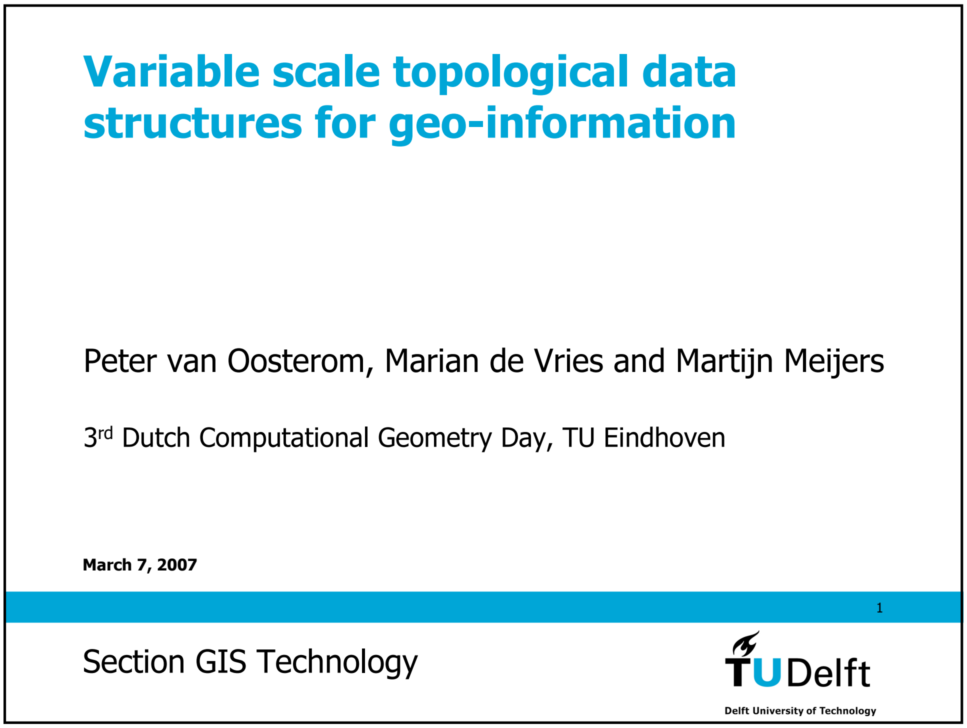


Variable scale topological data structures for geo-information

Peter van Oosterom, Marian de Vries and Martijn Meijers

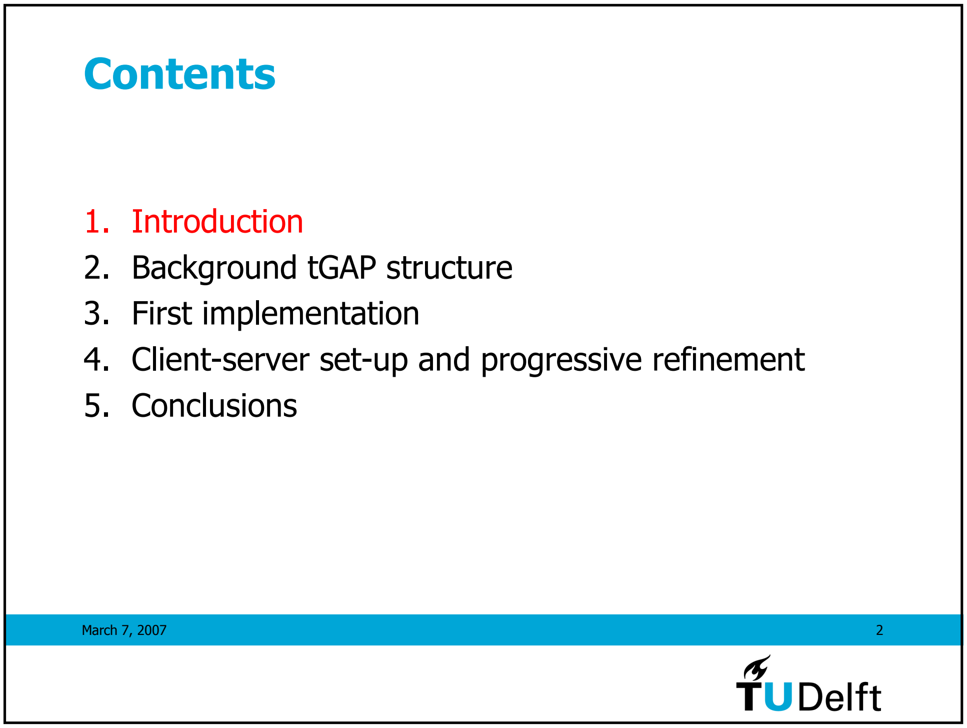
3rd Dutch Computational Geometry Day, TU Eindhoven

March 7, 2007

1

Section GIS Technology

Delft University of Technology



Contents

1. Introduction
2. Background tGAP structure
3. First implementation
4. Client-server set-up and progressive refinement
5. Conclusions

March 7, 2007

2

1. Introduction

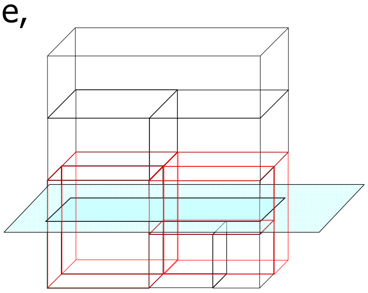
- Multi-scale databases: often multiple representation drawbacks: redundancy, fixed levels of detail
- Scaleless data structures: single representation with additional structure to access at any level of detail

- Often also spatial organization (clustering/indexing)
- Progressive transfer: keep sending more details (compare to raster formats: data pyramids, wavelets)

1. Generalized Area Partitioning-tree (GAP-tree) history

- In normal GAP-tree (van Oosterom 1993) areas are stored as independent polygons, drawback (computed) redundancy
- Vermeij et al.’03 proposed topological GAP-tree: edges and faces (with importance range, consider as height), reduced redundancy between neighbors

- Still some redundancy left: coordinates in higher level edge also present in lower (more detailed) level edges



1. tGAP structure (GAP-face tree + GAP-edge forest)

- Also coordinate redundancy between edges at different aggregation levels is removed

- Throughout remainder of presentation examples of the tGAP-structure (creation and use) will be shown

- Creation of the tGAP-tree is shown in pairs of steps
  1. removal of least important face (merge face)
  2. removal of edges, merge of edges (BLG-tree)

1. Proposed solution: tGAP structure

- Variable scale: infinite amount of levels
- Base level with most detailed geometry/topology
- Create links/structure on top

- Only theory → validate
- Last year at Autocarto 2005, presentation was concluded with question: What is the price of non-redundancy, that is, the many references?

