(Smooth) zoom and pan with vario-scale data

Client-server architecture options and benchmarking their performance

Martijn Meijers Lina Huang
Radan Šuba Peter van Oosterom

Faculty of Architecture and the Built Environment,

OTB Research, GIS technology

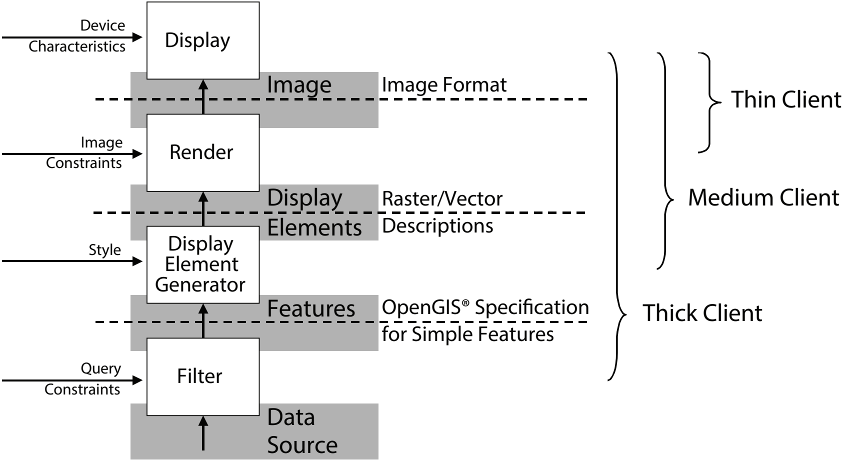
November 19, 2014 – Gemeente Amsterdam – User committee STW

Introduction

- Last meeting:
  Create large vario-scale data structure (2D geometry + 1D scale)
- How to visualize contents?
  Given that we want to use a client-server web architecture + large structure
- Today discuss:
  1. What are the options?
  2. What kind of pan/zoom interaction is possible (smooth transitions possible: yes / no)?
  3. How do the different options perform?

Rendering pipeline

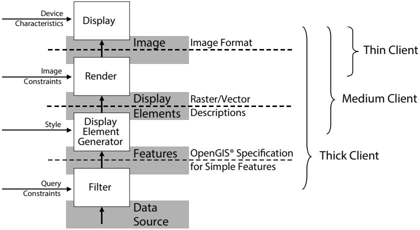
From data to map, i. e. on-screen graphics (OGC, 2000)



Rendering pipeline

Steps for making a map out of vario-scale data:

1. Select relevant topology primitives in data source — Slice at correct scale
2. Make features
   - Optional: clip edges
   - Form rings of edges, from rings form polygons
3. Transform to display elements, style and render polygons



Rendering pipeline

We tested 3 options:

Non-smooth transitions:

- Option A Making a map (every request independent from other requests)
- Option B New map dependent on earlier request(s) (to transfer less data)

Smooth transition:

- Option C Making a map with ‘gradual’/‘smooth’ transition to new map

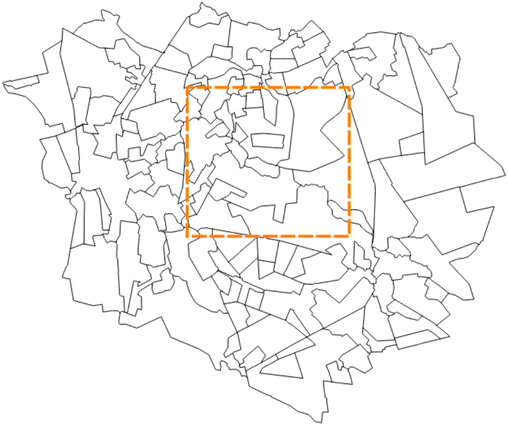
Making a map – Option A and B

Vario-scale structure (top view)

