

„Ambient Occlusion – ein Schritt zur realistischen Beleuchtung von 3D-Stadtmodellen“

Autoren

Jürgen Döllner, Henrik Buchholz, Haik Lorenz
Hasso-Plattner-Institut an der Universität Potsdam

Zusammenfassung

Der Beitrag gibt einen Überblick über innovative, echtzeitfähige Ansätze zur realistischen Beleuchtung von 3D-Stadtmodellen, mit denen die Qualität graphischer Darstellungen eine neue Stufe erreicht. Diese auf einem globalen Beleuchtungsmodell beruhenden Verfahren bilden derzeit einen Schwerpunkt in der Forschung im Bereich Visualisierung am Hasso-Plattner-Institut und ermöglichen für die Perzeption und Kognition optimierte Darstellungen komplexer 3D-Raummodelle. Im Folgenden werden die theoretischen Grundlagen und konkrete Ansätze zur approximativen Berechnung des Phänomens „globale Illumination“ erläutert. Die vorgestellten Verfahren wurden in ein Viewer- und Content-Management-System für CityGML-basierte 3D-Stadtmodelle erfolgreich integriert.

1 Einleitung

Die interaktive Visualisierung raumbezogener Informationen auf der Grundlage dreidimensionaler Raummodelle, wie zum Beispiel durch virtuelle 3D-Stadtmodelle und 3D-Landschaftsmodelle, etabliert sich derzeit als praxistaugliche und ausgereifte GIS-Basistechnologie. Die Einsatzfelder der 3D-Geovisualisierung umfassen dabei die Planung und Verwaltung (z. B. Stadtplanung, Funknetzplanung, Liegenschaftsverwaltung und Gebäudemanagement), die Simulation (z. B. für Hochwasser und Lärm), wie auch neue Anwendungen in der raumbezogenen Informationsvermittlung, etwa mittels 3D-Stadtplänen, 3D-Investorenberatungssystemen oder 3D-Werksinformationssystemen.

In den vergangenen Jahren war entscheidend, dass große 3D-Geodatenmengen überhaupt interaktiv visualisiert werden konnten – eine Fülle von spezialisierten Algorithmen wurde entwickelt, etwa für die Level-of-Detail-Darstellung von Geländemodellen, Geländetexturen und allgemeinen polygonalen Netzen. Auch die Frage der Hardware-technischen Voraussetzungen wurde dadurch beantwortet, dass heute leistungsstarke Computergraphik-Hardware zur Standardausstattung eines Arbeitsplatzrechners gehört – der Spiele-Industrie geschuldet.

Umso mehr stellt sich die Frage, wie 3D-Geovisualisierungssysteme qualitativ zu bewerten sind. Die computergraphische Qualität rückt insbesondere bei Präsentations-, Simulations- und Analyseanwendungen auf Grundlage virtueller 3D-Stadtmodelle in den Mittelpunkt. Eine der wesentlichen Schwachstellen heutiger IT-Lösungen stellt die mangelnde graphische Qualität dar, die sich für interaktive Systeme in den letzten 15 Jahre kaum positiv verändert hat, allenfalls durch verbesserte Eingangsdaten, wie z. B. höher aufgelösten Texturdaten für Gelände- und Fassadenmodelle.

Die Erklärung für die unveränderte computergraphische Qualität heutiger 3D-Geovisualisierungen liegt in der eingesetzten computergraphischen Softwaretechnologie. Im Kernstück der 3D-Rendering-Systeme, wie z. B. OpenGL, kommen *lokale*

Beleuchtungsmodelle zum Einsatz, die ihren Ursprung in den 70er und 80er Jahren haben; sie waren bestimmt durch die damalige Charakteristik der Computergraphik-Hardware. Seitdem wurden sie in der Kartographie und Geovisualisierung kaum grundlegend hinterfragt. Vielmehr galt es, diese technisch lange Zeit unveränderbaren Rahmenbedingungen bestmöglich zu nutzen. Beleuchtungsmodelle bilden heute keine unveränderbare Größe mehr – Texturierung, programmierbare Computergraphik-Hardware und korrespondierende Shading Languages (ROST 2004) erlauben die anwendungsspezifische Festlegung dieser qualitätsentscheidenden Funktionalität in interaktiven 3D-Geovisualisierungssystemen. Ein Ansatz einer solchen Neudefinition liegt in der Verwendung eines *globalen Beleuchtungsmodells*, das im Folgenden vorgestellt wird.

2 Potentiale globaler Beleuchtung in der 3D-Geovisualisierung

Beleuchtung und Schattierung auf Grundlage globaler Beleuchtungsmodelle spielen für die 3D-Geovisualisierung eine wesentliche Rolle, insbesondere für die perzeptive Qualität, die kognitive Verarbeitung und die Gestaltung raumbezogener Visualisierungen.

