Visualization, storage, analysis and distribution of massive aerial LiDAR point clouds

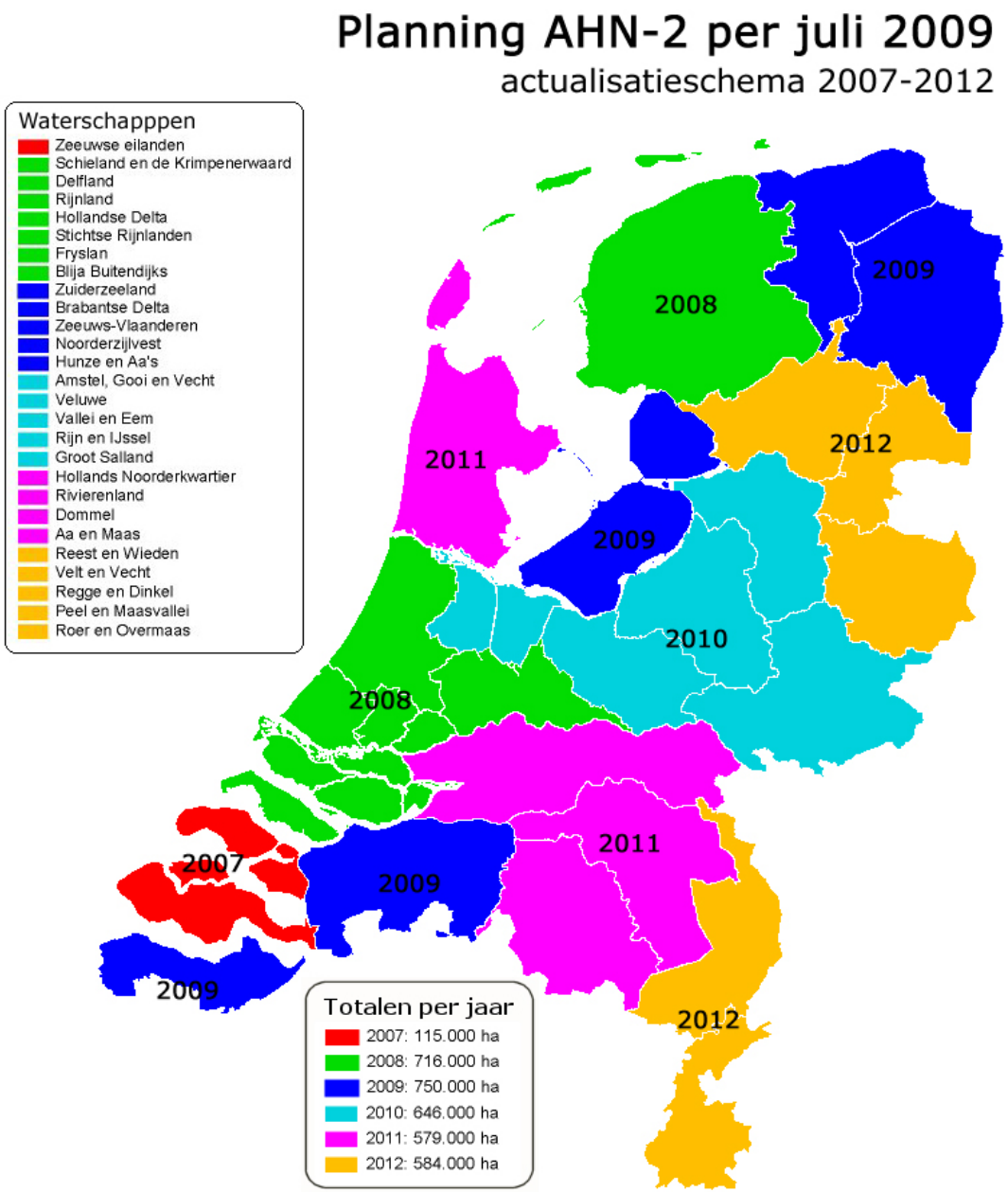
Gerwin de Haan and **Hugo Ledoux**

Technische Universiteit Delft

Data Visualization group
GIS technology group

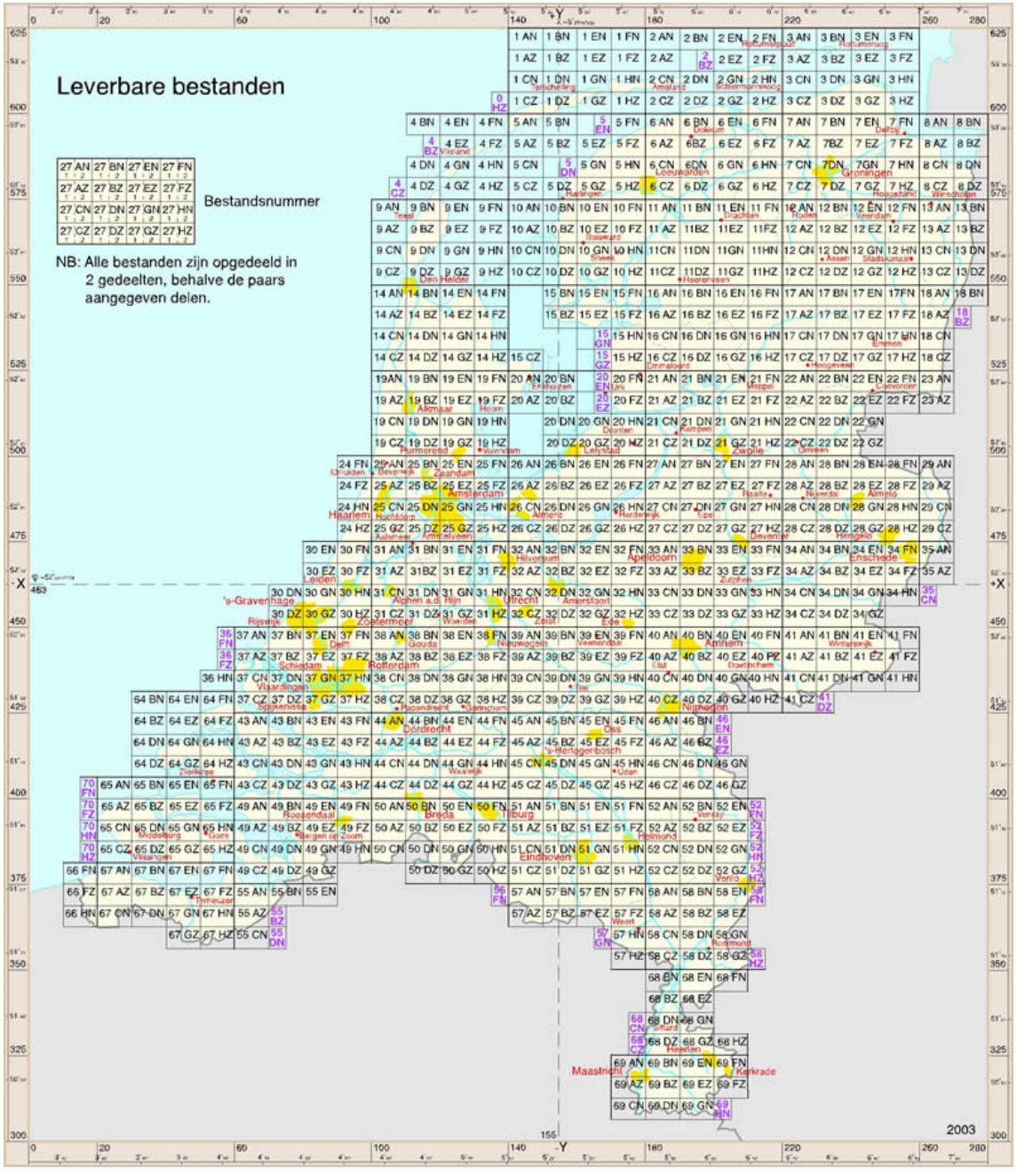
GeoWeb 2010, Vancouver
July 28 2010

AHN^2: A dataset covering totally the Netherlands

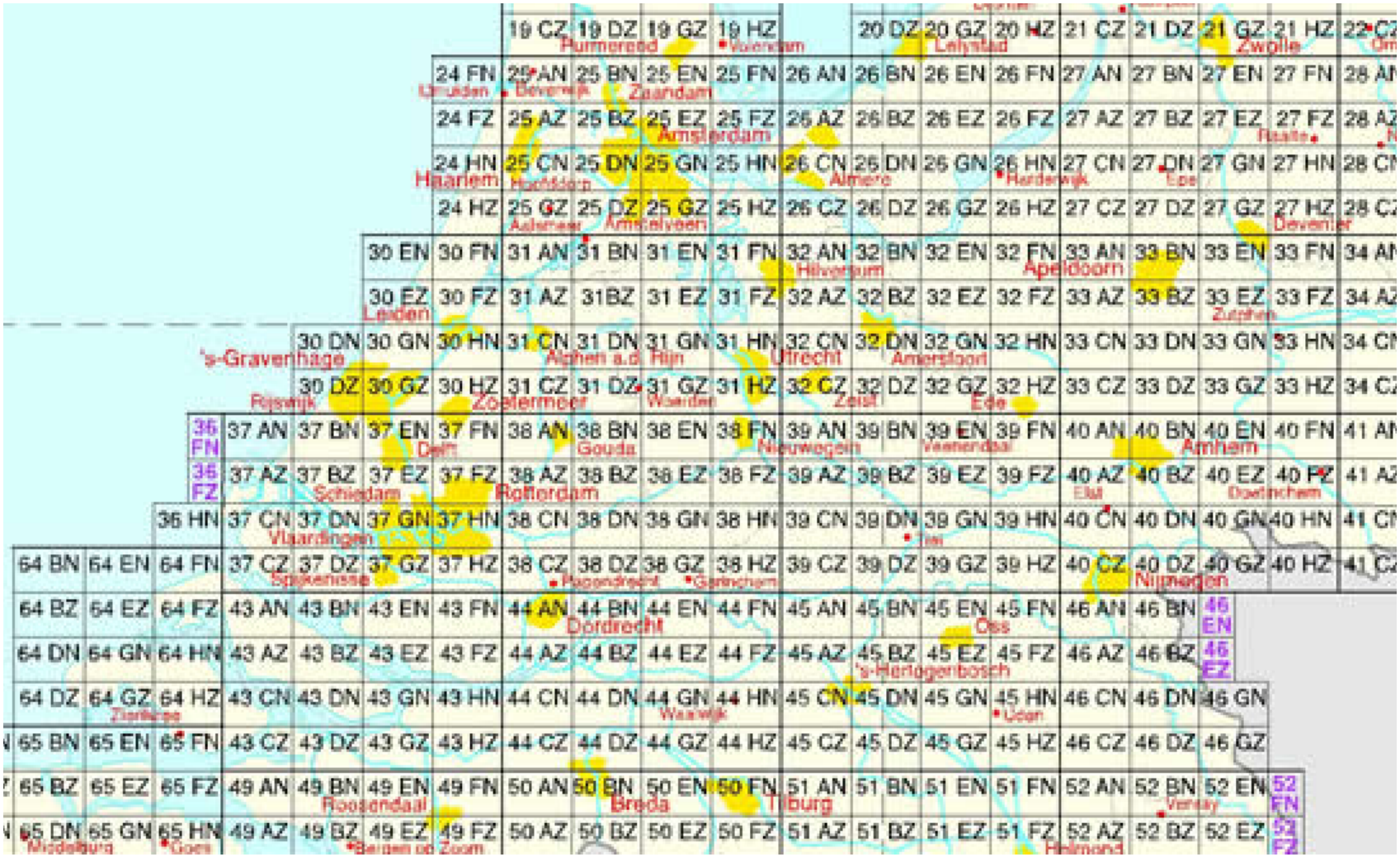


- compiled by the government
- (x, y, z) points
- raster (50cm grid available)
- at least 4 pts/m^2
- $>$ 150 billions points

AHN^2: A tiled dataset



AHN^2: A tiled dataset



File “g37en1_08.xyz”

```
82993.039,447517.568,-0.003
82993.334,447517.102,-0.042
82993.633,447516.614,-0.044
82992.494,447517.148,-0.005
82992.788,447516.682,-0.044
82993.689,447515.201,-0.022
82993.987,447514.719,-0.033
82994.282,447514.245,-0.052
82994.576,447513.770,-0.070
82994.884,447513.247,-0.012
...
...
...
```

Our challenges when dealing with massive point clouds

1. Real-time visualisation
2. Storage + spatial analysis
3. Distribution of such datasets

Can our tools/workflows scale to **massive** datasets?

Real-time Visualisation

Why visualise point clouds?

- Because it's a nice challenge!
- Because we need to:
  - Perform visual inspection of raw measurements
  - Gain insights from numbers

Rendering $\longrightarrow$ Visualisation $\longrightarrow$ Interaction

From DEM terrain...

(AHN^2 DEM 0.5m grid)

... to point cloud

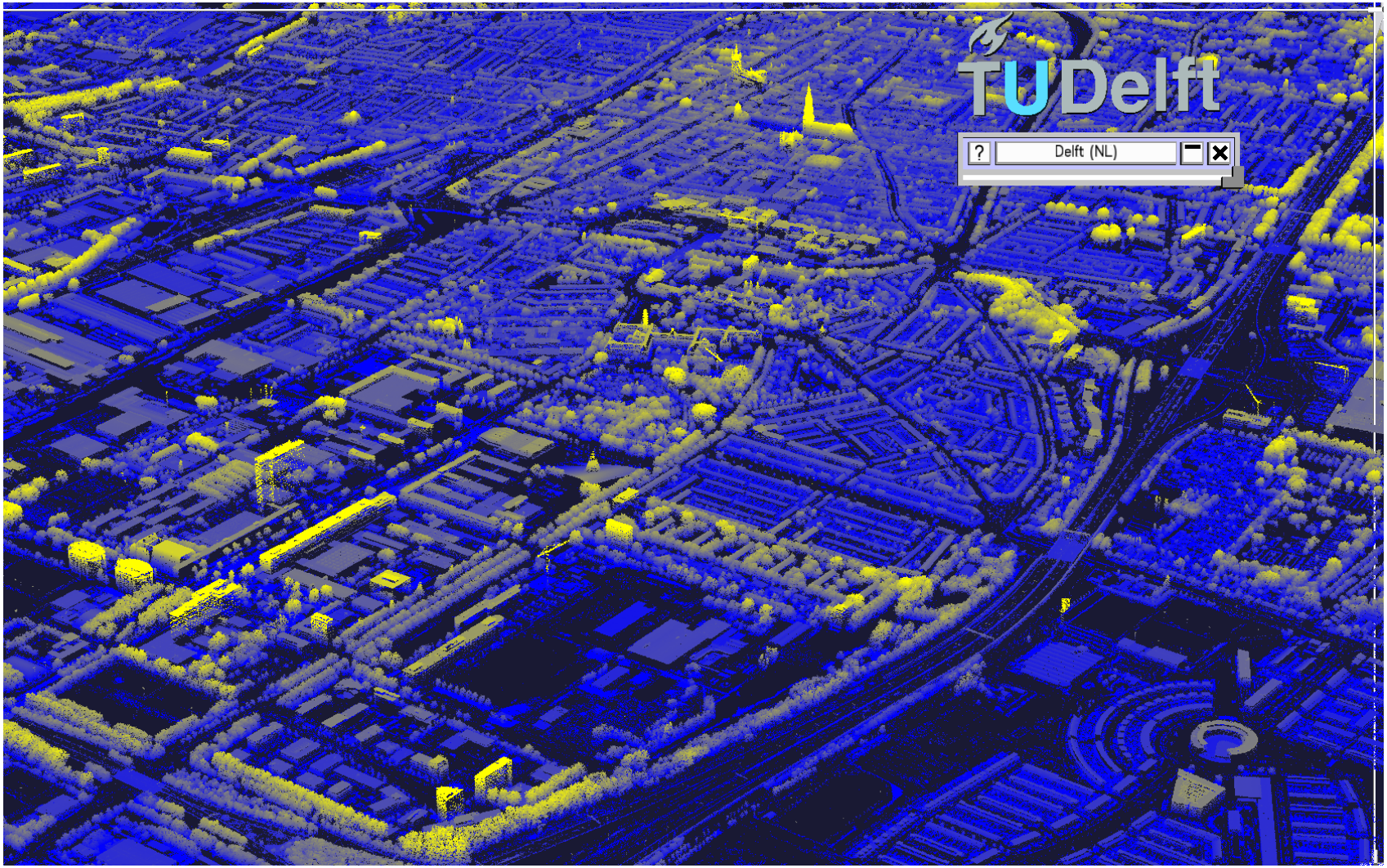
($\sim$ 2 millions points from AHN^2)

