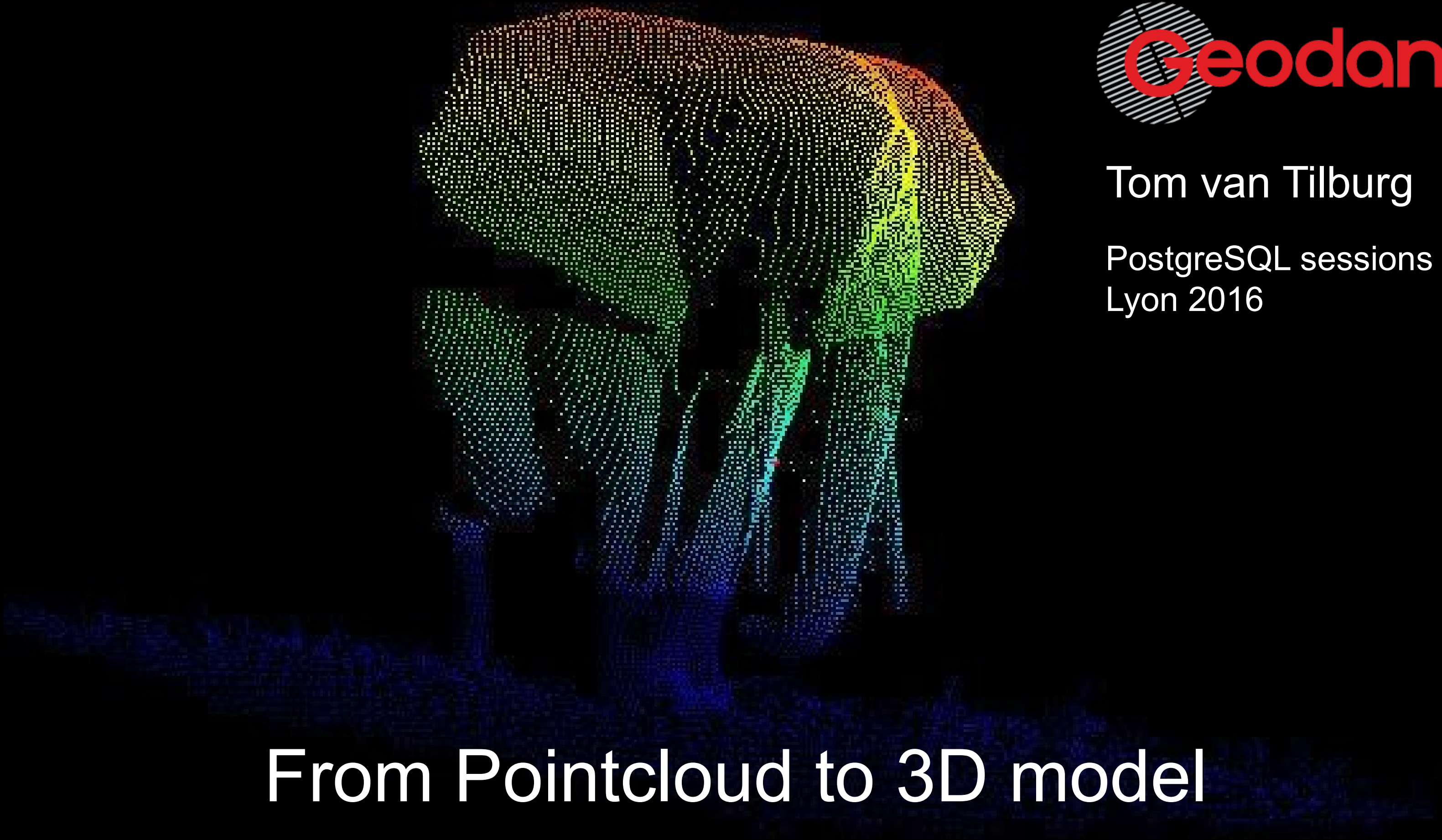


Tom van Tilburg

PostgreSQL sessions #8
Lyon 2016

From Pointcloud to 3D model with the help of postgres

Contents

What are pointclouds

The Dutch situation

Our experiences

3 use cases and demo

