Representing three-dimensional topography in a DBMS with a star-based data structure

Hugo Ledoux Martijn Meijers

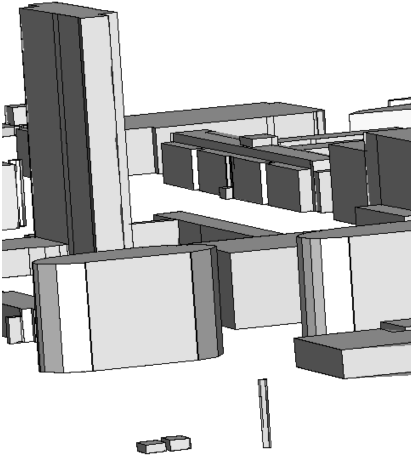
TU Delft
Delft University of Technology

3D GeoInfo 2012 (Québec City, Canada)
2012–06–16

Representing and storing 3D topography

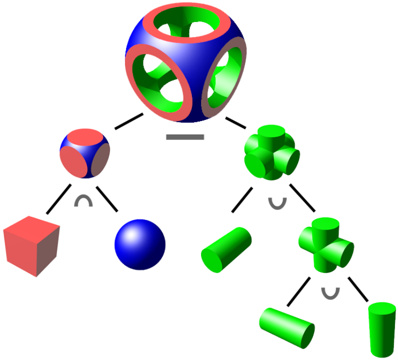
Our options:

1. *b-rep* (CityGML, 3D FDS)
2. CSG (IFC)
3. tetrahedralisation (TEN)

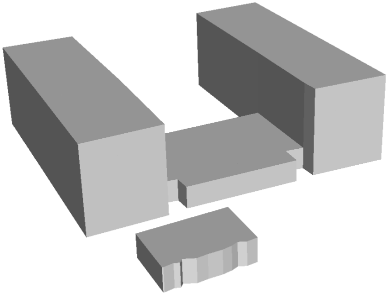


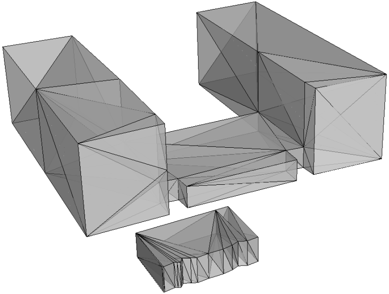
Our options:

1. *b-rep* (CityGML, 3D FDS)
2. CSG (IFC)
3. tetrahedralisation (TEN)



Representing and storing 3D topography





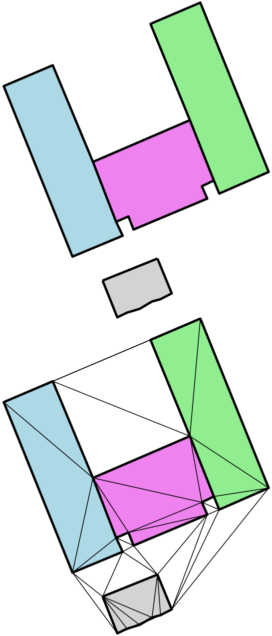
Our options:

1. *b-rep* (CityGML, 3D FDS)
2. CSG (IFC)
3. tetrahedralisation (TEN)

Constrained Delaunay tetrahedralisation (or TEN)

Advantages:

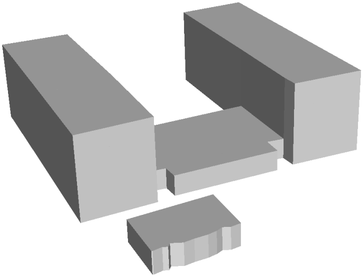
1. storage is simplified
2. spatial analysis is efficient
3. features can be represented
4. robust implementation
5. spatial relations between unconnected features explicitly stored

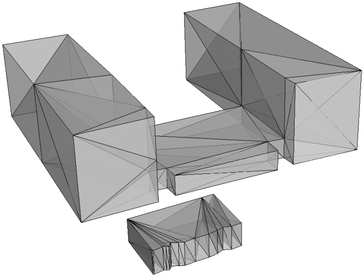


Constrained Delaunay tetrahedralisation (or TEN)

