

Interactive Ray Tracing

Steven Parker

William Martin

Peter-Pike J. Sloan

Charles Hansen

Brian Smits

Peter Shirley

Von Valentina Hans

Systemarchitektur

Master Task

```
initialize model
initialize ray tracing slaves on each free CPU
loop
  update viewing information
  lock queue
  place all primary rays in queue
  unlock queue
  when the queue is empty redraw screen and handle user input
end loop
```

Slave Task

```
initialize memory
loop
  if queue is not empty then
    lock queue
    pop ray request
    unlock queue
    compute RGB for pixel
    write RGB into frame buffer pixel
  end if
end loop
```

- parallel shared memory Implementation
- jeder Pixel wird unabhängig von allen anderen bearbeitet
- Probleme mit Synchronisation

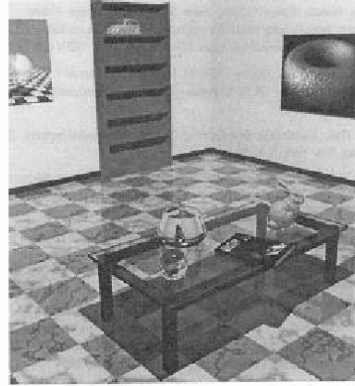
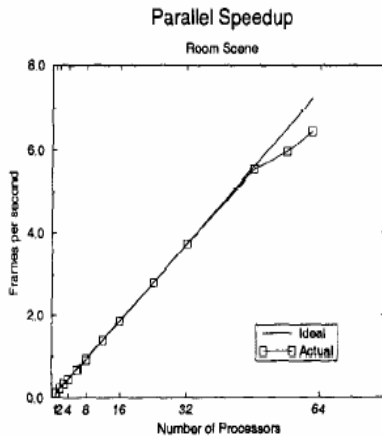
Systemarchitektur

- Unterteilung der Szene
 - Grid basiert
 - bounding volume
- Implementation läuft in einem konventionellen oder frameless Modus

Systemarchitektur: Konventionelle Operationen

- Gruppen von Strahlen werden jedem Prozessor zugewiesen
- Je größer die Gruppen desto weniger Synchronisation ist erforderlich
- Aber lange Ladebalancierungszeiten
- Anzahl in der Größe variierender Gruppen wird festgelegt

Systemarchitektur: Konventionelle Operationen



Systemarchitektur:Frameless Rendering

- statische Pixelverteilung in eine 1D-Renderingspur und wird in Klötze gebrochen
- Verteilung an die Prozessoren
- Buffer auf den Bildschirm zeichnen

Implementationsdetails:

- Licht und die Materialparameter bleiben statisch während der Blickpunkt geändert wird
- Änderung der traditionellen Schlüsselkomponenten eines Ray Tracers wie Beleuchtung, Materialmodelle, Schatten und Schnittpunktroutinen für Objekte

Implementationsdetails: Licht und Stoffe

- Wie werden in einem dynamischen Kontext Materialmodelle und Beleuchtung gehandhabt und geändert ?
- traditionelles Beleuchtungsmodell für eine Lichtquelle

$$L = k_d (l_a + s l_e \hat{\mathbf{n}} \cdot \hat{\mathbf{i}}) + s l_e k_h (\hat{\mathbf{h}} \cdot \hat{\mathbf{i}})^N + k_s L_s + k_t L_t$$

Implementationsdetails: Licht und Stoffe

- diffus: $k_h = k_s = k_t = 0$
- Metall: $k_d = k_t = 0$ und $k_h = k_s$
- Nicht leitend:

$$k_s(\hat{\mathbf{e}}, \hat{\mathbf{n}}) = R_0 + (1 - R_0)(1 - \hat{\mathbf{e}} \cdot \hat{\mathbf{n}})^5$$

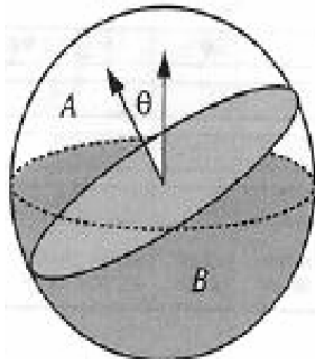
$$k_t(\hat{\mathbf{e}}, \hat{\mathbf{n}}) = 1 - k_s(\hat{\mathbf{e}}, \hat{\mathbf{n}}).$$

