

DISTRIBUTED RAYTRACING

Anwendungen des Algorithmus für fortgeschrittene Bildberechnung

Autor: Uwe Gleiß

V O R B E M E R K U N G E N

Um die kurzen Erklärungen dieses Skriptes verstehen zu können, ist es notwendig das grundlegende Raytracing-Verfahren verstanden zu haben. Ein hierfür geeignetes Skript ist auf der CoGra-FLG-Internetseite zu finden.

Distributed Raytracing steht frei übersetzt für verteilte Strahl(rück)verfolgung. Dabei wird der Raytracing-Algorithmus zur Erzielung verschiedener Effekte für einen einzelnen Bildpunkt mehrfach angewandt. Bei der Beschreibung der verschiedenen Anwendungen wird hier bewusst auf die dahinter stehende Mathematik verzichtet, Ziel ist ein genaueres Verständnis, das auch hilft, die Verfahren in der Praxis gezielter anzuwenden.

A N T I A L I A S I N G

Problem

Wird pro Bildpunkt genau ein Sehstrahl in die Szene geschickt, so entstehen z.B. an den Kanten von Objekten unschöne Treppelinien. Dies erklärt sich einfach dadurch, dass ein einzelner Strahl natürlich nur zwischen einem Treffer und einem Verfehlen des Objektes unterscheiden kann. Da der Bildpunkt eben kein mathematischer Punkt ist, müsste je nach Abstand seiner Mitte zur Kante der Bildpunkt aus einer Mischung der beiden Farben bestehen, um realistisch zu wirken.



Nahaufnahme ohne Antialiasing



Der gleiche Ausschnitt mit Antialiasing

Lösung

Pro Bildpunkt werden mehrere Strahlen ins Bild geschickt, die über diesen Bildpunkt entweder nach einem festen Muster, oder aber auch zufällig verteilt werden. Hier erklärt sich zum ersten Mal der Begriff Distributed Raytracing. Die Ergebnisse der Einzelberechnungen werden gemittelt, wodurch Punkte nahe einer Kante automatisch eine passende Mischfarbe erhalten.

Da selbst einfaches Antialiasing (vielleicht übersetzbar mit „Antistufung“) die Rechenzeit schnell nach oben treiben würde, bedient man sich noch eines einfachen Optimierungsverfahrens: Nur dann, wenn die Färbung eines Pixels stark von der seiner Nachbarn abweicht wird das Antialiasing überhaupt aktiviert.

Umsetzung in Cinema 4D

Hier wird generell zwischen zwei verschiedenen Verfahren unterschieden, zwischen denen in den Rendervoreinstellungen gewählt werden kann:

Bei „Geometrie“ werden nur Objektkanten geglättet. Ungenauigkeiten an Schattenkanten oder innerhalb von Texturen bleiben erhalten (was z.B. für unschöne Moirre-Effekte bei weit entfernten Mustern sorgt). Erst „Bestes“ Raytracing glättet das gesamte Bild, wobei angegeben werden kann, ab welchem, prozentualen Unterschied zweier Nachbarpunkte das Antialiasing eingesetzt werden soll.

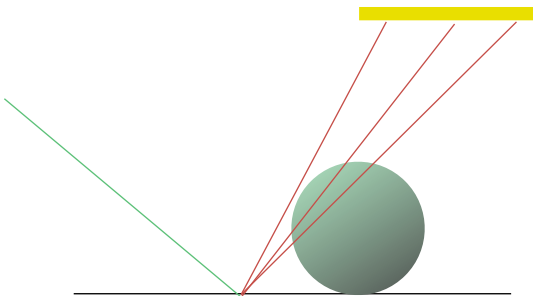
F L Ä C H E N S C H A T T E N

Problem

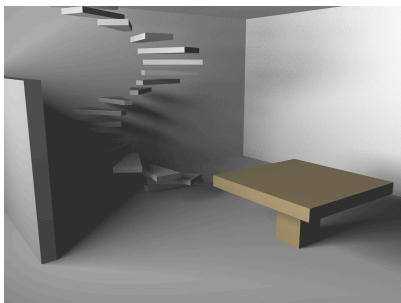
Beim klassischen Raytracing kommen Punktlichtquellen zum Einsatz. Diese erzeugen unvermeidlich sehr scharfe Schatten. Shadowmaps dagegen sorgen für weiche Schatten, allerdings sind diese an ihrem Rand gleichmäßig weich gezeichnet. Der Rand eines realen Schattens ist aber um so schärfer, je näher man sich an dem Objekt befindet, von dem er geworfen wird.

