3D Topography

A simplicial Complex-based Solution in a Spatial DBMS

Friso Penninga

GIS-t lunch meeting January 18, 2008

TU Delft

Presentation outline

- Introduction
- Previous presentation (01-09-06)
  - Characteristics simplicial complex-based approach
  - Simplicial complexes applied to 3D Topography
  - Implementation details
- Update operations (feature insertion)
- Test data sets
  - Tetrahedronisation of models
  - Storage requirements, comparison to Oracle 11g polyhedrons
- Future research & conclusions

TU Delft

Introduction (1/2)

PhD research within RGI-011

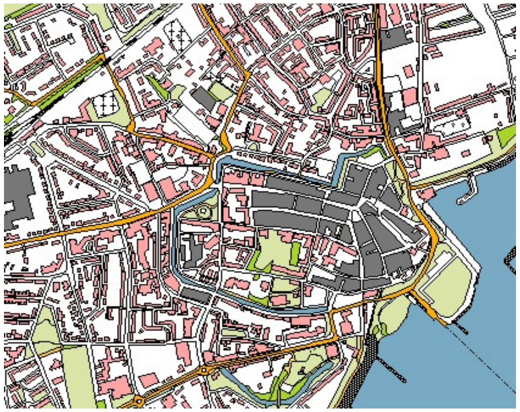
Objective RGI-011:

Enforcing a major break-through in the application of 3D Topography in corporate ICT environments due to structural embedding of 3D methods and techniques.

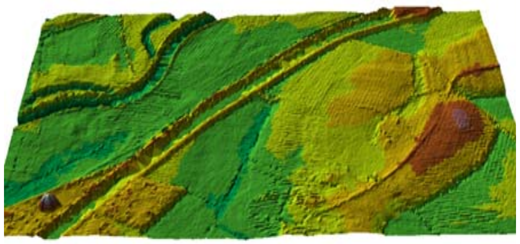
Research question my PhD:

How can a 3D topographic representation be realised in a feature-based triangular data model?

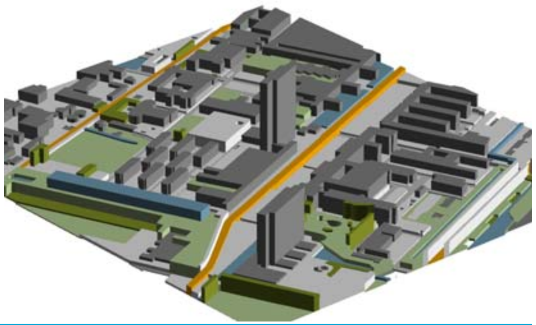
Introduction (2/2)



+



=



Two-step approach:

- How to develop a conceptual model that describes the real world phenomena (the topographic features)?
- How to implement this conceptual model, i.e. how to develop a suitable DBMS data structure?

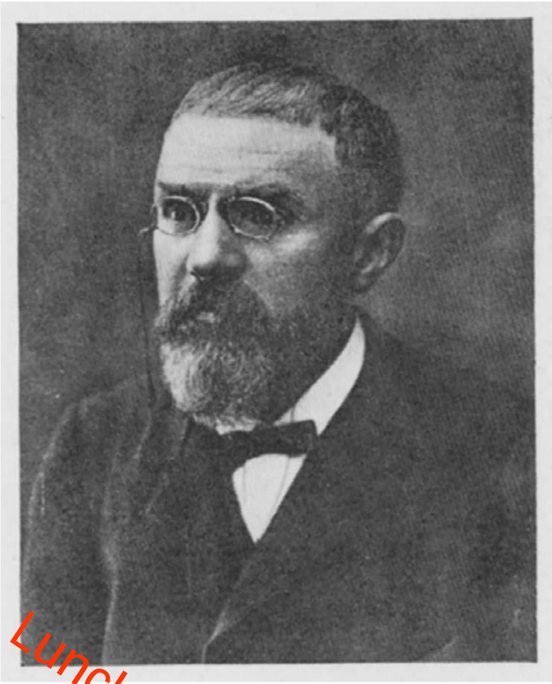
Focus:

storage, analysis, validation

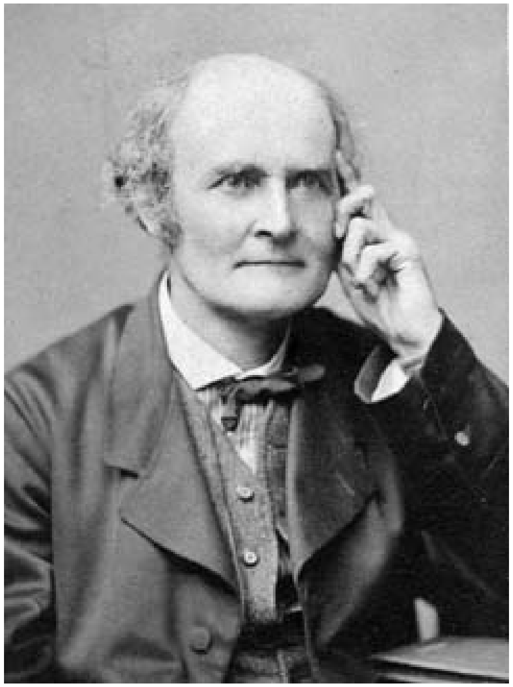
Presentation outline

- Introduction
- Previous presentation (01-09-06)
  - Characteristics simplicial complex-based approach
  - Simplicial complexes applied to 3D Topography
  - Implementation details
- Update operations (feature insertion)
- Test data sets
  - Tetrahedronisation of models
  - Storage requirements, comparison to Oracle 11g polyhedrons
- Future research & conclusions

Previous presentation



+



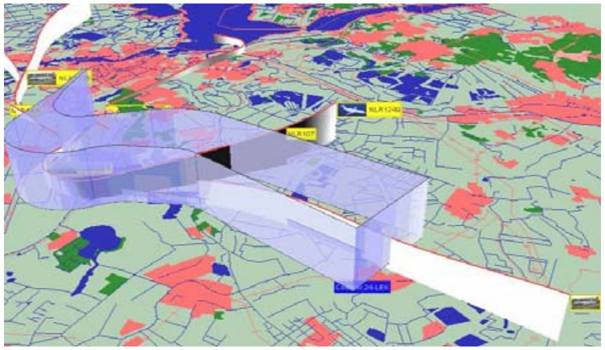
+

= 3D GIS?

TU Delft

Previous presentation

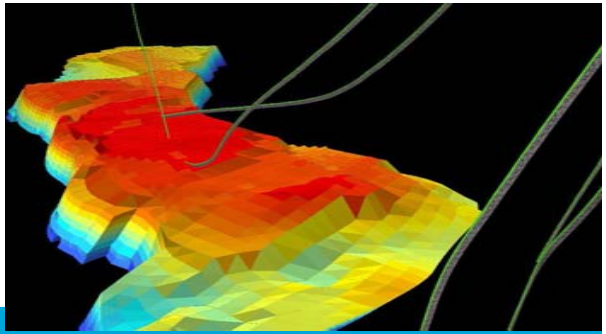
Characteristics sc-based approach (1/3)



Characteristic 1: Full decomposition of space

Two fundamental observations (Cosit'05 paper):

- ISO19101: a feature is an 'abstraction of real world phenomena'. These real world phenomena have by definition a volume
- Real world can be considered to be a volume partition (analogous to a planar partition: a set of non-overlapping volumes that form a closed modelled space)



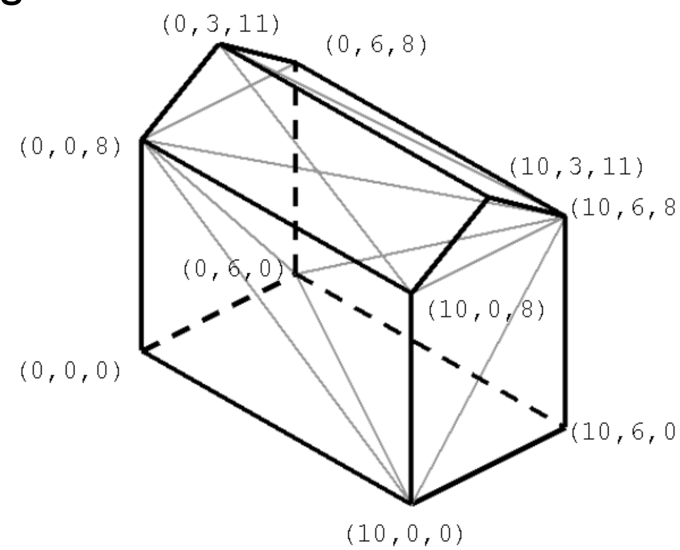
Result: explicit inclusion of earth and air

