

# Cloud basiertes Rendering von komplexen 3D-Stadtmodellen

MARKUS JUNG<sup>1</sup>

*Zusammenfassung: Dreidimensionale Stadtmodelle zeichnen sich meist durch eine sehr große Komplexität aus. Insbesondere aufgrund ihrer Texturen summierte sich ein solches Modell zu sehr großen Datenmengen. Für die Darstellung solch komplexer und datenintensiven Stadtmodellen werden entsprechend viel Ladezeit, sowie Speicherkapazität und Rechenpower benötigt. Im Rahmen meiner Masterthesis wurde ein Verfahren entwickelt um komplexe Modelle effizient über einen Webbrowser auf beliebigen Endgeräten darzustellen. Dazu wurde ein Prototyp implementiert. Das Rendering mittels Raytracing Verfahren, welches nicht auf dem Client, sondern auf dem Server ausgeführt wird, ermöglicht die Generierung sehr realitätsgetreuer Bilder. Zudem lässt sich dieses Verfahren in einer Cloudumgebung sehr gut parallelisieren und ist somit sehr gut skalierbar. Beim Image Based Provisioning wird nicht nur ein Farbbild generiert, sondern weitere Layer mit Informationen zur Tiefe, Normalen oder Objektzugehörigkeit jedes Pixels erstellt. Das Modell wird dabei in mehreren Bildpyramidenstufen gerendert und in Kacheln fixierter Größe unterteilt. Dadurch müssen auf den Client nur Bilder übertragen werden, welche konstant und damit kalkulierbar in ihrer Datenmenge sind. Mit diesem Verfahren lässt sich die Komplexität des 3D Modells von der Darstellung entkoppeln und der Betrachter kann trotzdem im Modell navigieren (Zoom, Pan) und Koordinatenmessungen oder Attributabfragen durchführen.*

## 1 Einleitung und Motivation

In den letzten Jahren wurden verschiedene webbasierte 3D-Geodienste, insbesondere auch virtuelle Globen entwickelt (CHRISTEN et al. 2012; COZZI & BAGNELL 2013). Mit diesen Geobrowsern konnte unter anderem gezeigt werden, dass es möglich ist im Webbrowser 3D-Stadtmodelle darzustellen. Aufgrund der teilweise sehr hohen Detailtreue weisen solche Modelle eine sehr große Datenmenge auf. So können 3D Modelle aus mehreren Tausenden oder Millionen Objekten und Texturen bestehen. Trotz Ansätzen mit Level of Detail oder dem dynamischen Nachladen räumlich aufgeteilter Segmente wie Kacheln, können komplexere 3D-Stadtmodelle zu erheblich längeren Ladezeiten und schlechterer Performance in der Darstellung führen (CHRISTEN et al. 2014; KLIMKE et al. 2014).

Auch Hardwareressourcen auf Smartphone und Tablets sind begrenzt nutzbar für eine performante interaktive 3D-Visualisierung. Für einen möglichst vielfältigen Nutzen der Geodaten ist es jedoch gerade wichtig, dass die Verteilung der Daten über verbreitete Hardware, Software und Netzwerke erfolgen kann. Die meisten Personen verfügen heute über mehrere mobile Geräte wie Notebooks, Tablets oder Smartphones und tragen diese ständig mit sich. Die Verteilung über das World Wide Web bietet einen zeit- und ortsunabhängigen flexiblen Zugang zu den Daten. Ein weiterer Vorteil der Darstellung über den Webbrowser ist, dass ein einfacher Zugang von den meisten Endgeräten ohne Installation weiterer Software ermöglicht wird.

---

<sup>1</sup> Fachhochschule Nordwestschweiz, Institut Vermessung und Geoinformation, Gründenstrasse 40, CH-4132 Muttenz, E-Mail: markus.jung@thurweb.ch

Aus diesen Gründen sind vermehrt neue, alternative Methoden zur 3D-Visualisierung im Webbrowser gefragt.

## 2 Technologien

In dieser Masterthesis wurde für die Darstellung von 3D Stadtmodellen über den Webbrowser ein bildbasierter Ansatz verwendet. Die Bilder werden in der Cloud prozessiert und zwischengespeichert. Für die Darstellung im Browser wurden die gespeicherten Bilder geladen und dargestellt. Das cloudbasierte Rendering erfolgt dabei mit dem Raytracing Verfahren. Verschiedene Interaktionen werden durch Image Based Provisioning ermöglicht.

