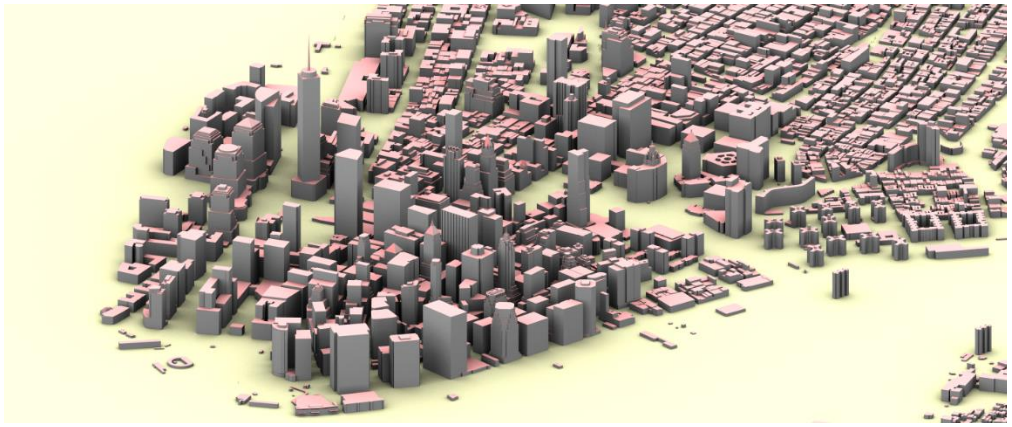
GeoPython 2016
Muttenz Switzerland June 22-24

Preprocessing 3D-City Models in the cloud



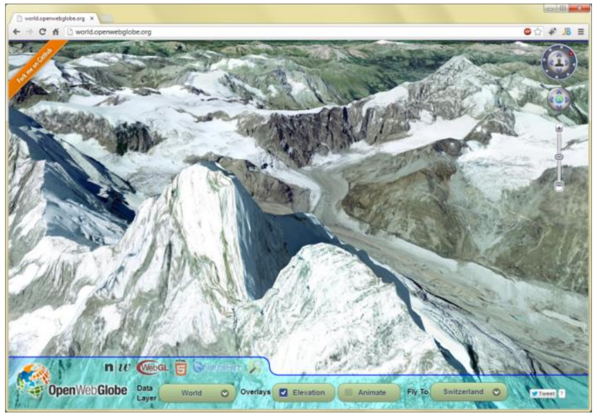
Martin Christen, Stephan Nebiker
FHNW – University of Applied Sciences and Arts Northwestern Switzerland

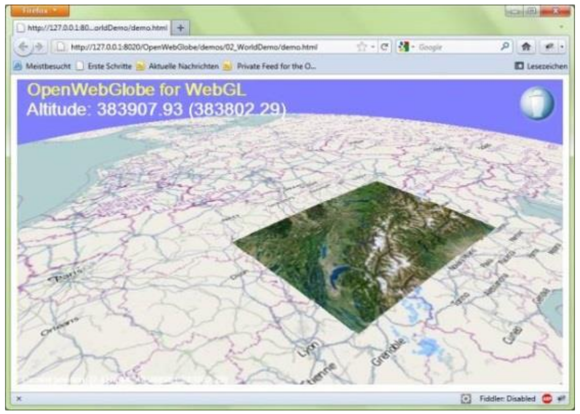
Institute of Geomatics Engineering

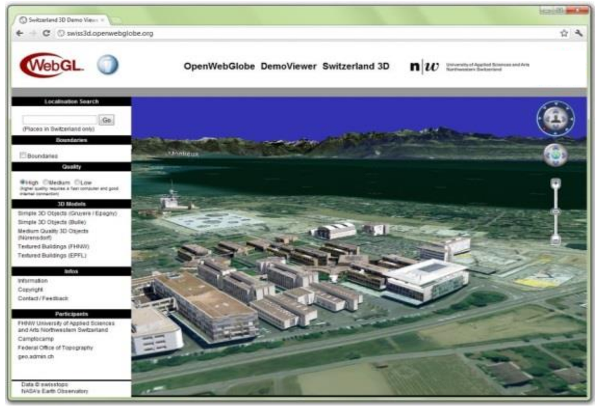
martin.christen@fhnw.ch
@MartinChristen

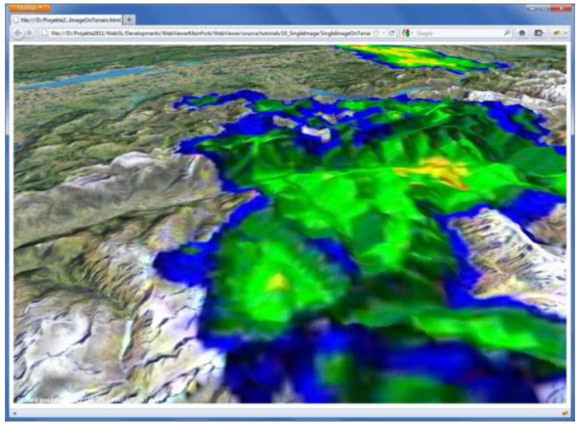
Previous Work

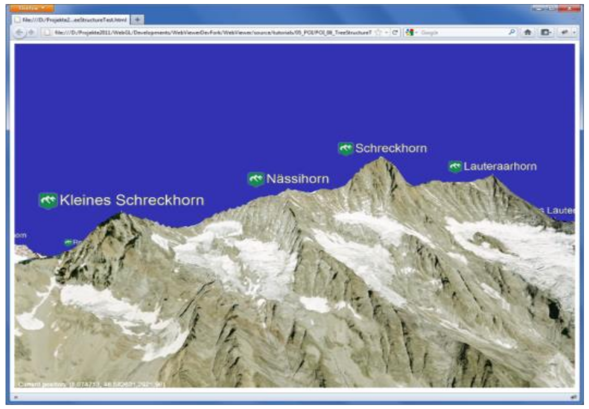
- Virtual Globe using WebGL
- Open Source Project started in April 2011
- Based on C++ version started in 2005
- JavaScript Library for rapid development of web-based geospatial 3D applications