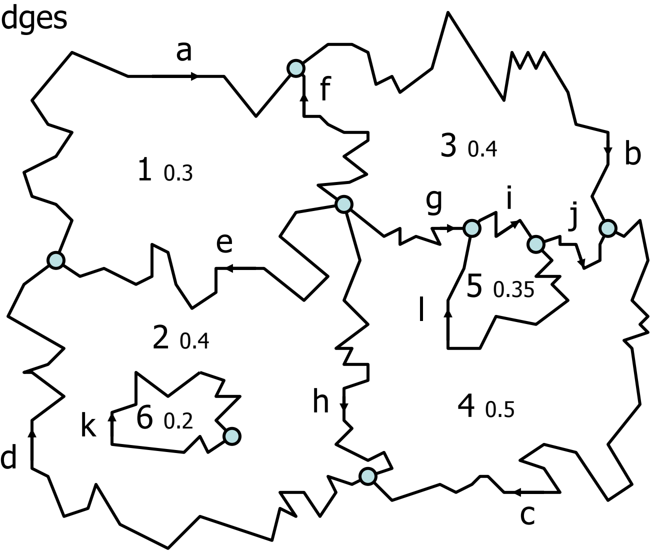
- Test and implement structure

Contents

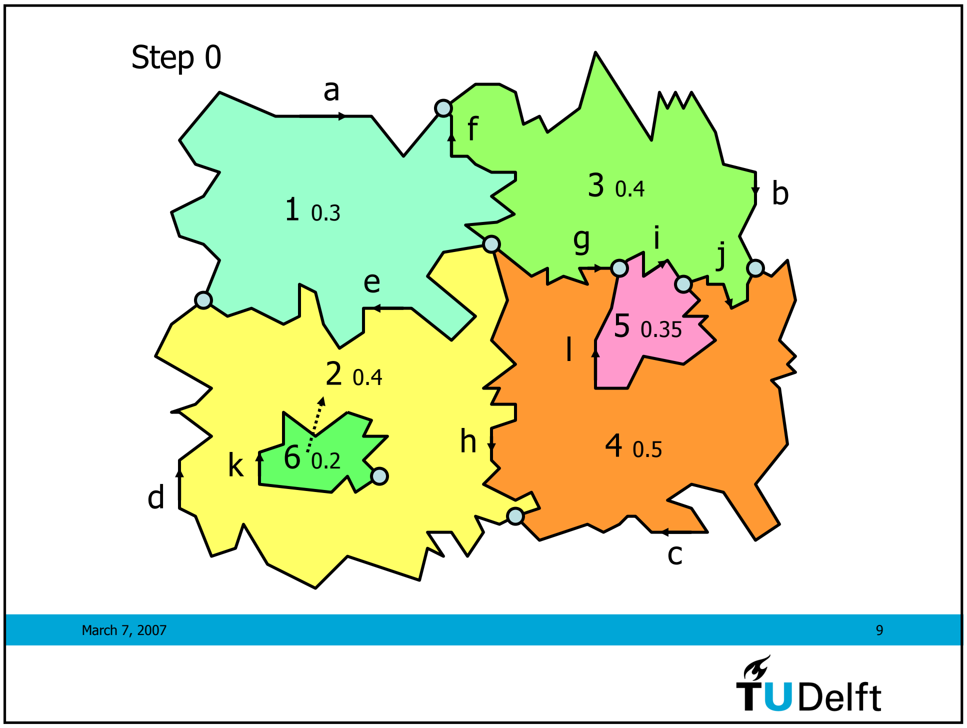
1. Introduction
2. Background tGAP structure
3. First implementation
4. Client-server set-up and progressive refinement
5. Conclusions

