Fachhochschule Nordwestschweiz
Hochschule für Architektur, Bau und Geomatik

OpenWebGlobe

3D Visualisierung von OpenStreetMap Daten mit OpenWebGlobe

MapData © OpenStreetMap contributors

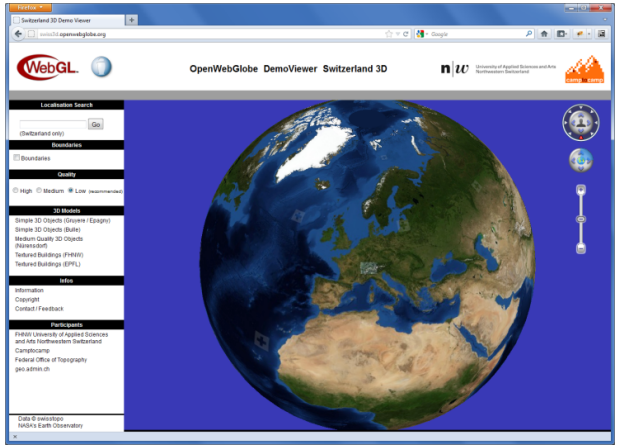
Martin Christen
Fachhochschule Nordwestschweiz
Institut Vermessung und Geoinformation

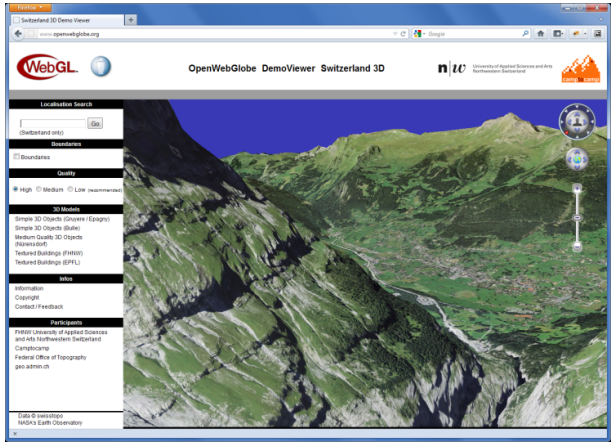
WebGL

- WebGL ist ein Industriestandard, welcher von der WebGL Working Group (Mitglieder: Apple, Google, Mozilla, Opera, ...) spezifiziert wird.
- WebGL erlaubt programmierbare 3D-Grafik im Web-Browser – ohne Verwendung von Plugins.
- WebGL basiert auf OpenGL ES 2.0, angepasst für JavaScript
- Die Spezifikation (Version 1.0) wurde am **3. März 2011** veröffentlicht**.**

OpenWebGlobe?

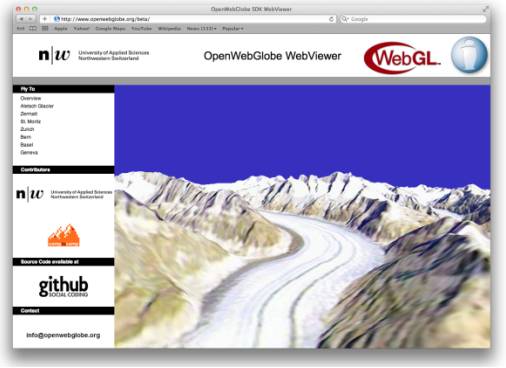
- Open Source Virtueller Globus auf Basis von WebGL/HTML5 zur Integration eines virtuellen Globus in eigene Web-Applikationen unter MIT Lizenz.
- Ermöglicht die Visualisierung grosser, eigener Bild- und Höhendatenbestände in einem virtuellen Globus
- Plug-In-frei im Browser ausführbar