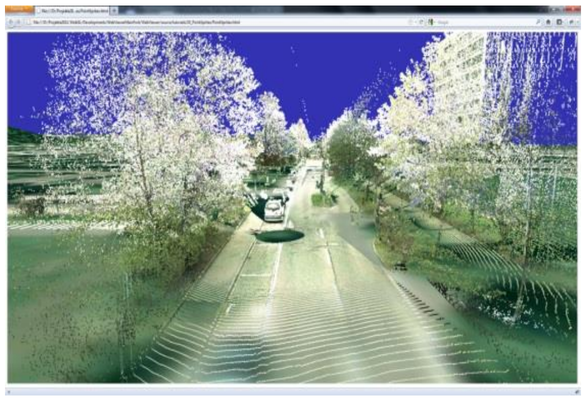


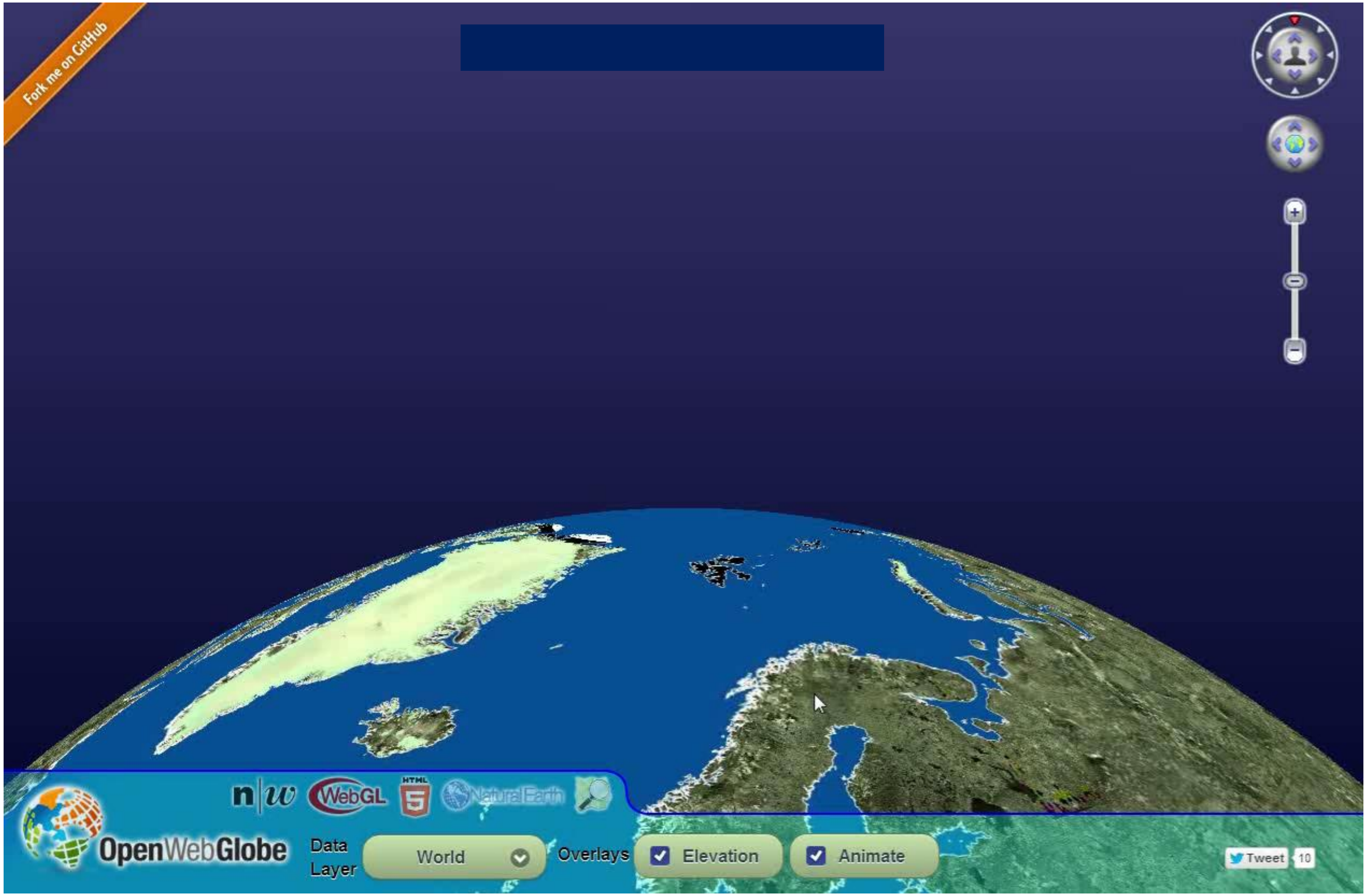




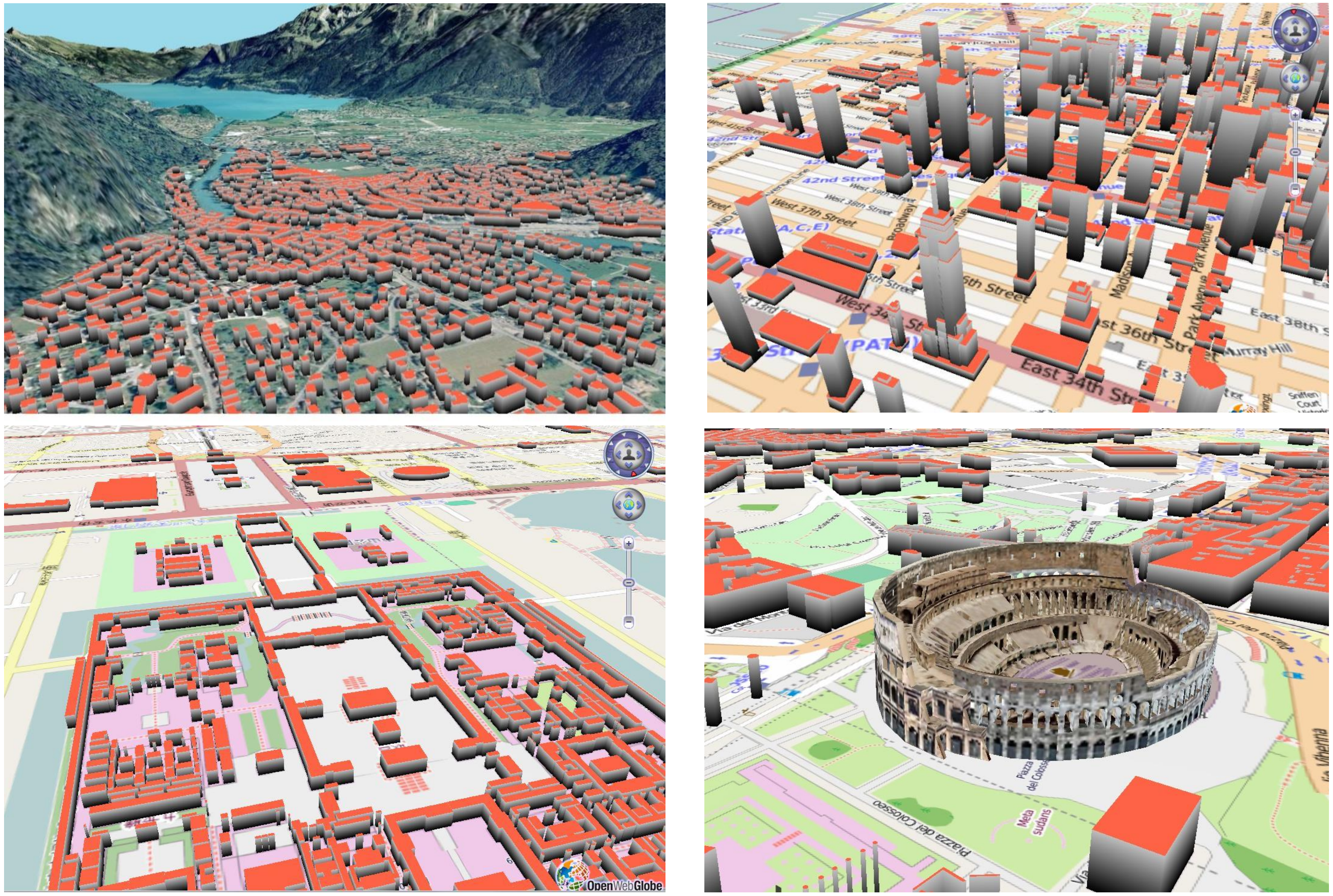







Fork me on GitHub
WebGL
HTML 5
NaturalEarth
OpenWebGlobe
Data Layer
World
Overlays
✓ Elevation
✓ Animate
Tweet 10

