

Realistic, Hardware-accelerated Shading and Lighting



Paper von Wolfgang Heidrich und Hans-Peter Seidel

Vortrag von UrsENZler
Betreut von Martin NÄf
GDV Seminar 2000

Motivation

- HW-Rendering bis heute:
 - Performance
- Zukunft:
 - Qualität
 - Funktionalität

Inhalt

- Lokale Beleuchtung
 - Alternative Beleuchtungsmodelle
- Globale Beleuchtung
 - Environment Maps
- Normal Maps
- Resultate
- Zukunft
- Persönliche Beurteilung

Lokale Beleuchtung

- Phong-Modell nicht ausreichend
 - Physikalisch bessere Modelle
 - Methode unabhängig vom Modell
- Unterstützt durch heutige Hardware
 - Wenig Speicher
 - Einfache Umsetzung

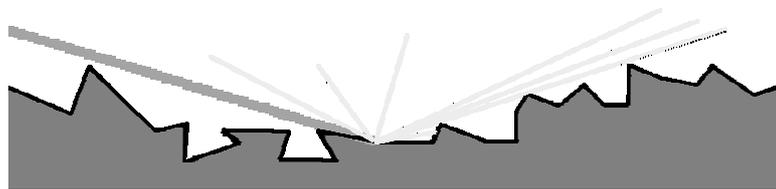
Torrance-Sparrow Model

$$I_0 = \frac{F \cdot G \cdot D}{\pi \cdot \cos \beta} \cdot I_i$$

F: Fresnel term

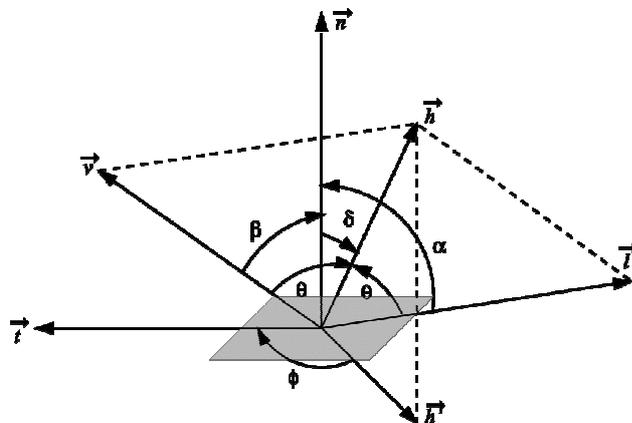
G: Geometry attenuation term

D: Micro facet distribution function



Torrance-Sparrow Model (2)

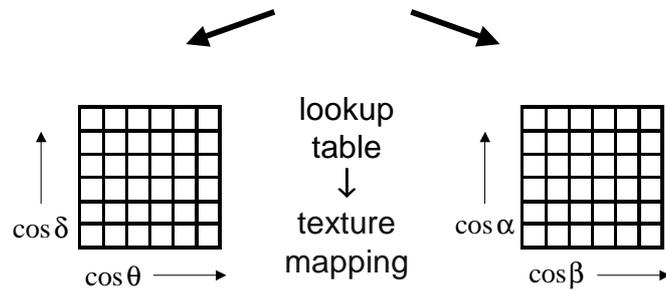
$$I_0 = \frac{F(\cos \theta) \cdot G(\cos \alpha, \cos \beta) \cdot D(\cos \delta)}{\pi \cdot \cos \beta} \cdot I_i$$



Lookup Tables

$$I_0 = F(\cos \theta) \cdot D(\cos \delta) \cdot \frac{G(\cos \alpha, \cos \beta)}{\pi \cdot \cos \beta} \cdot I_i$$

$$I_0 = T_1(\cos \theta, \cos \delta) \cdot T_2(\cos \alpha, \cos \beta) \cdot I_i$$



Implementation

- Spekular
 - Single-pass
 - 2 Texturen simultan
 - 2 Passes
 - Je 1 Textur mit alpha-blending
- Diffus
 - Zusätzliche Textur
 - Zusätzlicher Pass