Making a map – Option A and B

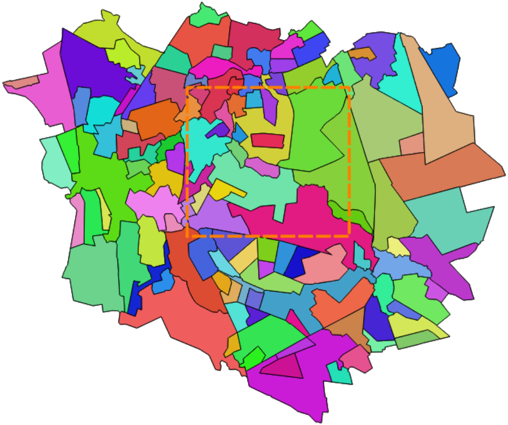
Viewport

Making a map – Option A and B



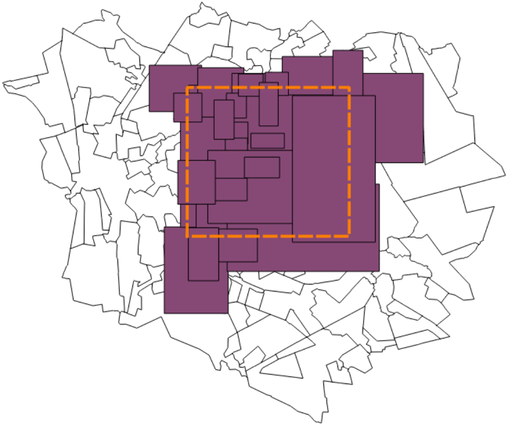
Take slice (determine where to slice) and clip (=viewport)