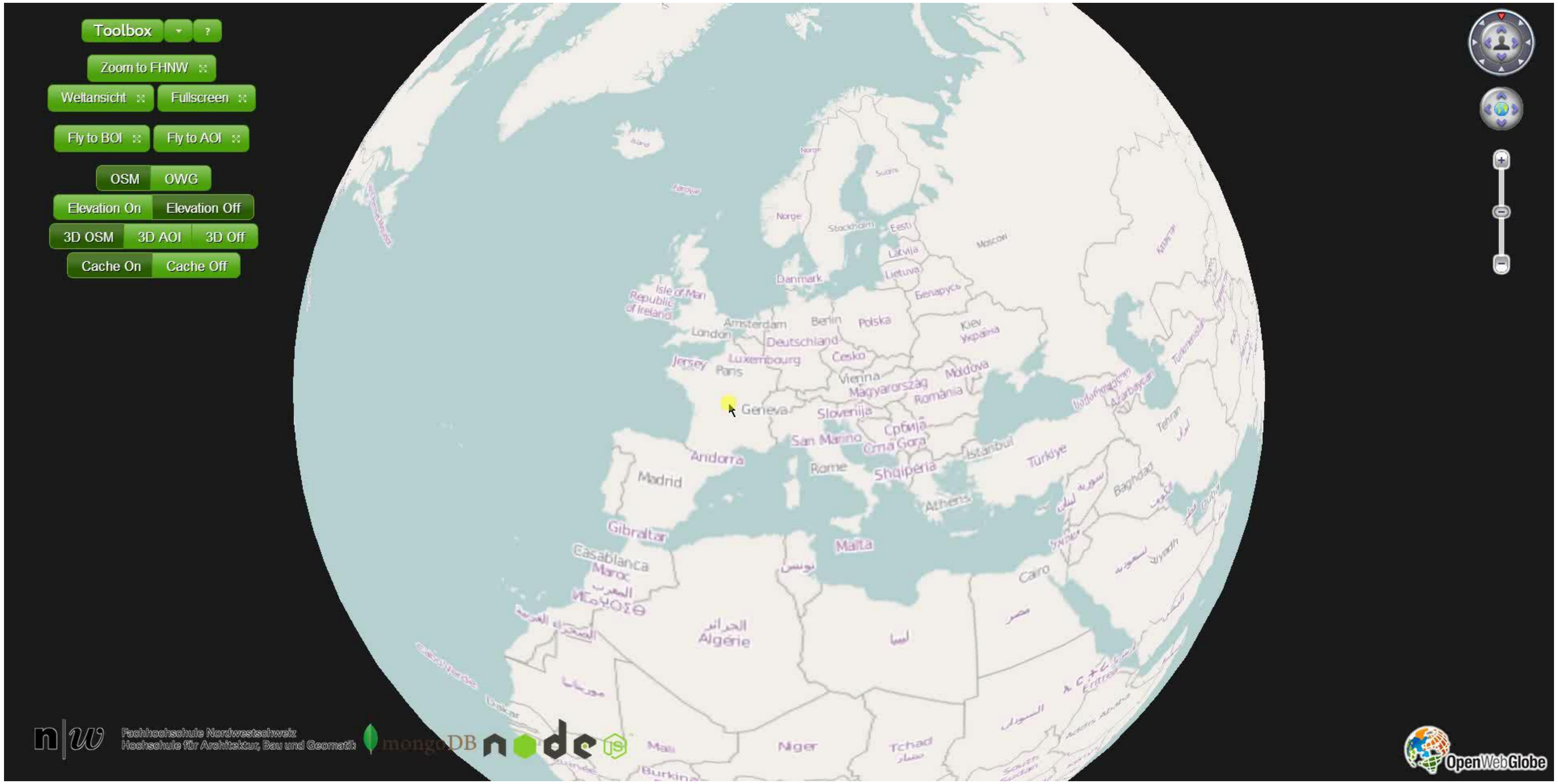
Streaming 3D-Geometry Tiles



BTh Hürbi/Daetwyler, MTh Lucas Oertli, 2013

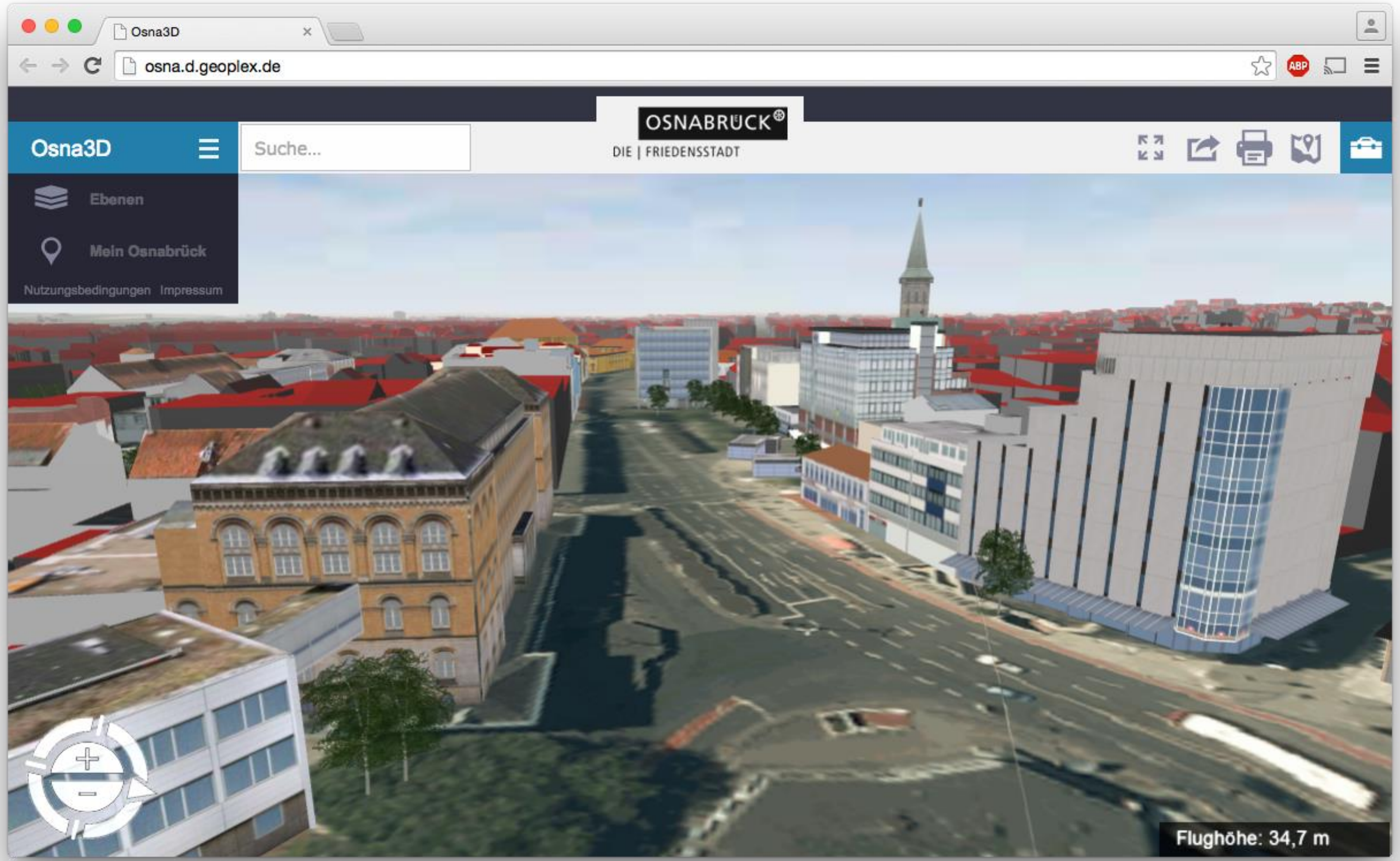
MapData © OpenStreetMap contributors

Streaming Example: 3D Geometry using OSM and “BOI” (worldwide streaming)



MTh Lucas Oertli, 2013

Streaming Example: Osnabrück (local streaming)



Created by Geoplex with Plexmap, based on OpenWebGlobe

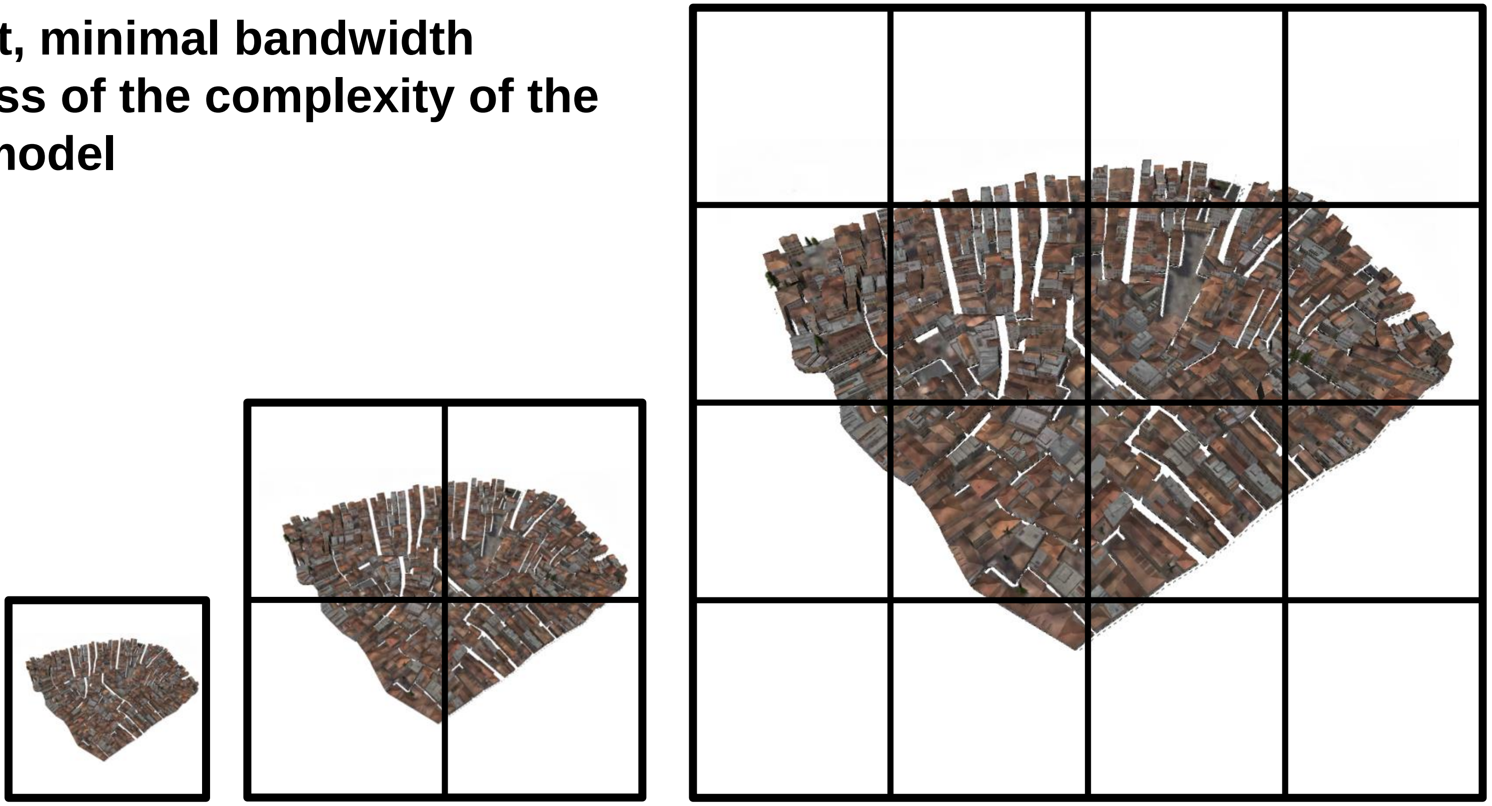
Some Problems I have with (Web-Based) Virtual Globes

- Unfortunately, WebGL compatibility is still an issue… a “fallback” is required
- Most people still prefer 2D Maps
- Navigation in 3D is too complicated for many users…
- In the “Geo-World”, 3D Models are usually not built by 3D game designers:
    - Often there are “too many” & “too big” textures per object
    - Different details per 3D-object
    - Remember “Google 3D Warehouse”
- Level of Detail: Generalization in 2D is accepted, but not in 3D!
- Limited number of people actually do have data **of the whole world**…
- Most virtual globe based applications I know are limited to a certain region/country/…
- Too slow (bandwidth/3D rendering) on mobile devices
- Too power consuming on mobile devices

Bringing together 2D Maps and 3D Globes

Concept: Prerender a 3D Scene using a high quality offline 3D renderer using an orthographic projection and create “2D” image tiles.