Texturkoordinaten

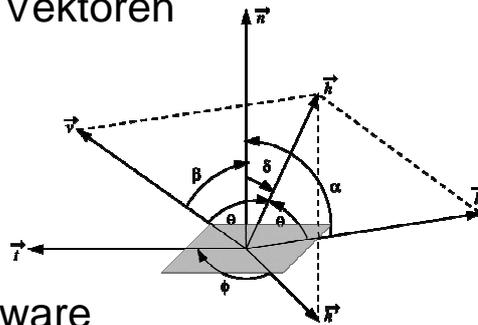
- Für normalisierte Vektoren

$$\vec{n} \cdot \vec{l} = \cos \alpha$$

$$\vec{n} \cdot \vec{v} = \cos \beta$$

$$\vec{h} \cdot \vec{n} = \cos \delta$$

$$\vec{h} \cdot \vec{l} = \cos \theta$$

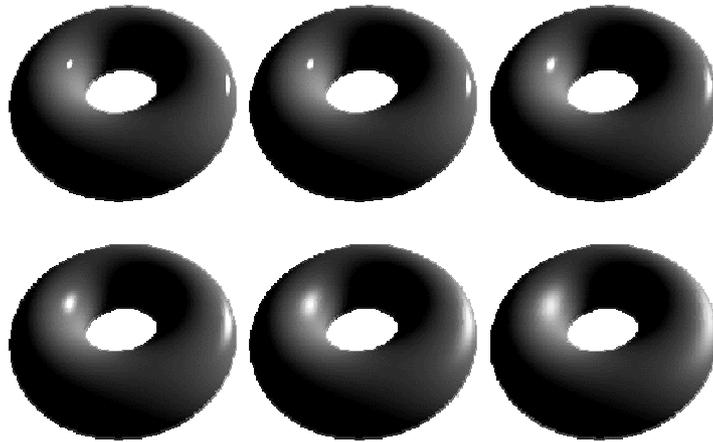


- Berechnet in Software
- Jedem Knoten als Texturcoordinate zugewiesen

Bemerkungen

- Licht- und Blickrichtung konstant
 - Einfache HW Implementation
- Approximativer Fehler
 - $F \cdot D \in [0, 1]$
 - $G / (\pi \cos \beta)$: Singularität für $\cos \beta \rightarrow 0$
 - Bereich in der Nähe der Singularität zu dunkel
- Unterstützt viele Beleuchtungs-Modelle
 - z.B. Phong

Beispiel



Torus mit verschiedener Rauheit berechnet mit dem Torrance-Sparrow Modell

Globale Beleuchtung

- Environment Maps
 - Erhöhung des Realismus
 - Guter Kompromiss zwischen
 - Qualität
 - Speicheranforderungen
 - Sphärisch
 - Schlechte Sampling-Rate
 - Kubisch
 - 6 Texturen
 - Anti-Aliasing bei Kanten schwierig

View-independant EM

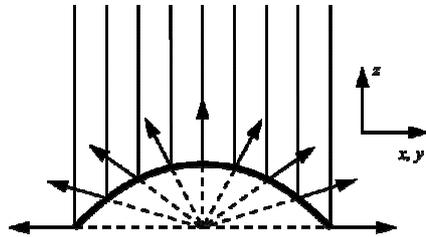
- Paraboloid

- $f(x, y) = \frac{1}{2} - \frac{1}{2}(x^2 + y^2), \quad x^2 + y^2 \leq 1$

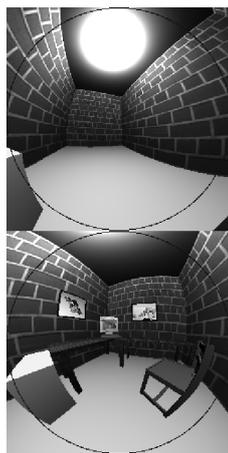
- 2 Hemisphären (2 Texturen)