Lösung

Eine Möglichkeit besteht darin, ein reales, flächiges Licht durch eine Vielzahl von Punktlichtern zu ersetzen. Damit aber so ein realistisch weicher Rand des Schattens entsteht sind recht viele Lichtquellen notwendig, was die Rechenzeit stark erhöht. Außerdem entsteht auch bei vielen Lichtern noch ein Mehrfachschatten (ähnlich denen eines Fußballspielers im Flutlicht). Mit diversen Verfahren könnten diese zwar weichgezeichnet werden, aber ein anderes Herangehen ist meist geschickter:



Vom Auftreffpunkt des Sehstrahls (grün) werden mehrere, zufällig verteilte Strahlen (distributed rays, rot) weiter zur Fläche der Lichtquelle verfolgt. Je mehr dieser Strahlen dabei auf ein Objekt treffen, desto mehr wird der ursprüngliche Auftreffpunkt demnach im Schatten liegen.



Beleuchtung mit einer großen Flächenlichtquelle

Durch die statistische Verteilung der Strahlen erhält der weiche Schatten ein körnig anmutendes Aussehen. Erst mit einer ausreichend großen Anzahl von Strahlen verschwindet dieser Effekt, was die Berechnungszeit erheblich steigern kann. Allerdings ist ein ansprechendes Ergebnis im Gegensatz zu einem Areal aus Punktlichtern oft schneller zu erreichen.

Umsetzung in Cinema 4D

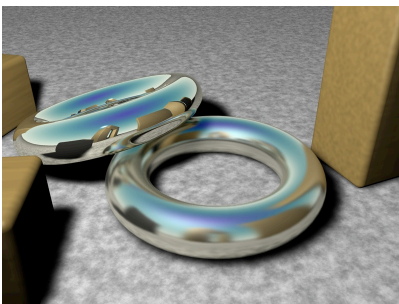
Cinema variiert die Anzahl der ausgeschickten Schattenstrahlen (Samples = Proben) je nach Notwendigkeit. Der Benutzer kann in den Einstellungen der Lichtquelle festlegen, in welchem Bereich dabei variiert werden darf. So kann der Rechenaufwand bzw. das Endergebnis recht gut gesteuert werden. Flächenschatten sind zwar aufwändig, steigern aber den Realismus zu einem noch recht erträglichen Preis in Sachen Zeitaufwand.

Die Form eines Flächenlichtes muss kein Rechteck sein, Cinema 4D bietet hier verschiedenste Möglichkeiten an, bis hin zur Verwendung von Objekten oder Splines, die dann die Verteilung der Beleuchtung steuern. Mit deren Hilfe kann anstatt einer klassischen Dreipunkt-Beleuchtung mit einer einzigen Lichtquelle eine interessante Ausleuchtung erzielt werden.

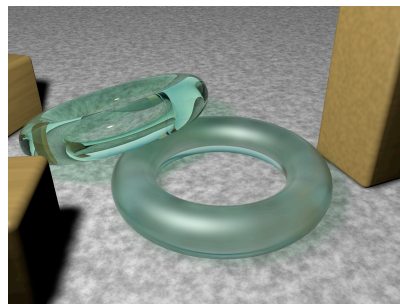
M A T T E R E F L E X I O N E N

Problem

Viele Oberflächen spiegeln ihre Umgebung nur diffus wieder und nicht wie Klarlack oder gar poliertes Metall. Dieser Effekt entsteht durch winzige Unebenheiten der Oberfläche, die allerdings nicht all zu sehr in ihrer Richtung voneinander abweichen. Es entsteht ein leicht verschwommenes Spiegelbild. Ganz ähnlich ist die Wirkung von Milchglas, das ja durchaus noch durchsichtig ist, aber alles dahinter nur verschwommen erkennen lässt (auch beschlagene Scheiben oder Spiegel zeigen ein solches Aussehen).