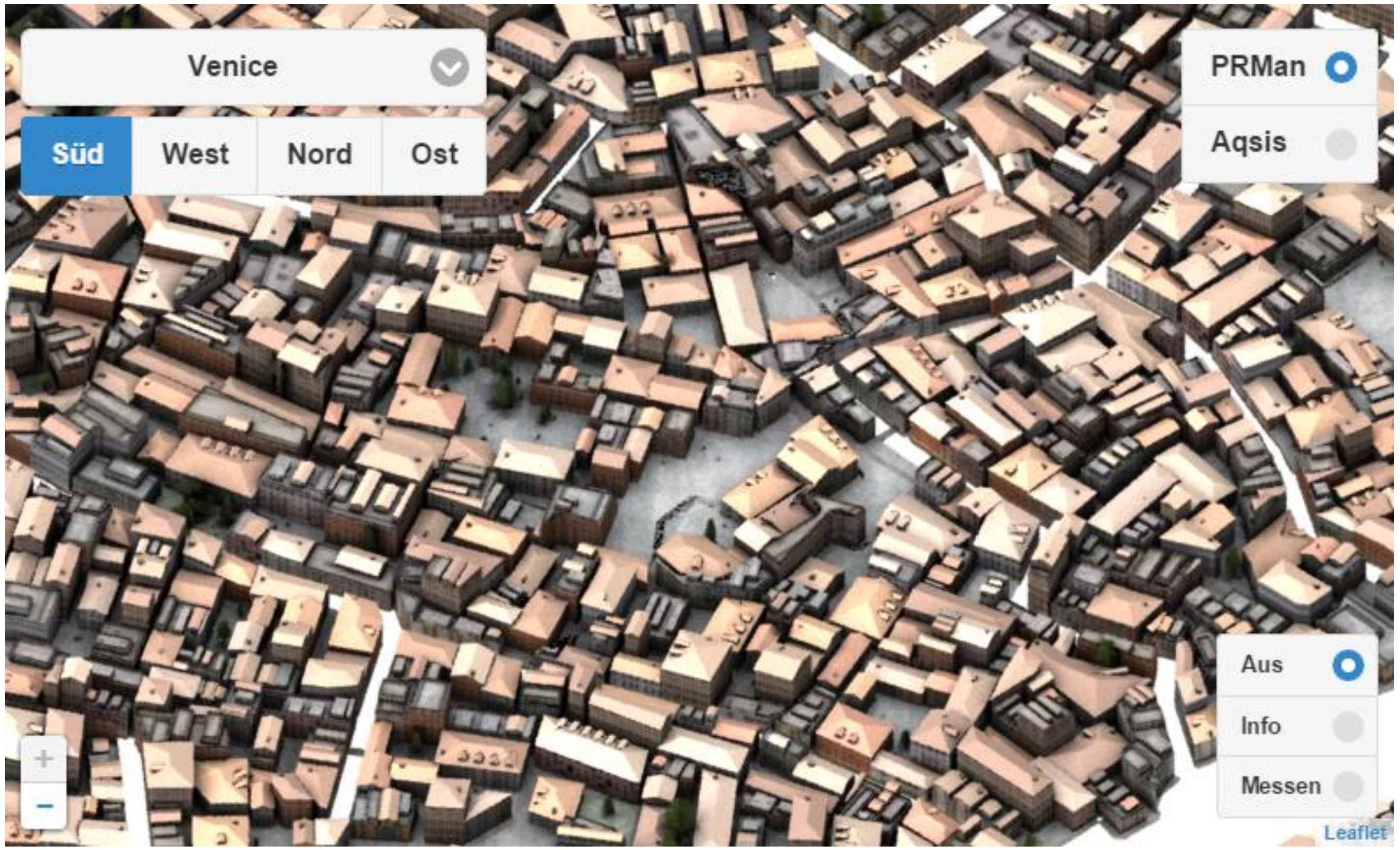
Constant, minimal bandwidth regardless of the complexity of the 3D city model



MTh Markus Jung, 2014

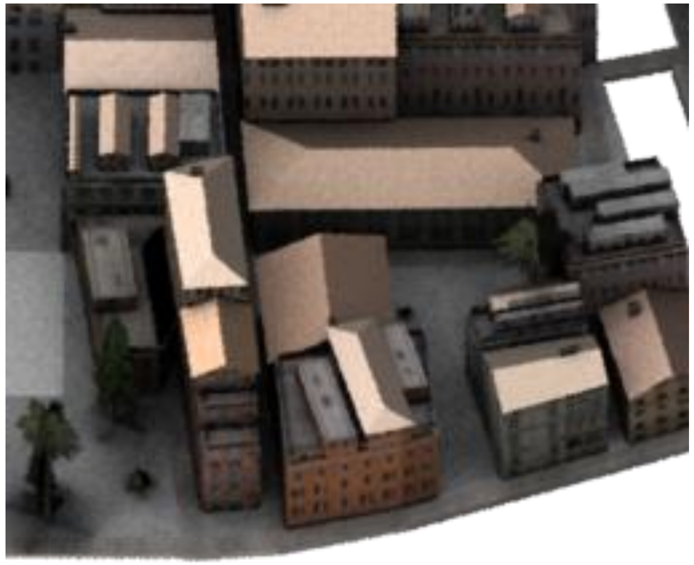
(Similar approaches were already done by Döllner et al. and also go back to some concepts by Sutherland)

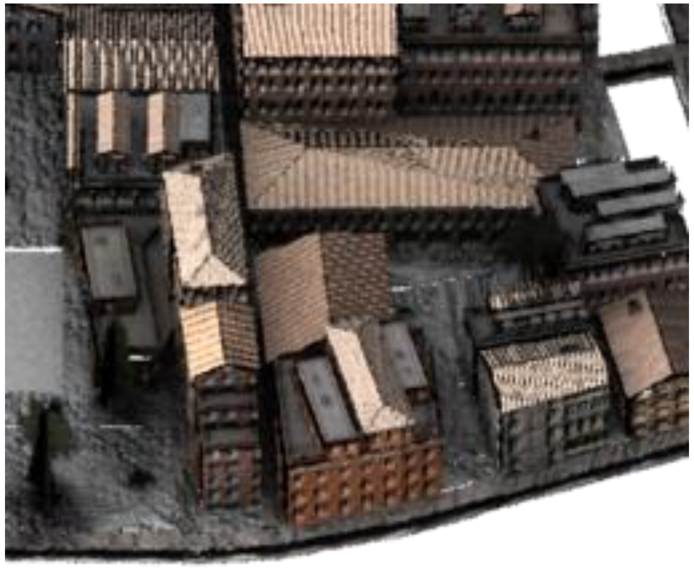
Display in the Webbrowser as “2D Map”



MTh Markus Jung, 2014

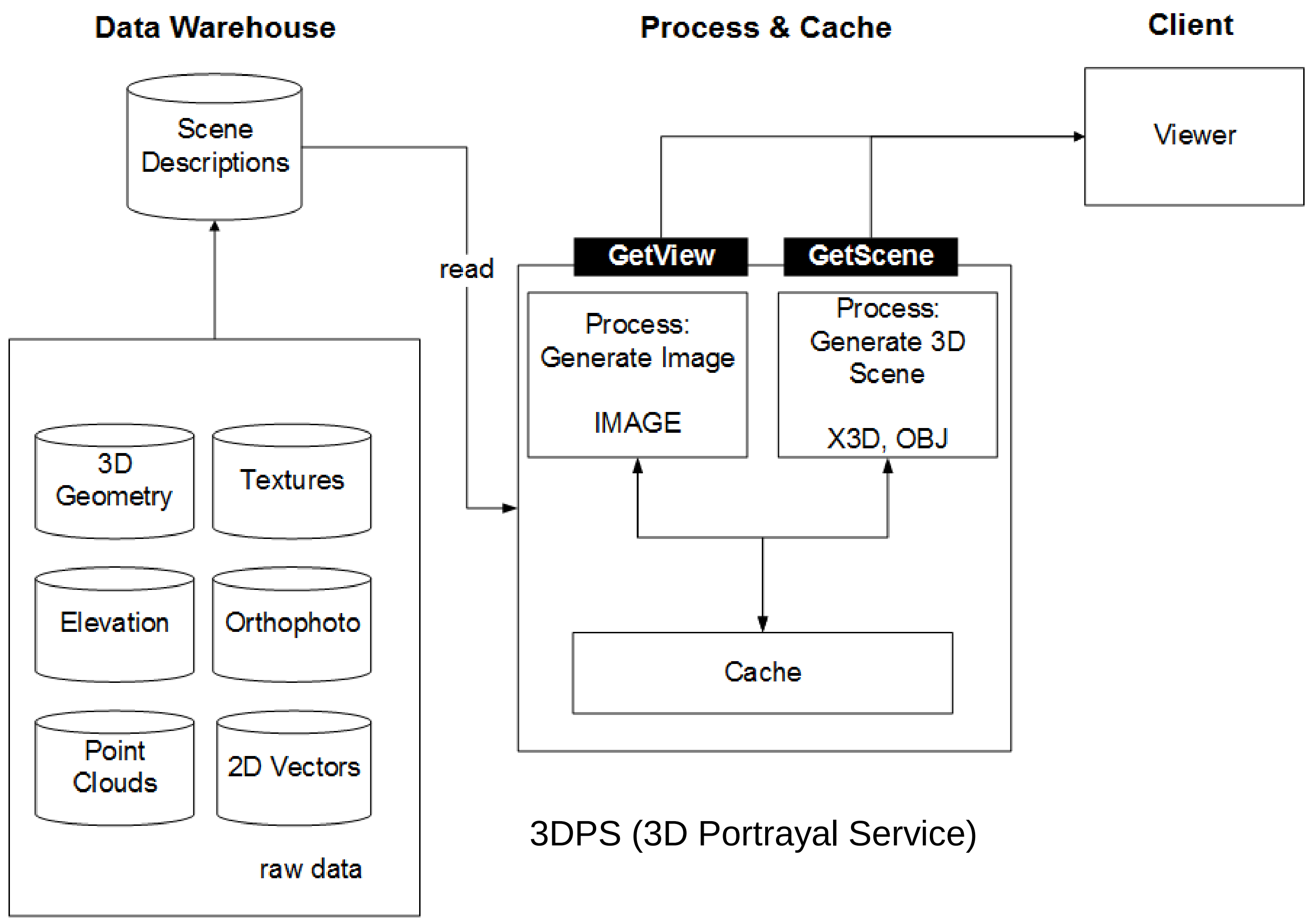
High Resolution Geometry doesn’t matter: Same download/render speed