TU Delft

edges



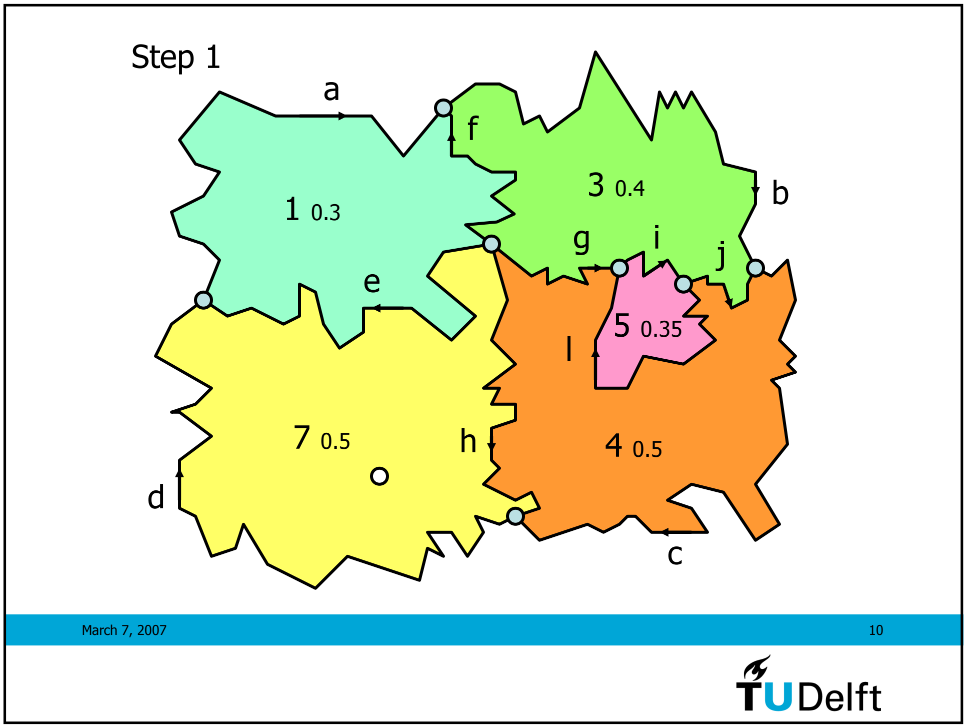
TU Delft



Step 0

March 7, 2007 9

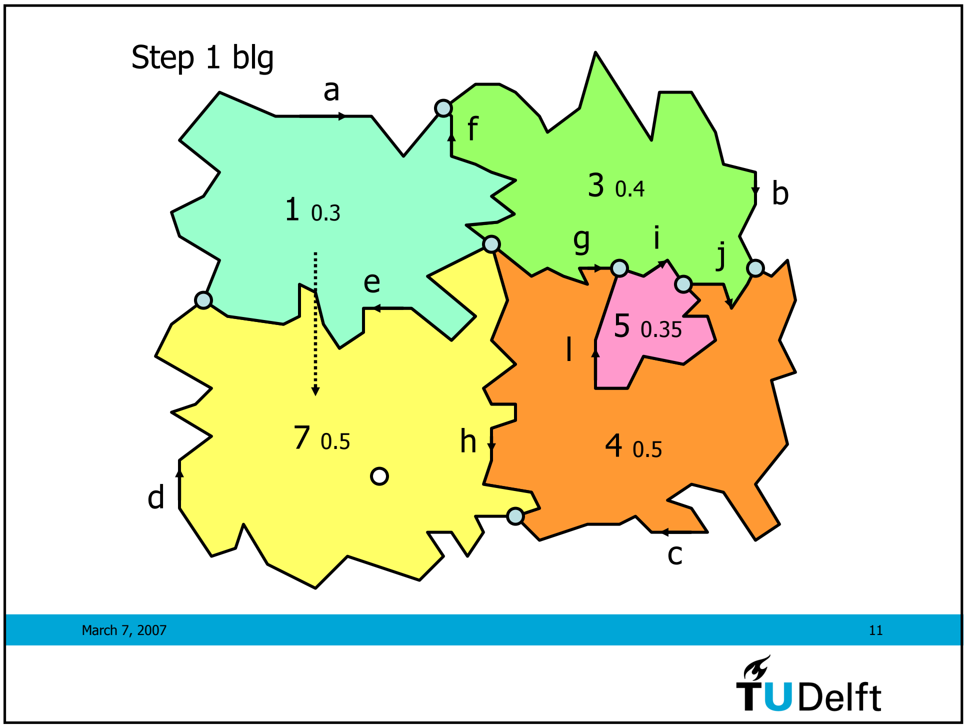
TU Delft



Step 1

March 7, 2007 10

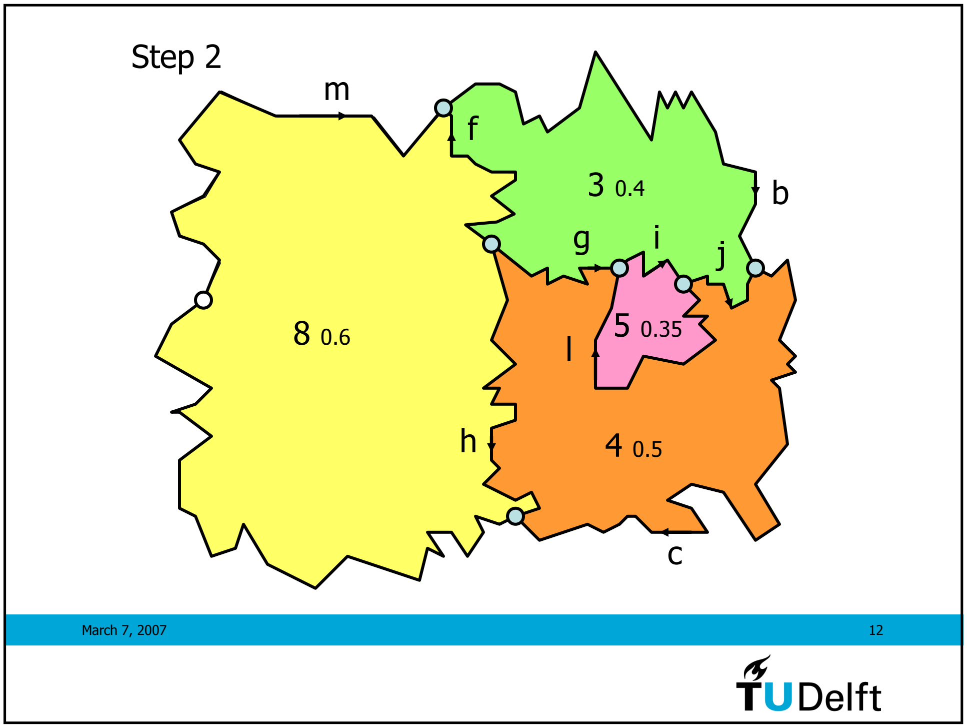
TU Delft



Step 1 blg

March 7, 2007

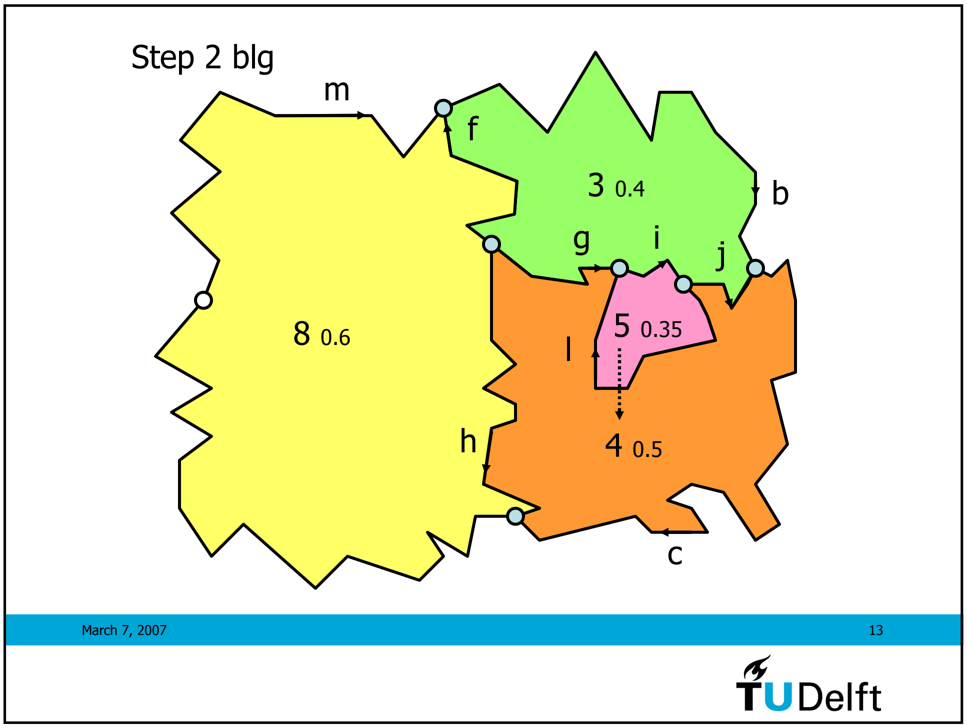
11



Step 2