Making a map – Option A and B



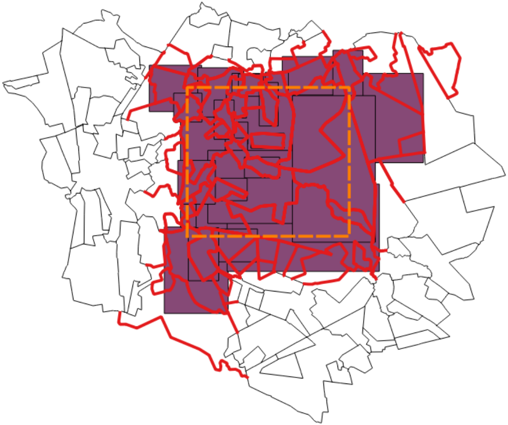
Form polygons

Making a map – Option A and B



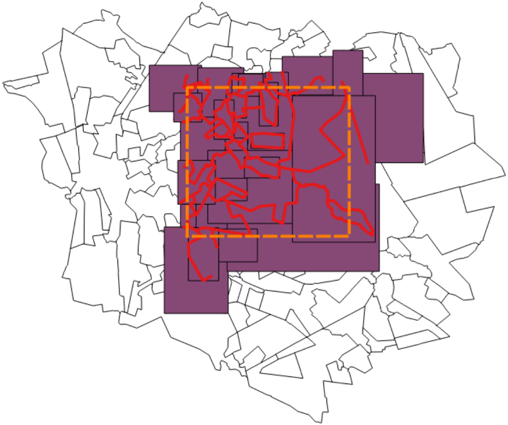
Faces that overlap

Making a map – Option A and B



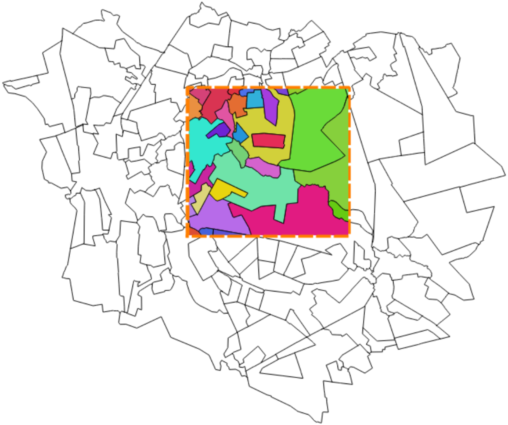
Edges that are needed for these faces

Making a map – Option A and B



Edges that are needed for the visible face parts

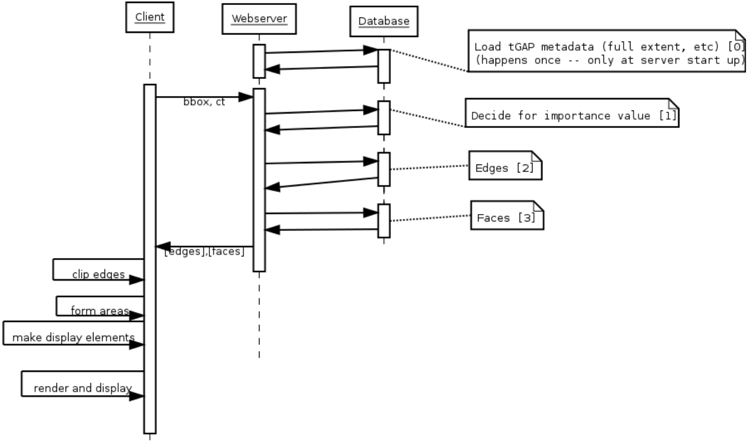
Making a map – Option A and B



Polygon parts on screen (inside viewport)

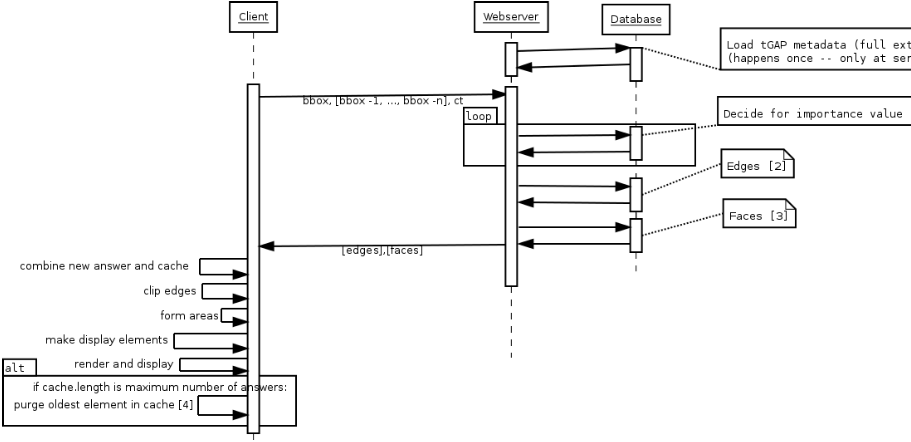
Data retrieval – Option A and B

Option A



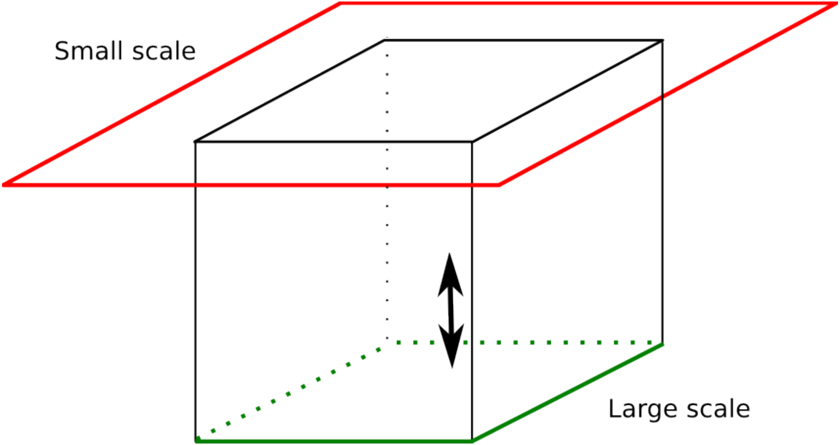
Data retrieval – Option A and B

Option B


Client
Webserver
Database
Load tGAP metadata (full ext
(happens once -- only at ser
bbox, [bbox -1, ..., bbox -n], ct
loop
Decide for importance value
Edges [2]
Faces [3]
[edges],[faces]
combine new answer and cache
clip edges
form areas
make display elements
render and display
alt
if cache.length is maximum number of answers:
purge oldest element in cache [4]