Our tools of choice

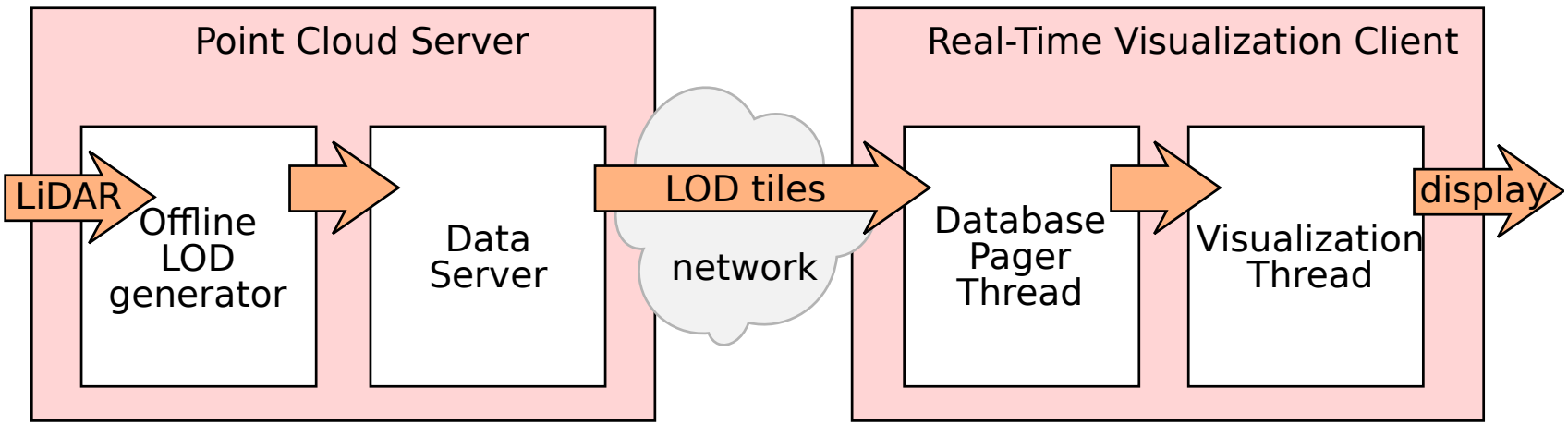
- OpenSceneGraph
  - most popular FOSS scene graph engine
  - based on flight-simulator software (not GIS)
- OpenSceneGraph-based VRMeer software
- Python abstraction layers