Advantages:

1. storage is simplified
2. spatial analysis is efficient
3. features can be represented
4. robust implementation
5. spatial relations between unconnected features explicitly stored

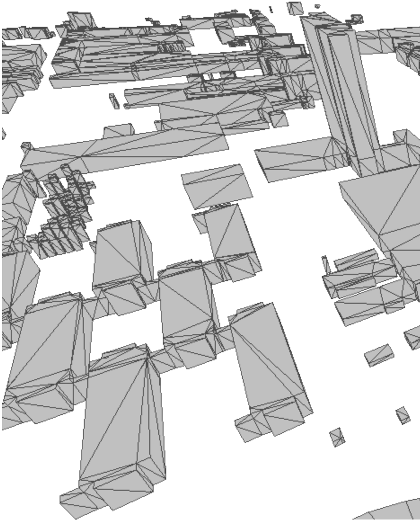




Constrained Delaunay tetrahedralisation (or TEN)

Advantages:

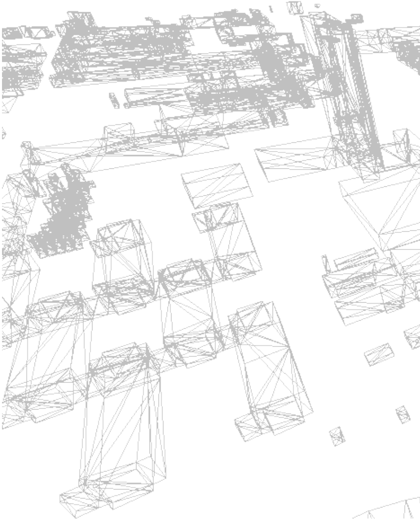
1. storage is simplified
2. spatial analysis is efficient
3. features can be represented
4. robust implementation
5. spatial relations between unconnected features explicitly stored



Constrained Delaunay tetrahedralisation (or TEN)

Advantages:

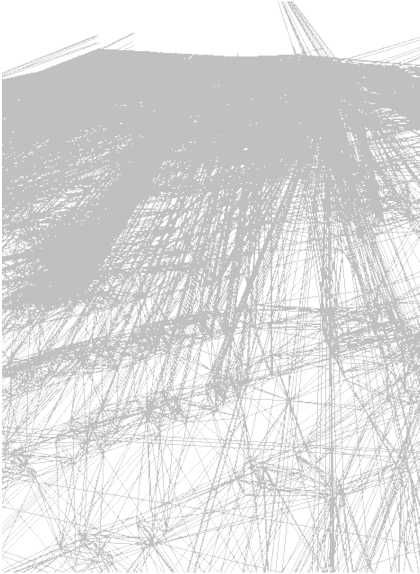
1. storage is simplified
2. spatial analysis is efficient
3. features can be represented
4. robust implementation
5. spatial relations between unconnected features explicitly stored



Constrained Delaunay tetrahedralisation (or TEN)

Advantages:

1. storage is simplified
2. spatial analysis is efficient
3. features can be represented
4. robust implementation
5. spatial relations between unconnected features explicitly stored



But...

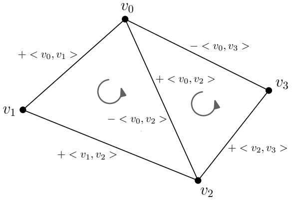
"An additional disadvantage of TEN is its much larger database size compared with other representations."

– S. Zlatanova *et al.* (2004)

Penninga (2008): Efficient storage is possible

Only vertices and tetrahedra are stored:

- akin to Simple Features
- 4 IDs per tetrahedron
- only 20% more storage than Oracle Spatial*

