

OpenWebGlobe: 3D-Visualisierung und Caching von globalen Stadtmodellen aus OpenStreetMap mittels Cloud-basiertem Framework

MARTIN CHRISTEN, KEVIN HÜRBI¹, STEPHAN NEBIKER¹

Proprietäre Virtuelle Globen wie Google Earth haben 3D Geodaten für ein breites Publikum geöffnet. Inzwischen gibt es auch Open Source Lösungen von virtuellen Globen, welche dem Betreiber unter anderem die Kontrolle über eigenen Geodaten ermöglichen. Die Erweiterbarkeit dieser proprietären Technologien um eigene Geobasisdaten wie z.B. verbesserte, hoch detaillierte Höhenmodelle oder globale 3D-Stadtmodelle, oder um andere eigene Funktionalität ist jedoch sehr eingeschränkt. Das quelloffene OpenWebGlobe Software Development Kit (www.openwebglobe.org) ermöglicht die Realisierung massgeschneiderter, leistungsfähiger Virtueller Globen mit lokalen bis globalen Geodatensätzen und eigener Funktionalität im Bereich Visualisierung und Prozessierung. Der OpenWebGlobe Viewer basiert auf den neuen Web-Technologien HTML5 und WebGL ist daher in den meisten modernen Web-Browsern ohne vorgängige Installation eines Plugins lauffähig. Die OpenWebGlobe prozessierung basiert

In diesem Paper stellen wir die neusten Entwicklungen des OpenWebGlobe Projekts vor. Es bietet ein hoch skalierbares, Cloud-basiertes Framework für den Aufbau und Betrieb von 3D Geoportalen mit selbst gewählten raumbezogenen Inhalten. Als Beispiel wird die globale 3D-Visualisierung von OpenStreetMap-Daten mit der on-the-fly-Cloud-basierten Prozessierung gezeigt - ein Ansatz für die Echtzeit-Verarbeitung und Caching von solchen hoch skalierbaren 3D-Szenen auf einem Web-basierten virtuellen Globus darstellt. Ein weitere Aspekt sind verschiedene Cloud-Computing-Funktionen, welcher für die Verarbeitung und visualisierung grosser Mengen von 2D- und 3D-Kartendaten an eine sehr grosse Menge an Clients bereitstellt. Außerdem wird gezeigt, wie ein traditionelles In-house Rechenzentrum mit Cloud-basierten Dienste für Caching kombiniert werden kann und eine sehr kostengünstig Methode für die Bereitstellung von grossen Mengen an up-to-date in 2D- und 3D Kartendaten ist..

1 Einleitung

Virtuelle Globen benötigen Gigabytes von Bild- und Höhendaten, welche vom Internet gestreamt und dargestellt werden müssen. Um die Downloadzeit so kurz wie möglich zu halten, werden nur die im Browserfenster aktuell sichtbaren Bild-, Höhendaten und 3D-Objekte in geeigneten Auflösungsstufen angefordert und heruntergeladen. Das OpenWebGlobe Projekt (www.openwebglobe.org) wurde an der FHNW Fachhochschule Nordwestschweiz am Institut für Vermessung und Geoinformatik (IVGI) initiiert. Als Open Source Projekt wurde es im April 2011 gestartet und verfügt über Mitentwickler von anderen Universitäten und aus der Privatwirtschaft (CHRISTEN, M., NEBIKER, S., 2010). Offene Datengrundlagen sind jedoch nach wie vor Mangelware. Dank Crowd-Sourcing Projekten wie OpenStreetMap (OPENSTREETMAP,

¹ Martin Christen, Kevin Hürbi, Stephan Nebiker, FHNW Fachhochschule Nordwestschweiz, Institut Vermessung und Geoinformation, Gründenstrasse 40, 4132 Muttenz, Schweiz; E-Mail: martin.christen@fhnw.ch, kevin.huerbi@fhnw.ch, stephan.nebiker@fhnw.ch

2014) gibt es freie weltweite Daten. Aus diesen Daten können neue Darstellungen realisiert werden. OpenStreetMap ist grundsätzlich ein 2D-Datensatz. Kombiniert man die Daten mit anderen freien Datensätzen so können beispielsweise weltweite 3D-Stadtmodelle realisiert werden. Um diese enormen Datenmengen auf einem Globus darzustellen sollte folgendes beachtet werden: Es muss ein Ansatz zur effizienten Datenübertragung (Streaming) realisiert werden um die Daten überhaupt darstellen zu können und die Daten müssen serverseitig schnell bereitgestellt werden können, möglichst durch ein Caching-Verfahren. In diesem Paper werden nun diese beiden Aspekte durchleuchtet.