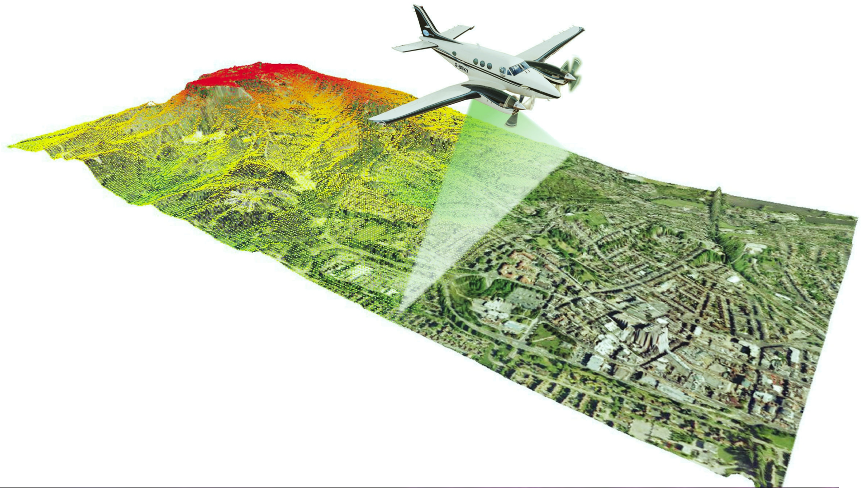
Discussion

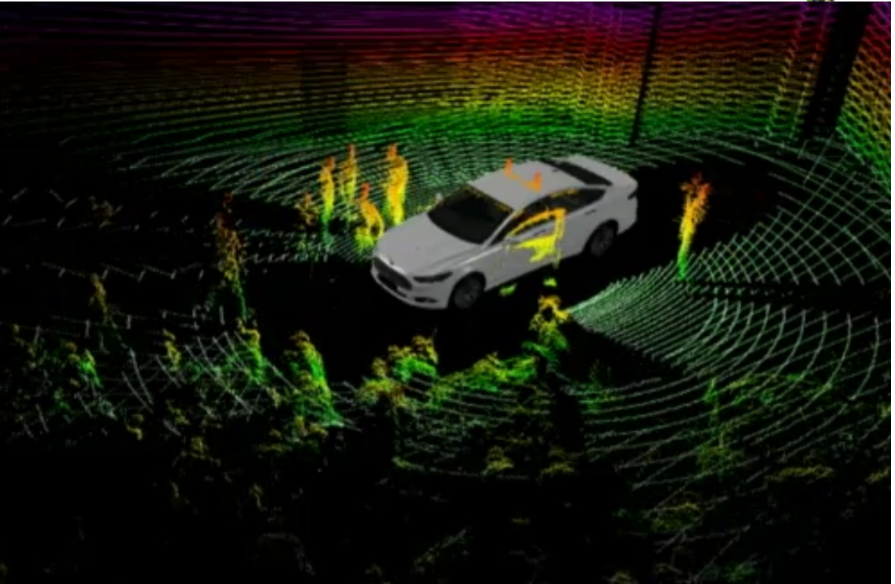
What are pointclouds

Often sensordata

Not raster, Not vector

Can be a small object, can be a country





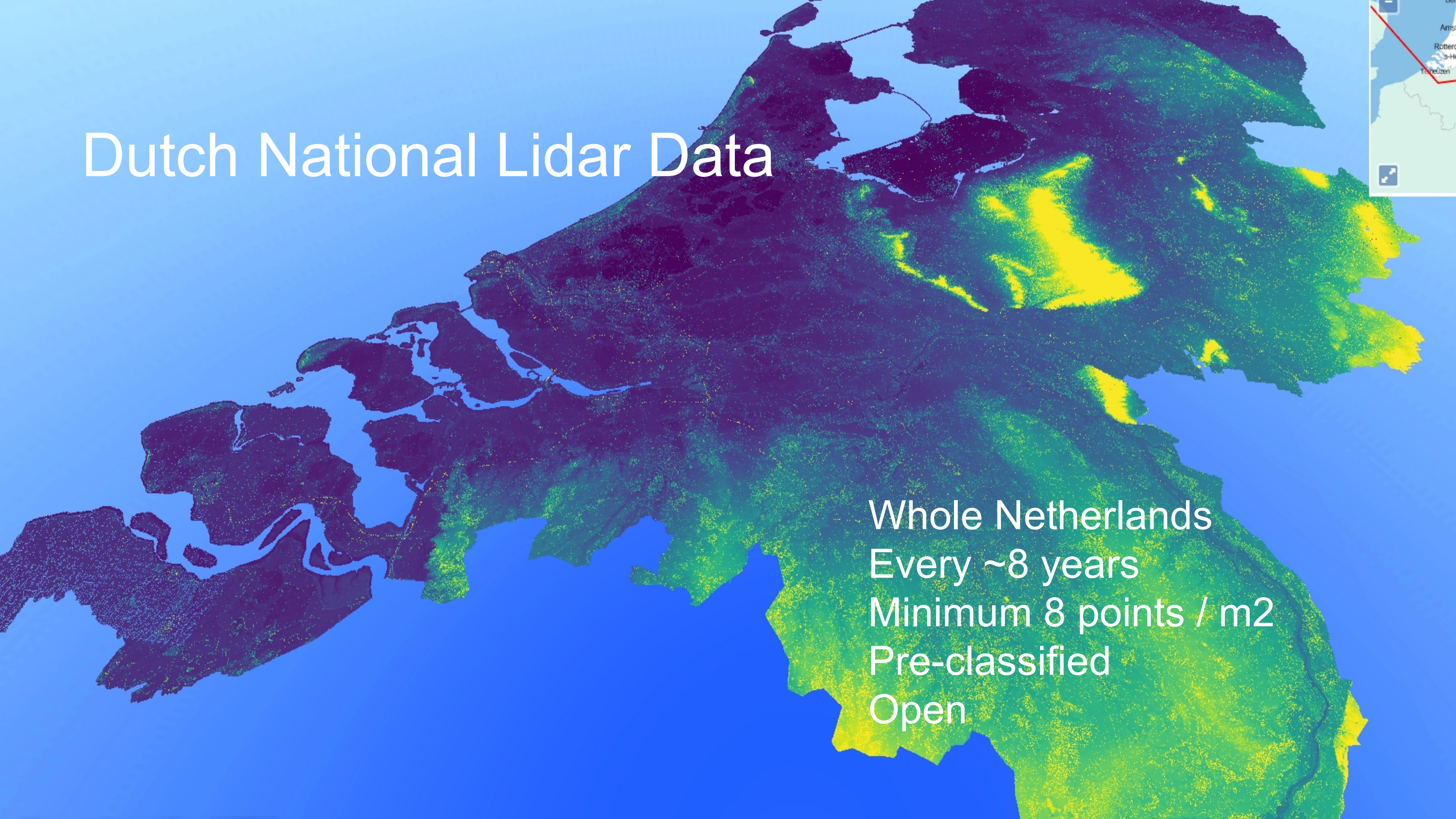
Laser scanner

Commonly contains:

- X,Y,Z
- Time
- Intensity
- Number of returns
- Return number

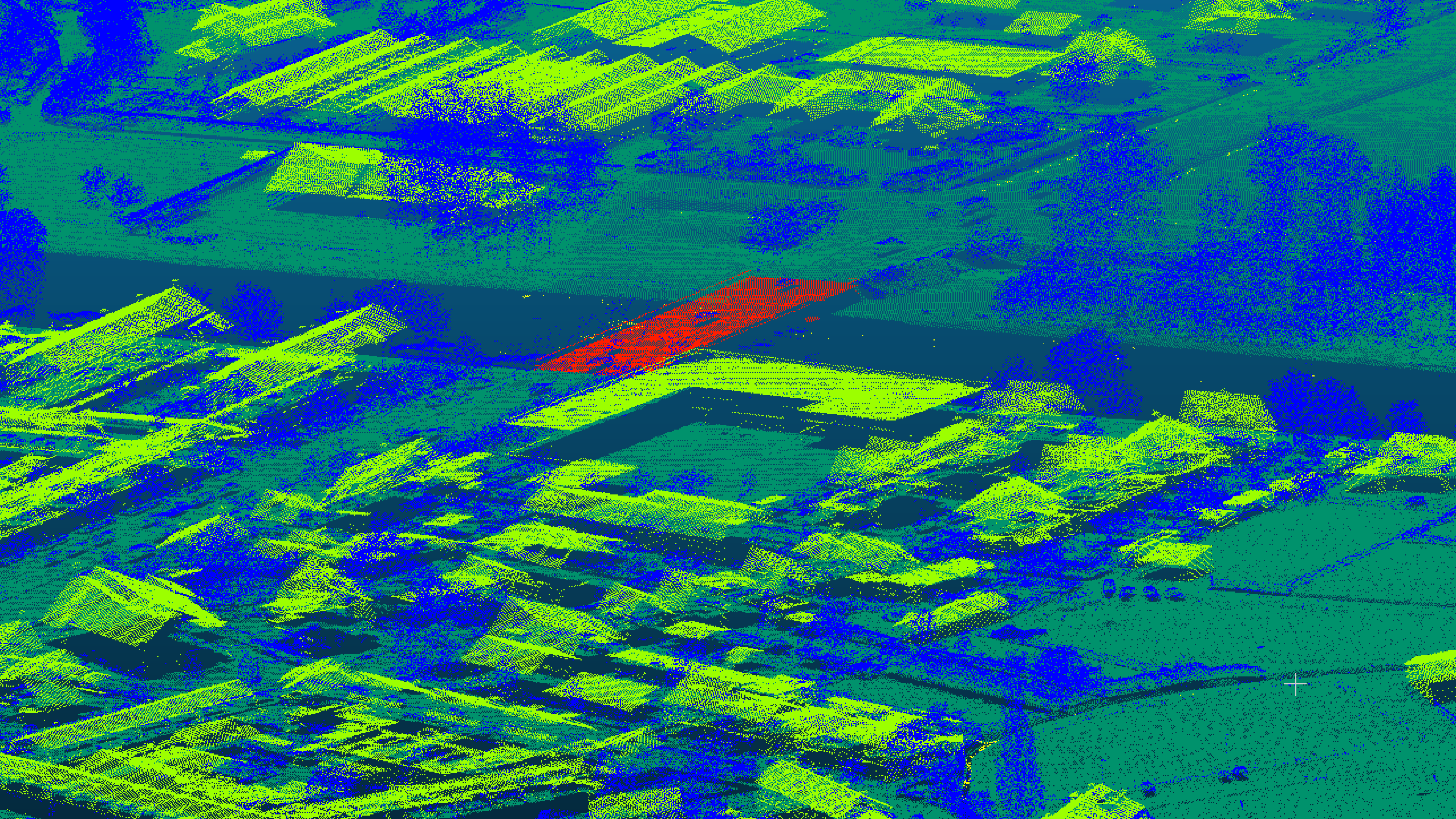
QT viewer

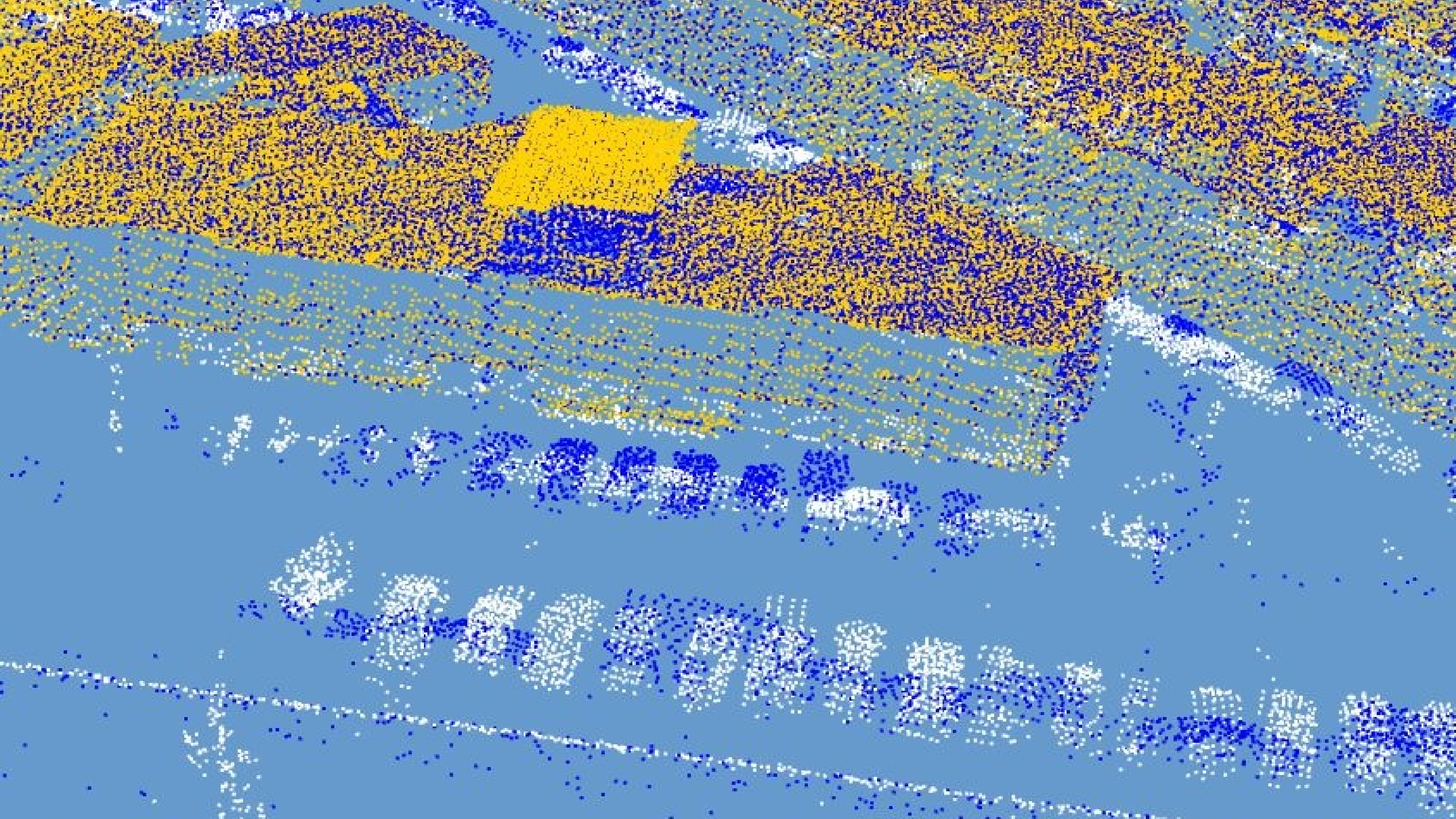
Dutch National Lidar Data



Whole Netherlands
Every ~8 years
Minimum 8 points / m2
Pre-classified
Open

W
S
N
E





Put *all* pointcloud data in postgres

PG Pointcloud

A PostgreSQL extension for storing point cloud (LIDAR) data

```
{