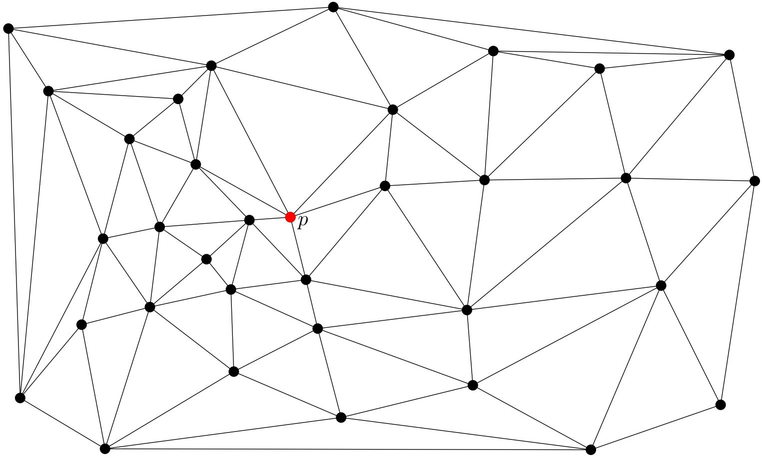


$$\partial C_2 = \partial S_{21} + \partial S_{22} = (< v_1, v_2 > - < v_0, v_2 > + < v_0, v_1 >) + (< v_2, v_3 > - < v_0, v_3 > + < v_0, v_2 >) = < v_1, v_2 > + < v_0, v_1 > + < v_2, v_3 > - < v_0, v_3 >$$

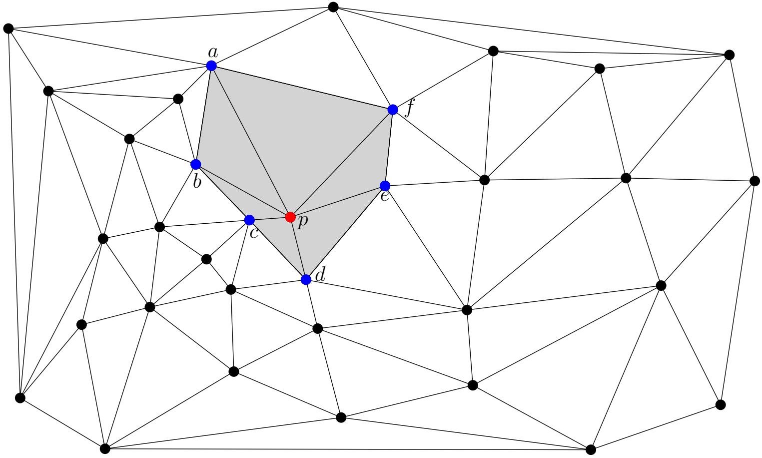
But...

- structure *not* topological
- spatial index should be added (e.g. R-tree)

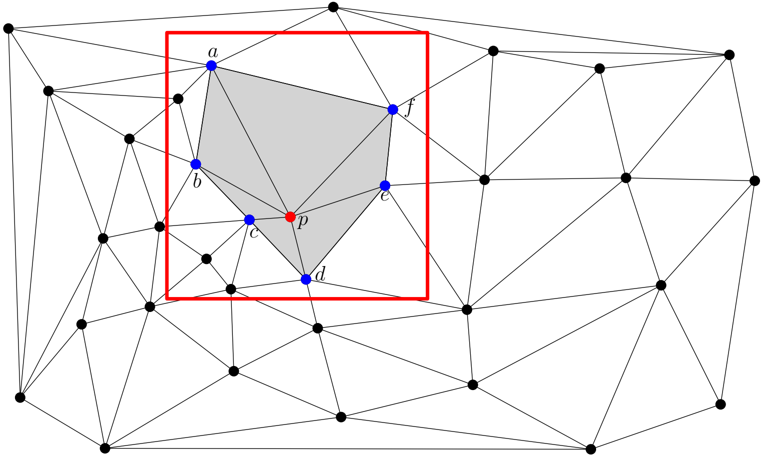
Our approach: storing *stars* in a 2D triangulation



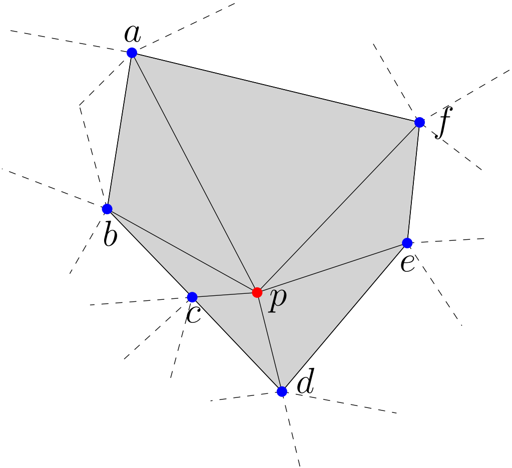
Our approach: storing *stars* in a 2D triangulation