- Besseres Sampling

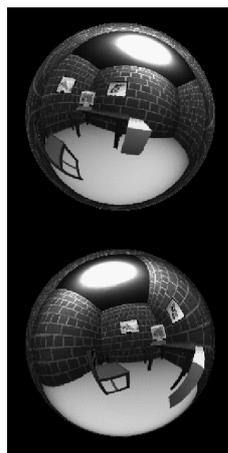
- Effiziente HW-Implementation



Beispiel



1 parabolische
Environment Map



Angewendet auf
2 Kugeln



Grenzbereich zw.
EM-Hälften

Materialeigenschaften

- Metallisch
 - Reflexion unabhängig vom Winkel des einfallenden Lichts
- Nicht-metallisch
 - Reflexion stark abhängig vom Winkel des einfallenden Lichts
 - Gewichtet mit Fresnel Term

Fresnel Term

- 1-dim Texture Map
- Variante 1
$$L_0 = F \cdot L_m + L_d$$

L_m : spiegelnder Anteil
 L_d : diffuser Anteil
- Variante 2
$$L_0 = F \cdot L_m + (1-F) \cdot L_d$$

Beispiel

$$L_0 = F \cdot L_m + L_d$$

$$L_0 = F \cdot L_m + (1-F) \cdot L_d$$



Refraction

1.5

200

Glossy Reflection & Transmission

- Prefiltered Environment Map
 - EM enthält vom Objekt ausgehendes Licht
- Rechenaufwendig
 - Kann nicht für jeden Frame gerechnet werden
 - View-independant Environment Maps

Beispiel

Parabolische
Environment
Map



Gefilterte Map

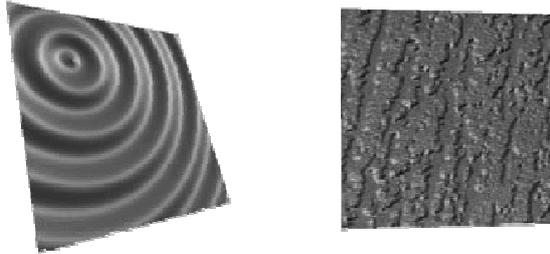
Angewendet
auf Torus

Transmission
durch Rechteck

Normal Maps

- Erhöhung der visuellen Komplexität
- Unterschiede zu Bump-Maps
 - Normalen sind direkt gegeben
(keine Berechnung der lokalen Normale
mehr notwendig)
 - Können direkt gemessen werden
 - Sind an Geometrie gebunden

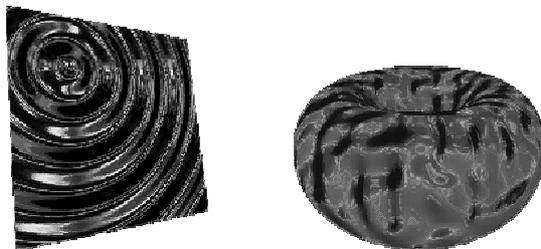
Beispiel



2 Phong beleuchtete Normal Maps
Rechts: von einer Tapete gemessen

Normal Maps (2)

- Alternative Beleuchtungsmodelle
 - Pixel Texture Extension
 - Lookup pro Pixel statt nur pro Knoten



Resultate

- Realistisches Schattieren und Beleuchten in HW
- Auf heutiger HW
- Frame Rates
 - 15-20 Hz SGI O2
 - >20 Hz SGI Octane MXE

Zukunft

- Mehrere Texturen gleichzeitig
- Pixel Texture Extension
 - Auf anderen Plattformen (nur SGI)
- Automatic texture generation modes
 - Winkelberechnungen in HW
- Automatic texture coordinate generation
 - Reflexionsvektor bei parabolischen EM

Zukunft (2)

- Neue Grafik-Pipeline
 - Phong ersetzen durch benutzergesteuertes Sample basiertes System
 - Materialeigenschaften als 2-dim Tabellen in HW laden
 - Indexberechnung für mehrere Lichtquellen gleichzeitig

Persönliche Beurteilung

- Vorteile
 - Bessere Beleuchtungsmodelle
 - Offen (kein spezifisches Modell)
 - Sehr schnell
- Nachteile
 - Heutige HW ist noch nicht soweit
 - Viele Passes notwendig
 - Pixel Texture Extension
 - Starke Modifikation der Pipeline