  `http://code.google.com/p/osgswig`

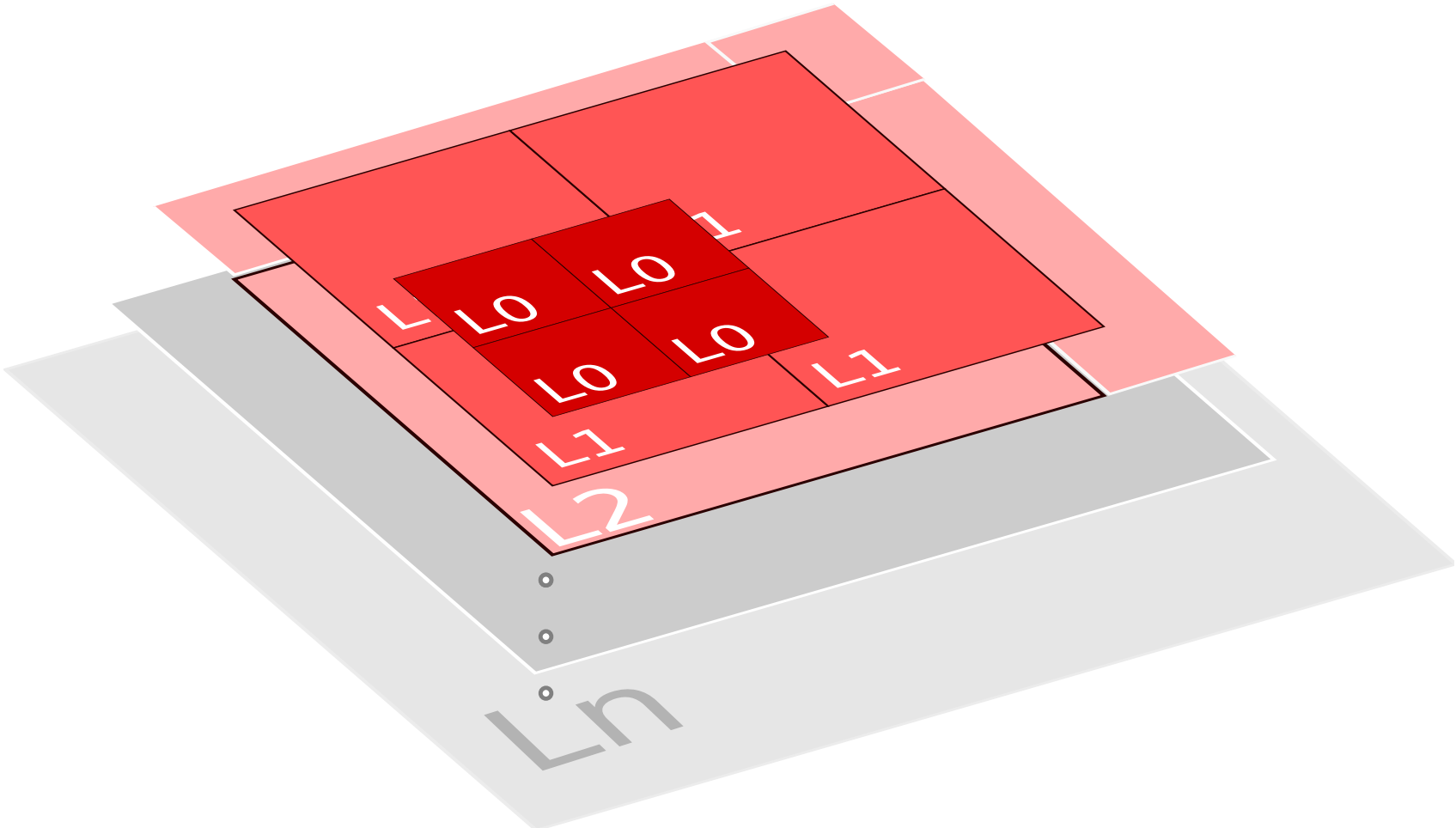
Demonstration video



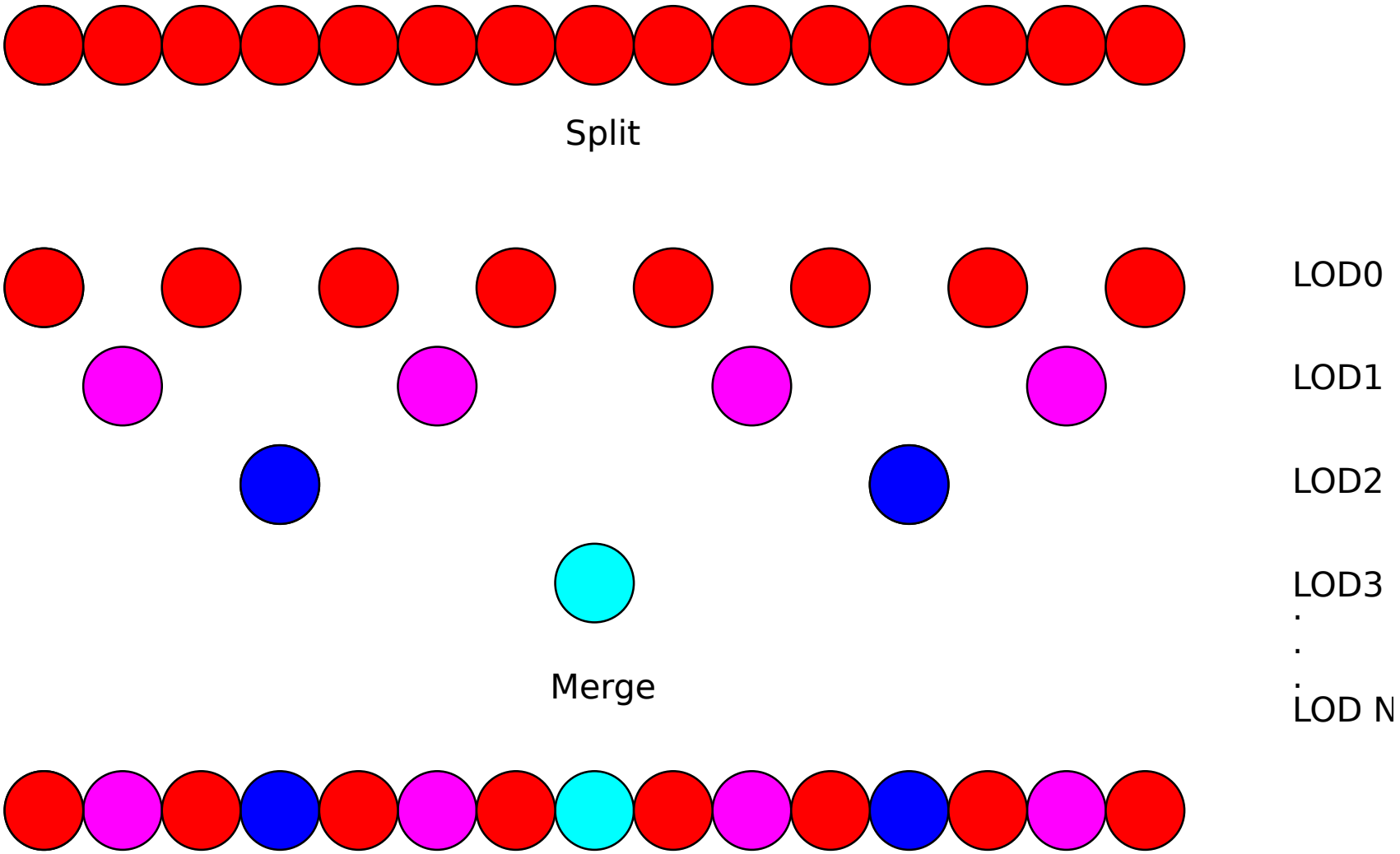
Visualisation pipeline overview



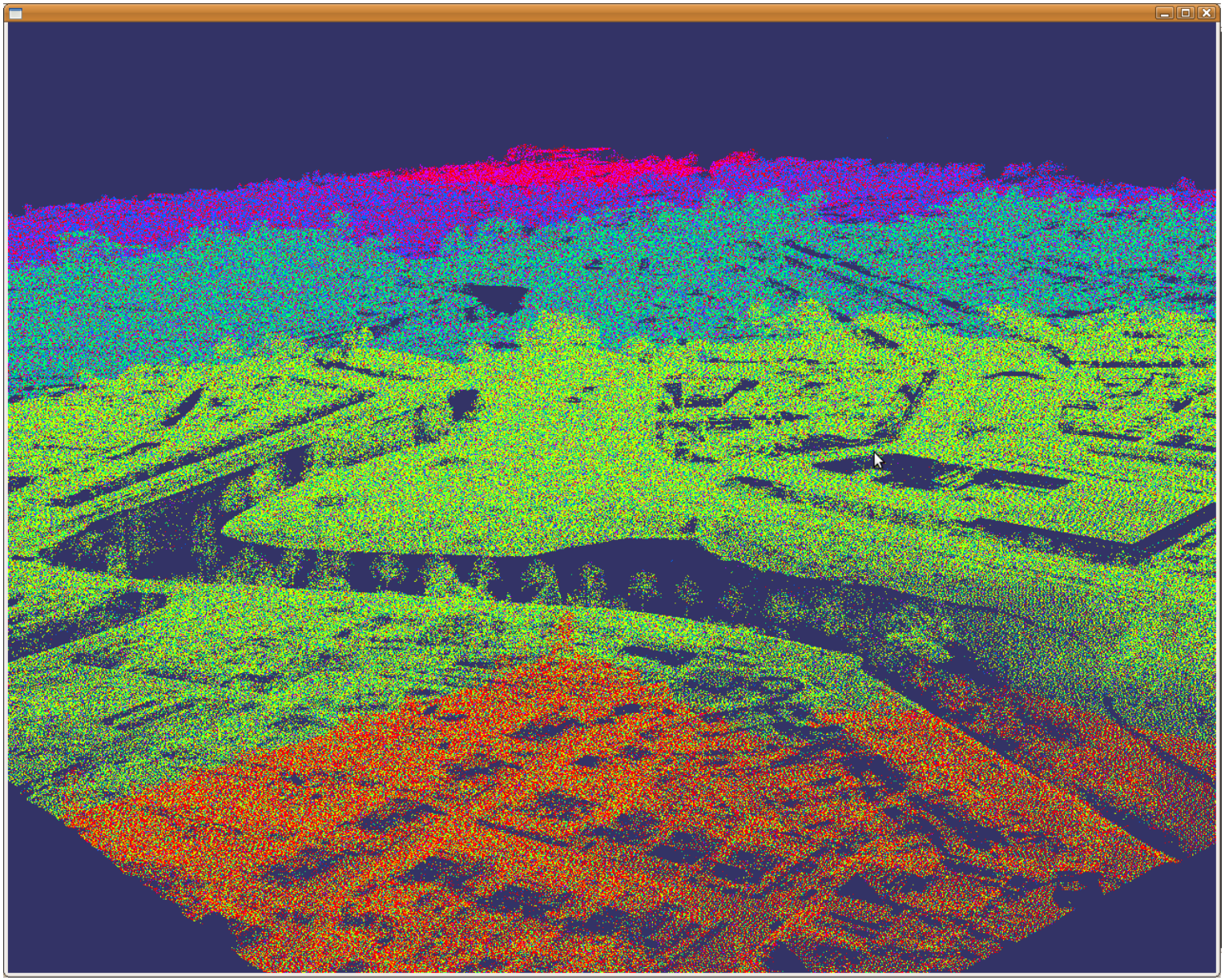
Multi-resolution data struture



Multi-resolution data struture

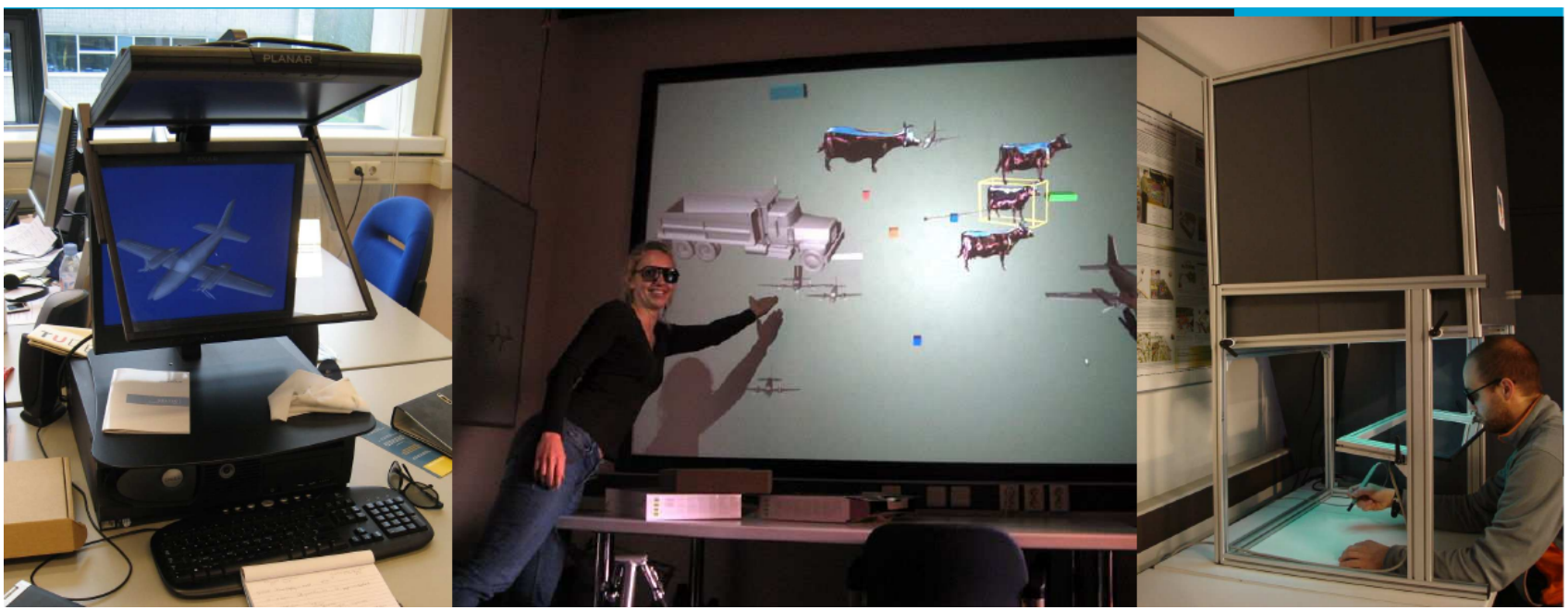


LOD effects



Interaction

- Stereoscopic display
- 3D interaction with space-mouse, Wii balance board interaction
- More info: PhD thesis *Techniques and Architectures for 3D Interaction*, Gerwin de Haan, TU Delft, 2009.

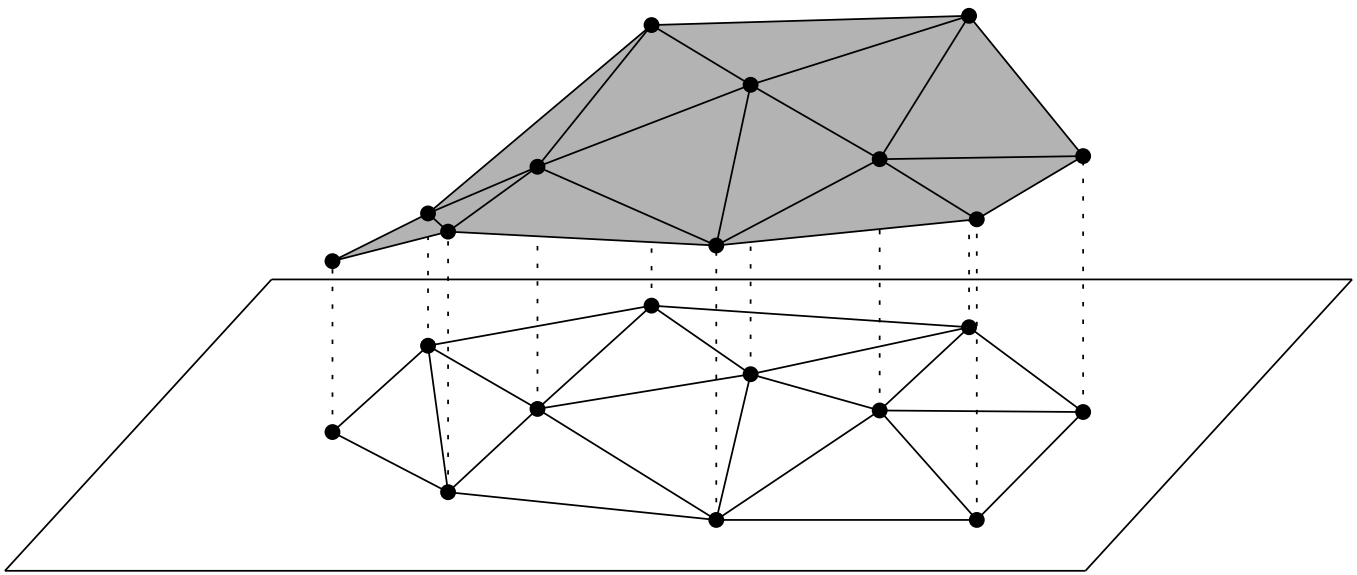


Interaction

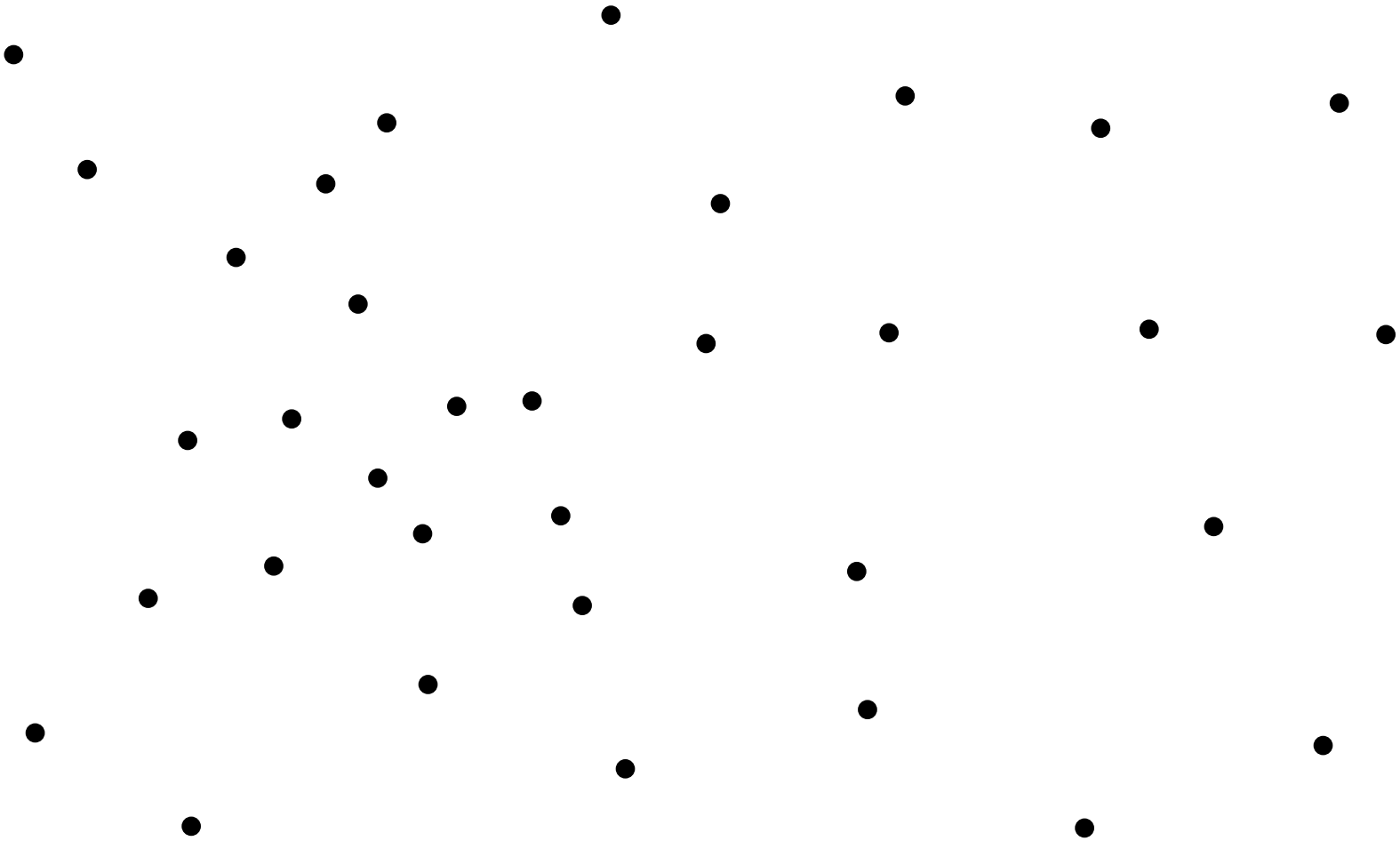
Analysis & Storage

Point clouds are often "2.5D surface"

LiDAR datasets are formed by scattered points in 3D space, which are the samples of a surface that can be projected on the horizontal plan.

