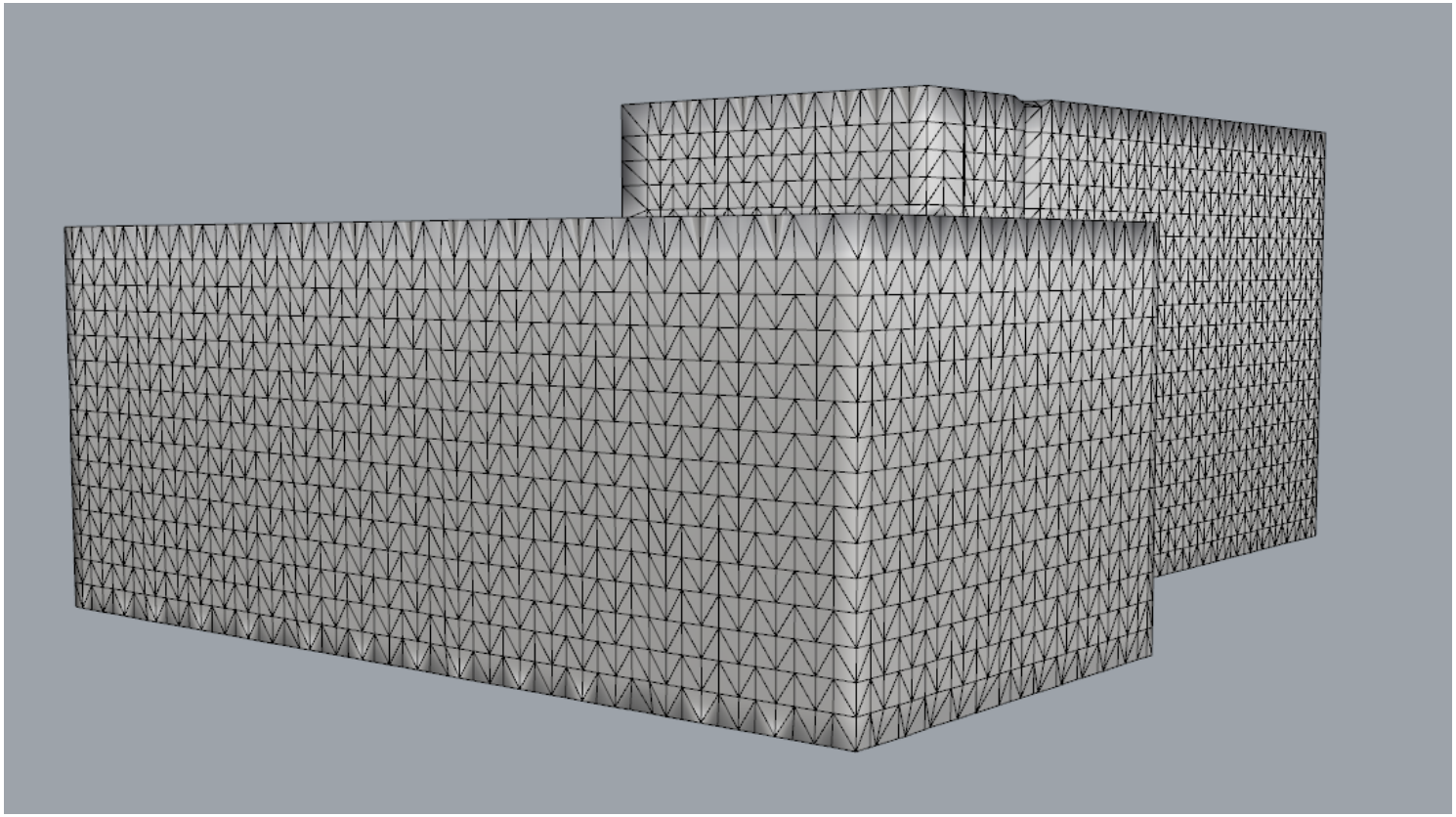
not all intersections:



2. Cube center

Approach 2: Dual Contouring

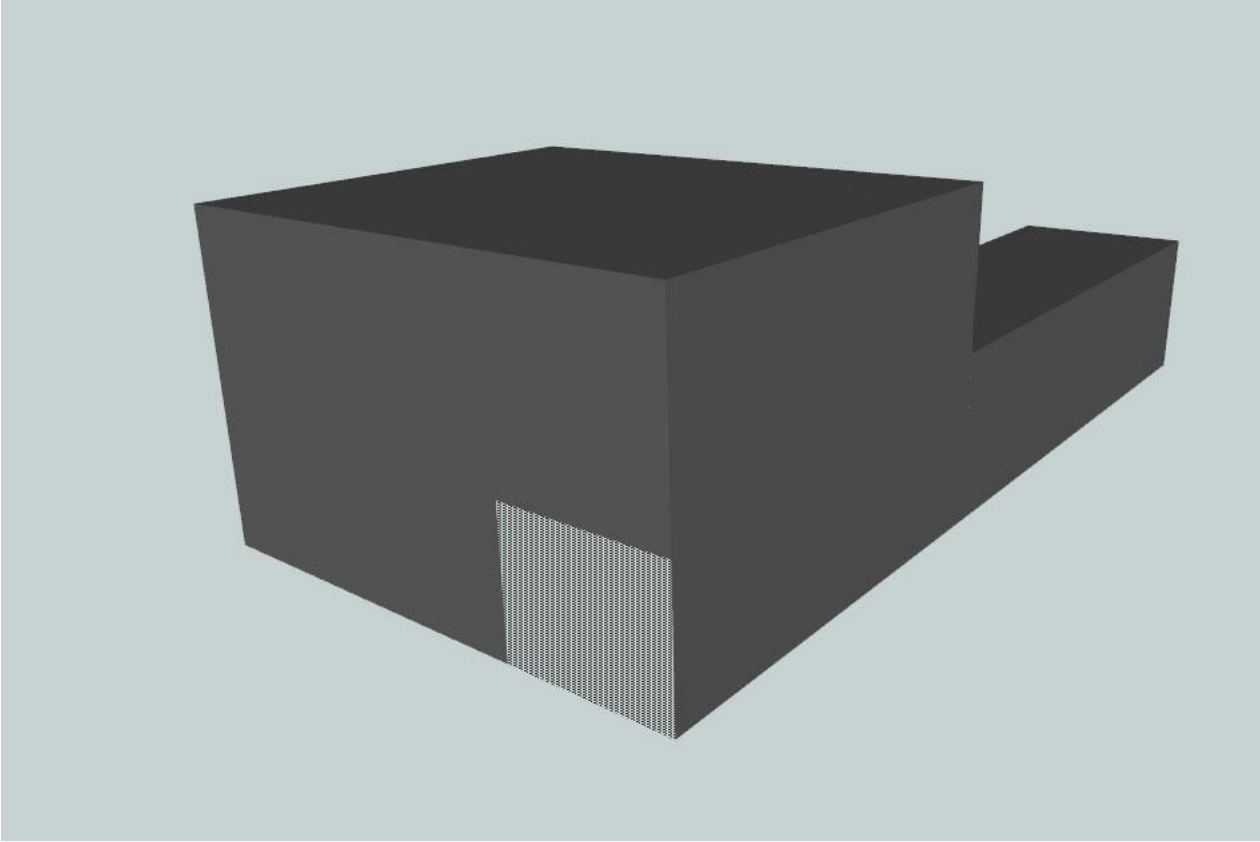
example



No shift in geometry & **sharp features**

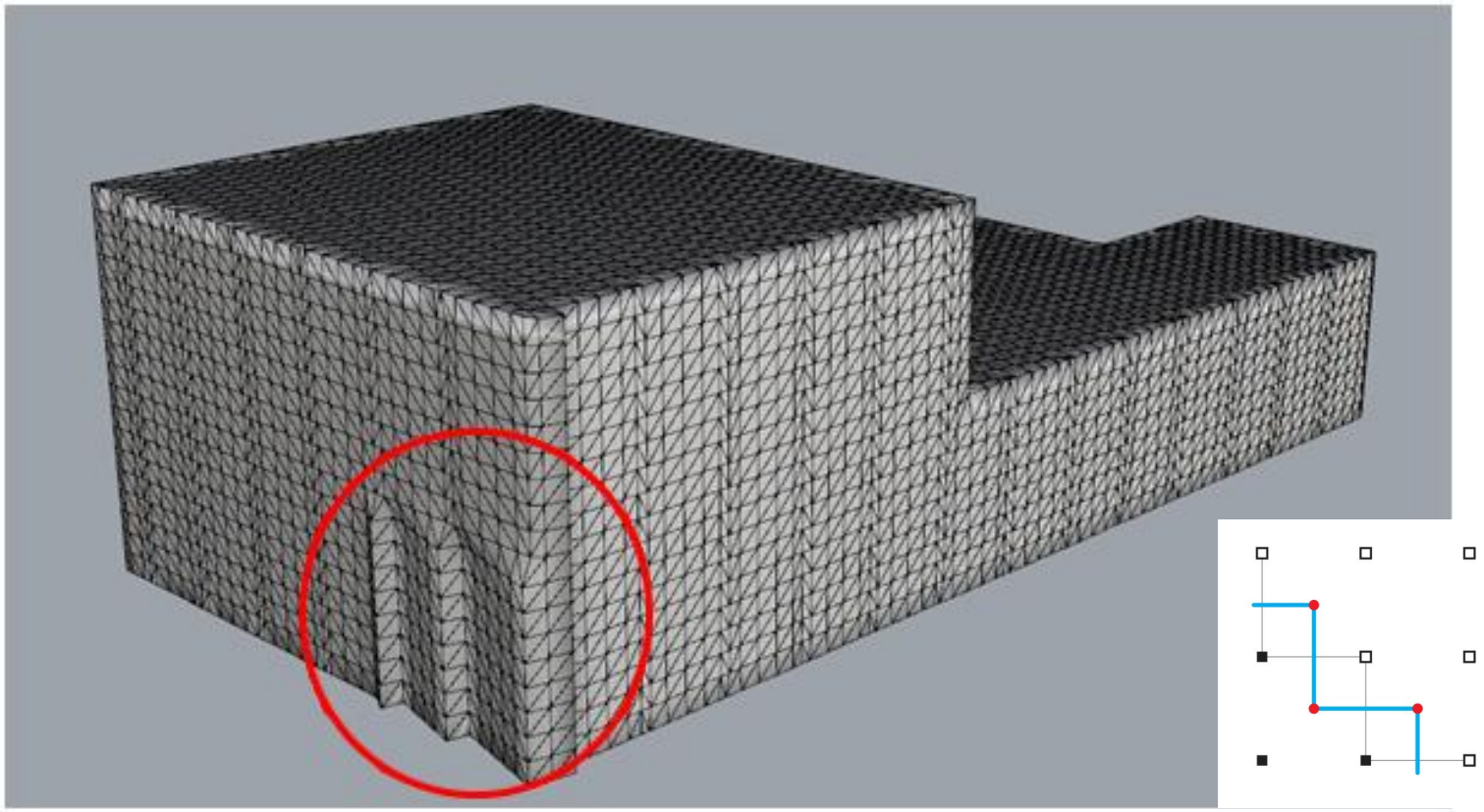
Approach 2: Dual Contouring

3D model with gap



Approach 2: Dual Contouring

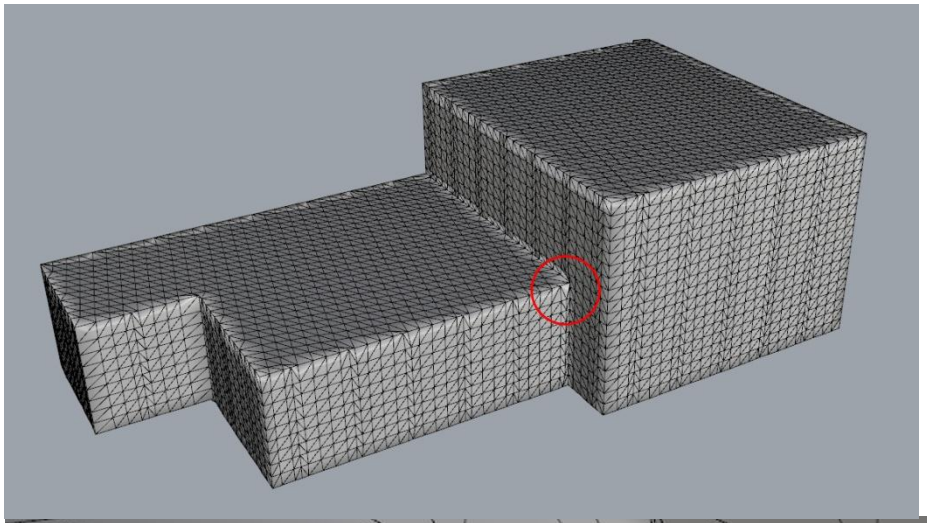
Result on gap

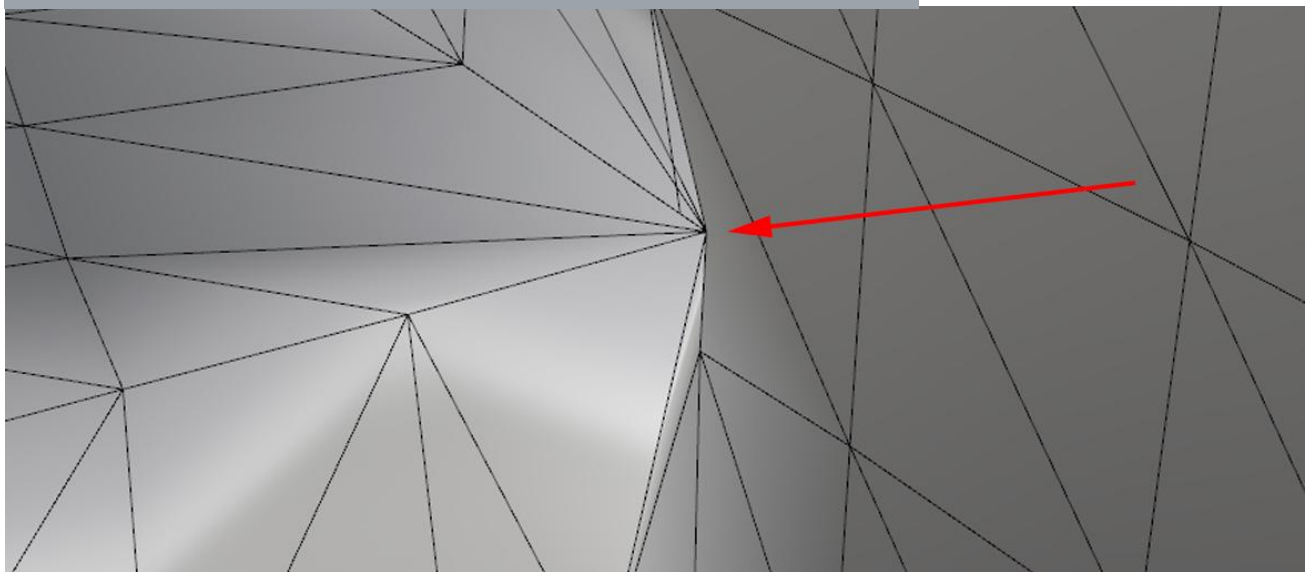


cube center assigned!

Approach 2: Dual Contouring

Issue: self-intersections!





Approach 2: Dual Contouring

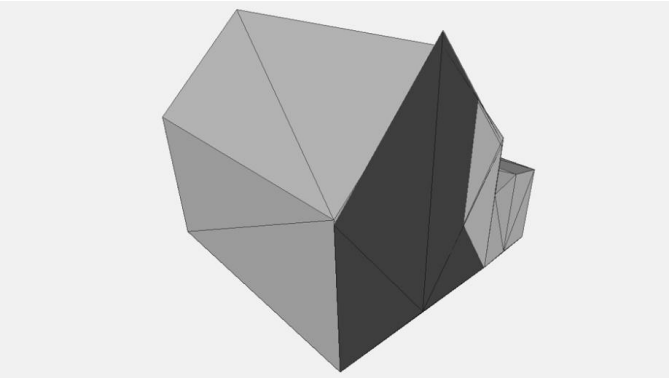
evaluation

- ✓ Sharp features reconstructed
- ✓ Oblique surfaces reconstructed
- ✓ Overshoots can be repaired

- Self-intersections in output (for regular dual contouring)
- Sometimes issues concerning the QEF computation
- Visible artefacts when no intersections are found