Das menschliche Auge nimmt elektromagnetische Strahlung wahr, nachdem diese mit den umgebenden Objekten interagiert hat. Eine präzise Simulation im Rechner ist aufgrund der Komplexität dieses Vorgangs nicht praktikabel. Z. B. strahlt eine 40W Glühlampe pro Sekunde ca. $1,1 \cdot 10^{20}$ Photonen ab, die im Modell berücksichtigt werden müssten. Statt einer korrekten physikalischen Simulation ist es für die Computergraphik allein entscheidend, einen glaubwürdigen Eindruck der Beleuchtung zu erreichen. Entsprechend dürfen grundlegende physikalische Gesetze, wie der Energieerhaltungssatz, verletzt werden, solange das Ergebnis, das erzeugte Bild, für die jeweilige Anwendung und Nutzergruppe glaubwürdig und angemessen erscheint.

Vereinfacht gesprochen berücksichtigt ein globales Beleuchtungsmodell die Lichtinteraktion zwischen allen Objekten einer Szene. Lokale Beleuchtungsmodelle hingegen berechnen ausschließlich die Lichtinteraktion zwischen Lichtquellen und Einzelobjekt. Für die Glaubwürdigkeit entscheidende Effekte wie indirekte Beleuchtung, Spiegelungen und Schattenwürfe werden dadurch nicht erfasst. Abb. 1 vergleicht beide Beleuchtungsmodelle anhand einer Beispielszene.

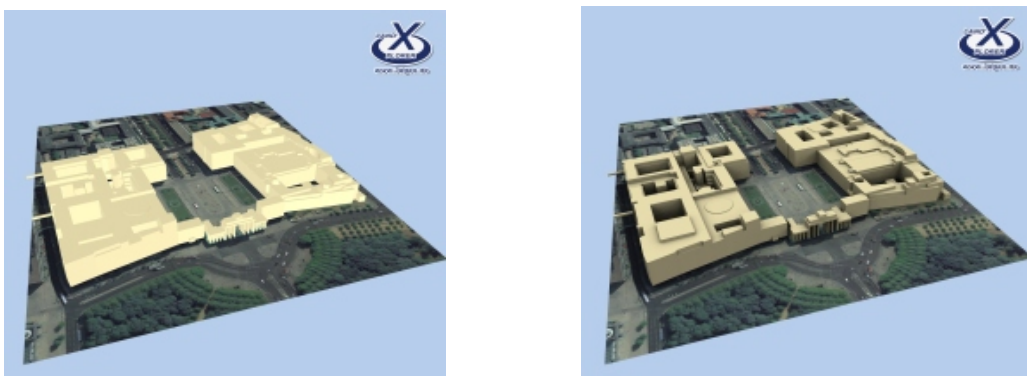


Abb. 1: 3D-Stadtmodell (CityGML-Modell Berlin, Pariser Platz) ohne und mit globaler Beleuchtung auf Basis von Ambient Occlusion.

2.1 Wahrnehmung von 3D-Objektdarstellungen

Der Mensch leitet bei der Wahrnehmung die Form eines dargestellten 3D-Objekts maßgeblich aus der im Bild kodierten Schattierung der Objektoberfläche ab („shape-from-shading“). Da

in zweidimensionalen Bildwiedergaben die stereoskopische Wahrnehmung ausscheidet, ist die Schattierung der wesentliche Parameter für die perzeptive Qualität. Globale Beleuchtungsmodelle kommen der physikalischen Lichtausbreitung nahe, wodurch räumliche Darstellungen für den Menschen perzeptiv einfacher verarbeitbar werden, da auf die allgemeine Seh- und Objekterkennungserfahrung zurückgegriffen werden kann.

2.2 Kognitive Verarbeitung von 3D-Raumsituationen

Die kognitive Verarbeitung von Darstellungen komplexer 3D-Raumsituationen wird durch globale Beleuchtungsmodelle ebenfalls verbessert und z. T. erst ermöglicht, da die räumliche Komposition der Einzelobjekte präzise vermittelt werden kann. Wesentliche Hinweise für Tiefe, Position und Nähe der Objekte werden im Bild, z. B. mittels ambierter Abdunklung oder Schattenwurf, direkt visuell kodiert.

2.3 Gestaltung geovirtueller Umgebungen

Die globale Beleuchtung bietet herausragende Möglichkeiten für die graphische Gestaltung. Selbst bei untexturierten 3D-Objekten erhalten Objektoberflächen eine Plastizität und Nuancierung. Weiter hilft globale Beleuchtung einen gestalterischen Bruch zwischen (z. B. photobasiert) texturierten und untexturierten Modellen visuell zu glätten. So wird es möglich, photorealistische 3D-Modelle aus Realaufnahmen und Modelldaten nahtlos zu kombinieren. Zudem können hochwertige abstrakte, kartographische Darstellungen von 3D-Stadtmodellen (z. B. Illustrationen) generiert werden.

3 Lokale Beleuchtungsmodelle

Das wichtigste lokale Beleuchtungsmodell ist das *Phong-Beleuchtungsmodell* (AKENINE-MÖLLER & HAINES 2002). Dieses empirische Modell, das nicht physikalisch begründet ist, berücksichtigt drei Arten von Beleuchtung:

- *Ambiente Beleuchtung* im Sinne einer allgemeinen Raumhelligkeit
- *Diffuse Beleuchtung* durch ungerichtete Reflexion des Lichts an der Oberfläche
- *Spekulare Beleuchtung* durch gerichtete Reflexion entlang einer Reflexionsrichtung.