Our approach: storing *stars* in a 2D triangulation

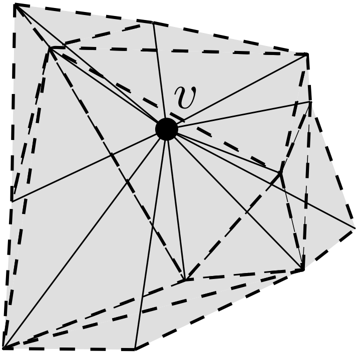


Our approach: storing *stars* in a 2D triangulation

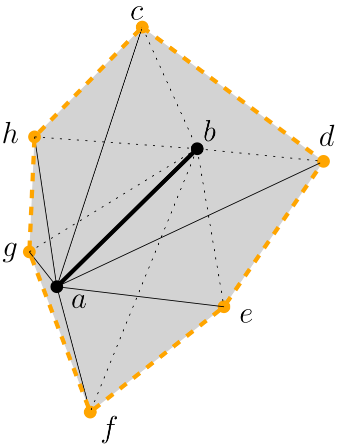


$$star(p) = abcdef$$

A *star* in 3D in a tetrahedralisation

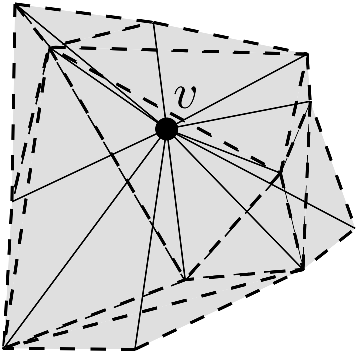


star for a vertex

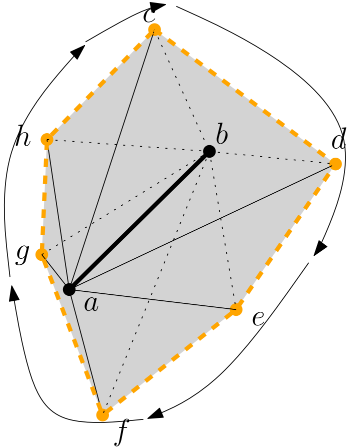


star for an edge

A *star* in 3D in a tetrahedralisation



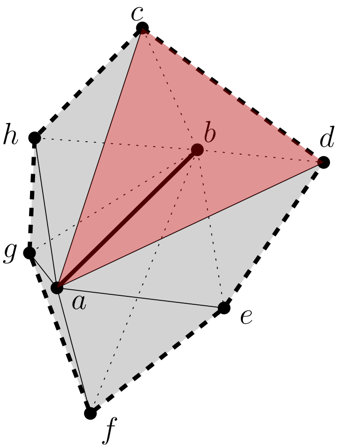
star for a vertex



star for an edge

The *star* of an edge in 3D

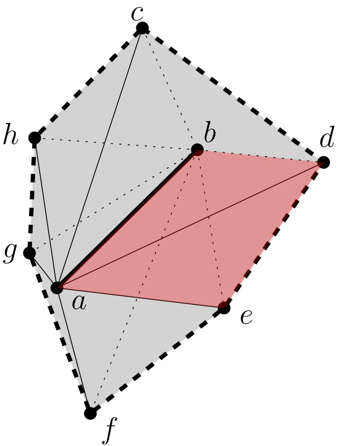
$star(ab) = cdefgh$



- $abcd$ = one tetra

The *star* of an edge in 3D