   “pcid" : 1,
   “pts" : [
       [0.02, 0.03, 0.05,
6],
       [0.02, 0.03, 0.05,
8]
   ]
}
```

https://github.com/pgpointcloud/pointcloud

Some difficulties

Laz files are up to 3 gb (!)

Total of 413 gb of laz files

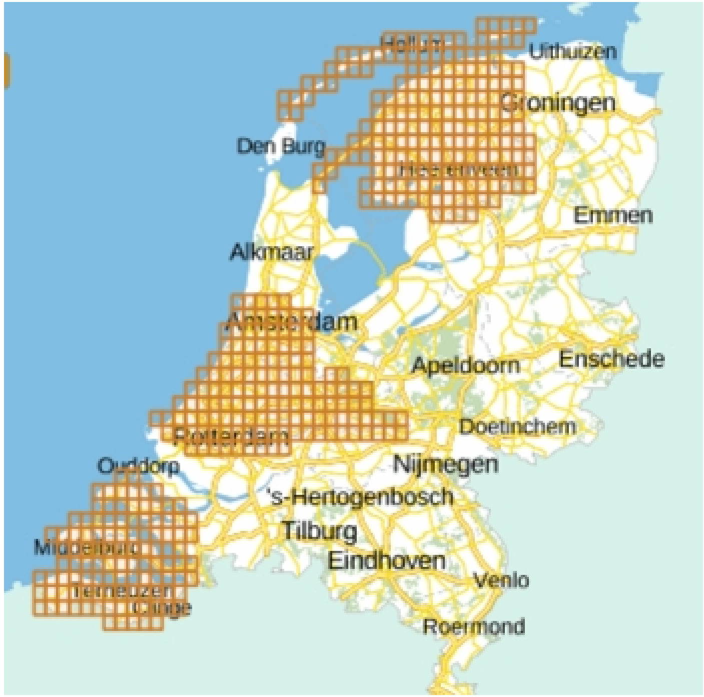
Laz -> pgpointcloud 4x bigger (dimensional compr)

Total of:

6 TB disk space

Weeks of loading time...

and we have another set coming….