Implementationsdetails: Licht und Stoffe

- glatt:

$$\begin{aligned} L = & k_d l_a (1 - k_s(\hat{\mathbf{e}}, \hat{\mathbf{n}})) + \\ & s k_d l_e (1 - (1 - \hat{\mathbf{n}} \cdot \hat{\mathbf{e}})^5) (1 - (1 - \hat{\mathbf{n}} \cdot \hat{\mathbf{i}})^5) + \\ & s k_h (\hat{\mathbf{h}} \cdot \hat{\mathbf{i}})^N + \\ & k_s(\hat{\mathbf{e}}, \hat{\mathbf{n}}) L_s, \end{aligned}$$

Implementationsdetails:Ambiente Beleuchtung



- Hintergrundbeleuchtung (I_a) aufgeteilt zwischen zwei Intensitäten A und B
- Winkel Theta variiert zwischen 0 und Pi

Implementationsdetails:Ambiente Beleuchtung

- Theta = 0, $I_a = A$
- Theta = Pi/2, $(A + B)/2$

$$I_a(\theta) = \begin{cases} \left(1 - \frac{\sin \theta}{2}\right) A + \left(\frac{\sin \theta}{2}\right) B & \text{if } \theta < \frac{\pi}{2}, \\ \left(\frac{\sin \theta}{2}\right) A + \left(1 - \frac{\sin \theta}{2}\right) B & \text{otherwise.} \end{cases}$$

Implementationsdetails: Schatten

- Bei Ray Tracing - harte Schatten
- Methoden, die sanfte Schatten erzeugen vergrößern dramatisch die Berechnungszeit relativ zu der Berechnung der harten Schatten

Implementationsdetails: Schatten

- Verbesserung von Leistung durch
 - Zusammenfassen von Lichtstrahlen zu Bündeln
 - zur Beschleunigung: Strahlen im Bündel vorberechnen
- im Bündel den Durchschnitt bilden

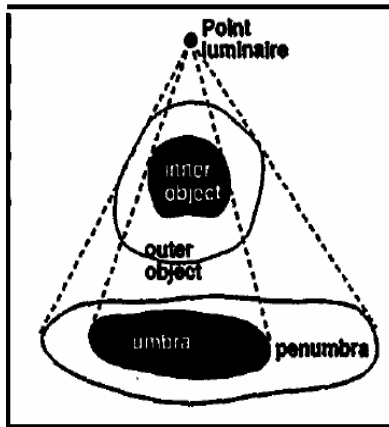
Implementationsdetails: Schatten

- Halbschatten, ist der Schatten eines halbdurchsichtigen Objekts, welches nicht zusätzlich von einem undurchsichtigen Objekt beschattet wird
- Der Algorithmus erzeugt Schatten eines sanftschattigen Objekts
- kugelförmige Lichtquelle

Implementationsdetails: Schatten

- Die Transparenz des halbdurchsichtigen Objekts erhöht sich von nicht Transparenz des undurchsichtigen Objekts zu der vollen Transparenz an der Grenze des halbdurchsichtigen Objekts
- die Steigung der Intensität im Halbschatten natürlich machen

Implementationsdetails: Schatten



- $s=0$ an der Fläche des undurchsichtigen Objekts und erhöht sich nicht linear mit der Distanz bis $s=1$ an der Grenze der Fläche des halbdurchsichtigen Objekts

Fazit:

- ist interaktiv auf wenigen Prozessoren
- nähert sich der realen Zeit für komplexe Umgebungen auf 64 oder mehr Prozessoren
- ist ideal um dynamische Effekte zu zeigen
- Arbeit mit großen Modellen z.B. Architekturdesign
- Lebensfähige Annäherung an high end parallel machine
- Softwareimplementation erlaubt es neue Optionen für interaktive Displays (Schattenvariationen, frameless rendering, anspruchvolles Licht) zu verwenden

Literatur:

- **Interactive Ray Tracing:** Steven Parker, William Martin, Peter-Pike J. Sloan, Charles Hansen, Brian Smits, Peter Shirley
<http://www.cs.utah.edu>
- **Computergraphik 2 Skript WS 02/03:**
Stefan Müller