March 7, 2007

12

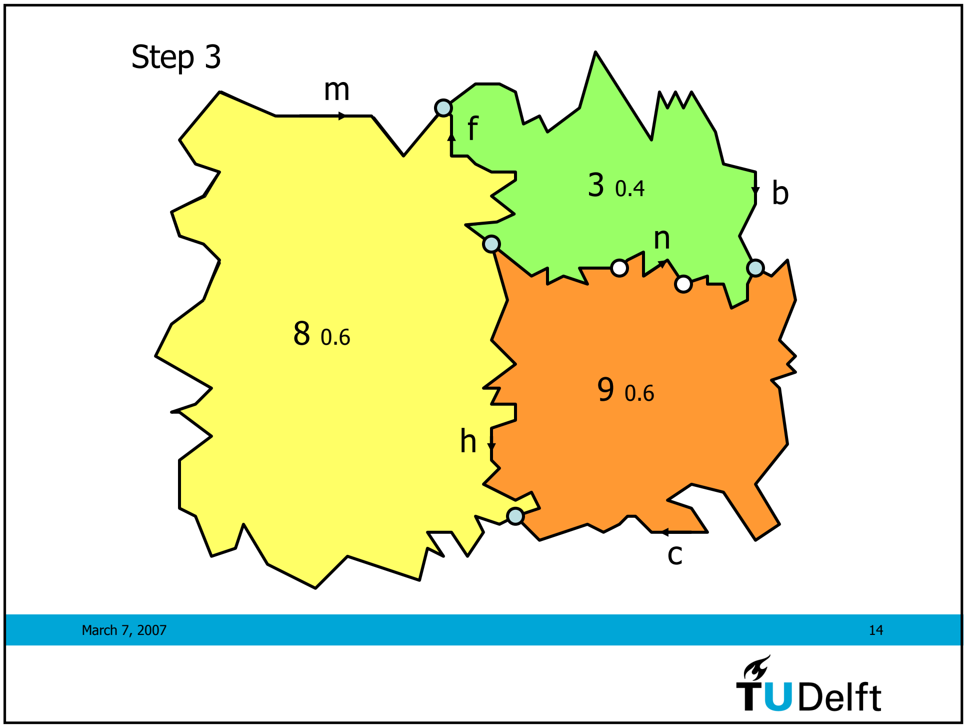


Step 2 blg

March 7, 2007

13

TU Delft

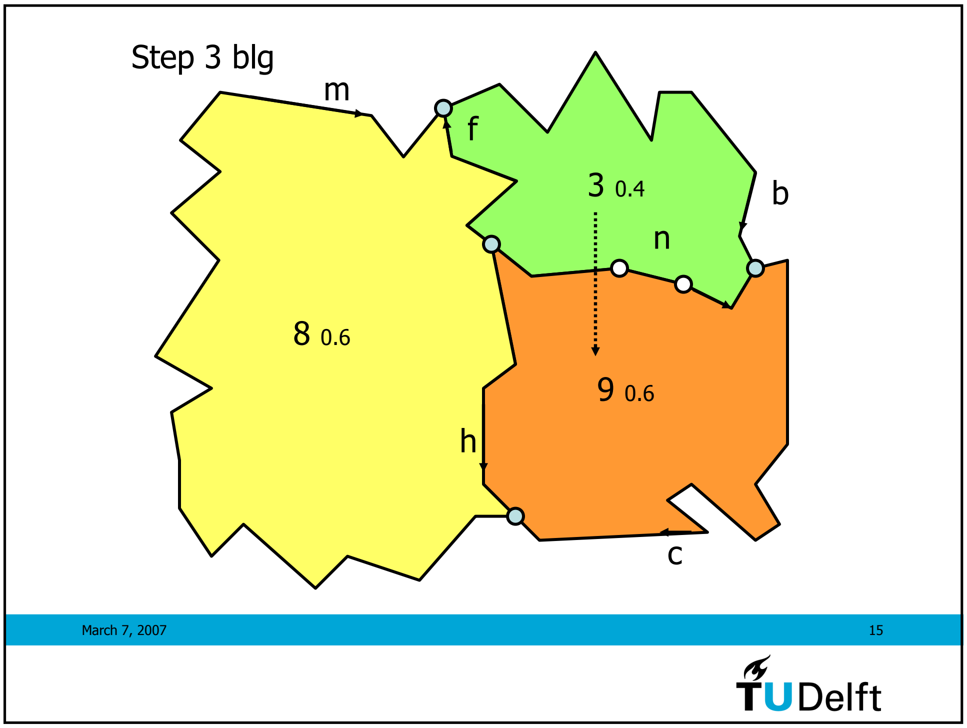


Step 3

March 7, 2007

14

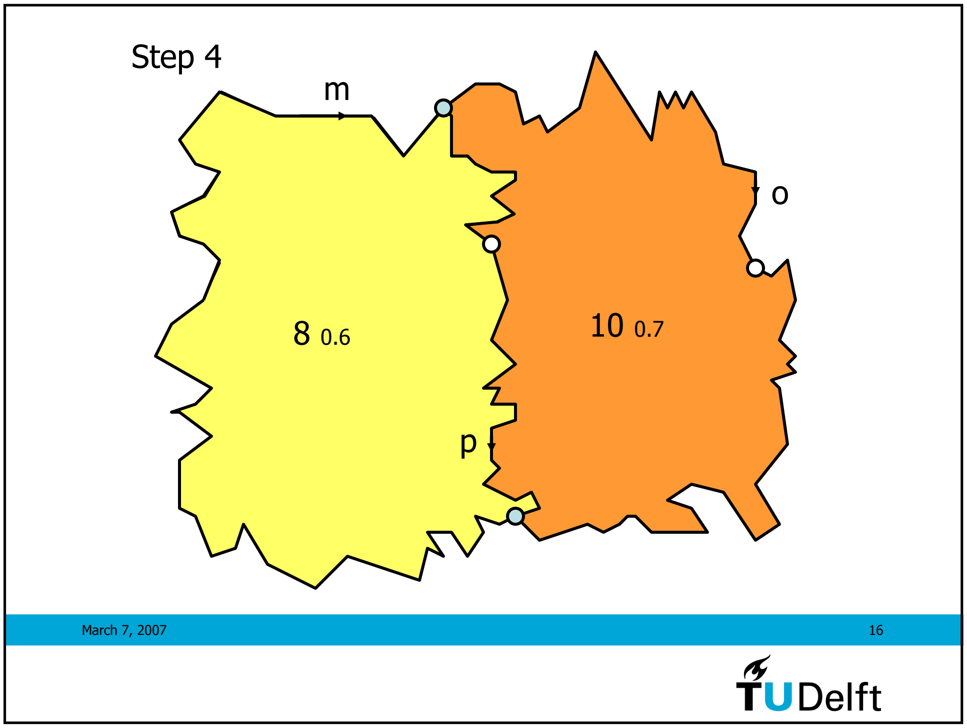
TU Delft



Step 3 blg

March 7, 2007

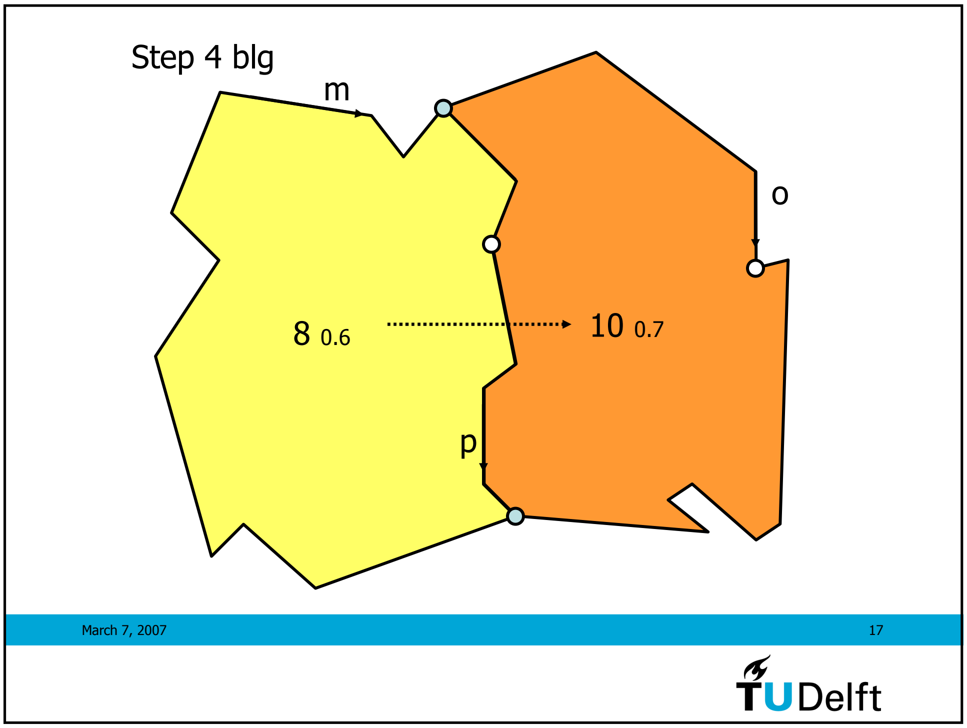
15



Step 4

March 7, 2007