TU Delft

Previous presentation
Characteristics sc-based approach (2/3)

Characteristic 2: constrained TEN

object boundaries represented by constraints

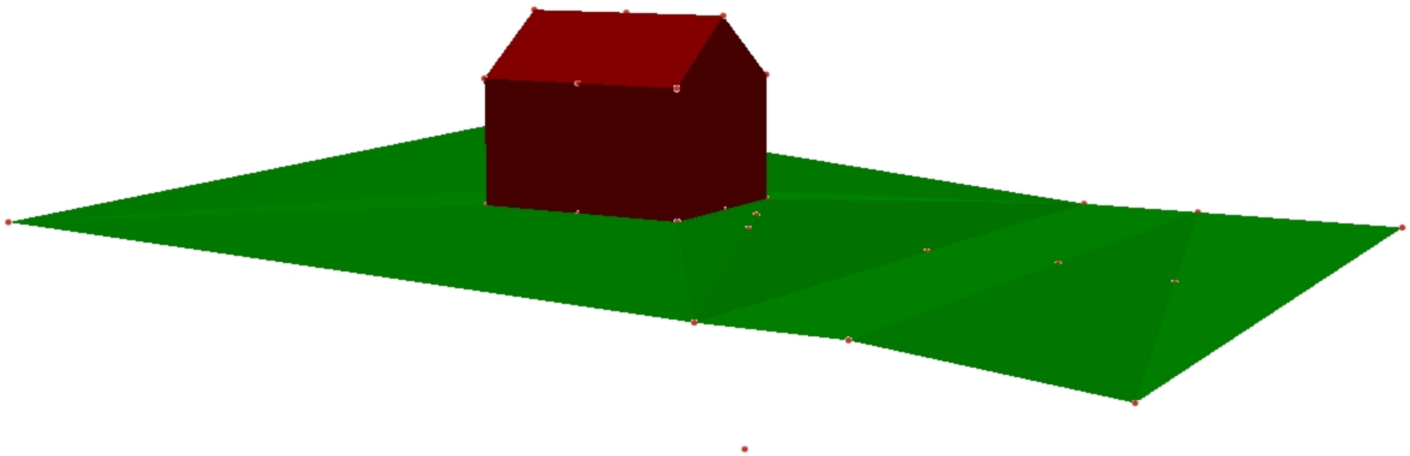


Advantages of TEN:

- Well defined: a n-simplex is bounded by n + 1 (n - 1)-simplexes.
- Flatness of faces: every face can be described by three points
- A n-simplex is convex (which simplifies amongst others point-in-polygon tests)

Previous presentation

Example: small data set



Previous presentation

Characteristics sc-based approach (3/3)

Characteristic 3: based on Poincaré simplicial homology

solid mathematical foundation (SDH'06 paper):

Simplex S_n defined by ($n+1$) vertices: $S_n = <v_0,\ldots v_n>$

The boundary ∂ of simplex S_n is defined as sum of *(n-1)* dimensional simplexes (note that 'hat' means skip the node):

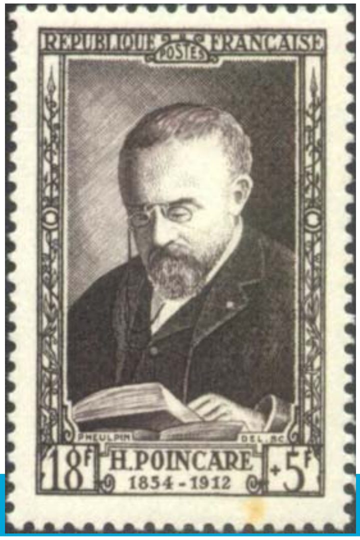
$$\partial S_n = \sum_{i=0}^{n} (-1)^i < v_0,\ldots,\hat{v}_i,\ldots,v_n >$$

remark: sum has n+1 terms

$S_1 = <v_0, v_1>$ $\quad\quad$ $\partial S_1 = <v_1> - <v_0>$

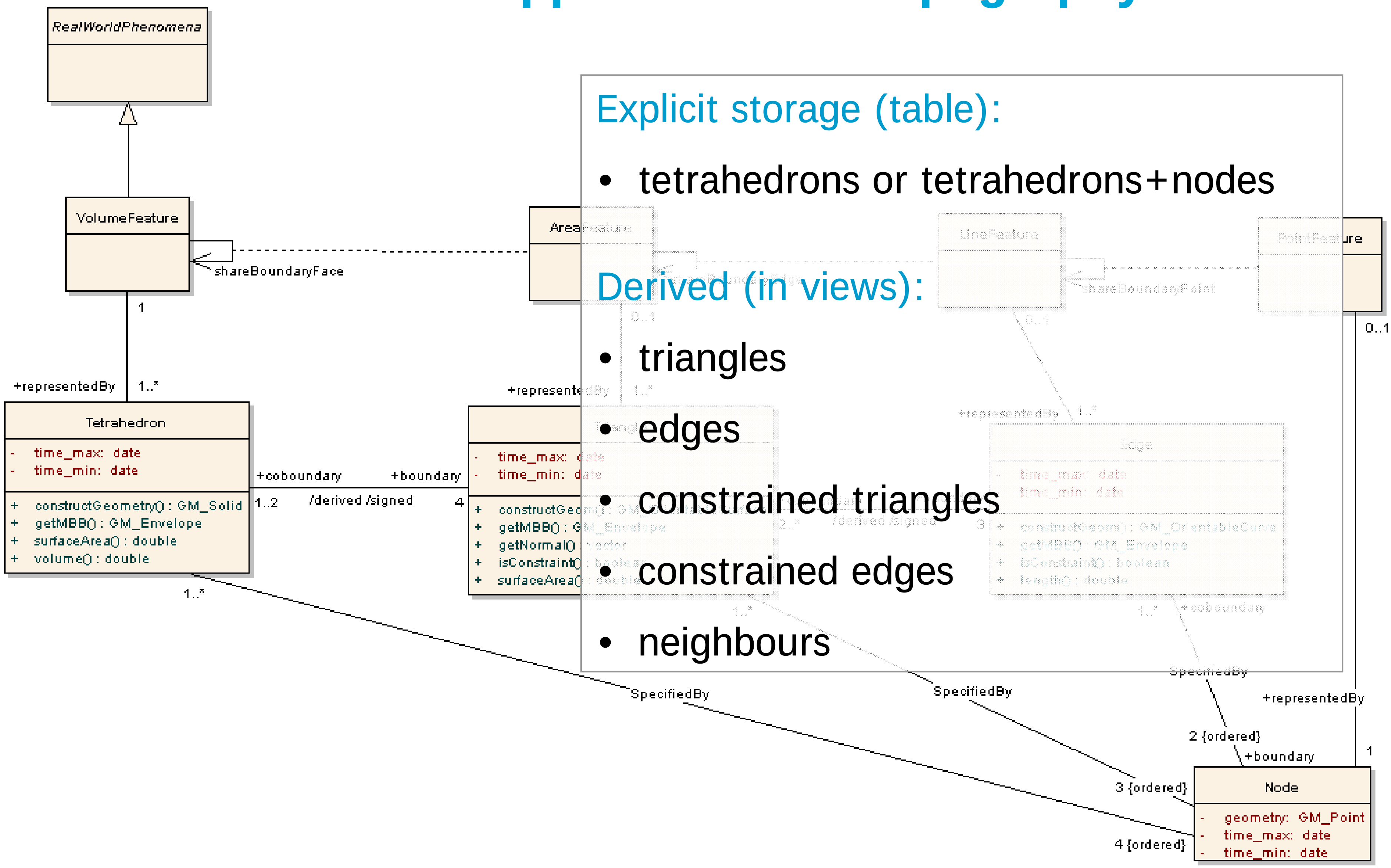
$S_2 = <v_0, v_1, v_2>$ $\quad\quad$ $\partial S_2 = <v_1, v_2> - <v_0, v_2> + <v_0, v_1>$

$S_3 = <v_0, v_1, v_2, v_3>$ $\quad\quad$ $\partial S_3 = <v_1, v_2, v_3> - <v_0, v_2, v_3> + <v_0, v_1, v_3> - <v_0, v_1, v_2>$



Previous presentation

SC-based approach to 3D topography



Explicit storage (table):

- tetrahedrons or tetrahedrons+nodes

Derived (in views):