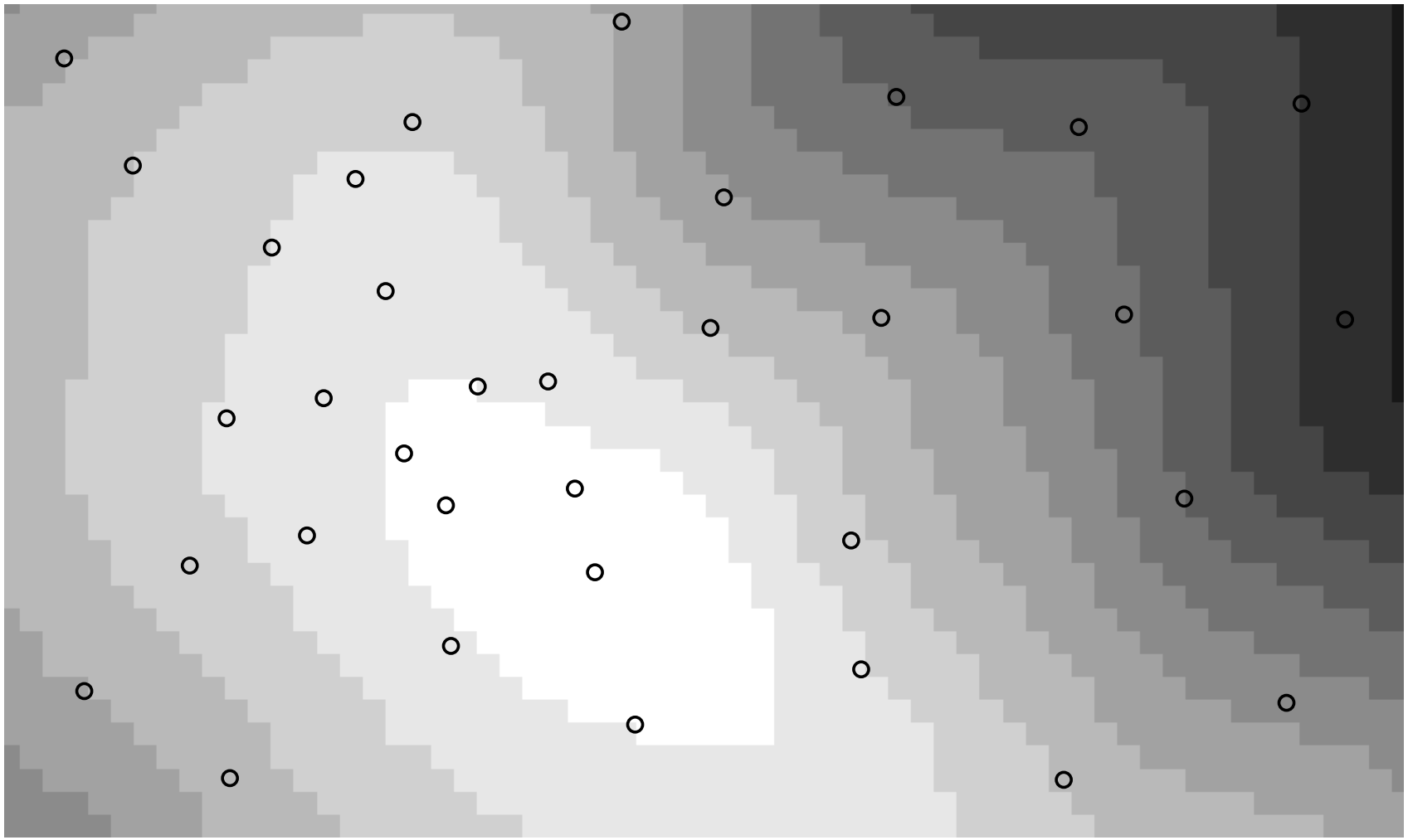


Reconstruction of the surface



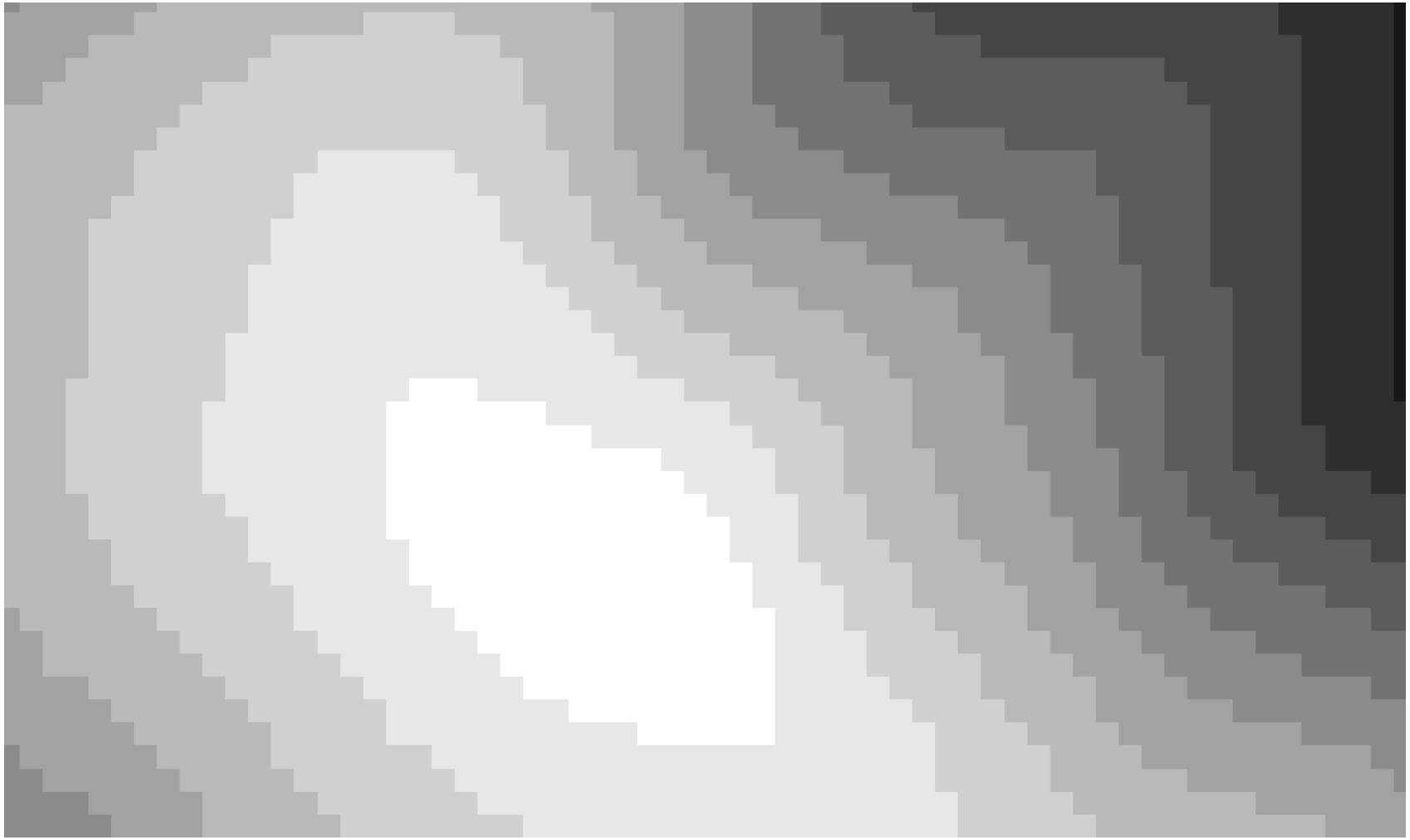
Original LiDAR points

Reconstruction of the surface



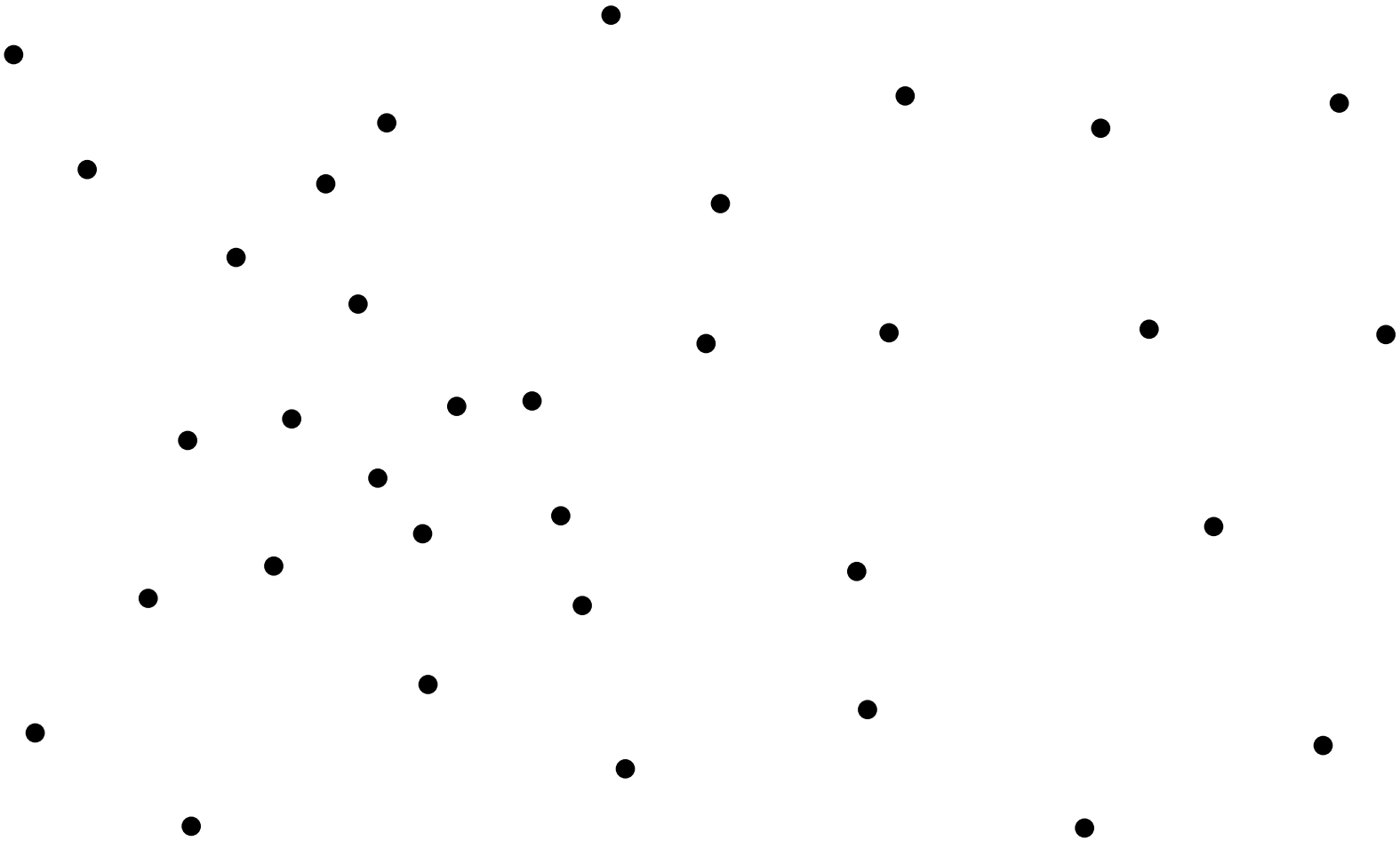
Raster representation

Reconstruction of the surface



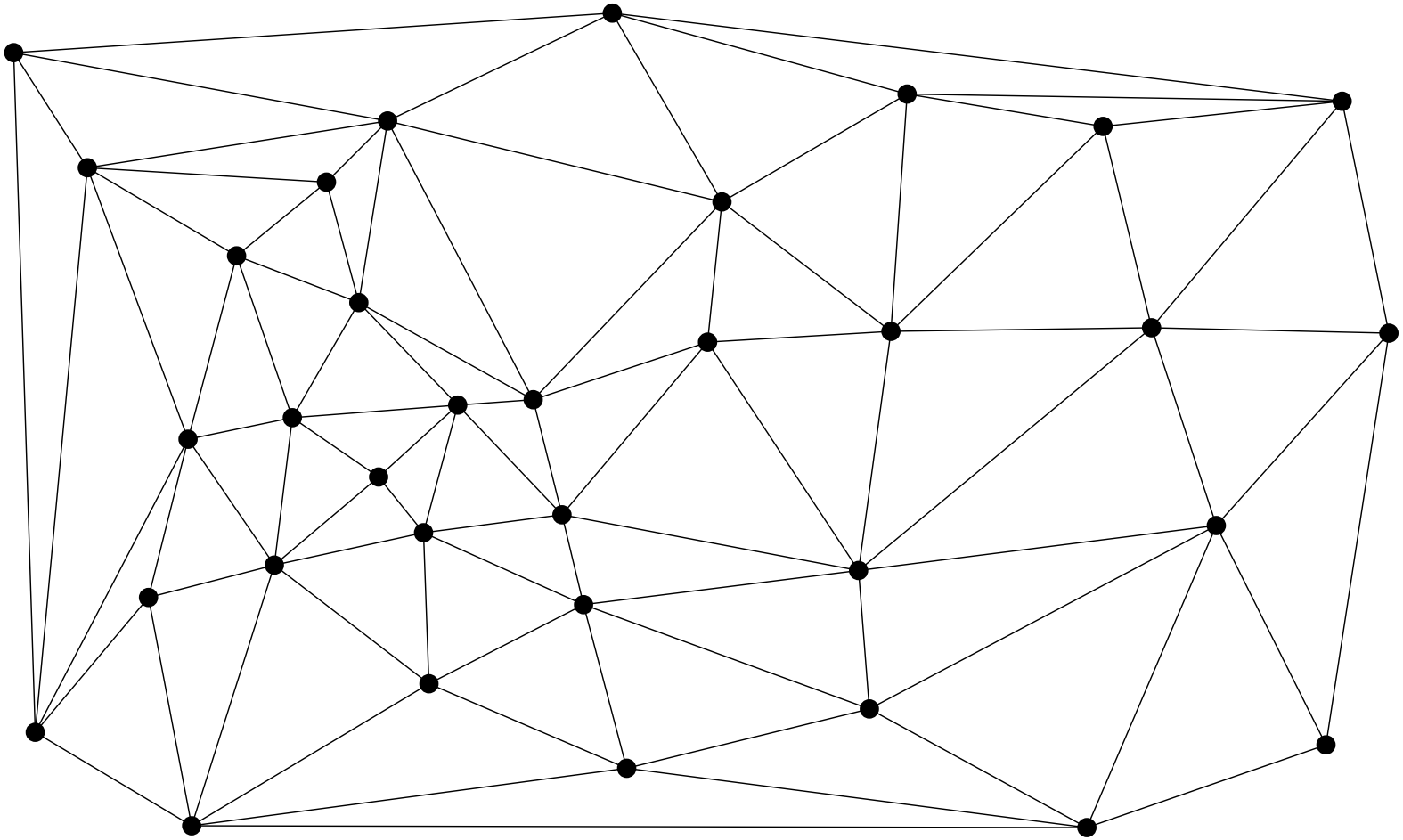
Raster representation

Reconstruction of the surface



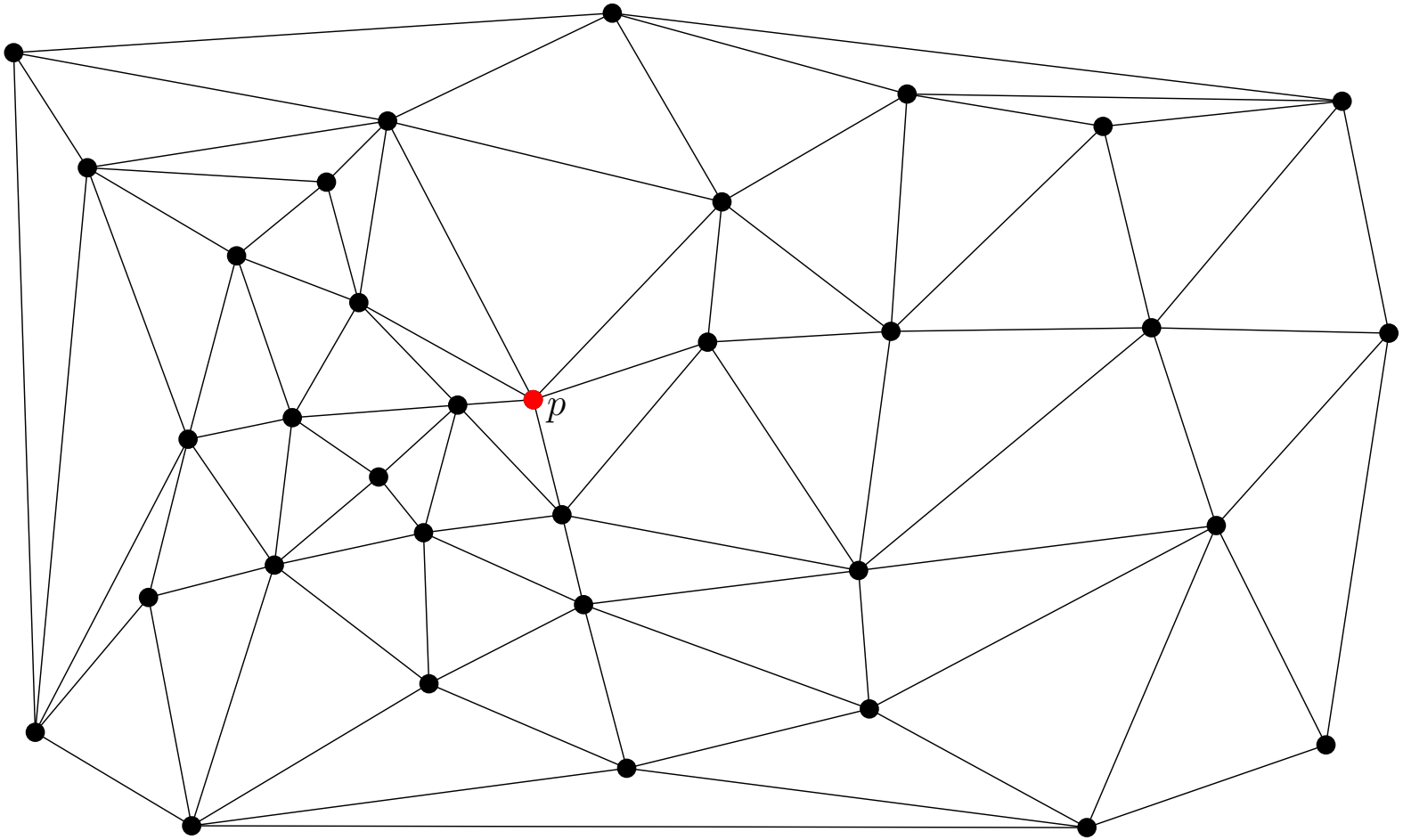
Original LiDAR points

Reconstruction of the surface



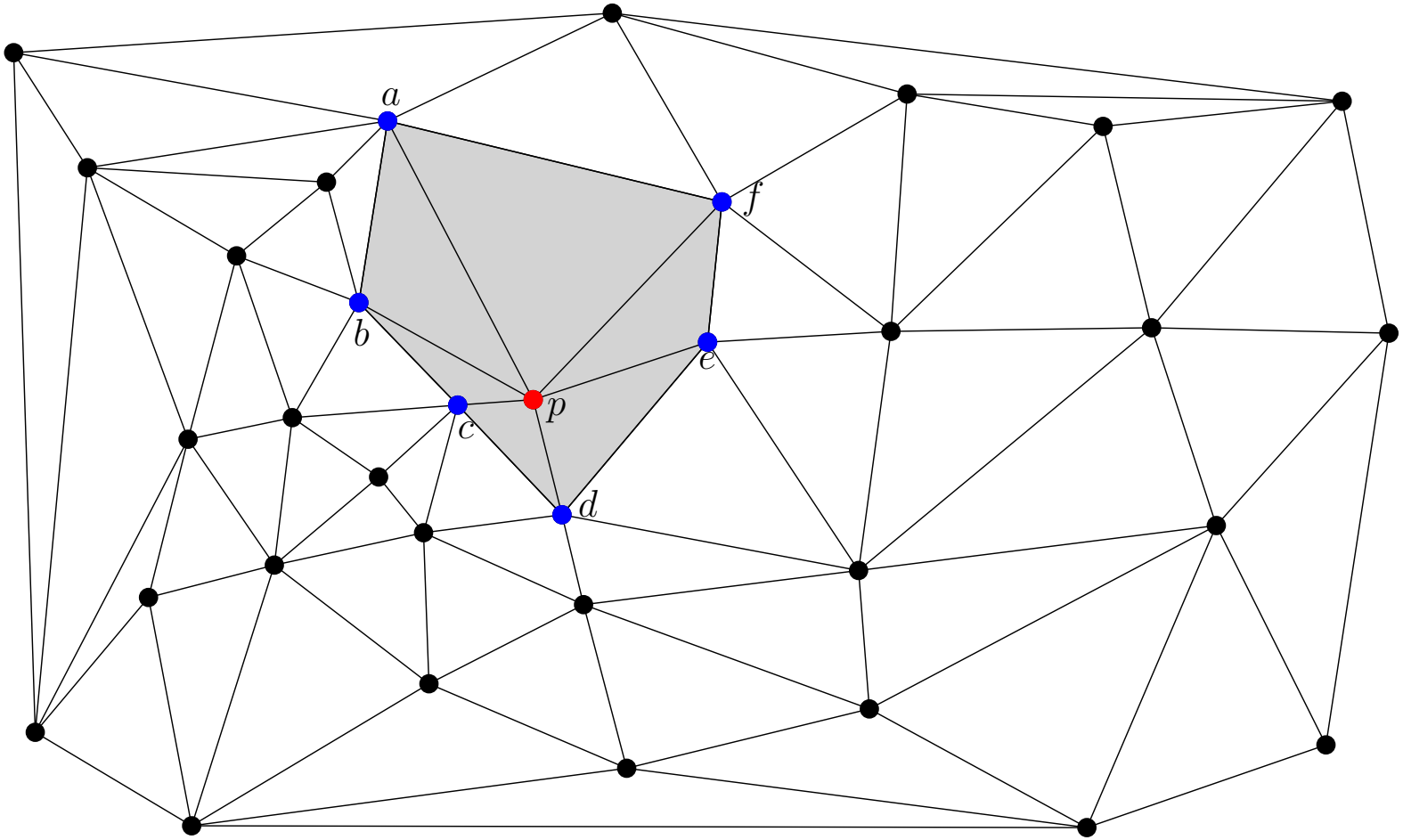
TIN (with Delaunay triangles)