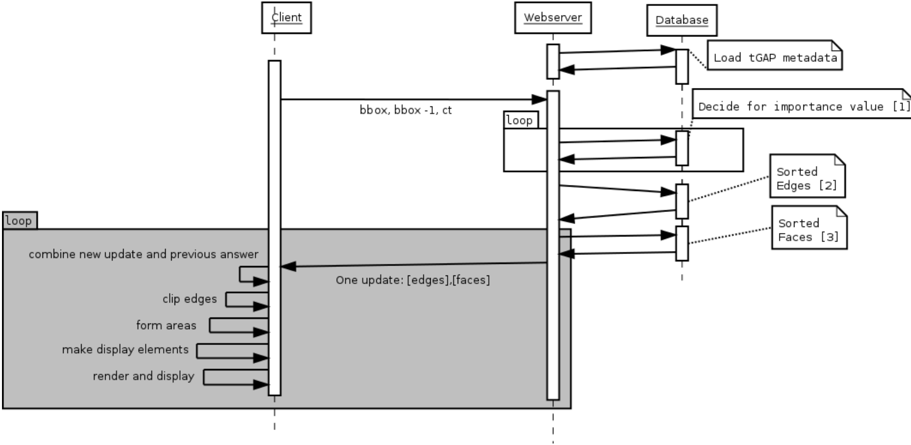
Making a map – Option C

Making a map – Option C



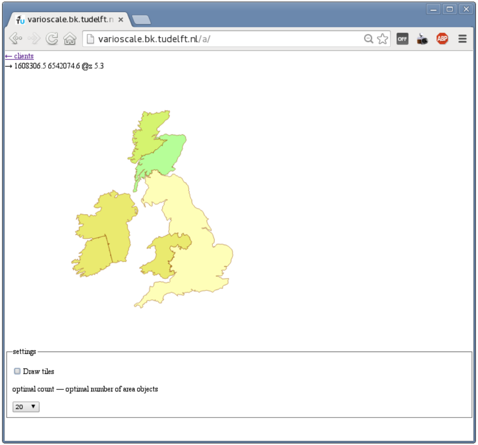
Data retrieval – Option C

Option C



Demo

`http://varioscale.bk.tudelft.nl/`



← clients

→ 1608306.5 6542074.6 @z 5.3

settings

□ Draw tiles

optimal count — optimal number of area objects

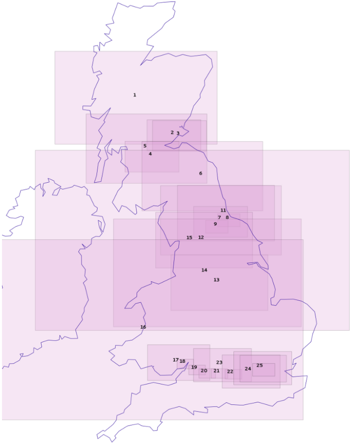
20 ▼

Experiment

- Loaded Corine Land cover for UK
- Make requests to webserver simulating ‘user interaction’
- Measure:
  1. Response time: time to first byte and to last byte (duration)
  2. Response size
  3. (Number of edges and faces)

Experiment

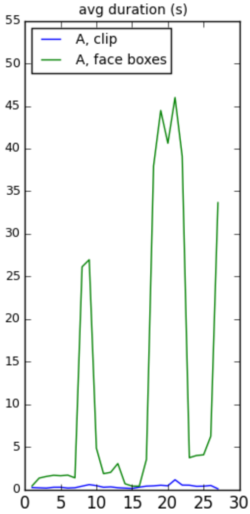
Rectangles = user path (zoom and pan actions of a user)

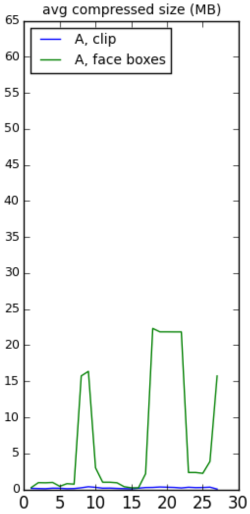


Note: smaller rectangle is larger scale (zoomed in more)

Experiment

Clipping – blue versus Not clipping – green (Option A only)