- triangles
- edges
- constrained triangles
- constrained edges
- neighbours

Previous presentation

Implementation details

$$\partial S_n = \sum_{i=0}^{n} (-1)^i < v_0, \ldots, \hat{v}_i, \ldots, v_n >$$

Boundary operator implemented in PL/SQL procedure

Procedure used to define views with triangles, edges, constrained triangles (object boundaries!), constrained edges, e.g.:

```
create or replace view triangle as
    select deriveboundarytriangle1(tetcode) tricode,
    tetcode fromtetcode from tetrahedron
    UNION ALL
    select deriveboundarytriangle2(tetcode) tricode,
    tetcode fromtetcode from tetrahedron
    UNION ALL
    ...
```

TUDelft

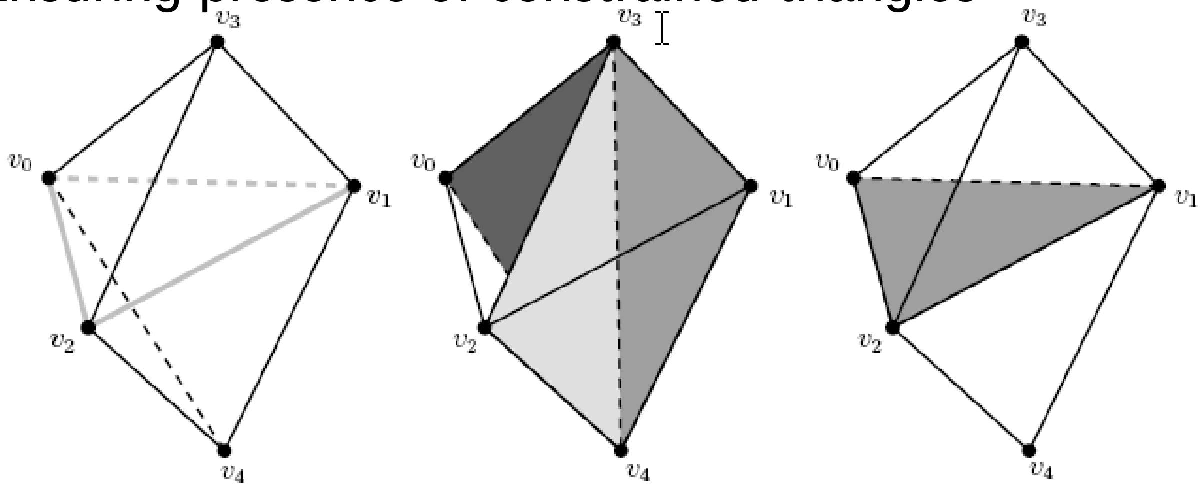
Presentation outline

- Introduction
- Previous presentation (01-09-06)
  - Characteristics simplicial complex-based approach
  - Simplicial complexes applied to 3D Topography
  - Implementation details
- Update operations (feature insertion)
- Test data sets
  - Tetrahedronisation of models
  - Storage requirements, comparison to Oracle 11g polyhedrons
- Future research & conclusions

Update operations

Four steps in feature insertion

1. Feature boundary triangulation: input for step 2
2. Inserting constrained edges with new approach (following slides)
3. Ensuring presence of constrained triangles



4. Interior modelling + reclassifying tetrahedrons

Update operations

Insertion of constrained edge in a TEN

Usual approach: insert nodes, using flipping for edge recovery

Might fail in topographic TEN:

many constraints in close proximity ⟶ flipping not always possible

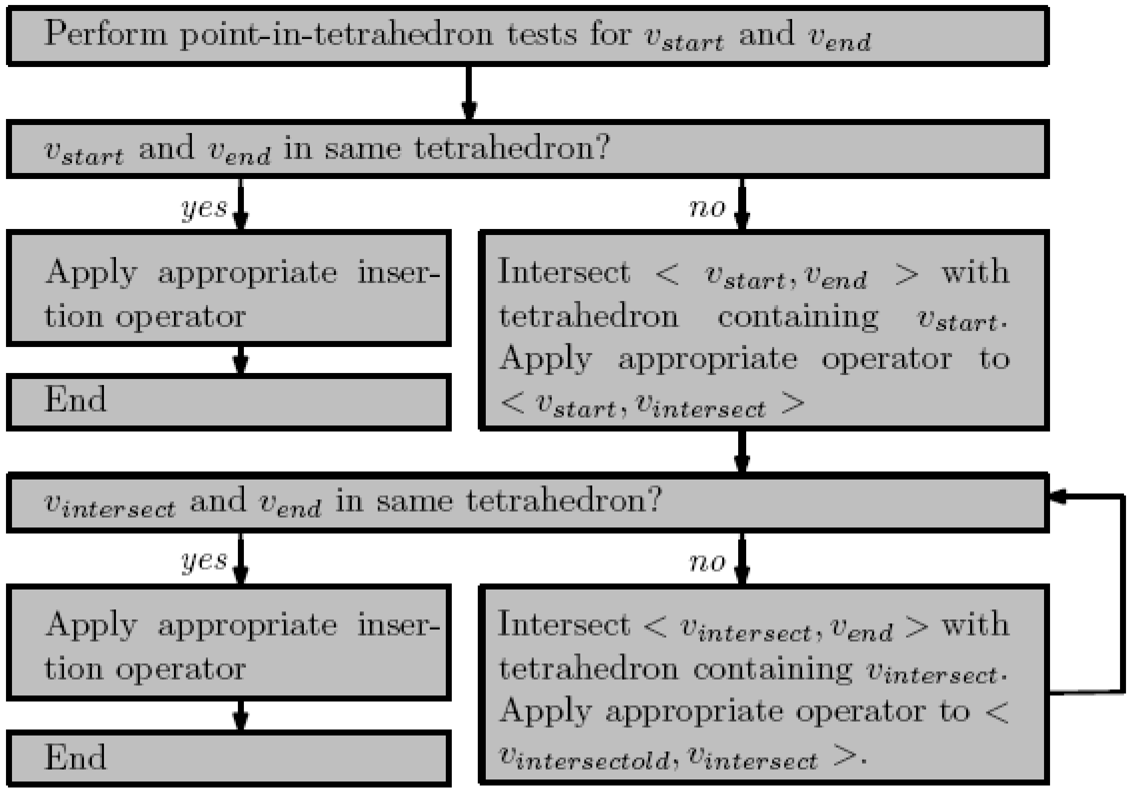
New approach: insert a complete constrained edge

Nine unique cases: *exhaustive + mutually exclusive*

Node lies on	Node	Edge	Triangle	Tetrahedron
Node	I_{00}	I_{01}	I_{02}	I_{03}
Edge	(I_{01})	I_{11}	I_{12}	I_{13}
Triangle	(I_{02})	(I_{12})	I_{22}	I_{23}
Tetrahedron	(I_{03})	(I_{13})	(I_{23})	I_{33}

