Point Cloud Workshop at OTB

Wilko Quak
Hugo Ledoux
Theo tijssen
Maarten Vermeij

2008-10-17

OTB Research Institute for Housing, Urban and Mobility Studies

TU Delft
Delft University of Technology

Point Clouds

New research for GISt

2008-10-17

OTB Research Institute for Housing, Urban and Mobility Studies

TU Delft
Delft University of Technology

Agenda

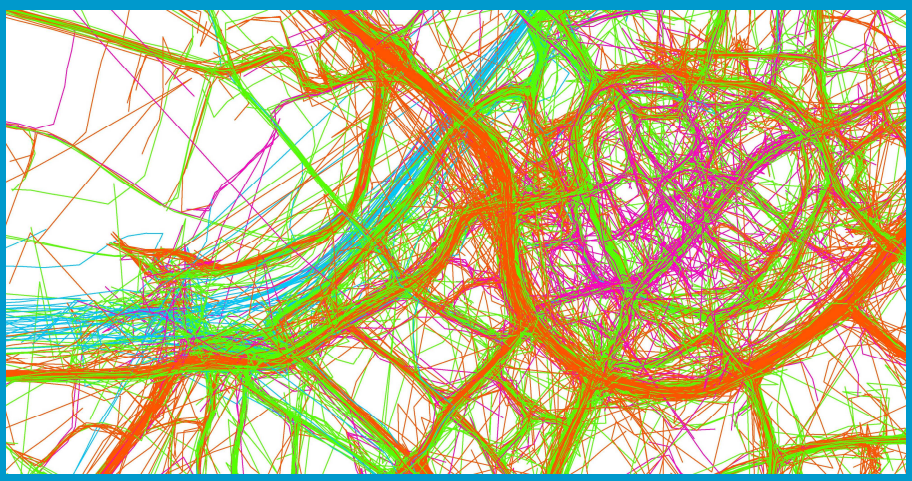
- Wilko: Introduction
- Theo: Oracle Storage
- Hugo: Point Cloud Modeling
- Maarten: Point Cloud visualization
- Wilko: On MonetDB

OTB Research Institute for Housing, Urban and Mobility Studies

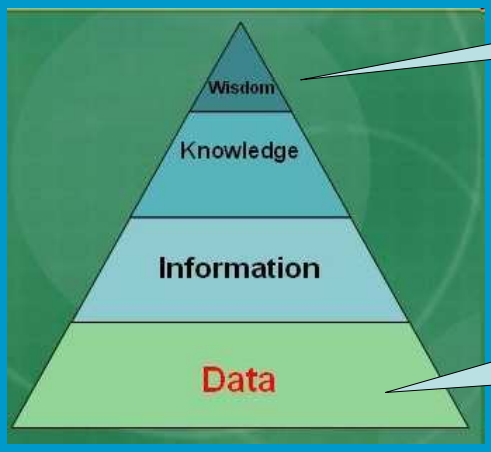
TU Delft

Point Clouds

- Examples
- GISt and point clouds (why and what)



Why are point cloud interesting



Should grow in quality

Grows in size (point clouds do, parcels don't)

Section GISt and point clouds

- What are we going to do?
- What is the state-of-the-art?
- Should we team up with other institutes?
- How do we collaborate?

Proposals

- Figure out what state of the art is:
  - Collect a bunch of relevant articles and discuss them (on Fridays).
- How can we collaborate:
  - Find common platform (DBMS)
  - Related to open-source ideas.
- Related Research institues:
  - Scientific visualization group at TUDelft EWI.
  - CWI for DBMS storage.
  - Sensor Network group at Wageningen.

Research Question

- Should we store the raw point clouds in a DBMS?
- Can we store them efficiently?
- What operations do we want to perform on the point clouds?