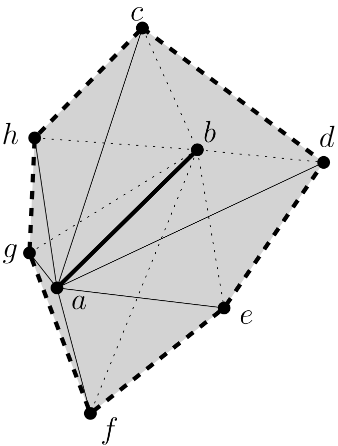
$star(ab) = cdefgh$



- $abcd$ = one tetra
- $abde$ = another tetra

The *star* of an edge in 3D

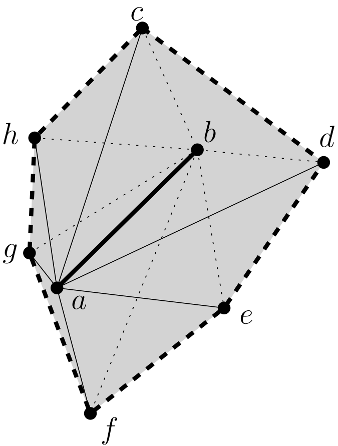
$\boxed{star(ab) = cdefgh}$



- *abcd* = one tetra
- *abde* = another tetra
- *abcd* and *abde* are adjacent

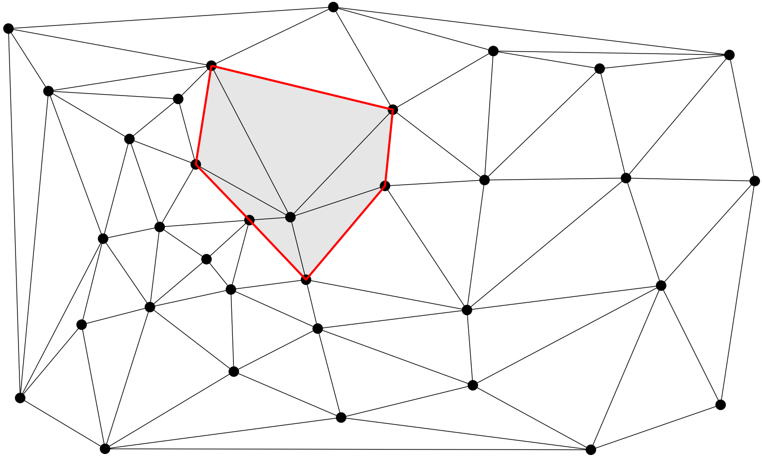
The *star* of an edge in 3D

$\boxed{star(ab) = cdefgh}$

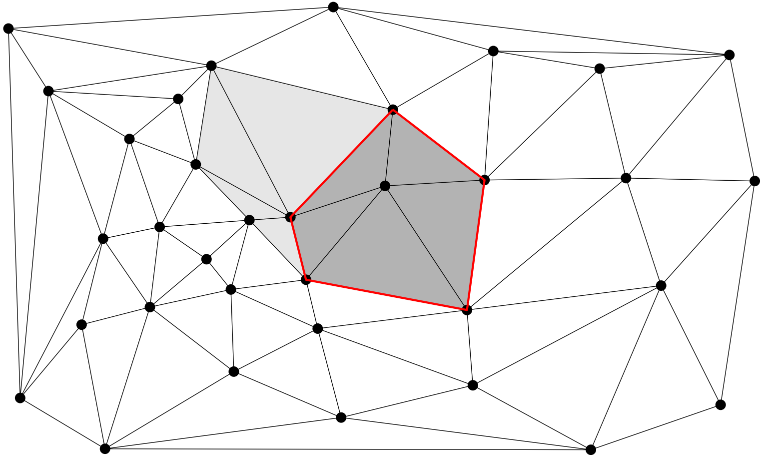


- *abcd* = one tetra
- *abde* = another tetra
- *abcd* and *abde* are adjacent
- each tetra is in 6 edges

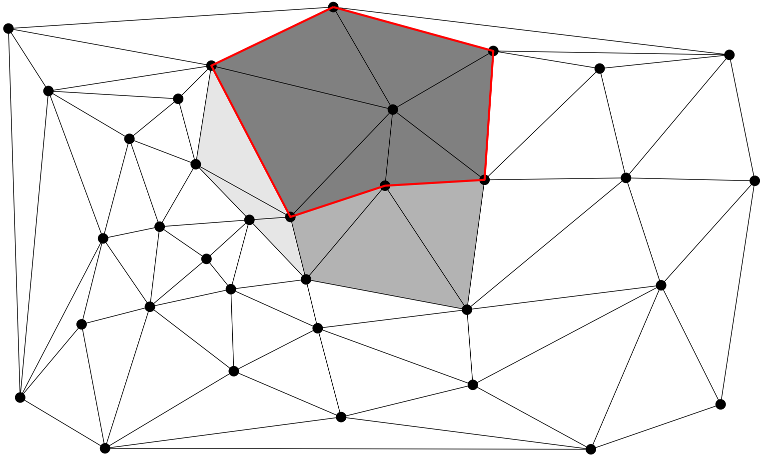
Every star is stored → implicit triangles



