

3D-Geoapplikationen im Browser

Überblick & Erfahrungen

Michael Holthausen & Daniel Koch, terrestris GmbH & Co. KG

FOSSGIS 2019, Dresden, 14.03.2019

Gliederung

- Über...
- Was sind 3D-Geoapplikationen?
- Technische Lösungen
- Datenformate & Schnittstellen
- Vergleich von Bibliotheken

Michael Holthausen



- M. Sc. Geographie
- Anwendungsentwickler
@terrestris

✉ holthausen@terrestris.de
⌚ @mholthausen

Daniel Koch



- M. Sc. Geographie
- Lead developer @terrestris
- Kernentwickler react-geo
- Kernentwickler SHOGun

✉ koch@terrestris.de

⌚ [@dnlkoch](https://twitter.com/dnlkoch)

terrestris



✉ info@terrestris.de
⌚ [@terrestris](https://twitter.com/terrestris)
🐦 [@terrestrisde](https://twitter.com/terrestrisde)

- terrestris.de
- OpenSource GIS aus Bonn
- Entwicklung, Projekte & Support/Schulung
- Beratung, Planung, Implementierung & Wartung

Was sind 3D- Geoapplikationen?

“

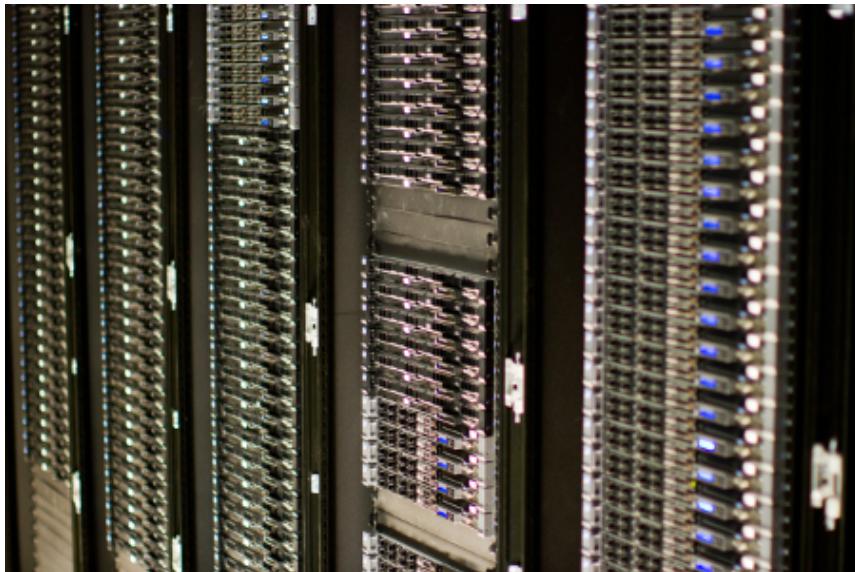
*We live in a 3D world. People move,
think, and experience in three dimensions*

Eine Einführung

- Viele Medieninhalte liegen in 3D vor, werden aber auf flachen Bildschirmen präsentiert
- Animationsfilme werden aus computergenerierten 3D-Bildern erstellt
- Online-Kartendienste lassen uns die Welt in 3D erkunden

- Ursprung bereits in den 1960er Jahren
- Umspannt viele Bereiche des Alltags

Voraussetzungen früher:



Und heute?

- 3D-fähige Hardware in jedem Computer und mobilen Endgerät
- universell verfügbar & kostenlos:



Google Chrome Anwendung "100,000 Stars"

- Browserbasierte 3D-Simulation unserer stellaren Nachbarn in der Milchstraße
- Wissenschaftliche Genauigkeit der Daten kann nicht garantiert werden
- HTML5 Elemente wie WebGL, CSS3D und Web Audio



L

Take a tour.



?

"100,000 Stars", Google Data Arts Team

Anwendung im Raum

- Darstellung von texturierten Gebäuden oder Objekten
- Darstellung von Geländemodellen
- Visualisierung von geologischen Untergrundschichten einer bestimmten Region
- Einbindung von Orthofotos
- u. v. a. m.