Implementation

Test 20 buildings



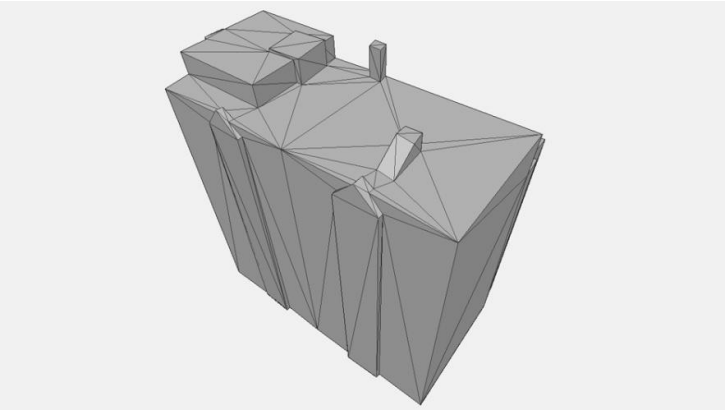
Non-manifold edge

Approach 1: Marching Cubes

✓18/20 repaired

Approach 2: Dual Contouring

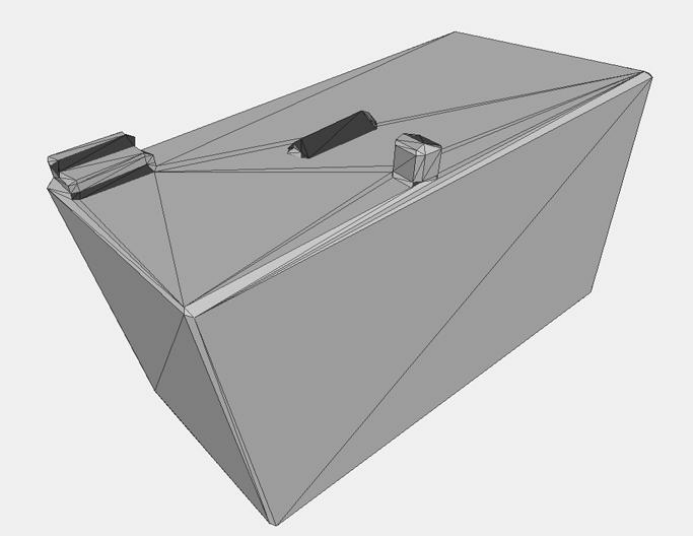
✓6/20 repaired

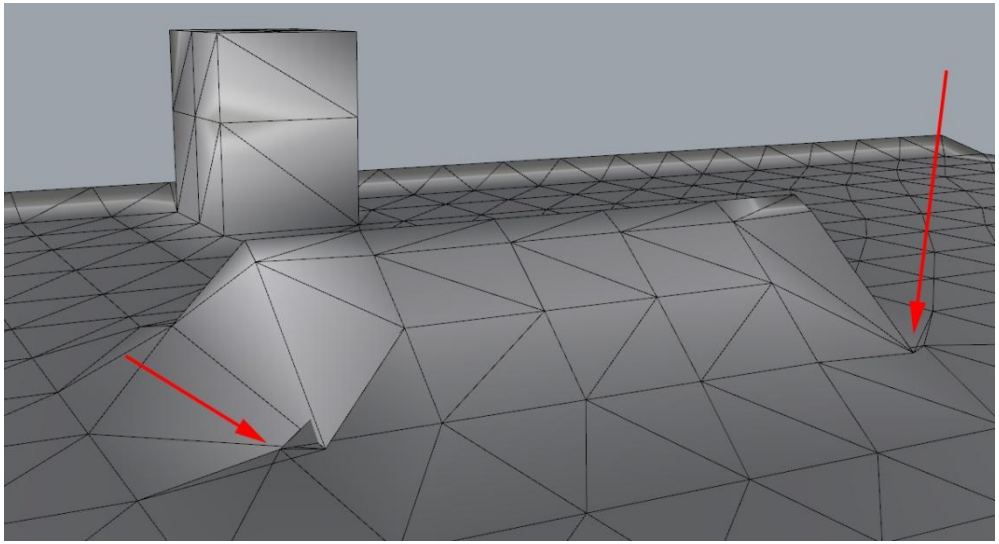


Same consecutive points

Implementation

Test 20 buildings





Approach 1: Marching Cubes

✓ repaired

Approach 2: Dual Contouring

self-intersection

Rotterdam Heijplaat

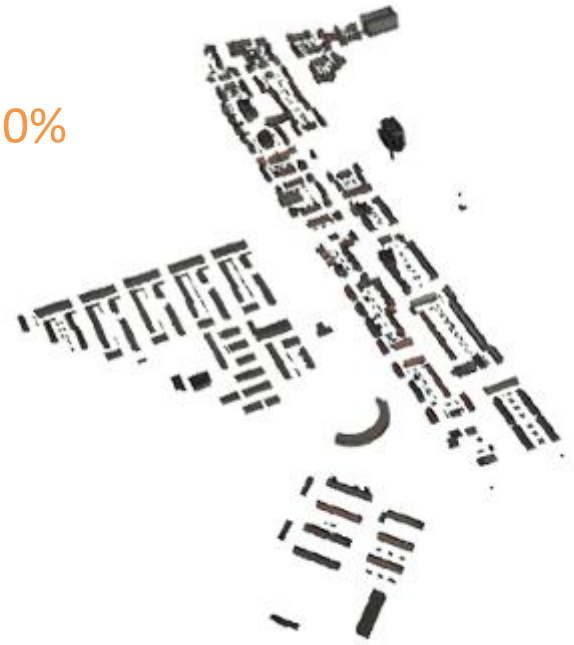

Rotterdam Heijplaat
1207 buildings
116 valid (10 %)