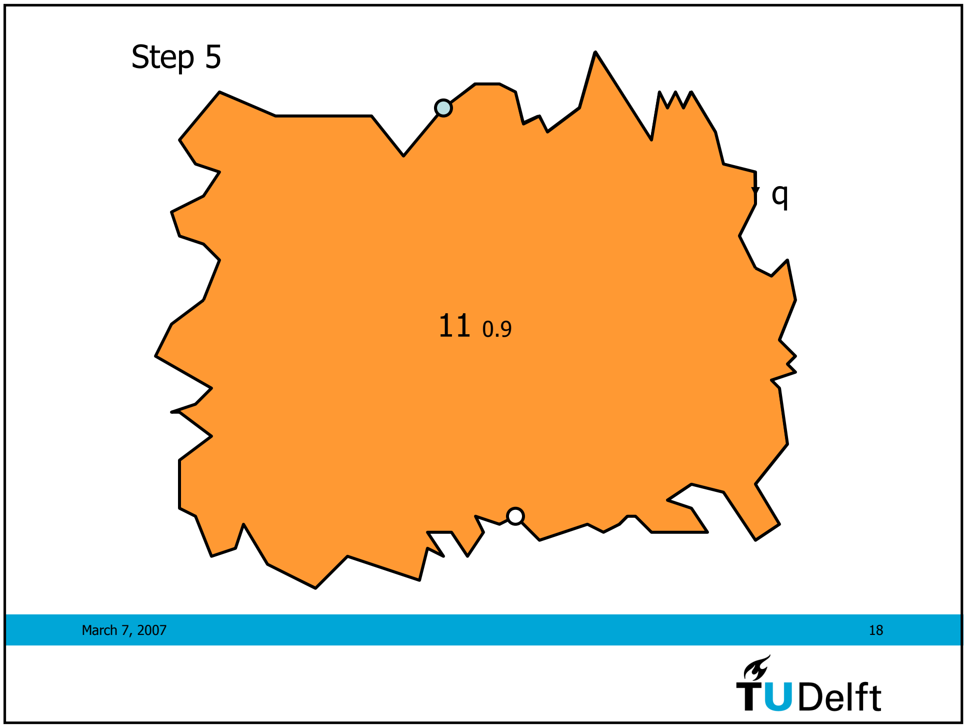
16



Step 4 blg

March 7, 2007 17

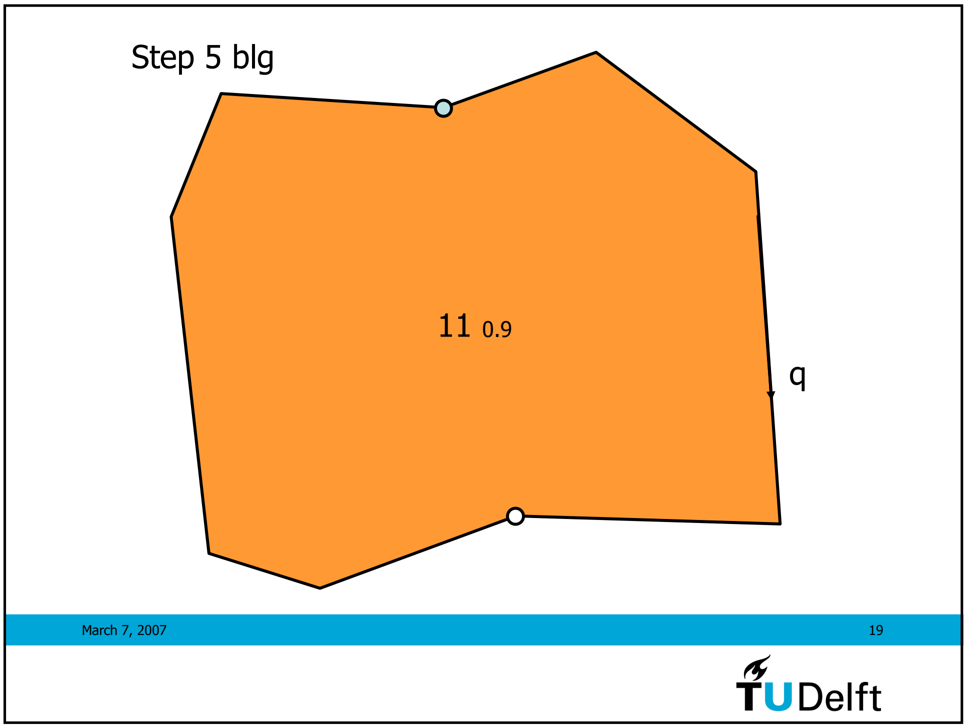
TU Delft



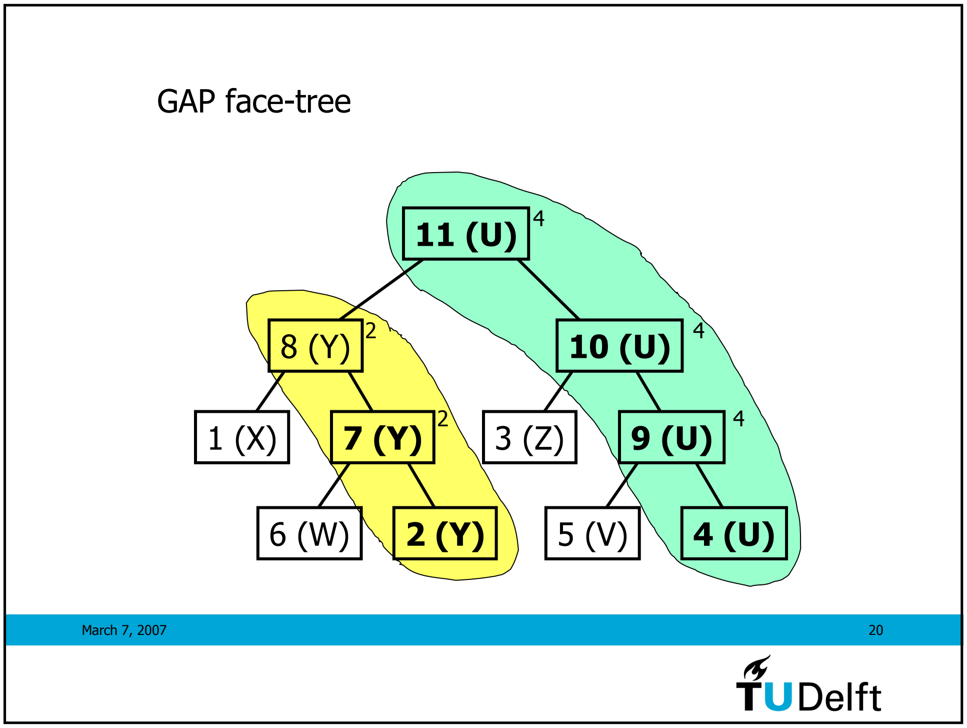
Step 5

March 7, 2007 18

TU Delft

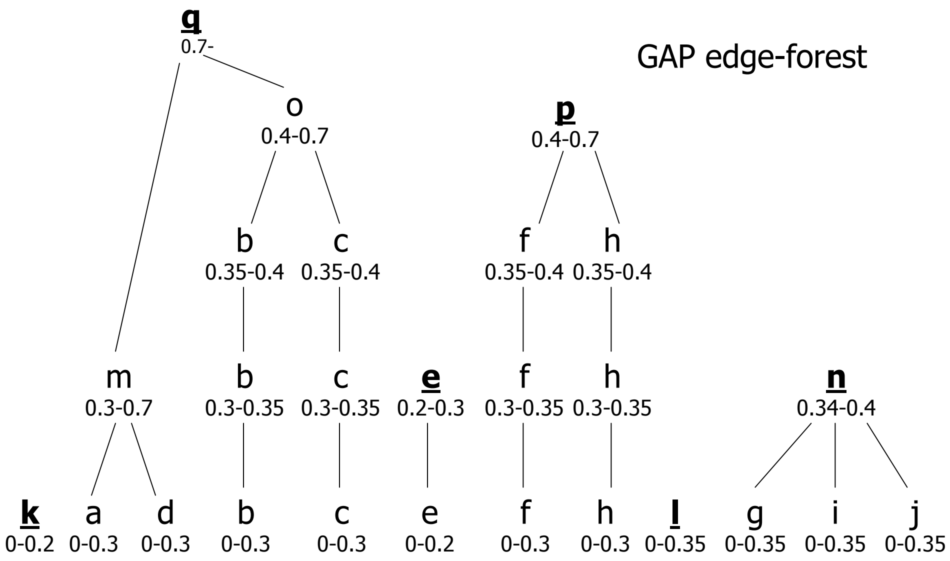


Step 5 blg



GAP face-tree

GAP edge-forest



q
0.7-

o
0.4-0.7

