netherlands eScience center
by SURF & NWO

TU Delft

nD-PointCloud, what else

nD-PointCloud

Peter van Oosterom, Martijn Meijers, Vitali Diaz Mercado, Edward Verbree (TU Delft)
Thijs van Lankveld, and Nauman Ahmed (Netherlands eScience Center)

Congres AHN & Beeldmateriaal, 16 October 2025, Amersfoort

Motivation nD-PointCloud

- point cloud data sets are often used for monitoring
  → dynamic point clouds
  → time added as additional organizing dimension

- organizing point cloud data in levels of Importance (LoI) is an approach to manage large data sets
  → LoI: discrete (multi-scale/dLoI) or continuous (vario-scale/cLoI)
  → scale treated as additional organizing dimension

- how to manage higher dimensional point clouds (4D, 5D, …)?

Agenda

- **nD-PointCloud**
  - **foundations**
  - **PostgreSQL implementation**
  - **nD Convex polytope query**
  - **Apache Parquet**

AHN3 colored by 'intensity gradient'

- practical results
  - Potree conversions AHN 1 to 5
  - COPC/VPC conversions
  - fast direct point cloud-based change detection

- future work
  - main publications

nD-PC foundations

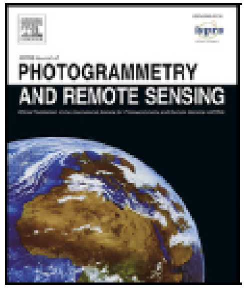
ISPRS Journal of Photogrammetry and Remote Sensing 194 (2022) 119–131

ELSEVIER

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs



Check for updates

Organizing and visualizing point clouds with continuous levels of detail

Peter van Oosterom [a,*], Simon van Oosterom [b], Haicheng Liu [c], Rod Thompson [a], Martijn Meijers [a], Edward Verbree [a]

[a] *Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, The Netherlands*
[b] *Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands*
[c] *Alibaba Group, China*

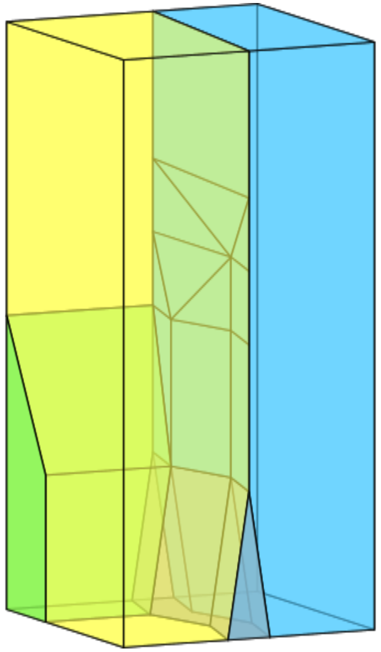
ARTICLE INFO

Keywords:
nD point clouds
Continuous level of detail (cLoD)
Space Filling Curve (SFC)
Perspective view selection

ABSTRACT

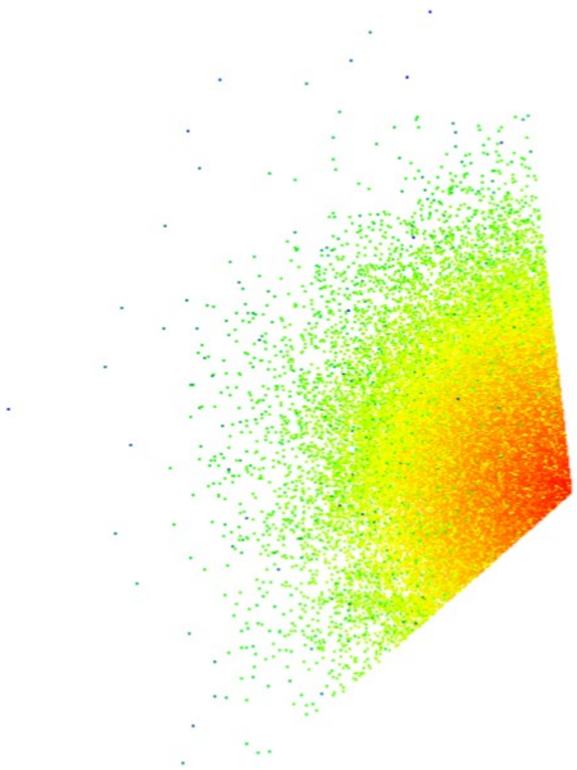
Point clouds contain high detail and high accuracy geometry representation of the scanned Earth surface parts. To manage the huge amount of data, the point clouds are traditionally organized on location and map-scale; e.g. in an octree structure, where top-levels of the tree contain few points suitable for small scale overviews and lower levels of the tree contain more points suitable for large scale detailed views. The drawback of this solution is that it is based on discrete levels, causing visual artifacts in the form of data density shocks when creating

Vario-scale for point cloud data



- lesson from vario-scale research: add one continuous dimension to the vector geometry to represent scale -> continuous LoI (2D data vario-scale represented by 3D geometry)

- apply this to point cloud data
  1. compute the cLoI value
  2. add this as organizing dimension, either x,y,imp (z and others attributes) or x,y,z,imp (and others as attributes) or …
  3. Cluster/index the 3D, 4D, .. nD points
  4. Define perspective view selections, view frustum with one more dimension: the further, the higher cLoI's



cLoI computation, getting rid of discrete levels → real continuous levels, nD case

- for ideal continuous distribution function over levels (nD):

$$f(l,n) = \frac{2^{(n-1)l}(n-1)\ln 2}{2^{(n-1)(L+1)}-1}$$

for l between 0 and $L+1$
and n number of dimensions

- this function has Cumulative Distribution Function (CDF):

$$F(l,n) = \frac{2^{(n-1)l}-1}{2^{(n-1)(L+1)}-1}$$

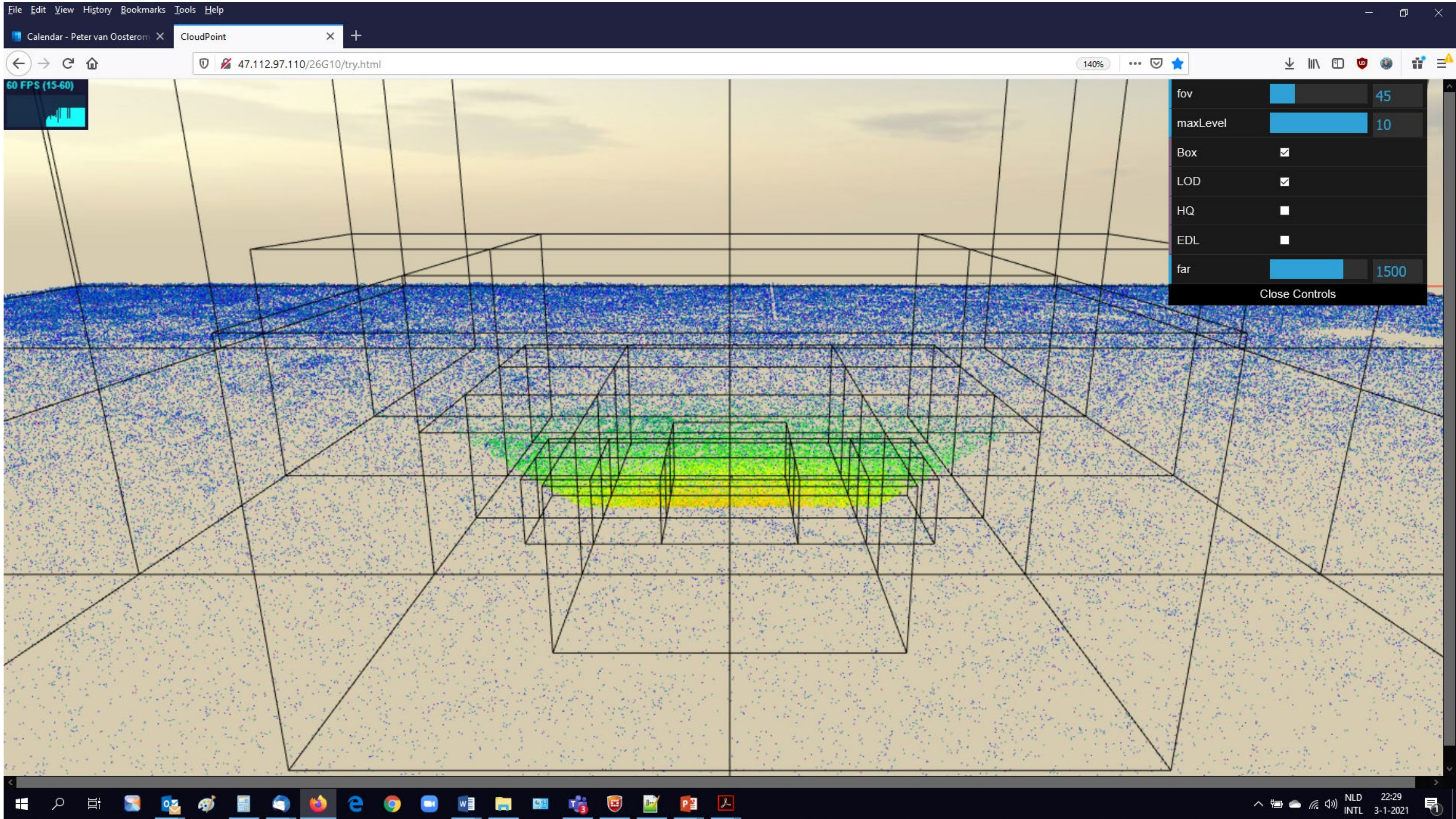
for l between 0 and $L+1$
and n number of dimensions

- using random generator U (uniform between 0 and 1) to generate level l **(cLoI)** (between 0 and $L+1$) for next point in nD space:

$$l = \frac{\ln((2^{(n-1)(L+1)}-1)U+1)}{(n-1)\ln 2}$$

cLoI in 3D web application

developed by Xuefeng Guan (Wuhan Univ) after 1 year visit TUD



note: colour is density (3rd person view)

Cluster/index by Space Filling Curves (SFCs)

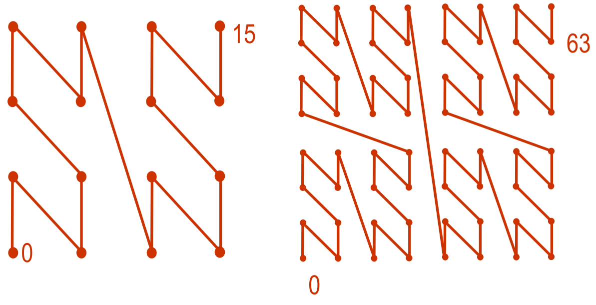
- from nD to 1D (computer memory addresses) and back…

- apply linear ordering to a multidimensional domain (spatial clustering)
- organize a flat table efficiently
- full resolution keys: avoid storing x,y[,z] + t/cLoI
  → recovered from SFC key

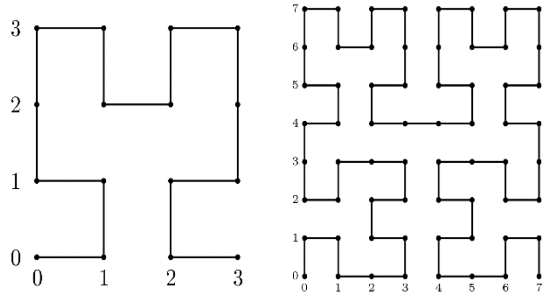
- use Index Organized Table (cluster data stored in the B-Tree index)
- queries need to be re-written to SFC-ranges, benefit from spatial clustering → efficient

- SFCs based on hyper-cubes
  - Morton/Hilbert both nD and quadrant recursive
  - Consider relative scaling of dimensions
  - Space reserved on the hypercube for future data

Morton (Peano)

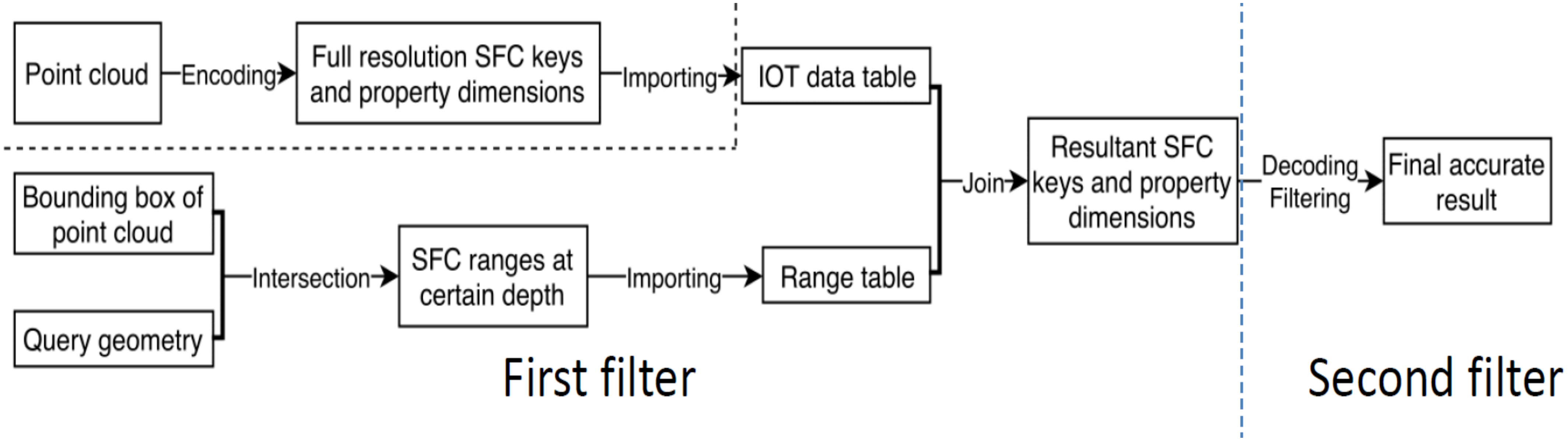


Hilbert



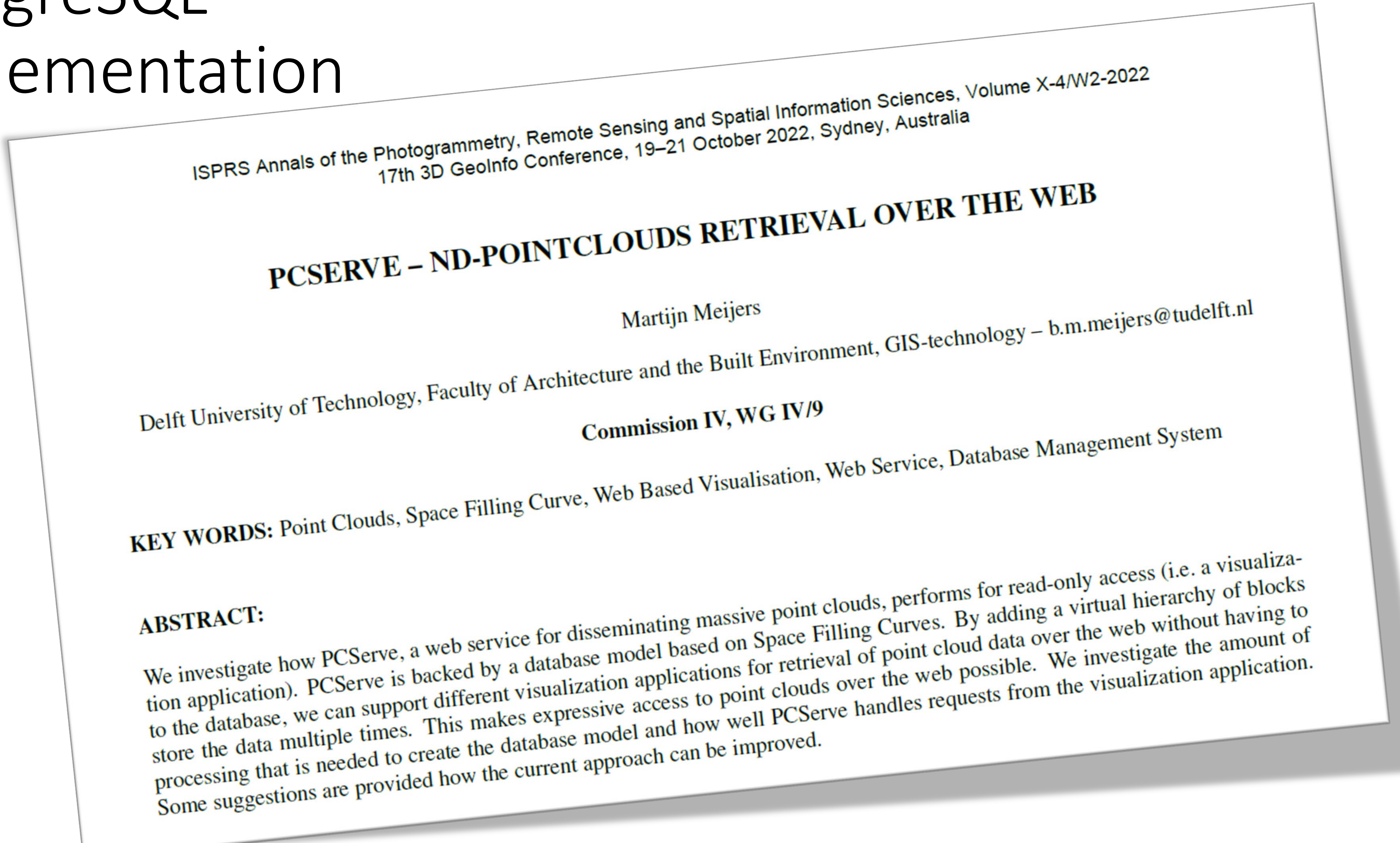
Overview: load & query point clouds

1. loading (upper right of figure)
    - convert nD points to SFC keys
    - store in IOT table (organized on SFC key)



2. querying (remainder)
    - convert query geometry to SFC ranges,
    - first filter: join SFC range table with IOT table
    - second filter (optional): test points inside query geometry

PostgreSQL Implementation



ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W2-2022
17th 3D GeoInfo Conference, 19–21 October 2022, Sydney, Australia

PCSERVE – ND-POINTCLOUDS RETRIEVAL OVER THE WEB

Martijn Meijers

Delft University of Technology, Faculty of Architecture and the Built Environment, GIS-technology – b.m.meijers@tudelft.nl