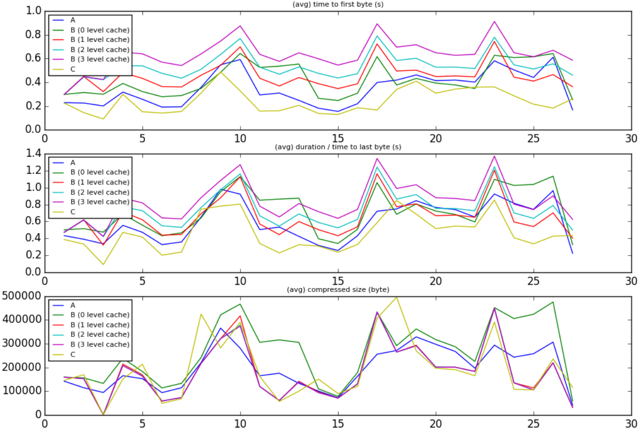




Experiment

Option A versus B(0,1,2,3) versus C

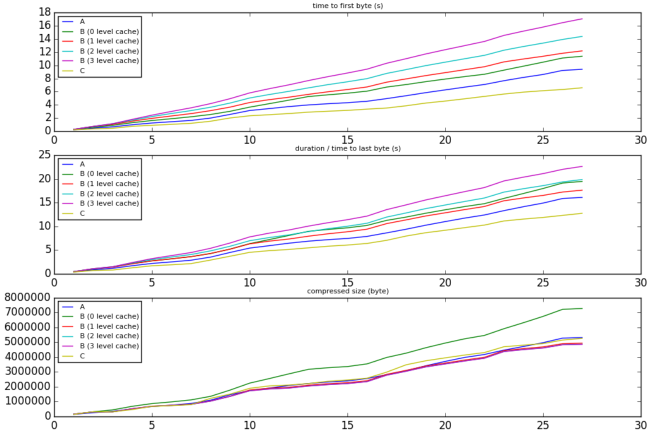
North to South over whole of Great Brittain, zooming in, out and panning, average measurements



Experiment

Option A versus B(0,1,2,3) versus C

North to South over whole of Great Brittain, zooming in, out and panning, measurements (cumulative)



Results summary

- Not clipping edges leads to excess data.

Results summary

- Not clipping edges leads to excess data.
- Allows for interactive use at arbitrary map scale: response for options A, B1, C on average < 1 second.

Results summary

- Not clipping edges leads to excess data.
- Allows for interactive use at arbitrary map scale: response for options A, B1, C on average < 1 second.
- Option A: fits in seamlessly with (back end of) current architectures [(Tiled) WMS, WFS].

Results summary

- Not clipping edges leads to excess data.
- Allows for interactive use at arbitrary map scale: response for options A, B1, C on average < 1 second.
- Option A: fits in seamlessly with (back end of) current architectures [(Tiled) WMS, WFS].
- Option B: sufficient to remember one previous request, otherwise too expensive to query with no additional savings.

Results summary

- Not clipping edges leads to excess data.
- Allows for interactive use at arbitrary map scale: response for options A, B1, C on average < 1 second.
- Option A: fits in seamlessly with (back end of) current architectures [(Tiled) WMS, WFS].
- Option B: sufficient to remember one previous request, otherwise too expensive to query with no additional savings.
- Option C: makes smooth zoom transition possible: vario-scale structure very suitable.

Future work

- Improve implementations: client-side not completely finished.

Future work

- Improve implementations: client-side not completely finished.
- Smooth pan: add data while dragging (similar as with smooth zoom).

Future work

- Improve implementations: client-side not completely finished.
- Smooth pan: add data while dragging (similar as with smooth zoom).
- Optimize (coordinates are using many digits, independent from scale level).

Future work

- Improve implementations: client-side not completely finished.
- Smooth pan: add data while dragging (similar as with smooth zoom).
- Optimize (coordinates are using many digits, independent from scale level).
- Inconsistent line simplification (some lines much more simplified than others).

Future work

- Improve implementations: client-side not completely finished.
- Smooth pan: add data while dragging (similar as with smooth zoom).
- Optimize (coordinates are using many digits, independent from scale level).
- Inconsistent line simplification (some lines much more simplified than others).
- More gradual transitions.

Thank you for your attention

- Delft University of Technology
  Faculty of Architecture and the Built Environment
  OTB Research
  GIS Technology

- dr. ir. Martijn Meijers
  b.m.meijers@tudelft.nl
  tel. (+31) 15 27 85642

References

OGC (2000). OpenGIS Web Map Server Interface Implementation Specification.