Agenda

- Wilko: Introduction
- Theo: Oracle Storage
- Hugo: Point Cloud Modeling
- Maarten: Point Cloud visualization
- Wilko: On MonetDB

Problem: Redundancy and noise in point clouds (thanks Rod)

Redundant data. Good compression

Interesting part. Reasonable compression

Noise. Uncompressable

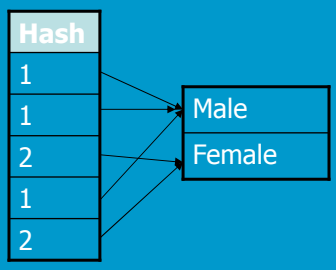
MonetDB and point clouds

•Adapt MonetDB string storage trick for point clouds

What the user sees:

Table
Male
Male
Female
Male
Female

Implementation:



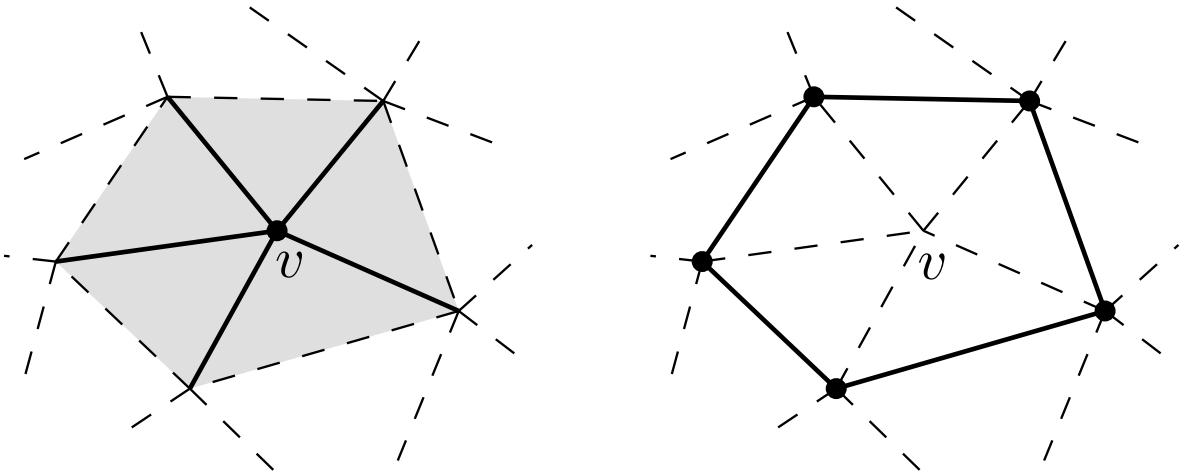
A star-based data structure for storing very large TINs

Hugo Ledoux

TU Delft
Technische Universiteit Delft

Lunch meeting
October 20, 2008

What's a star anyway?