p
0.4-0.7

b
0.35-0.4

c
0.35-0.4

f
0.35-0.4

h
0.35-0.4

m
0.3-0.7

b
0.3-0.35

c
0.3-0.35

e
0.2-0.3

f
0.3-0.35

h
0.3-0.35

n
0.34-0.4

k
0-0.2

a
0-0.3

d
0-0.3

b
0-0.3

c
0-0.3

e
0-0.2

f
0-0.3

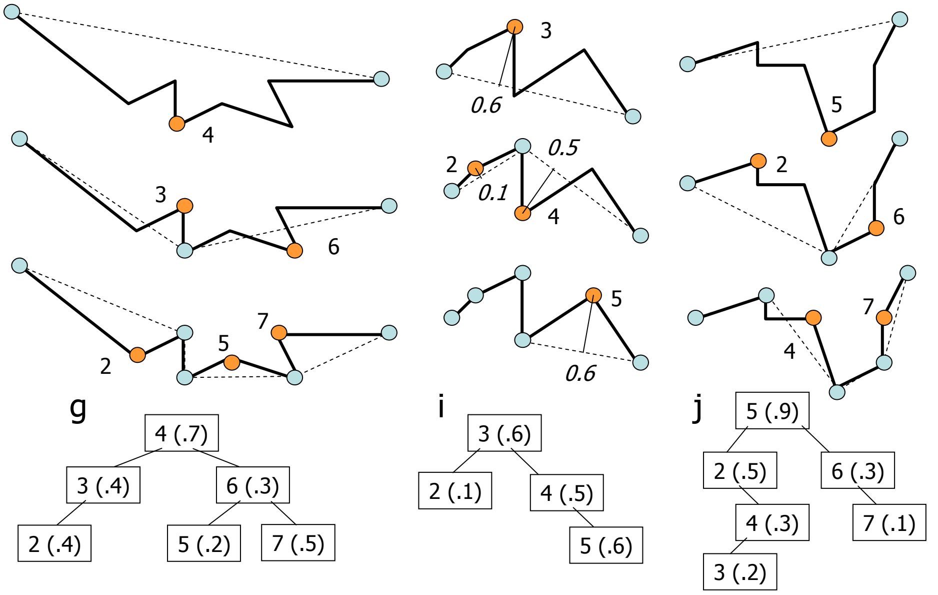
h
0-0.3

l
0-0.35

g
0-0.35

i
0-0.35

j
0-0.35