Technische Lösungen

Ziele

- Level of Detail (LOD)
- Übertragung als Stream

Clientseitig



-
- Hardwarebeschleunigtes 3D-Rendering mit JavaScript
 - Basiert auf der bewährten Grafik-API OpenGL
 - Alle Desktopbrowser
 - Die meisten mobilen Browser



-
- Transformationen, Übergänge und benutzerdefinierte Filter
 - Hardwarebeschleunigte 3D-Rendering- und Animationsfunktionen
 - Alle Desktop- & mobilen Browser



-
- 2D-Grafiken auf der Oberfläche eines DOM-Elements (Canvas)
 - zusätzliche JavaScript Bibliotheken rendern 3D-Effekte
 - Alternative zu WebGL oder CSS3 3D

Welche Technologie?

- Was soll mit der Anwendung abgebildet werden?
- Welche Plattformen sollen unterstützt werden?
- Welche Performance soll erreicht werden?
- Wie hoch soll der Grad der Interaktivität sein (messen, bewegen, navigieren)?
- u. v. a. m.

Serverkomponenten



GeoServer

- W3DS (Web 3D Service)
- WVS (Web View Service)

Beide Dienste sind *deprecated* und werden nicht weiter entwickelt



3D City DB

- Kostenlose Geodatenbank zum Speichern, Darstellen und Verwalten von virtuellen 3D-Stadtmodellen
- Implementiert den CityGML-Standard
- Produktive und kommerzielle Nutzung seit mehr als 14 Jahren
- Direkter Export in den Formaten KML, COLLADA und glTF

Datenformate & Schnittstellen

OGC Standards

3DPS (3D Portrayal Service)

- Seit 2017 OGC Standard
- Große Menge raumbezogener 3D-Daten im Web
- Keine Definition eines bestimmten Formates
- Stellt ein Framework bereit zur Bestimmung der Interoperabilität



CityGML

- Seit 2008 OGC Standard
- Offenes Datenmodell und XML-basiertes Format
- Speicherung und Austausch von virtuellen 3D-Stadtmodellen
- Gemeinsame Definition der grundlegenden Bestandteile eines 3D-Stadtmodells
- CityJSON: kompakte Beschreibung von CityGML-Daten

I3S (Indexed 3D Scene Layers)

- Seit 2017 OGC Standard
- Spezifikation für Speicherung und Übertragung von heterogenen 3D (Geo-)Daten
- Unterstützung von diversen 3D Datentypen (3D Objekte, Punktwolken, etc.)
- Scene Layer (Container für Daten)
- Scene Service (REST API) vs. Scene Layer Package (SLPK)

Web3D Consortium

X3D (Extensible 3D Graphics)

- Ersetzt das Vorgängerformat VRML (Virtual Reality Modelling Language)
- Standard für 3D-Grafiken im Web
- Kompatibel mit anderen Standards (DOM, XML)
- Szenen werden direkt in das HTML-Markup geschrieben
- Bevorstehendes Upgrade auf Version 4

Beispiel einer 3D-Box mit X3D

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.3//EN" "http://www.web3d.org/specifications/x3d-3.3.dtd">
<X3D profile='Interchange' version='3.3' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
xsd:noNamespaceSchemaLocation =' http://www.web3d.org/specifications/x3d-3.3.xsd '>
  <head>
    <meta name='title' content='Box.x3d'/>
    <meta name='description' content='Simple Box shape used as test scene for GeoInline/Inline loading and
unloading.'/>
  </head>
  <Scene>
    <Transform translation=' -1.0 0.0 0.0 ' >
      <Shape>
        <Appearance>
          <Material diffuseColor='0.0 1.0 0.0' />
        </Appearance>
        <Box />
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

X3D Beispiel (Box)

Bisher ohne Standardisierung

CZML (Cesium Language)

- JSON basiertes Format
- Punkte, Flächen, Modelle und andere grafische Grundelemente
- Beschreibung einer raumzeitlich-dynamischen grafischen Szene
- Effizientes Streaming & einfach zu parsen
- Hauptsächlich zur Anzeige im Browser über CesiumJS

Beispiel zur Positionierung eines Punktes mit CZML

```
{  
  "id": "point",  
  "availability": "2012-08-04T16:00:00Z/2012-08-04T16:05:00Z",  
  "position": {  
    "epoch": "2012-08-04T16:00:00Z",  
    "cartographicDegrees": [  
      0, -70, 20, 150000,  
      100, -80, 44, 150000,  
      200, -90, 18, 150000,  
      300, -98, 52, 150000  
    ]  
  },  
  "point": {  
    "color": {  
      "rgba": [255, 255, 255, 128]  
    }  
  }  
}
```