2 Streaming von OSM Daten

Wie die meisten Kartendienste lädt OpenWebGlobe die Daten mittels Kacheln eines Quadrees in der sphärischen Mercator-Projektion (SNYDER, J. P., 1987). Sobald der Nutzer eine definierte Zoomstufe erreicht, werden die 3D-Gebäudemodelle für jede Kartenkachel serverseitig erstellt. Damit die Abfrage von Hunderten bis Tausenden von 3D-Gebäudemodellen pro Kartenkachel realisiert werden kann, muss die Serverkomponente möglichst skalierbar sein. Der Serverdienst wurde mit node.js erstellt und komplett in JavaScript implementiert.

OpenWebGlobe übergibt zuerst die Koordinaten der aktuell geladenen Kartenkacheln an das Berechnungsskript. Das Programm holt die Gebäudegrundrisse, ermittelt die jeweilige Terrainhöhe und Gebäudehöhe, berechnet die 3D-Gebäudemodelle und gibt diese schlussendlich dem Webbrowser im OpenWebGlobe-Format zurück. In der Abbildung 1 ist die Architektur der Webapplikation in der Übersicht dargestellt.

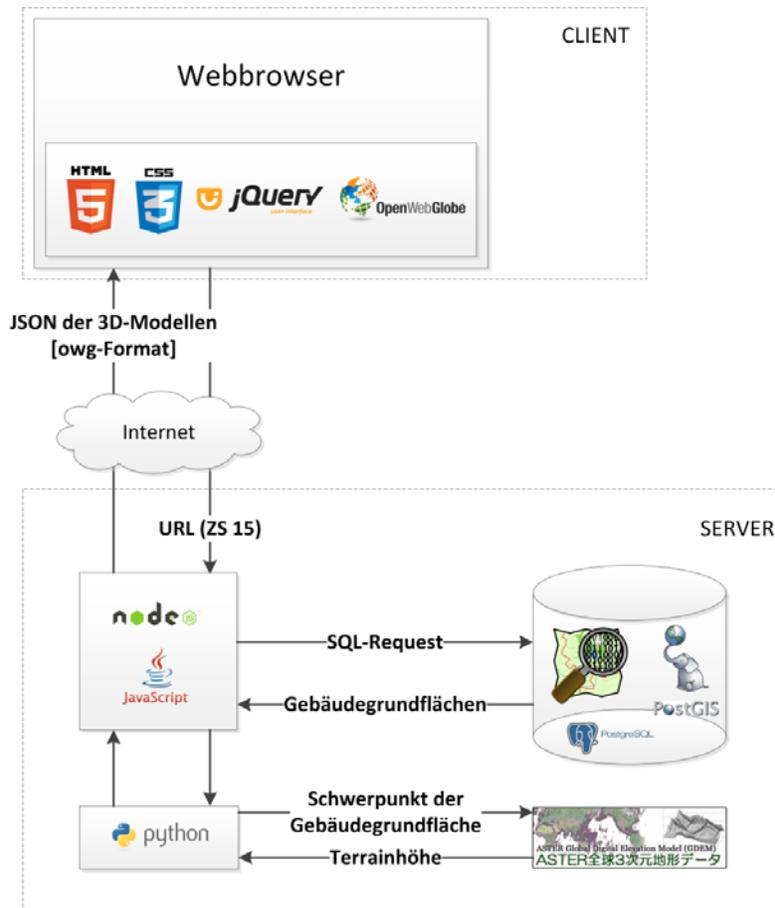


Abb. 1: Architektur der Webapplikation

Sämtliche Gebäudegrundflächen im gewünschten Koordinatenausschnitt werden von OpenStreetMap aus den PostgreSQL- und PostGIS-Datenbanken bezogen. Die Berechnung des 3D-Gebäudemodells erfolgt in der node.js-Applikation iterativ über jede Grundfläche aus der SQL-Abfrage. Damit aus der 2D-Grundfläche ein 3D-Modell gerechnet werden kann, ist eine Höheninformation (Kapitel 3.2: Bestimmung der Gebäudehöhe) nötig. Für die Erstellung der 3D-Gebäudemodelle wird die extrahierte Grundfläche kopiert und bei der Z-Koordinate der Höhenwert addiert. Daraus resultiert die Grund- und die Deckfläche eines 3D-Modells. Zum Schluss werden die Mantelflächen der Aussenwände und von allfälligen Innenhöfen hinzugefügt.