Reconstruction of the surface



TIN (with Delaunay triangles)

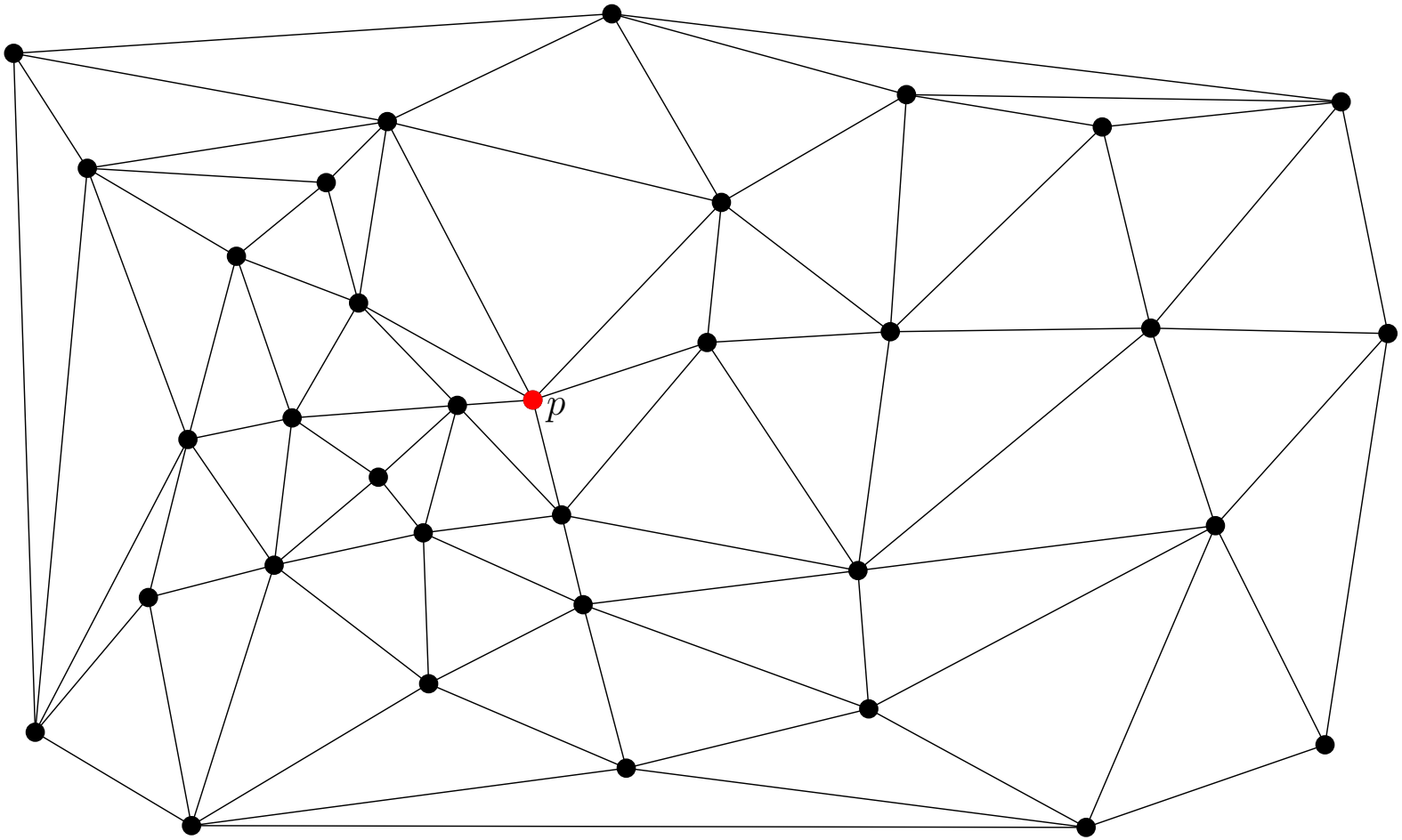
Reconstruction of the surface



TIN (with Delaunay triangles)

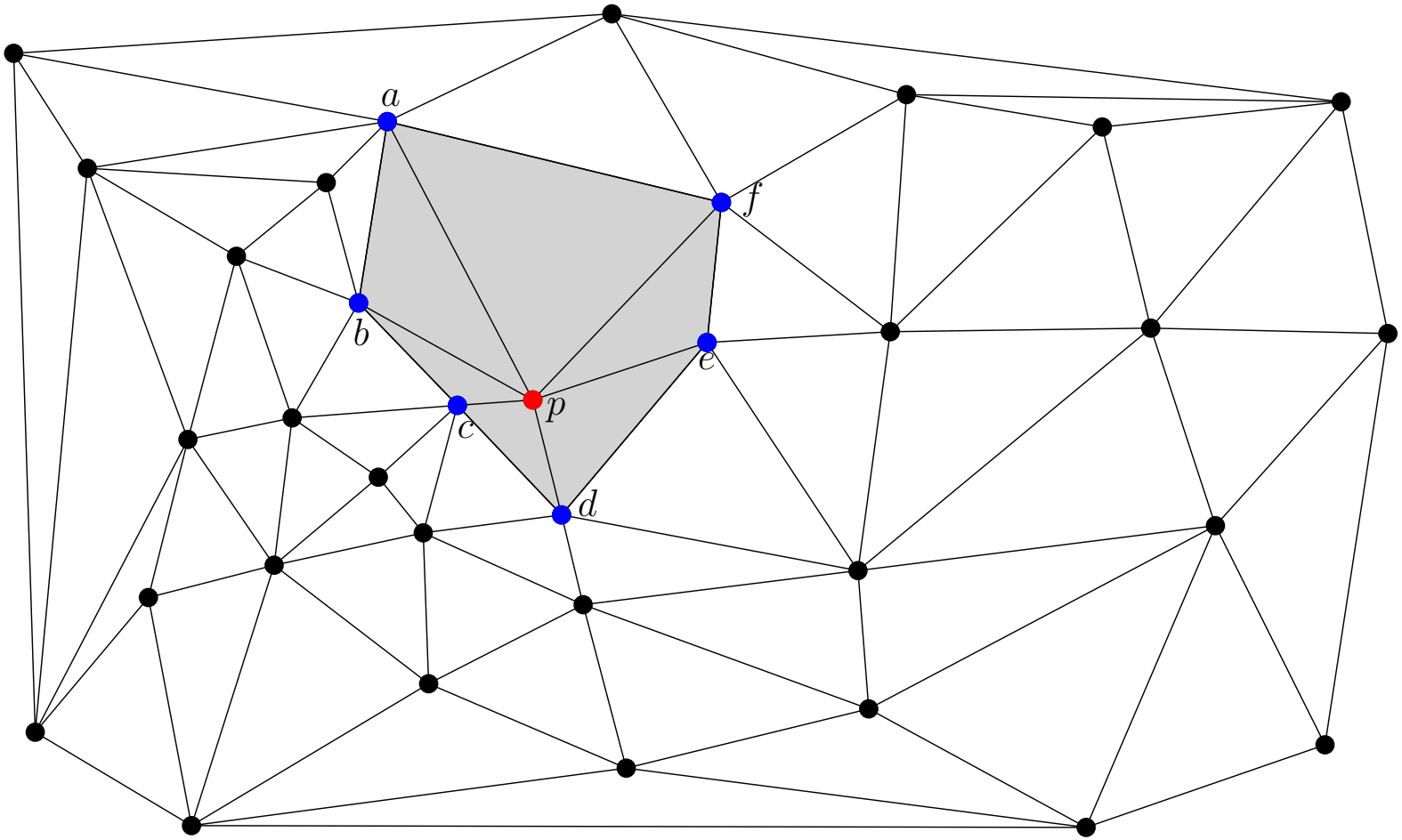
A star-based data structure

- Goes beyond the usual "store points and edges/triangles"
- Ideas come from data structures for compression of graphs



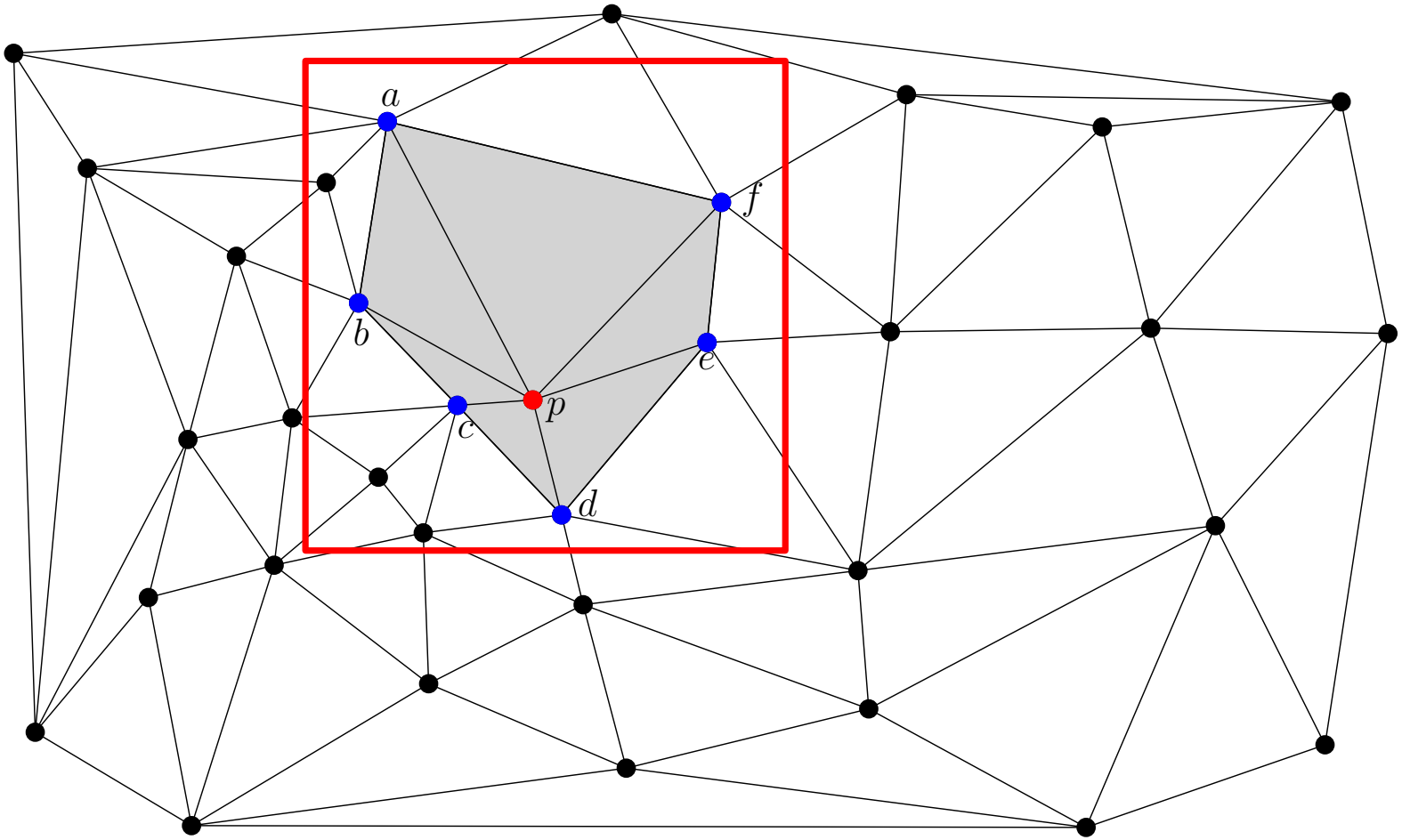
A star-based data structure

- Goes beyond the usual "store points and edges/triangles"
- Ideas come from data structures for compression of graphs