Implementation

Rotterdam Heijplaat dataset

Existing dataset

116/1207 repaired = 10%

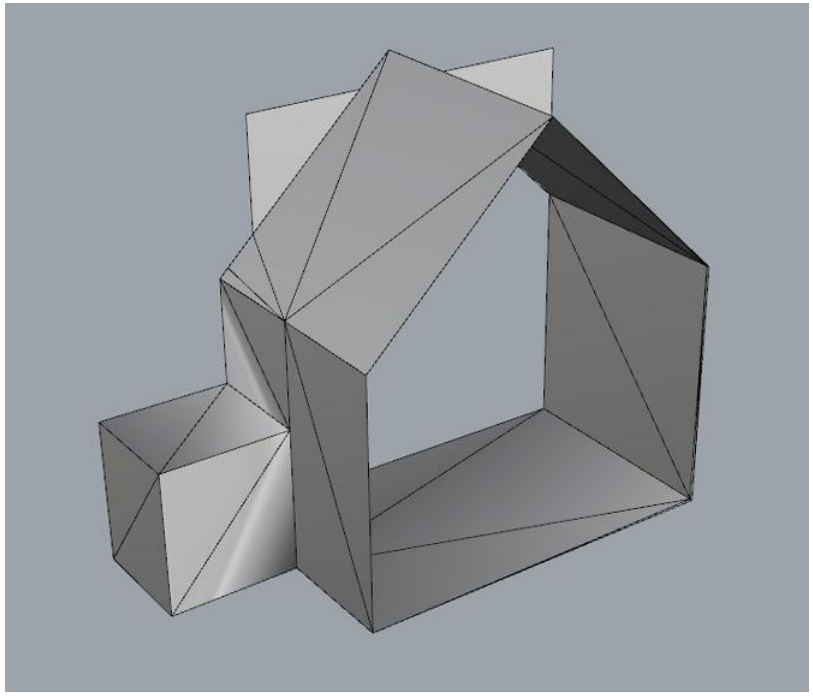


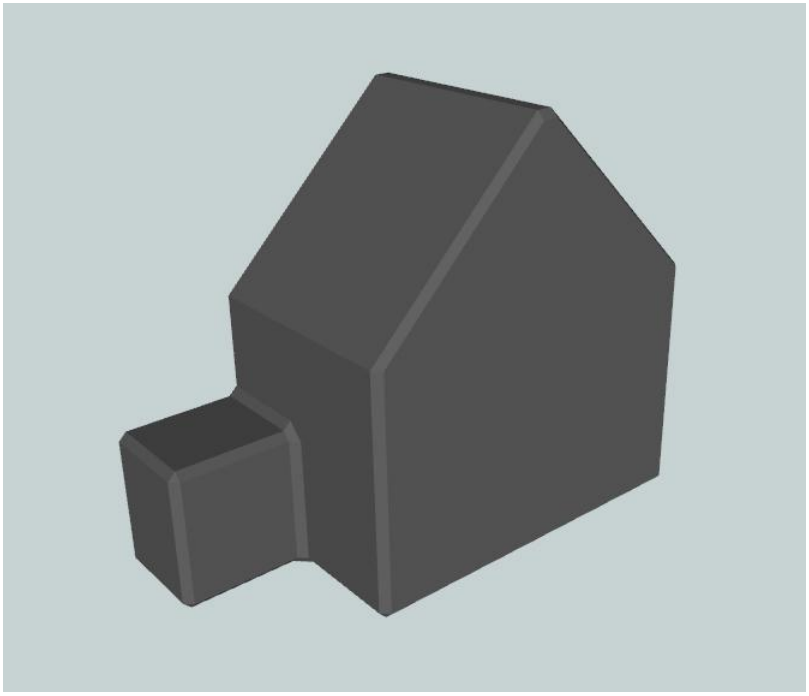
Approach 1: Marching Cubes

✓1159/1207 repaired = 96%

Implementation

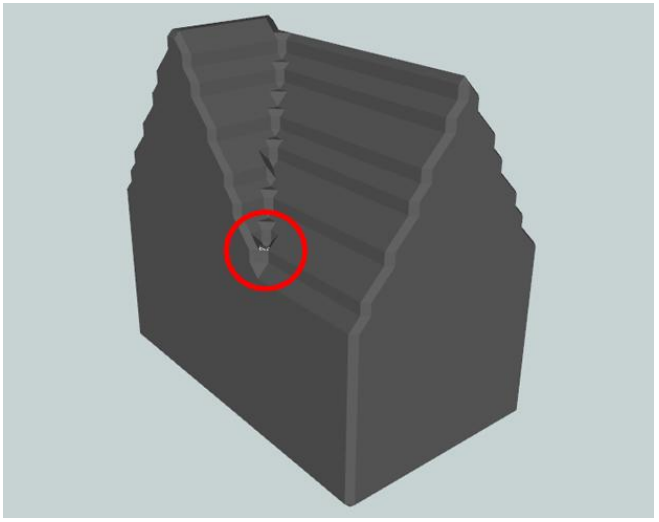
Rotterdam Heijplaat dataset

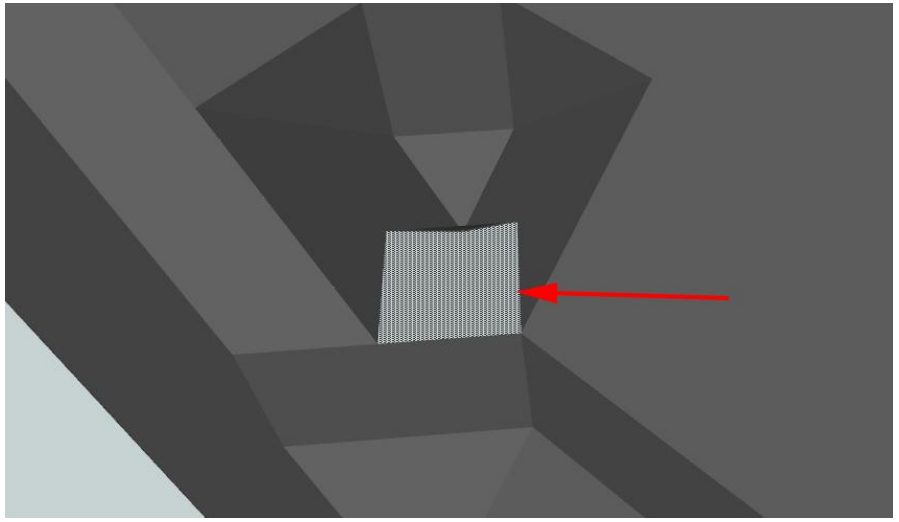




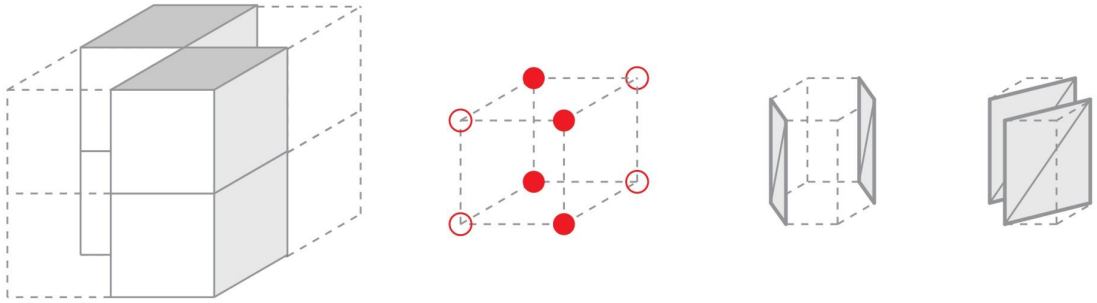
Implementation

Rotterdam Heijplaat dataset



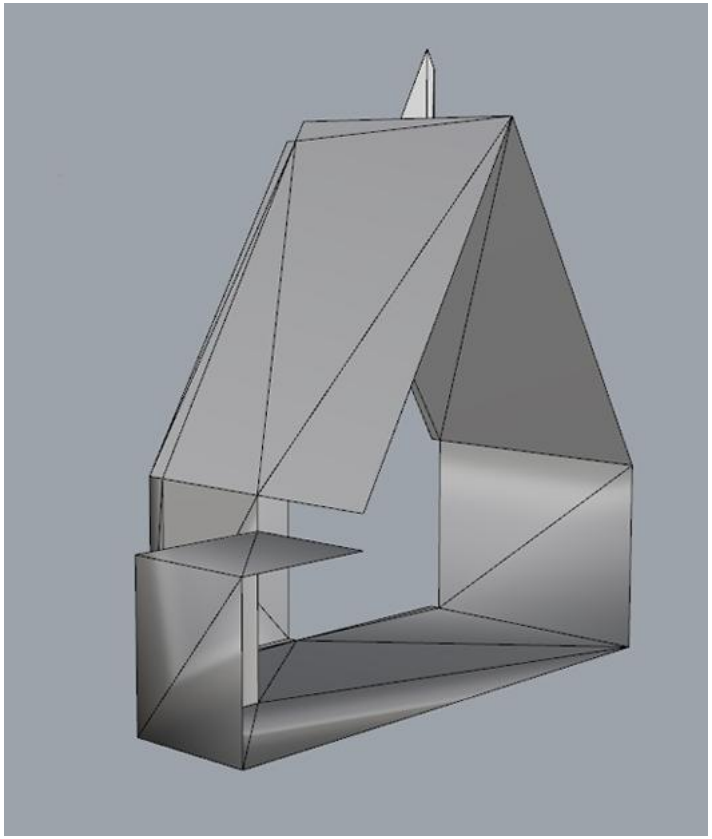


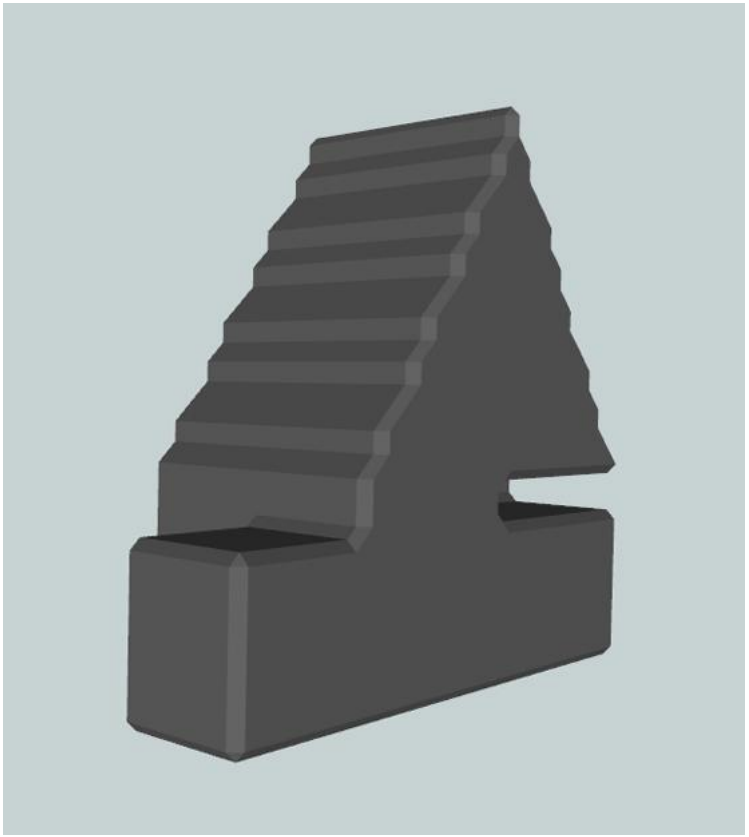
Repair failed due to ambiguity



Implementation

Rotterdam Heijplaat dataset