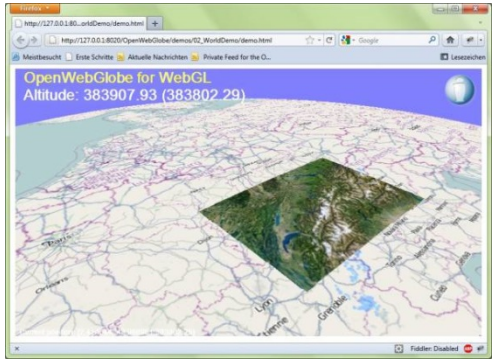
OpenWebGlobe Web Viewer

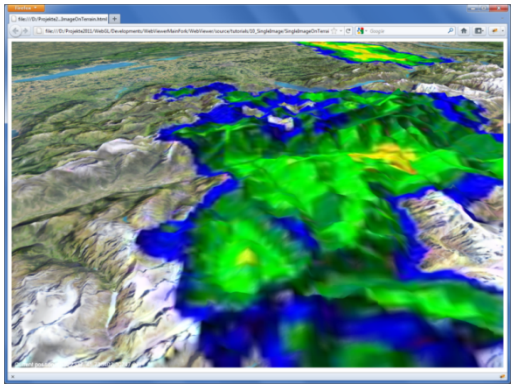
- JavaScript Bibliothek mit Funktionen zur Visualisierung der Daten, Kontrolle der Ansichten und zur Einbindung eigener Inhalte
- Läuft Plug-In frei in WebGL fähigen Browsern

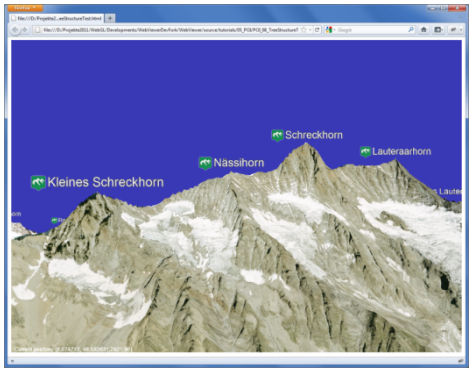


Layers / Contents

- Bild- und Höhendaten
- POIs
- 3D Modelle
- HTML5 Canvas Element
- Overlays
- Punktwolken







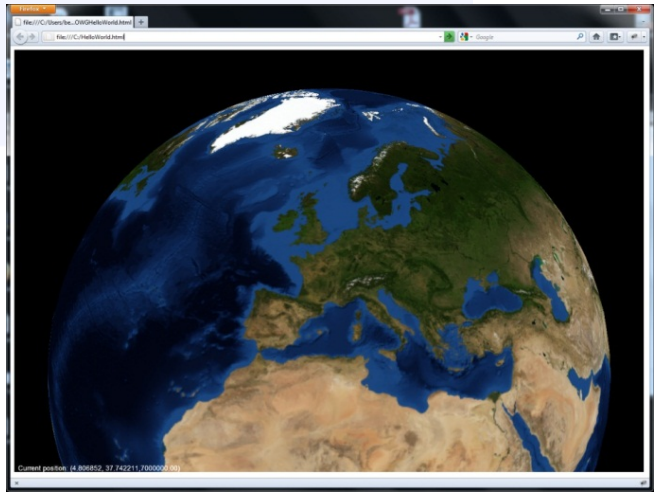
OpenWebGlobe: Datenprozessierung und Viewer

Die OpenWebGlobe SDK erlaubt dem Entwickler:

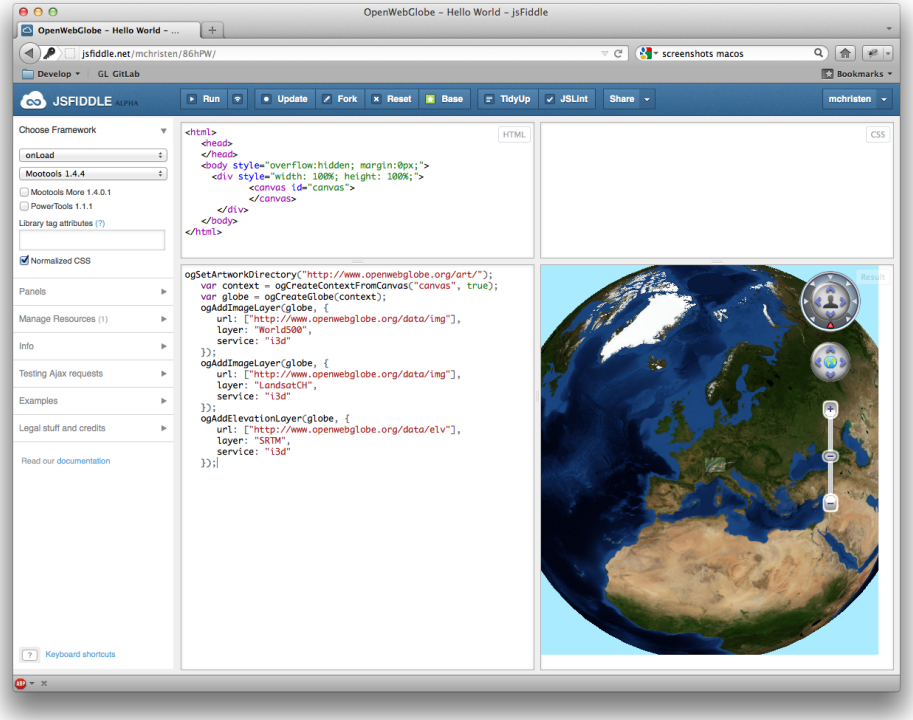
- Einfache Integration grosser, eigener Bild- und Höhendatenbestände
- Volle Kontrolle über die eigenen Daten und deren Aktualisierung
- Individuelles Datenhosting auf beliebigen Web-Servern
- Anpassung und Erweiterung der bestehenden Funktionalität dank offenem Quellcode möglich (MIT Lizenz)
- Einfache Implementierung eigener Applikationen mit virtuellem Globus

OpenWebGlobe in eigener Webseite

```
1  <html>
2  <script type="text/javascript" src="openwebglobe.js"></script>
3  <script type="text/javascript">
4
5  function main()
6  {
7     var ctx = ogCreateContextFromCanvas("canvas", true);
8     var globe = ogCreateGlobe(ctx);
9     var imgBlueMarble500 =
10    {
11       url     : ["http://www.openwebglobe.org/data/img"],
12       layer   : "World500",
13       service : "i3d"
14    };
15    ogAddImageLayer(globe, imgBlueMarble500);
16 }
17 </script>
18 </head>
19 <body onload="main()">
20    <canvas id="canvas"></canvas>
21 </body>
22 </html>
```



Erste Schritte



Entwickeln/Testen direkt im Browser!

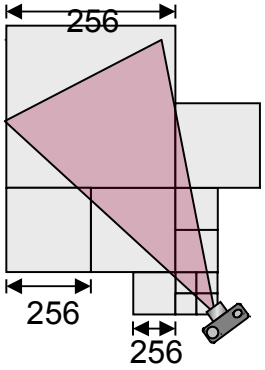
Hello World:
http://jsfiddle.net/mchristen/86hPW/

