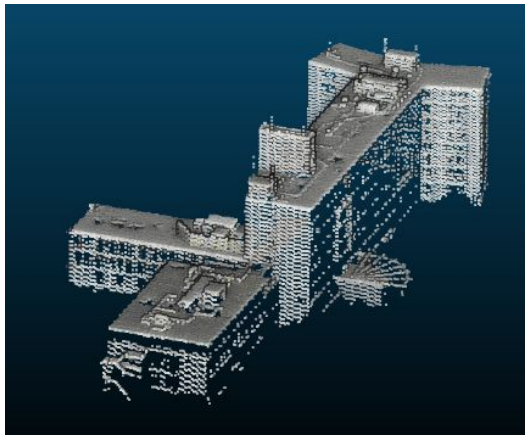
Creating a Simplified Large Scale 3D Model in PostGIS for Noise-calculations,

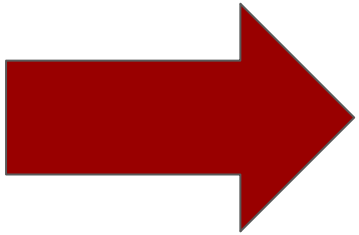
Based on Lidar and Cadastral Data

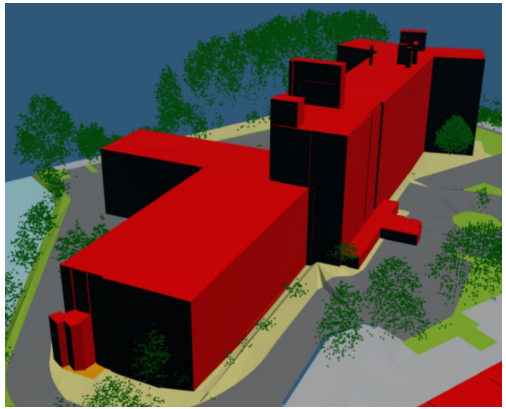
tom.van.tilburg@geodan.nl

Geodan

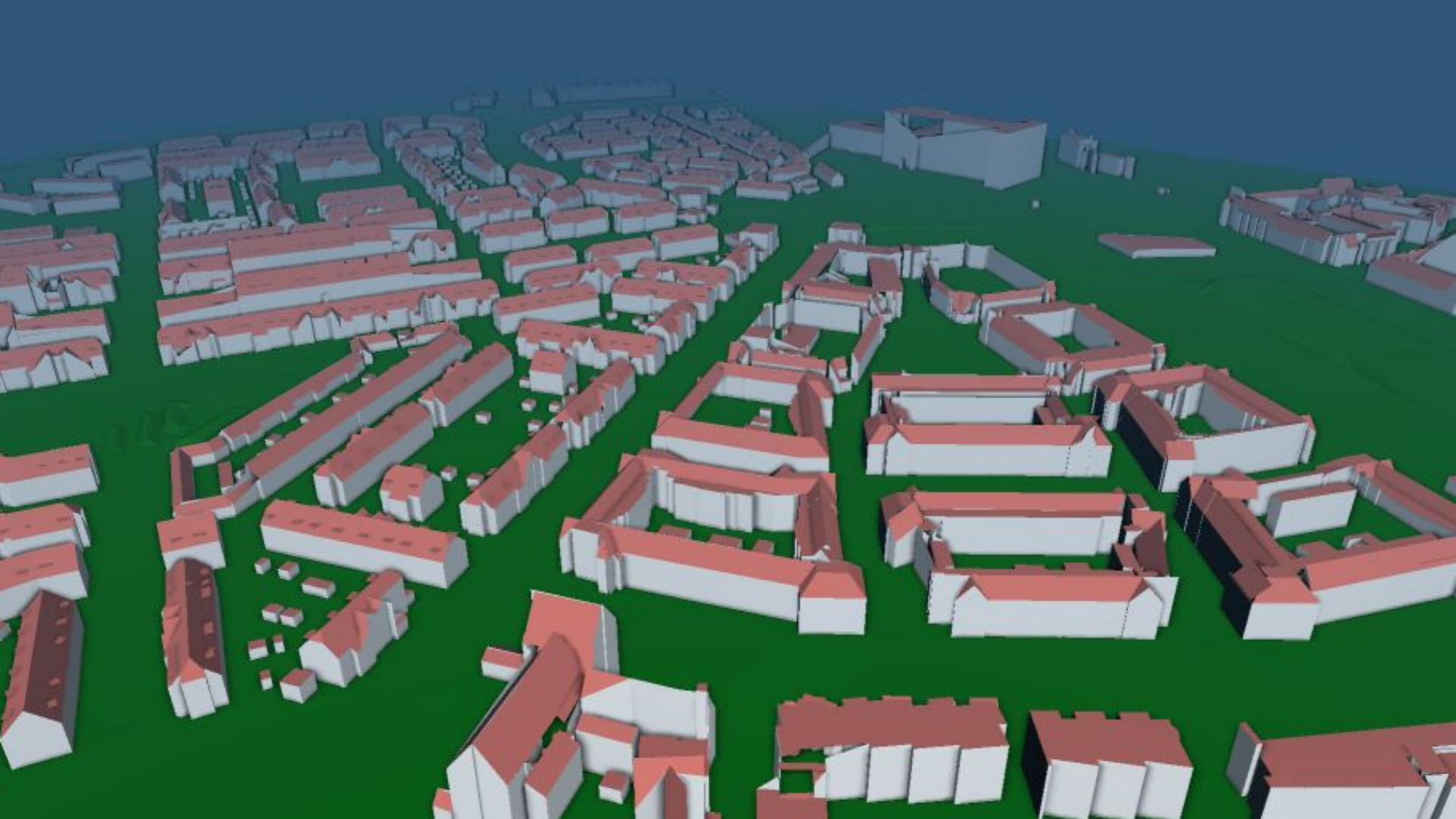
TL;DR



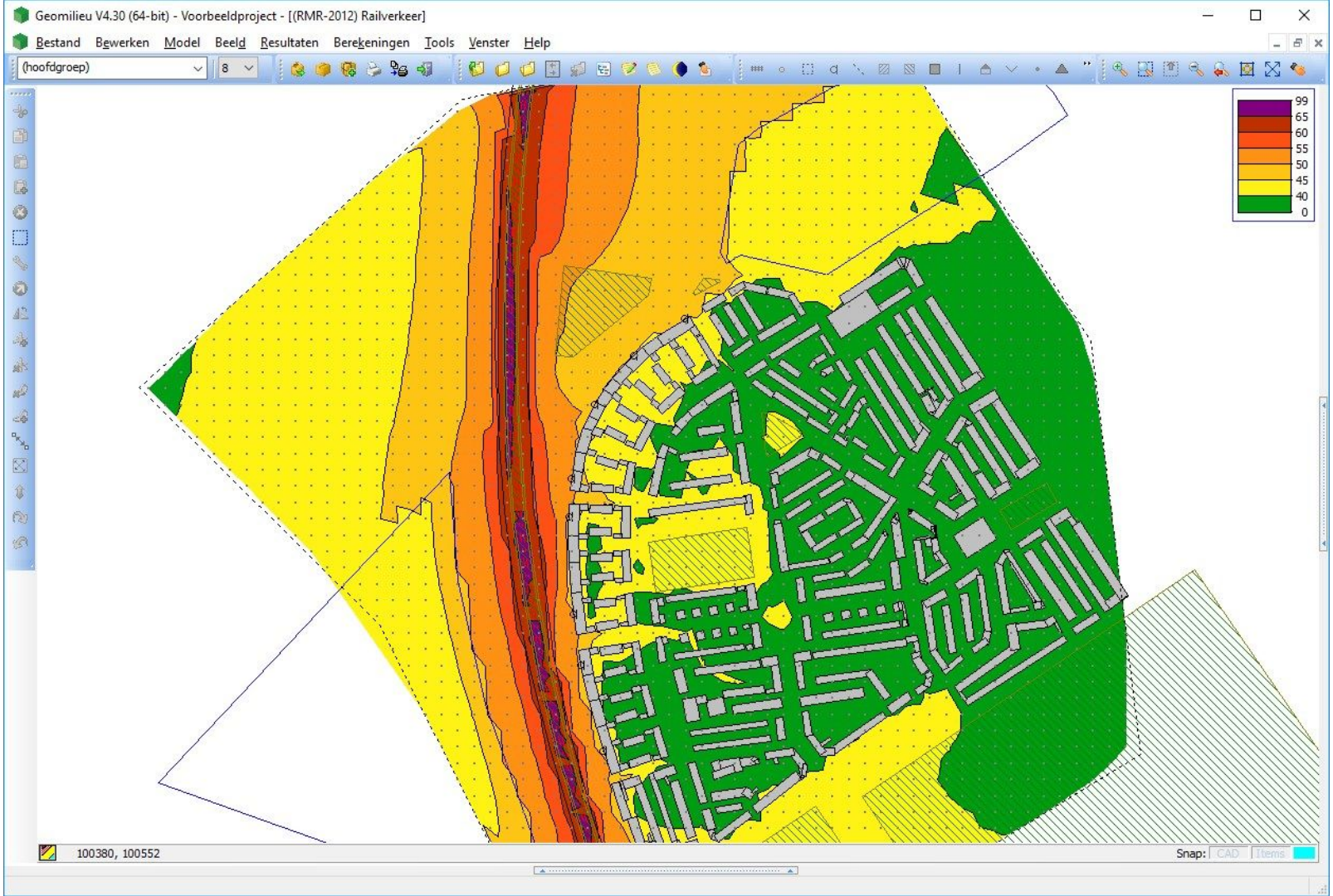




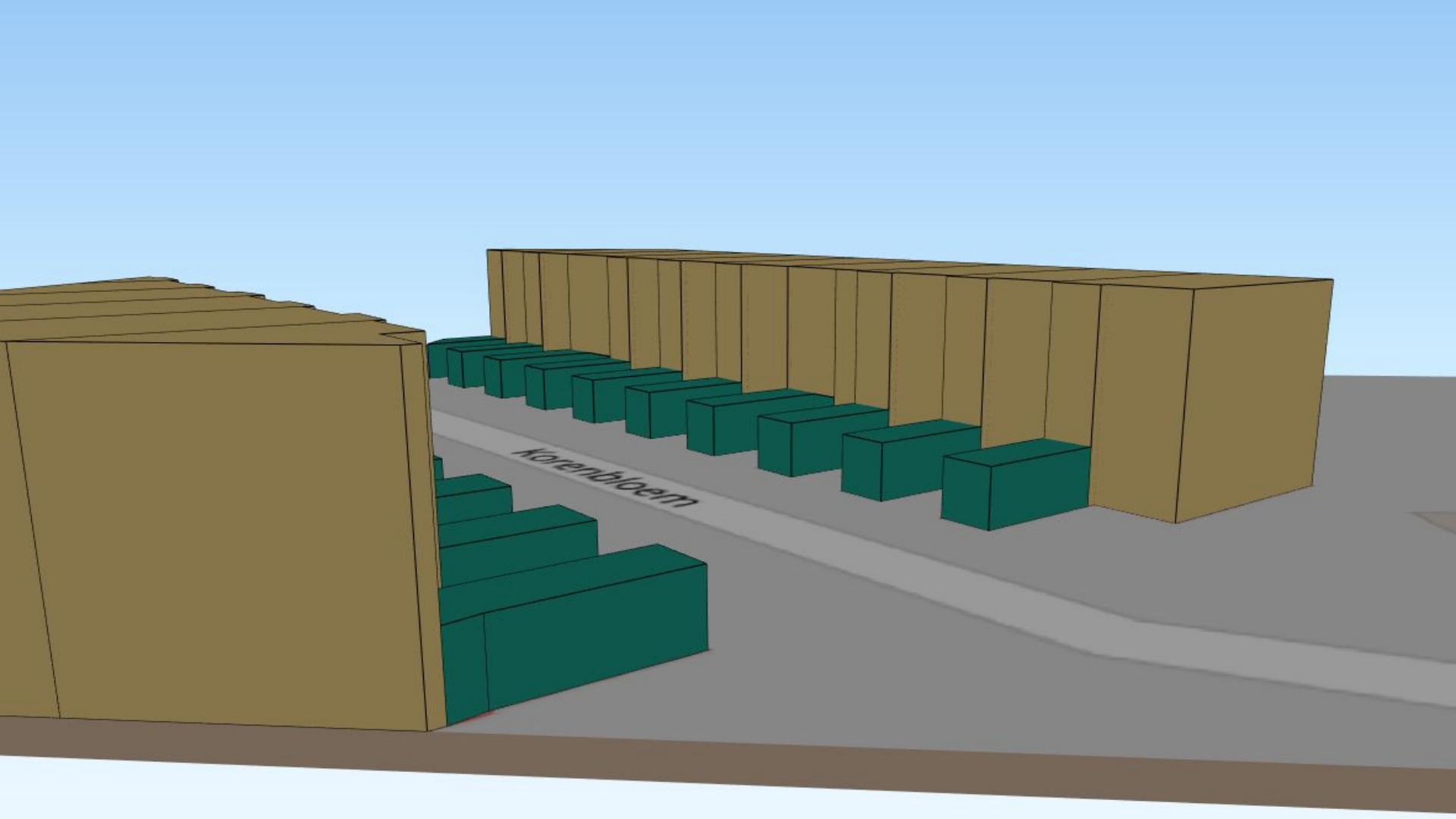