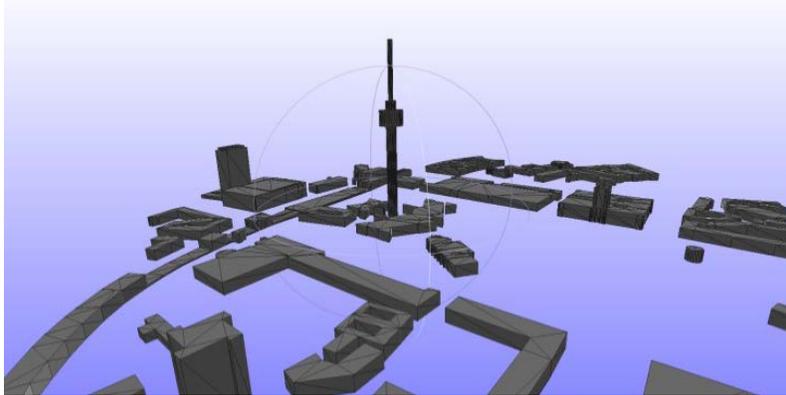


Abb. 2: 3D-Gebäudemodelle in MeshLab visualisiert

Damit die 3D-Gebäudemodelle im OpenWebGlobe angezeigt werden können, werden sie in einem internen und effizienten JSON basierten Format ausgegeben. In diesem JSON-Objekt wird ein weiteres Objekt pro 3D-Gebäudemodell erstellt. Dieses Modell enthält u.a. eine Liste mit den 3D-Koordinaten der Modelleckpunkte. Die Flächen der 3D-Gebäudemodelle werden als Dreiecksflächen dargestellt. Die Triangulation der Flächen erfolgt mit der JavaScript-Bibliothek poly2tri. (POLY2TRI, 2014). Sämtliche Flächen im JSON-Objekt werden somit als Dreiecke gespeichert und müssen Client-Seitig nicht mehr optimiert werden. (DÄTWYLER M., HÜRBI K., 2013).

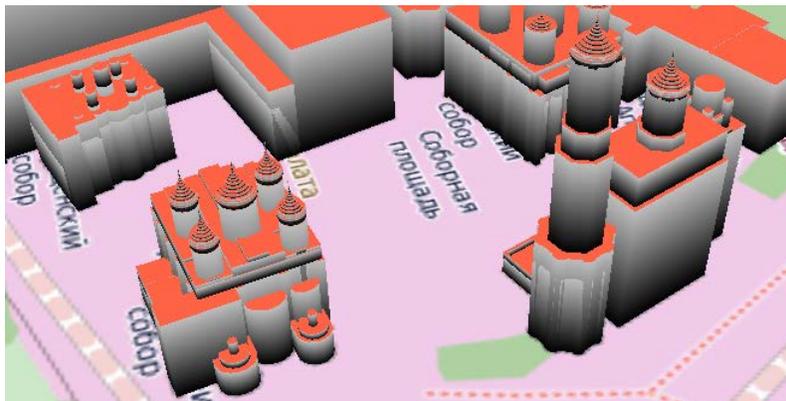


Abb. 3: 3D-Gebäudemodelle in OpenWebGlobe

Die OpenStreetMap Daten sind jedoch im Moment nicht für 3D Daten ausgelegt, deswegen ist es schwierig, gute 3D Modelle zu erstellen. Bei einigen Gebäuden haben sich die freiwilligen OSM Erfasser sehr grosse Mühe gegeben, wie es am Beispiel der Cathedral of Christ the Savior in Moskau (Abbildung 3) zu sehen ist. Im Moment werden in OpenStreetMap geschwungene Seitenflächen in der Vertikalen nicht unterstützt, deshalb wurde ein weiterer Workflow erstellt, welche 3D Modelle aus anderen Quellen in den Service integrieren kann (OERTLI, L., 2013)