Die Intensität I an einem Oberflächenpunkt ergibt sich wie folgt:

$$I = I_a k_a O_a + f_{att} I_p \left[k_d O_d (n \cdot l) + k_s O_s (r \cdot v)^s \right]$$

- Materialien werden durch die Kombination der verschiedenen Beleuchtungsarten dargestellt. Parameter sind die Materialfarbe aufgeteilt in ambienten, diffusen und spekularen Anteil (O_a, O_d, O_s), deren jeweiliger Einfluss (k_a, k_d, k_s), sowie die Schärfe der spekularen Reflexion s .
- Der erste Term $I_a k_a O_a$ berechnet die Umgebungsbeleuchtung oder *ambiente Beleuchtung*, die die gesamte Lichtstreuung durch mehrfache Reflexionen approximiert; der Term hängt weder von der Position und Orientierung der Fläche noch von den Lichtquellen ab.
- Der zweite Term $k_d O_d (n \cdot l)$ berechnet diffuse Reflexion, die nur von der Einfallrichtung des Lichts, nicht aber der Ausfallrichtung abhängt, berechnet durch das Skalarprodukt $n \cdot l$ zwischen Oberflächennormale und Lichtrichtung unter der Annahme, dass sich zwischen Oberflächenpunkt und Lichtquelle kein Hindernis befindet.
- Der letzte Term $k_s O_s (r \cdot v)^s$ berechnet spekulare Reflexion, z. B. die Lichtspiegelung bei metallischen oder polierten Oberflächen. Als Gewichtungsfunktion wird hier das Skalarprodukt $r \cdot v$ aus Reflektionsrichtung und Blickrichtung verwendet. Zusätzlich wird

es potenziert, um die Schärfe der Reflexion zu steuern. Da es sich um ein lokales Beleuchtungsmodell handelt, wird tatsächlich nur Licht der Lichtquelle gespiegelt, nicht aber das benachbarter Objekte.

- Zusätzlich bestimmt der Wert f_{att} die Intensitätsabschwächung (*Attenuation*) durch die Entfernung zwischen Lichtquelle und Oberfläche.

Weiter müssen in 3D-Szenen virtuelle Lichtquellen modelliert werden. Folgende idealisierten Lichtquellentypen finden bei lokalen Beleuchtungsmodellen Einsatz:

- *Ambiente Lichtquelle*, die konzeptionell ungerichtetes Licht konstant für alle Punkte der Szene aussenden.
- *Parallele Lichtquellen*, die konzeptionell Licht in Form paralleler Strahlen für die gesamte Szene aussenden (etwa analog zum Sonnenlicht).
- *Punktlichtquellen*, die konzeptionell ihr Licht von einem Punkt aus gleichmäßig in alle Richtungen strahlen.
- *Spot-Lichtquellen*, die als spezialisierte Punktlichtquellen wie ein Scheinwerfer nur einen Lichtkegel werfen.

Lichtparameter sind eine Intensität I_p sowie eine positions- und typabhängige Lichtrichtung l . Insgesamt bilden diese Modelle für Lichtquellen, Beleuchtung und Schattierung ein leistungsstarkes Trio, deren Hauptmangel allerdings in der konstanten Abschätzung der ambienten Beleuchtung liegt.

Ausgehend von punktuell ermittelten Beleuchtungsintensitäten erfolgt die *Schattierung* der Objektoberflächen, die diese Intensitätswerte auf Flächenstücke (i. a. Dreiecke) überträgt. Drei Schattierungsmodelle finden sich in heutigen 3D-Renderingsystemen:

- *Flat Shading* benötigt einen einzigen Intensitätswert, um ein Flächenstück einheitlich damit zu färben. Aufgrund der geringen Qualität der Ergebnisse wird dieses Verfahren kaum mehr eingesetzt.
- *Gouraud Shading* benötigt für alle Eckpunkte Intensitätswerte und schattiert ein Flächenstück, indem es diese Werte linear über die Fläche interpoliert. Die Beleuchtungsberechnung ist abhängig von der Modellraumkomplexität (Anzahl Dreiecke) eines Objekts.
- *Phong Shading* benötigt für alle sich aus der Rasterisierung eines Flächenstücks ergebenden Pixel (bzw. Fragmente) einen Intensitätswert. Die Beleuchtungsberechnung ist abhängig von der Bildraumkomplexität (Anzahl Fragmente) des Objekts.