S
N
E



WHY?


Geomilieu V4.30 (64-bit) - Voorbeeldproject - [(RMR-2012) Railverkeer]
Bestand Bewerken Model Beeld Resultaten Berekeningen Tools Venster Help
(hoofdgroep)
8
99
65
60
55
50
45
40
0
100380, 100552
Snap:
CAD
Items

X 10 million


Korenbloem

PostGIS

Software

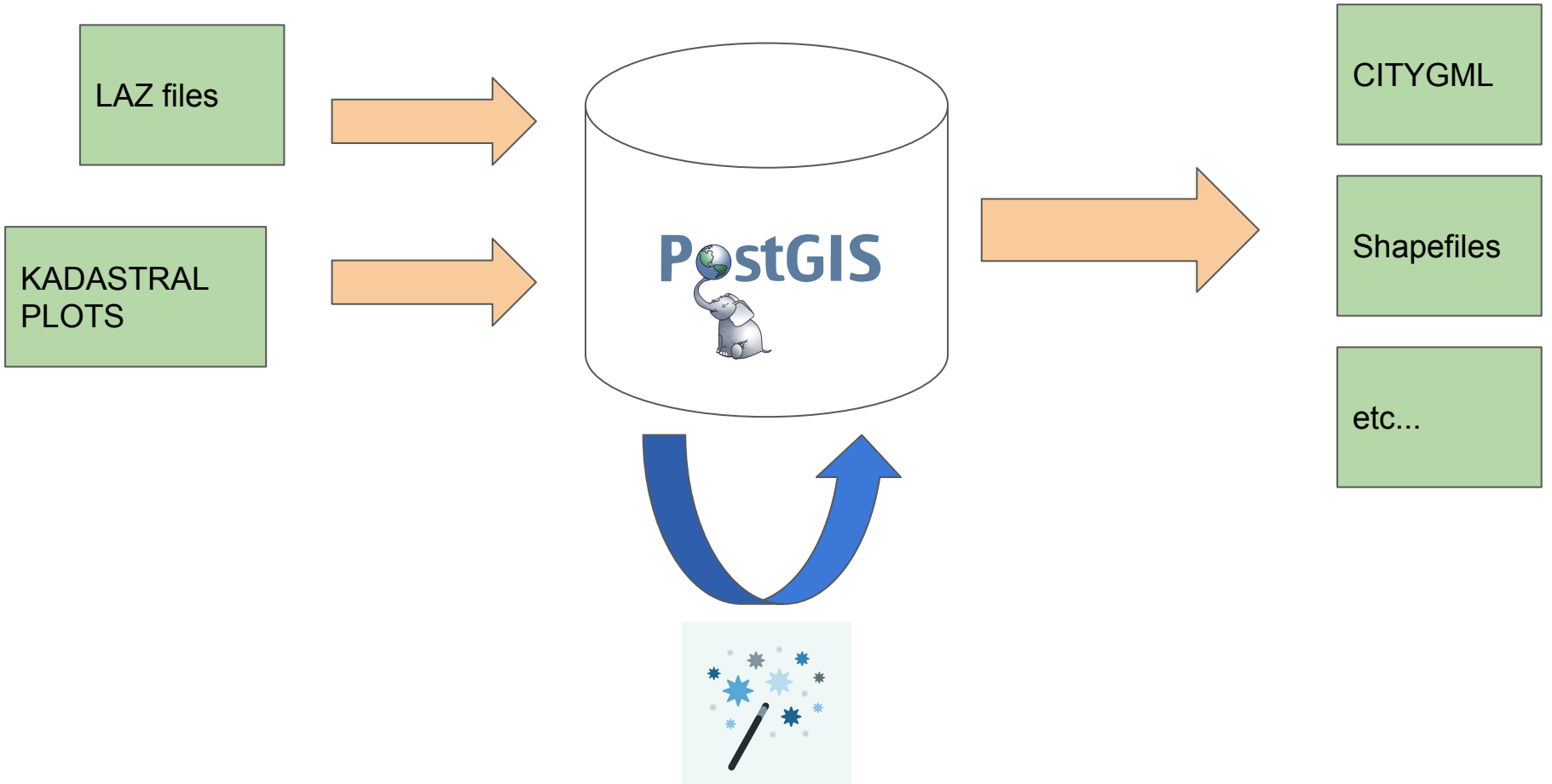
- PDAL
- PGPOINTCLOUD
- SFCGAL
- POSTGIS
- GDAL
- GRASS

Postgres extensions:

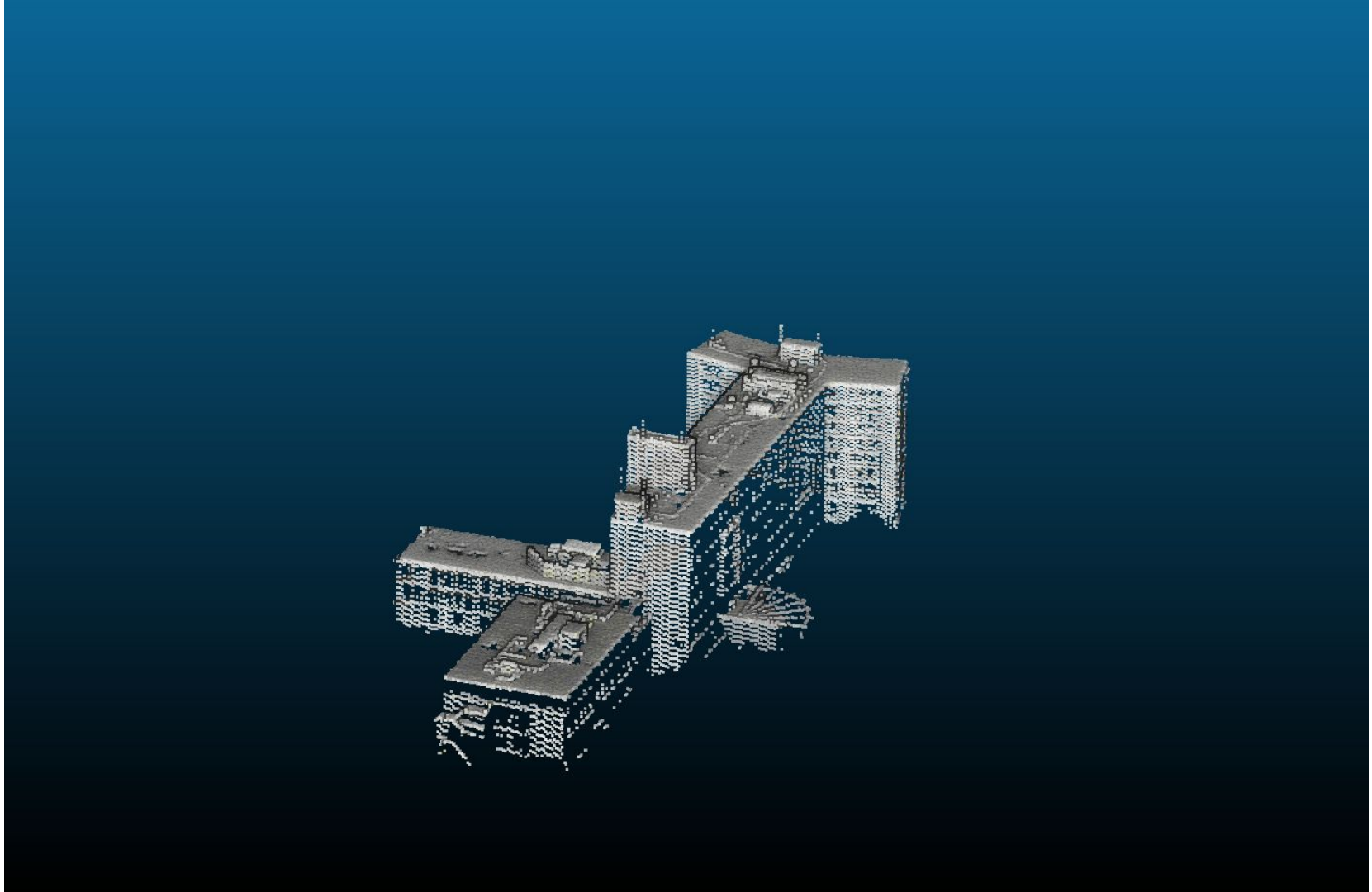
- PLV8
- PLpython

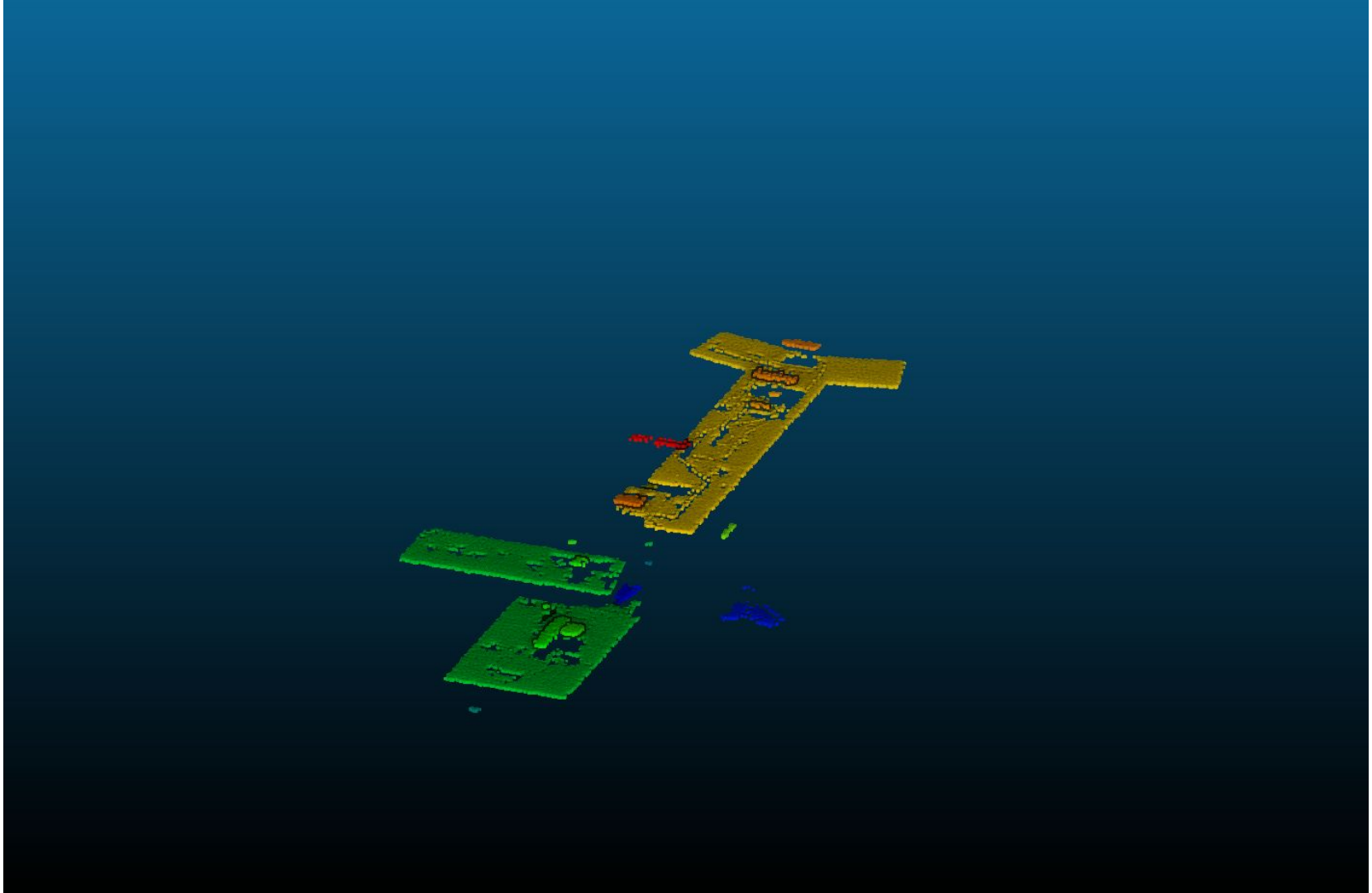
in the python extension:

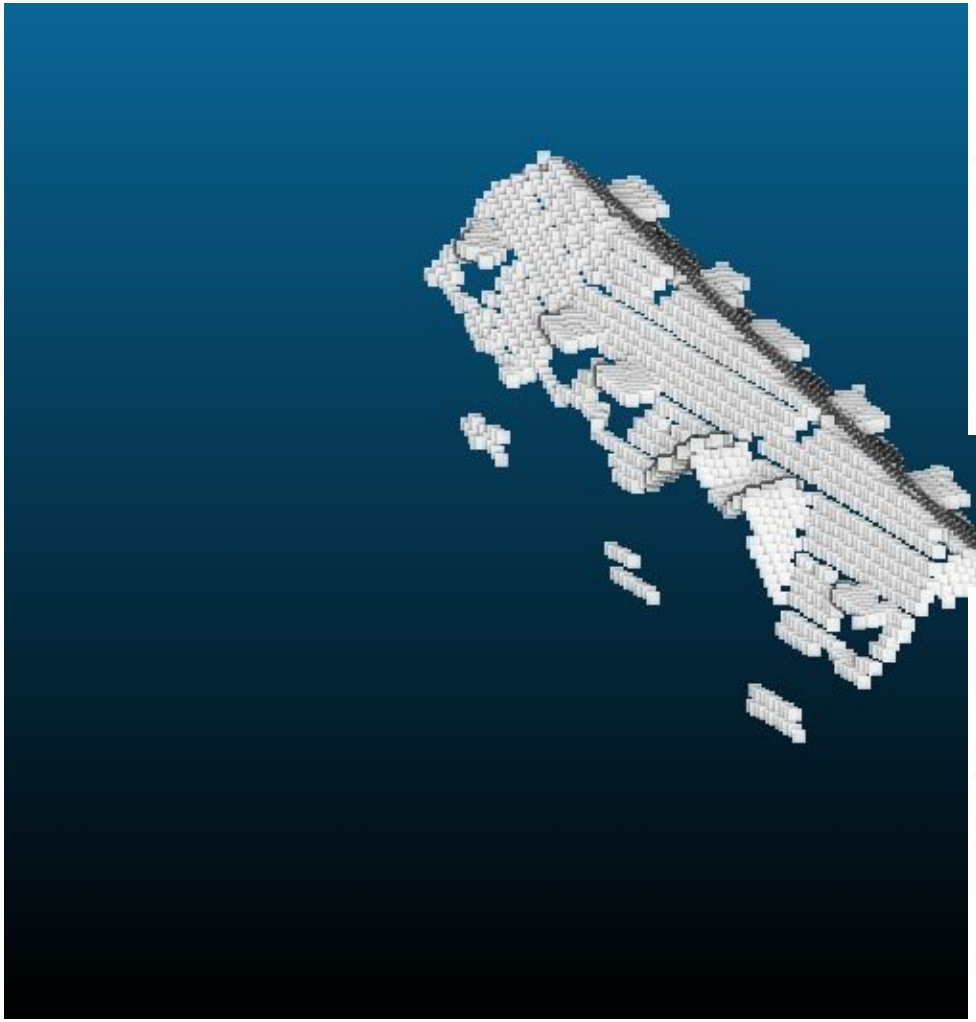
- numpy
- sklearn
- do it yourself algorithms

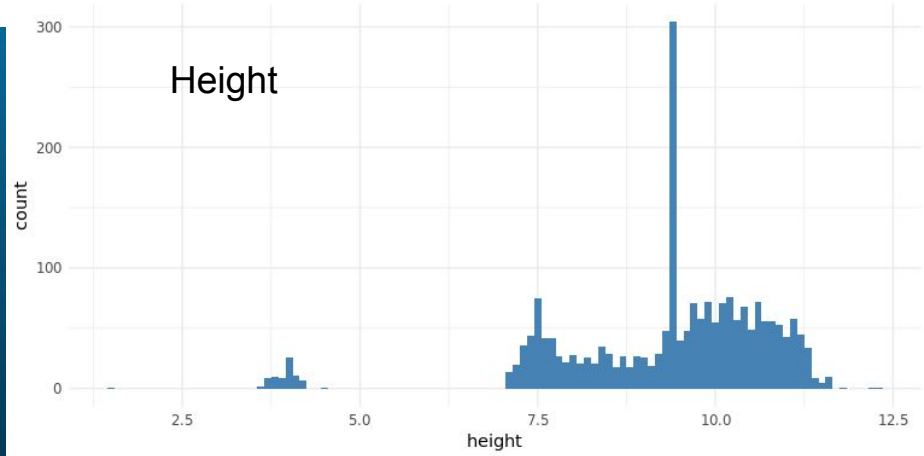

LAZ files
KADASTRAL PLOTS
PostGIS
CITYGML
Shapefiles
etc...

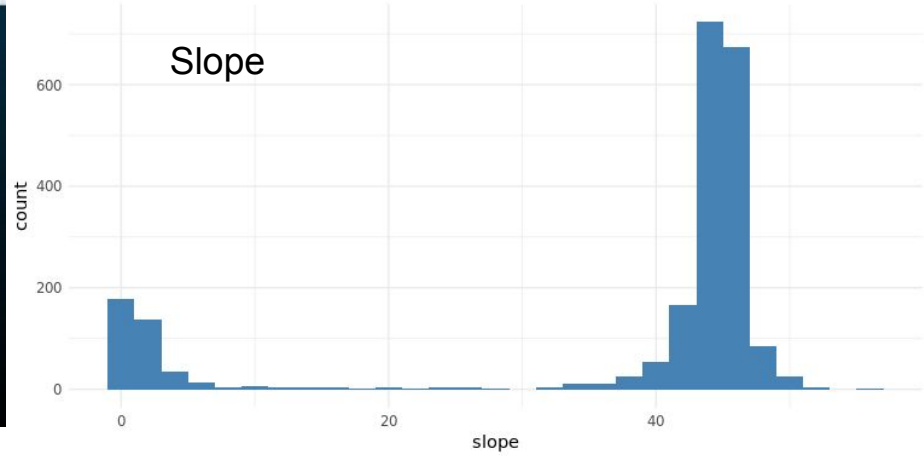
```
CREATE OR REPLACE FUNCTION noisemodel.dbscan3d(points text, eps double precision, minpoints integer)
RETURNS SETOF float[] AS
$$

from sklearn.cluster import DBSCAN
import numpy as np

def dbscan3d(points, eps=0.5, min_samples=5, metric='manhattan',
            algorithm='auto', leaf_size=30, p=None, n_jobs=1):
   geom = points[:,:3]
   db = DBSCAN(eps, min_samples, metric, algorithm,
               leaf_size, p, n_jobs)
   db.fit(geom)
   labels = np.zeros((len(points), 5))
   labels[:, :4] = points
   labels[:, 4] = db.labels_
   return labels

arrpoints = np.array(eval(points))
return dbscan3d(arrpoints, eps, minpoints)
$$
LANGUAGE plpythonu;
```

