

# Image Based Provisioning für grosse CityGML-basierte 3D-Stadtmodelle

Die zunehmende Verfügbarkeit von 3D-Stadtmodellen in offenen Standards führt zur Frage, wie diese effizient visualisiert und den Benutzenden bereitgestellt werden. Durch neue Erfassungstechnologien werden immer mehr und detailliertere Geometrien und Texturen erzeugt. Mit dem Image Based Provisioning können diese hochkomplexen 3D-Daten effizient auf mobilen Endgeräten zur Verfügung gestellt werden. In einer neu entwickelten Python-Applikation werden die mit Blender-Skripten vorgerenderten Kacheln des G-Buffers für hochdynamische Anwendungen eingesetzt.

## CityGML-3D-Stadtmodell Berlin

Als Grundlage dient das frei verfügbare Stadtmodell von Berlin. Die Kombination aus verschiedenen Beleuchtungstechniken verleiht dem Modell eine natürliche Erscheinung. Die Kacheln werden in der Parallelprojektion gerendert und einzeln gespeichert.

## G-Buffer

Basierend auf dem G-Buffer (Saito & Takahashi 1990) wird mit speziell entwickelten Python-Skripten in Blender die Szene mit unterschiedlichen Bildebenen gerendert. Für die aktuelle Anwendung werden eine RGB-Map, eine Tiefenkarte, Normal-Map und eine Objekt-ID-Map mit der programmierbaren Open-Source 3D-Software Blender erstellt (siehe Abbildung rechts).

## 3D-Koordinaten im 2D-Bildraum

Durch die Modelview- und Projektionsmatrizen können 2D-Bildpunkte mit Einbezug einer Tiefenkarte effizient mittels Matrizenmultiplikation in 3D-Weltkoordinaten gerechnet werden. Mit dieser Information kann das Modell in Form einer Punktwolke rekonstruiert werden.

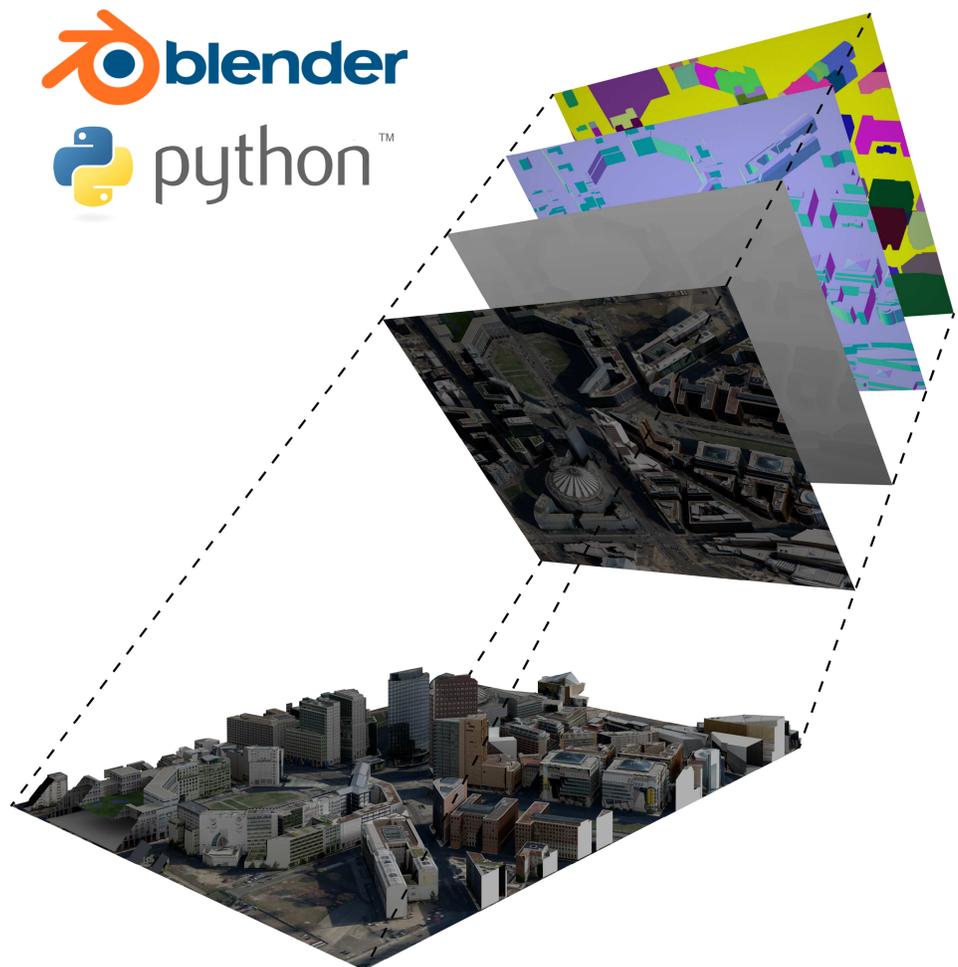
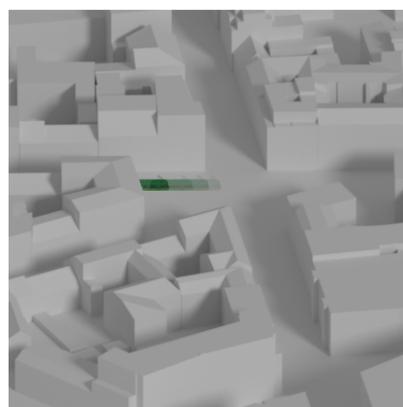