MTh Markus Jung, 2014

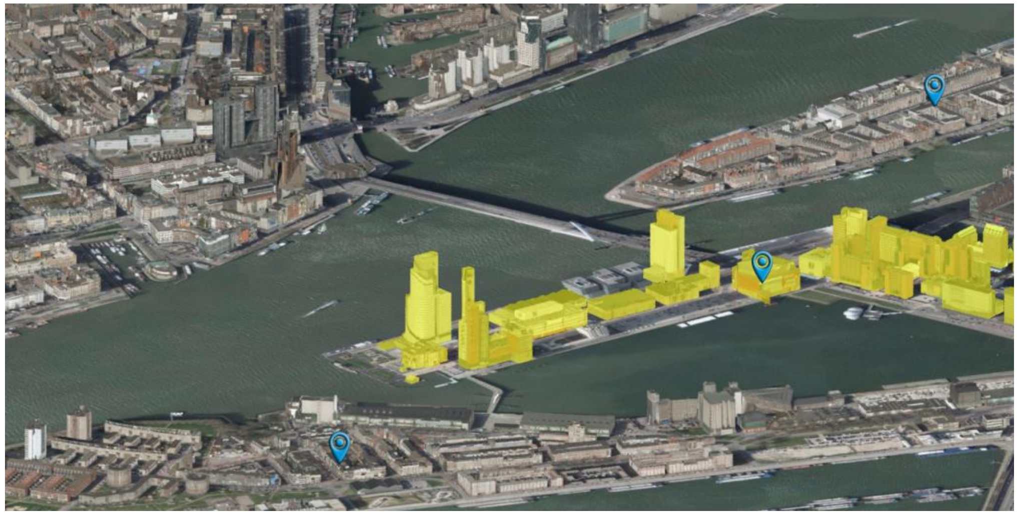
The 3dmaps.ch Project: Bringing it all together!



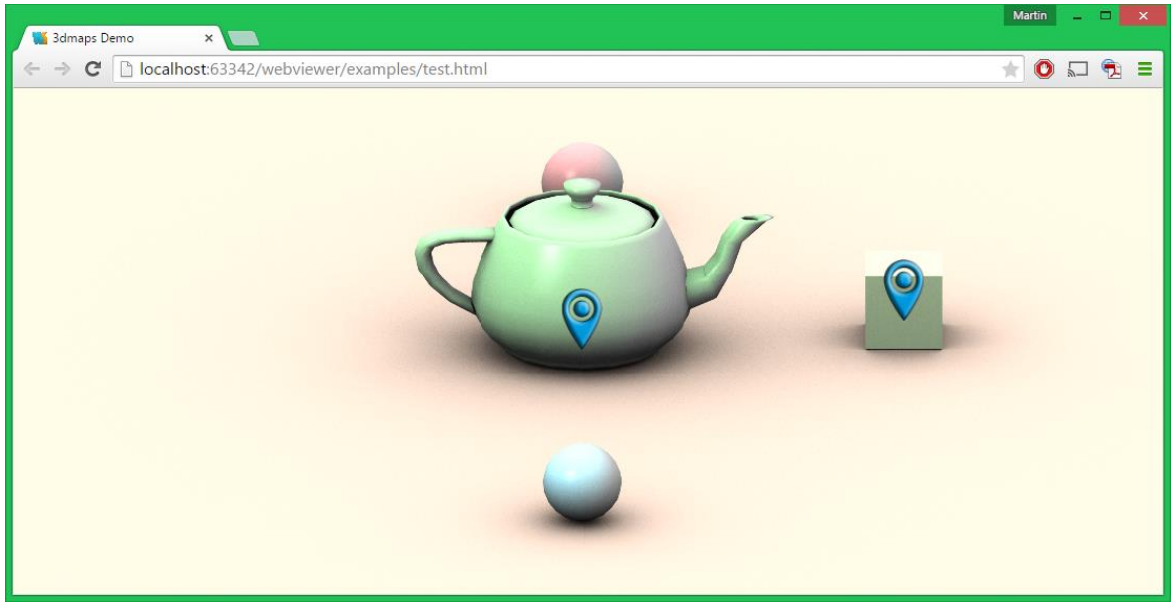
3DPS (3D Portrayal Service)

The Viewer

- The viewer basically uses the same concepts as a “2D Map Viewer”
- map3d.js supports WebGL
  There is also a pure canvas version available as fallback
- Operations like “highlighting” are highly customizable. Basically it is an image processing operation which runs on the GPU (WebGL Version). If there is no WebGL available, the operation is done using JavaScript.