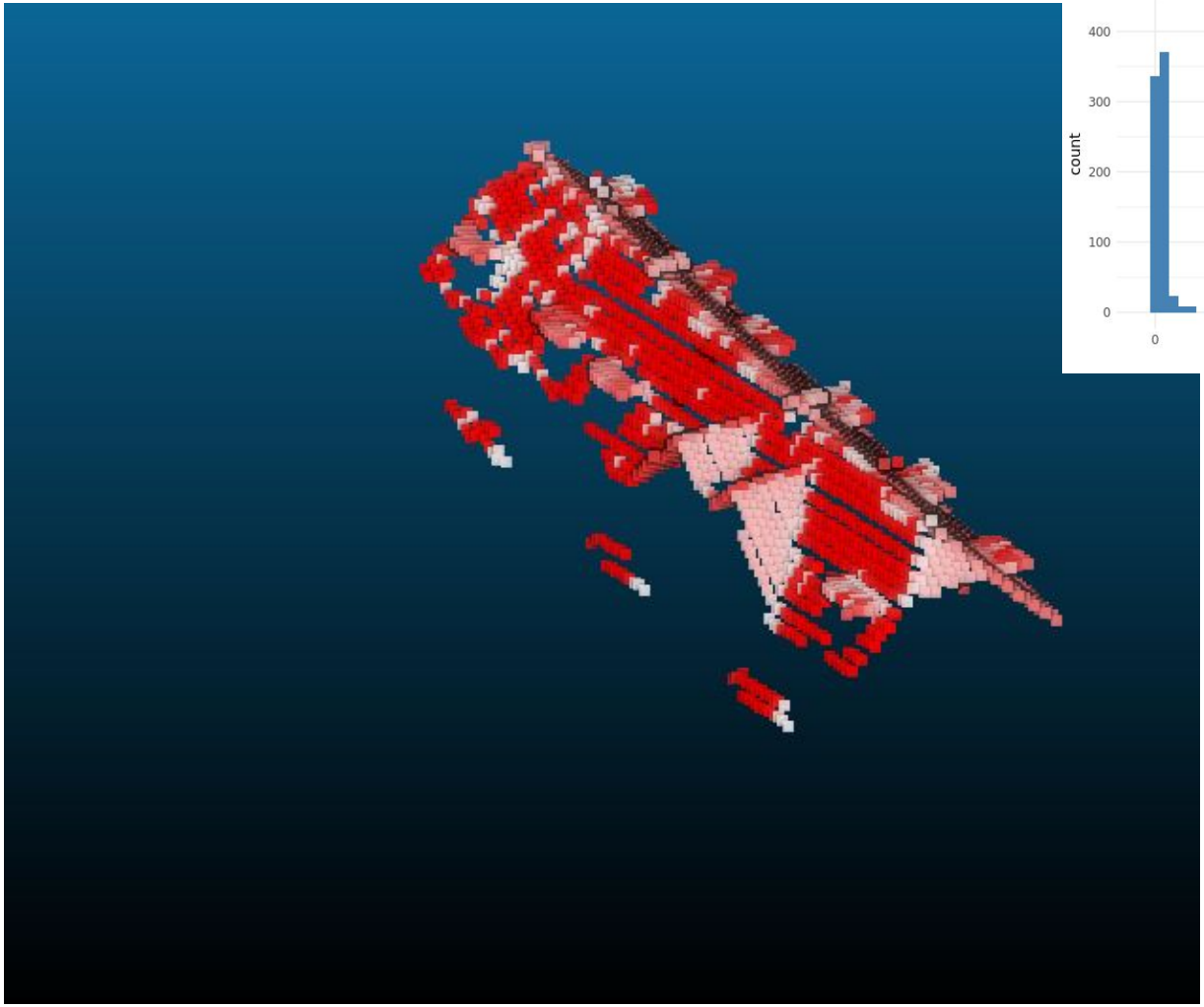


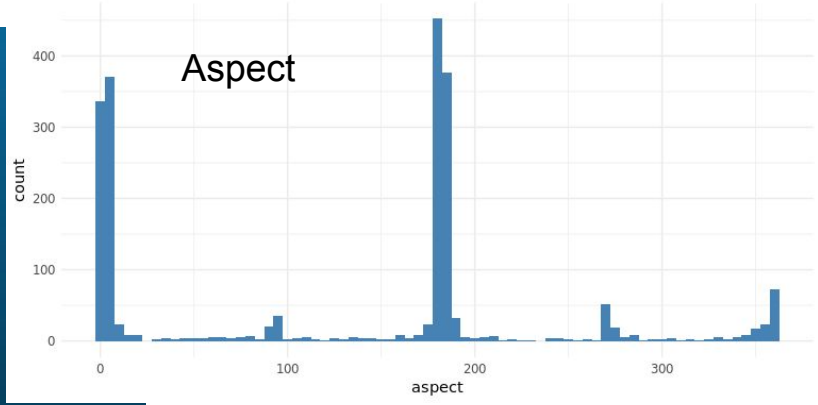


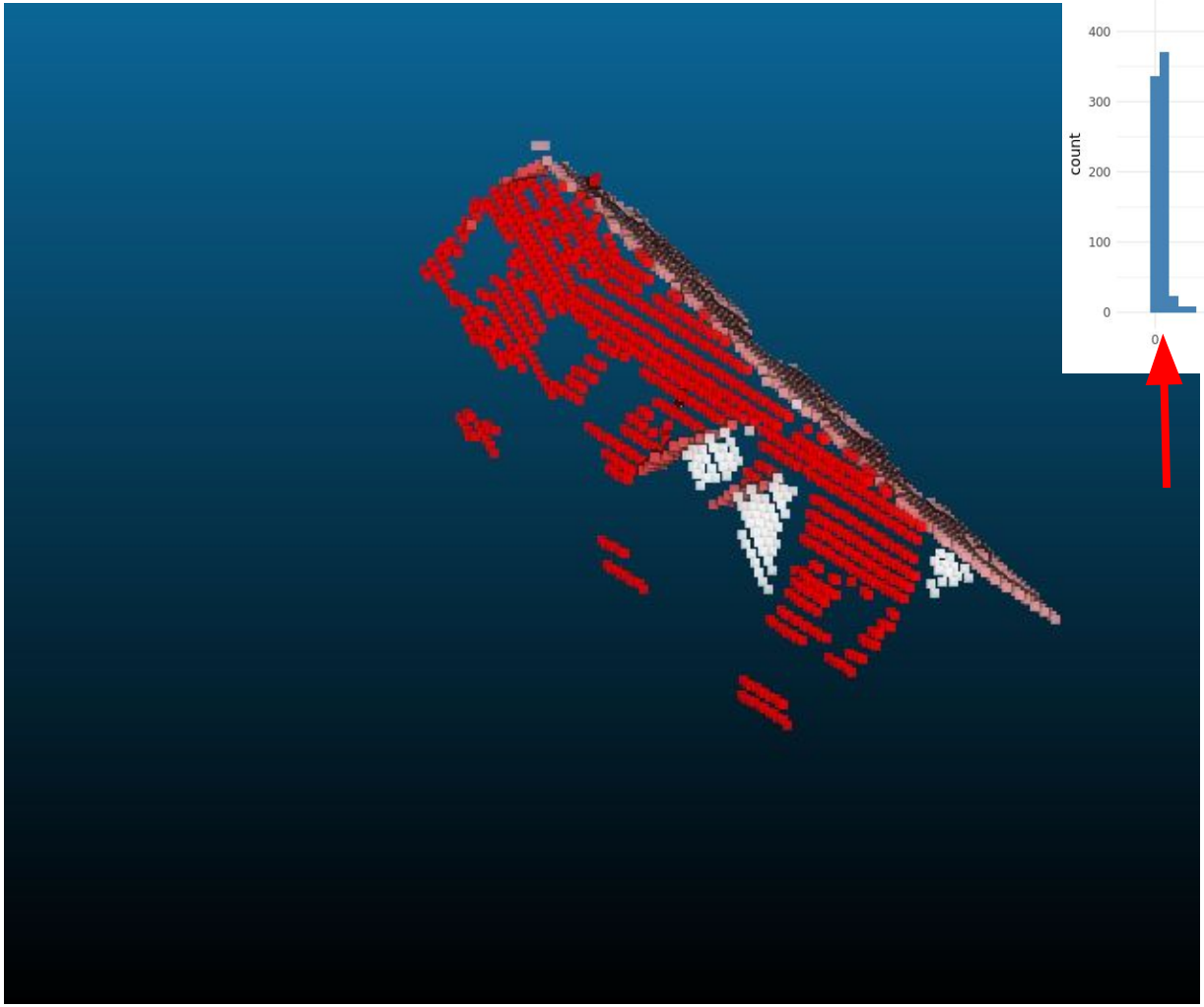


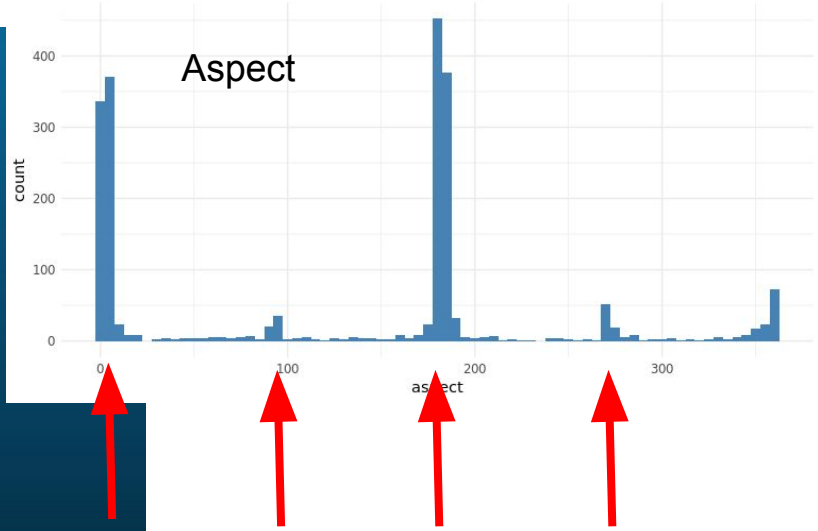

Height
300
200
100
0
count
2.5
5.0
7.5
10.0
12.5
height