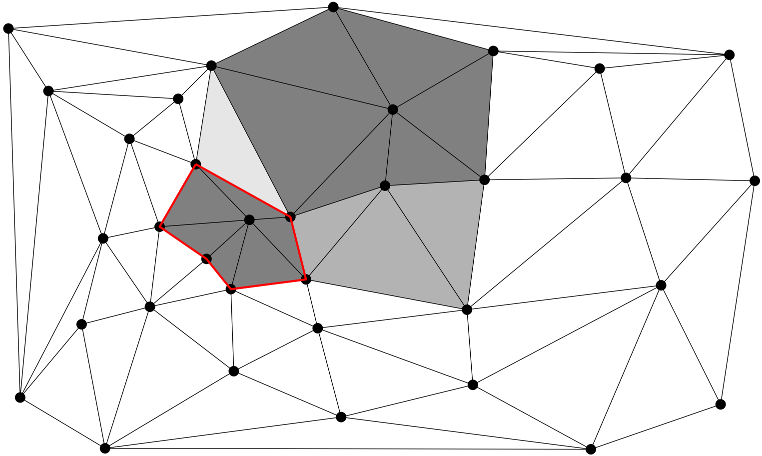
Every star is stored → implicit triangles



Every star is stored → implicit triangles

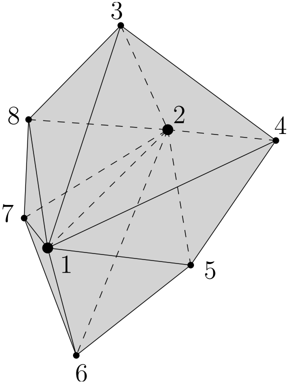


Every star is stored → implicit triangles



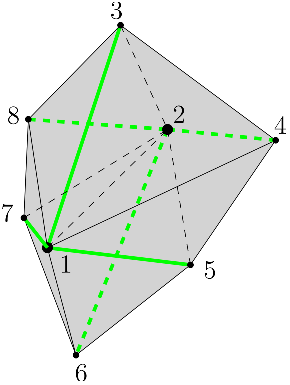
Compression = storing only *representative* edges

- $|E| \cong (7/6)|T|$
- using smart idea of Blandford *et al.* (2005)
- representative edge = both vertex labels are either odd or even
- half of edges will be stored
- each triangle and each tetra must be in at least 1 star
- only 3 labels per tetra on average—Penninga (2008) needs 4



Compression = storing only *representative* edges

- $|E| \cong (7/6)|T|$
- using smart idea of Blandford *et al.* (2005)
- representative edge = both vertex labels are either odd or even
- half of edges will be stored
- each triangle and each tetra must be in at least 1 star
- only 3 labels per tetra on average—Penninga (2008) needs 4



DBMS storage = easy and efficient

Vertex table

id	x	y	z
1	5.0	2.5	6.0
2	5.0	11.5	6.0
3	5.0	6.0	12.0
...	...	...	...
8	1.0	6.0	8.0