Update operations

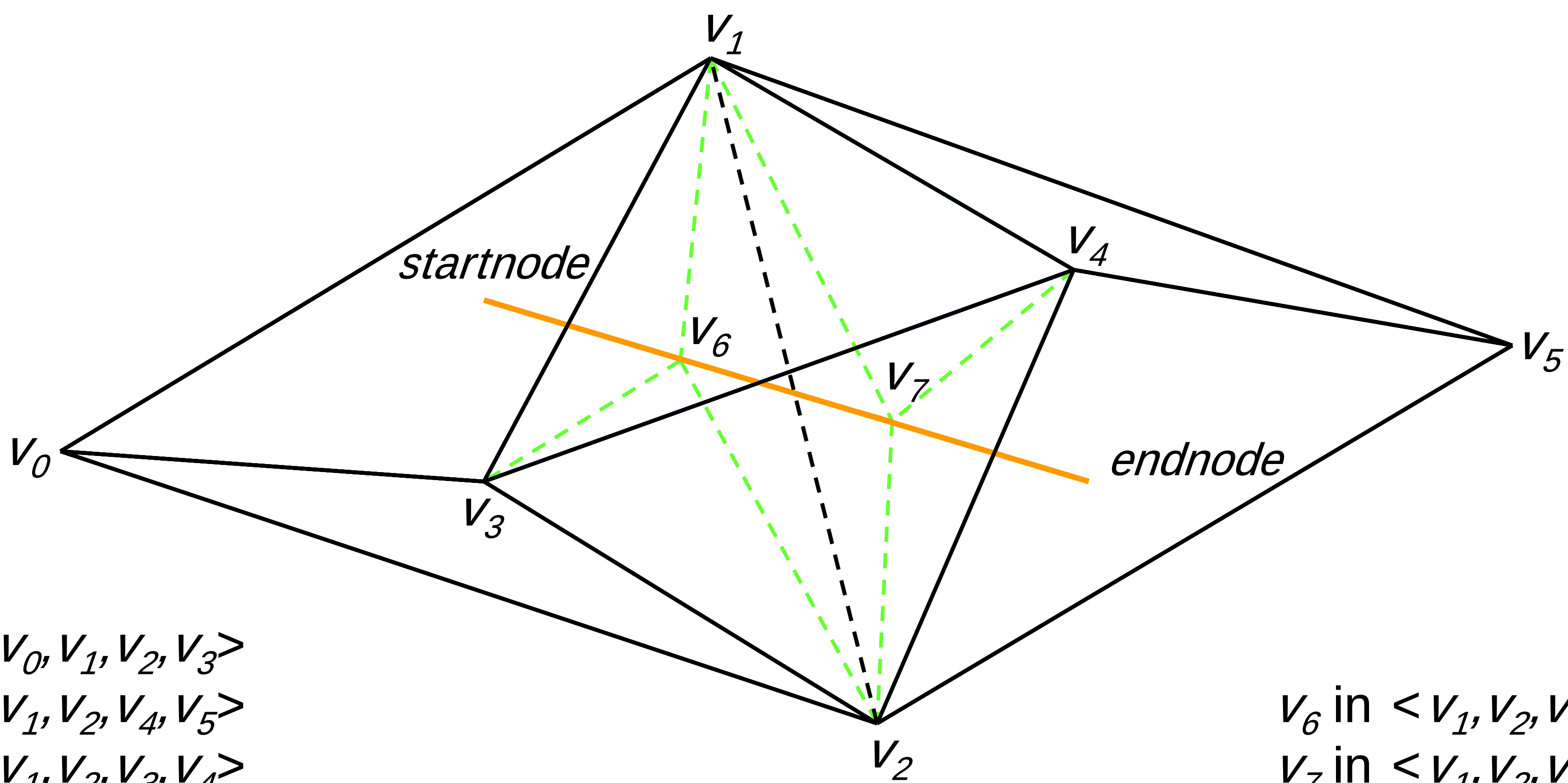
Insertion of constrained edge in a TEN



New approach: act as local as possible with minimal impact

Update operations

Example: constrained edge insertion



$<v_0, v_1, v_2, v_3>$
$<v_1, v_2, v_4, v_5>$
$<v_1, v_2, v_3, v_4>$

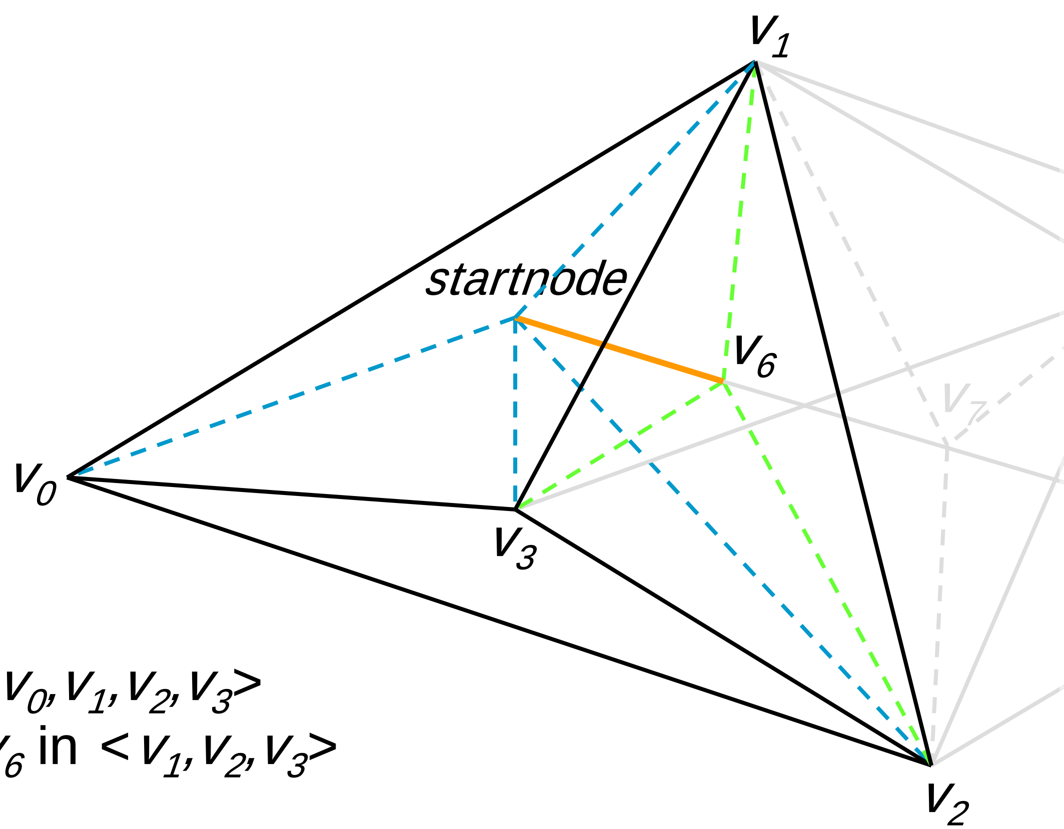
v_6 in $<v_1, v_2, v_3>$
v_7 in $<v_1, v_2, v_4>$

i.e. combination of intersection cases:
I_{23} - I_{22} - I_{23}

TU Delft

Update operations

Example: constrained edge insertion



$< v_0, v_1, v_2, v_3 >$

v_6 in $< v_1, v_2, v_3 >$

Tetrahedron split in six new tetrahedrons:

$< v_0, v_1, v_2,$ *startnode*$>$

$<$ *v0,*$v_1, v_3,$*startnode*$>$

$< v_0, v_2, v_3,$*startnode*$>$

$<$*startnode*$, v_6, v_1, v_2 >$

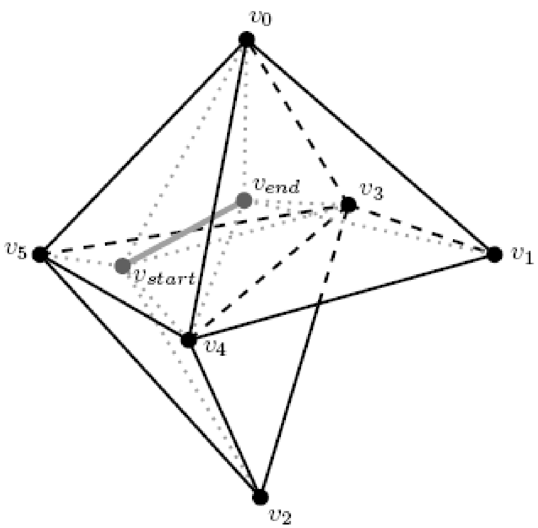
$<$*startnode*$, v_6, v_1, v_3 >$

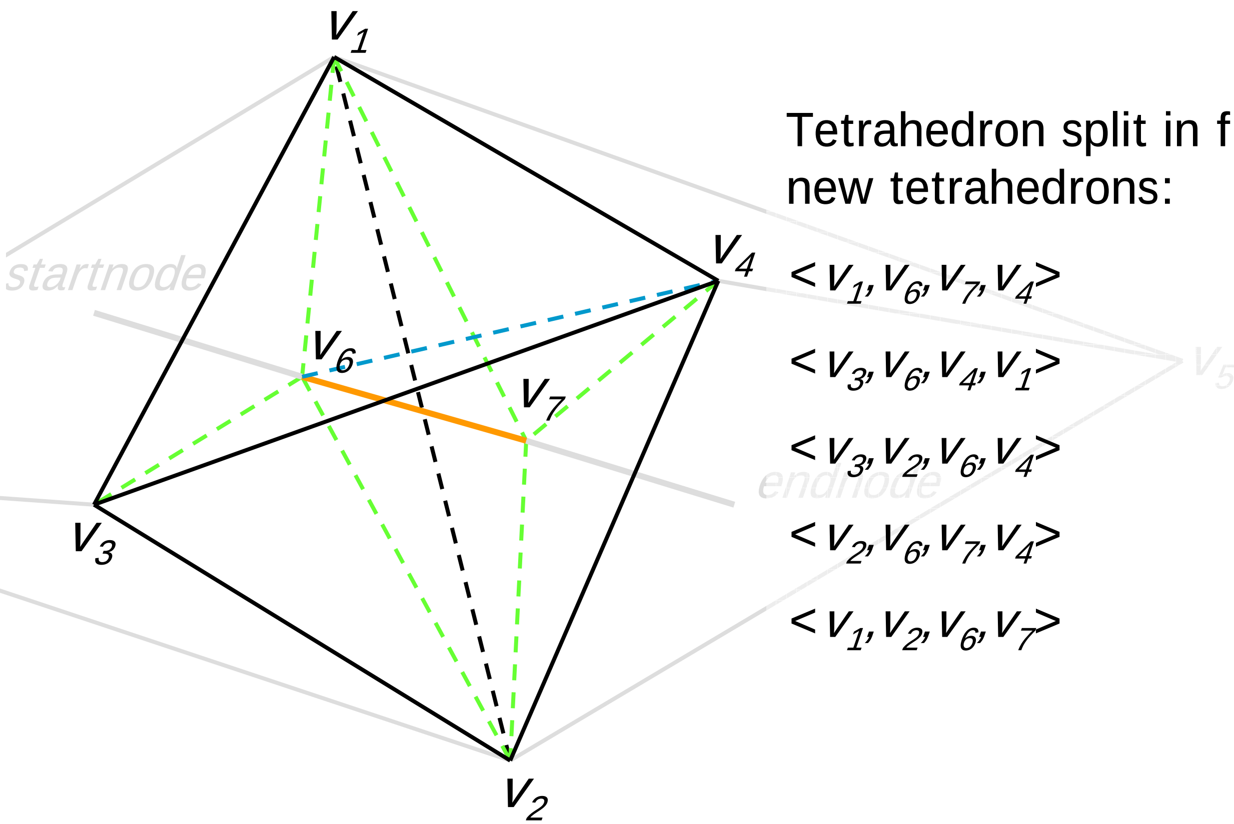
$<$*startnode*$, v_6, v_2, v_3 >$

![TU Delft]

Update operations

Example: constrained edge insertion





Tetrahedron split in five new tetrahedrons:

$<V_1, V_6, V_7, V_4>$

$<V_3, V_6, V_4, V_1>$

$<V_3, V_2, V_6, V_4>$

$<V_2, V_6, V_7, V_4>$

$<V_1, V_2, V_6, V_7>$

$<V_1, V_2, V_3, V_4>$
V_6 in $<V_1, V_2, V_3>$
V_7 in $<V_1, V_2, V_4>$

Update operations

Example: constrained edge insertion

Tetrahedron split in six new tetrahedrons:

$<v_2, v_4, endnode, v_5>$

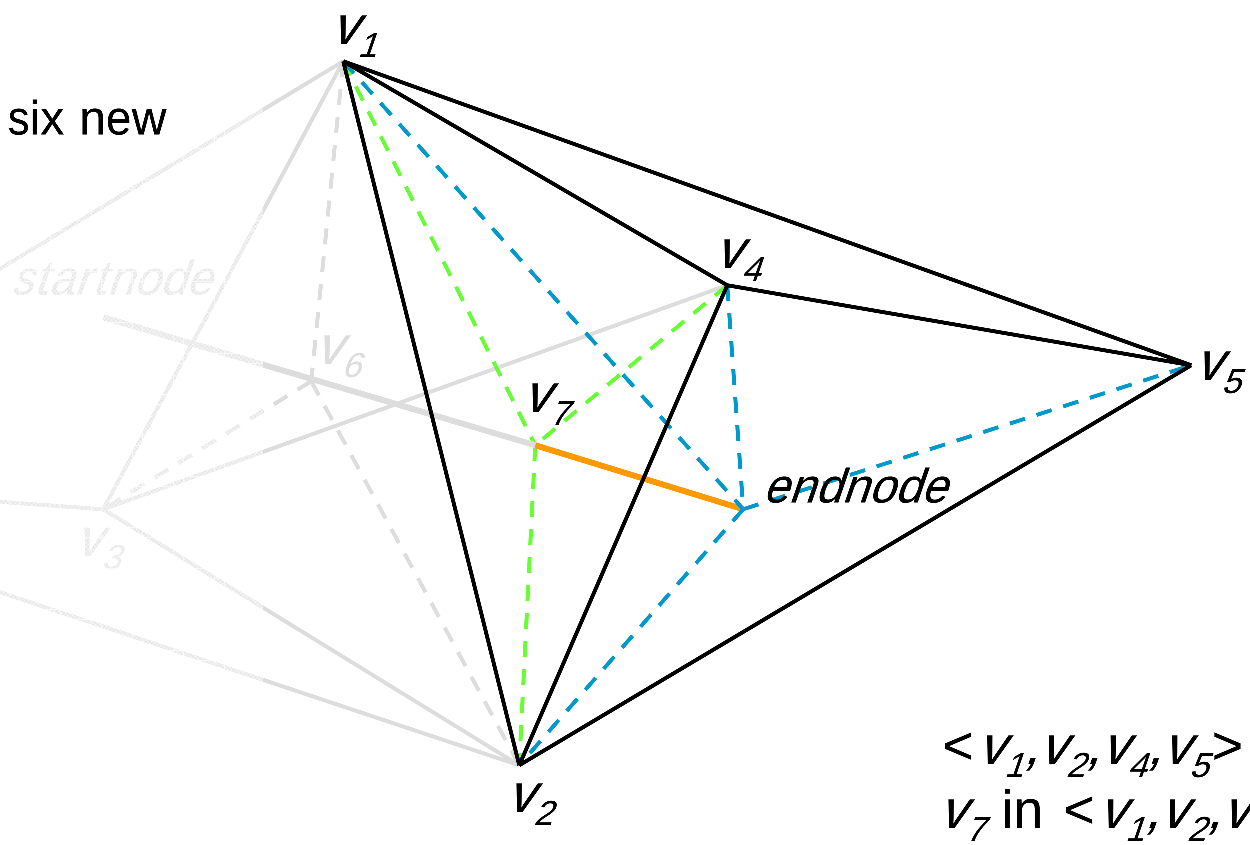
$<v_4, v_5, endnode, v_1>$

$<endnode, v_7, v_4, v_1>$

$<v_7, endnode, v_1, v_2>$

$<v_2, v_4, v_7, endnode>$

$<endnode, v_1, v_2, v_5>$



$<v_1, v_2, v_4, v_5>$

v_7 in $<v_1, v_2, v_4>$

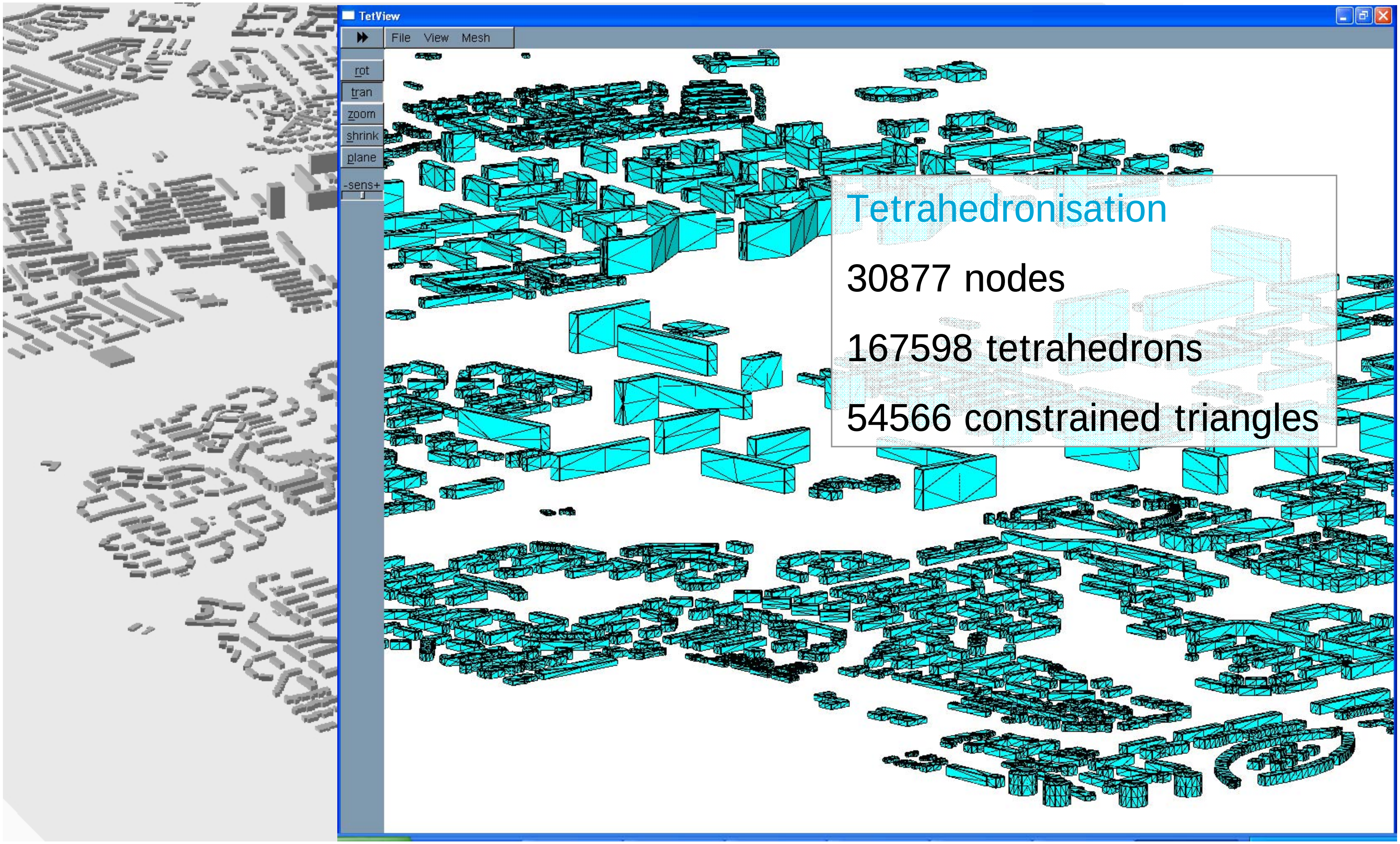
Presentation outline

- Introduction
- Previous presentation (01-09-06)
  - Characteristics simplicial complex-based approach
  - Simplicial complexes applied to 3D Topography
  - Implementation details
- Update operations (feature insertion)
- Test data sets
  - Tetrahedronisation of models
  - Storage requirements, comparison to Oracle 11g polyhedrons
- Future research & conclusions

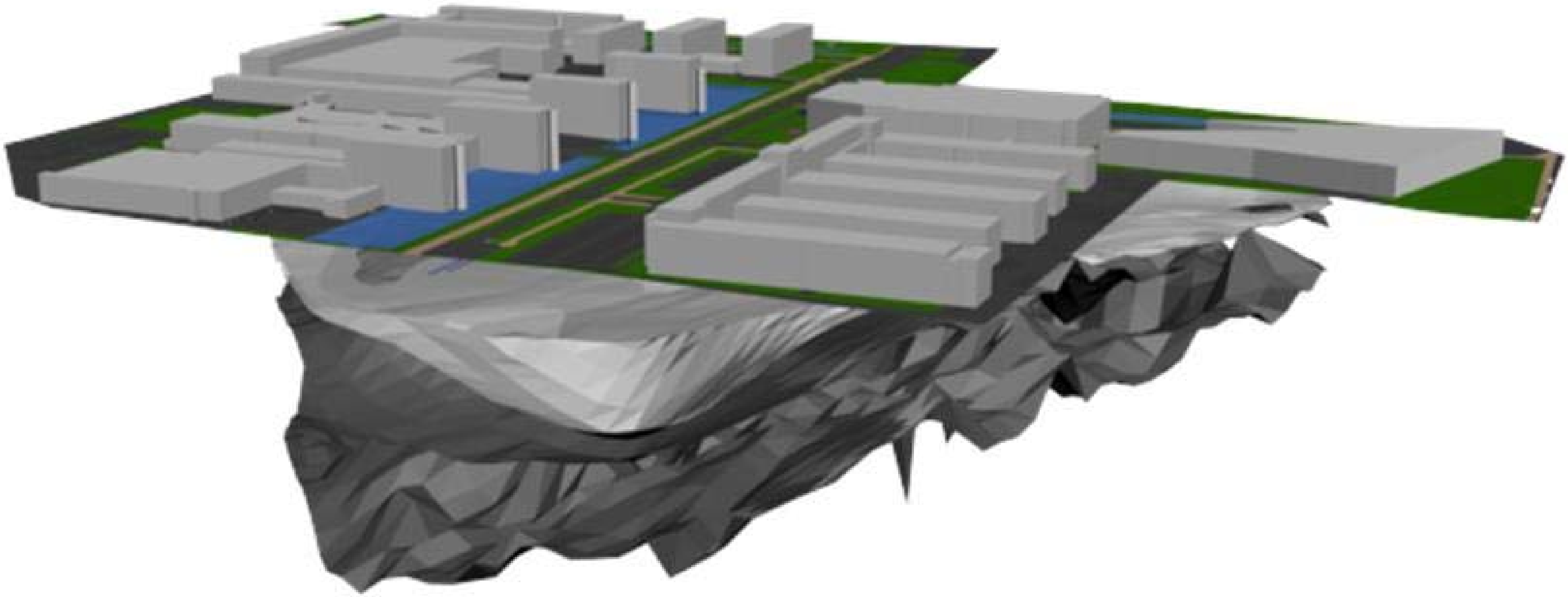
TU Delft

Test data sets (1/2)

Rotterdam data set

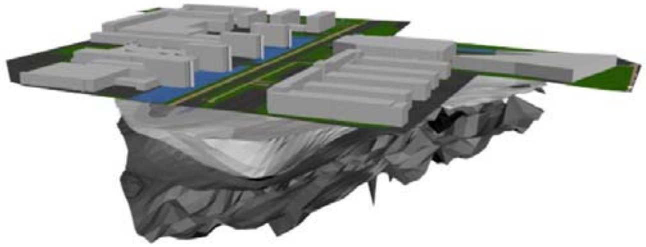


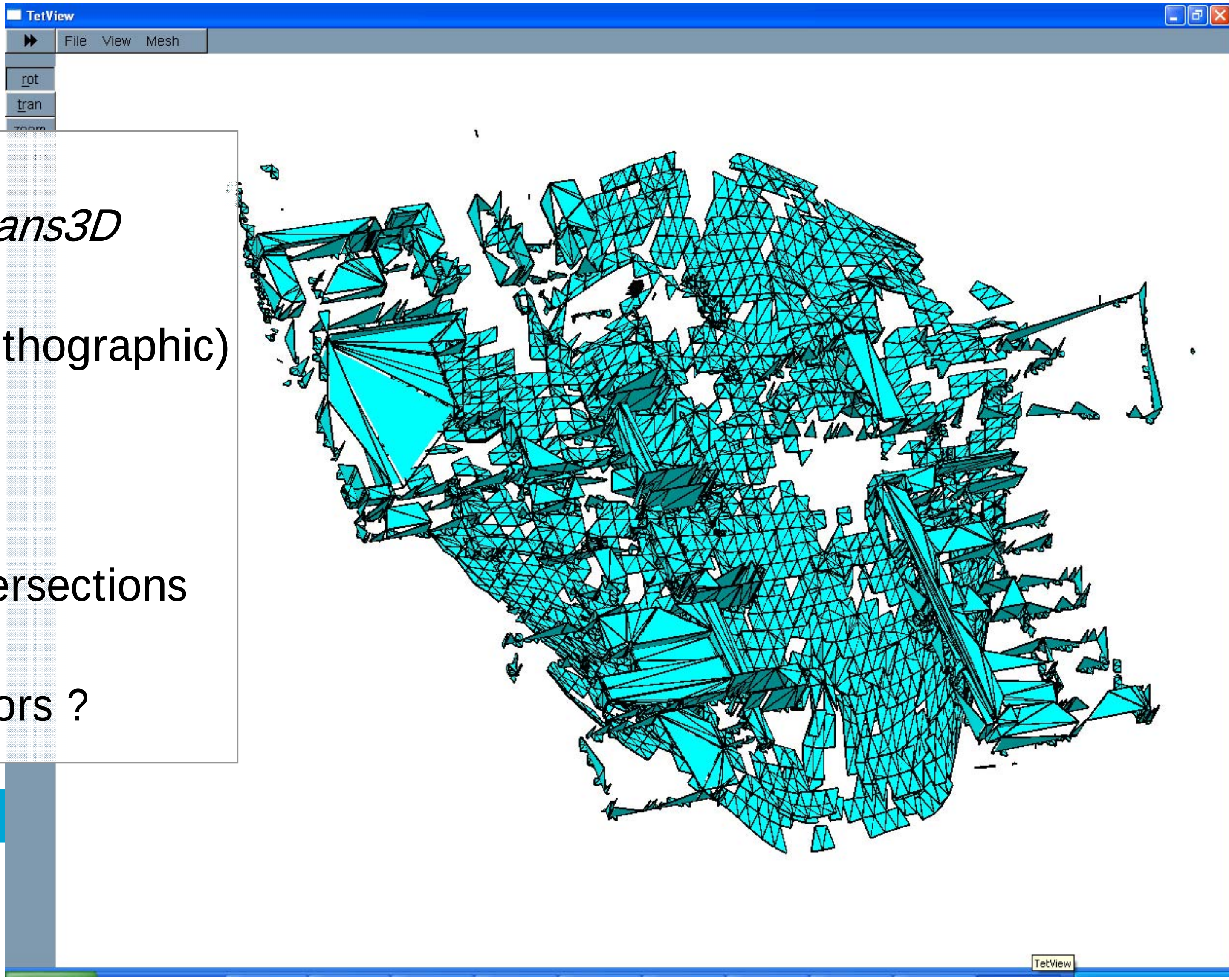
Test data sets (2/2)
Campus data set



Test data sets (2/2)