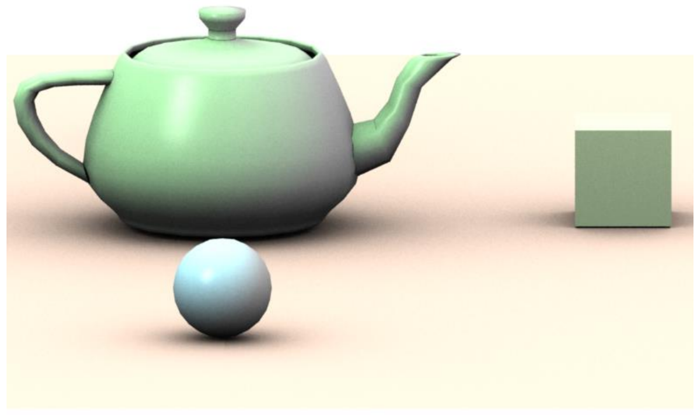


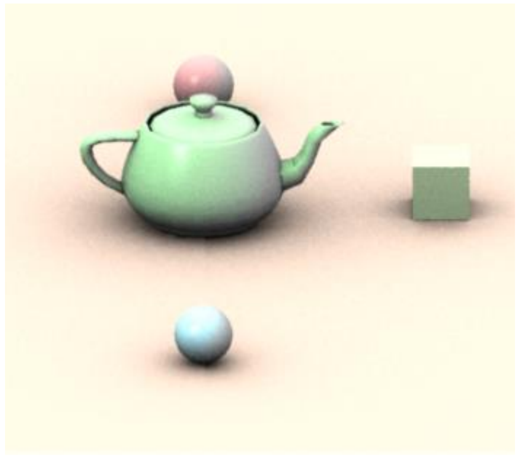
Viewer API
map3d.js Library

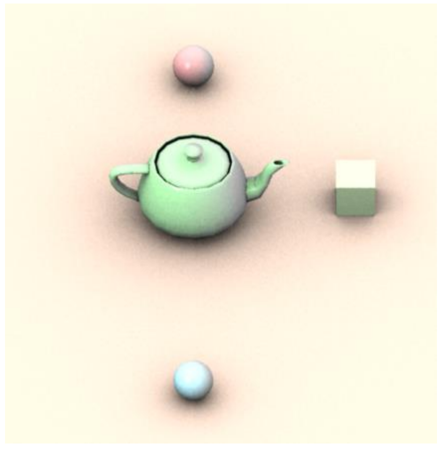


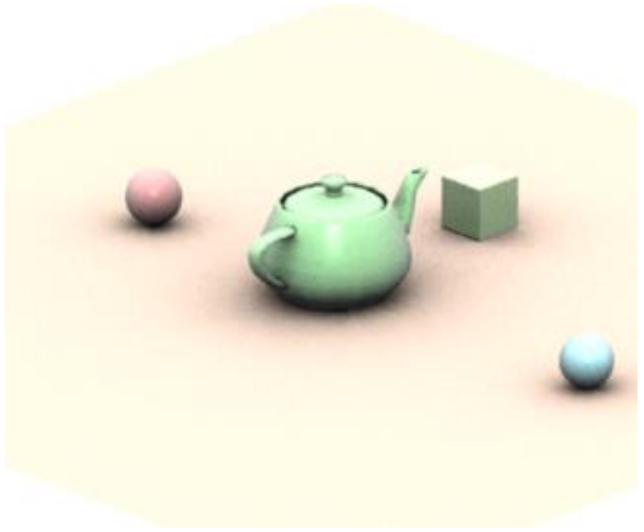
```
var map = new map3d.map("mapcanvas");

var layer = new map3d.imageLayer([
  "http://t1.3dmaps.ch/tiles/teatime",
  "http://t2.3dmaps.ch/tiles/teatime",
  "http://t3.3dmaps.ch/tiles/teatime",
  "http://t4.3dmaps.ch/tiles/teatime"])  });

layer.addTo(map);

var teapot_marker = new map3d.marker("Green Teapot", [0,0,0]);
teapot_marker.addTo(map);

var cube_maker = new map3d.marker("Green Cube", [80.5, 11.5, 10.5]);
cube_maker.addTo(map);
```

Different prerenderings for different pitch/view direction









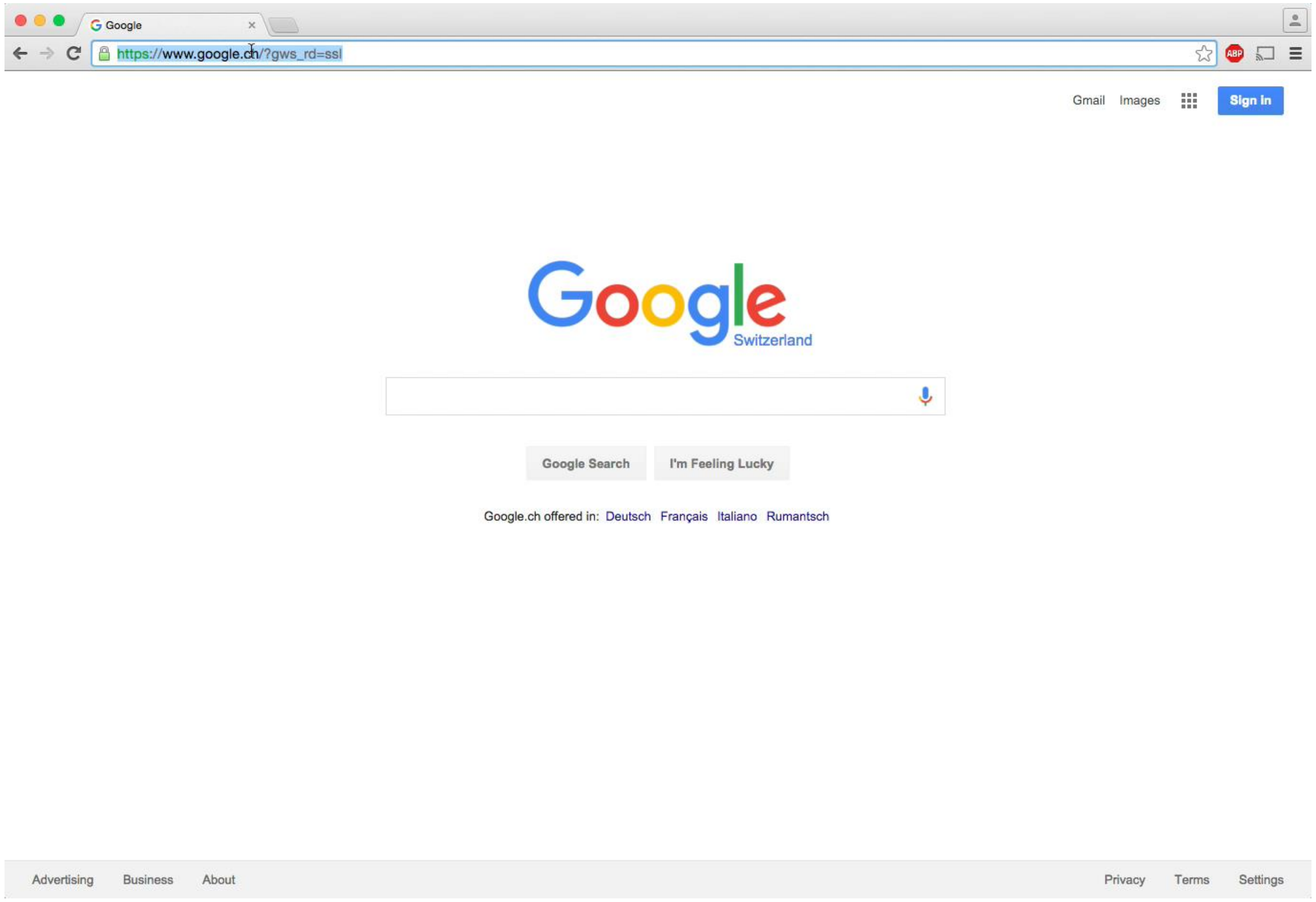
Every Prerendering needs storage, but with todays cloud storage pricing this is not really an issue anymore!

Use case 1: Rotterdam Dataset

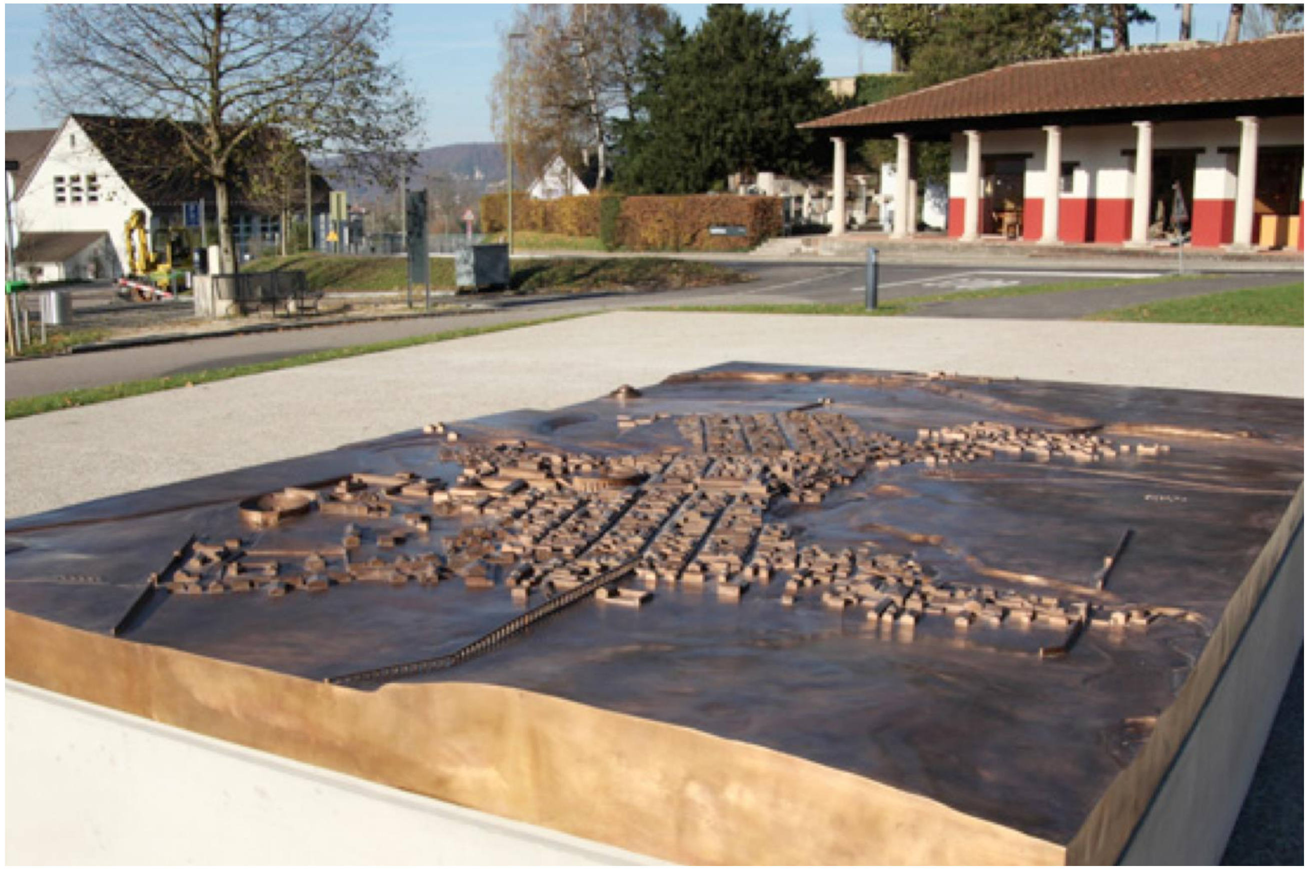
90 CityGML files with a total size of 2.72 GB

26'474 textures with a size of 1024x1024, an uncompressed total data volume of around 77 GB

Orthophoto uncompressed 430 GB



Use case 2: The Roman city of Augusta Raurica



A digital reconstruction of the historical Roman City of Augusta Raurica, created at the institute of Geomatics Engineering at the FHNW. 3D-Printed to create a bronze model.