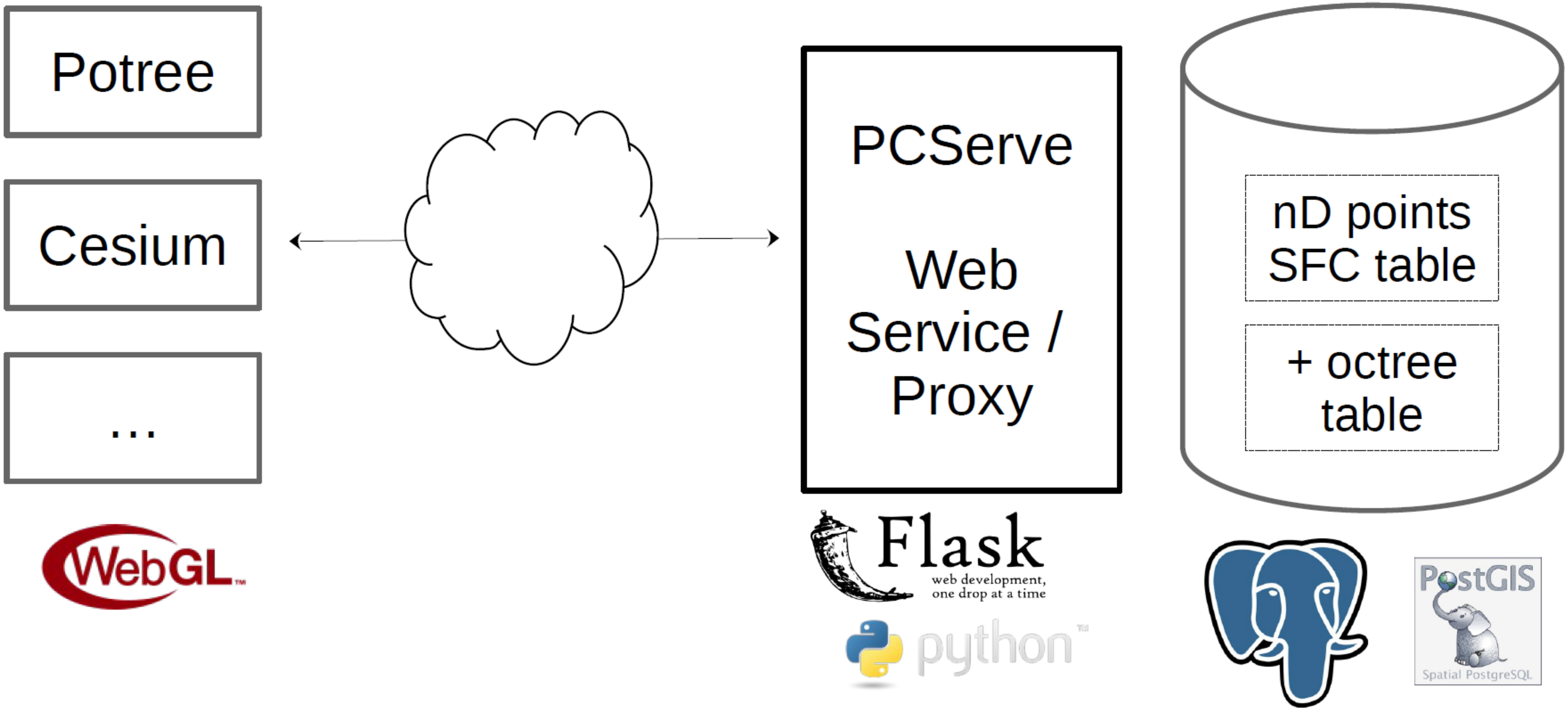
Commission IV, WG IV/9

KEY WORDS: Point Clouds, Space Filling Curve, Web Based Visualisation, Web Service, Database Management System

ABSTRACT:

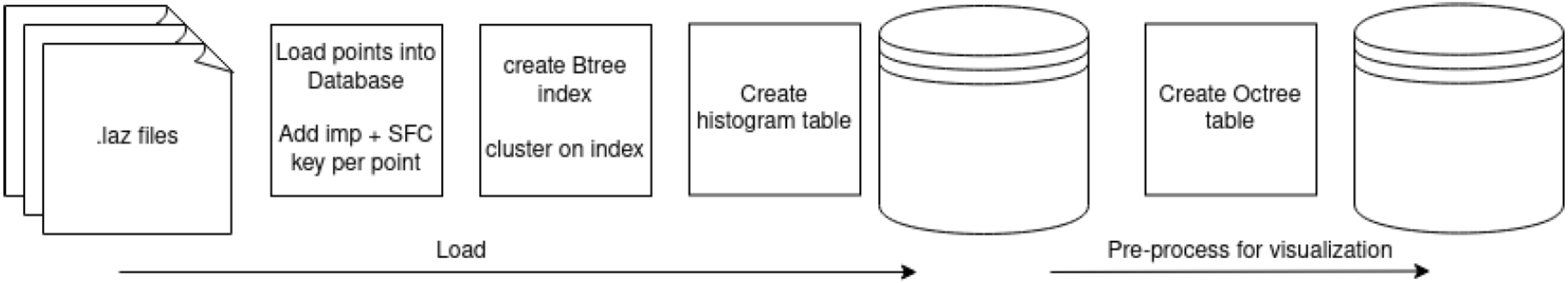
We investigate how PCServe, a web service for disseminating massive point clouds, performs for read-only access (i.e. a visualization application). PCServe is backed by a database model based on Space Filling Curves. By adding a virtual hierarchy of blocks to the database, we can support different visualization applications for retrieval of point cloud data over the web without having to store the data multiple times. This makes expressive access to point clouds over the web possible. We investigate the amount of processing that is needed to create the database model and how well PCServe handles requests from the visualization application. Some suggestions are provided how the current approach can be improved.

Implementation: PCServe



PCServe: DB preparation

- load, add cLoD and organize by SFC
- **virtual octree** to support potree viewer as there is not yet a cLoI viewer
- on request generate octree nodes with SFC ranges



nD Convex polytope query

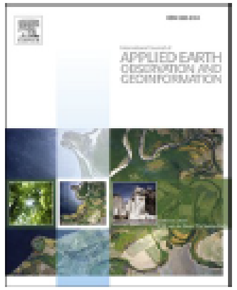
International Journal of Applied Earth Observations and Geoinformation 105 (2021) 102625

ELSEVIER

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Executing convex polytope queries on nD point clouds

Haicheng Liu [*], Rodney Thompson, Peter van Oosterom, Martijn Meijers

Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, the Netherlands

A R T I C L E I N F O

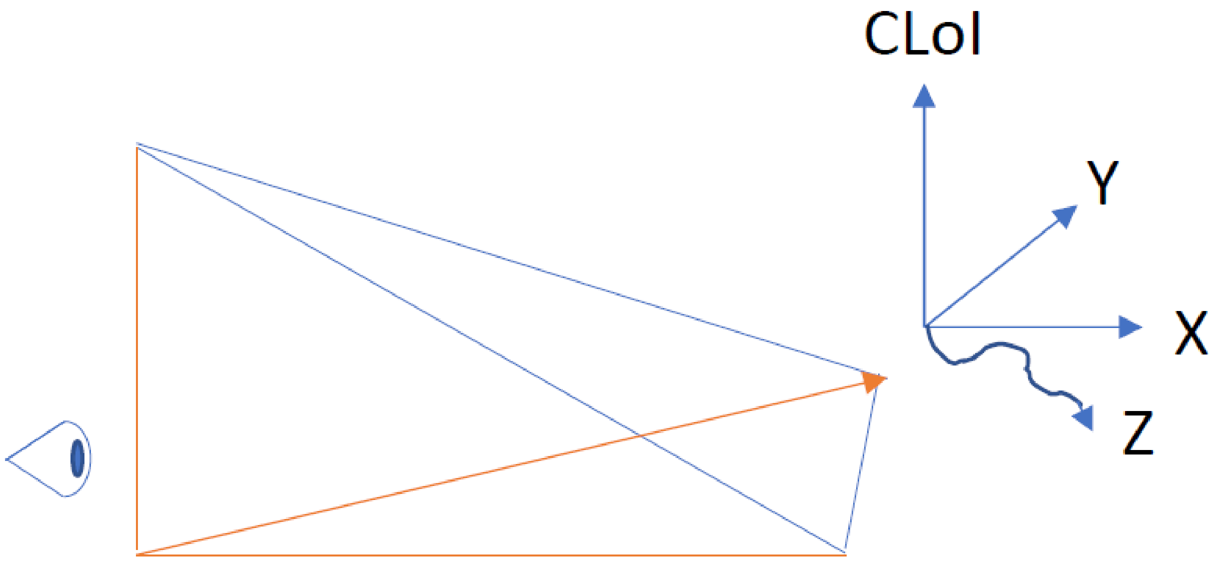
Keywords:
nD point clouds
Polytope query
Spatial data structures
CPLEX
Perspective view selection

A B S T R A C T

Efficient spatial queries are frequently needed to extract useful information from massive nD point clouds. Most previous studies focus on developing solutions for orthogonal window queries, while rarely considering the polytope query. The latter query, which includes the widely adopted polygonal query in 2D, also plays a critical role in many nD spatial applications such as the perspective view selection. Aiming for an nD solution, this paper first formulates a convex nD-polytope for querying. Then, the paper integrates three approximate geometric algorithms – SWEEP, SPHERE, VERTEX, and a linear programming method CPLEX, developing a solution based

nD query by convex polytopes

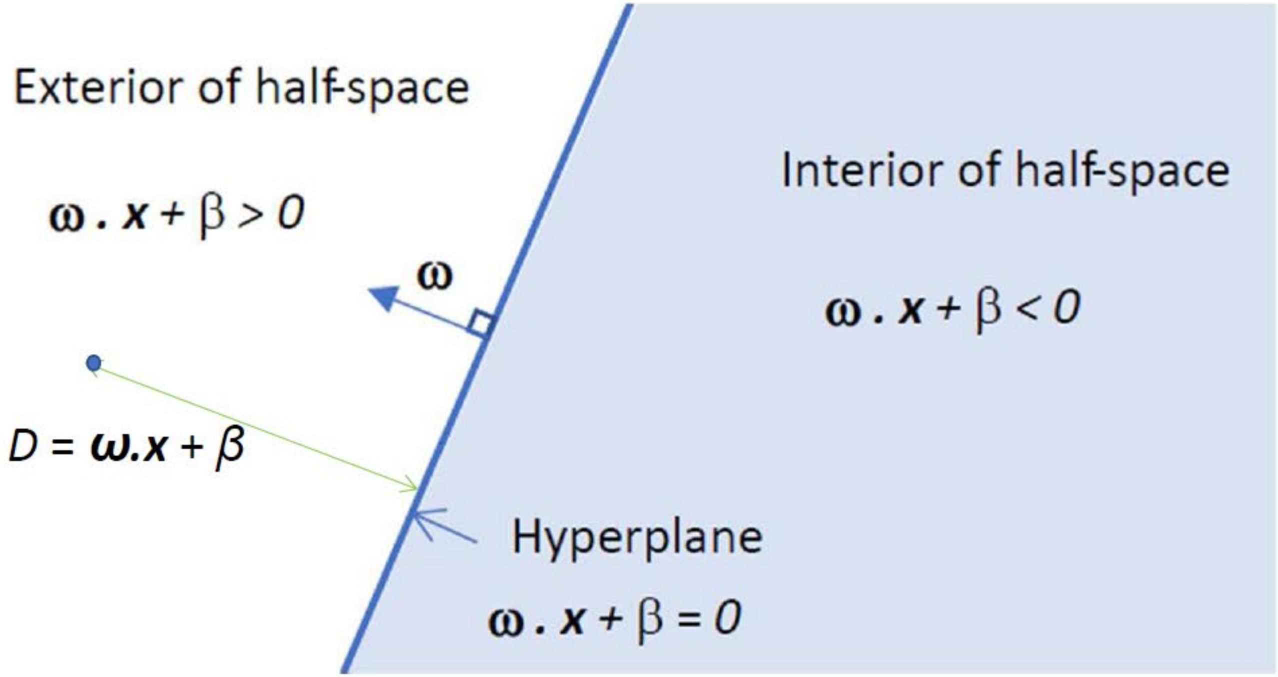
- convex polytope is more selective than a hyper-rectangle/sphere
- well defined in nD, based on half-space intersection
- fast implementation based on SFC/virtual 2^n-tree
- example view frustum selection



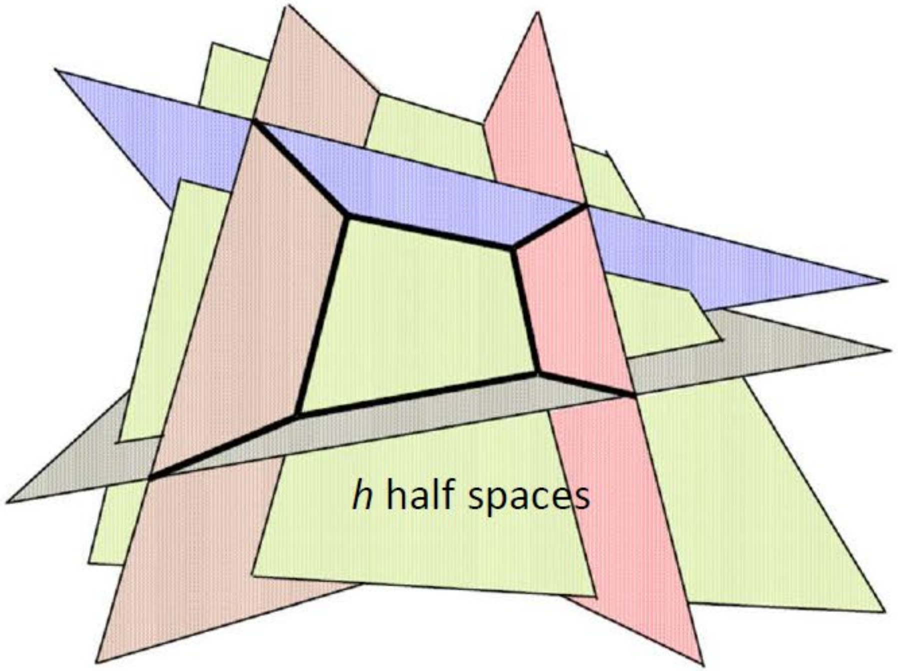
4D View Frustum with CLoI
(only drawn in 3D)

Test overlap half-space with virtual node 2^n-tree

- half-space = nD plane inequation
- virtual node 2^n-tree = hyperbox (range of SFC-keys)
- plane normal vector used which 2 corners hyperbox used for test



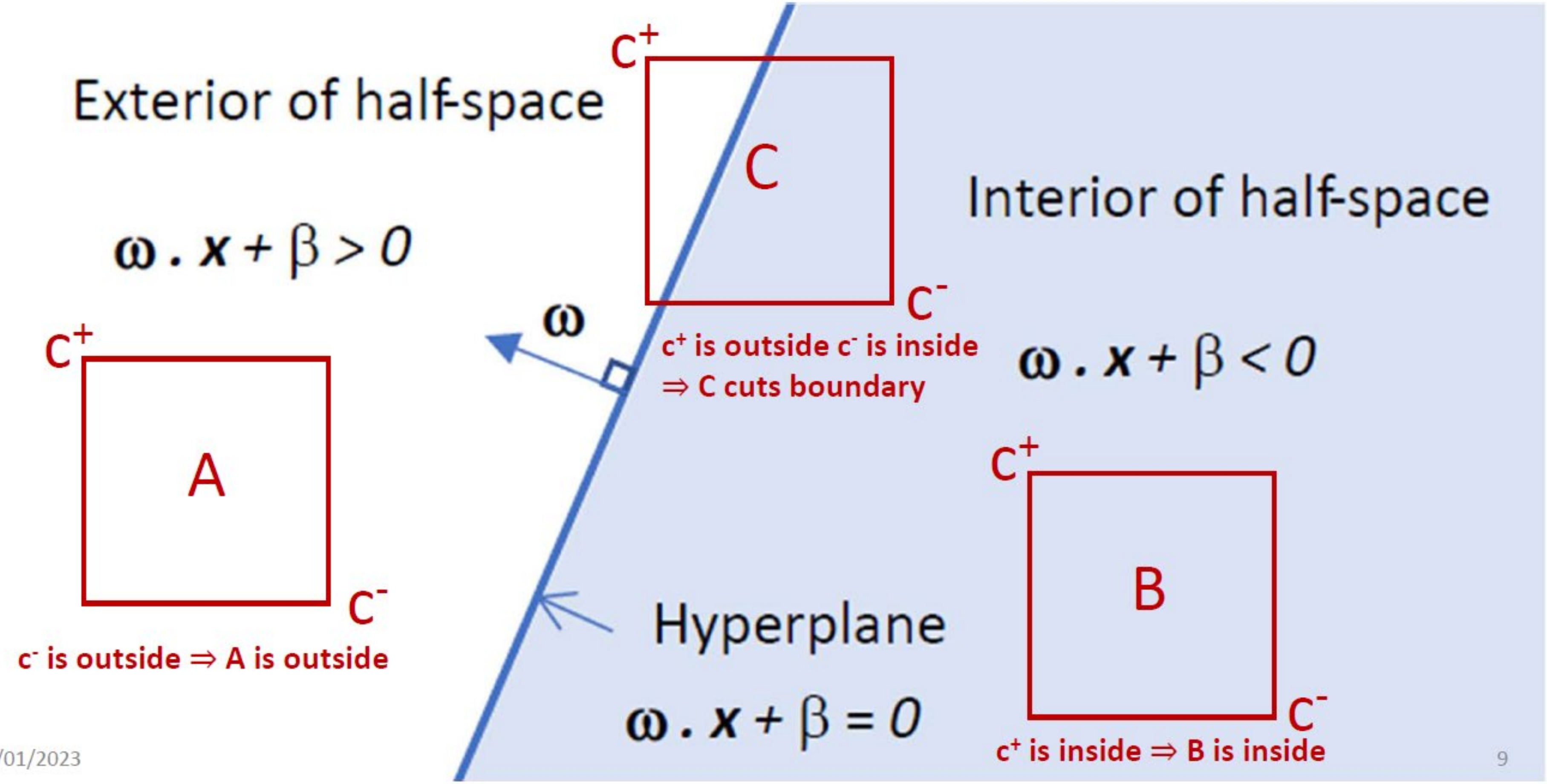
This question "Is a point within a halfspace" is of $O(n)$ in nD space



So the question of "Is a point within a region defined by h half spaces" is $O(nh)$.

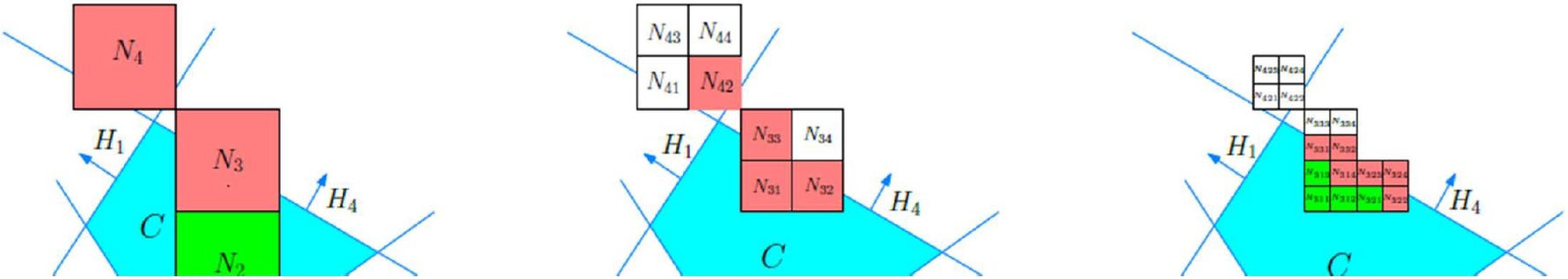
Just 2 corner points

- test nD box inside nD half-space: out, partial, in
- in nD box there are 2^n points (e.g. 5D → 32 points), but just 2 tested



Full query: recursive 2^n-tree traversal

- in-box give SFC-range (based on level in tree this can be larger/smaller)
- partial-box is refined (unless max depth/bottom level is reached)
- resulting set of SFC ranges is joined with SFC based nD points
  → either report individual points, or point groups

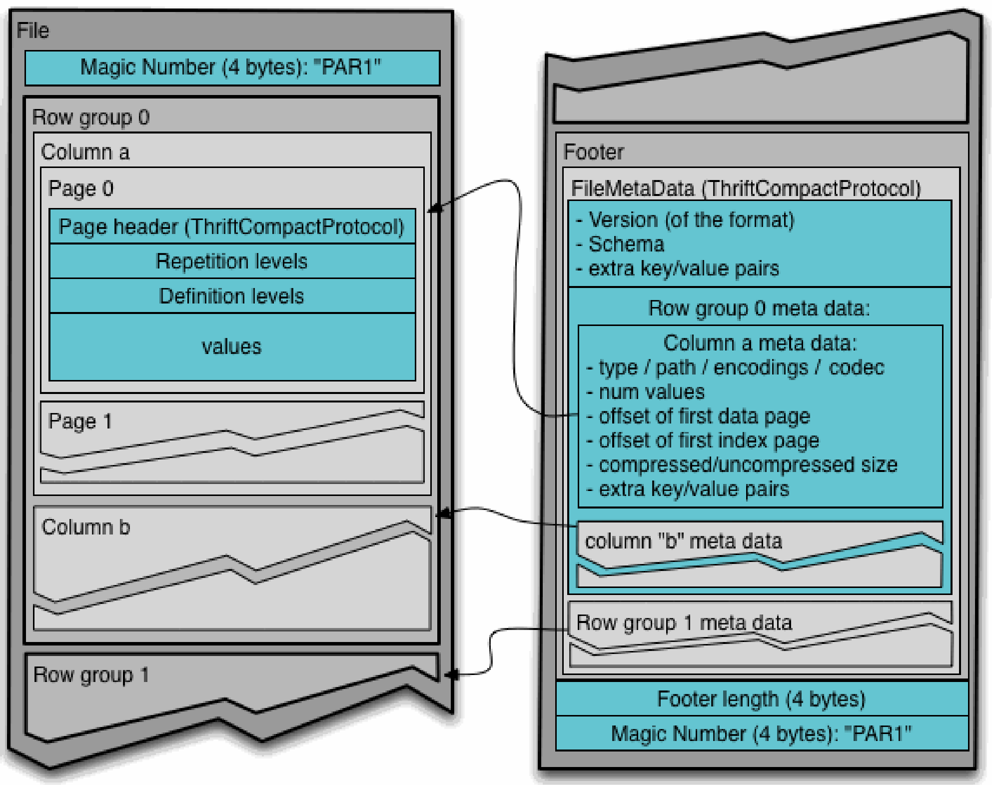


Apache Parquet (as alternative store)