4 Echtzeitfähige Simulation photorealistischer Effekte

Mit lokalen Beleuchtungsmodellen allein wären kaum die heute erzielbaren, photorealistisch wirkenden Szenendarstellungen möglich. Die Texturierungsfunktionalität moderner Computergraphik-Hardware schafft es, vielfältige photorealistische Effekte in Echtzeit zu imitieren. Entsprechende Verfahren ergänzen lokale Beleuchtungsmodelle durch Simulation ausgewählter physikalische Effekte für spezielle Situationen, z. B. durch Umgebungsspiegelung und Lichtbrechung an Wasseroberflächen oder Lichtausbreitung in Wolken oder Schatten. Die notwendigen Parameter werden dazu i. a. in 2D-Texturen kodiert, die zusammen mit den lokalen Beleuchtungsmodellen beim Rendering, meist in mehreren Durchläufen (Multipass-Rendering), ausgewertet werden. Bekannte Verfahren (AKENINE-MÖLLER & HAINES 2002) umfassen:

- *Bump Mapping* zur Simulation der Erscheinung einfacher Mikrostrukturen in Oberflächen (z. B. Muster oder Kratzer).

- *Displacement Mapping* zur Modellierung geometrischer Deformationen bei komplexen Mikrostrukturen.
- *Material Mapping* zur Modellierung unterschiedlicher Reflexionseigenschaften.
- *Environment Mapping* für Spiegelungen der Umgebung auf Oberflächen.
- *Shadow Mapping* zur bildbasierten Ermittlung von Schattenbereichen in einem Bild.

Mit Phong-Beleuchtung, Phong-Schattierung und texturbasierten Renderingverfahren lassen sich Bilder hoher visueller Komplexität synthetisieren. Abb. 2 nutzt z. B. Umgebungsreflexion auf der Wasseroberfläche und detaillierte botanische 3D-Vegetationsmodelle, um den Grad an Photorealismus zu erhöhen. Allerdings lösen diese Verfahren nicht das zentrale Problem, die ambiente Beleuchtung in Szenen glaubwürdig wiederzugeben.



Abb. 2: Ansicht eines 3D-Stadtmodells mit lokaler Beleuchtung ergänzt durch ein texturbasiertes Renderingverfahren für Umgebungsreflexionen auf der Wasseroberfläche.

5 Globale Beleuchtungsmodelle

Die globale Beleuchtung hat zum Ziel, indirekte Beleuchtungseinflüsse, also die Interaktion zwischen Objekten, nachzubilden. Dazu wird an einem Oberflächenpunkt alles einfallende Licht – direkt oder indirekt – berechnet. Allgemein lässt sich diese Funktion als rekursives Integral darstellen:

$$I(x, x') = V(x, x') \left[\varepsilon(x, x') + \int_S \rho(x, x', x'') I(x', x'') dx'' \right]$$

- $I(x, x')$ ist die Lichtintensität, die von einem Punkt x' zu einem Punkt x übertragen wird.
- $V(x, x')$ beschreibt die Sichtbarkeit bzw. Distanz zweier Punkte. Ist der direkte Weg blockiert, gilt $V(x, x') = 0$, ansonsten $V(x, x') = f_{att}$.
- $\varepsilon(x, x')$ codiert das von x' zu x emittierte Licht.
- $\rho(x, x', x'')$ beschreibt den Lichtanteil, der von einem Punkt x'' über x' zu x übertragen wird. Um diese indirekte Beleuchtung vollständig zu erfassen, muss über alle Punkte x'' der Szene S integriert werden.
- Diese Formel ist rekursiv, da zur Beleuchtungsberechnung an einem Punkt die Beleuchtung aller anderen Punkte der Szene benötigt werden.

Eine exakte Lösung dieses Integrals lässt sich nicht berechnen. Stattdessen gibt es Approximationsverfahren, die für jeweils ausgewählte Effekte geeignet sind. Die beiden klassischen Verfahren sind:

- *Raytracing* (Strahlverfolgung) verfolgt Lichtstrahlen vom Auge „zurück“ zur Lichtquelle und kann so insbesondere Spiegelungen und harte Schatten sehr gut darstellen, was den Charakter der frühen Computergraphik stark geprägt hat.
- *Radiosity* (gleichmäßige Lichtausbreitung) hingegen kommt einer Lichtsimulation näher, indem es die Ausbreitung einer konstanten Lichtmenge schrittweise berechnet als Lichttransfer von einem Flächenstück auf alle anderen Flächenstücke. Mit jedem Schritt nähert sich die Lösung einem Gleichgewichtszustand an. Dieses Verfahren eignet sich vor allem für diffuse Beleuchtung und wird in 3D-Architekturmodellen häufig eingesetzt, wobei die Berechnung für ausgewählte Einzelbilder und nicht in Echtzeit erfolgt.

Neuere Verfahren der globalen Beleuchtung setzen statistische Methoden, sogenannte *Monte-Carlo-Methoden*, ein. Sie simulieren dabei Lichtteilchen und deren Interaktion mit der Umgebung. Um nicht jedes einzelne Photon berechnen zu müssen, verwenden sie Wahrscheinlichkeiten und Methoden der Statistik, um von einer vergleichsweise kleinen Menge an Photonen auf das Endergebnis zu schließen. Hauptvertreter ist das *Photon Mapping* (JENSEN 2001), das sich speziell für Brechungseffekte wie Kaustiken oder Lichtstreuung in Wolken eignet. Ein sehr junges Verfahren ist das spektrumbasierte Rendering, das Farben über ein Spektrum repräsentiert und so farbtreue und natürliche Darstellungen erreicht.