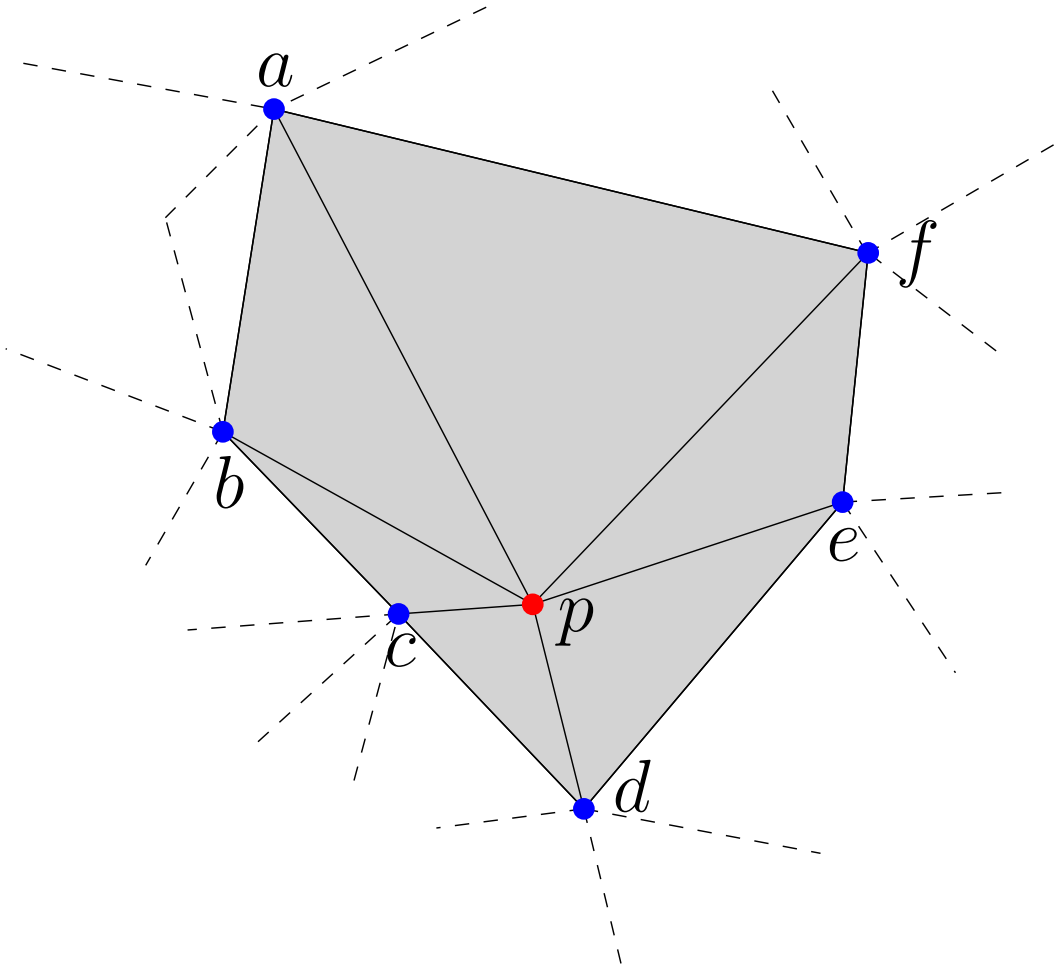
A star-based data structure

- Goes beyond the usual "store points and edges/triangles"
- Ideas come from data structures for compression of graphs



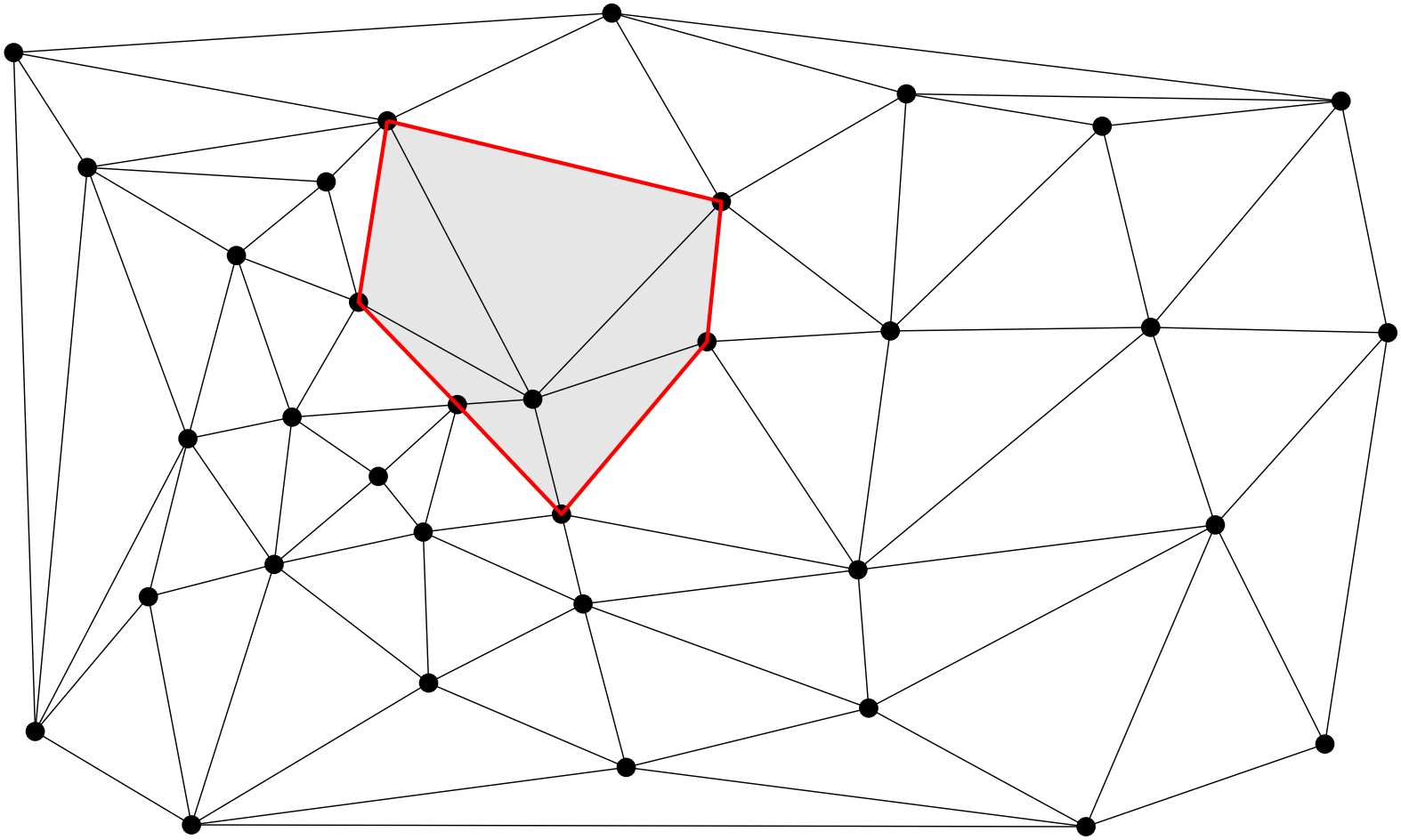
A star-based data structure

- Goes beyond the usual "store points and edges/triangles"
- Ideas come from data structures for compression of graphs

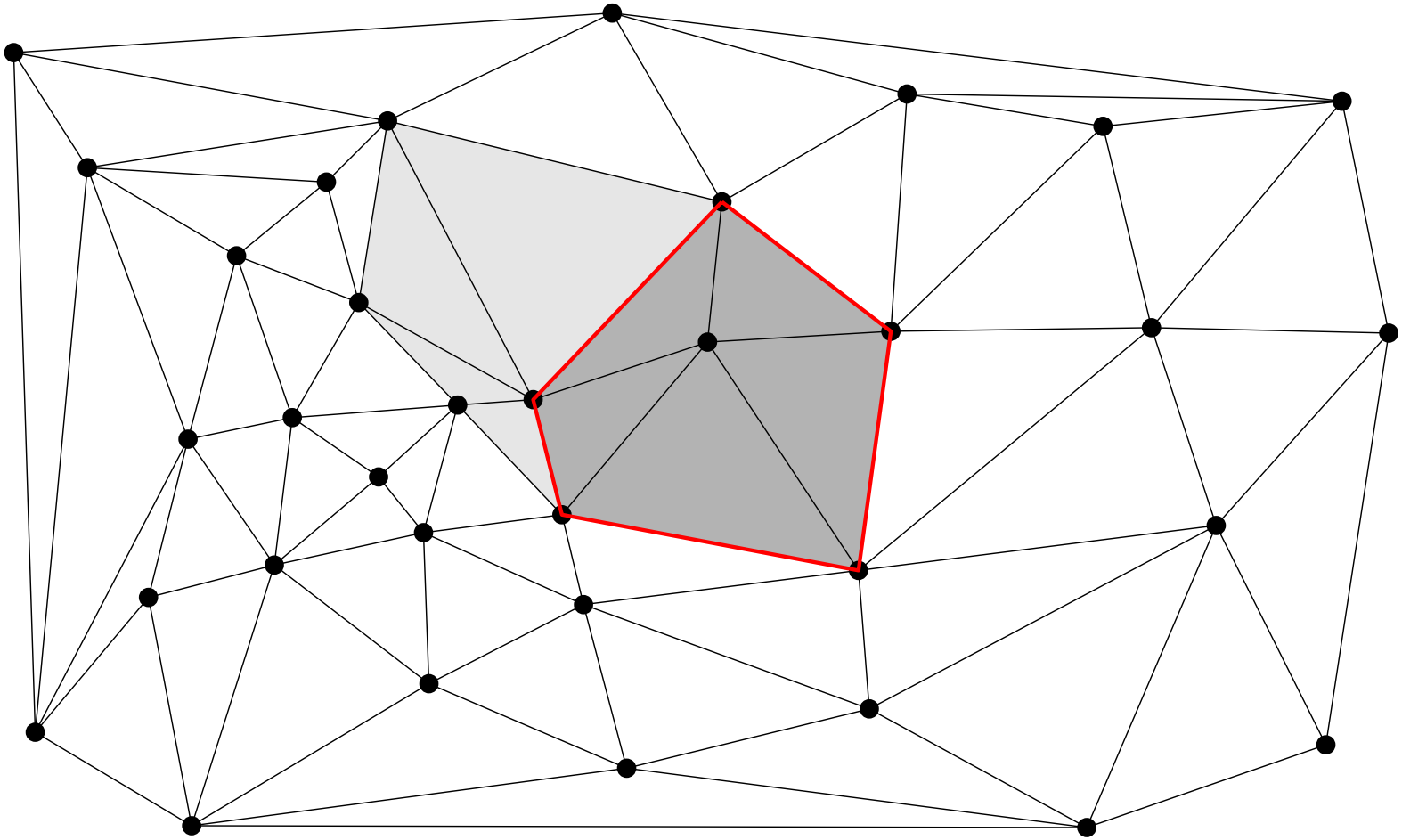


$$star(p) = abcdef$$

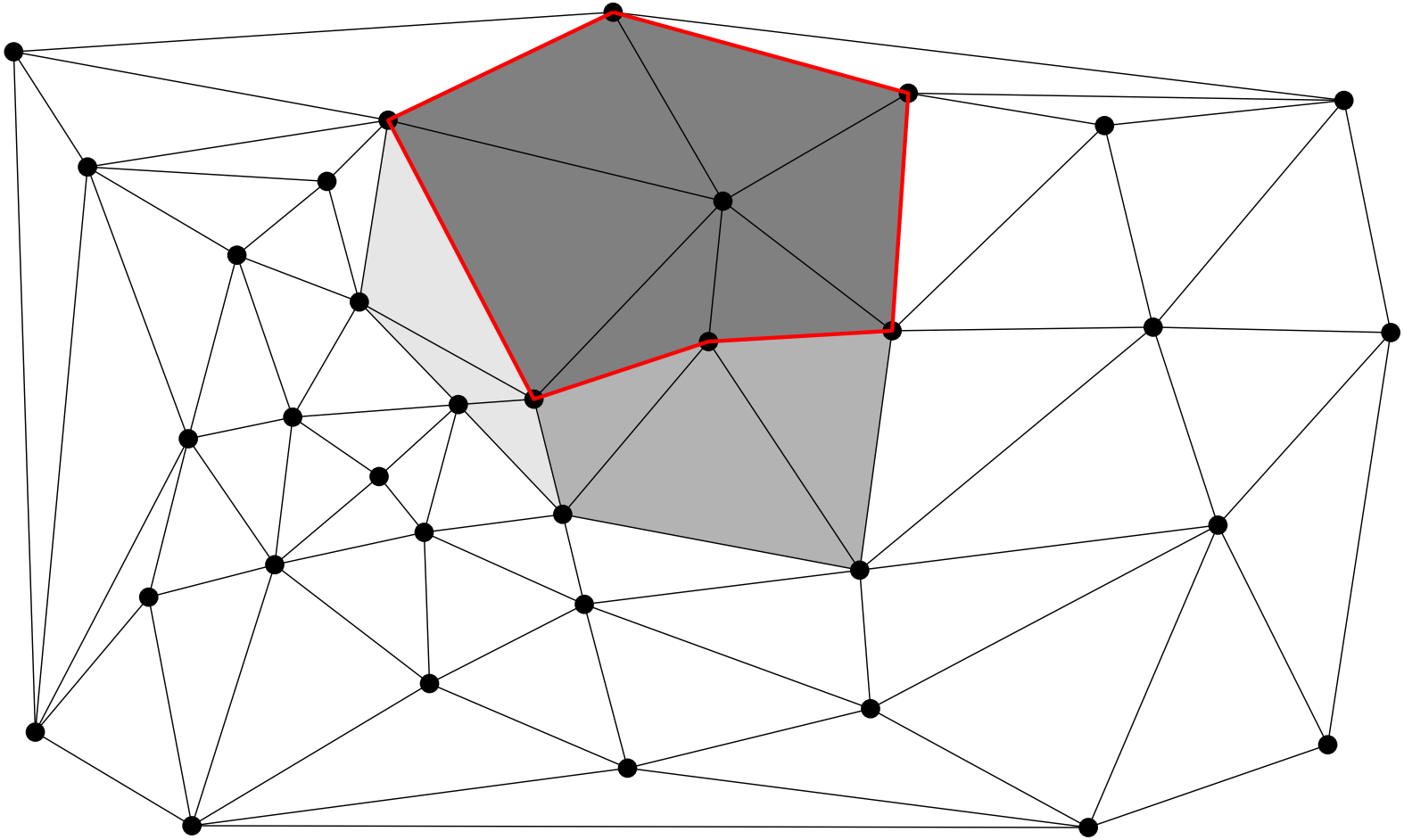
Every star(v) is stored $\longrightarrow$ implicit triangles