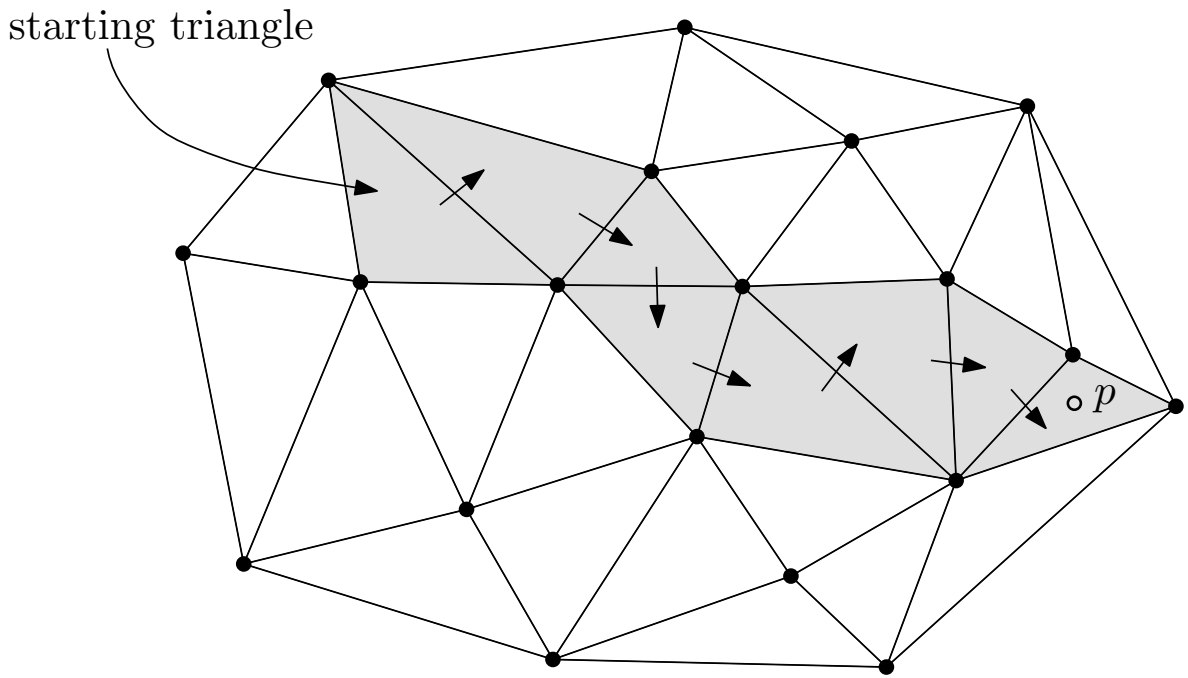
$\mathrm{star}(v)$ $\mathrm{link}(v)$

1. The whole TIN is stored in the DBMS
2. It goes beyond the usual "store points and edges and triangles": triangles are *implicitly* stored
3. Basically, the star of every vertex is stored
4. Based on recent advances in the compression of graphs

Advantages

1. No spatial index needed: fetching of triangles based on "walking"
2. One table with $id - x - y - z - binary$
3. Can be combined with spatial indexes also, since the binary column doesn't have to be filled
4. Ideas are readily extensible to higher dimensions
5. With only that, we obtain a structure that is fully "topologic", i.e. we can obtain the adjacency between points/triangles efficiently.
6. We can update/manipulate very easily the structure (simplification algorithms could be build over it directly in the database for instance)