Valid output but uwanted result (~5%?)

Conclusion

To which extent is it possible to
automatically repair a geometrically invalid
3D City Building Model using a voxel-based method?

Conclusion

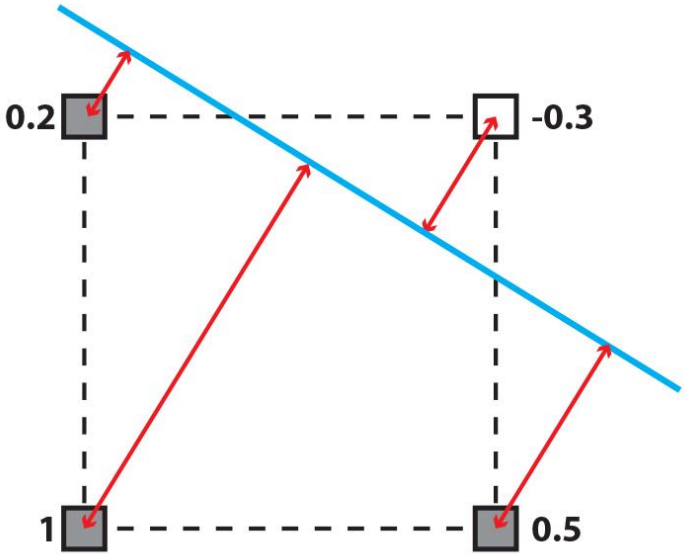
- **Voxelization:**
  Correct building volumes can be almost guaranteed

- **Approach 1:** Marching Cubes
  Strong repair capability but:
  - geometry shifted
  - stair stepping
  - corners rounded

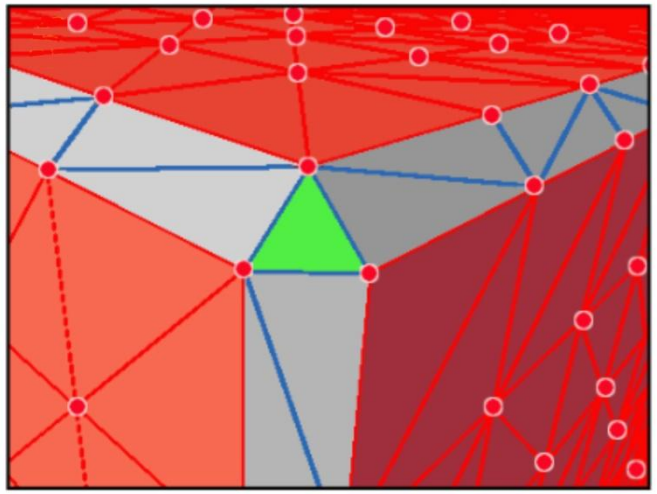
- **Approach 2:** Dual Contouring
  Sharp features & oblique surfaces reconstructed but:
  - contains self-intersections
  - artefacts may be created