- columnar store format, for serialized data (disk storage)
- structured (+ nested) data, compressed
- has a main memory cousin (uncompressed columnar)
  - Apache Arrow → Data frame libraries can read Parquet from disk into Arrow(2D, heterogeneous tabular data)
- you can interact via SQL with the Parquet files using query engine (DuckDb, Apache Drill)

Apache Parquet - structure

- table split over:
  - Row groups → Columns
  - Column Chunks → Pages
- encoding data + Compression at Page level
- metadata on Min/Max of Row groups (+ recent addition: Pages), stored at end of file (block ranges index, if data sorted!)
- can be fetched in parts with http head followed by http range requests (if server support)
- C++, Java, Rust libraries read / write
- https://parquet.apache.org/



Some nD PC data (load similar to PostgreSQL)

SFC key **Attributes**

sfc_key blob	gps_time double	scan_angle float	intensity uint16	return_number uint8	number_of_returns uint8	classification uint8	scan_direction uint8	is_edge_of_flight_line uint8
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\x8ES\xC…	494147.3131107338	27.0	170	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\xFC\x04…	493868.6779278455	16.002	118	1	1	2	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\xFC\xCD…	494146.86342112836	27.0	22	1	1	2	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\xFC\xF9…	494146.83669732255	27.0	37	1	1	2	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\xFF,2\xF8	493869.0259201808	16.002	64	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\xFF\xCB…	494146.41376256285	28.002	30	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,N\xFF\xD7…	493868.97246252943	15.0	184	1	1	2	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,O\xF7X\x0…	493870.1446405349	16.002	230	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,P\x0E\x8C…	493870.0054991533	15.0	66	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,P\x0E\xA0…	493870.1339485246	16.002	195	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,P\x0E\xBB…	493870.0858279384	15.0	48	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,P\x0F;\xF1&	493870.03758022806	16.002	200	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,P\x10o\x8A}	493869.855603322	16.002	47	1	1	6	1	0
\x00\x00\x00\x00\x10\x02~\x8C\xC3,P\x10r\x93=	493869.74327855057	13.998	200	1	1	2	1	0
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7ws…	164798893.0589545	-10.002	171	1	1	2	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wt…	164798893.02829105	-10.002	16	1	2	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wt…	164798893.07124695	-10.002	39	1	2	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wt…	164798893.0466985	-10.002	129	1	1	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wu…	164798892.99145013	-10.002	49	2	2	2	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wupz	164798892.9668992	-10.002	60	1	2	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wu…	164798892.96690187	-10.002	18	1	2	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wu…	164798892.9668992	-10.002	133	2	2	2	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wu…	164798892.98529118	-9.0	95	1	3	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wv…	164798892.99755764	-9.0	164	2	2	2	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7wvy(	164798892.99755764	-9.0	43	1	2	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7ww3M	164798893.0466725	-9.0	23	1	2	1	1	0
\x00\x00\x00\x00\x10\x02~\x9F\xFF\xEB\xD7ww7g	164798893.0589441	-9.0	155	1	1	2	1	0

6083217 rows (40 shown) 9 columns

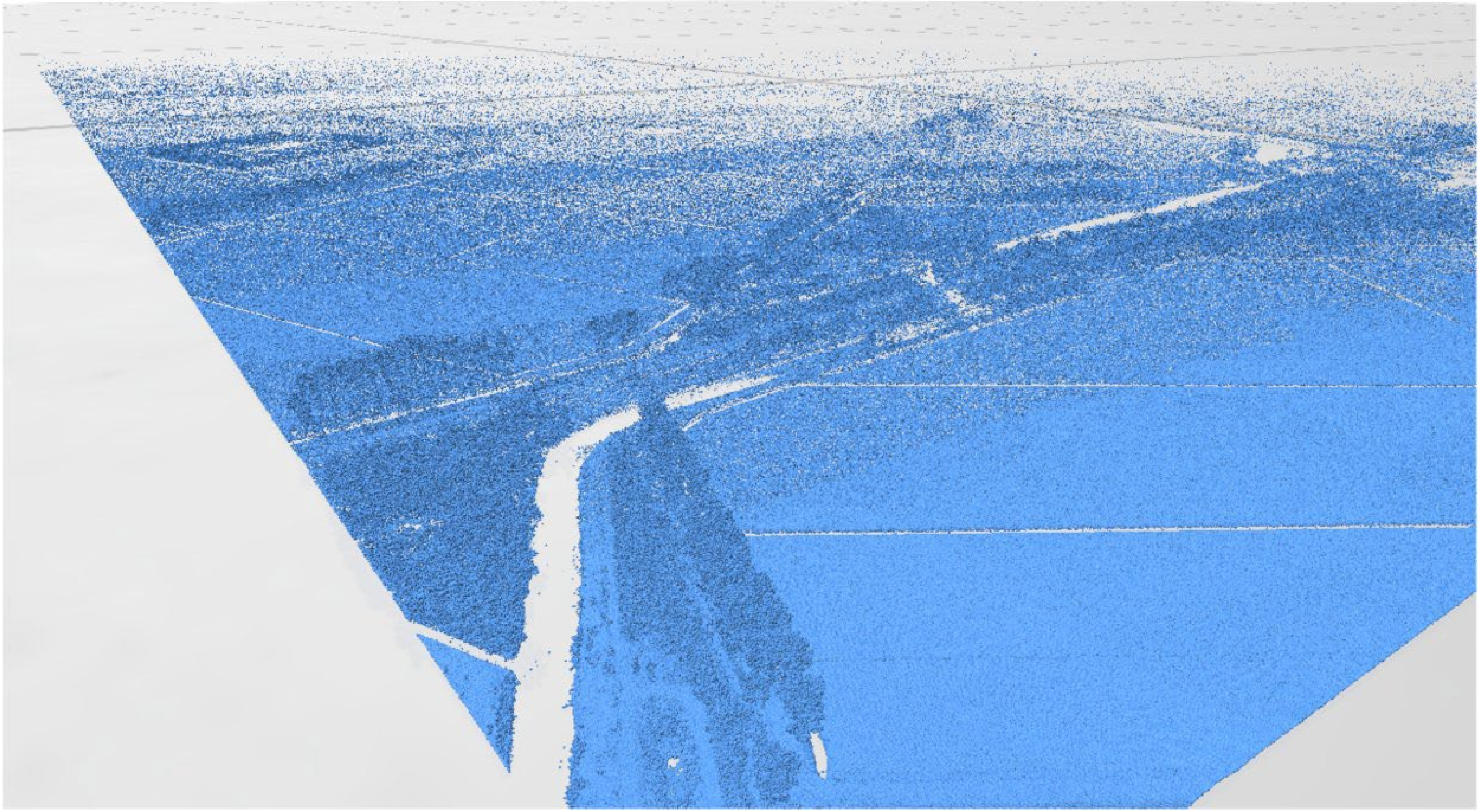
Row groups + **Columns** = Column Chunks

Querying Parquet

- Column Chunks divided into Pages
- encoding/compression can differ per column data type

- custom merge join with half spaces→query ranges and data (SFC keys)Read relevant parts from Parquet file, that contains sorted data
- heavy use of Min/Max index data on both Row groups as well as Column pages in footer of Parquet file
- fetch in phases:
  1. matching records based on SFC key column (a. Row Group → b. Column page)
  2. retrieve additional attributes for query result points from disk
- decode / de-quantize keys

Integrated Space / cLoI query

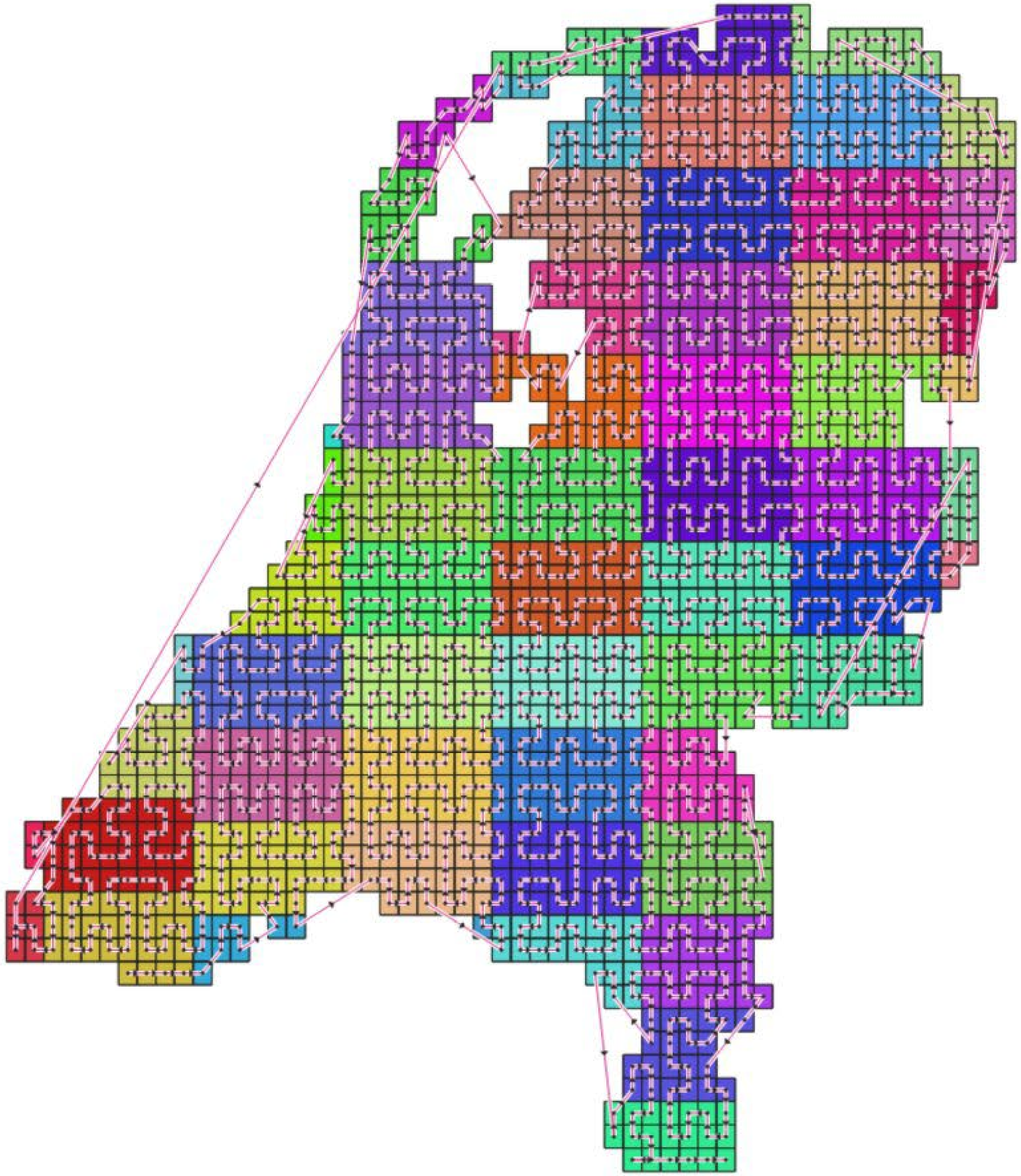


Agenda

- nD-PointCloud
  - foundations
  - PostgreSQL implementation
  - nD Convex polytope query
  - Apache Parquet

- **practical results**
  - **Potree conversions AHN 1 to 5**
  - **COPC/VPC conversions**
  - **fast direct PC-based change detection**

- future work
  - main publications



Dell PowerEdge R7525→ ronna
physical server for benchmarks, parallel computing

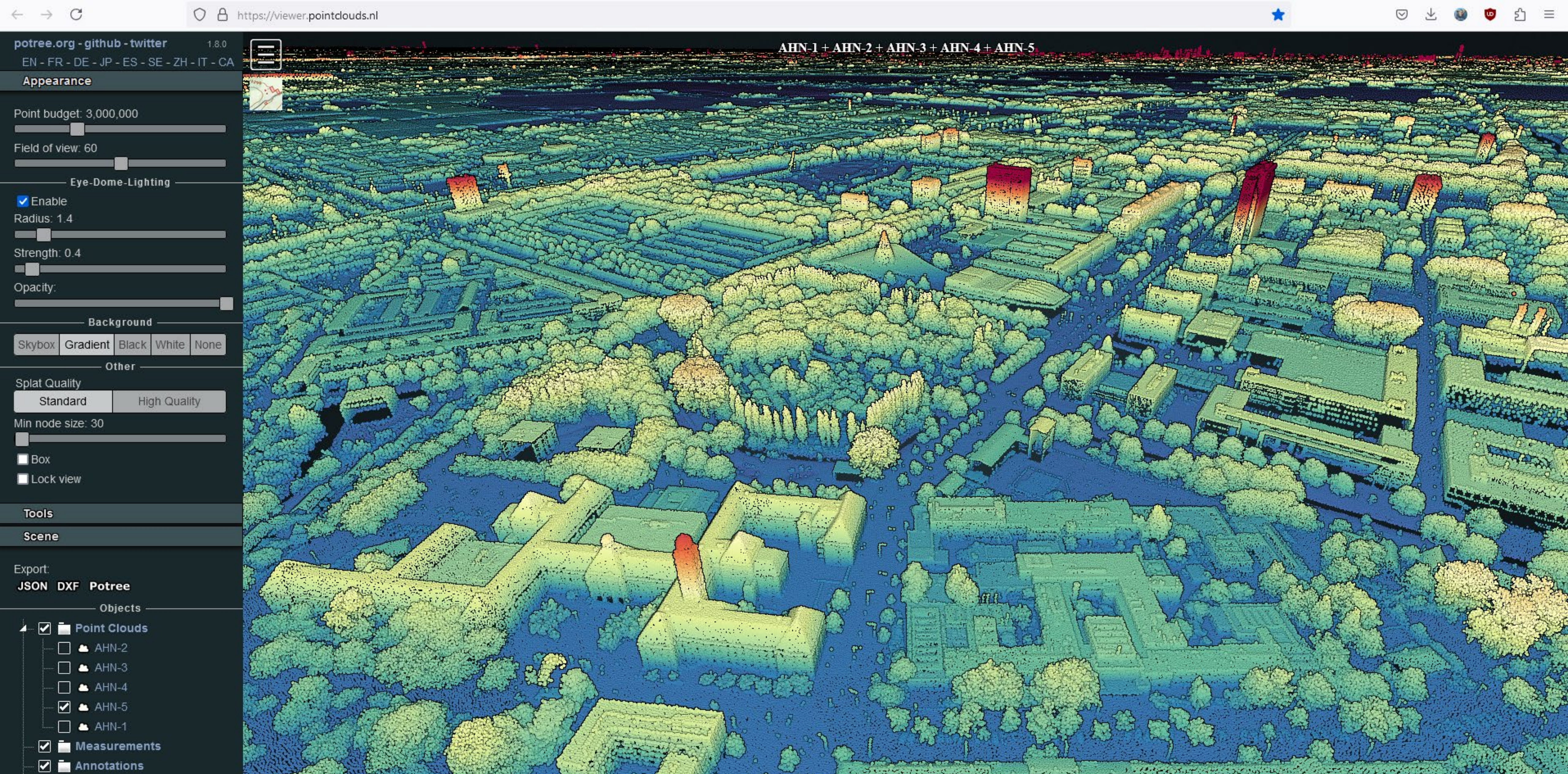
Dell PowerEdge R7525 server

1. 2 x 24-core AMD EPYC 7443 processors (96 threads), 2.85GHz
2. 512 GB buffered memory (RDIMM)
3. Ubuntu operating system

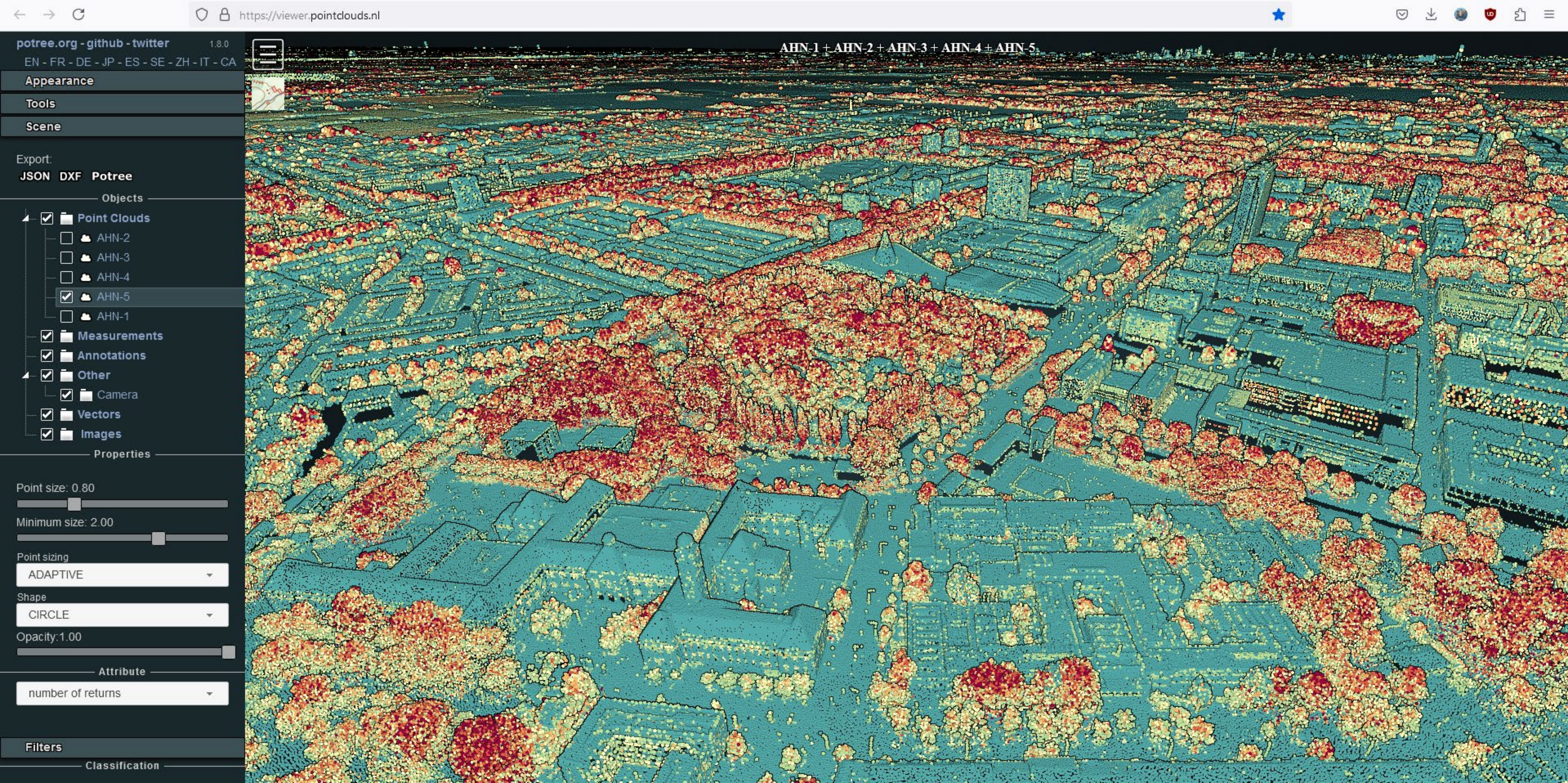
- Disk storage – direct attached
  1. 15,4 Tb SSD SAS
  2. 19,2 Tb SSD SATA
  3. 180,0 Tb Hard Drive SAS, 7.2K rpm in RAID-5