Walking



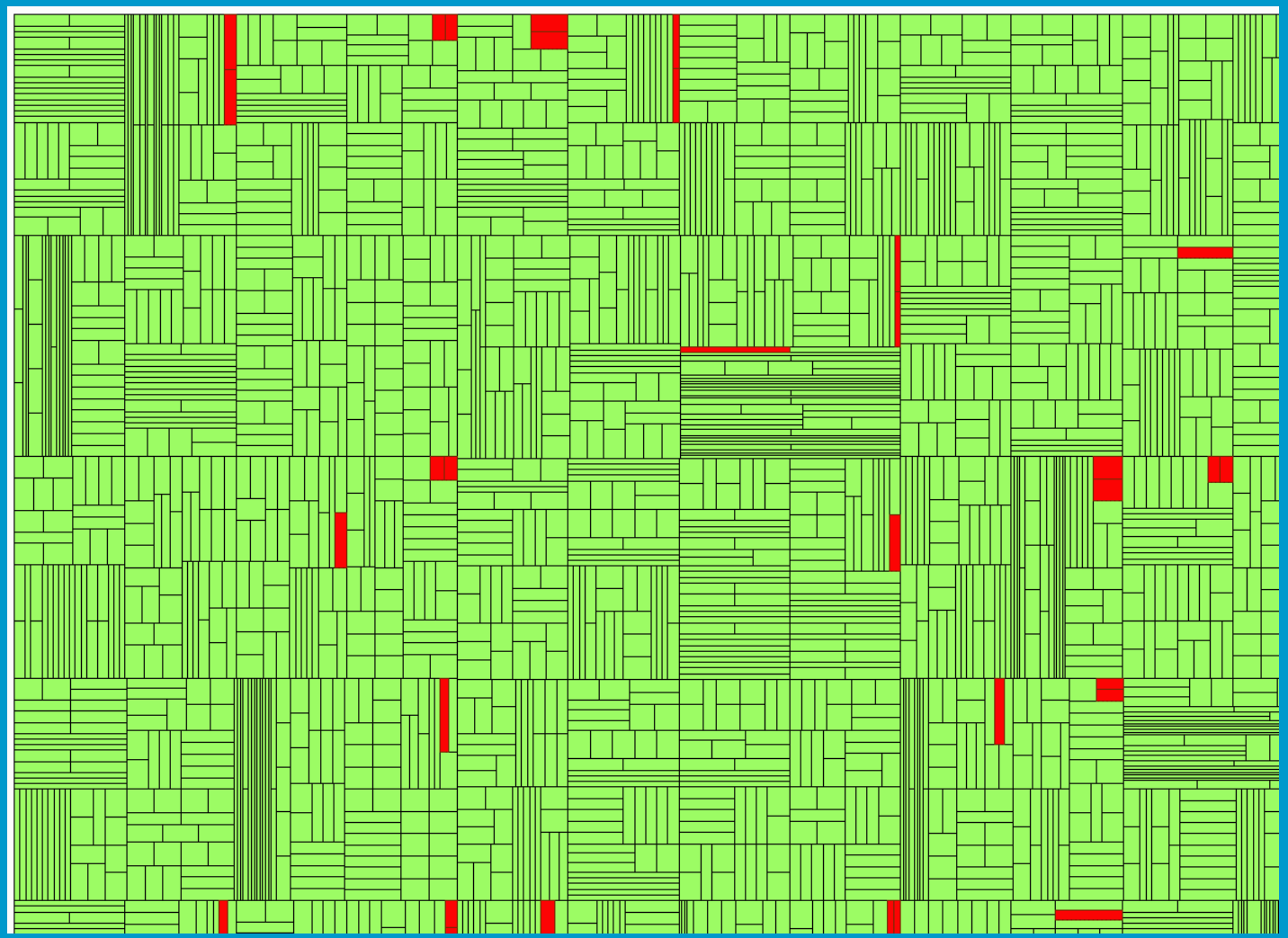
Point clouds (TIN) in Oracle 11g

Functionality:

- Create point cloud (SDO_PC datatype) and store points
  - prepare points in table
  - initialize point cloud
  - load points
- Select points (clip) based on spatial query
- Retrieve subset of points as SDO_GEOMETRY multipoint

TU Delft

Point cloud is “partitioned” in blocks



TU Delft

Points are stored in block table

```
SQL> describe BLKTAB;

 Name                      Null?    Type
 ------------------------- -------- ---------------------
 OBJ_ID                             NUMBER
 BLK_ID                             NUMBER
 BLK_EXTENT                         MDSYS.SDO_GEOMETRY
 BLK_DOMAIN                         MDSYS.SDO_ORGSCL_TYPE
 PCBLK_MIN_RES                      NUMBER
 PCBLK_MAX_RES                      NUMBER
 NUM_POINTS                         NUMBER
 NUM_UNSORTED_POINTS                NUMBER
 PT_SORT_DIM                        NUMBER
 POINTS                             BLOB
```


TU Delft

TIN block table

Name	Null?	Type
OBJ_ID		NUMBER
BLK_ID		NUMBER
BLK_EXTENT		MDSYS.SDO_GEOMETRY
BLK_DOMAIN		MDSYS.SDO_ORGSCL_TYPE
PCBLK_MIN_RES		NUMBER
PCBLK_MAX_RES		NUMBER
NUM_POINTS		NUMBER
NUM_UNSORTED_POINTS		NUMBER
PT_SORT_DIM		NUMBER
POINTS		BLOB
TR_LVL		NUMBER
TR_RES		NUMBER
NUM_TRIANGLES		NUMBER
TR_SORT_DIM		NUMBER
TRIANGLES		BLOB

TU Delft