### 2.1 Image Based Provisioning

SAITO & TAKAHASHI (1990) präsentierten das Prinzip von geometrischen Puffern (G-Buffer). Diese bestehen aus einem Set von Ansichten, welche jede davon eine geometrische Eigenschaft beinhaltet. Es wird teilweise auch von Multi Layer Images gesprochen (Abb. 1). Diese aus mehreren Layern bestehenden Bilder werden für das Image Based Provisioning verwendet. Die 3D Modelle werden dazu auf einem Server zu Multi Layer Images gerendert. Diese Bilder werden auf den Client gesendet und können dadurch auch auf Geräten mit geringerer Hardwareleistung visualisiert werden. Obwohl es sich um Bilder handelt, ist dank den verschiedenen Layern trotzdem eine Interaktion möglich. Die Tiefenkarte ermöglicht ein Messen im dreidimensionalen Raum. Anhand des Layer zur Objektidentifikation lassen sich Objekte anwählen und weitere thematische Daten dazu abrufen. Das Open Geospatial Consortium (OGC) definiert zurzeit dazu den 3D Portrayal Service, ein Standard zur Verbreitung von 3D-Geodaten über das Internet (OPEN GEOSPATIAL CONSORTIUM 2015).

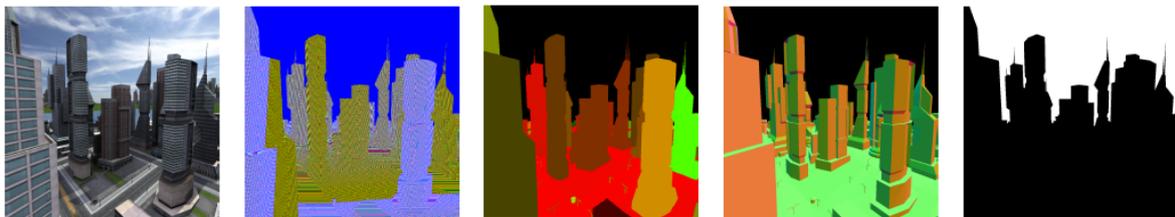


Abb. 1: Multi Layer Images: (v.l.n.r) RGB, Tiefenkarte, Objektidentifikation, Normalen, Maske (HAGEDORN 2009)

### 2.2 Raytracing

Beim Rendering wird aus einer dreidimensionalen Szene mit definiertem Blickpunkt ein Bild gerendert. Raytracing ist ein solches Renderingverfahren das von APPEL (1968) vorgestellt und von WHITTED (1980) verbessert wurde. Dabei wird für einen definierten Kamerastandpunkt für jeden Pixel mittels einer Strahlverfolgung die Farbe berechnet. Entlang des Strahles wird das erste sichtbare Objekt ermittelt und die Farbe berechnet. Für die Farbberechnung werden einerseits die Textur und Oberflächenbeschaffenheit des Objektes verwendet. Zudem werden weitere Strahlen vom Objekt ausgesendet um Beleuchtung, Schattierung, Reflexionen und Transparenz zu simulieren. Dadurch können sehr realistische Bilder aus Modellen generiert werden, weshalb Raytracing oft in Filmen (z.B. Ice Age, Robots, Ratatouille) verwendet wird (Abb. 2). Ein großer

Vorteil ist auch die Möglichkeit die Berechnung zu parallelisieren, wodurch sie die Berechnungsdauer verkürzen lässt.



Abb. 2: Bild aus dem Film Ratatouille (2007), berechnet mittels Raytracing

### 3 Rendering

Das Rendering beim erstellten Prototypen erfolgte nach dem RenderMan Standard mit der Software PRMan. Beides wurde von den Pixar Animation Studios entwickelt (PIXAR 2016). Weiter wurden einige Untersuchungen mit Aqsis vorgenommen, welches eine Open Source Alternative zu PRMan ist.

#### 3.1 Datenaufbereitung

Als Ausgangsdaten für den Prototyp wurden mehrere 3D Modelle aus CityEngine verwendet. Diese wurden in das offene Dateiformat Wavefront OBJ exportiert, ein verbreitetes Format für die Speicherung und den Austausch von 3D Daten. Anschliessend wurden diese Daten in RenderMan spezifische Formate konvertiert, welche Geometrien und Texturen des Modells beinhalten. Zusätzlich wurden noch verschiedene Shader definiert. In den Shadern werden die beim Rendering benötigten Berechnungen der Farbwerte, Lichtverhältnisse, etc. programmiert.