In Echtzeitanwendungen lassen sich solche globalen Beleuchtungsmodelle nicht unmittelbar einsetzen, da Objektinteraktionen die benötigte Rechenleistung potenzieren. Die Rechenzeit pro Bild lässt sich i. a. eher in Stunden als in Sekunden messen. Für Szenen, in denen sich weder Objekte noch Lichtquellen ändern, kann in einem Vorberechnungsschritt ein globales Beleuchtungsmodell angewendet und die Ergebnisse gespeichert werden. Enthält die Szene dynamische Komponenten, lassen sich trotzdem häufig Anteile der globalen Beleuchtung vorberechnen und dann mit den Ergebnissen lokaler Beleuchtungsmodelle kombinieren. Die vorberechneten Ergebnisse müssen blickwinkelunabhängig sein und können mittels Texturen oder als Eckpunktparameter dem Echtzeitrendering übergeben werden.

6 Globale Beleuchtung mit Ambient Occlusion

Ambient Occlusion („Umgebungsverdeckung“) ist ein hardwarenahes Verfahren zur Approximation globaler Beleuchtungsberechnungen (LANDIS 2002). Der Begriff „Ambient Occlusion“ bezeichnet ursprünglich die Abschwächung von ambientem Licht aufgrund von Abschattung durch nahe Objekte. Das Verfahren simuliert für ein gegebenes 3D-Modell eine allgemeine, von außen kommende Umgebungsbeleuchtung, z. B. die Beleuchtung durch Tageslicht. Die Helligkeit einer Objektoberfläche wird danach berechnet, wieviel Umgebungslicht jeden Punkt der Oberfläche direkt erreicht. Dadurch werden z. B. die Wände von Innenhöfen nach unten i. a. dunkler (Abb. 3). Ein Punkt auf dem Boden eines Innenhofes ist dunkler, weil das Tageslicht aus den meisten möglichen Einstrahlrichtungen durch die angrenzenden Häuser blockiert wird.

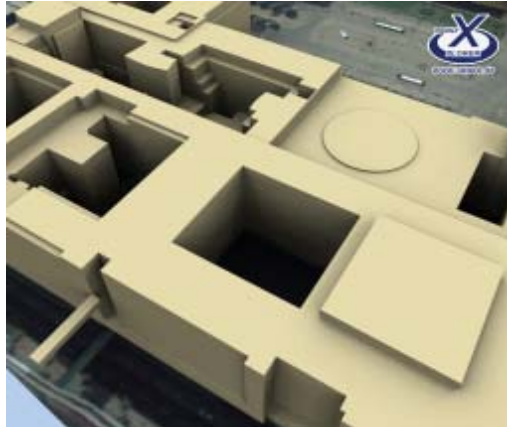


Abb. 3: Beispiel für die Abdunklung der Lichtintensität an den Innenhofwänden zum Boden hin.

Theoretisch wird die Helligkeit eines Punktes x , die durch direkte Tageslicht-Einstrahlung entsteht, wie folgt approximiert (KONTKANEN & LAINE 2005):

$$A(x, n) = \frac{1}{\pi} \int_{\Omega} V(x, \omega) [\omega \cdot n] d\omega$$

Dabei gibt n die Oberflächennormale im Punkt x an, und es wird über die Halbkugel Ω um die Richtung n integriert. Die Halbkugel Ω kann dabei als Gesamtheit aller möglichen Lichteinfallrichtungen verstanden werden. Für eine Lichtrichtung ω bezeichnet $V(x, \omega)$ die *Sichtbarkeitsfunktion*. Diese ist 0, wenn das Licht aus Richtung ω durch ein anderes Objekt verdeckt ist und andernfalls 1. Die Gewichtung mit dem Skalarprodukt $(\omega \cdot n)$ bewirkt, dass senkrecht einfallendes Licht stärkeren Einfluss hat.

Um eine Berechnung praktisch zu ermöglichen, wird eine Näherung verwendet. Dazu wird die Menge Ω aller möglichen Einstrahlrichtungen ersetzt durch eine endliche Menge von Einstrahlrichtungen $\Omega' \subset \Omega$, die gleichmäßig über die Halbkugel Ω verteilt sind. Somit ergibt sich:

$$A(x, n) \approx \frac{1}{|\Omega'|} \sum_{\omega \in \Omega'} V(x, \omega) [\omega \cdot n]$$

Je mehr repräsentative Lichtrichtungen gewählt werden, d. h., für größere $|\Omega'|$, steigt die Genauigkeit und damit die Qualität der Darstellung, aber auch der Berechnungsaufwand.

Die Helligkeit kann nur für eine endliche Anzahl von Oberflächenpunkten berechnet werden. Deshalb wird die Lichtberechnung in den Bildsyntheseprozess integriert, d. h. die Helligkeitsberechnung wird für jedes berechnete Pixel durchgeführt, an dem das Modell sichtbar ist. Für die Bestimmung der Helligkeit eines Pixels p wird dazu die Beleuchtungsberechnung für den Punkt x_p durchgeführt, der im Pixel p sichtbar ist. Die Berechnung von $V(x, \omega)$ wird durch einen Strahlentest erreicht. Schneidet der Strahl mit Ursprung in x und Richtung ω ein anderes Objekt, wird $V(x, \omega)$ zu 0 ausgewertet, sonst zu 1.