- Network NAS 26,0 TB on svm104.storage.tudelft.net

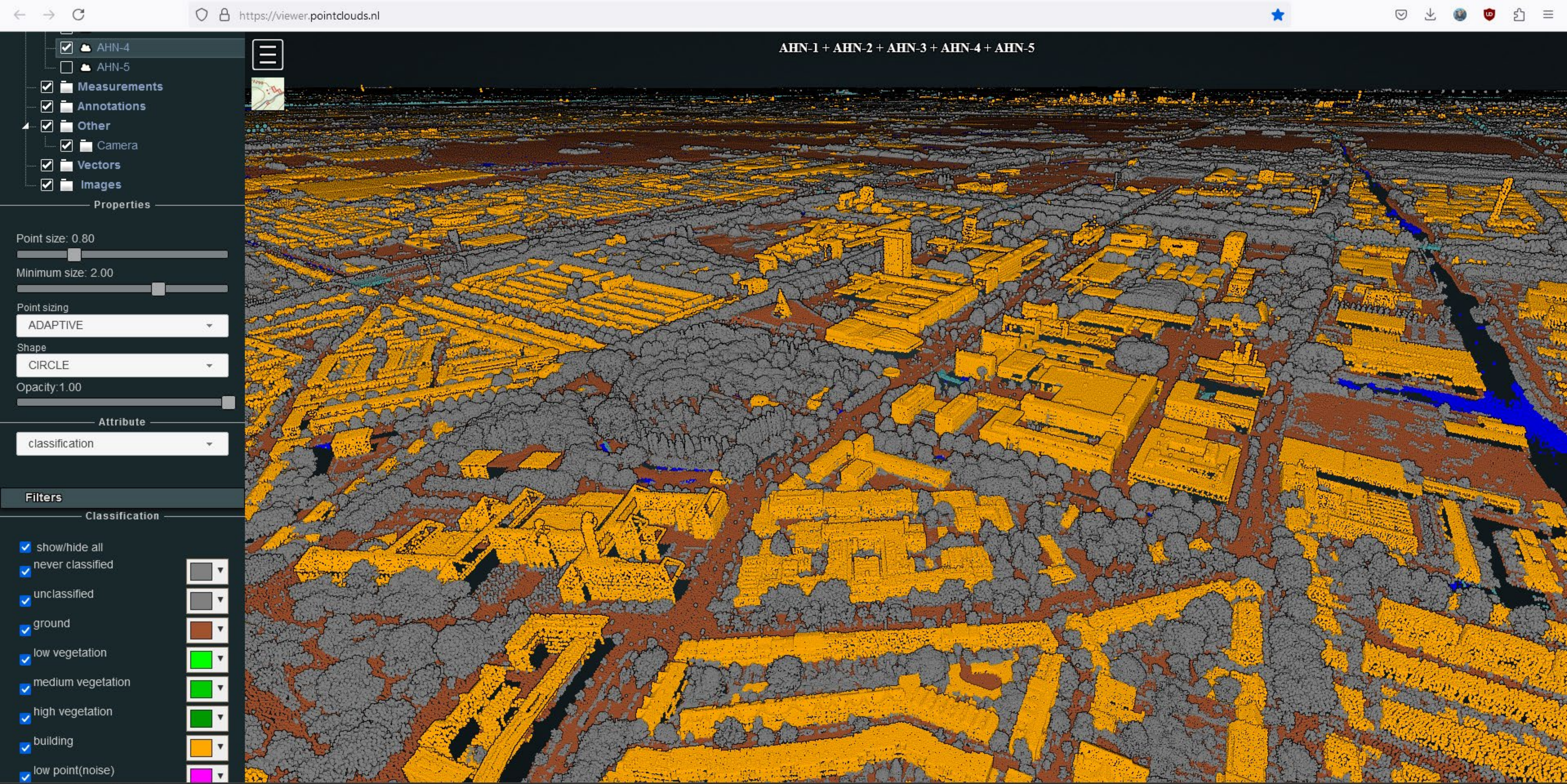
http://viewer.pointclouds.nl all AHN versions in potree



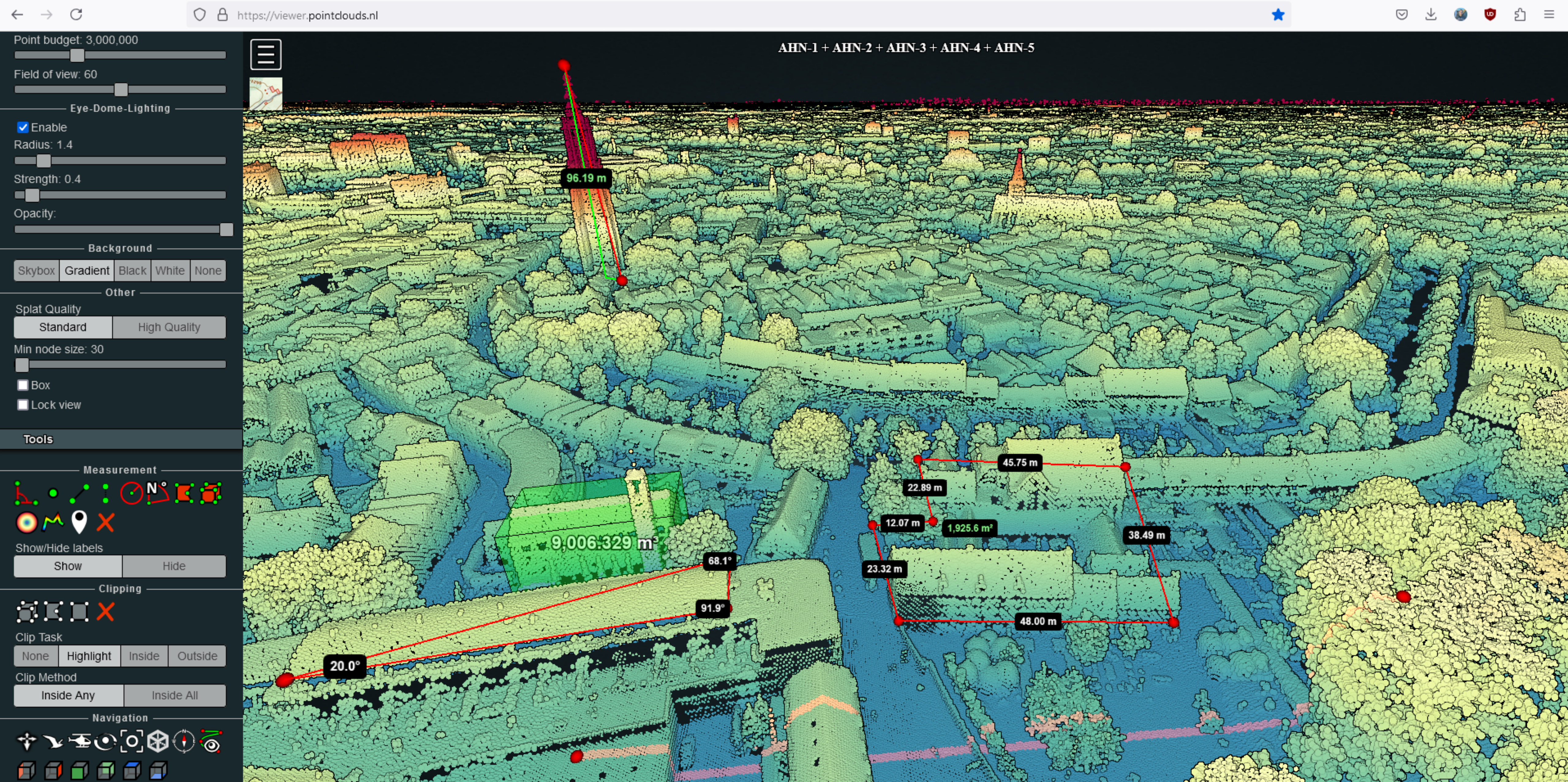
All potree tools, e.g. show number of returns : 1-7



.. or classification



Various tools: height, area, volume, angles, profile,..

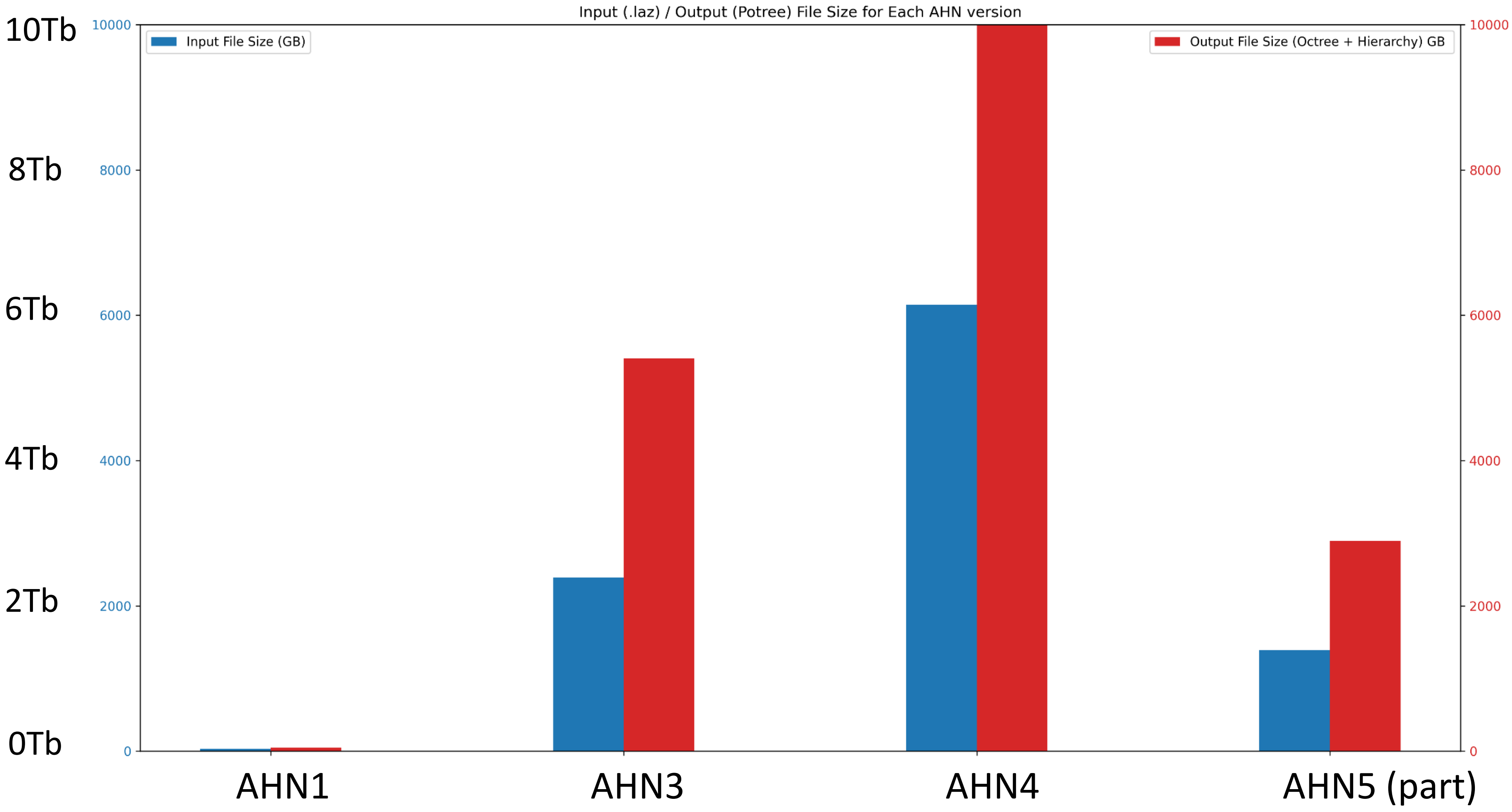


potree conversion on ronna

name	#points	#files in	Input size (Gb)	Output size (Gb)	Duration (hours)
AHN1	11.984.853.767	1358	33,1	44,2	0,17
AHN3	557.925.797.136	1374	2390,7	5035,8	12,29
AHN4	947.364.043.509	1381	6145,4	9931,8	33,38
AHN5 part	289.944.615.278	499	1394,3	2695,9	8,59

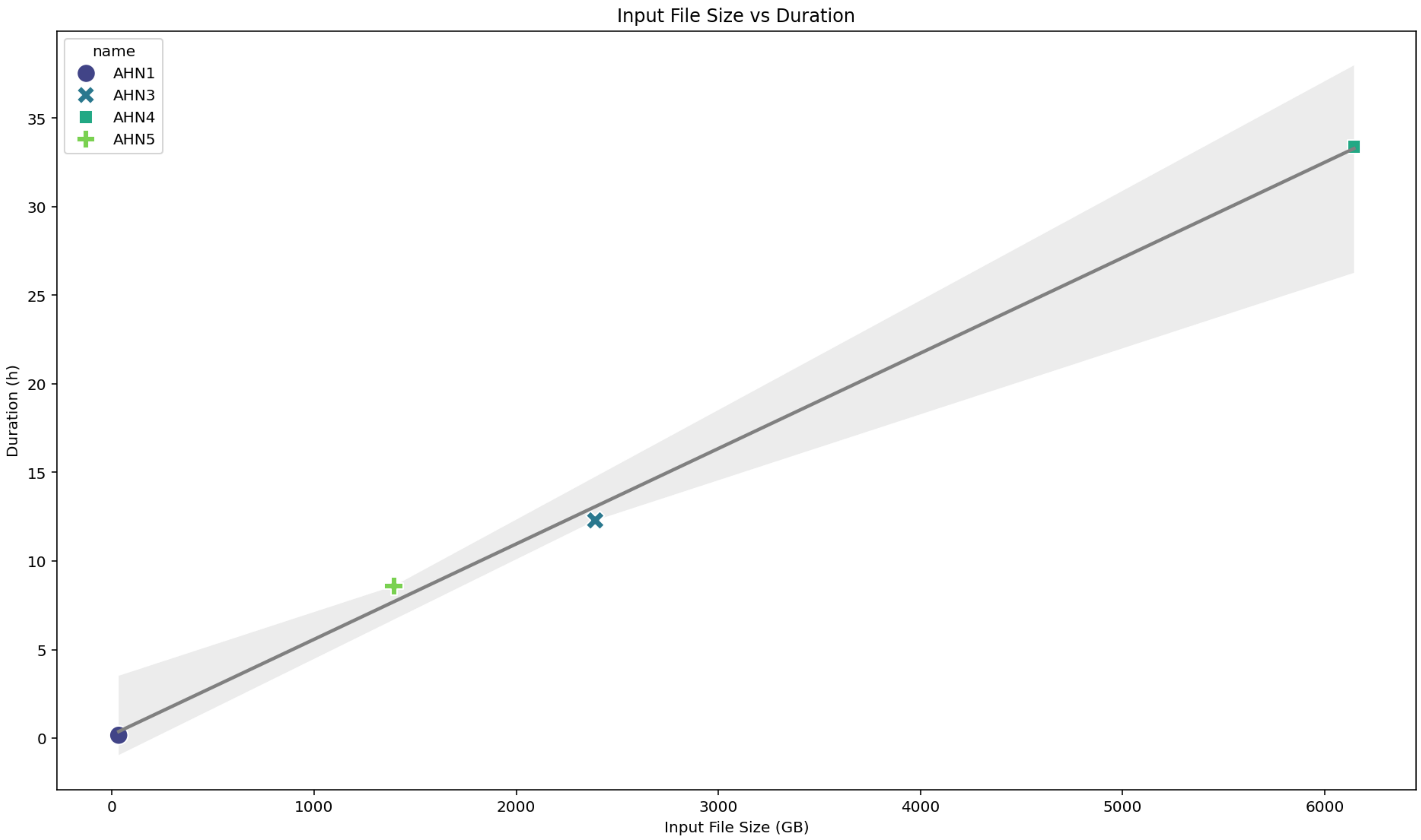
AHN2 converted before (previous project)
Duration conversion on ronna, multi-user

AHN-potree conversion statistics: in-/output size



Note about issue memory leak potree, due to parallel sort in library (in some systems/versions)

AHN-potree conversion: input size vs duration



Virtual Point Clouds and Cloud Optimized Point Cloud

- Virtual Point Cloud (VPC) file = SpatioTemporal Asset Catalogs (STAC)
- extension on top of GeoJSON with PC metadat
- contains links to COPC .laz files (or EPT, Entwine Point Tile)

- Cloud Optimized Point Cloud (COPC) file = data
- COPC is .laz which has embedded Octree
- specs on https://copc.io/ (by Hobu, Inc.)