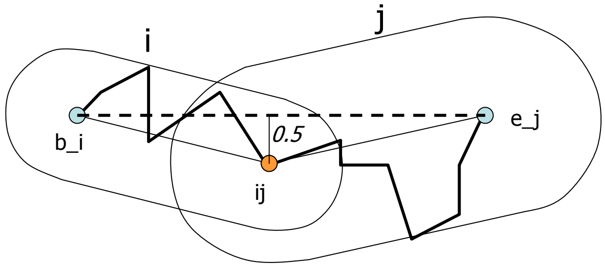


4 (.7)
3 (.4)
6 (.3)
2 (.4)
5 (.2)
7 (.5)

3 (.6)
2 (.1)
4 (.5)
5 (.6)

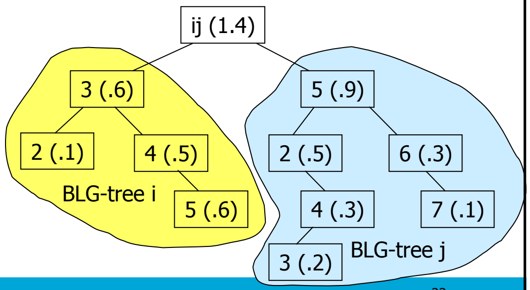
5 (.9)
2 (.5)
6 (.3)
4 (.3)
7 (.1)
3 (.2)

Join BLG-tree's
of edges i and j



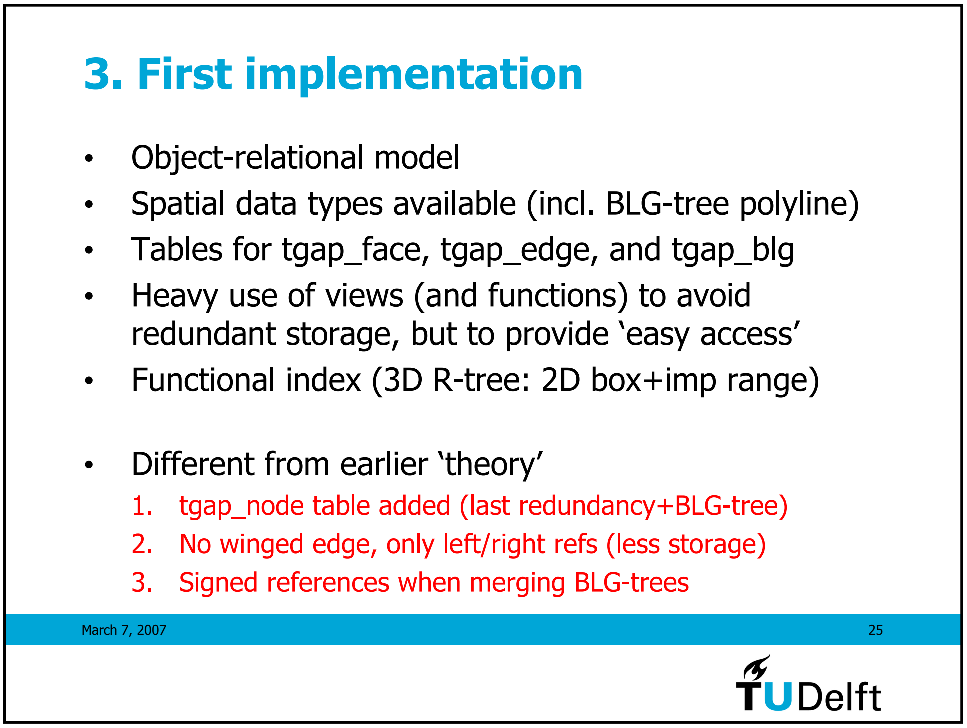
err_ij=dist(point(ij), line(b_i,e_j))
+ max(err_i, err_j)=