Das Ambient-Occlusion-Verfahren erreicht realistische Beleuchtungseffekte und ist im Vergleich zu aufwendigeren Verfahren wie Radiosity oder Photon-Mapping deutlich schneller berechenbar. Eine Einschränkung des Verfahrens besteht darin, dass indirekte Beleuchtung, d. h., der Effekt, dass angestrahlte Flächen auch Licht an ihre Umgebung abgeben, nicht berücksichtigt wird. Dadurch ist das Verfahren weniger geeignet für die Simulation von Innenraum-Beleuchtung, z. B. für Räume mit Deckenflutern. Für Außenansichten hingegen

ist es sehr gut geeignet, da hier die wahrgenommene Schattierung maßgeblich durch die direkte Tageslicht-Einstrahlung hervorgerufen wird.

In der soweit beschriebenen Ursprungsform ist das Ambient-Occlusion-Verfahren nicht für eine Darstellung in Echtzeit geeignet. Zum einen erfordert die Berechnung eines Strahlentests pro Lichtrichtung und Pixel zu hohen Rechenaufwand, zum anderen muss bei jeder Änderung der Blickperspektive die Beleuchtung für jedes Pixel neu berechnet werden.

7 Ambient Occlusion für virtuelle 3D-Stadtmodelle

Um das Ambient-Occlusion-Verfahren für virtuelle 3D-Stadtmodelle einsetzen zu können, muss der Rechenaufwand pro Bild möglichst gering ausfallen. Eine größtmögliche Darstellungsgeschwindigkeit wird erreicht, wenn die Helligkeitsinformationen vorberechnet und dem 3D-Stadtmodell als „Light Map“-Sammlung zur Verfügung stehen. Für die Darstellung in Echtzeit werden 3D-Stadtmodelle normalerweise durch Dreiecke repräsentiert. Die Integration vorberechneter Intensitätsinformationen in ein 3D-Stadtmodell kann eckpunktbezogen oder texturbasiert erfolgen.

7.1 Eckpunkt-bezogene Berechnung

Die Beleuchtungsberechnung wird für jedes Dreieck pro Eckpunkt durchgeführt. Das Ergebnis wird als Intensitätswert gespeichert und durch Gouraud-Shading auf die Fläche angewendet.

Eckpunktbezogene Berechnungen sind in Renderingverfahren sehr verbreitet und können für beschränkt große Modelle meist effizient realisiert werden (SATTLER ET AL. 2004). Für 3D-Stadtmodelle ist diese Variante jedoch problematisch. Die Genauigkeit der Beleuchtungsberechnung hängt von der Größe der Dreiecke ab, d. h. für große Dreiecke wird die Berechnung zu ungenau. Z. B. würde eine geringe Abschattung durch einen kleinen Wand-Vorsprung durch die Interpolation auf die gesamte angrenzende Wand verteilt. Feinere Unterteilung des Modells in mehr Dreiecke würde die geometrische Komplexität drastisch erhöhen und damit die Darstellungsgeschwindigkeit entsprechend einschränken. Hinzu kommt, dass bei 3D-Stadtmodellen in der Praxis keine einheitliche Granularität der geometrischen Modelle gegeben ist.

7.2 Berechnung in Texturen

Die Beleuchtungsberechnung erfolgt in Texturen, die während der Darstellung auf die Oberflächen der einzelnen Objekte im Stadtmodell projiziert werden. Dazu wird zunächst eine *Textur-Parametrisierung* benötigt, d. h. eine Abbildungsvorschrift, die jedem Dreieck einen Texturbereich zuordnet. Die Textur-Parametrisierung muss außerdem eindeutig sein, d. h. jede Position innerhalb der Texturen darf höchstens einem Punkt im Modell zugewiesen werden.

Liegt eine Textur-Parametrisierung vor, kann die Helligkeitsberechnung pro Textur-Pixel erfolgen. Die Texturen werden wie folgt berechnet:

- Für jedes Textur-Pixel p wird die zugeordnete Position x_p im Modell ermittelt, d. h. der Punkt im Modell, dessen Farbe durch die Farbe von p bestimmt wird.
- Die Helligkeit wird für x_p berechnet und als Farbe von p in der Textur gespeichert.

Abb. 4 zeigt ein Beispiel einer auf diese Weise berechneten Beleuchtungstextur, die in einem Vorberechnungsschritt entsteht und dadurch nicht zur Beeinträchtigung der eigentlichen Darstellungsgeschwindigkeit führt.

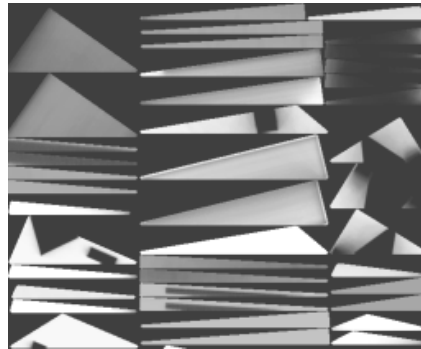


Abb. 4: Beispiel einer Beleuchtungstextur für eine Menge von Dreiecken.