- load VPC in **QGIS** as a single point cloud layer (rather than each file as a separate map layer)

Example VPC file

JSON Raw Data Headers

Save Copy Collapse All Expand All (slow) Filter JSON


```
type:                 "FeatureCollection"
▾ features:
  ▾ 0:
    type:             "Feature"
    stac_version:     "1.0.0"
  ▾ stac_extensions:
    ▾ 0:              "https://stac-extensions.github.io/pointcloud/v1.0.0/schema.json"
    ▾ 1:              "https://stac-extensions.github.io/projection/v1.1.0/schema.json"
    id:               "C01C.copc"
  ▾ geometry:
    ▾ coordinates:
      ▾ 0:
        ▾ 0:
          0:          5.17868789843972
          1:          53.38616375438354
          2:          39.53252257234287
        ▸ 1:          [_]
        ▸ 2:          [_]
        ▸ 3:          [_]
        ▸ 4:          [_]
      type:           "Polygon"
  ▸ proj:geometry:    {_}
  ▸ proj:bbox:        [_]
  ▾ properties:
      datetime:       "2014-09-15T00:00:00Z"
      pc:count:       295582241
      pc:encoding:    "?"
    ▸ pc:schemas:     [_]
      pc:type:        "lidar"
    ▸ proj:bbox:      [_]
    ▾ proj:geometry:
      ▾ coordinates:
        ▾ 0:
            0:        141125.351
            1:        600000
            2:        -1.872
          ▸ 1:        [_]
          ▸ 2:        [_]
          ▸ 3:        [_]
          ▸ 4:        [_]
        type:         "Polygon"
    ▾ proj:wkt2:      'COMPOUNDCRS["Amersfoort / RD New + NAP height",PROJCRS["Amersfoort / RD New",BASEGEOGCRS["Amersfoort",DATUM["Amersfoort",ELLIPSOID["Bessel 1841",6377397.155,299.1528128,LENGTHUNIT["metre",1,ID["EPSG",9001]],ID["EPSG",7004]],ID["EPSG",6289]],DEFININGTRANSFORMATION["Amersfoort to ETRS89 (9)",ID["EPSG",9282]],ID["EPSG",4289]],CONVERSION["RD New",METHOD["Oblique Stereographic",ID["EPSG",9809]],PARAMETER["Latitude of natural origin",52.1561605555558,ANGLEUNIT["degree",0.0174532925199433,ID["EPSG",9102]],ID["EPSG",8801]],PARAMETER["Longitude of natural origin",5.38763888888917,ANGLEUNIT["degree",0.0174532925199433,ID["EPSG",9102]],ID["EPSG",8802]],PARAMETER["Scale factor at natural origin",0.9999079,SCALEUNIT["unity",1,ID["EPSG",9201]],ID["EPSG",8805]],PARAMETER["False easting",155000,LENGTHUNIT["metre",1,ID["EPSG",9001]],ID["EPSG",8806]],PARAMETER["False northing",463000,LENGTHUNIT["metre",1,ID["EPSG",9001]],ID["EPSG",8807]],ID["EPSG",19914]],CS[Cartesian,2,ID["EPSG",4499]],AXIS["Easting (X)",east],AXIS["Northing (Y)",north],LENGTHUNIT["metre",1,ID["EPSG",9001]],ID["EPSG",28992]],VERTCRS["NAP height",VDATUM["Normaal Amsterdams Peil",ID["EPSG",5109]],CS[vertical,1,ID["EPSG",6499]],AXIS["Gravity-related height (H)",up],LENGTHUNIT["metre",1,ID["EPSG",9001]],GEOIDMODEL["ETRS89 to NAP height (2)",ID["EPSG",9283]],ID["EPSG",5709]],ID["EPSG",7415]]'
    links:            []
  ▾ assets:
    ▾ data:
      ▸ href:         "https://data.pointclouds…ahn-copc/3/C01C.copc.laz"
      ▾ roles:
```


STAC PointCloud
https://github.com/stac-extensions/pointcloud
https://stacspec.org/en

extent of COPC .laz in WGS'84

extent of COPC .laz in EPSG:7415 (RD+NAP)

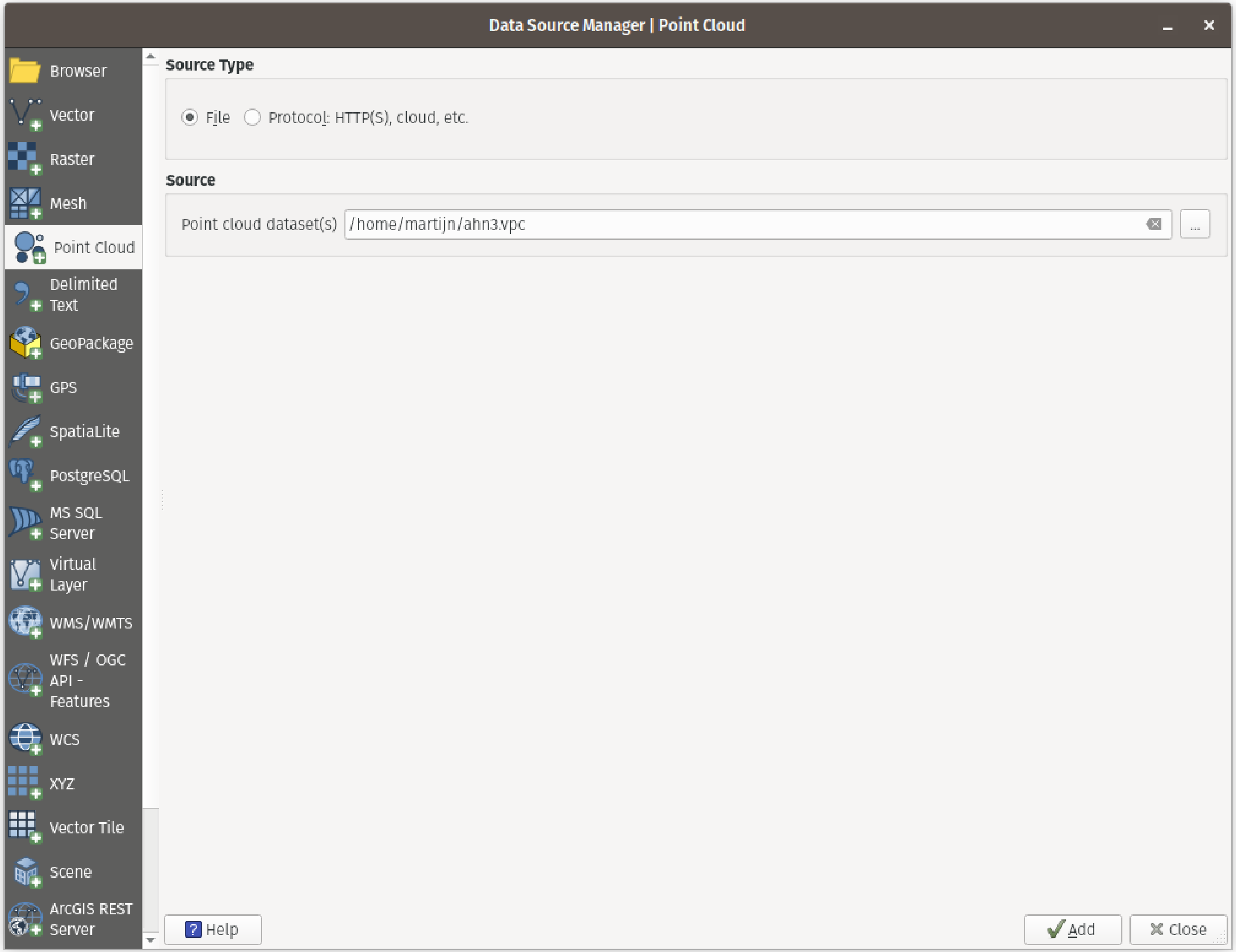
EPSG:7415 WKT2

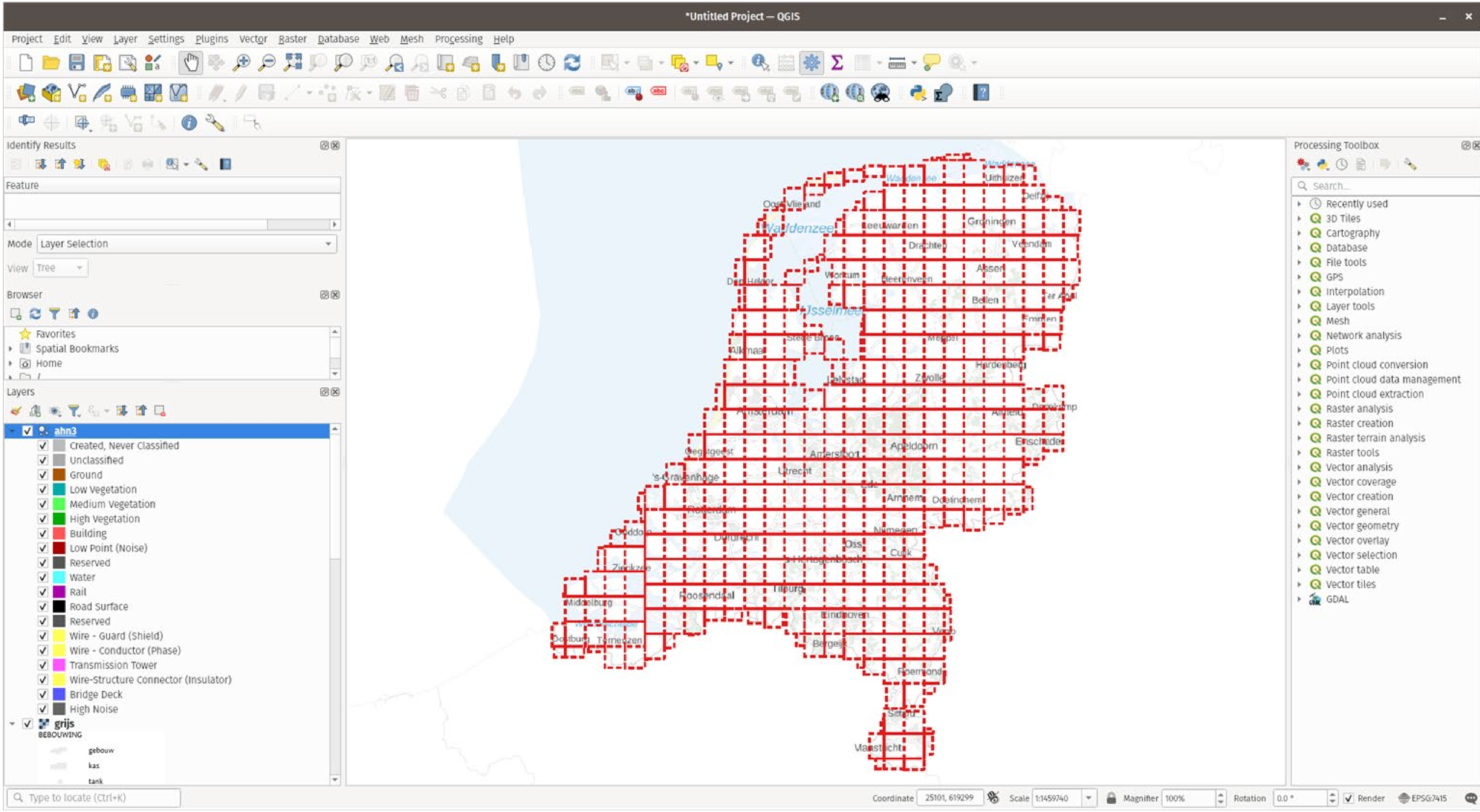
link to COPC .laz (which has embedded Octree)

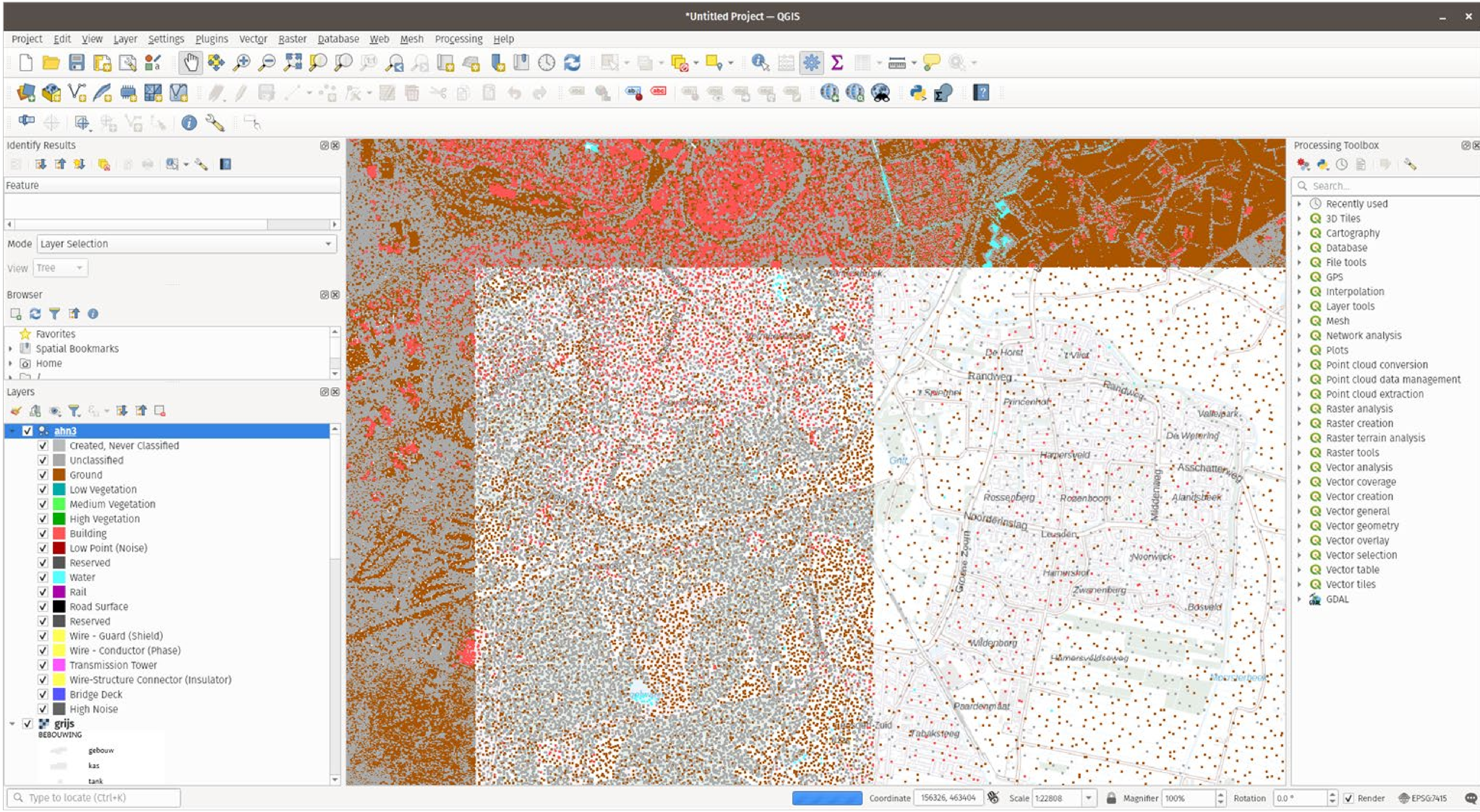
To use VPC/COPC in QGIS start with download .vpc file

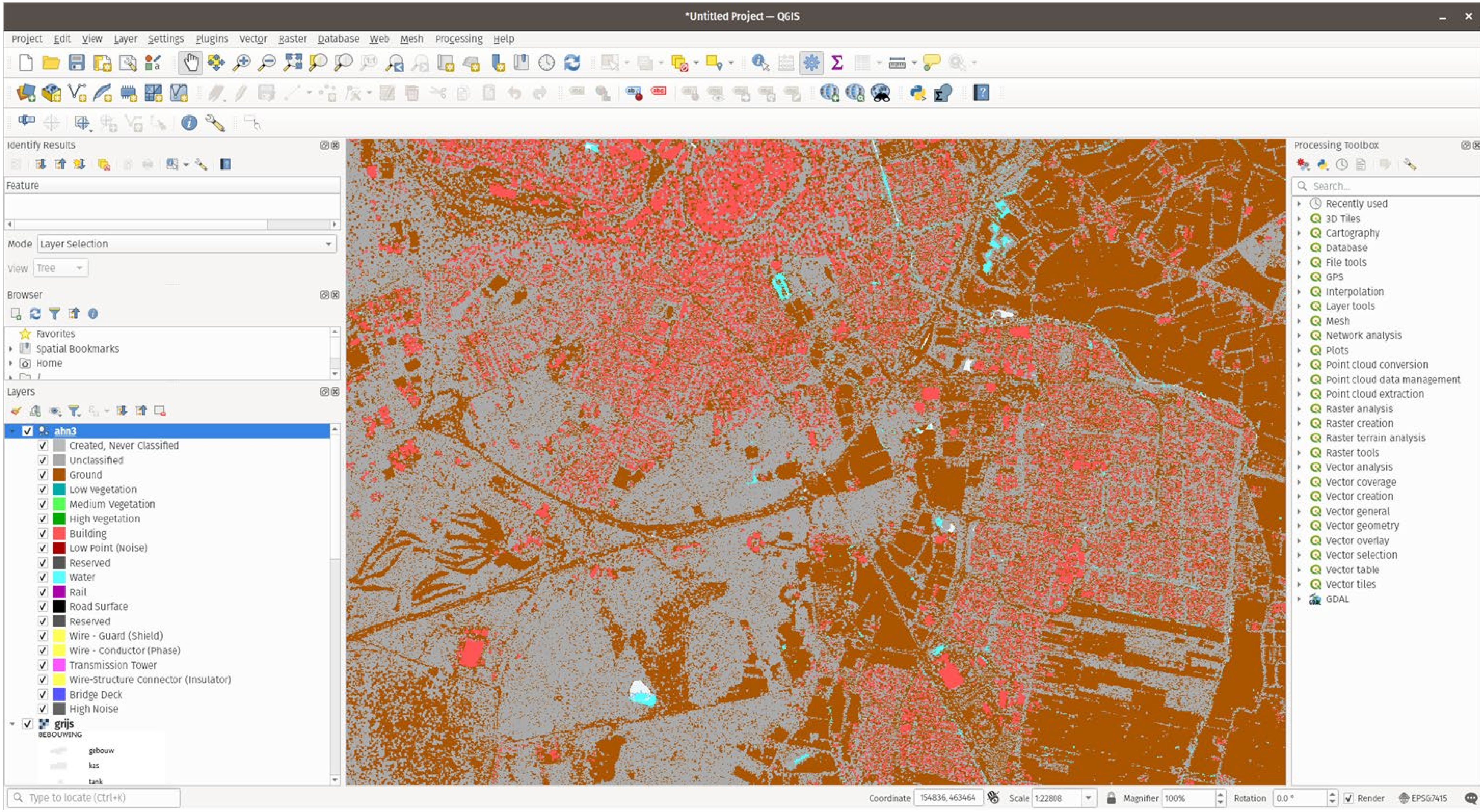
```
$ wget https://data.pointclouds.nl/data/nl/ahn-copc/3/ahn3.copc.vpc.json -O ahn3.vpc
--2024-10-10 14:03:00--  https://data.pointclouds.nl/data/nl/ahn-copc/3/ahn3.copc.vpc.json
Resolving data.pointclouds.nl (data.pointclouds.nl)... 131.180.126.49
Connecting to data.pointclouds.nl (data.pointclouds.nl)|131.180.126.49|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2428491 (2.3M) [application/json]
Saving to: ‘ahn3.vpc’


ahn3.vpc
100%[===========================================================================================
===========>]   2.32M  --.-KB/s    in 0.02s


2024-10-10 14:03:00 (109 MB/s) - ‘ahn3.vpc’ saved [2428491/2428491]
```