0.5+0.9=1.4



Contents

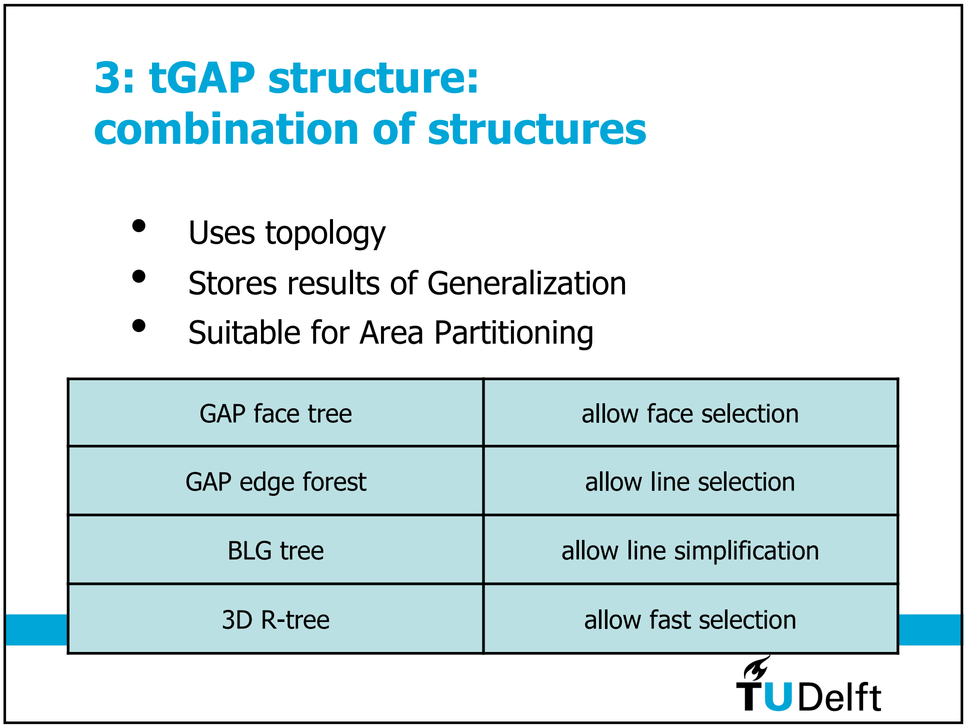
1. Introduction
2. Background tGAP structure
3. First implementation
4. Client-server set-up and progressive refinement
5. Conclusions



3. First implementation

- Object-relational model
- Spatial data types available (incl. BLG-tree polyline)
- Tables for tgap_face, tgap_edge, and tgap_blg
- Heavy use of views (and functions) to avoid redundant storage, but to provide ‘easy access’
- Functional index (3D R-tree: 2D box+imp range)

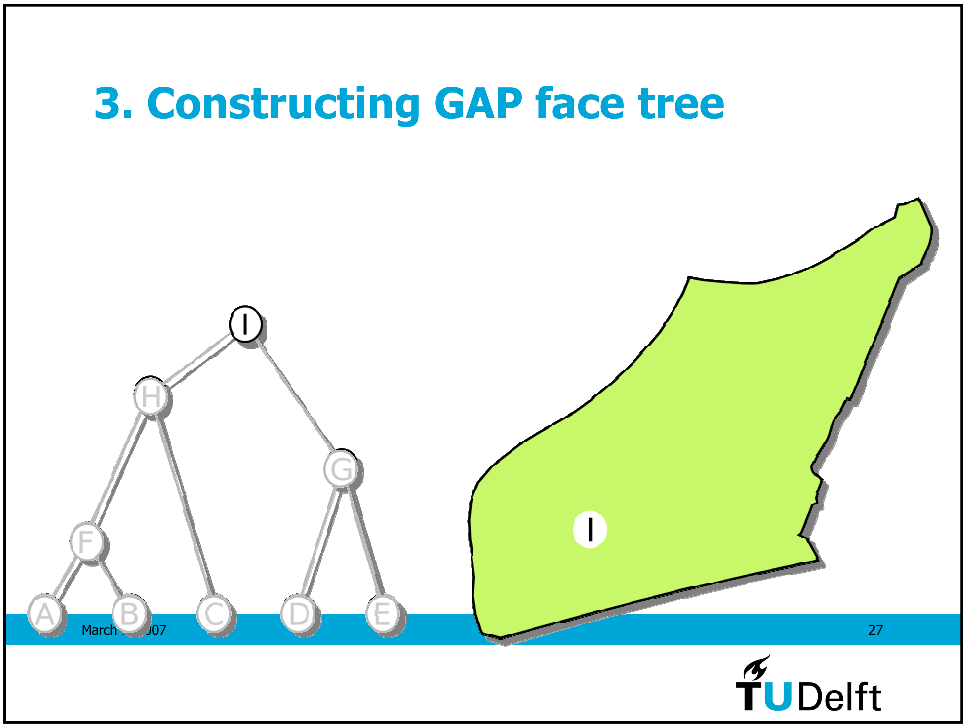
- Different from earlier ‘theory’
  1. tgap_node table added (last redundancy+BLG-tree)
  2. No winged edge, only left/right refs (less storage)
  3. Signed references when merging BLG-trees



3: tGAP structure: combination of structures

- Uses topology
- Stores results of Generalization
- Suitable for Area Partitioning

GAP face tree	allow face selection
GAP edge forest	allow line selection
BLG tree	allow line simplification
3D R-tree	allow fast selection



3. Constructing GAP face tree

F 1 A+B A

```
ERROR: rangecheck
OFFENDING COMMAND: string

STACK:

66038
33018
32512
33019
```