Future work

Distance field + edge sharpening



+

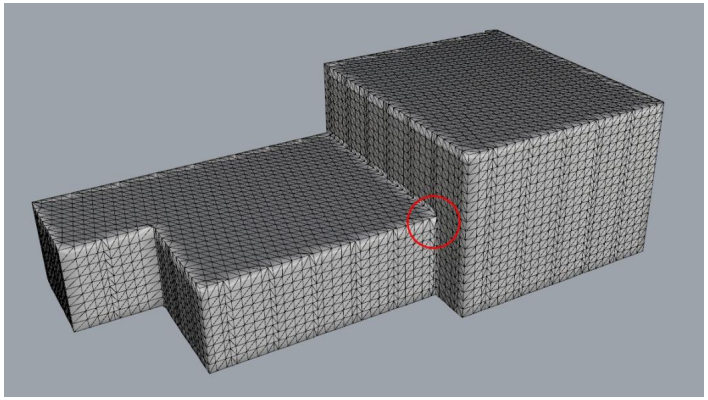


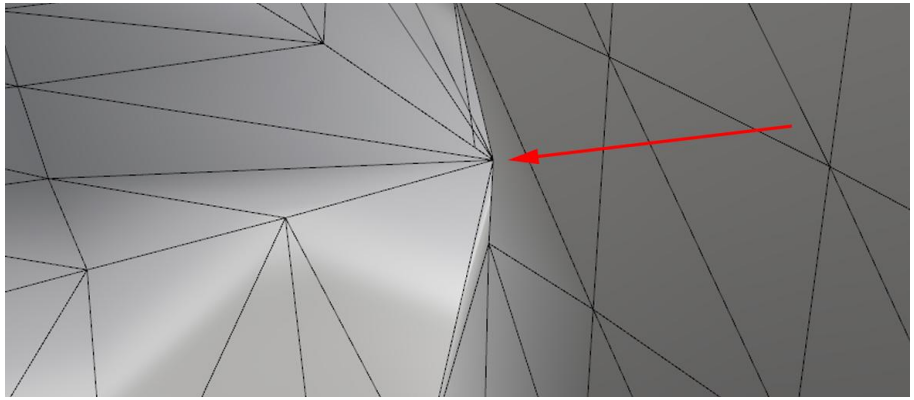
Approach 1: Marching Cubes

solution for oblique surfaces and sharp features

Future work

Dual contouring non-intersecting





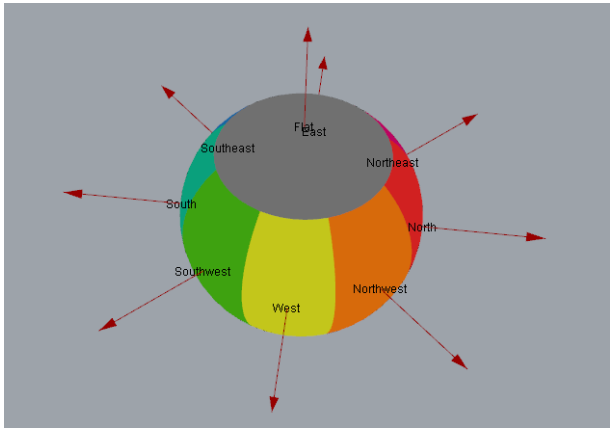
Dual Contouring (Ju et al., 2002 & Schaefer et al. 2002)
Known for **self-intersections**

Isosurfaces over simplicial partitions of multiresolution grids (Schaefer et al., 2010)
manifold & self-intersection free adaptation of Dual Contouring

Approach 2: Dual Contouring
adaption to avoid self-intersections

Future work

1. Geometry based attribute assumptions

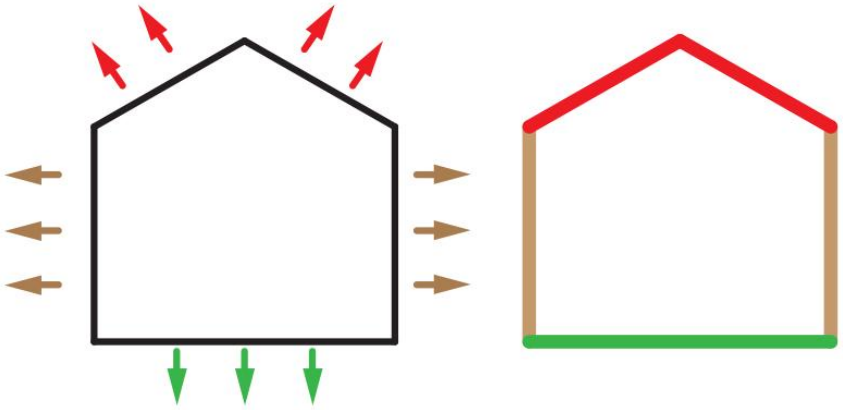


(3D modelling, P. Nourian, 2014)

TABLE 2.1: Normal vector limitations on semantic surfaces

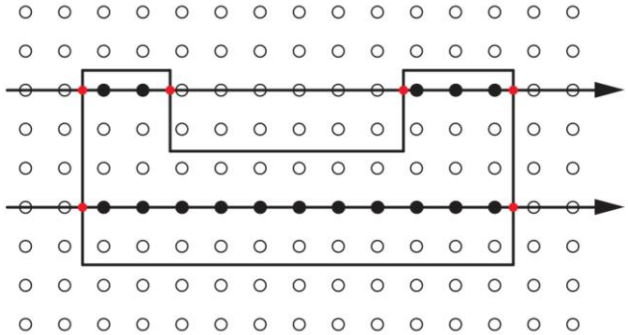
Surface type	Allowed direction(s)
WallSurface	All
RoofSurface	All
ClosureSurface	All
GroundSurface	Only down
OuterCeilingSurface	Only down
OuterFloorSurface	Only up
Opening	All