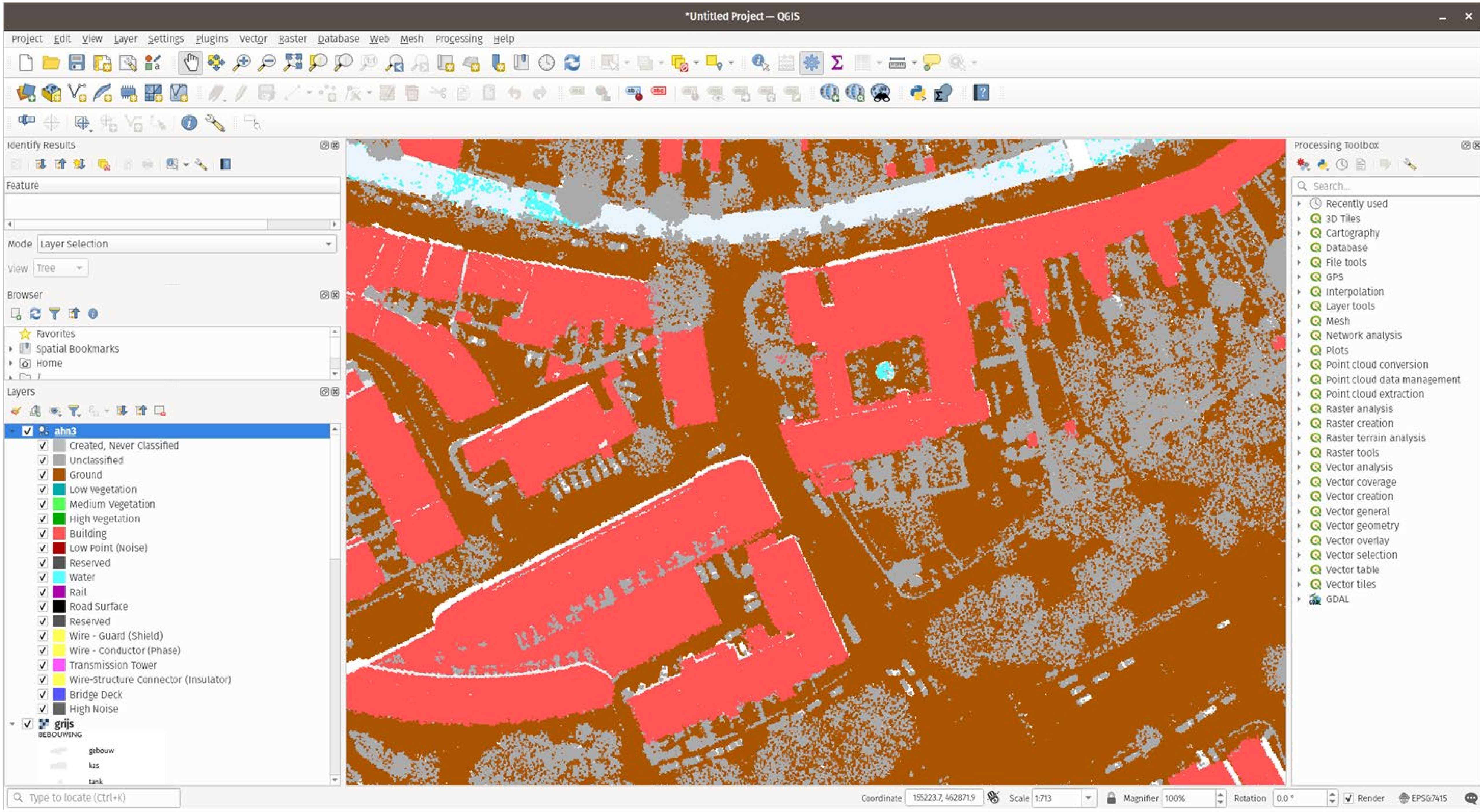
Open VPC in QGIS

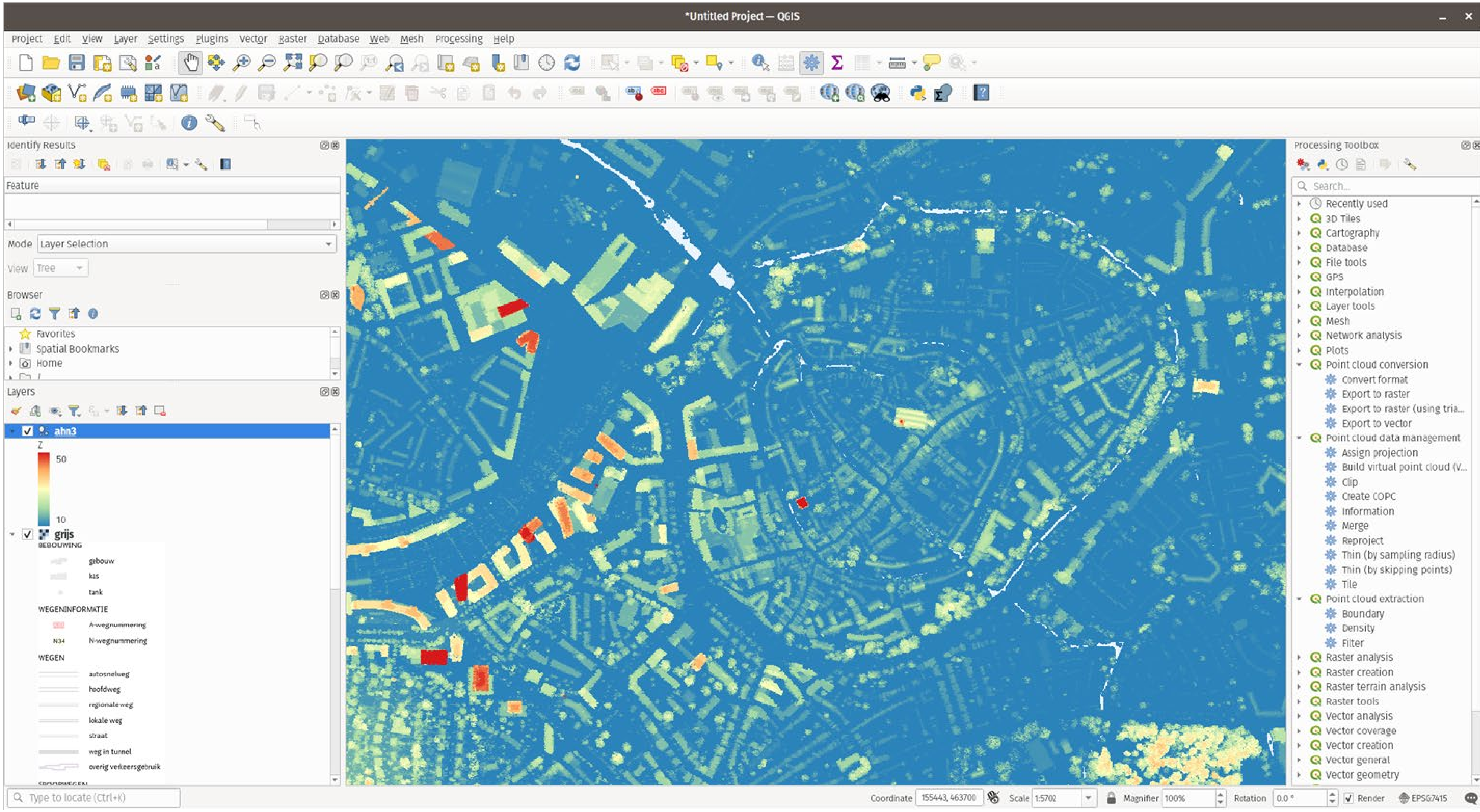


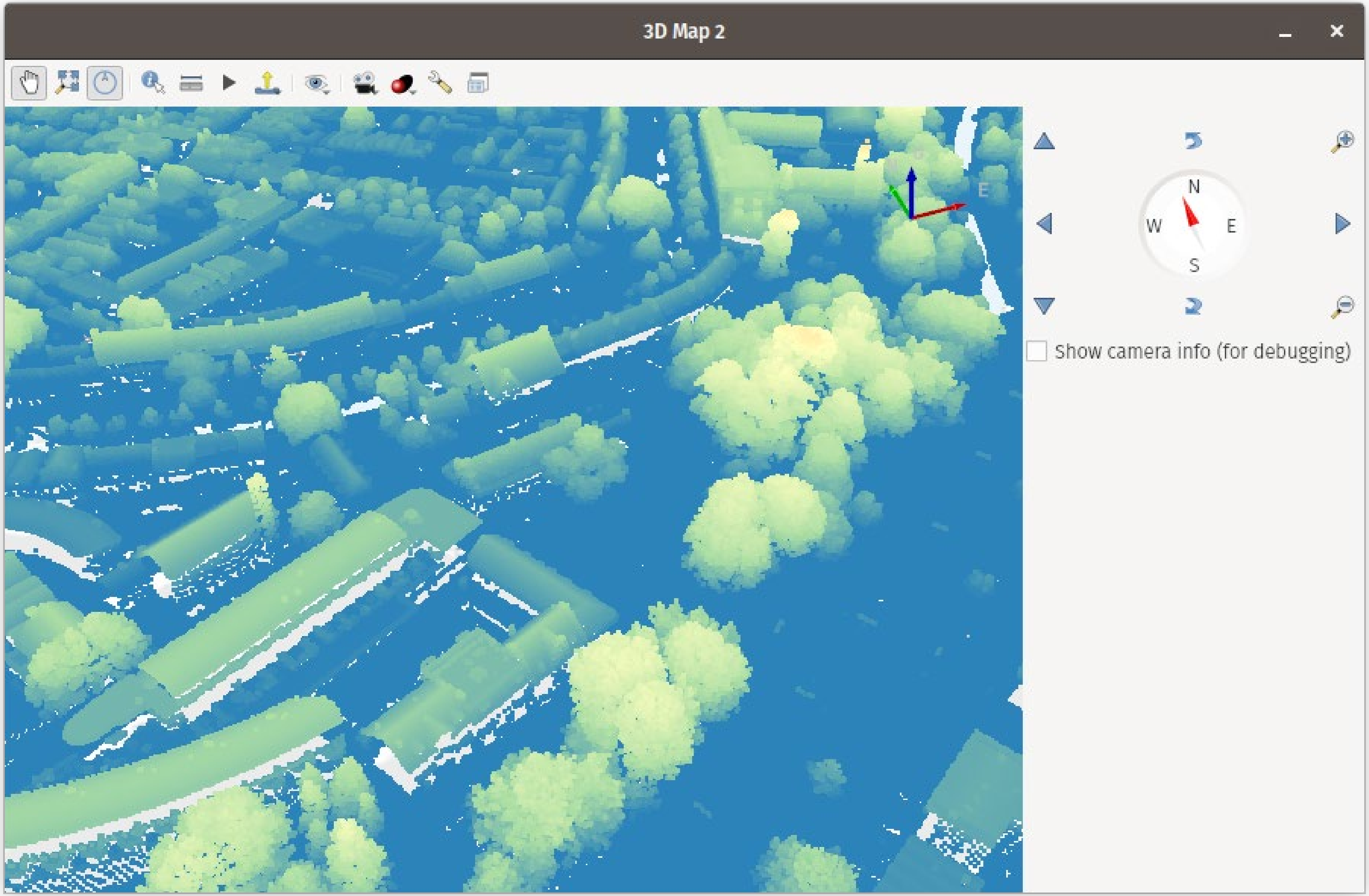

*Untitled Project — QGIS
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
Web
Mesh
Processing
Help
Identify Results
Feature
Mode
Layer Selection
View
Tree
Browser
Favorites
Spatial Bookmarks
Home
Layers
ahn3
Created, Never Classified
Unclassified
Ground
Low Vegetation
Medium Vegetation
High Vegetation
Building
Low Point (Noise)
Reserved
Water
Rail
Road Surface
Reserved
Wire - Guard (Shield)
Wire - Conductor (Phase)
Transmission Tower
Wire-Structure Connector (Insulator)
Bridge Deck
High Noise
grijs
BEBOUWING
gebouw
kas
tank
Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Point cloud data management
Point cloud extraction
Raster analysis
Raster creation
Raster terrain analysis
Raster tools
Vector analysis
Vector coverage
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
Vector tiles
GDAL
Type to locate (Ctrl+K)
Coordinate
25101, 619299
Scale
1:1459740
Magnifier
100%
Rotation
0.0 °
Render
EPSG:7415


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Identify Results
Feature
Mode Layer Selection
View Tree
Browser
Favorites
Spatial Bookmarks
Home
Layers
ahn3
Created, Never Classified
Unclassified
Ground
Low Vegetation
Medium Vegetation
High Vegetation
Building
Low Point (Noise)
Reserved
Water
Rail
Road Surface
Reserved
Wire - Guard (Shield)
Wire - Conductor (Phase)
Transmission Tower
Wire-Structure Connector (Insulator)
Bridge Deck
High Noise
grijs
BEBOUWING
gebouw
kas
tank
De Horst
't Vliet
Randweg
Princenhof
Valleipark
De Watering
Hamersveld
Asschatterweg
Rossenberg
Rozenboom
Noorderinslag
Leusden
Middenweg
Groene Zoom
Noorwijck
Hamerskop
Zwanenburg
Bosveld
Wildenburg
Hamersveldseweg
Paardenmaat
Tabaksteeg
Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Point cloud data management
Point cloud extraction
Raster analysis
Raster creation
Raster terrain analysis
Raster tools
Vector analysis
Vector coverage
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
Vector tiles
GDAL
Type to locate (Ctrl+K)
Coordinate 156326, 463404
Scale 1:22808
Magnifier 100%
Rotation 0.0 °
Render
EPSG:7415


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Identify Results
Feature
Mode Layer Selection
View Tree
Browser
Favorites
Spatial Bookmarks
Home
Layers
ahn3
Created, Never Classified
Unclassified
Ground
Low Vegetation
Medium Vegetation
High Vegetation
Building
Low Point (Noise)
Reserved
Water
Rail
Road Surface
Reserved
Wire - Guard (Shield)
Wire - Conductor (Phase)
Transmission Tower
Wire-Structure Connector (Insulator)
Bridge Deck
High Noise
grijs
BEBOUWING
gebouw
kas
tank
Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Point cloud data management
Point cloud extraction
Raster analysis
Raster creation
Raster terrain analysis
Raster tools
Vector analysis
Vector coverage
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
Vector tiles
GDAL
Type to locate (Ctrl+K)
Coordinate 154836, 463464
Scale 1:22808
Magnifier 100%
Rotation 0.0 °
Render
EPSG:7415


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Identify Results
Feature
Mode Layer Selection
View Tree
Browser
Favorites
Spatial Bookmarks
Home
Layers
ahn3
Created, Never Classified
Unclassified
Ground
Low Vegetation
Medium Vegetation
High Vegetation
Building
Low Point (Noise)
Reserved
Water
Rail
Road Surface
Reserved
Wire - Guard (Shield)
Wire - Conductor (Phase)
Transmission Tower
Wire-Structure Connector (Insulator)
Bridge Deck
High Noise
grijs
BEBOUWING
gebouw
kas
tank
Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Point cloud data management
Point cloud extraction
Raster analysis
Raster creation
Raster terrain analysis
Raster tools
Vector analysis
Vector coverage
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
Vector tiles
GDAL
Type to locate (Ctrl+K)
Coordinate 155223.7, 462871.9
Scale 1:713
Magnifier 100%
Rotation 0.0 °
Render
EPSG:7415


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Identify Results
Feature
Mode Layer Selection
View Tree
Browser
Favorites
Spatial Bookmarks
Home
Layers
ahn3
Z
50
10
grijs
BEBOUWING
gebouw
kas
tank
WEGENINFORMATIE
A-wegnummering
N-wegnummering
WEGEN
autosnelweg
hoofdweg
regionale weg
lokale weg
straat
weg in tunnel
overig verkeersgebruik
Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Convert format
Export to raster
Export to raster (using tria...
Export to vector
Point cloud data management
Assign projection
Build virtual point cloud (V...
Clip
Create COPC
Information
Merge
Reproject
Thin (by sampling radius)
Thin (by skipping points)
Tile
Point cloud extraction
Boundary
Density
Filter
Raster analysis
Raster creation
Raster terrain analysis
Raster tools
Vector analysis
Vector coverage
Vector creation
Vector general
Vector geometry
Type to locate (Ctrl+K)
Coordinate 155443, 463700
Scale 1:5702
Magnifier 100%
Rotation 0.0 °
Render
EPSG:7415


3D Map 2
N
W
E
S
Show camera info (for debugging)

Statistics for AHN to COPC and VPC (on ronna)

	#files in	input	processing	#files out	output
AHN3	1374	2.4Tb	14 days 5 hrs	382	3.1Tb
AHN4	1381	6.1TB	26 days *)	381	5.1Tb

*) estimation (as process was interrupted, total 54 days)

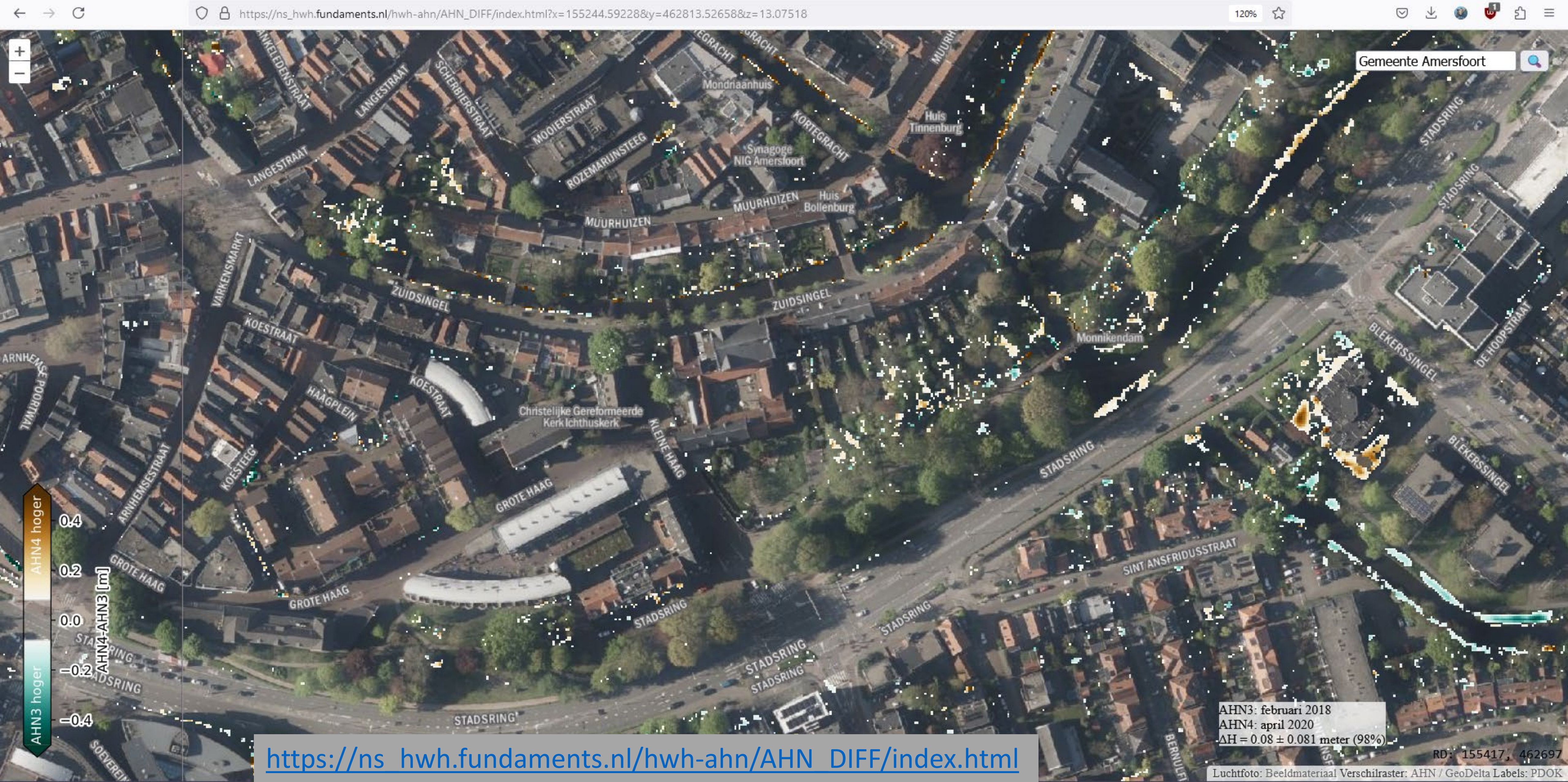
processing time linear with input size

ronna is multi-user system, so timings not exact

for AHN3: COPC > LAZ input, for AHN4: COPC < LAZ input

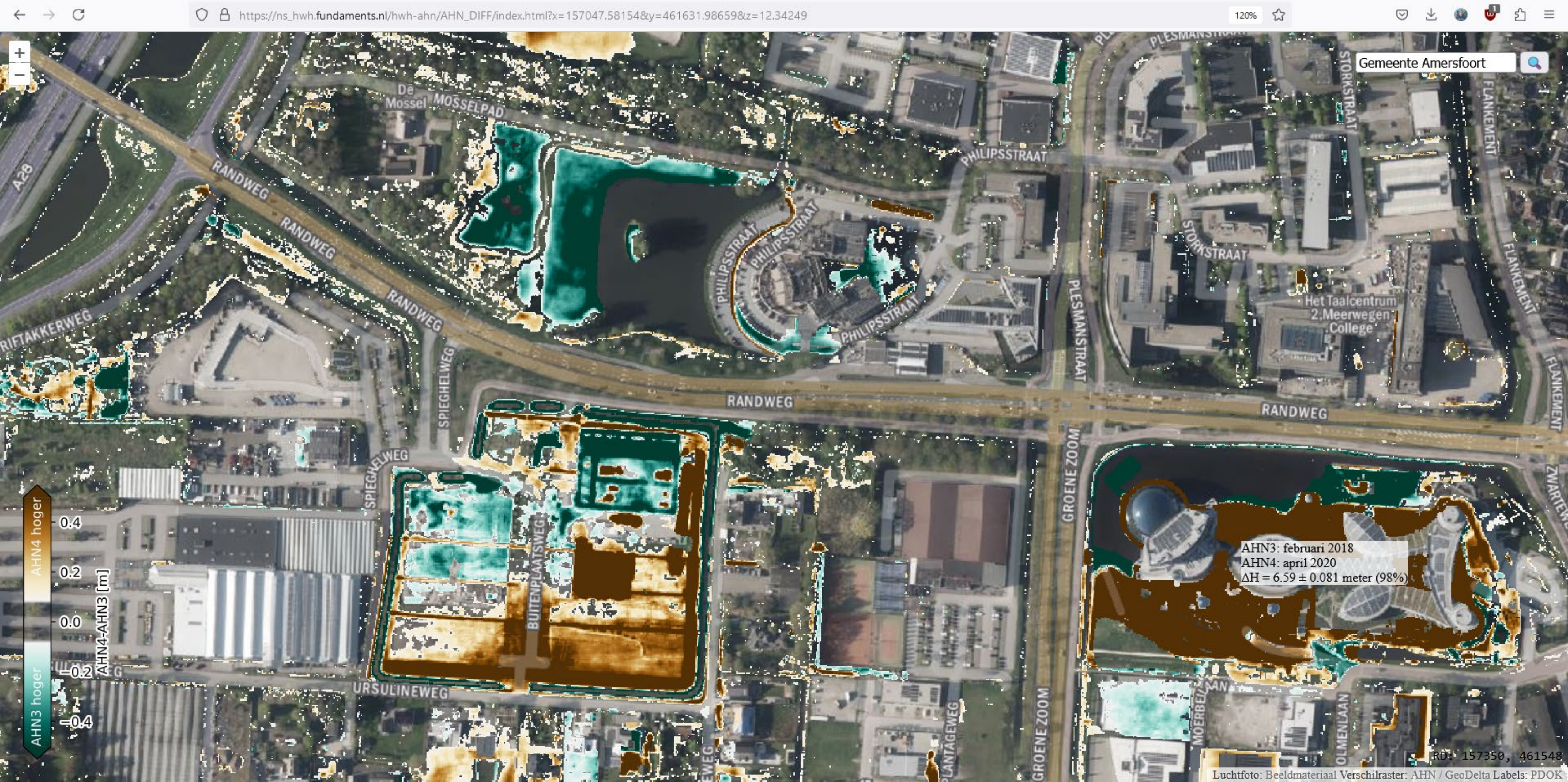
GitHub to source https://github.com/nd-pc/process_ahn

Raster based change detection



https://ns_hwh.fundaments.nl/hwh-ahn/AHN_DIFF/index.html

Nicer example AFAS HQ, Leusden



Fine, but…

- grid is less detailed than PC
- grid can only be displayed in 2D
- grid needs to be computed first

- much better to have direct point cloud change detection

Comparison of Cloud-to-Cloud Distance Calculation Methods - Is the Most Complex Always the Most Suitable?

