

Beleuchtungsmodelle für 3D-Stadtmodelle mit GPU-optimiertem Rendering in der Cloud

Raytracing (dt. Strahlverfolgung) ist eine Methode zur Verdeckungsrechnung in der Computergrafik. In dieser Master Thesis wurde ein GPU basierter Raytracer zum Rendern von 3D-Stadtmodellen entwickelt. Dabei basiert die Intersektionsberechnung auf der Radeon Rays SDK von AMD. Es sind zwei Beleuchtungsmodelle implementiert. Zum einen das Blinn-Phong Beleuchtungsmodell und zum anderen Ambient Occlusion. Die Funktionen sind in C++ programmiert und für eine einfachere Bedienung in Python integriert. Die Einstellungen für das Rendern eines Modells werden in einem Textfile übergeben.

Backwards Raytracing

Beim Backwards Raytracing werden von der Kamera aus durch jedes Pixel Strahlen in die Szene geschickt und mit den Objekten verschnitten. Das Verfahren eignet sich, um realistische Beleuchtung zu simulieren, ist jedoch äusserst rechenintensiv. Die Millionen von Schnittberechnungen werden mit der Radeon Rays SDK durchgeführt. Dies ist eine Open Source C++ Bibliothek von AMD, mit welcher es möglich ist parallelisierte Intersektionsberechnungen auf der GPU durchzuführen. In Abb. 1 ist der Ambient Occlusion Algorithmus dargestellt. Die Schattierung des Pixels wird in Abhängigkeit der Umgebungsverdeckung des Schnittpunktes ermittelt, welche durch zufällig ausgesendete Strahlen bestimmt wird.

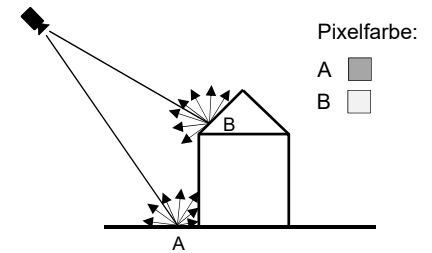


Abb. 1: Ambient Occlusion nach Christensen (2002) und Landis (2002)

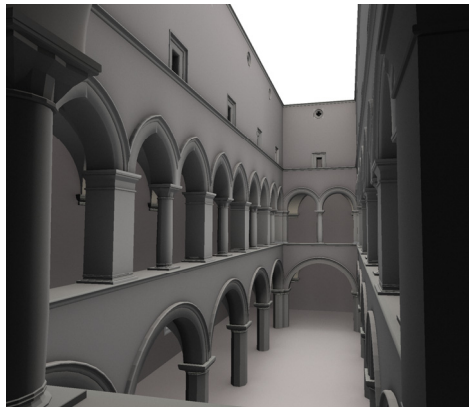


Abb. 2: Modell Sponza, Blinn-Phong überlagert mit Ambient Occlusion



Python Binding



Beleuchtungsmodelle implementiert in C++



Radeon Rays SDK Raytracing Kernel

Abb. 3: Übersicht Systemarchitektur

GPU Raytracer

Der entwickelte Raytracer lässt sich mit Python ausführen (siehe Abb. 3). In einem JSON-File werden die Einstellungen zum Rendern erfasst. Die implementierten Funktionen, wie zum Beispiel Primärstrahlen, Schattenstrahlen, Ambient Occlusion, usw. sind in C++ programmiert. Die Funktionen wurden als DLL kompiliert und in Python mit dem Modul ctypes eingebunden.

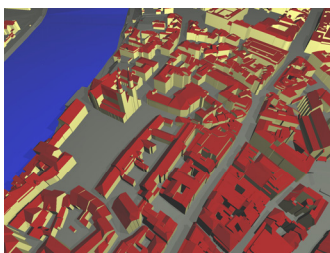


Abb. 4: Basel mit Blinn-Phong Beleuchtungsmodell

Resultate

Es wurden zwei Beleuchtungsmodelle umgesetzt. Zum einen das Blinn-Phong Modell (siehe Abb. 4) und zum anderen Ambient Occlusion (siehe Abb. 5). Die beiden Resultate können auch überlagert werden, was in Abb. 2 dargestellt ist. Die Berechnungszeit auf der GPU ist etwa halb so lang wie auf der CPU.



Abb. 5: Berlin mit Ambient Occlusion

Fazit

Mit dem entwickelten Raytracer ist es möglich, unter Verwendung von Python 3D-Stadtmodelle mit verschiedenen Beleuchtungsmodellen und Einstellungen auf der GPU zu rendern. Neben den oben erwähnten Beleuchtungsmodellen ist es auch möglich zwischen einer perspektivischen und orthogonalen Kamera zu wählen. Die Renderingzeit ist abhängig von den gewählten Einstellungen und beträgt einige Sekunden bis wenige Minuten. Dabei können mithilfe der Radeon Rays SDK auf einer GeForce GTX 1080 Grafikkarte 90 Millionen Schnittberechnungen pro Sekunde durchgeführt werden.

Referenzen

Christensen, P. H. (2002): Note #35: Ambient Occlusion, Image-Based Illumination, and Global Illumination. (PhotoRealistic RenderMan Application Notes).
Landis, H. (2002): „RenderMan in Production“. ACM SIGGRAPH 2002, Course 16.

Autor: Markus Fehr
Examinator: Martin Christen
Experte: Grégory Jaegy

© FHNW Institut Vermessung und Geoinformation
Master Research Unit Geoinformationstechnologie