#### 3.2 Transformation

Für das Rendering der Stadtmodelle wurde eine Orthogonalprojektion verwendet. Dazu wurden vordefinierte Schrägansichten auf das Modell gerendert (Abb. 3). Für jedes Bild wurden die Kameraausrichtung und der Bildausschnitt berechnet. Die Berechnung beinhalteten das Parsing der Koordinaten und Ermittlung der Bounding Box. Daraus konnten dann Transformationsmatrix für die Kamerapositionierung und der Gesamtausschnitt des Modells berechnet werden. Um das Modell auch auf mobilen Geräten effizient darstellen und darin navigieren zu können, wurden diese mittels Kachelung und Bildpyramiden in mehrere tausend Einzelbilder unterteilt. Für jedes Einzelbild wurde eine RIB Datei gemäss RenderMan Standard generiert und anschliessen mit PRMan gerendert. Nebst den Farbwerten wurden für jedes Bild weitere Layer generiert (Tiefenkarte, Normalen, Objektidentifikation). Anhand dieser zusätzlichen Layer lassen sich auch in den Bildern Koordinatenmessungen durchführen, Beleuchtungen simulieren oder Attribute zu Objekten anzeigen.

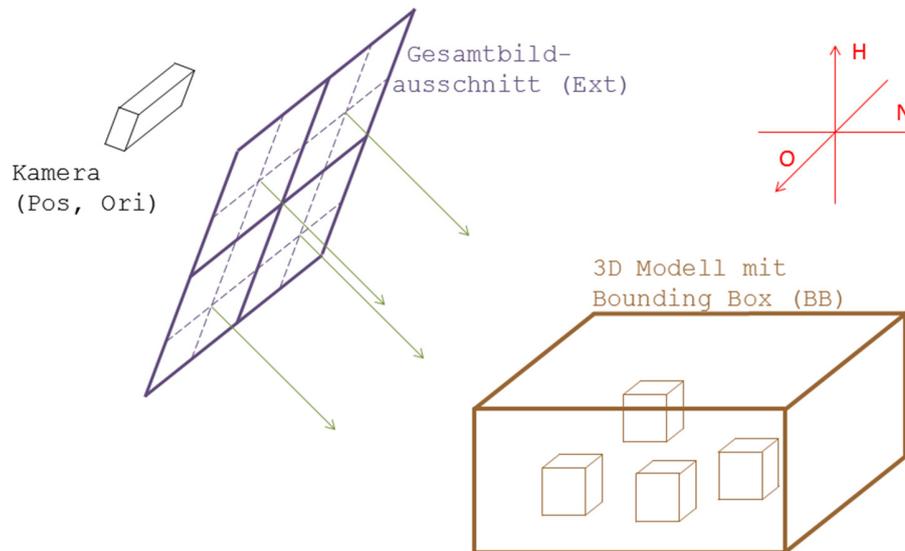


Abb. 3: Positionierung der Kamera und des Bildausschnitts zum 3D Modell

## 4 Webanwendung

Anhand des im letzten Kapitel vorgestellten serverseitigen Renderings lassen sich sehr komplexe Stadtmodelle über herkömmliche Webbrowser auf jedem beliebigen Gerät darstellen. Dies wurde mit einer einfachen Webanwendung aufgezeigt. Dazu wurde ein node.js Webserver aufgesetzt. Für die Darstellung des Modells wurde die Javascript Bibliothek Leaflet verwendet. Darin kann sich ein Nutzer verschiedene Ansichten auf ein Modell anzeigen lassen und darin navigieren (Zoom, Pan). Die angezeigten Kacheln werden dazu laufend vom Server geladen. Sofern vorhanden werden die Kacheln aus dem Zwischenspeicher (Cache) geladen oder ansonsten gerendert. Mittels zwei weiteren Service lassen sich im Webbrowser räumliche und attributive Eigenschaften zu den Objekten abrufen. Diese werden serverseitig aus den verschiedenen Layern ermittelt. Eine vereinfachte Version der Webanwendung (Anzeige von zwischengespeicherten Kacheln ohne Messfunktionalität und Attributanzeige) kann über [mth2014.3dmaps.ch](http://mth2014.3dmaps.ch) aufgerufen werden.