Abbildung: Rendering des G-Buffers im texturierten Stadtmodell (RGB-Map, Tiefenkarte, Normal-Map und Objekt-ID-Map)



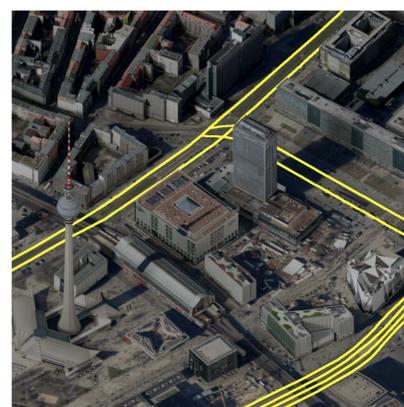
## Real-Time Simulation

Da von jedem Pixel die 3D-Koordinate berechnet werden kann, lassen sich viele Echtzeit-Anwendungen wie z.B. eine Hochwassersimulation realisieren. Dank den vorgerenderten Kacheln ist dies auch mit hochkomplexen Daten möglich.



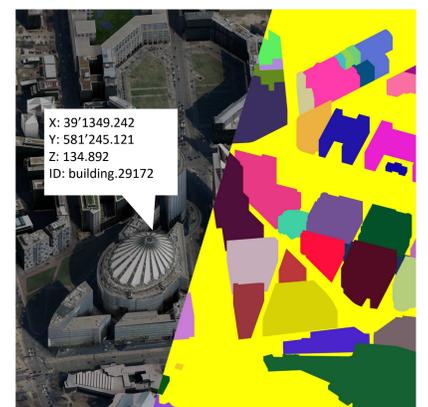
## Sichtbarkeitsanalyse

Durch einen Distanzvergleich in der Tiefenkarte kann dynamisch überprüft werden, ob ein Objekt sichtbar ist oder von einem Gebäude o.Ä. verdeckt wird. Die Objekte können so Interaktiv bewegt werden.



## Überlagerung Vektordaten

Stützpunkte einer Linie können in das Bildkoordinatensystem umgerechnet werden. Durch einen erweiterten Bresenham-Algorithmus wird pixelweise die Linie gezeichnet und auf deren Sichtbarkeit überprüft.



## Präzise Objektinformationen

Im ID-Layer hat jedes Objekt im ganzen Stadtmodell einen eindeutigen Farbwert. Über diesen Identifikator können pixelpräzise Informationen zu den CityGML-Objekten abgefragt werden.

Referenzen: Saito, Takafumi und Takahashi, Tokiichiro (1990): Comprehensible Rendering of 3-D Shapes. In: Computer Graphics. 24 (4), <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.4139&rep=rep1&type=pdf>.

**Autor:** Remo Ackermann  
**Examinator:** Prof. Martin Christen  
**Experte:** Robert Wüest