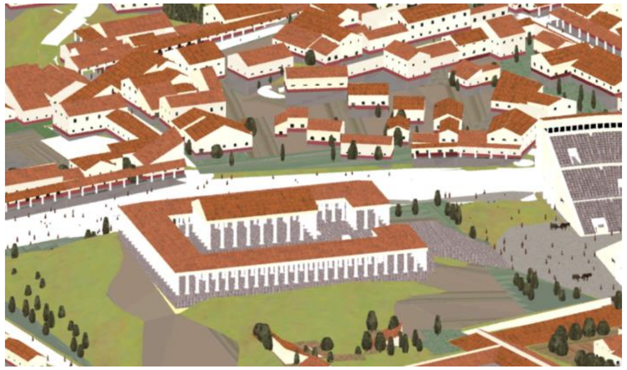
The 3D Model

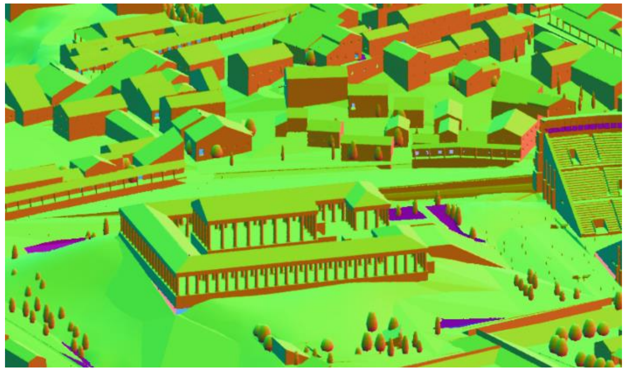
About 4000 geospatial objects (buildings, roads, vegetation features, terrain, …) at three levels of detail.

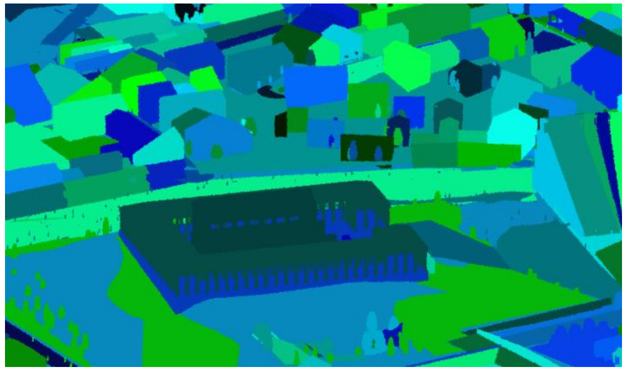
3D Geometry: 172 MB

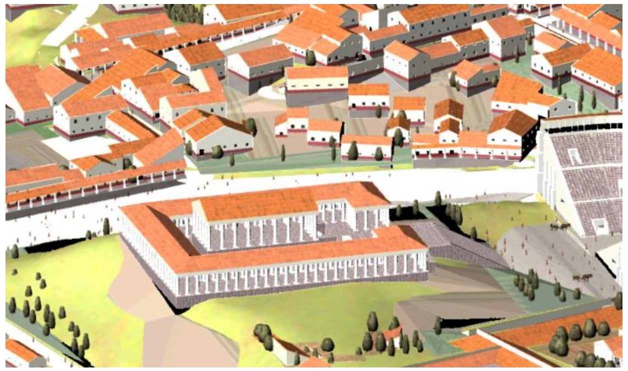
Textures: 350 MB

Prerendering the Model: Color Map, Normal Map, Id-Map, Depth Map







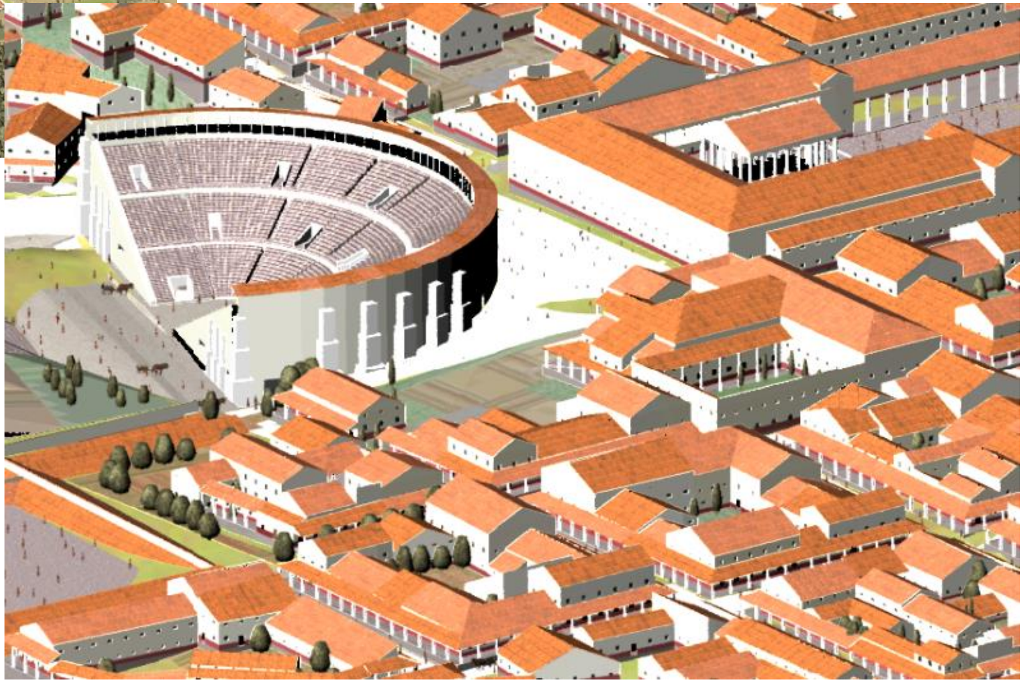


Dynamic Lighting

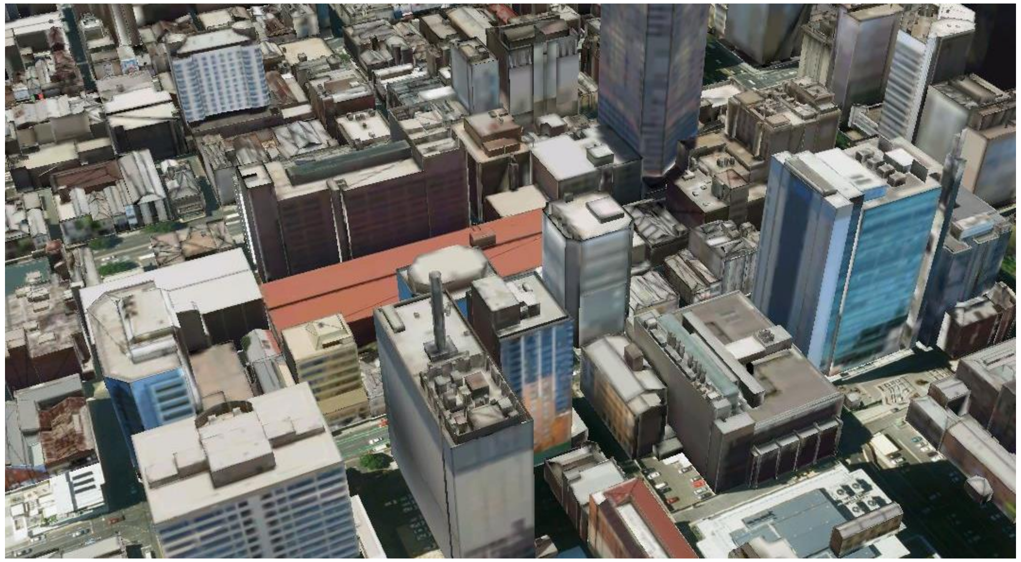
Normal Map: for Object Identification: Highlighting, special effects, …