## 5 Cloud-Computing

Beim entwickelten Prototypen werden die Kacheln vorgerendert und für die Darstellung aus dem Cache geladen. Für das Rendering in Echtzeit werden leistungsfähige Server benötigt. Abb. 4 zeigt eine Cloudarchitektur, die mehrere Vorteile vereint. Das Rendering eines einzelnen Bildes kann auf mehreren Servern parallel prozessiert werden. Ein Webserver verarbeitet die verschiedenen Nutzeranfragen. Sofern vorhanden können die benötigten Daten aus dem Cache geladen werden. Ansonsten wird das Bild auf mehreren Servern gerendert und für künftige Anfragen im Cache abgelegt. Ein Monitoring überwacht die Antwortzeit und die Auslastung der Server fürs Rendering. Die einzelnen Server lassen sich in der Cloud nach Bedarf hinzu- oder abschalten, wodurch die Anwendung sehr kosteneffizient und skalierbar betrieben werden kann.

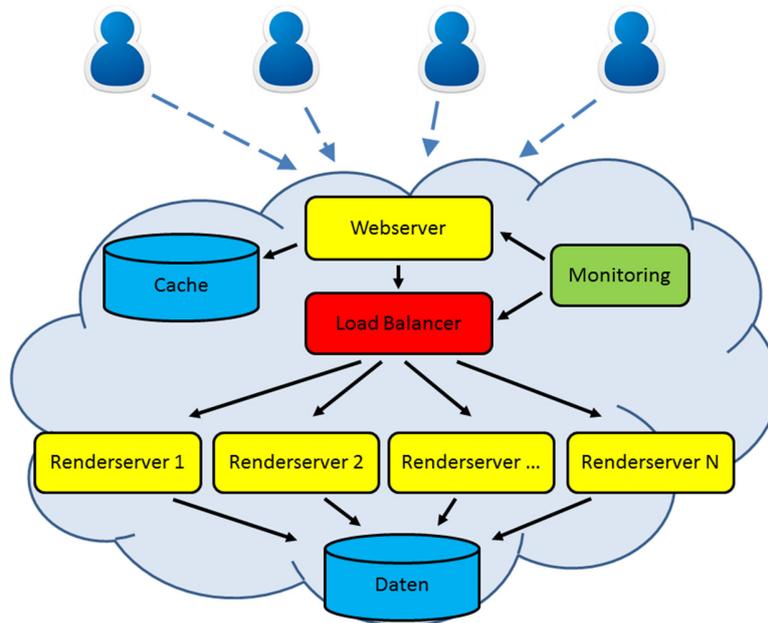


Abb. 4: Parallelisierung des Rendering in der Cloud

## 6 Resultate

Durch das serverseitige Rendern kann die Komplexität eines 3D Modells von seiner Darstellung entkoppelt werden. Somit lassen sich auch sehr grosse Stadtmodelle auf beliebigen Endgeräten ohne grosse Hardwareleistung darstellen. Nachfolgend zwei gerenderte Kacheln mit der Software PRMan (Abb. 5) und Aqsis (Abb. 6). Mit PRMan konnten sehr realitätsgetreue Bilder generiert werden. Grund für den auffälligen Qualitätsunterschied ist die bessere Berechnung der Lichtverhältnisse mittels Ambient Occlusion in PRMan.



Abb. 5: PRMan mit Ambient Occlusion



Abb. 6: Aqsis ohne Ambient Occlusion

Das Rendering von mehreren Layern (Farbwerte, Tiefenkarte, Objektzugehörigkeit) ermöglicht auch in den Bildern verschiedene Interaktionen und Abfragen. Abb. 7 und Abb. 8 zeigen Koordinatenmessungen sowie Attributabfragen in der Webanwendung.

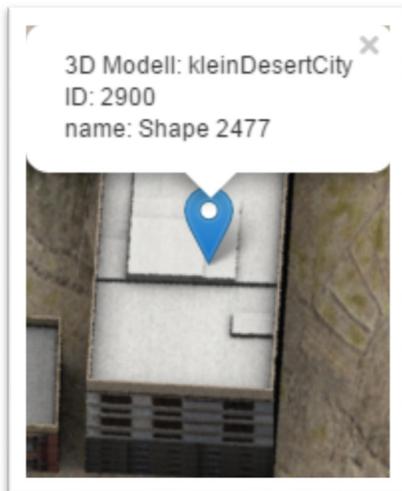


Abb. 7: Abfrage von Attributen

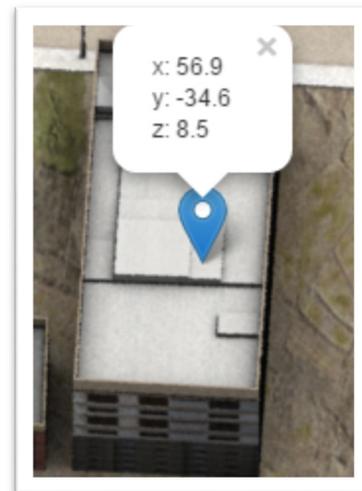


Abb. 8: Koordinatenmessung



Abb. 9: 3D-Stadtmodell Rotterdam mit Orthophoto: Highlight mittels Objektidentifikationslayer (CHRISTEN 2016)