Campus data set





DXF

↓ *AccuTrans3D*

STL

↓ (StereoLithographic)

TetGen

↓

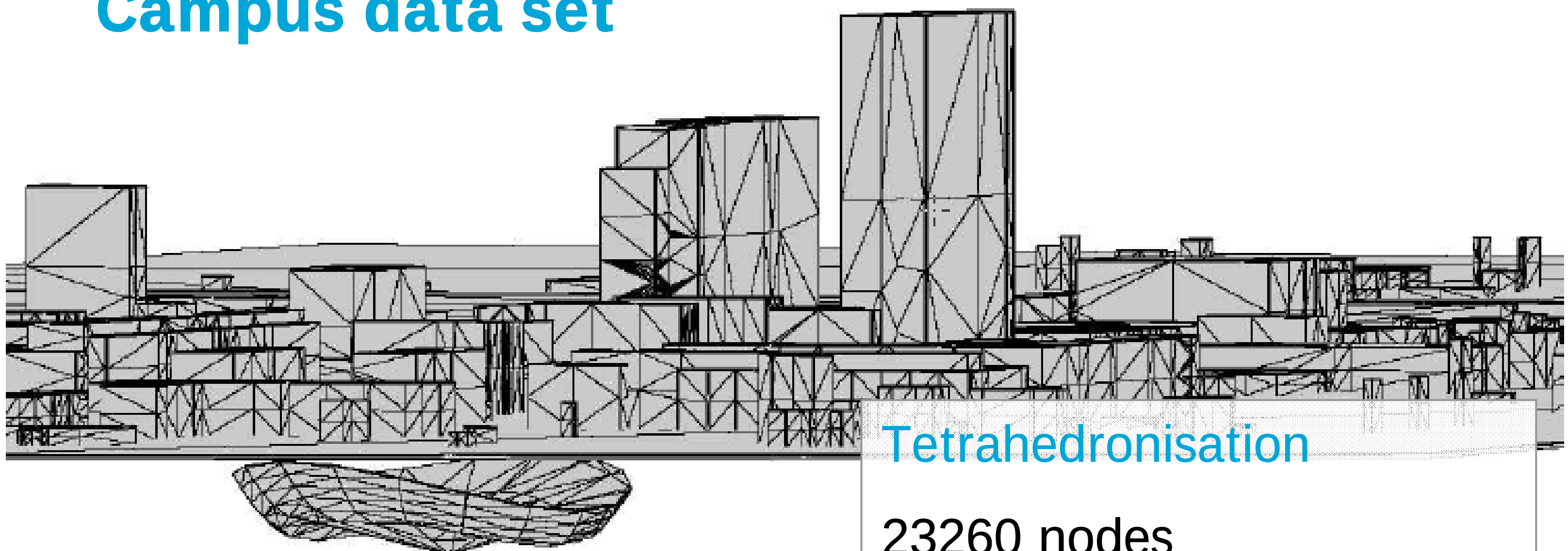
errors...

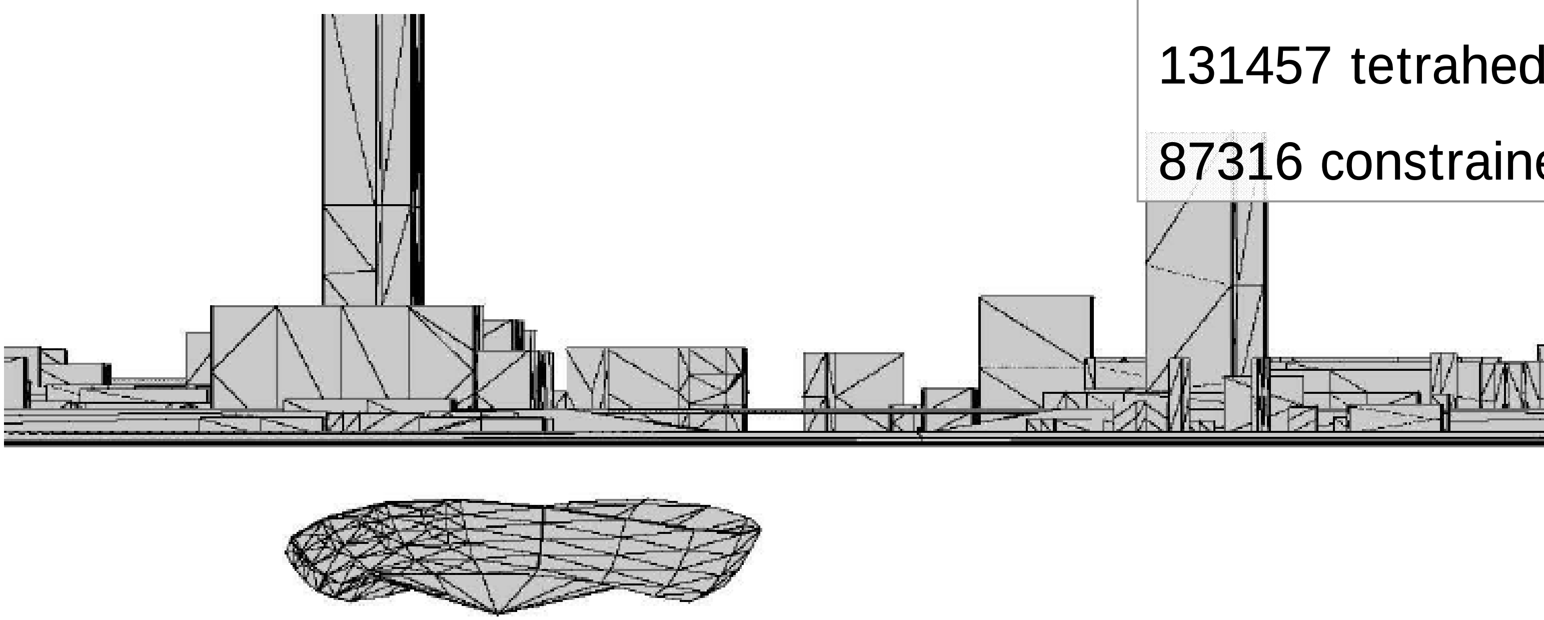
>50.000 intersections

round off errors ?

Test data sets (2/2)

Campus data set





Tetrahedronisation

23260 nodes

131457 tetrahedrons

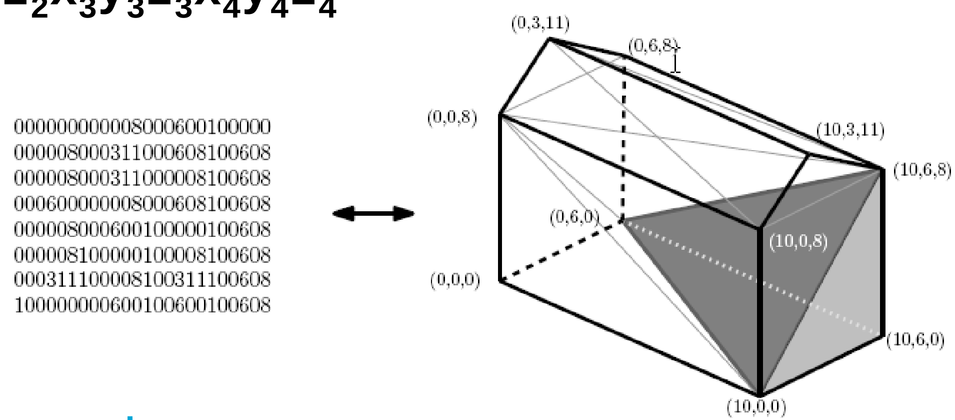
87316 constrained triangles

Data storage requirements (1/2)
Two alternative approaches

Coordinate concatenation:

describe tetrahedrons by node geometries:

$x_1y_1z_1x_2y_2z_2x_3y_3z_3x_4y_4z_4$



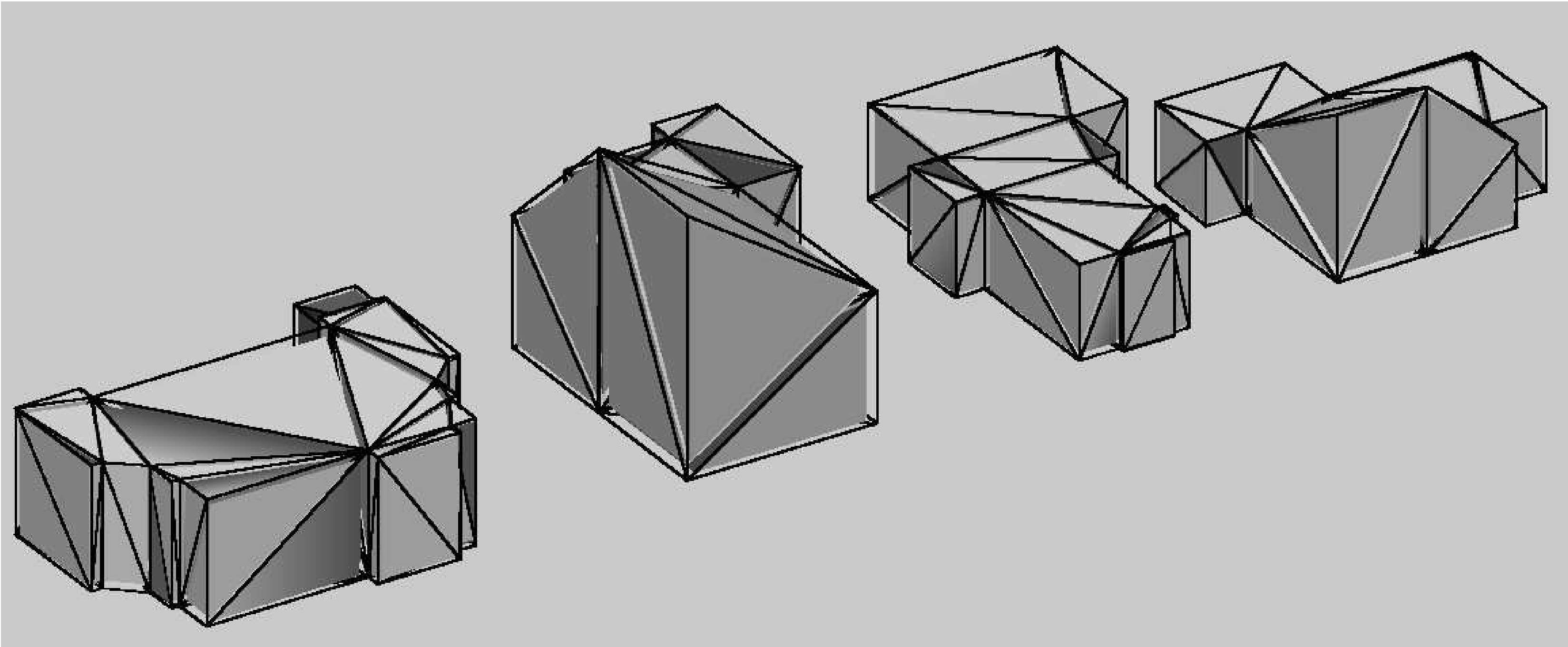
Identifier concateation:

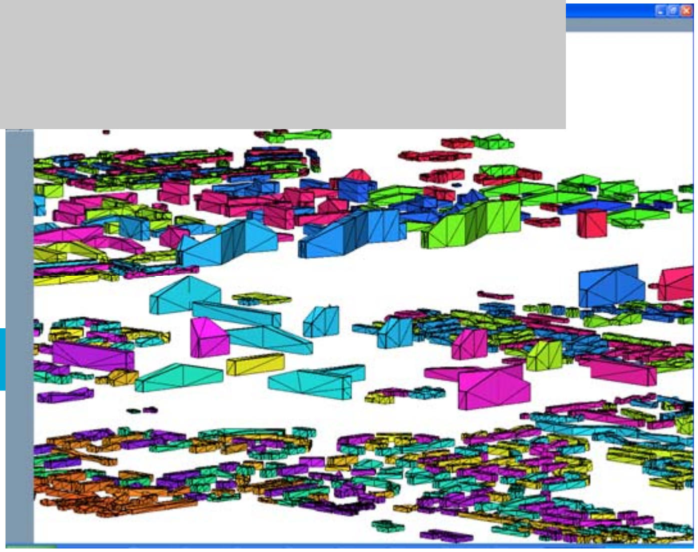
describe tetrahedrons by node id's:

$id_1id_2id_3id_4$ with $id_1{:}x_1y_1z_1$, $id_2{:}x_2y_2z_2$, etc.

Data storage requirements (2/2)

Comparing alternatives and polyhedrons





Presentation outline

- Introduction
- Previous presentation (01-09-06)
  - Characteristics simplicial complex-based approach
  - Simplicial complexes applied to 3D Topography
  - Implementation details
- Update operations (feature insertion)
- Test data sets
  - Tetrahedronisation of models
  - Storage requirements, comparison to Oracle 11g polyhedrons
- Future research & conclusions

Future research

Spatial clustering and indexing

Basic idea:

Why add a meaningless unique id to a node, when its geometry is already unique?

1 Bitwise interleaving coordinates → Morton-like code → sorting these codes → spatial clustering

2 Use as spatial index → no addtional indexes (R-tree/quad tree)

Objective: reducing storage requirements

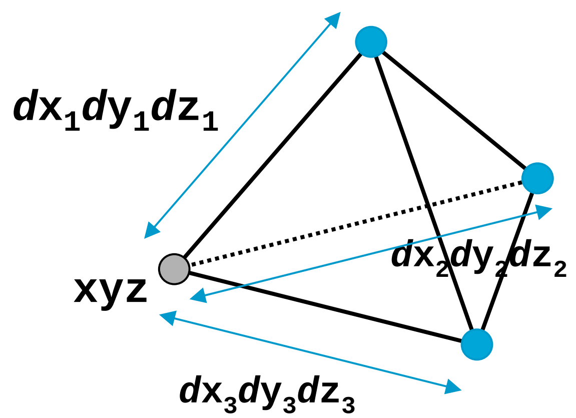
Future research

Coordinates vs. coord. differences

Four nodes of a tetrahedron will be relatively close:
only small differences in coordinates

Alternative tetrahedron description:

$\mathbf{xyz}\,dx_1dy_1dz_1dx_2dy_2dz_2dx_3dy_3dz_3$



Description is based on geometry
(so still unique) but smaller

Conclusions

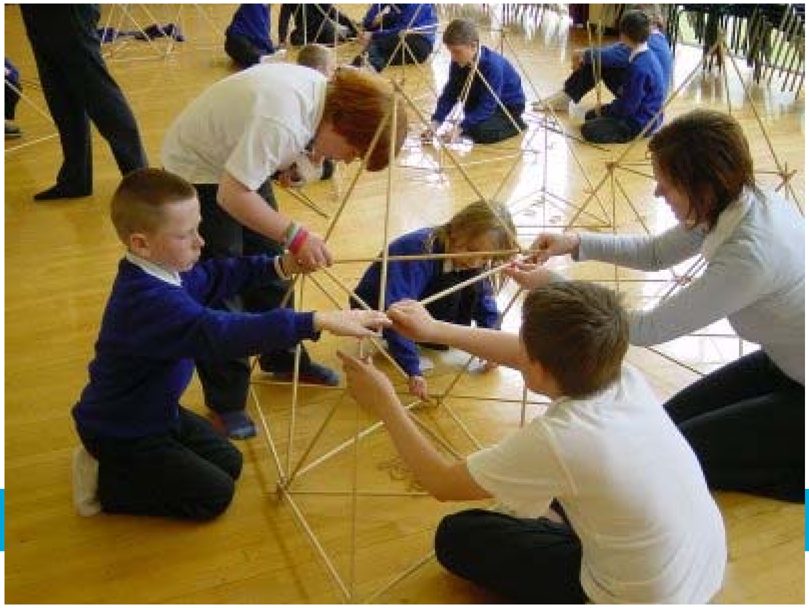
The PhD project results in a a new topological approach to data modelling, based on a tetrahedral network.

- Operators and definitions from the field of simplicial homology are used to define and handle this structure of tetrahedrons.
- Simplicial homology provides a solid mathematical foundation
- Simplicial homology enables one to derive substantial parts of the TEN structure efficiently, instead of explicitly storing all primitives.
- DBMS characteristics as the usage of views, functions and function-based indexes are extensively used to realise this potential data reduction.

Conclusions

- A proof-of-concept implementation was created:

  prevailing view that tetrahedrons are more expensive in terms of storage, is not correct when using the proposed approach (incl. proposed improvements (binary, delta's, etc.)



Using tetrahedrons is easy!

TU Delft

Thesis defence

Thursday June 19, 2008 – 15.00 h

Keep the evening free…