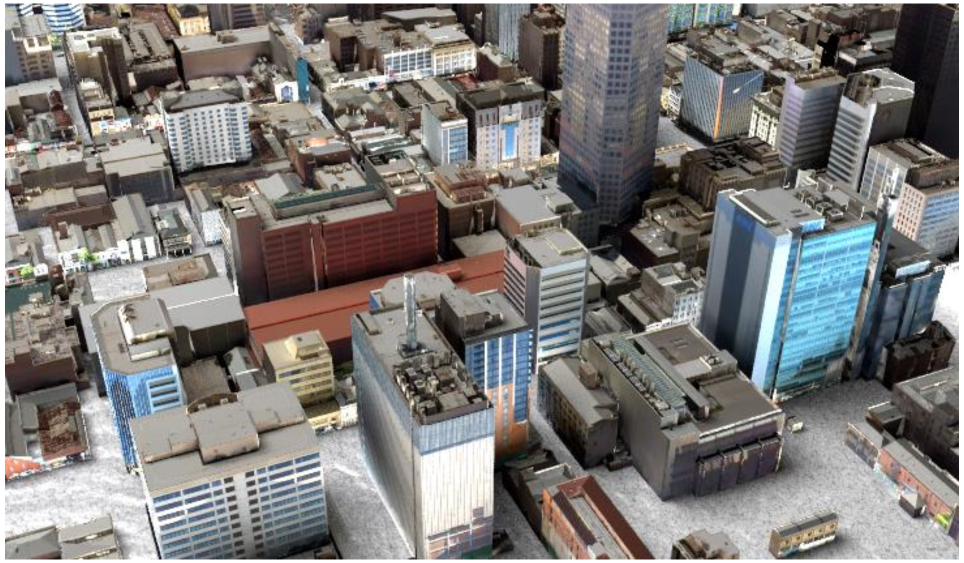
Depth Map: for 3D Position, special effects, …

3D View in the (mobile) webbrowser with dynamic Lighting



Use-Case 3: Comparing Visual Quality (Adelaide, Australia)


Klassischer 3D Webviewer


3D-Map

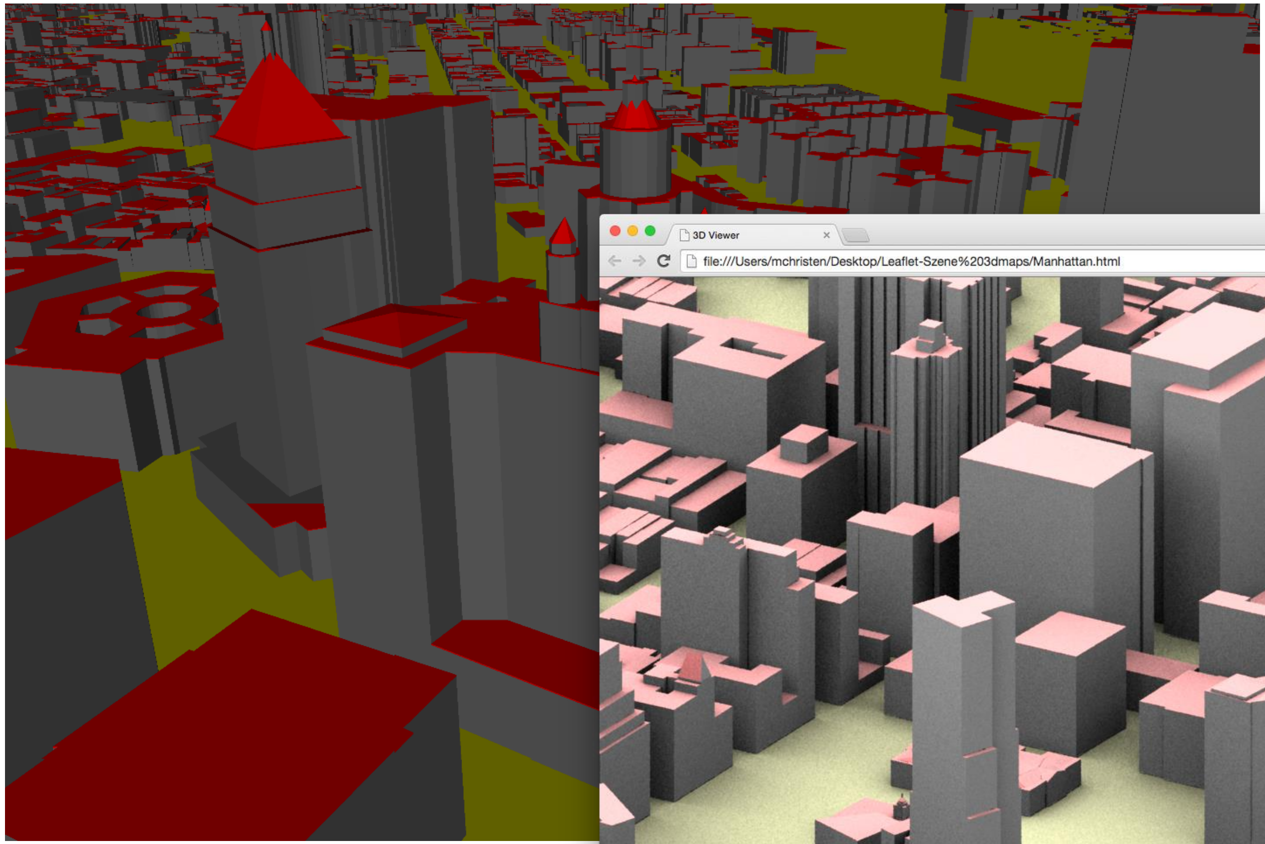
(MTh Christian Lindenberger 2015)

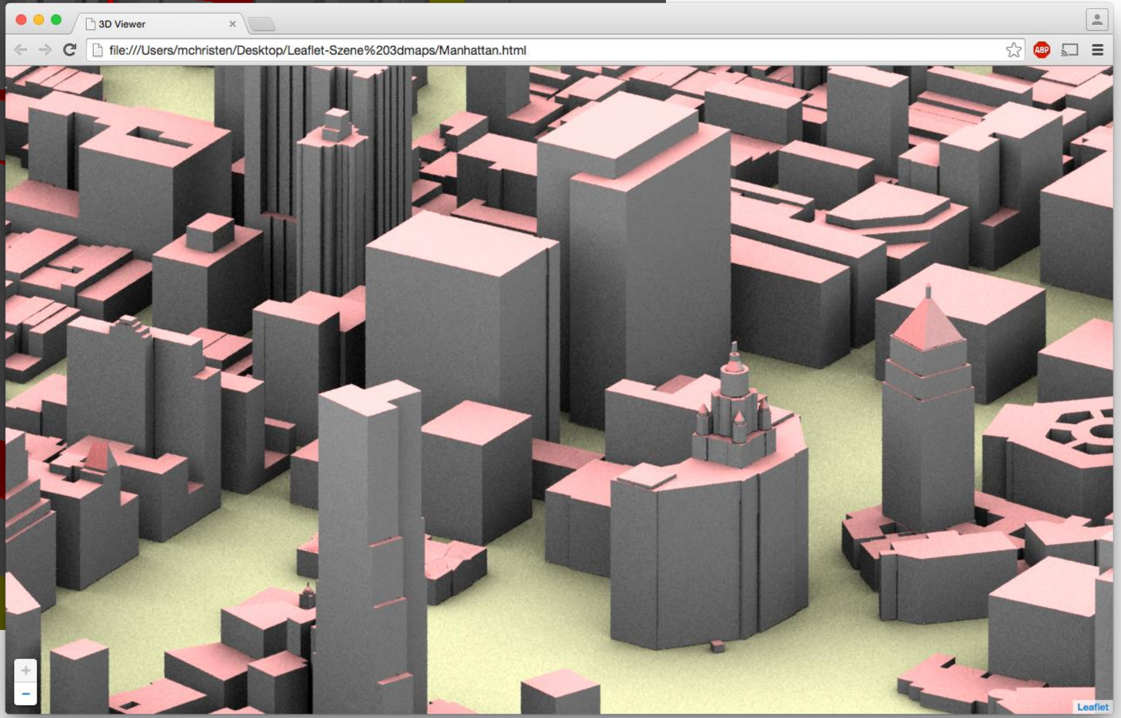
Outlook (1)

- Implement more effects and lighting models,
  dynamic snow/water/etc. using depth map & normal map
- Layer management (include point clouds, mix different layers using depth map
- Add realtime content ("mix real 3D Object") using depth-map
- Release map3d.js as Open Source (not before 2016)

Outlook (2)

More Rendering Effects, for example Screen Space Ambient Occlusion (SSAO)




3D Viewer
file:///Users/mchristen/Desktop/Leaflet-Szene%203dmaps/Manhattan.html
Leaflet

BTh, Daniel Rettenmund 2015

Outlook (3)

OpenWebGlobe 2

- “isometric maps” will be one of many features of OpenWebGlobe 2
- “isometric maps” will be the default 3D Viewer in OpenWebGlobe 2
- Switch to “real 3D” anytime
- State is saved:
  - The best matching viewpoint is selected when switching
  - If you highlight an object in “isometric mode”, it will be highlighted in “Real 3D” mode too.

Conclusion

- Using Image Provisioning (“isometric maps”) we have a “3D Streaming” with constant speed, no matter how complex the geometry/textures are.
- It runs on nearly every mobile phone with a webbrowser
- Thanks to cloud storage, we can have near unlimited number of views with different “look-at-angles”
- Navigation is really easy!
  Everyone who can handle 2D maps can handle isometric 3D Maps (pan & zoom)
- If WebGL is available it uses WebGL
  Otherwise: fallback to HTML5 Canvas.
- No limitation in numbers of users (viewers) at the same time

Questions