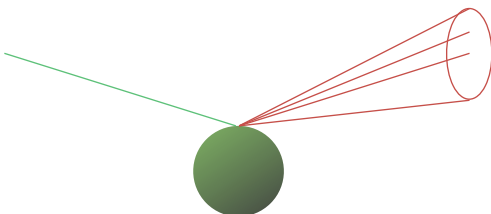
Normale und mattierte Spiegelung



Transparenz ohne und mit Streuung

Lösung

Die Unebenheiten durch ein Relief im Material der Oberfläche zu simulieren klappt oft nur unzureichend. Auch hier kann verteiltes Raytracing den gewünschten Effekt erzielen (natürlich wieder auf Kosten der Rechenzeit).



Trifft ein Sehstrahl auf die spiegelnde bzw. auf die transparente Oberfläche, so wird wieder ein ganzes Bündel von statistisch verteilten Strahlen weitergeschickt. Diese Verlaufen alle in einem Kegel um die eigentlich korrekte Bahn des reflektierten bzw. gebrochenen Strahls. Je weiter dieser Kegel geöffnet ist, desto stärker ist der resultierende Effekt.

Umsetzung in Cinema 4D

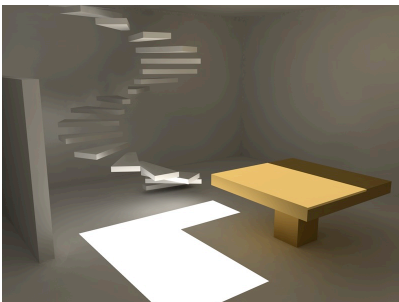
Die Streuung wird im Spiegelungs- bzw. im Transparenzkanal aktiviert. Auch hier kann gesteuert werden, wie viele Teststrahlen verwendet werden, um diese Effekte zu berechnen. Bei der Streuung selbst sollte man nicht all zu hohe Werte verwenden, da sonst der Effekt so stark wird, dass man ihn auch trickreich durch andere Techniken nachahmen kann, die wesentlich weniger Rechenzeit verschlingen (in den obigen beiden Bildern wurde ca. 20% Streuung eingesetzt). Es sollte auch daran gedacht werden, die anderen Materialkanäle passend zu verändern (mattes Glas besitzt z.B. ein viel breiteres und auch schwächeres Glanzlicht).

G L O B A L I L L U M I N A T I O N

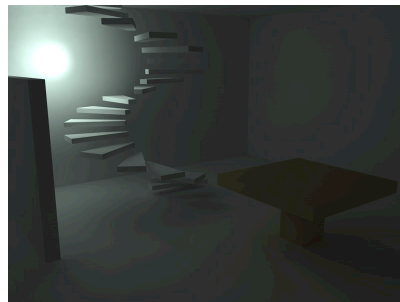
Problem

Der Begriff Global Illumination wird oft als Zusammenfassung für verschiedene Effekte benutzt (Radiosity, HDR-Lighting, Photons). An dieser Stelle geht es um die so genannte Radiosity. Damit wird die Tatsache beschrieben, dass in der Realität nicht nur Lichtquellen Licht aussenden, sondern auch jede beleuchtete Fläche ihrerseits wieder zur indirekten Lichtquelle wird. Dadurch wird ein Zimmer hell, obwohl nur auf den Boden direktes Licht durch das Fenster fällt. Nahestehende Objekte werden in der Farbe eines stark beleuchteten Nachbarobjektes eingefärbt.