7.3 Optimierung

Die explizite Berechnung der Strahlentests (s. letzter Abschnitt) würde dazu führen, dass die Vorberechnung je nach Modellkomplexität, verwendeter Hardware und Anzahl der Lichtrichtungen mehrere Stunden oder Tage dauert. Die rechenintensive Auswertung der Strahlentests muss daher optimiert werden. Dazu wird für jede Lichtrichtung ω eine Parallelprojektion des Modells aus der Richtung ω erzeugt. Für jedes Pixel wird dabei der Abstand von der virtuellen Kamera ermittelt. Das Ergebnis ist ein *Tiefenbild*, d. h. ein Bild, das anstelle von Farben Abstandswerte enthält. Das Tiefenbild wird verwendet, um effizient zu bestimmen, ob ein Oberflächenpunkt p aus der Richtung ω sichtbar ist. Unter Einbeziehung der Optimierung kann der Algorithmus vereinfachend wie folgt zusammengefasst werden:

```
// erzeuge Graustufenbilder
// initialisiert mit Helligkeit 0
T1,...,Tn = prepareTextures();

For each  $\omega$  in  $\Omega'$  {
    // erzeuge Tiefenbild durch
    // Parallelprojektion aus Richtung  $\omega$ 
    DepthImage D $\omega$  = createDepthImage( $\omega$ );
    for i=1...n {
        for each pixel p in T $_i$  {
            // errechne den Punkt im Modell,
            // auf den p abgebildet wird
            Vector3D x $_p$  = mapTo3DSpace(p);

            // errechne Oberflächennormale in x $_p$ 
            Vector3D n = computeSurfaceNormal(x $_p$ );

            // berechne Sichtbarkeit
            // mit Hilfe des Tiefenbildes
            // Ergebnis:
            // 1 für sichtbar; 0 für nicht sichtbar
            double V = computeVisibilityValue(x $_p$ ,  $\omega$ , D $\omega$ );

            // addiere Helligkeitsbeitrag
            // für x $_p$  aus Richtung  $\omega$ 
            p.brightness += V * dotProduct( $\omega$ , n);
        }
    }
}
```

Beleuchtungstexturen tragen wesentlich zur Komplexität einer 3D-Szene bei. Bei größeren Stadtmodellen (ab ca. 1.000 Gebäude) entstehen so große Mengen an Beleuchtungstexturen, dass diese nicht mehr direkt verarbeitet werden können, da zum einen der Texturspeicher der Hardware begrenzt ist und zum anderen das Umschalten zwischen Texturen (texture switching) hohe Laufzeitkosten verursacht. Hier wird in unserer Implementierung auf ein skalierendes Multiresolutionsverfahren für Texturdaten, der Textur-Atlas-Ansatz (BUCHHOLZ & DÖLLNER 2005) zurückgegriffen, sodass unbeschränkt große texturierte Stadtmodelle interaktiv visualisiert werden können.

8 Photorealistisches Rendering von CityGML-Stadtmodellen

In einem gemeinsamen Projekt mit der 3D Geo GmbH, Potsdam wird derzeit das echtzeitfähige, interaktive 3D-Viewer- und Content-Management-System LandXplorer für CityGML-basierte virtuelle 3D-Stadtmodelle entwickelt. Die vorgestellten Verfahren zur globalen Beleuchtung und weitere texturbasierte photorealistische Renderingverfahren sind dort im Rendering-Kern enthalten.



Abb. 5: Anwendung von Ambient Occlusion für texturierte Modelle im LandXplorer CityGML Viewer.

Der LandXplorer CityGML Viewer erlaubt es mittlerweile Beleuchtungstexturen für ein 3D-Stadtmodell automatisch zu berechnen, wie in Abb. 5 demonstriert. Falls das Modell bereits Detail-Texturen, z. B. Fassaden- und Dachtaturen, besitzt, werden diese in der Visualisierung mit den Beleuchtungstexturen kombiniert. Die Vorberechnungszeit für kleinere Modelle liegt auf einem aktuellen Standard-PC unterhalb einer Minute. Beispielsweise wurden Bodenplatte und Gebäudemodelle auf einem AMD Athlon 4300+ und GeForce 7900 GT in insgesamt etwa 13 Sekunden vorberechnet. Die Erzeugung der Beleuchtungstexturen erfolgt vollautomatisch, d. h. es sind keine Konfigurationseinstellungen notwendig. Nach einmaliger Vorberechnung der Beleuchtungstexturen ist eine Bewegung durch das virtuelle 3D-Modell in Echtzeit möglich. Eine Neuberechnung ist nur dann erforderlich, wenn das Modell geometrisch verändert wird.