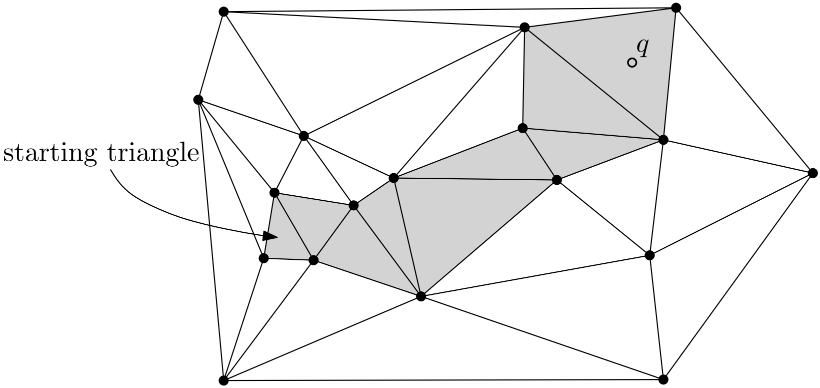
Representative edge table

start	end	star[]
2	4	$\{\emptyset, 3, 1, 5\}$
5	7	$\{6, 1, 2\}$
1	7	$\{\emptyset, 8, 2, 5, 6\}$
1	5	$\{\emptyset, 6, 7, 2, 4\}$
...	...	...
1	3	$\{\emptyset, 4, 2, 8\}$

PostgreSQL table

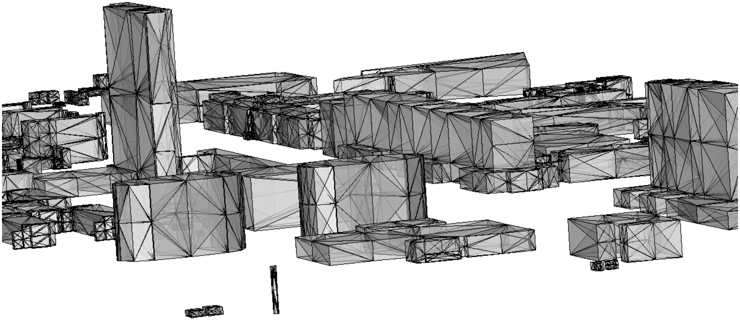
```
-- Vertex table
CREATE TABLE pgtet_vertex (
    gid bigint,
    x numeric,
    y numeric,
    z numeric
);
ALTER TABLE pgtet_vertex ADD PRIMARY KEY (gid);

-- Edge table
CREATE TABLE pgtet_edge (
    start bigint,
    end bigint,
    link bigint[] -- array of integers
);
ALTER TABLE pgtet_edge ADD PRIMARY KEY (from_gid, to_gid);
```

No auxiliary R-tree index needed: “walking” is used



(Can be made efficient [MSZ99])

Experiments with the TU Delft campus dataset



input 3D model		CDT				star
vertices	constraints	vertices	edges	triangles	tetrahedra	representative edge
5 978	3 982	6 938	56 291	95 420	47 707	25 697

Experiments with the TU Delft campus dataset

Some facts and statistics:

- 370 solids (building's footprints extruded)
- 1000 vertices added by the tetrahedralisation
- 20% compacter than Penninga's (or than Simple Features)
- average size of a star = 4.9 (min = 3; max = 28)

input 3D model		CDT				star
vertices	constraints	vertices	edges	triangles	tetrahedra	representative edge
5 978	3 982	6 938	56 291	95 420	47 707	25 697

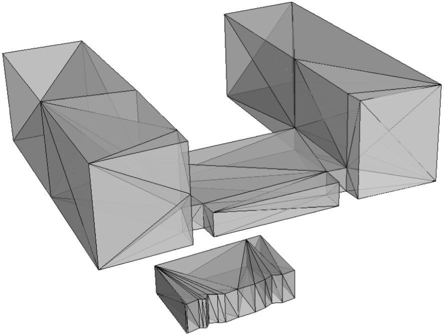
Thanks for your attention

Hugo Ledoux

h.ledoux@tudelft.nl

Martijn Meijers

b.m.meijers@tudelft.nl



References

Daniel K. Blandford, Guy E. Blelloch, David E. Cardoze, and Clemens Kadow.
Compact representations of simplicial meshes in two and three dimensions.
International Journal of Computational Geometry and Applications, 15(1):3–24, 2005.

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.

Friso Penninga.
3D Topography: A Simplicial Complex-based Solution in a Spatial DBMS.
PhD thesis, Delft University of Technology, Delft, the Netherlands, 2008.

Sisi Zlatanova, Alias Abdul Rahman, and Wenzhong Shi.
Topological models and frameworks for 3D spatial objects.
Computers & Geosciences, 30(4):419–428, 2004.