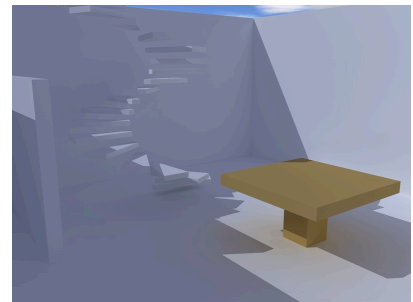
Da Licht in dieser Weise von jedem Objekt auf jedes Objekt fallen kann würde der Rechenaufwand für eine exakte Lösung schnell mörderisch groß und damit eine Berechnung völlig ausgeschlossen. Auf der anderen Seite ist es gerade all dieses hin und her wandernde Licht, das eine Szene real erscheinen lässt.



Beleuchtung durch ein Oberlicht



Eine einzelne Lichtquelle an der Wand



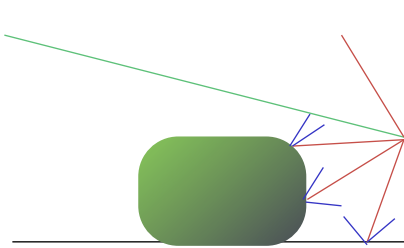
Beleuchtung durch einen Cinema 4D Himmel

Lösung

Eine vollständige Lösung würde bedeuten, dass vom Auftreffpunkt eines Sehstrahls wiederum ein komplettes Bild der umgebenden Szene berechnet werden müsste. Dabei wäre auch noch der Abstand zu den einzelnen anderen Elementen zu berücksichtigen, um korrekt zu bestimmen, wie stark der Auftreffpunkt durch all diese Anteile aufgehellt wird. Die notwendigen Berechnungen wären damit aber noch lange nicht abgeschlossen, denn all diese Oberflächenpunkte werden ja wiederum von ihrer Umgebung beeinflusst, usw.

Eine Methode wäre nun, all diese gegenseitige Beeinflussung in ein atemberaubend großes Gleichungssystem zu integrieren, das dann nur noch gelöst werden muss (PovRay z.B. geht grob betrachtet diesen Weg). Zum Glück kann aber auch das oben zuerst angedeutete Verfahren umgesetzt werden, wenn man es genügend einschränkt:

Es genügt, vor der eigentlichen Bildberechnung nur für einige, geschickt gewählte Punkte die Helligkeiten aus ihrer Umgebung zu berechnen. Später verwendet man für die dazwischen liegenden Bildpunkte entsprechende Mischungen dieser Werte. Außerdem genügt es erneut, ein Bündel von zufällig verteilten Strahlen vom Auftreffpunkt in die Szene zu schicken (im Extremfall einige Hundert und nicht mehrere Tausend, wie es bei einem kompletten Bild aus Sicht des Auftreffpunktes notwendig würde).



Zuletzt kann auch noch bei der nie endenden Weiterverfolgung des Lichtes gespart werden: meist genügt es, zwei (wie in der Zeichnung) bis drei Schritte weiter zu gehen, wobei jedes Mal auch die Anzahl der statistischen Strahlen weiter reduziert werden kann, da der Einfluss auf den ursprünglichen Punkt mit jedem solchen Schritt weiter abnimmt.

Umsetzung in Cinema 4D

Global Illumination (kurz GI) wird in den Rendervoreinstellungen aktiviert. Hier kann die grundlegende Genauigkeit und damit die Anzahl der Punkte gesteuert werden, für die die Umgebungshelligkeit abgetastet wird (Cinema 4D verteilt diese ersten, abzutastenden Punkte selbst an Hand der Geometrie möglichst sinnvoll). Wie auch schon bei den anderen Verfahren mit verteiltem Raytracing ist die Anzahl der Teststrahlen einstellbar, wobei Cinema 4D selbst steuert, wo eher weniger und wo mehr Strahlen verwendet werden. Lediglich die Unter- und Obergrenze ist einstellbar (was aber auch genügt).