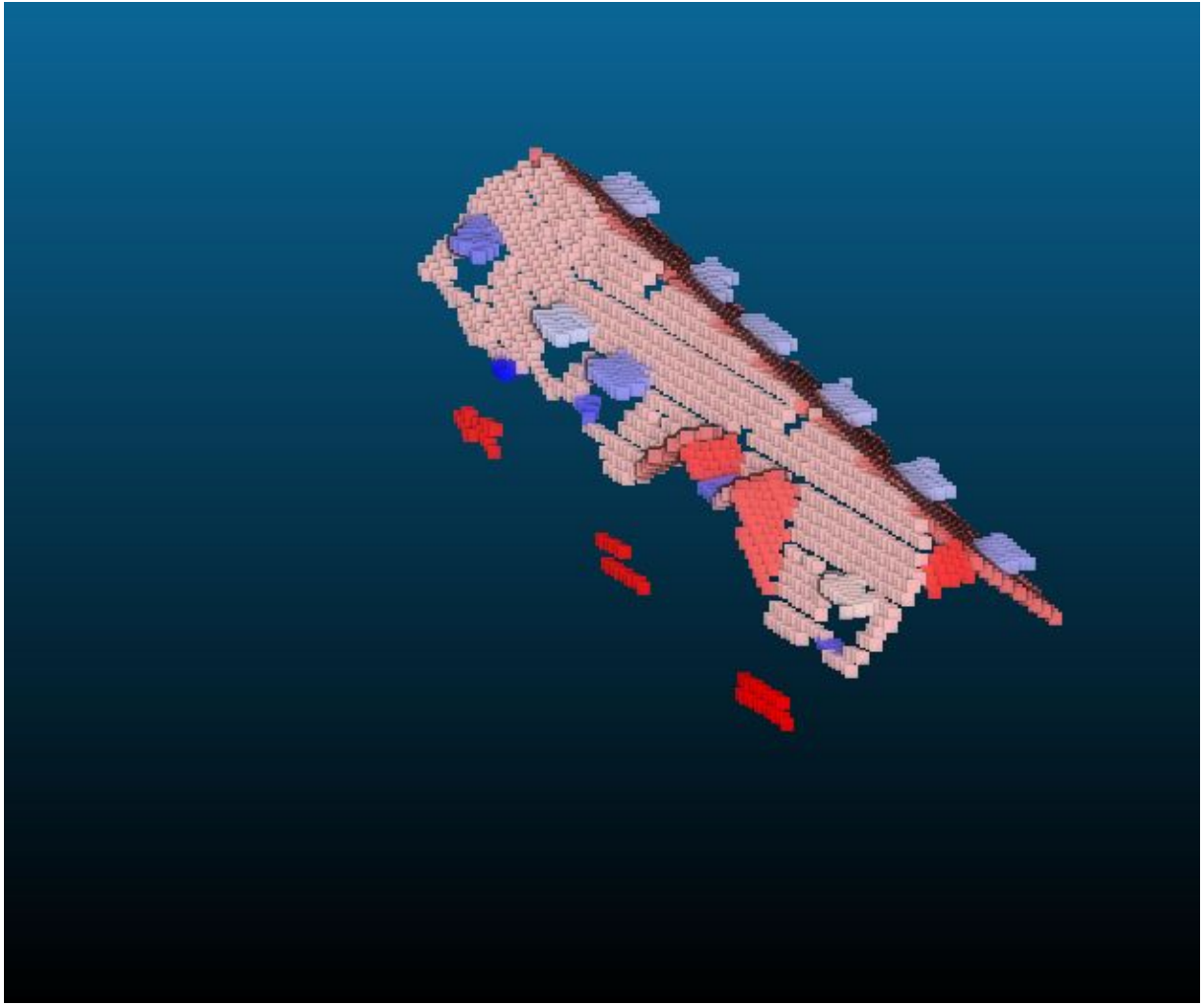

Slope
600
400
200
0
count
0
20
40
slope

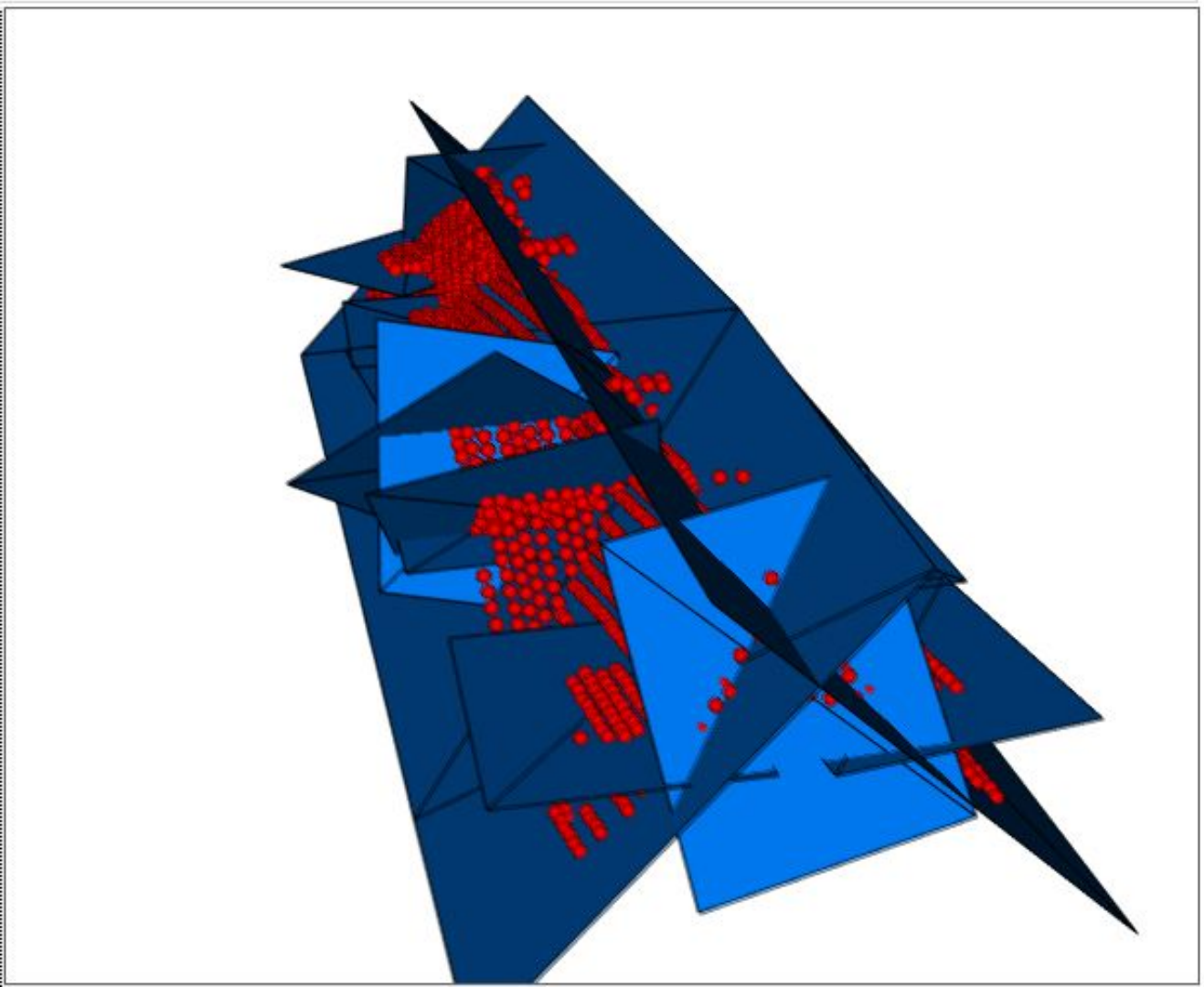


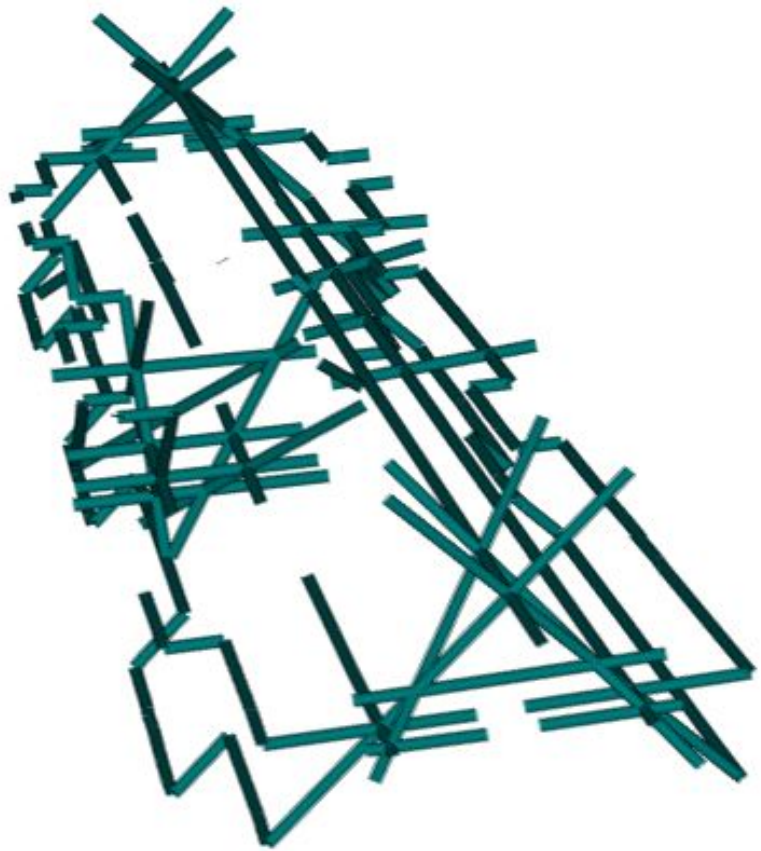

Aspect
count
400
300
200
100
0
0
100
200
300
aspect