## 7 Fazit und Ausblick

Mit dem entwickelten Verfahren und dem Prototyp konnte gezeigt werden, dass sich auch sehr komplexe 3D-Stadtmodelle effizient auf verschiedenen Endgeräten darstellen lassen. Image Based Provisioning und Raytracing ermöglichen die Komplexität eines Modells von der Übertragung und der Darstellung zu entkoppeln. In der prototypischen Webanwendung werden zwischengespeicherte vordefinierte Ansichten angezeigt. Optimierungen für das Rendering in

Realtime wurden mit der Cloudarchitektur vorgestellt. Diese Rasterdaten liessen sich im Client mit weiteren Raster oder Vektordaten erweitern. Dazu können verschiedenen Szenarien simuliert und dargestellt werden. Somit lassen sich diverse Anwendungen dreidimensional darstellen und verbreiten, beispielweise Verkehrssimulationen, Überschwemmungen, Stadtentwicklungen, Überbauungen und nur einige zu nennen.

Die Arbeit wurde inzwischen weiterentwickelt, um 3D-Stadtmodelle aus City-GML und Orthophotos zu unterstützen (CHRISTEN 2016), was in der Abb. 9 zu sehen ist. Geplant sind unter anderem auch die Visualisierung von Punktwolken, sowie Animationen bei denen Echtzeit-3D mit der Schrägansicht kombiniert werden kann.

## 8 Literaturverzeichnis

- APPEL, A., 1968: Some techniques for shading machine renderings of solids. AFIPS Conference Proc. **32**, 37-45
- CHRISTEN, M., NEBIKER, S. & LOESCH, B., 2012: Web-Based Large-Scale 3D-Geovisualisation Using WebGL. International Journal of 3-D Information Modeling **1** (3), 16-25.
- CHRISTEN, M., HÜRBI, K. & NEBIKER, S., 2014: OpenWebGlobe: 3D-Visualisierung und Caching von globalen Stadtmodellen aus OpenStreetMap mittels Cloud-basiertem Framework. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band 23, E. Seyfert, E. Gülch, C. Heipke, J. Schiewe, M. Sester, (Hrsg.), 34. Wissenschaftlich-Technische Jahrestagung der DGPF, Beitrag 250.
- CHRISTEN, M., 2016: Visualisierung von 3D-Geodaten im (mobilen) Webbrowser. Geoinformationssysteme 2016, Kolbe, T.H., Bill, R. & Donaubaue, A. (Hrsg.), Wichmann, ISBN 978-3-87907-610-9.
- COZZI, P. & BAGNELL, D., 2013: A webgl globe rendering pipeline. GPU Pro 4: Advanced Rendering Techniques **4**, 39-48.
- HAGEDORN, B., 2009: Service-Based, Interactive Portrayal of 3D Geovirtual Environments. Proceedings of the 4<sup>th</sup> Ph.D. Retreat of the HPI Research School on Service-oriented Systems Engineering, Meinel, C., Plattner, H., Döllner, J., Weske, M., Polze, A., Hirschfeld, R., Naumann, F. & Giese, H. (Hrsg.), Potsdam, Universitätsverlag, 13:1-13:9.
- KLIMKE, J., HAGEDORN, B. & DÖLLNER, J., 2014: Scalable Multi-Platform Distribution of Spatial 3D Contents. International Journal of 3-D Information Modeling **3**, 3, 35-49.
- OPEN GEOSPATIAL CONSORTIUM (OGC), 2015: OGC seeks public comment on candidate 3D Portrayal Service Standard, OGC Press Release, <http://www.opengeospatial.org/pressroom/pressreleases/2165>, letzter Zugriff 25.02.2016.
- PIXAR, 2016: Renderman. <https://renderman.pixar.com/view/renderman>, letzter Zugriff 02.03.2016.
- SAITO, T. & TAKAHASHI, T., 1990: Comprehensible Rendering of 3-D Shapes. Proceedings of the 17<sup>th</sup> Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '90. New York, NY, USA: ACM, 197-206
- WHITTED, T., 1980: An Improved Illumination Model for Shaded Display. Commun. ACM, **23** (6), 343-349