Abb. 4: Beispiele für Gebäude aus anderen Quellen

3 Caching der 3D-Kacheln

Die Datenbankabfrage für die Gebäudegrundrisse aus der OpenStreetMap aus der PostgreSQL Datenbank ist relativ zeitaufwändig und vor allem nicht so einfach skalierbar. Ein möglicher Ansatz zur Verbesserung der Performance ist das Speichern sämtlicher Abfragen. Die Abfragen könnten in einzelnen Files gespeichert werden, jedoch würde dies bei grösseren Mengen von Einzeldateien zu Problemen führen (CHRISTEN, M., NEBIKER, S., 2011A).

Der Ablauf des Caching wird in Abbildung 5 gezeigt und läuft folgendermassen ab: Zunächst wird eine 3D-Objekt-Kachel angefordert. Ist diese bereits im Cache vorhanden und ist das optionale Ablaufdatum noch nicht erreicht, so wird diese Kachel aus der NoSQL Datenbank MongoDB (MONGODB, 2014) gelesen und zurückgegeben. Ist die Kachel noch nicht im Cache, so wird diese aus dem Grundriss der OpenStreetMap Daten generiert. Optional können einzelne Gebäude durch besser aufgelöste 3D-Objekte ersetzt werden. Dies wurde beispielsweise für Sehenswürdigkeiten wie die Freiheitsstatue in New York, die Pyramiden in Gizeh, der Tower in London usw. realisiert. In diesem Fall wurden die Daten vorgängig in einem 3D-Modellierungspaket optimiert und in direkt in der MongoDB - also im Cache - gespeichert. Somit könnten alle interessanten Gebäude weltweit zusätzlich modelliert und mit OpenWebGlobe dargestellt werden. Ein Vorteil dieses Ansatzes ist auf die On-The-Fly Prozessierung der Daten - somit wird eine vorgängige Datenprozessierung (CHRISTEN, M., 2012) überflüssig.