Vitali Diaz, Peter van Oosterom, Martijn Meijers, Edward Verbree, Nauman Ahmed, and Thijs van Lankveld

Abstract Cloud-to-cloud (C2C) distance calculations are frequently performed as an initial stage in change detection and spatiotemporal analysis with point clouds. There are various methods for calculating C2C distance, also called inter-point distance, which refers to the distance between two corresponding point clouds captured at different epochs. These methods can be classified from simple to complex, with more steps and calculations required for the latter. Generally, it is assumed that a more complex method will result in a more precise calculation of inter-point distance, but this assumption is rarely evaluated. This paper compares eight commonly used methods for calculating the inter-point distance. The results indicate that the accuracy of distance calculations depends on the chosen method and a characteristic related to the point density, the intra-point distance, which refers to the distance between points within the same point cloud. The results are helpful for applications that analyze spatiotemporal point clouds for change detection. The findings will be helpful in future applications, including analyzing spatiotemporal point clouds for change detection.

Keywords Cloud-to-cloud distance calculation · Change detection · Spatiotemporal analysis

This article was selected based on the results of a double-blind review of an extended abstract

V. Diaz (✉) · P. van Oosterom · M. Meijers · E. Verbree
GIS Technology, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft 2628 BL, The Netherlands
e-mail: v.diazmercado@tudelft.nl; vitalidime@gmail.com

P. van Oosterom
e-mail: P.J.M.vanOosterom@tudelft.nl

N. Ahmed · T. van Lankveld
Netherlands eScience Center, Amsterdam 1098 XH, The Netherlands

T. H. Kolbe et al. (eds.), *Recent Advances in 3D Geoinformation Science*, Lecture Notes in Geoinformation and Cartography, https://doi.org/10.1007/978-3-031-43699-4_20

Direct PC-based change detection (not convert to grids and show changes in 3D)

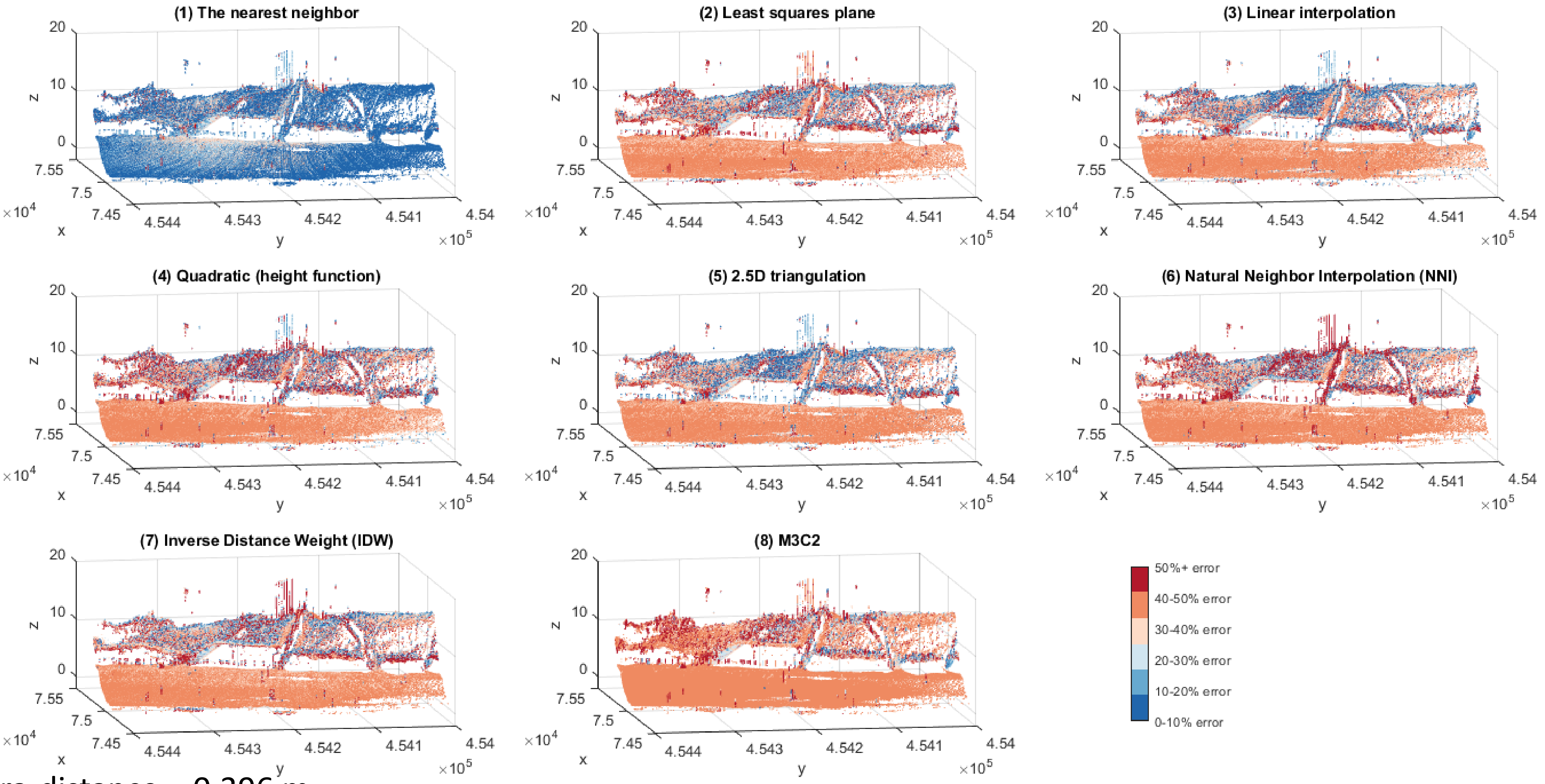
- tests to find the suitable PC-based method: controlled offsets and real epochs → comparison based on error and time cost
- range of algorithms (as available in CloudCompared)
- reimplemented in Matlab for full control
- testing with various data sets (AHN, CostScan, lake, bunny)
- conclusion from results: 3D Nearest Neighbour(NN) outperforms/equal quality to more complex algorithms regarding similar results and less time cost. Relatively efficient for implementation, 'real' NN kd-tree based.

- Then, further speed-up will be challenged for massive implementation for further change detection at nationwide applications over the AHN series.

Experiment 1: CoastScan data with controlled offset

comparing the 8 different algorithm

dx=0.2
dy=0.2
dz=0.2



Average intra-distance = 0.396 m

Experiment 2: Preliminary Results (2D NN)

Artificial displacement

dx=-0.25

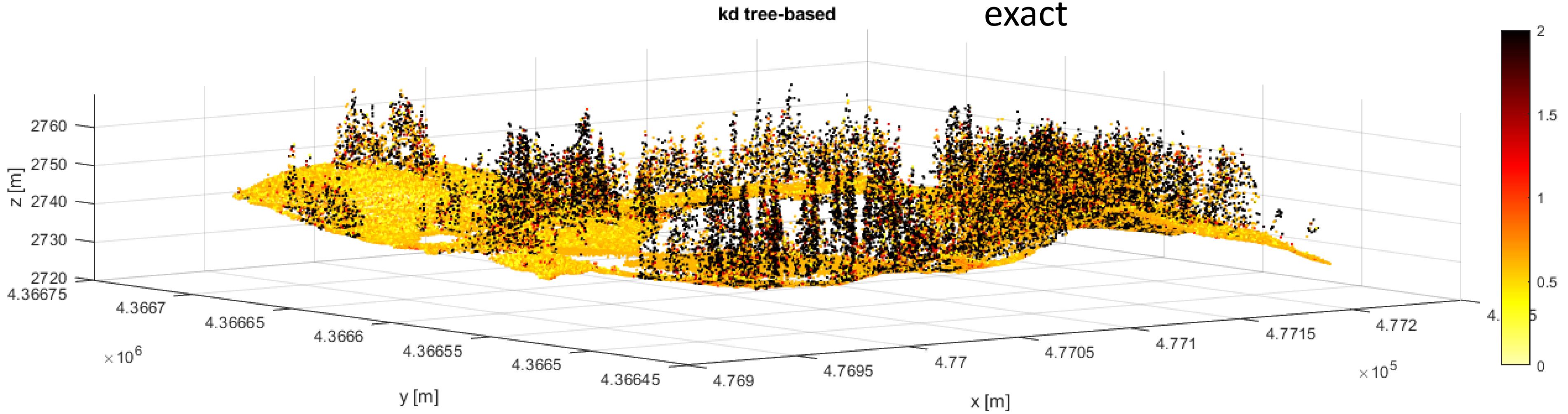
dy=-0.25

dz=-0.5

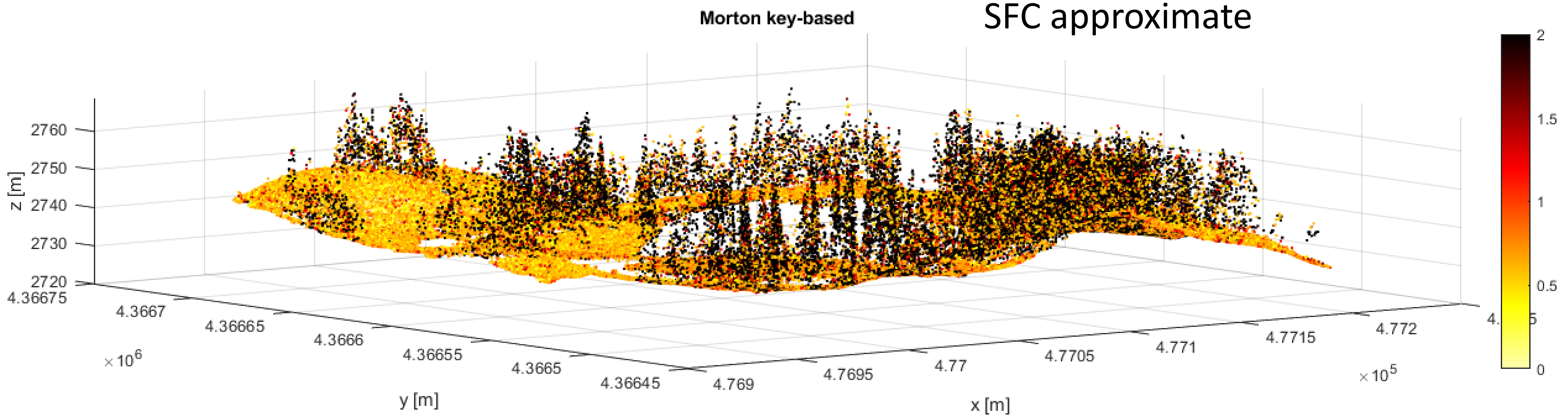
d=0.6124

(lesson learnt: do not do 2D, but 3D NN)

exact

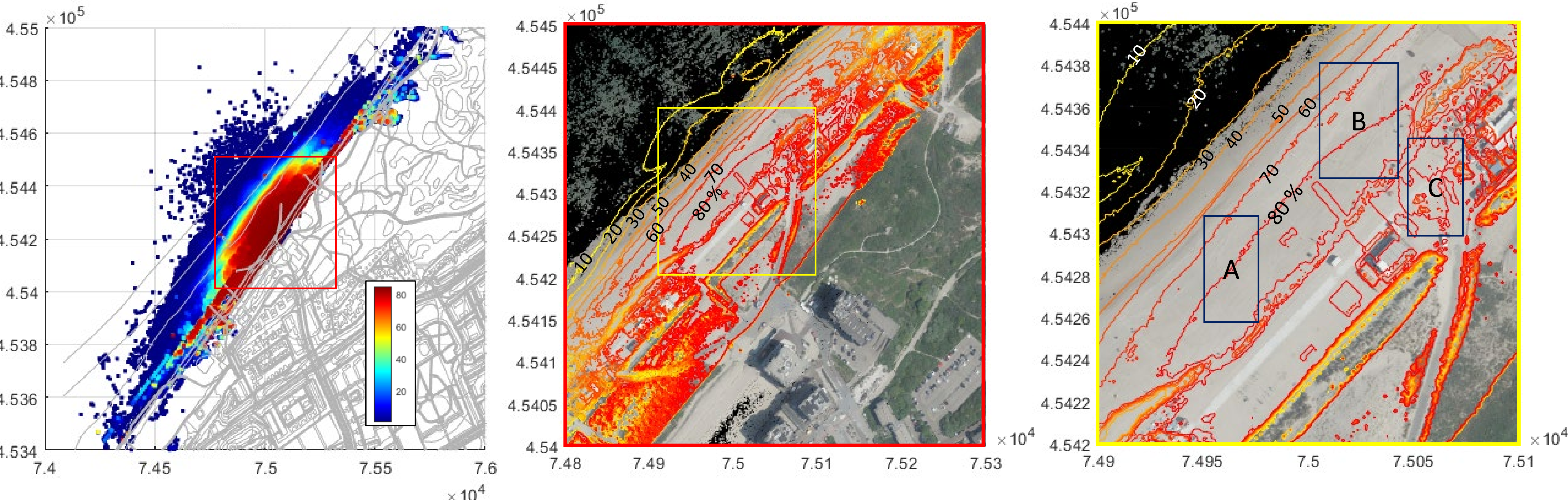


SFC approximate

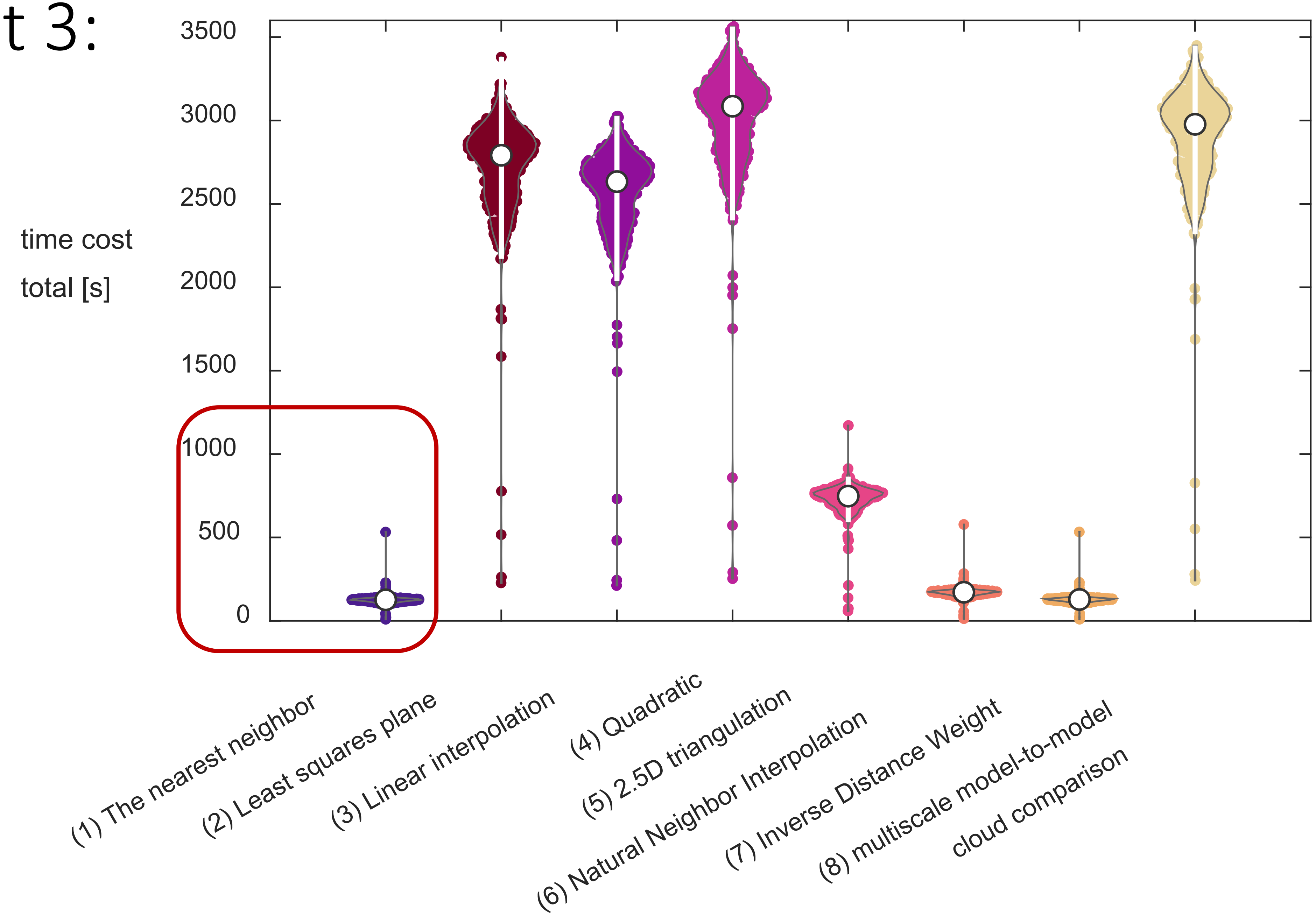


Experiment 3: spatio-temporal CoastScan data

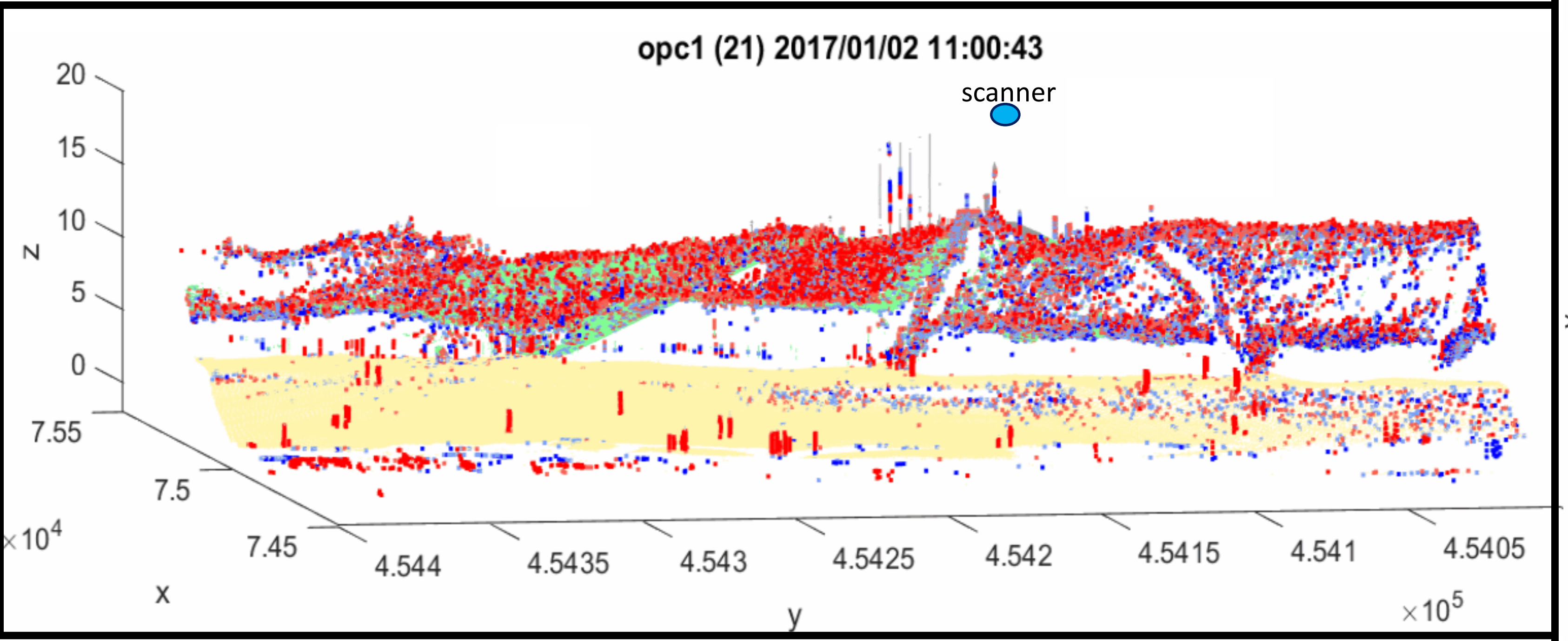
period 2017/01/01 to 2017/04/18
2,187 epochs