Als Alternative zum obigen Verfahren kann auch der Modus „Stochastisch“ gewählt werden. Dabei werden schon die ersten Strahlen (in der Zeichnung grün) zufällig und nicht an Hand der Geometrie verteilt. Je nach Szene und auch gewünschtem Endergebnis kann mal die eine, mal die andere Methode sinnvoller sein, die rein stochastische Methode sorgt für ein körnig wirkendes Bild, was ja durchaus im Sinne des Künstlers sein kann.

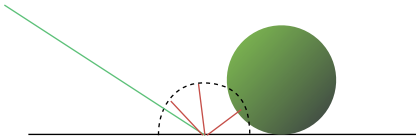
A M B I E N T O C C L U S I O N

Problem

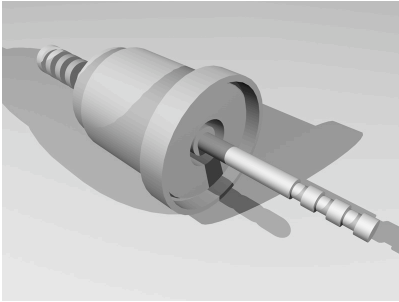
In diesem Fall entsteht das Problem hauptsächlich aus dem Verlangen, Ergebnisse in ähnlicher Qualität wie mit Global Illumination schneller erreichen zu wollen.

Lösung

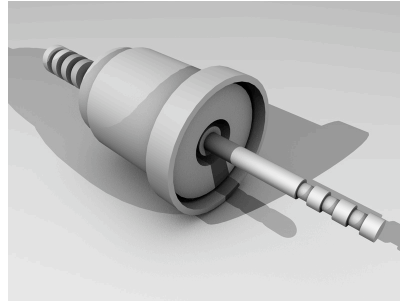
„Ambient Occlusion“ könnte man sehr frei übersetzen mit: „Verdeckung des Umgebungslichtes“. Das wiedergespiegelte Licht der Umgebung erreicht vor allem Ecken und Kerben relativ schlecht. Anstatt nun solche Bereiche mittels Radiosity (s.o. bei Global Illumination) zu ermitteln, kann man den Rechenaufwand erheblich reduzieren, indem vom Auftreffpunkt eines Sehstrahls nur die umliegende Geometrie abgetastet wird (nicht deren Helligkeit).



Je mehr der Teststrahlen innerhalb eines einstellbaren Radius auf umliegende Oberflächen treffen, um so mehr wird die Helligkeit des Auftreffpunktes verringert. Dieser Algorithmus kann durchaus rechenintensiv werden, bleibt aber hinter dem Aufwand für Radiosity deutlich zurück.



Ohne Ambient Occlusion



Mit Ambient Occlusion

Umsetzung in Cinema 4D

Auch Ambient Occlusion kann in den Rendervoreinstellungen aktiviert werden. Die Einstellungsmöglichkeiten erklären sich zum größten Teil selbst (nach Studium der obigen Erklärung). Der Effekt kann aber auch gezielt nur für einzelne Objekte genutzt werden, indem man ihn z.B. im Leuchtenkanal eines Materials verwendet.

P H O T O N S

Problem

Da Raytracing von der Kamera ausgeht und dann vom Auftreffpunkt aus den Weg mit einem einzelnen Strahl weiter zum Licht verfolgt, kann es einige Effekte nicht darstellen, die aber in der Optik völlig normal sind. Ein solcher Effekt ist die Tatsache, dass spiegelnde und transparente Materialien das Licht je nach Form des Objektes streuen oder bündeln. Bei nur einem Strahl zur Lichtquelle kann unmöglich festgestellt werden, dass am Auftreffpunkt das Licht durch die darüber liegende Linse gebündelt wird.