- Seit 2016 OGC Community Standard, auf dem Weg zum OGC Standard
- Konzipiert für das Streaming und Rendern von 3D-Inhalten auf einen Globus
- Photogrammetrie, 3D-Gebäude, BIM/CAD (Bauwerksdatenmodellierung) und Punktwolken
- Definiert eigene Binärfomate für das Streaming
- Styling-Spezifikationen

3D Tiles

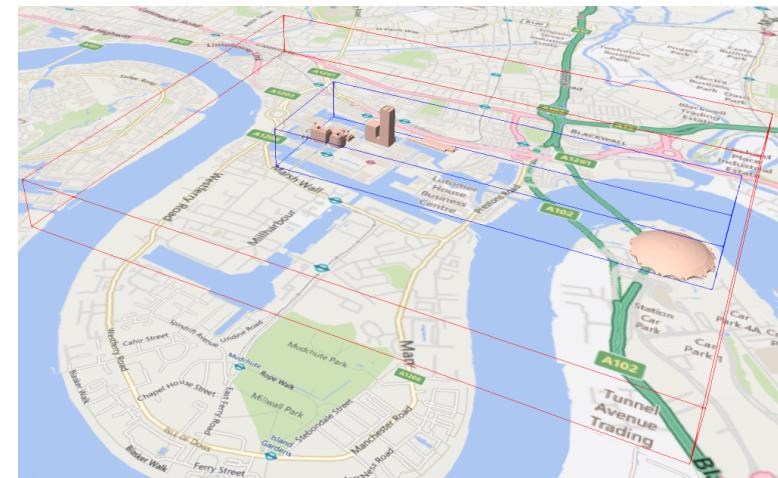
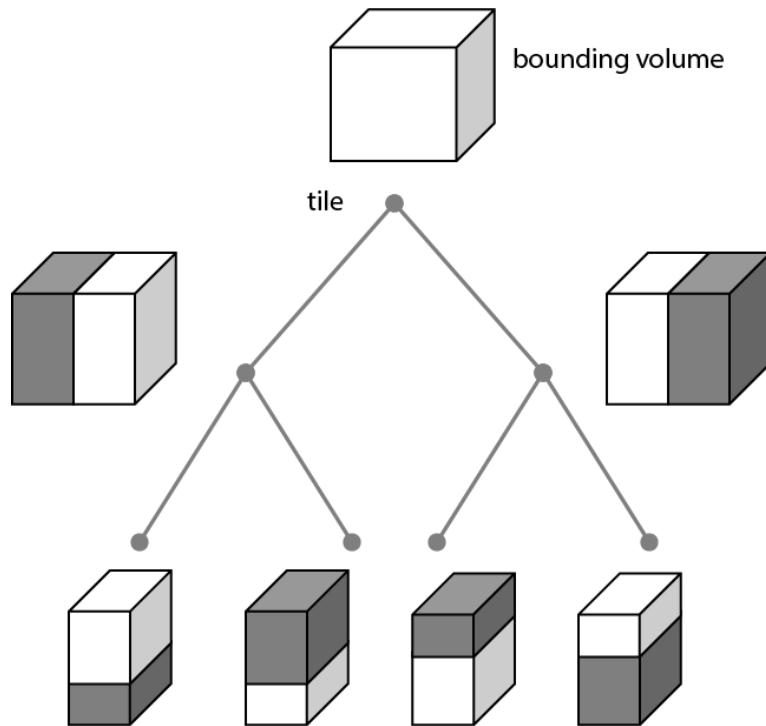
- b3dm (Batched 3D Model)
 - heterogene 3D-Modelle
 - z.B. für verschiedene Gebäude in einer Stadt
- i3dm (Instanced 3D Model)
 - effizientes Streaming und Rendern einer großen Anzahl von Modellen mit geringfügigen Abweichungen

3D Tiles

- pnts (Point Cloud)
 - effizientes Streaming von Punktwolken zur 3D-Visualisierung
- cmpt (Composite)
 - flexibles Streaming kombinierter, heterogener Datensätze
 - z.B. aus b3dm- und i3dm-Modellen

3D Tiles

Hierarchical Level of Detail (HLOD)
zur optimalen Wiedergabe räumlicher Daten





- *JPEG of 3D*
- Konzipiert für das Streaming und Rendern von 3D-Inhalten auf einen Globus
- Besteht aus:
 - Einer JSON Datei (.gltf), Szenenbeschreibung & Verweis auf externe Dateien
 - Binärdateien (.bin), Geometrie- und Animationsdaten
 - Bilddateien (.jpg, .png, usw.), Texturen