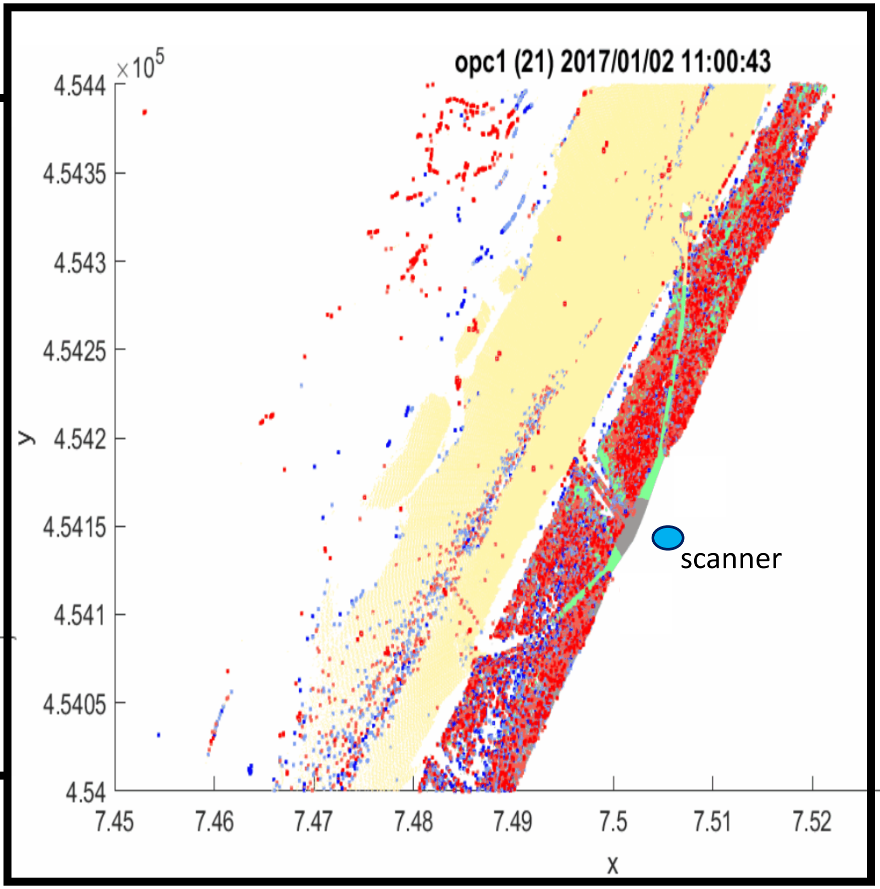


Experiment 3: Time cost



Experiment 3: Spatio-temporal CoastScan data





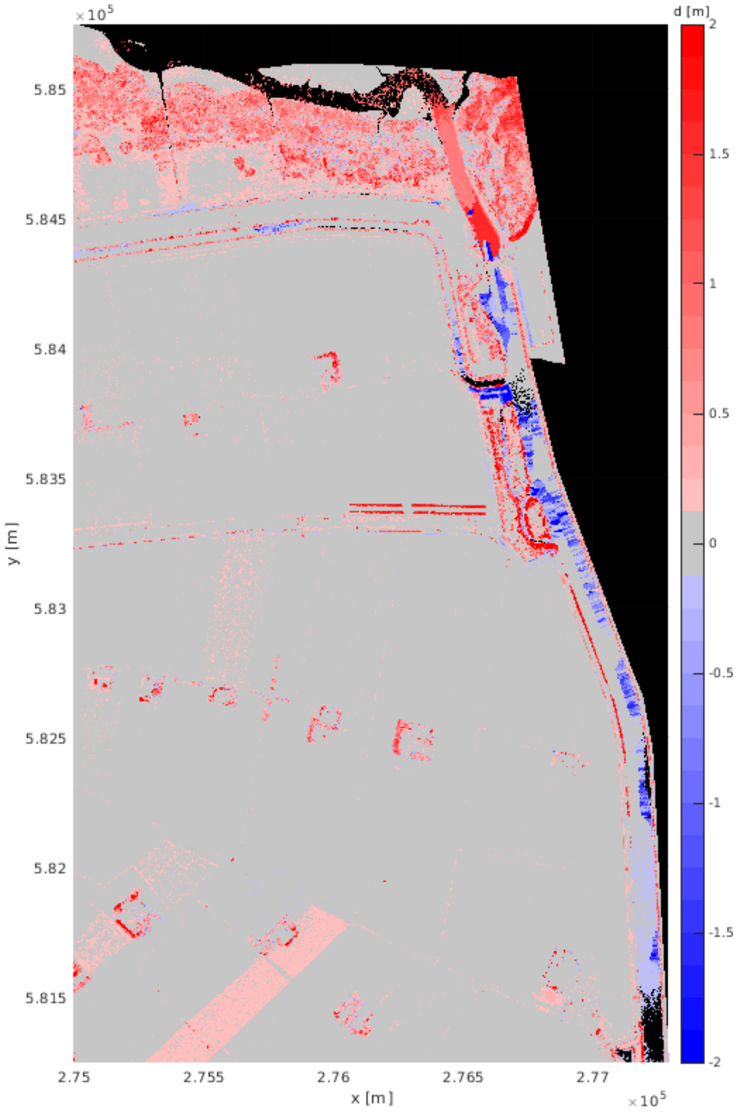
3D Nearest Neighbour(NN):
changes are detected efficiently and correctly

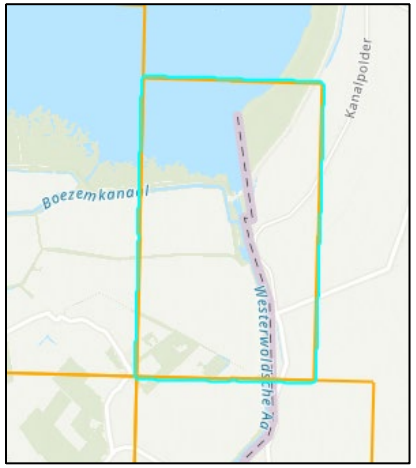
SFC-based ultra-fast change detection

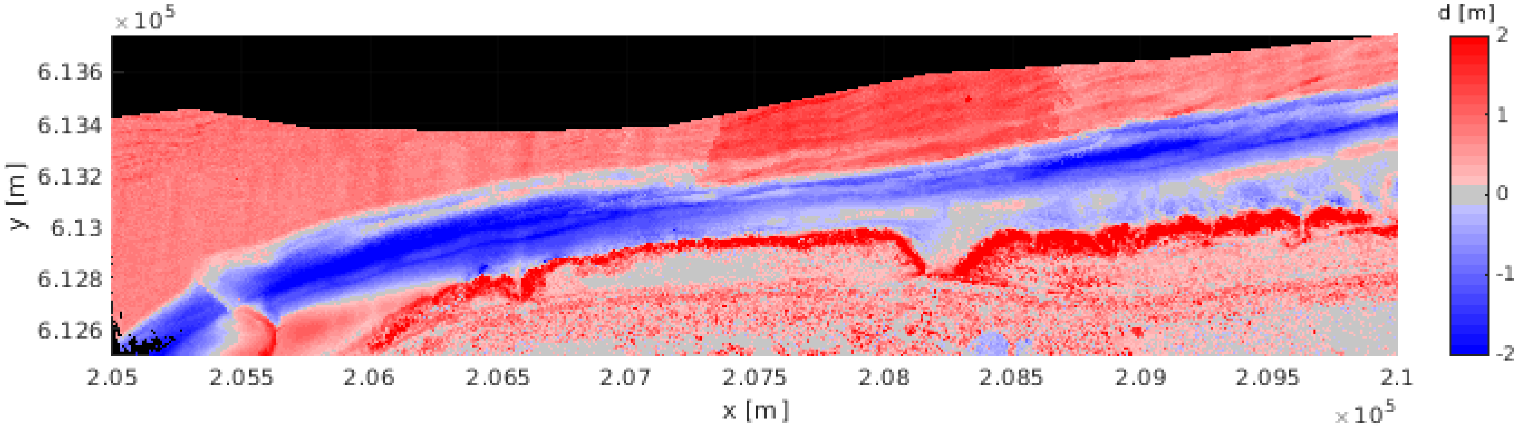
- per epoch, preparation of PC data set
  1. scaling and offsetting (SFC always in a cube)
  2. SFC key calculation
  3. sorting SFC key column

- batch process NN distance calculation (advanced merge join 2 epochs): for every point P in epoch 2:
  1. search the previous and next SFC key of the calculated SFC key in epoch 1
  2. decode and unscale these two SFC keys (previous and next)
  3. find approximate nearest neighbor (NN) of the 2 candidates
     (option: box query in epoch 1 with P as center and distance P-NN as radius)
  4. output d and dx,dy,dz

Preliminary results: AHN4-3





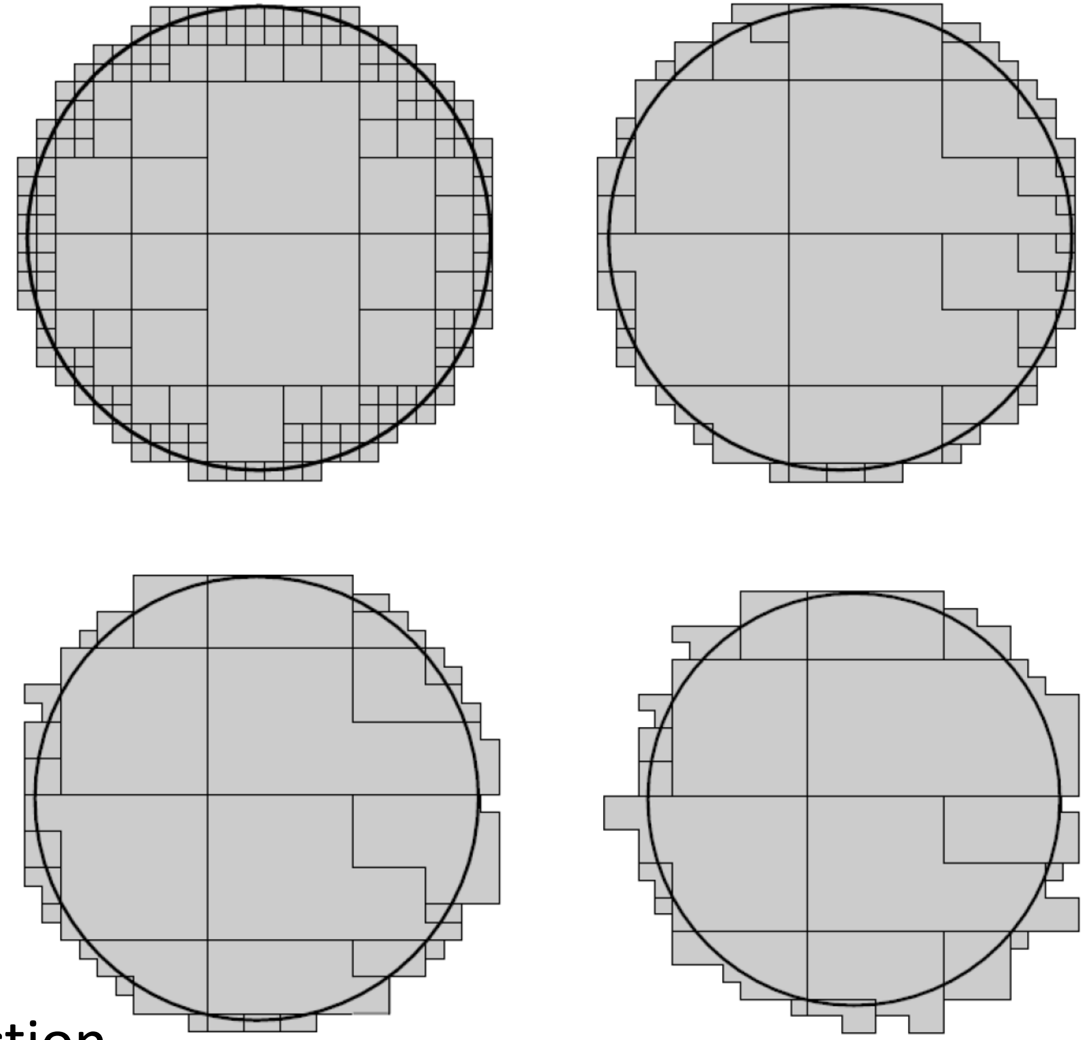


Agenda

- nD-PointCloud
  - foundations
  - PostgreSQL implementation
  - nD Convex polytope query
  - Apache Parquet

- practical results
  - Potree conversions AHN 1 to 5
  - COPC/VPC conversions
  - fast direct point cloud-based change detection

- **future work**
  - **main publications**



Conclusion

- designed and tested nD-PC organization
  (with cLoI) in resp. DBMS and files, and good results obtained
- explored and cleaned spatio-temporal point cloud data, and analyzed in detail the possible change detection options → NN preferred
- current practice tools not yet cLoI-aware and space first (not time)
- active at various meetings (OGC, 3D GeoInfo, FOSS4G, EGU) and publications
- Geomatics/GIMA students doing (thesis) projects with point clouds
- **open invitation to the nD-PointCloud project concluding symposium:**
  12 November 2024, 13-16 hours
  room B, Faculty of Architecture and the Built Environment, TU Delft

Future work

- completion of publication in pipeline
- standardization of format and protocols (binary Parquet files)
- cLoI aware viewers and other clients (computations),
  using selection with flat hyperplanes (convex polytope)
- on-the-fly CRS transformations of selections, or pre-computing of whole data sets (on HPC/HTC)
- explore cLoI to integrate datasets from different scales (after geo-referencing/CRS transformation)
- nationwide AHN change detection (buildings, vegetation, ..) by
  - adding 1 just or all 3 directions
  - store result integrated (dimension of attribute) or separate from points
  - interactive of massive preprocessing (on HPC/HTC), but fast use
  - backward of forward changes
  - for CoastScan, option to skip epochs (hours, days, weeks, months)

Publications (1/2)

- Vitali Diaz, Peter van Oosterom, Martijn Meijers, Edward Verbree, Nauman Ahmed, Thijs van Lankveld, Comparison of Cloud-to-Cloud Distance Calculation Methods - Is the Most Complex Always the Most Suitable?, Chapter in: Recent Advances in 3D Geoinformation Science, Lecture Notes in Geoinformation and Cartography, Springer Nature Switzerland, pp. 229-334, 2024.
- Algan Yasar, Robert Voûte, Edward Verbree, Direct Use of Indoor Point Clouds for Path Planning and Navigation Exploration in Emergency Situations, Chapter in: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-4/W11-2024, pp. 175-181, 2024.
- Zhenyu Liu, Peter van Oosterom, Jesús Balado, Arjen Swart, Bart Beers, Data frame aware optimized Octomap-based dynamic object detection and removal in Mobile Laser Scanning data, In: Alexandria Engineering Journal, 74, pp. 327-344, 2023.
- Vitali Diaz, Peter van Oosterom, Martijn Meijers, Edward Verbree, Nauman Ahmed, Thijs van Lankveld, Comparison of point distance calculation methods in point clouds - Is the most complex always the most suitable?, In: Proceedings of the 18th International 3DGeoInfo Conference 2023, Munich, Germany, pp. 329-334, 2023.
- Ioannis Dardavesis, Edward Verbree, Azarakhsh Rafiee, Indoor localisation and location tracking in indoor facilities based on LiDAR point clouds and images of the ceilings, In: Proceedings of the 26th AGILE Conference on Geographic Information Science, 2023, GIScience Series, 4(4), Delft, The Netherlands, pp. 1-15, 2023
- Vidushi Bhatt, Sharath Chandra Madanu, Shen Qiwei, Susanne Epema, Gees Brouwer, Pointcloud based anatomy, MSc Geomatics synthesis project, Technical report, Delft University of Technology, pp. 55, 2023.
- **Haicheng Liu, nD-PointCloud Data Management - continuous levels, adaptive histograms, and diverse query geometries, PhD thesis, Delft University of Technology, pp. 207, 2022.**

Publications (2/2)

- Peter van Oosterom, Simon van Oosterom, Haicheng Liu, Rod Thompson, Martijn Meijers, Edward Verbree, Organizing and visualizing point clouds with continuous levels of detail, In: ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier BV, 194, pp. 119-–131, 2022.
- Martijn Meijers, PCServe – nD-PointClouds Retrieval over the Web, In: ISPRS - Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 17th 3D GeoInfo Conference, Sydney, Australia, Copernicus GmbH, X-4/W2-2022, pp. 193-200, 2022.
- Vitali Diaz, Haicheng Liu, Peter van Oosterom, Martijn Meijers, Edward Verbree, Fedor Baart, Maarten Pronk, Thijs van Lankveld, Point clouds and Hydroinformatics, 2022 (Abstract from EGU General Assembly 2022, Vienna, Austria, 23–27 May 2022).
- Zhenyu Liu, Peter van Oosterom, Jesús Balado, Arjen Swart, Bart Beers, Detection and reconstruction of static vehicle-related ground occlusions in point clouds from mobile laser scanning, In: Automation in Construction, Elsevier BV, 141, pp. 104461, 2022.
- Haicheng Liu, Rodney Thompson, Peter van Oosterom, Martijn Meijers, Executing convex polytope queries on nD point clouds, In: International Journal of Applied Earth Observations and Geoinformation, Elsevier, 105(102625), pp. 1-11, 2021.
- Guan-Ting Zhang, Edward Verbree, Xiao-Jun Wang, An Approach to Map Visibility in the Built Environment From Airborne LiDAR Point Clouds, In: IEEE Access, Institute of Electrical and Electronics Engineers (IEEE), 9, pp. 44150-44161, 2021.
- H. Liu, P. Van Oosterom, B. Mao, M. Meijers, R. Thompson, An efficient nD-Point Data Structure for Querying Flood Risks, Chapter in: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Copernicus GmbH, XLIII-B4-2021, pp. 367-374, 2021.

Thanks for your attention!