OpenStreetMap:
http://jsfiddle.net/mchristen/3htXe/

POI setzen:
http://jsfiddle.net/mchristen/rkqNy/

Latest OpenWebGlobe SDK: http://www.openwebglobe.org/js

Daten-Prozessierung mit Cloud



Download der sichtbaren Datenfragmente

Prozessierung und Caching von Tiles

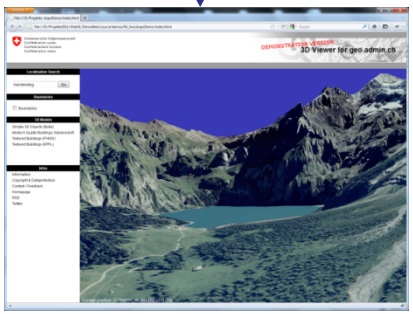
mongoDB

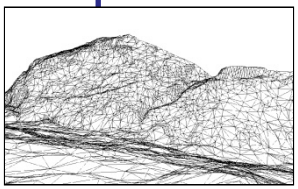
node.js

Web Viewer

- HTML5
- WebGL
- JavaScript

Upload **Rohdaten** in Cloud



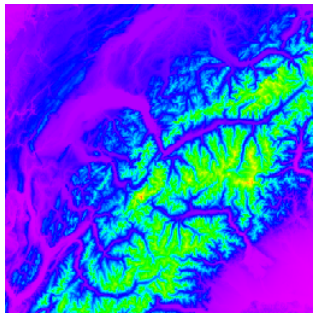


Bilddaten, Höhendaten,
3D-Modelle, Punktwolken, POI

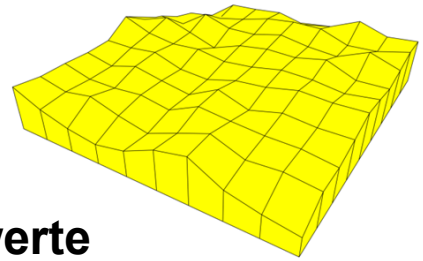
Prozessierung von Tiles?

2D Image Tile
RGB(A)
256x256 Pixels (meistens...)
LOD Beschränkung: Anzahl Pixel

2D Vector Tile
Enthält 2D Geometrie, aufgrund Performance wird es aber oft auch als Bild gerastert (oder Kombination)
LOD Beschränkung: Anzahl Vektoren (oder Pixel)



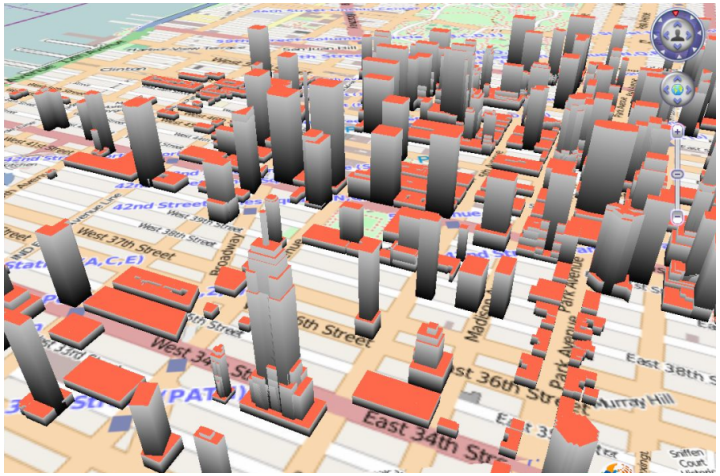
2D Elevation Tile (-> für 3D Tiles)
Höhenwerte
z.B. 17x17 Werte im Raster
LOD Beschränkung: Anzahl Höhenwerte

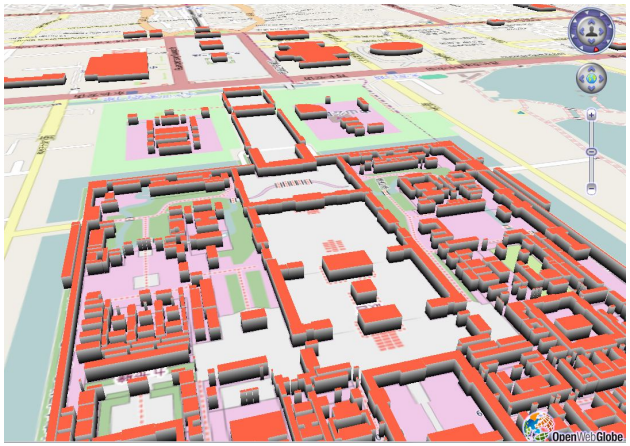


3D Geometrie Tile ?