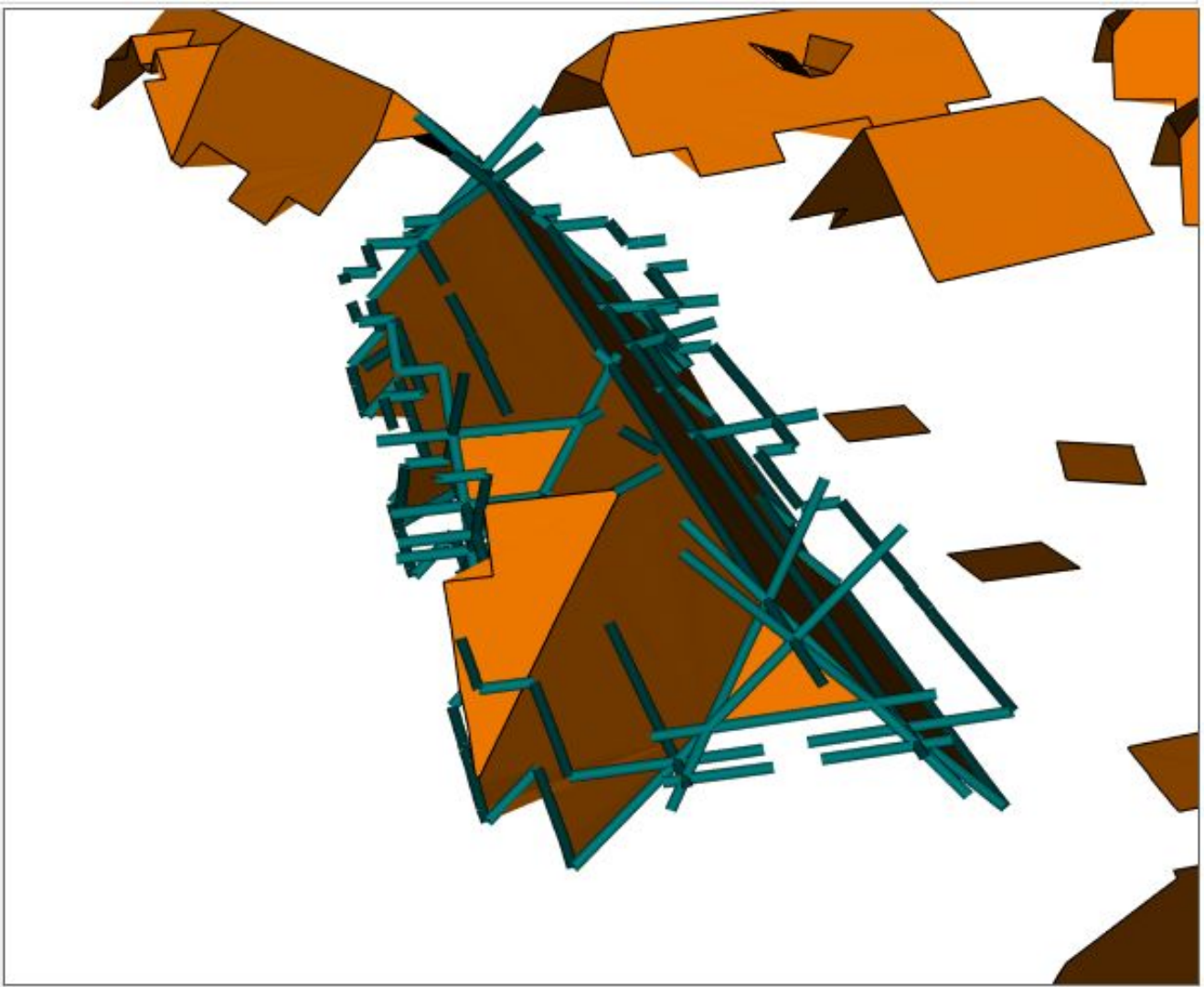


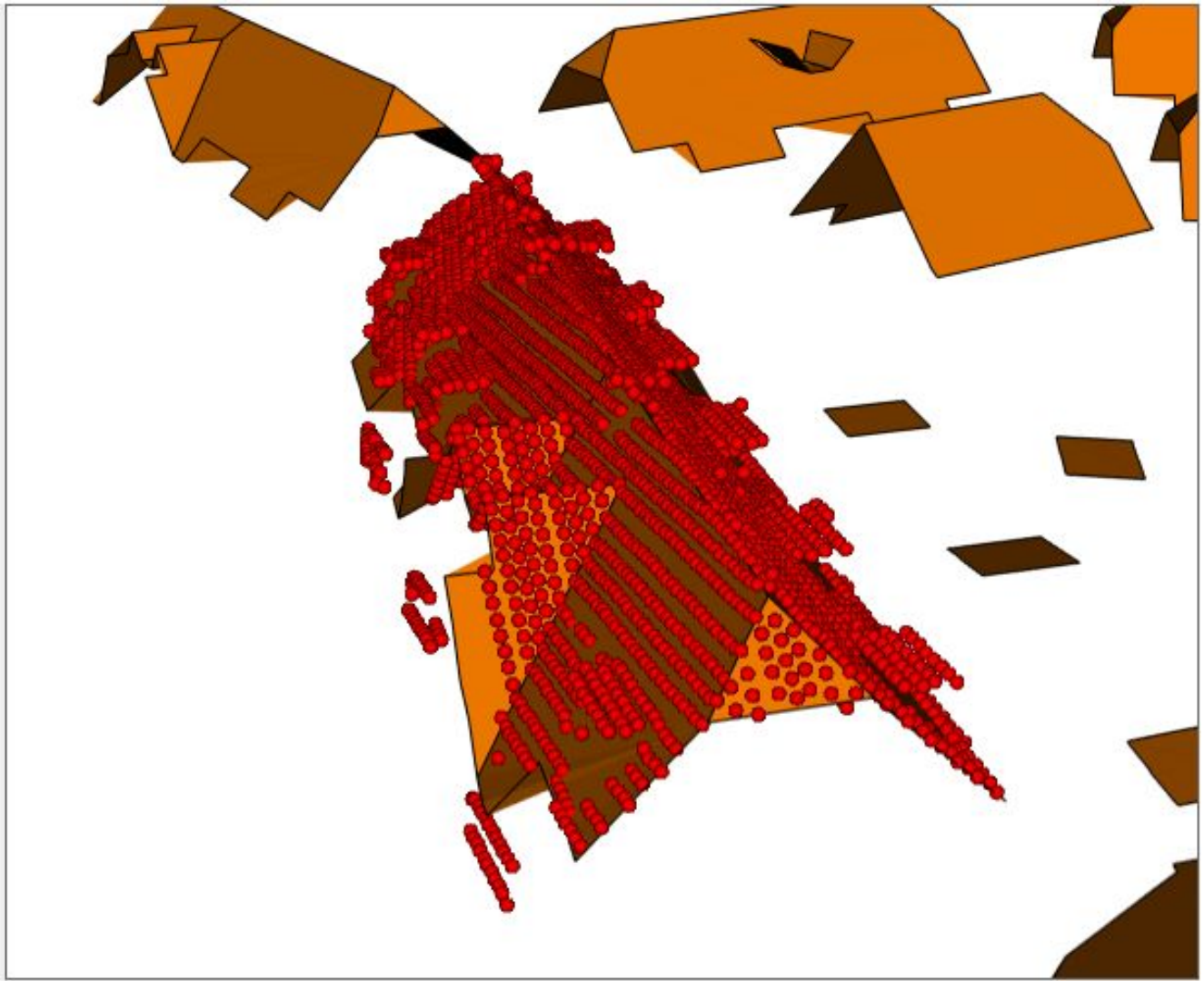

Aspect
count
400
300
200
100
0
0
100
200
300
aspect

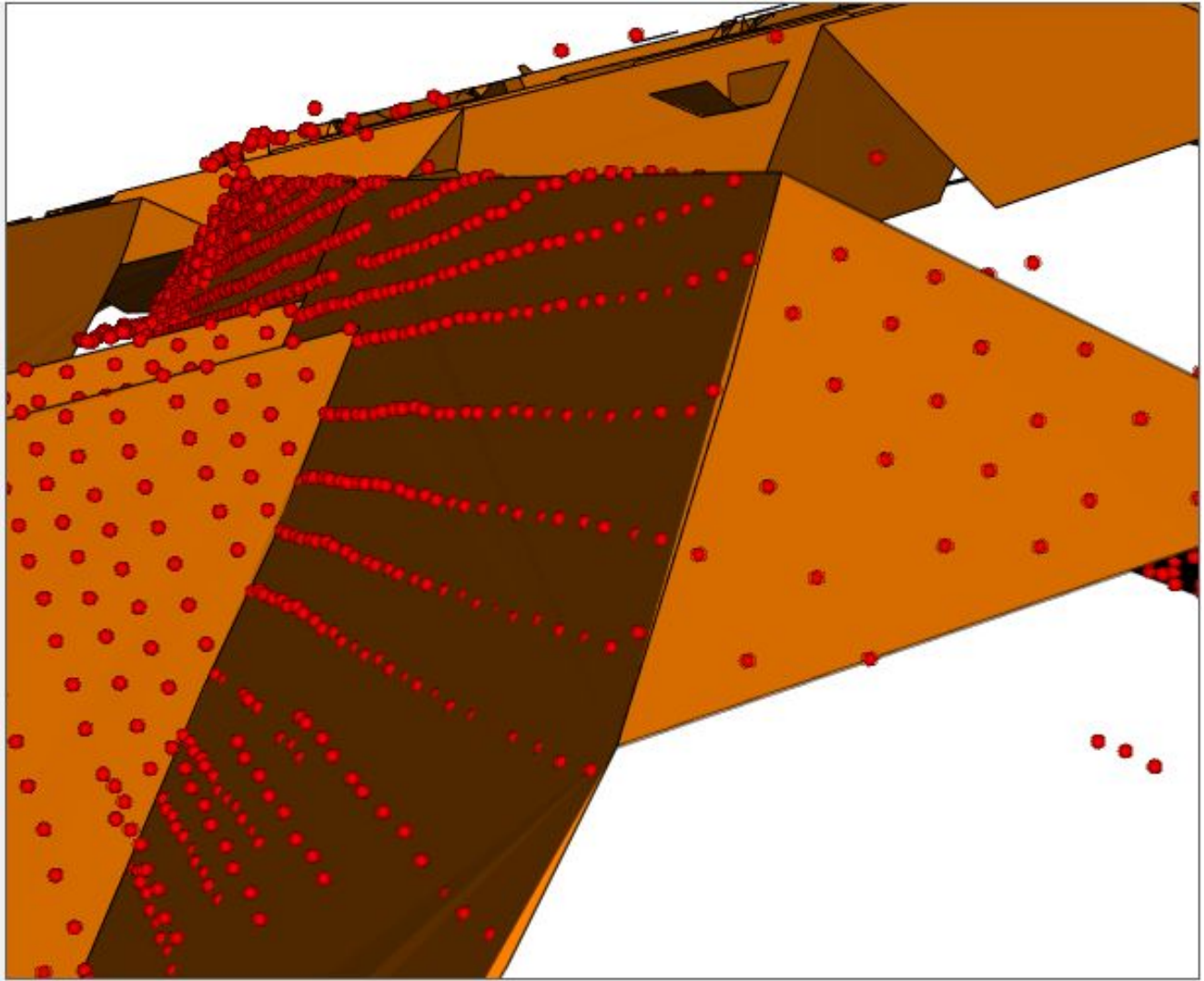












Where are we?

LOD1.3 for NetherLands