How we do it

6TB of SSD

PDAL pipeline with chipper (100 Gb in memory)

600 pts per patch, gist index on bbox

Every laz file in its own table, combine in one view

Pitfalls

Parallel loading

-> data gets dispersed on disk

(Thanks to *Giuseppe Broccolo)*

Putting all 6TB (~6 billion records) in 1 table

-> headache!

Some nice surprises

Indexing patches is fast

Indices are effective

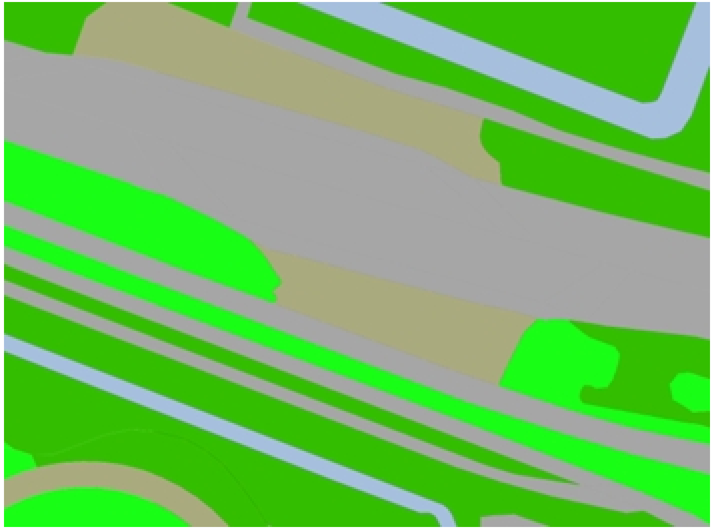
Filtering with pgpointcloud not as slow as expected

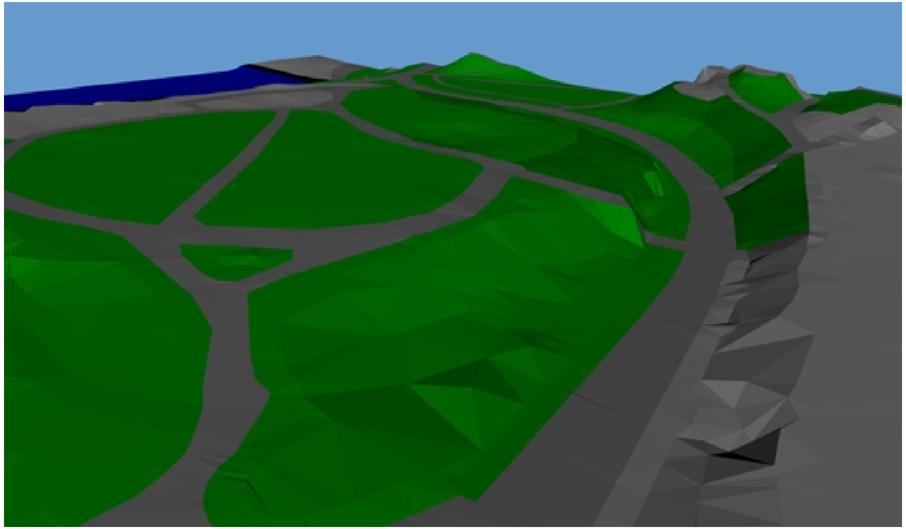
Some good SFCGAL functions to make use of

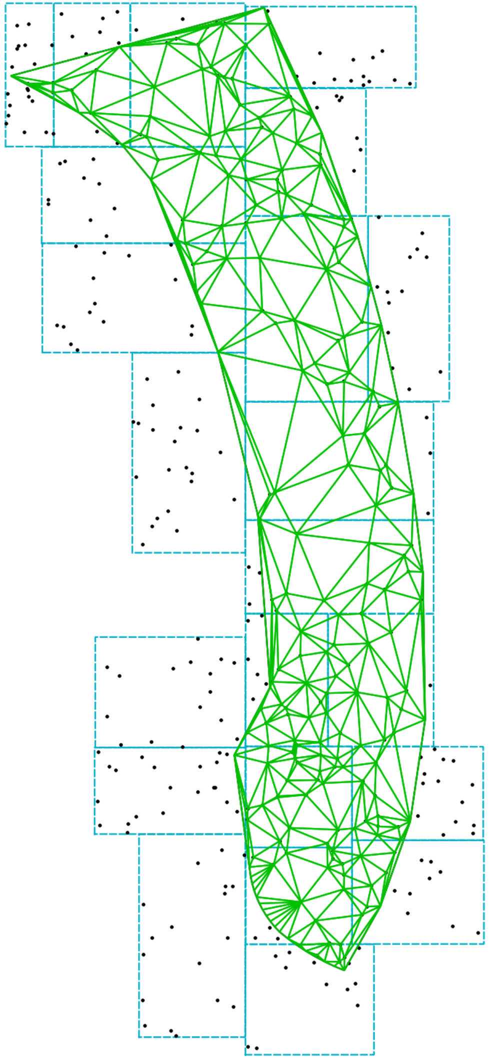
Hidden feature: ST_Triangulate2DZ

Use case 1:

Make 3D out of 2D kadastral data







More on this process in the next talk

```
)
,basepoints AS (
        SELECT id,geom FROM polygonsz
        WHERE ST_IsValid(geom)
)
,triangles AS (
        SELECT
                id,
                ST_MakePolygon(
                        ST_ExteriorRing(
                                (ST_Dump(ST_Triangulate2DZ(ST_Collect(a.geom)))).geom
                        )
                )geom
        FROM basepoints a
        GROUP BY id
)
,assign_triags AS (
        SELECT          a.*, b.type, b.class
        FROM triangles a
        INNER JOIN polygons b
        ON ST_Contains(b.geom, a.geom)
        ,bounds c
        WHERE ST_Intersects(ST_Centroid(b.geom), c.geom)
        AND a.id = b.id
)

SELECT _south::text || _west::text || p.id as id, p.type as type,
        ST_AsX3D(ST_Collect(p.geom),5) geom
FROM assign_triags p
GROUP BY p.id, p.type
```

Demo time!

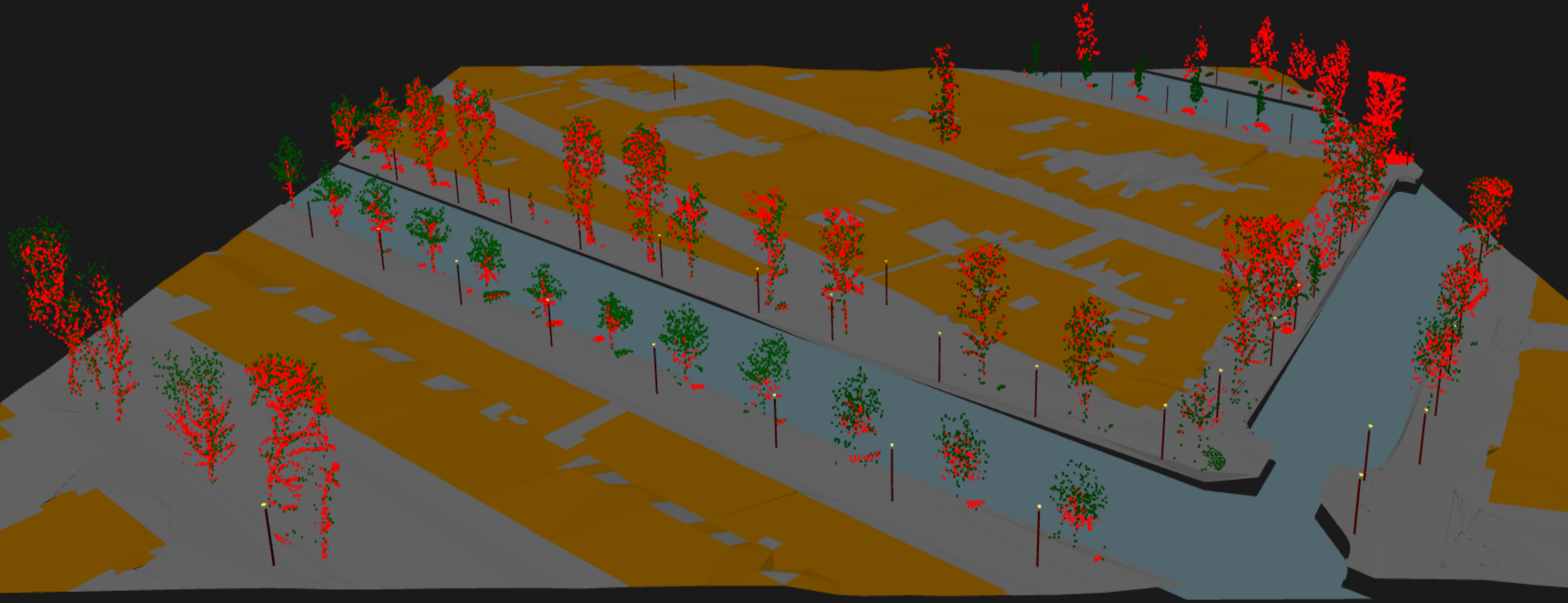
Use case 2:

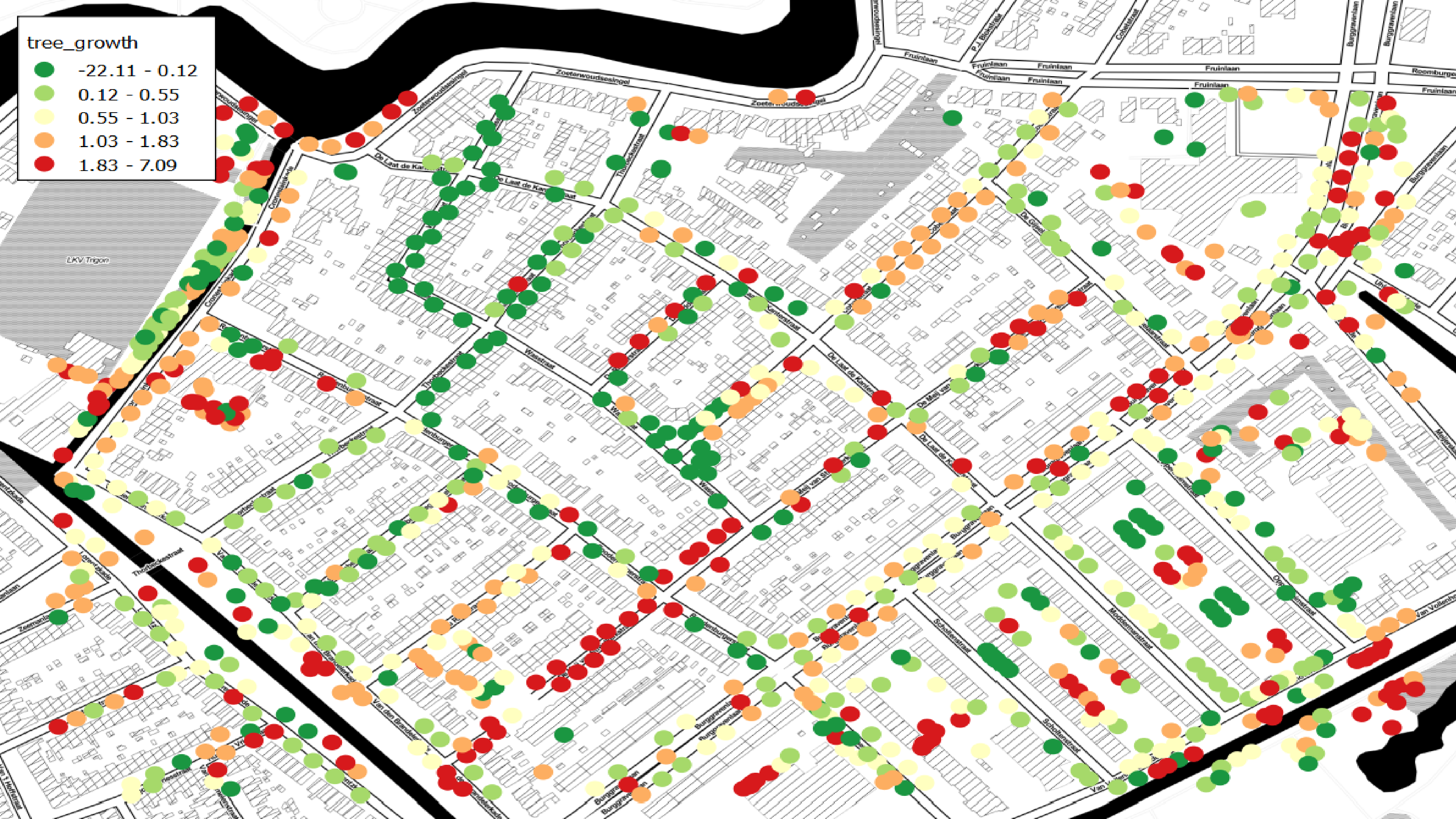
Finding tree growth

Finding tree cluster

```
SELECT
    Geometry(pt) AS geom ,
    ST_ClusterDBScan(Geometry(pt), eps := 2.5, minpoints := 370) over () AS cid
    FROM points a
    WHERE PC_Get(pt,'ReturnNumber') < PC_Get(pt,'NumberOfReturns') -1
    AND PC_Get(pt,'Intensity') < 150
```