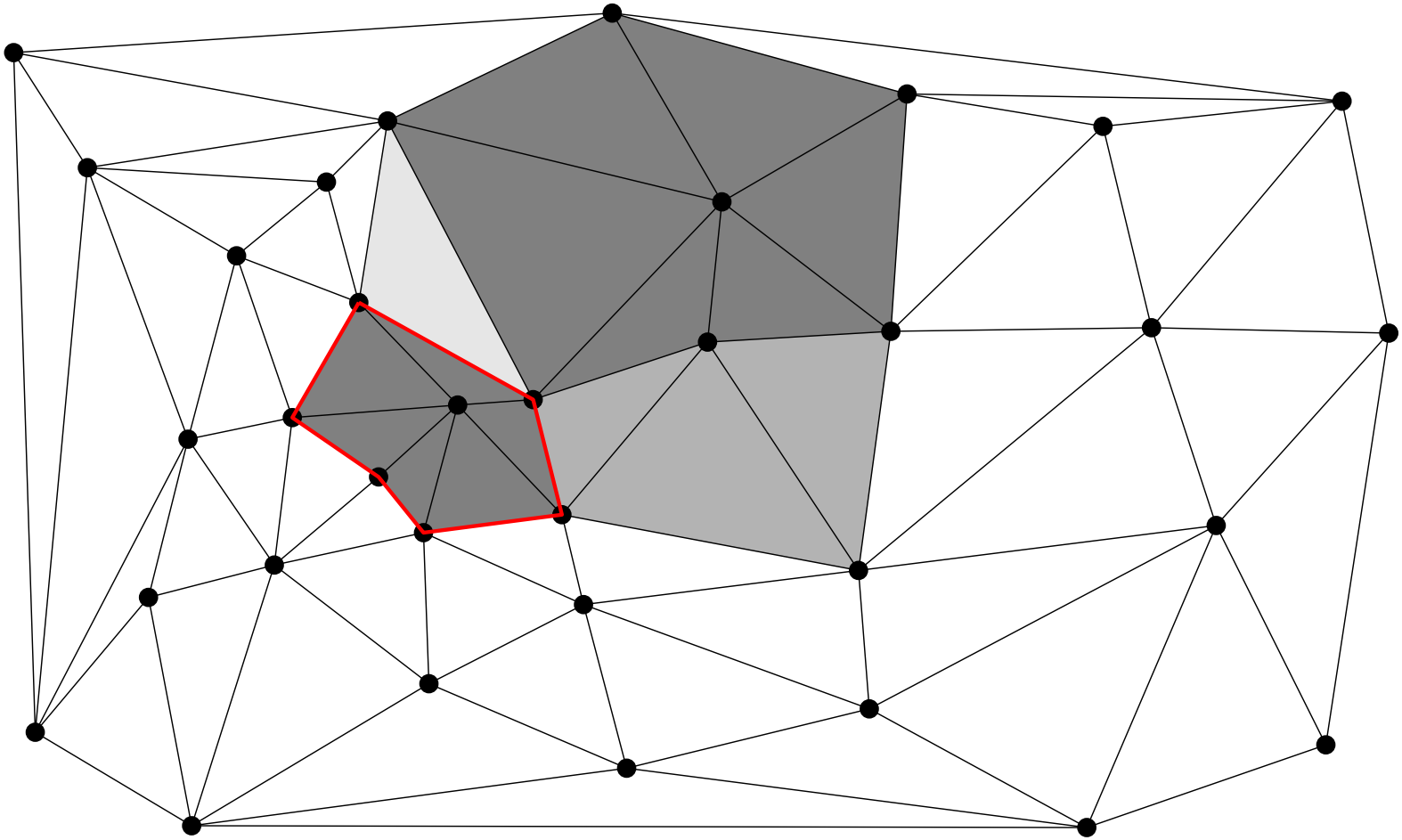
Every star(v) is stored $\longrightarrow$ implicit triangles



Every star(v) is stored $\longrightarrow$ implicit triangles



Every star(v) is stored $\longrightarrow$ implicit triangles



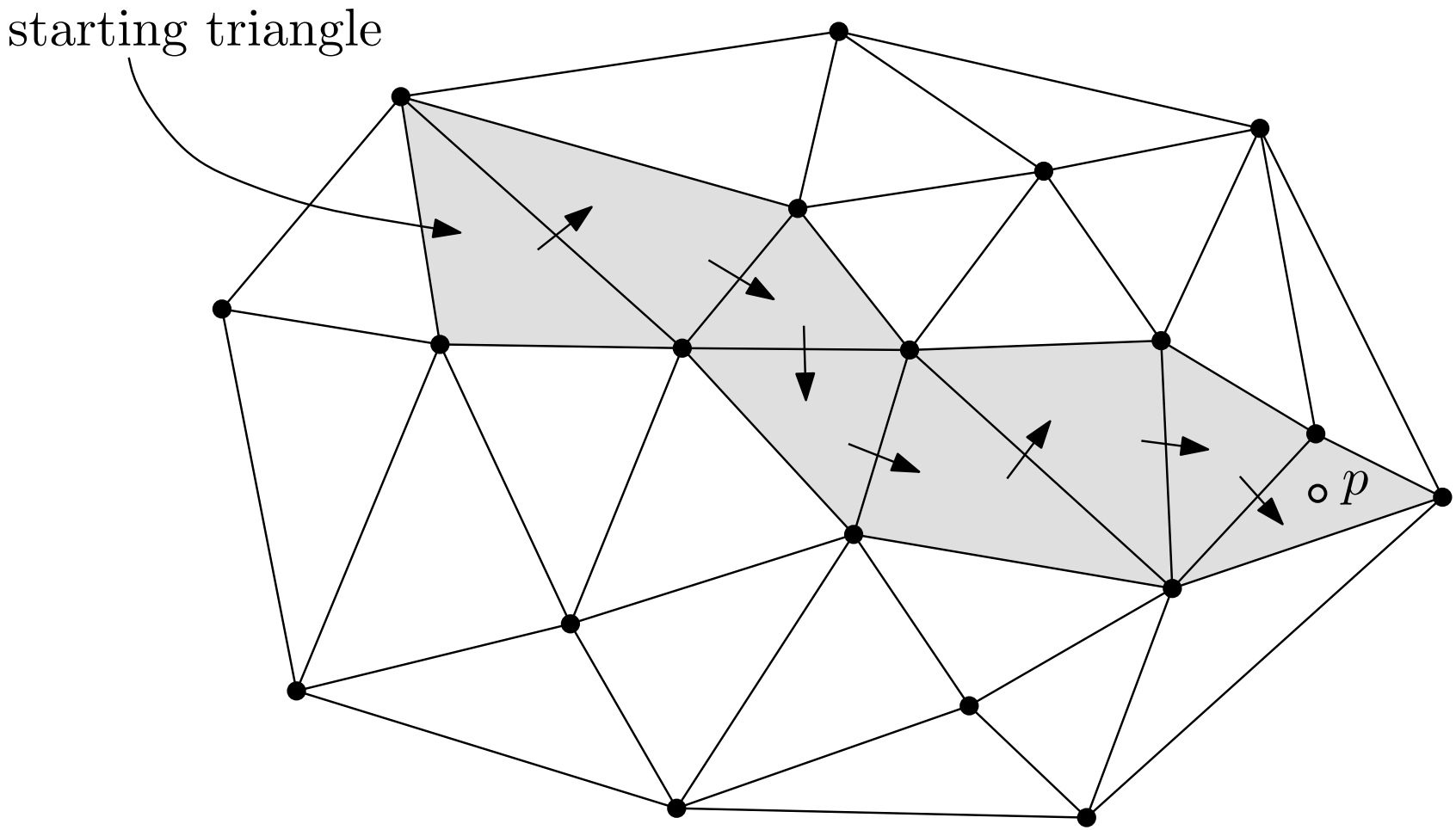
Stars in a DBMS

ID	x	y	z	star
1	3.21	5.23	2.11	2–44–55–61–23
2	5.19	29.01	4.55	7–98–111–233–222
3	22.43	15.99	8.19	99–101–73–23
...	...	...	...	...
5674	221.19	15.23	37.81	309–802–793–1111

Advantages:

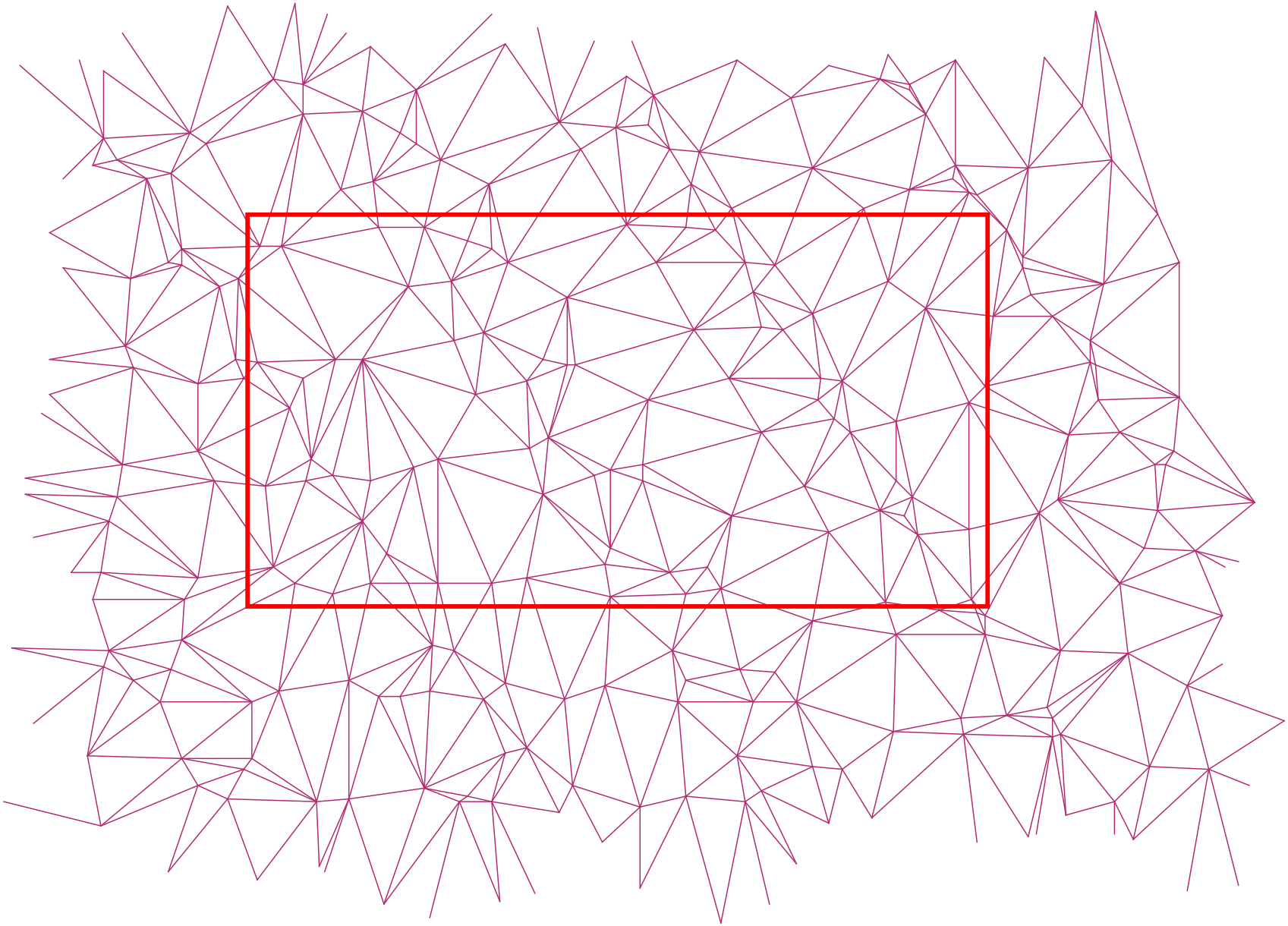
1. Only one table with $id - x - y - z - star$
2. No spatial index needed: fetching of triangles based on "walking"
3. Star column need not be filled ($\sim$ Simple Features)
4. Local updates are possible (insertion and removals)
5. Ideas are readily extensible to 3D for storing manipulating tetrahedra

Point Location = "Walking" in the triangulation



(Can be made efficient with some tricks [MSZ99])

Range Queries: also uses the triangulation



One problem: how to create that DT in the first place?