9 Ausblick

Die ersten Erfahrungen mit Ambient Occlusion zur Beleuchtung von 3D-Stadtmodellen fielen positiv aus. Nutzer kommentierten, dass selbst einfache Blockmodelle an Plastizität und graphischer Qualität gewinnen. Insgesamt wird dem Betrachter die präzise Wahrnehmung räumlicher Kompositionen deutlich vereinfacht, wodurch sich z. B. Positionierungsfehler

leichter identifizieren lassen. Abb. 6 zeigt bzgl. des Geländemodells leicht schwebende Gebäude. Dieser Effekt wird bei lokaler Beleuchtung vollständig maskiert, durch Ambient-Occlusion jedoch mittels Verschattungsinformationen unmittelbar sichtbar.

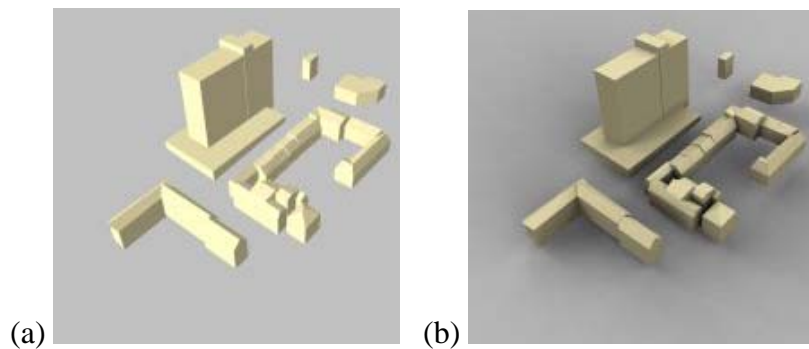


Abb. 6: Mit Ambient Occlusion sind Positionierungsfehler wie z.B. schwebende Häuser im Vergleich zu lokaler Beleuchtung sofort erkennbar.

Die Implementierung des vorgestellten Verfahrens zur globalen Beleuchtung innerhalb der LandXplorer-Softwareplattform für 3D-Geovisualisierung wird weiter ausgebaut und verbessert:

- Die Beleuchtung wird zur Zeit für gleichmäßigen Lichteinfall aus allen Richtungen berechnet, was in etwa den Lichtverhältnissen bei vollständig wolkenbedecktem Himmel entspricht. Durch unterschiedliche Gewichtung der Lichtrichtungen werden später auch vorgegebene Sonnenstände in die Beleuchtung mit einfließen können.
- Momentan wirkt die Lichtberechnung nur auf die Helligkeit des Modells. Wenn zukünftig die Himmelsfarbe (z. B. bei Dämmerung) berücksichtigt wird, kann für unterschiedliche Tageszeiten eine realistischere Beleuchtung erzielt werden.
- Weitere echtzeitfähige photorealistische, physikalisch-basierte 3D-Renderingverfahren, wie z. B. für Wasseroberflächen und Vegetationszonen, werden derzeit in das LandXplorer-System integriert.
- Die berechneten Beleuchtungsintensitäten werden zukünftig als Fassaden-Texturatlas exportiert und können so die graphische Qualität auch für andere 3D-Visualisierungs- und Simulationssysteme zugänglich machen.

Die Arbeiten zeigen zugleich, dass 3D-Geovisualisierung längst von der reinen Anwendung computergraphischer Funktionen zur aktiven Ausgestaltung und Spezialisierung computergraphischer Algorithmen übergeht. Nur so kann in der Zukunft weiter die Qualität von 3D-Geovisualisierung für die menschliche Wahrnehmung und Kognition optimiert werden. Ambient Occlusion leistet hierzu einen wichtigen Beitrag.

Weiterführende Links

www.hpi.uni-potsdam.de/3d Web-Site des Fachgebiets Computergraphische Systeme am Hasso-Plattner-Institut der Universität Potsdam.

www.3dgi.de Web-Site der Nachforschergruppe 3D-Geoinformationen, gefördert durch das BMBF im Rahmen des InnoProfile-Programms.

www.3dgeo.de Web-Site mit frei erhältlichem CityGML-Viewer.

www.citygml.org Web-Site der CityGML-Initiative.

Literatur

Akenine-Möller, T., Haines, E.: Real-Time Rendering. Second Edition, A K Peters, 2002.

Buchholz, H., Döllner, J.: View-Dependent Rendering of Multiresolution Texture-Atlases. Proceedings of the IEEE Visualization 2005, Minneapolis, pp. 215-222, Okt. 2005.

Foley, J., van Dam, A., Feiner, S., Hughes, J.: Computer Graphics, Principles and Practice. Second Edition, Addison Wesley, 1990.

Jensen, H.: Realistic Image Synthesis Using Photon Mapping. AK Peters, 2001.

Kontkanen, J., Laine, S.: Ambient Occlusion Fields. Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games, pp. 41-48, ACM Press, 2005.

Landis, H.: Production-Ready Global Illumination. Proceedings of ACM SIGGRAPH 2002, Course Notes #16 (RenderMan in Production), pp. 87-102, July 2002.

Rost, R.: OpenGL Shading Language. Addison-Wesley, 2004.

Sattler, M., Sarlette, R., Zachmann, G., Klein, R.: Hardware-Accelerated Ambient Occlusion Computation. In Vision, Modeling, and Visualization 2004, pp. 331-338, 2004.