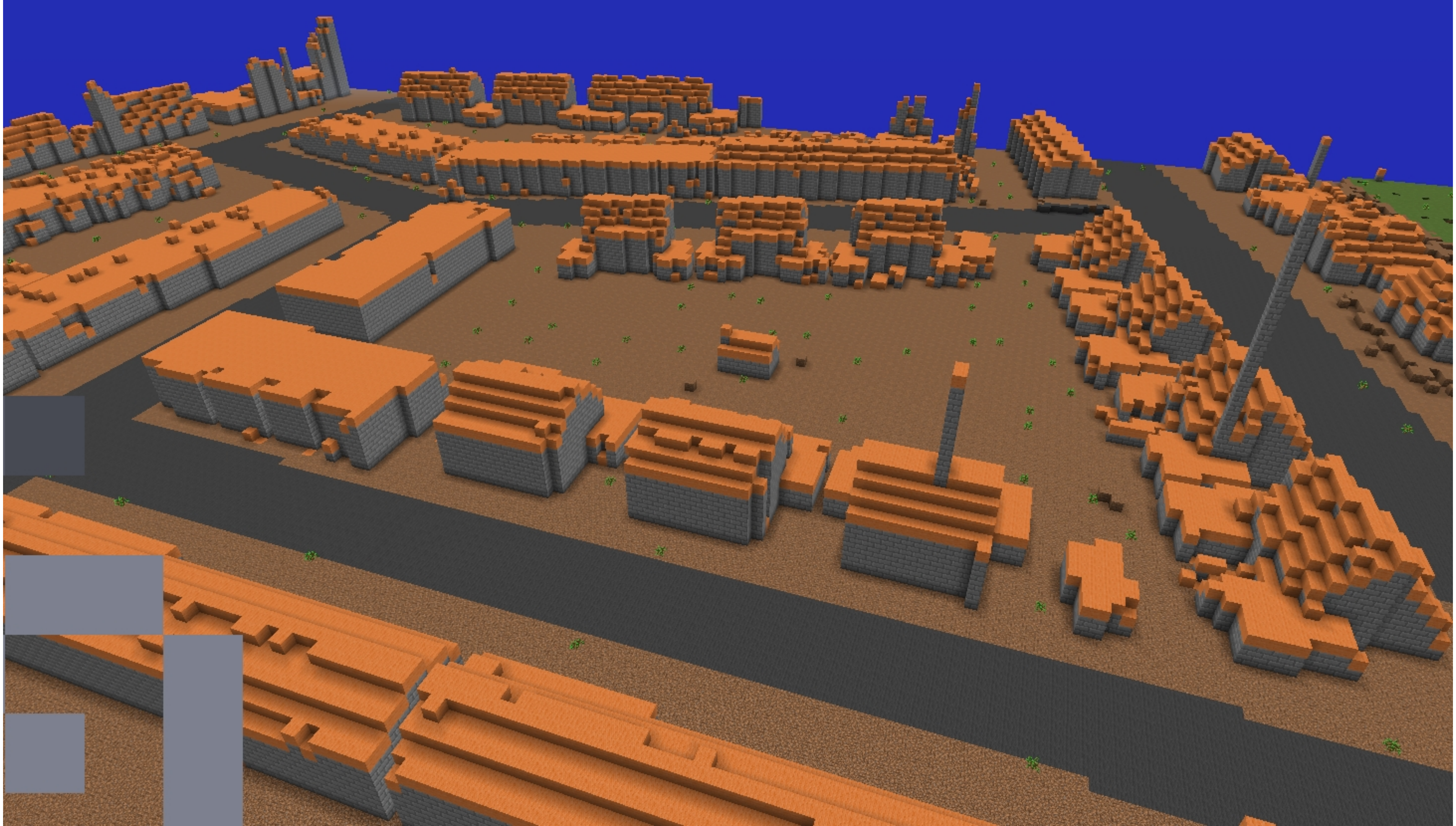
BGT 3D



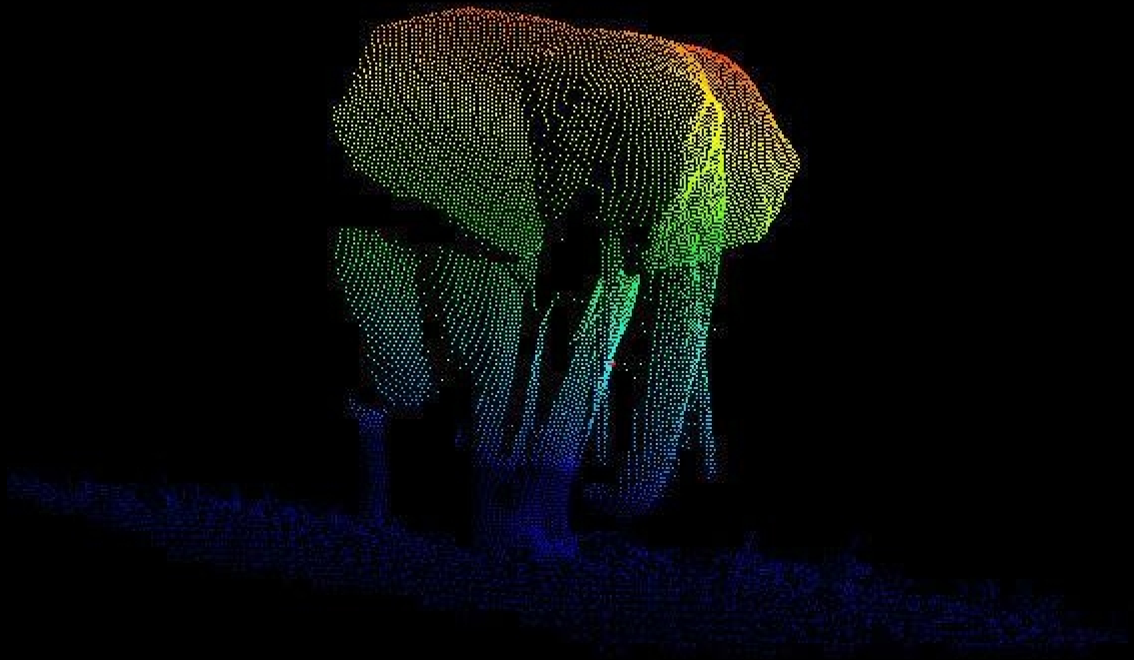

tree_growth
● -22.11 - 0.12
● 0.12 - 0.55
● 0.55 - 1.03
● 1.03 - 1.83
● 1.83 - 7.09
Zoeterwoudsesingel
Fruinlaan
P.J. Blokstraat
Cobetstraat
Burggravenlaan
Roomburgerlaan
De Laat de Kanterstraat
Thorbeckestraat
LKV Trigon
Wasstraat
De Mey van Streefkerkstraat
Van Vollenhovenkade

Use case 3:

Make minecraft levels



Fin



Discussion

Is a database really the best place for massive pointcloud storage ?

Would a columnstorage perform better in loading and indexing ?

Use-cases, use-cases, uses-cases……