Run time:

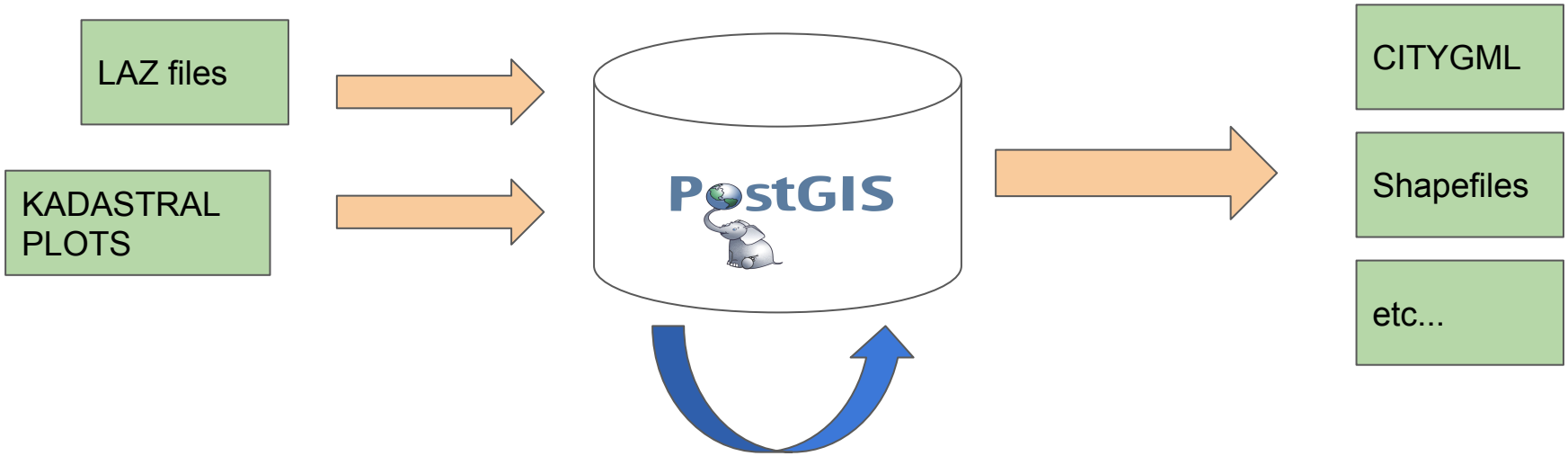
5 minutes / km2

Amsterdan = 20 hours

Where do we go?

LOD2.2

- Gable roofs
- Dormers
- Porches



Conclusion

A geospatial database is suitable for large scale 3D processing

Fin