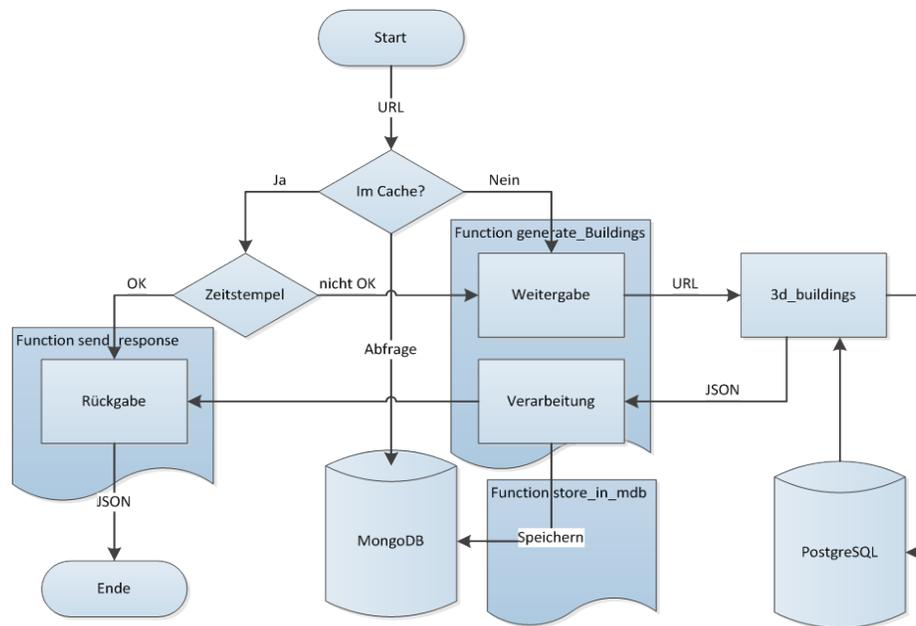


Abb. 5: Ablauf für das Caching der 3D-Tiles

4 Fazit und Ausblick

Proprietäre virtuelle Globen ermöglichen eine Reihe von Anwendungen und sind einfach zu bedienen. Wenn jedoch grosse Mengen eigener Daten integriert und spezifische Anwendungen oder Dienstleistungen erstellt werden möchte, stösst man rasch an technische und organisatorische Grenzen. Der quelloffene OpenWebGlobe auf der Basis von HTML5 und WebGL bietet da eine Alternative. Das SDK ermöglicht die massgeschneiderte Erstellung eines virtuellen Globus und der darauf basierenden Anwendung oder Dienstleistung mittels eigener oder frei verfügbarer Geobasisdaten. Mit OpenWebGlobe können verschiedene Anwendungen realisiert werden. So wurde gezeigt, dass es möglich ist einen globalen Layer mit Stadtmodellen aus OpenStreetMap zu integrieren. Um die Performance zu verbessern wurde auch ein Caching Mechanismus eingebaut. Eine NoSQL Datenbank wie MongoDB eignet sich sehr gut zur Speicherung von Key-Value Paaren, in diesem Fall von gekachelten 3D Geometrien. In Zukunft sollen auch andere Kacheltypen (insbesondere Bild-, Höhen- und Vektor-Kacheln) in MongoDB gespeichert werden.

5 Literaturverzeichnis

- CHRISTEN, M., 2014: OpenWebGlobe - eine SDK zur Darstellung massiver Daten im Webbrowser dank WebGL, WebGL Publishing
- SNYDER, J. P. 1987: Map Projections: A Working Manual, U.S. Geological Survey Professional Paper 1395, U.S. Geological Survey.
- CHRISTEN, M., NEBIKER, S., 2010. Neue Entwicklungen im Bereich Virtueller Globen am Beispiel der i3D-Technologie. Dreiländertagung der OVG, DGPF und SGPF. Wien

- CHRISTEN, M., NEBIKER, S., 2011A. Large Scale Constraint Delaunay Triangulation. *Lecture Notes in Geoinformation and Cartography*. Springer, Heidelberg, pp. 57-72.
- CHRISTEN, M., NEBIKER, S., 2011B. OpenWebGlobe SDK, an Open Source High Performance Virtual Globe SDK for Open Maps. *Proceedings of the 1st European State of the Map*, Wien.
- CHRISTEN, M., 2012. Processing Geodata using the OpenWebGlobe Tools. <https://github.com/downloads/OpenWebGlobe/DataProcessing/dataprocessing.pdf> (02.04.2012)
- DÄTWYLER M., HÜRBI K., 2013. 3D-Visualisierung von Gebäuden aus OpenStreetMap in OpenWebGlobe, Bachelor Thesis, FHNW - Fachhochschule Nordwestschweiz, Institut Vermessung und Geoinformation
- OERTLI, L., 2013. OpenWebGlobe : Visualisierung und Caching von 3D-Stadtmodellen, Master Thesis, FHNW - Fachhochschule Nordwestschweiz, Institut Vermessung und Geoinformation
- WÜEST, R., 2012. Paralleles Pre-processing und optimiertes Rendering globaler Openstreetmap-Daten in OpenWebGlobe, 2012. Master Thesis, FHNW - Fachhochschule Nordwestschweiz, Institut Vermessung und Geoinformation

Online Quellen:

- MONGODB, 2014: NoSQL Datenbank, <http://www.mongodb.org/> (letzter Zugriff: 10.1.2014)
- OPENSTREETMAP, 2014: OpenStreetMap Project Wiki, URL: http://wiki.openstreetmap.org/wiki/Main_Page (letzter Zugriff: 1.2.2014)
- POLY2TRI, 2014: <https://github.com/r3mi/poly2tri.js> (letzter Zugriff: 25.1.2014)
- WEBGL 2011, 2011: WebGL Specification, Version 1.0, 10 February 2011. URL: <https://www.khronos.org/registry/webgl/specs/1.0/> (letzter Zugriff: 20.01.2014).