Vergleich von Bibliotheken

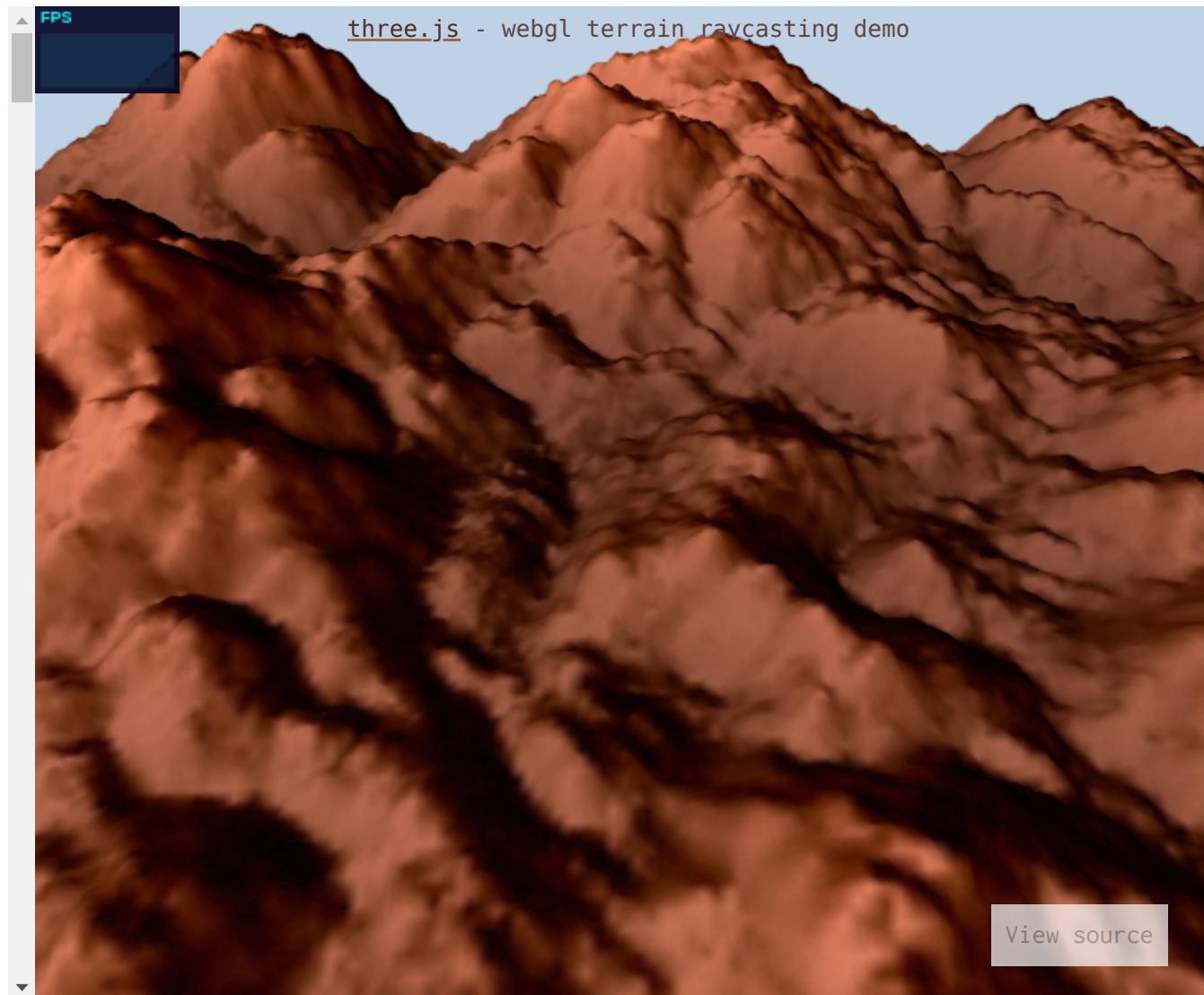
three.js

three.js / examples

Type to filter X

webgl

- animation / cloth
- animation / keyframes
- animation / skinning / blending
- animation / skinning / morph
- animation / multiple
- camera
- camera / array
- camera / cinematic
- camera / logarithmicdepthbuffer
- clipping
- clipping / advanced
- clipping / intersection
- decals
- depth / texture
- effects / anaglyph
- effects / ascii
- effects / parallaxbarrier
- effects / peppersghost
- effects / stereo
- framebuffer / texture
- geometries
- ...



three.js Beispiel

x3dom

Information

Description:

This 3D model is rendered by the **X3DOM**-BVHRefiner node which refines and loads hierarchical data dynamically corresponding to the view point.