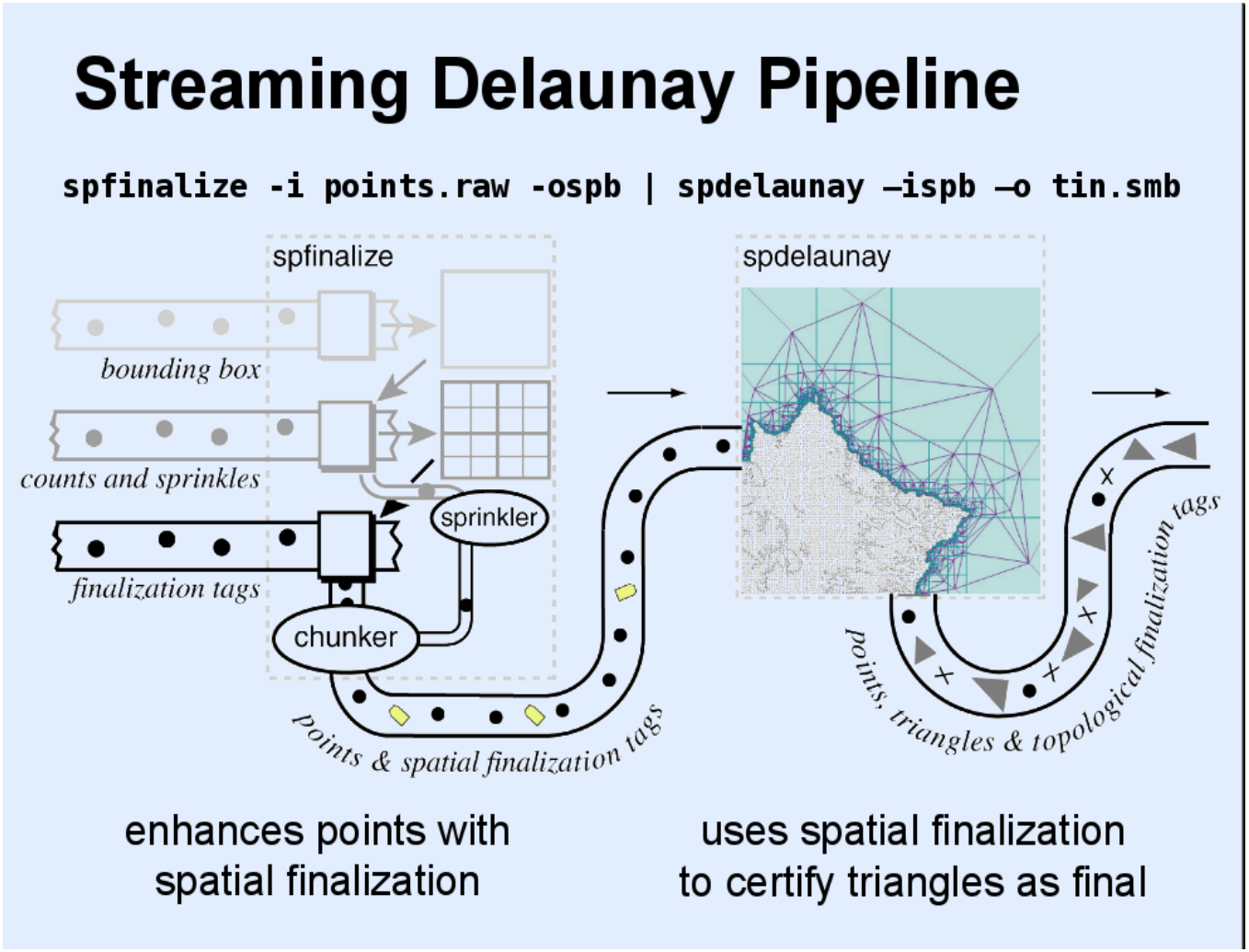
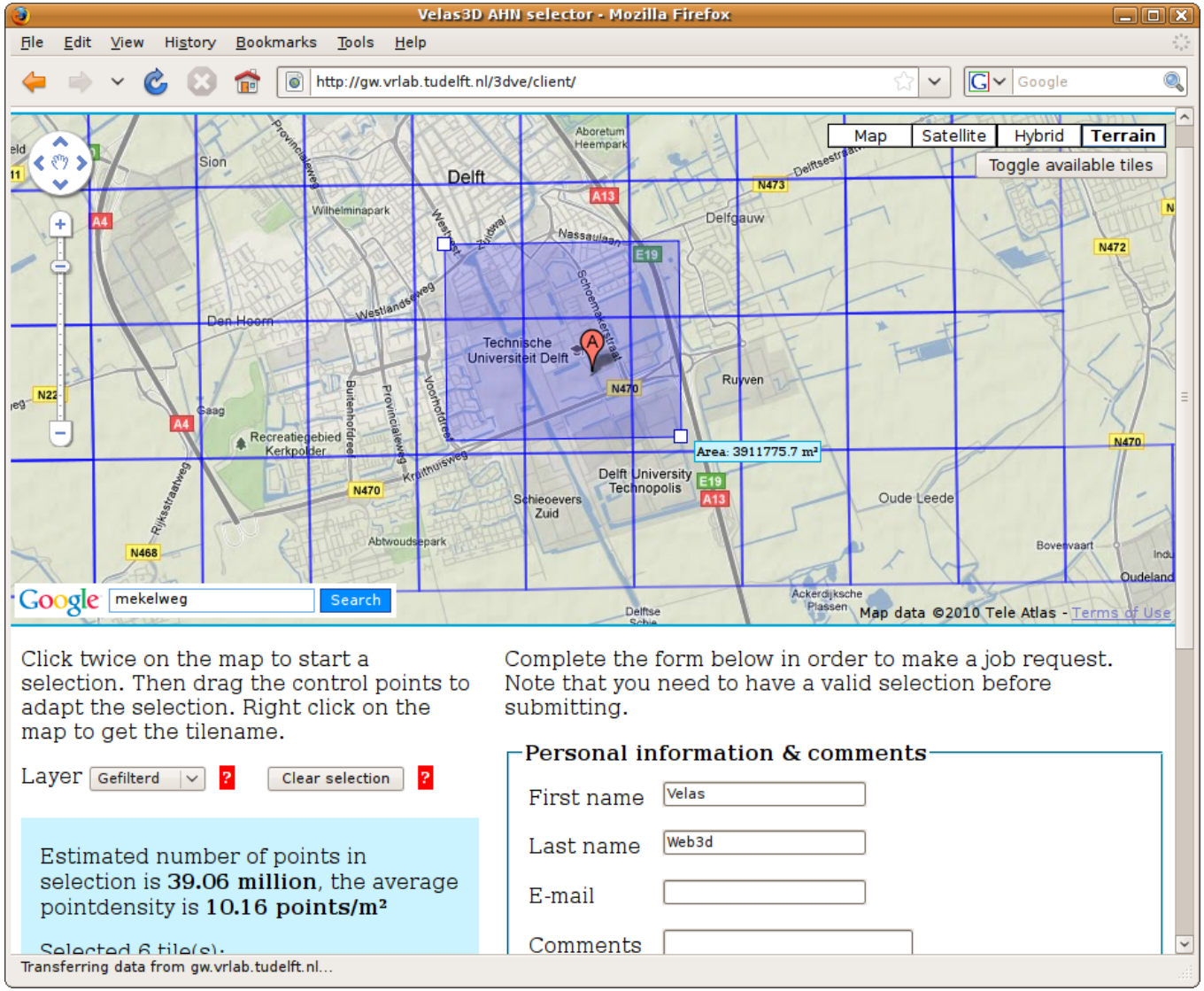


Figure from Martin Isenburg's presentation at GIScience 2006 [ILSS06]

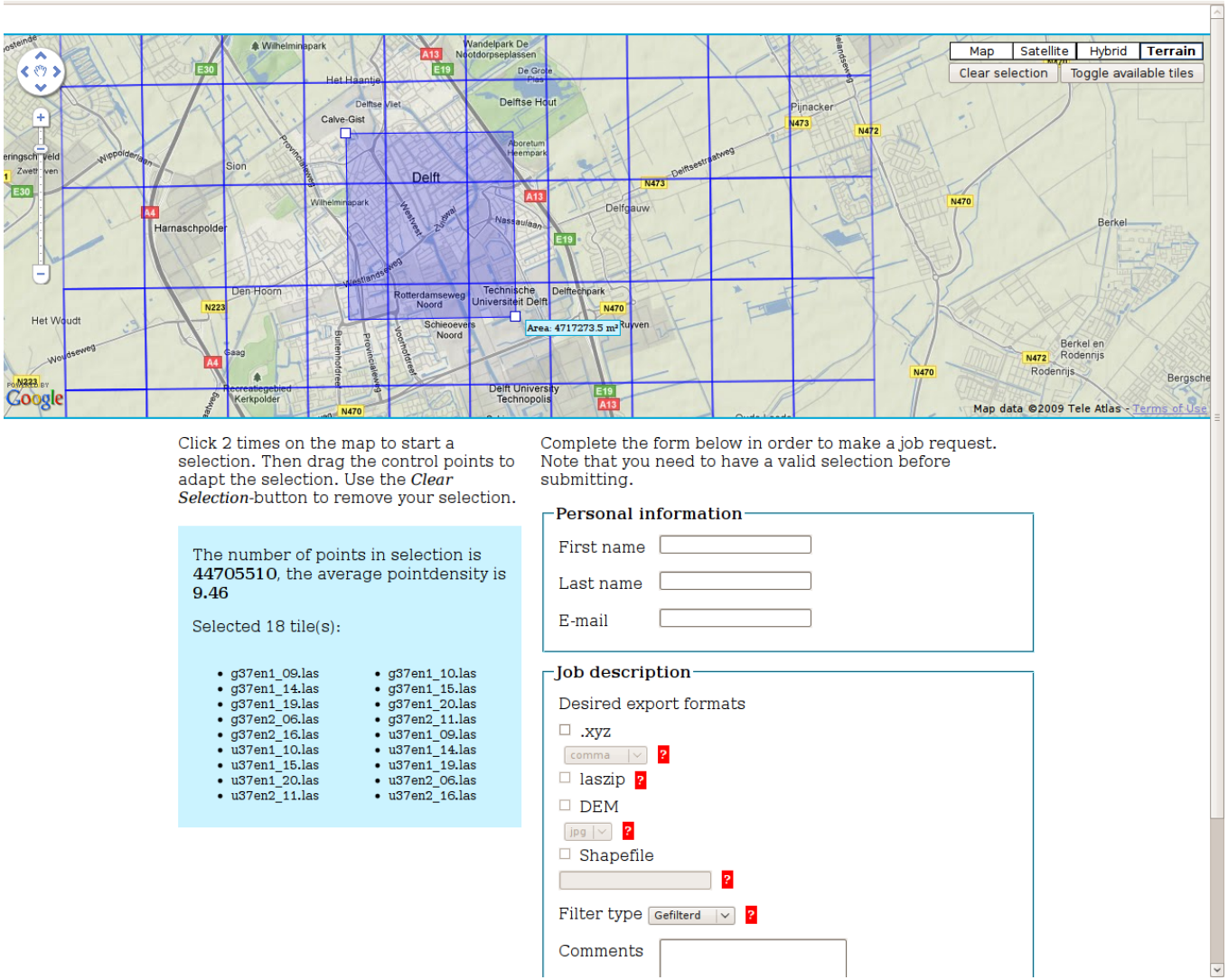
Distribution of the Data

Velas3D: An online AHN^2 selector



http://gw.vrlab.tudelft.nl/3dve/client

Velas3D: An online AHN^2 selector



Click 2 times on the map to start a selection. Then drag the control points to adapt the selection. Use the ***Clear Selection***-button to remove your selection.

The number of points in selection is **44705510**, the average pointdensity is **9.46**

Selected 18 tile(s):

- g37en1_09.las
- g37en1_10.las
- g37en1_14.las
- g37en1_15.las
- g37en1_19.las
- g37en1_20.las
- g37en2_06.las
- g37en2_11.las
- g37en2_16.las
- u37en1_09.las
- u37en1_10.las
- u37en1_14.las
- u37en1_15.las
- u37en1_19.las
- u37en1_20.las
- u37en2_06.las
- u37en2_11.las
- u37en2_16.las

Complete the form below in order to make a job request. Note that you need to have a valid selection before submitting.

Personal information

First name

Last name

E-mail

Job description

Desired export formats

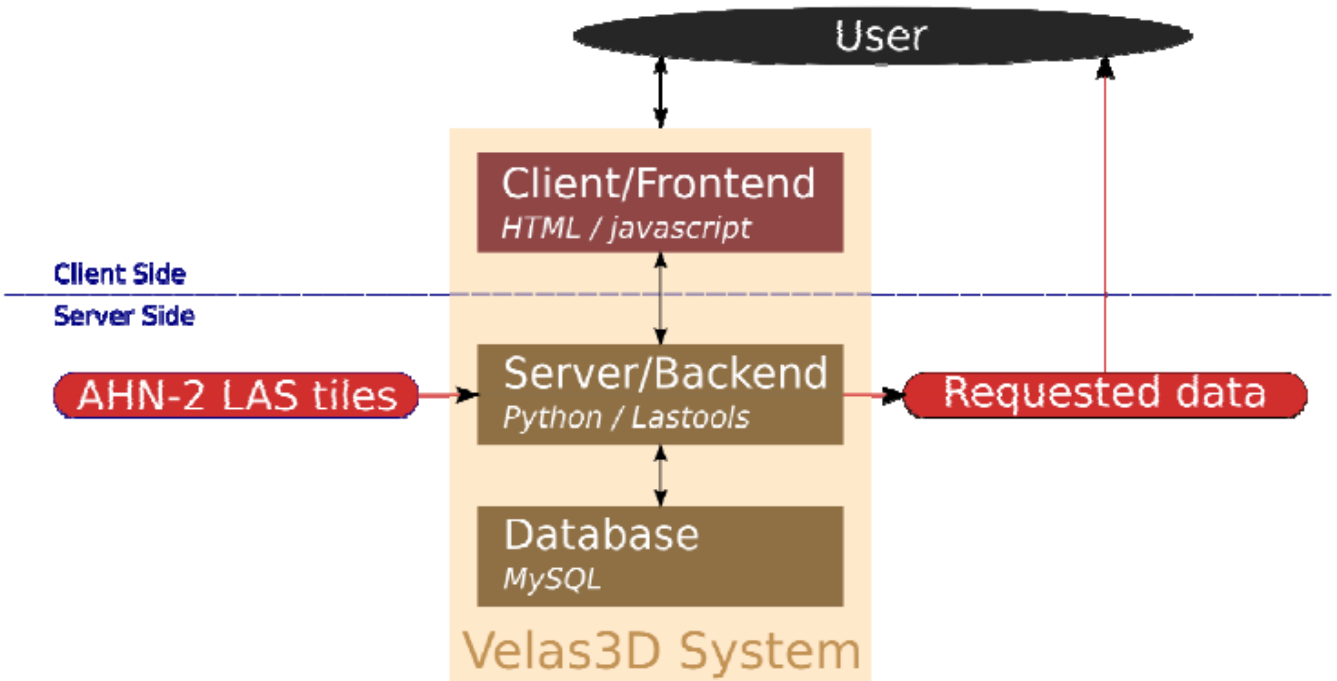
- .xyz (comma)
- laszip
- DEM (jpg)
- Shapefile

Filter type: Gefilterd

Comments

`http://gw.vrlab.tudelft.nl/3dve/client`

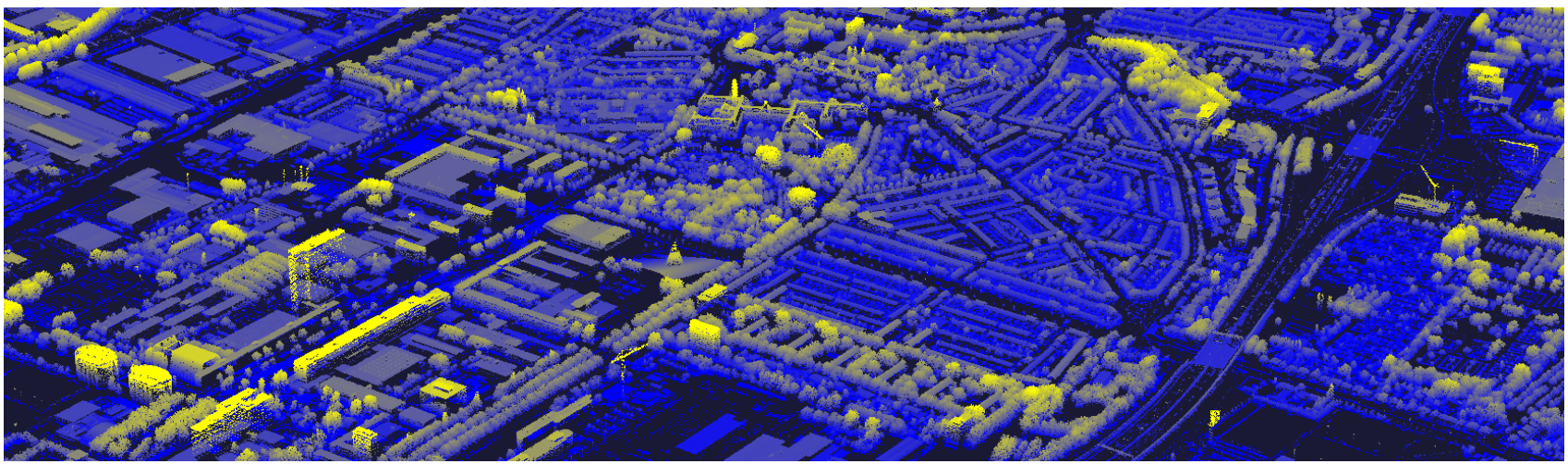
Velas3D: An online AHN^2 selector



Thanks for your attention

Hugo Ledoux
h.ledoux@tudelft.nl

Gerwin de Haan
g.dehaan@tudelft.nl



References

Martin Isenburg, Yuanxin Liu, Jonathan Richard Shewchuk, and Jack Snoeyink.
Streaming computation of Delaunay triangulations.
ACM Transactions on Graphics, 25(3):1049–1056, 2006.

Ernst P. Mücke, Isaac Saias, and Binhai Zhu.
Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations.
Computational Geometry—Theory and Applications, 12:63–83, 1999.