- 3D Geometrie in Tile...
- Beliebige Geometrie (Vertex Buffer, Index Buffer, Texturreferenzen) in einer Kachel
- Bei Gebäuden: Schwerpunkt des Grundriss bestimmt welche Kachel verwendet wird
- **LOD Beschränkung: Anzahl Dreiecke pro Kachel** (und Anzahl Referenzen auf Texturen)

Streamen von Geometrie-Tiles (aus OSM Daten)



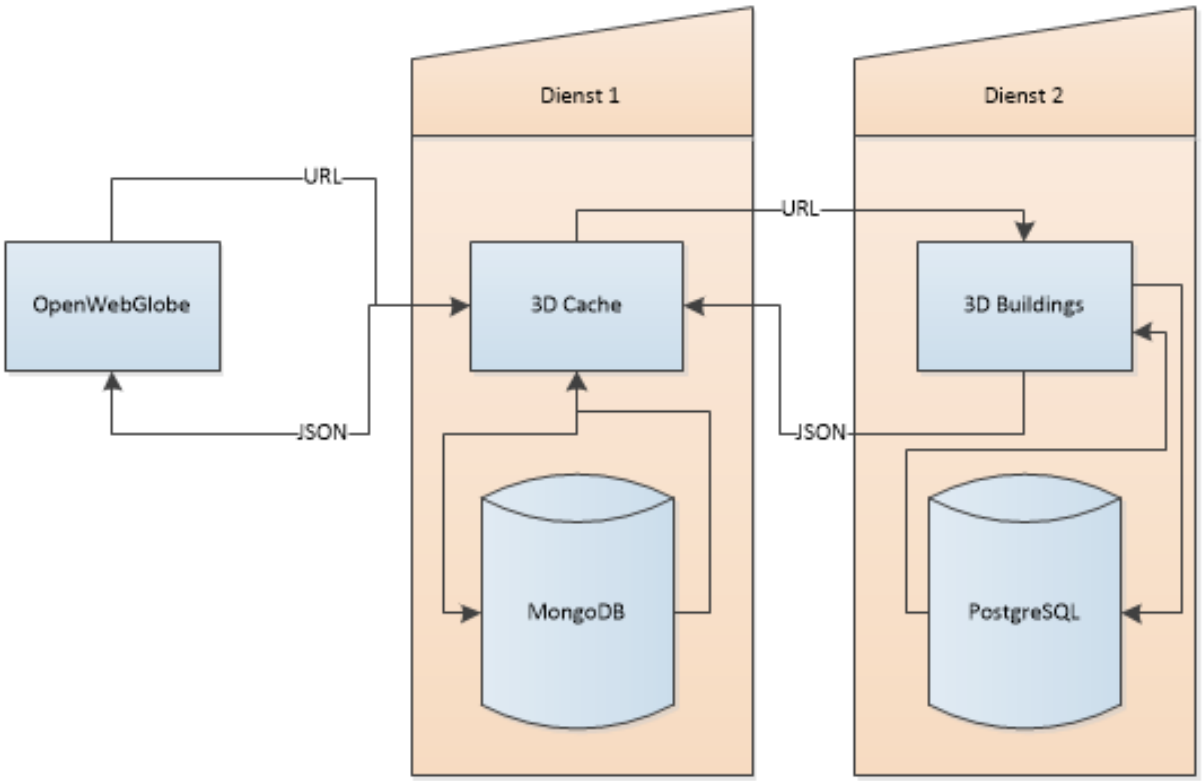


BTh Hürbi/Dätwyler, MTh L. Oertli

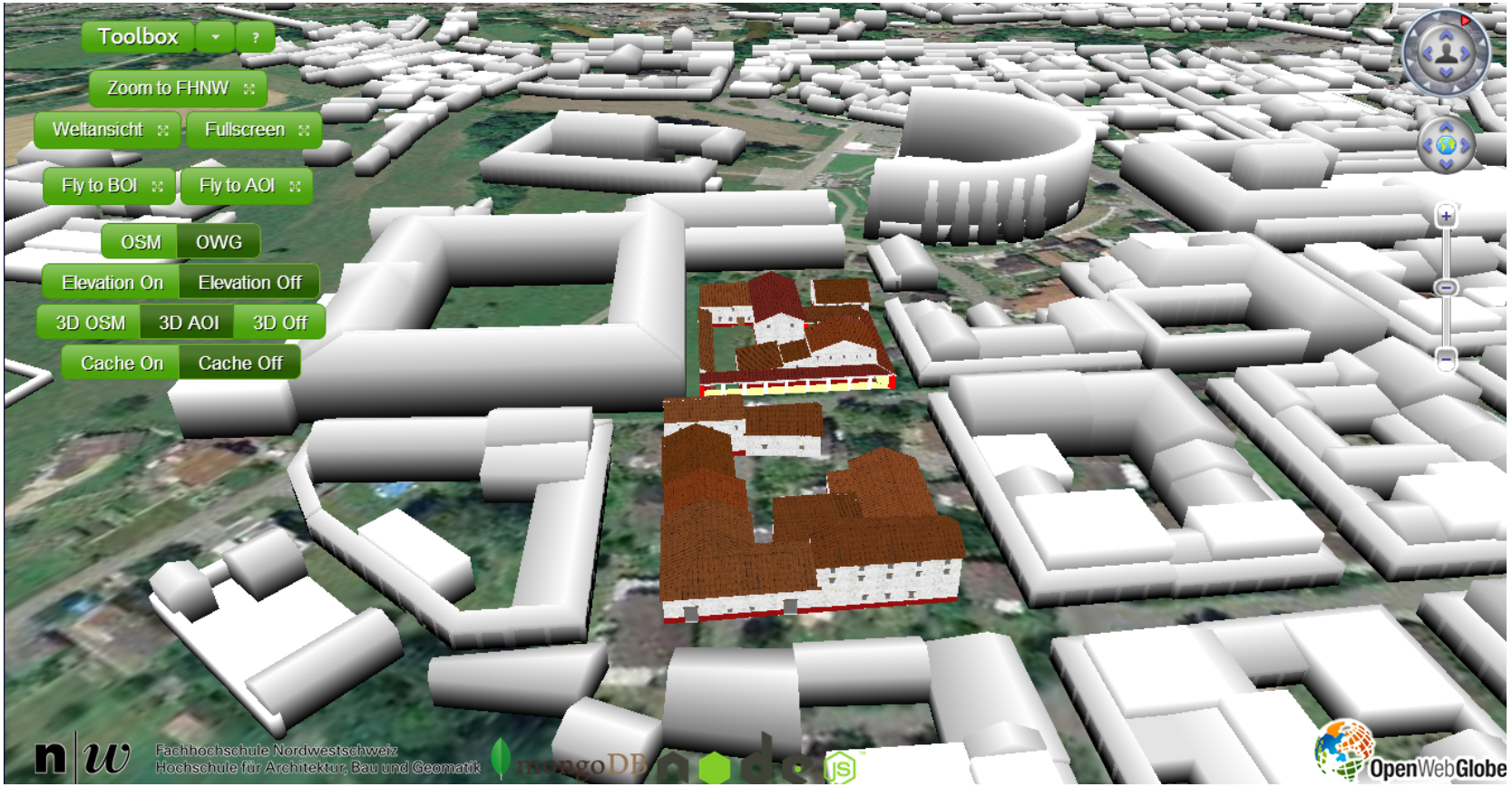
MapData © OpenStreetMap contributors

Kombination OSM & Modelliertes Modell

Aufbau des node.js Servers



Beispiel mit Historischen Daten



Zusammenfassung und Ausblick

- **Verwendung 3D Geometrie-Tiles für weltweites Rendering**
- **Vorrechnen/Caching von OSM Daten direkt aus Postgres Datenbank**
- **Caching über MongoDB (oder S3)**

Weitere Entwicklungen in Planung:

- Verbesserte Darstellung von OSM Gebäuden (z.B. Dachformen)
- Einbezug anderer Daten (z.B. Strassen, Bäume etc.)

Vielen Dank

- http://www.openwebglobe.org
- https://github.com/OpenWebGlobe