3D-Model Statistics:

Model:	Puget Sound
DataSet:	4096 x 4096 Pixel
Points:	16.777.261 (~16.5 Mio.)

Rendering Statistics:

Rendering speed:	0 fps
Currently drawn points:	0.0

Parameters

CesiumJS

History of Wildfire Severity
(Click on a cylinder to view details)

Wildfires from 1984 to 2016

Display Options

Show non-forest (e.g., shrubland) fires
 Show cumulative fire events

Timeline playback

[Running @ 3.00 seconds per year]

Use buttons for playback control
Spacebar toggles play/pause

Predominant Burn Severity

High Moderate Low

Color corresponds with largest burn severity assessment area

Area of Fire Boundary

Small Med Large

Cylinder base area is proportional to fire boundary acreage

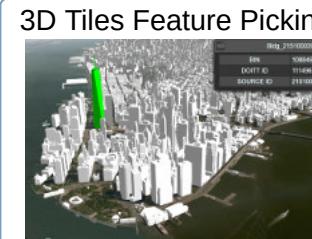
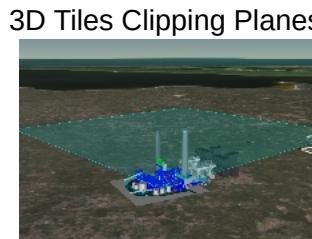
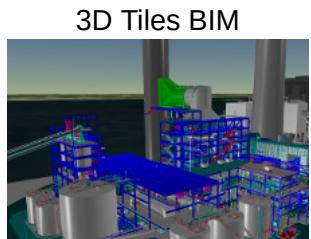
CesiumJS Beispiel (History of Wildfire Severity, oregonhowl.org)

CesiumJS 3D Tiles

[New](#)[Run \(F8\)](#)[Suggest \(Ctrl-Space\)](#)[Info](#)[Save As](#)[Share](#)[Import Gist](#)[Open in New Window](#)[View as Thumbnail](#)[Search Gallery](#)[JavaScript code](#)[HTML body & CSS](#)

```
1 // A demo of interactive 3D Tiles styling
2 // Styling language Documentation: https://cesium.com/tutorials/styling-language/
3 // Building data courtesy of NYC OpenData | https://data.cityofnewyork.us/
4 var viewer = new Cesium.Viewer('cesiumContainer',
5     {
6         terrainProvider: Cesium.createWorldTerrain()
7     });
8
9 viewer.scene.globe.depthTestAgainstTerrain = true;
10
11 // Set the initial camera view to look at Manhattan
12 var initialPosition = Cesium.Cartesian3.fromDegrees(-74.0, 40.7, 1000);
13 var initialOrientation = new Cesium.HeadingPitchRoll(0, -15, 0);
14 viewer.scene.camera.setView({
15     destination: initialPosition,
16     orientation: initialOrientation,
17     ...});
```

Cesium 1.56.1

[Gallery](#)[Console](#)[Showcases](#)[New in 1.56.1](#) [Tutorials](#) [3D Tiles](#) [Post Processing](#) [Beginner](#) [ion Assets](#) [Geometries](#) [DataSources](#) [CZML](#)

CesiumJS Beispiel (3D Tiles)

Zusammenfassung

	three.js	x3dom	CesiumJS
Beispiele/Tutorials	✓	✓	✓
Dokumentation & API	✓	✓	✓
Support	✓	○	✓
Community			
Updates	✓	○	✓
Raumbezug	✗	✗	✓

Zusammenfassung

- Erhöhte Aktivitäten im Kontext der Web-Visualisierung von 3D-Inhalten erkennbar
- Viele unterschiedliche (standardisierte) Datenformate
- Teils schwer oder nicht untereinander austauschbar
- Alle Bibliotheken arbeiten ohne zusätzliche Plugins (WebGL)

Vielen Dank

Fragen & Anmerkungen?

Impressum

Impressum

Autoren

Michael Holthausen

terrestris GmbH & Co. KG

Kölnstr. 99

53111 Bonn

holthausen@terrestris.de

Daniel Koch

koch@terrestris.de

Lizenz

Diese Folien sind unter CC BY-SA veröffentlicht.

Vortragsfolien, PDF-Version, git repository