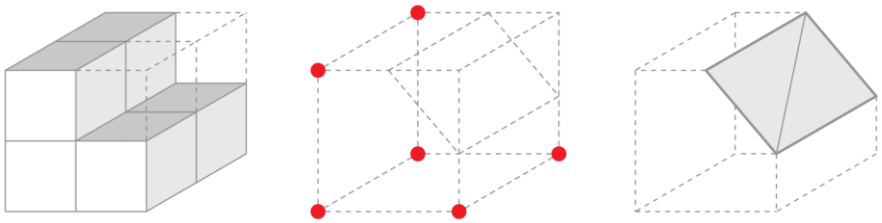
(S. Donkers, 2013)



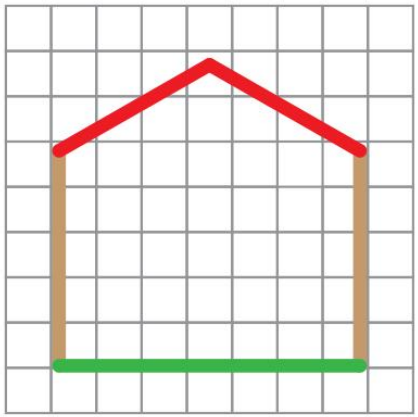
Future work

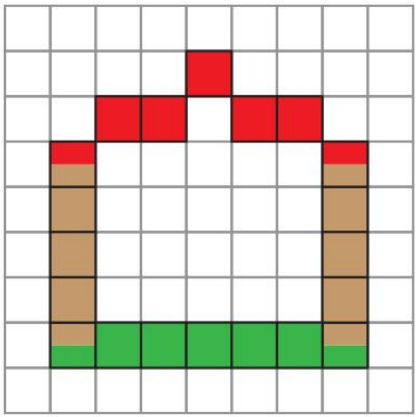
2. Voxel based atttribute preservation





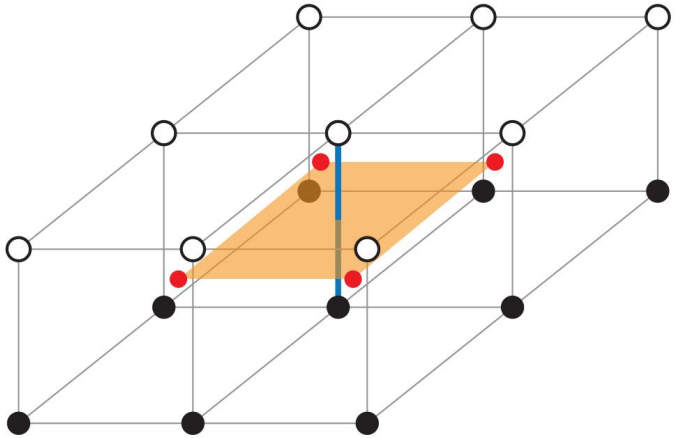
Approach 1: Marching Cubes



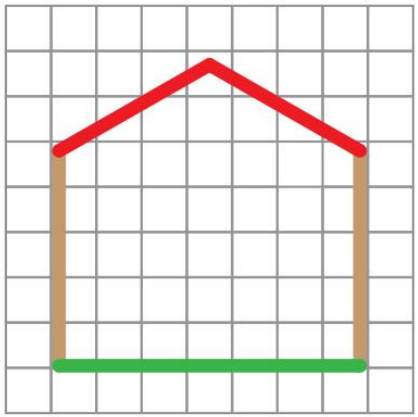


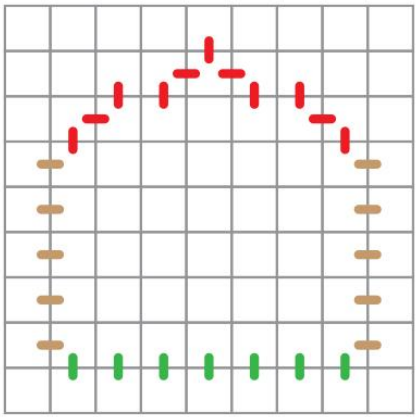
Future work

3. Edge based attribute preservation



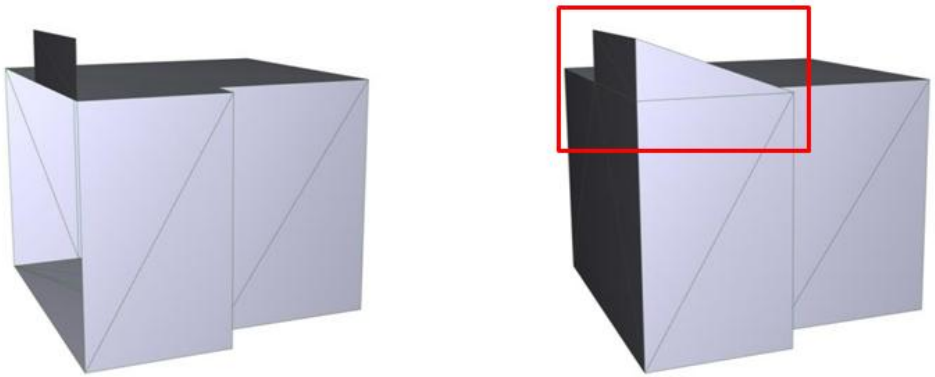
Approach 2: Dual Contouring



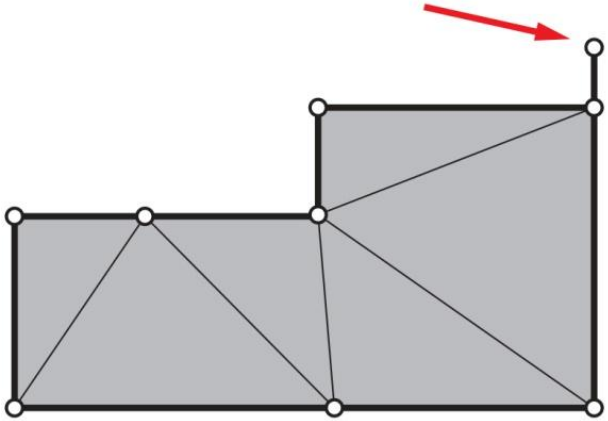


Future work

Shrink-wrapping hybrid method



Shrink-wrapping (J. Zhao et al., 2013)



Use vertice-volume distance?

Questions