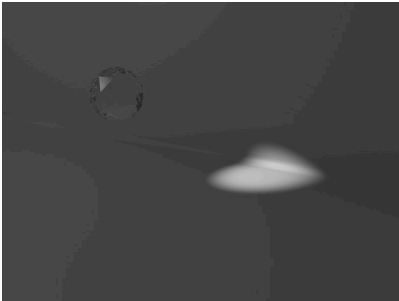
All die Lichtfiguren, die durch Spiegelungen oder Brechung entstehen, wie der Brennpunkt einer Linse, aber auch die hübschen Lichtspiele auf dem Tisch, geworfen durch die Brechung in einem geschliffenen Weinglas, bezeichnet man als Kaustiken. Viele Raytracingprogramme bezeichnen deshalb solche Effekte auch als Caustics.

Lösung

In diesem Fall nutzt man distributed Raytracing in umgekehrter Richtung. Von der Lichtquelle, deren Licht die Kaustiken erzeugen soll werden vor der Berechnung der Szene eine bestimmte Anzahl an Strahlen (hier physikalisch nicht ganz korrekt als Photonen bezeichnet) in die Szene hinein verfolgt. Der Weg über Reflexion und Brechung wird dabei korrekt ermittelt, wobei die Energie des Photons meist auch ein wenig abnimmt.

Trifft das Photon schließlich auf eine Oberfläche mit diffuser Reflexion (entspricht in Cinema dem Kanal Farbe), so wird seine Position und die an dieser Stelle ankommende Energie gespeichert. All diese Berechnungen laufen meist recht schnell ab (einige 100.000 Photonen sind meist unproblematisch), allerdings wird für die Zwischenspeicherung der Photonendaten eine nicht unerhebliche Menge an Speicher benötigt.

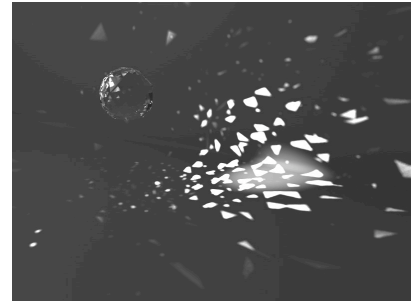
Bei der Bildberechnung wird nun zusätzlich zu den sonst üblichen Verfahren die Umgebung eines Auftreffpunktes eines Sehstrahls nach Photonen abgesucht, die dort „gelandet“ sind. Die Energie dieser Photonen wird summiert und das Ergebnis benutzt, um die Oberfläche passend aufzuhellen. Werden zu wenige Photonen verwendet, oder ist der Radius der Umgebung zu klein, innerhalb der gesucht wird, so werden einzelne Photonen als Punkte sichtbar, was meist unerwünscht ist.



Szene ohne Photonen, ein Spotlicht



Szene mit wenigen Photonen



Ausreichende Anzahl von Photonen

Umsetzung in Cinema 4D

Damit Caustics wirklich sichtbar werden sind mindestens zwei Schritte notwendig:

Die Szene muss mindestens eine Lichtquelle enthalten, die auch Photonen erzeugt. Die Details sind in den Lichtquelleneigenschaften einzustellen. Zu empfehlen ist ein Spot, da die Photonen gleichmäßig über den Beleuchtungsbereich verteilt werden. Eine Punktlichtquelle würde also einen Großteil der Photonen wirkungslos in den Himmel schicken. Oft ist es sinnvoll eine Lichtquelle zu kopieren und dann die Kopie als Spot zu nutzen, der ausschließlich Photonen mit passender Richtung erzeugt und nicht zusätzlich die Szene beleuchtet. Für Letzteres dient die Originallichtquelle.

Zusätzlich müssen die Kaustiken auch noch in den Rendervoreinstellungen aktiviert werden, um Wirkung zu zeigen.

WEITERE ANWENDUNGEN

Depth of Field (DOF)

Tiefenunschärfe, wie sie reale Kameras unvermeidlich erzeugen, kann ebenfalls mittels distributed Raytracing erzeugt werden. Allerdings geht dies oft mit geschickter Nachbearbeitung (im Raytracingprogramm selbst oder danach) schneller und fast ebenso überzeugend.

Motionblur

Bewegungsunschärfe kann distributed Raytracing (hier über die Zeit verteilt) nutzen, um bewegte Objekte weichzeichnen.