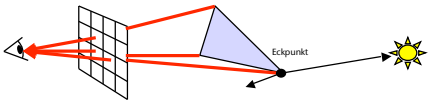
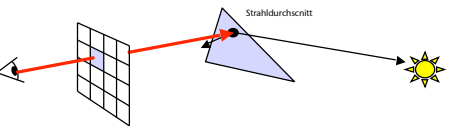


Scankonvertierung vs. Raytracing

- Scan-Konvertierung: Auswerten eines Strahls, der durch jeden Eckpunkt eines Objektes gesendet wird

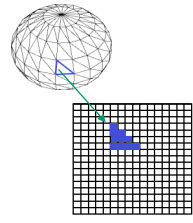


- Raytracing: Auswerten eines Strahls, der durch einen Bildschirmpixel gesendet wird



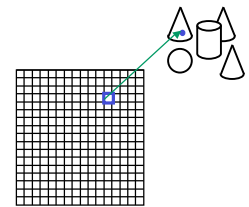
G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 95

- Zum Umwandeln einer Szene mittels Scan-Konvertierung ...



... scan-konvertiere jedes Dreieck

- Zum Umwandeln einer Szene mittels Raytracing ...



... verfolge für jedes Pixel einen Strahl

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 96

Vor- und Nachteile

- Scan-Konvertierung:
 - schnell (da nur Eckpunkte)
 - wird unterstützt von aktuellen Grafikkarten
 - geeignet für Echtzeitanwendungen
 - ad-hoc Lösung für Schatten, Transparenz
 - Keine Interreflexion
- Raytracing:
 - noch rel. langsam (Suche nach Schnittpunkten zwischen Strahlen und Objektprimitiven)
 - bisher von keiner kommerziellen Hardware unterstützt
 - Offline-Rendering-Verfahren
 - Allgemeine Lösung für Schatten, Transparenz und Interreflektion, Clipping und Culling

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 97

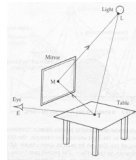
Bewertung des (einfachen) Raytracings

- Vorteile:
 - Eignet sich besonders für Szenen mit hohem spiegelndem und transparentem Flächenanteil
 - Kann beliebige Objektrepräsentationen verarbeiten (z.B. CSG, Rauch, ...)
 - Einzige Anforderung: man muß Schnitt zwischen Strahl und Objekt und die Normale in diesem Schnittpunkt berechnen können
 - Berechnung von Schatten, Reflexionen und Transparenzen sind ein inhärenter Teil des Raytracing-Algorithmus
 - Keine explizite perspektivische Transformation oder Clipping nötig

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 98

Nachteile

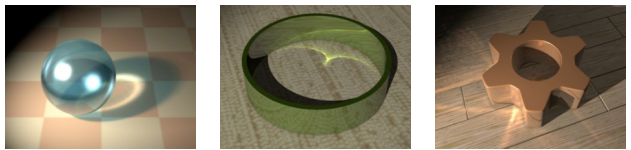
- Sehr viele Strahlen
 - Naives Ray-Casting: $O(p \cdot n \cdot l)$, $p = \# \text{ Pixel}$, $n = \# \text{ Polygone}$, $l = \# \text{ Lichtquellen}$
 - Anzahl Strahlen wächst exponentiell mit Rekursionstiefe!
- Keine indirekte Beleuchtung (Spiegel, "color bleeding" = diffuse indir. Bel.)
- Keine weichen Halbschatten
- Shading muß bei jeder Änderung der Kamera neu berechnet werden, obwohl diese nur von den Lichtquellen und den Objekten abhängen
- Für alle diese Nachteile wurden natürliche verschiedene Abhilfen vorgeschlagen



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 99

Beispiel für das Problem der indirekten Beleuchtung: Kaustiken

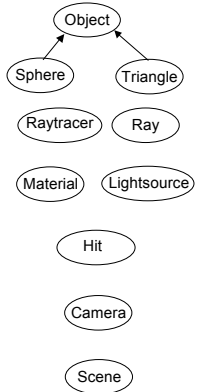
- Konzentration von Licht
- Lichtstrahlen treffen sich in einem Punkt
- Raytracing wird ineffektiv
- Nur 1 reflektierter Strahl wird betrachtet

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 100

Software-Architektur eines einfachen Raytracers

- Szene: einfache Liste von Kugeln, Dreiecken, etc.
- Schnittpunkt Strahl—Objekt
- Diffuse und spekulare Reflexion (Phong-Modell)
- Sekundärstrahlen
- Mögliche Erweiterungen
 - Lichtbrechung, Transparenzen
 - Besseres Oberflächenmodell (Fresnel)
 - Andere Objekte (Kegel, Zylinder, Polygon, ...)
 - Szene einladen



```

graph TD
    Object((Object)) --- Sphere((Sphere))
    Object --- Triangle((Triangle))
    Raytracer((Raytracer)) --- Sphere
    Raytracer --- Triangle
    Raytracer --- Ray((Ray))
    Raytracer --- Material((Material))
    Raytracer --- Lightsource((Lightsource))
    Hit((Hit)) --- Raytracer
    Camera((Camera)) --- Raytracer
    Scene((Scene)) --- Raytracer
  
```

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 101

Typische Raytracer-Klassen

- Lightsource (hier nur Einfache Punktlichtquelle)


```
Vector m_location; // Position
Vector m_color; // Farbe
```
- Material


```
Vector m_color; // Farbe der Oberfläche
float m_diffuse; // Diffuser / Spekularer
float m_specular; // Reflexionskoeff. [0..1]
float m_phong; // Phong-Exponent
```
- Ray

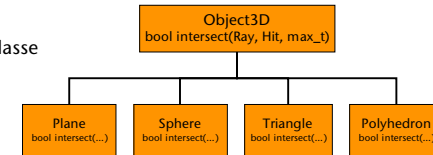

```
Vector m_origin; // Aufpunkt des Strahls
Vector m_direction; // Strahlrichtung
```

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 102

- Hit: Speichert Informationen über den Schnittpunkt

```
Ray m_ray;           // Strahl
float m_t;           // Geradenparameter t
Object* m_object;    // Geschnittenes Objekt
Vector m_location;   // Schnittpunkt
Vector m_normal;     // Normale am Schnittpunkt
```

- Object:
Abstrakte Basisklasse
für alle
Geometrie-
objekte



```
// Schnittpunkt von Strahl mit Objekt
virtual bool closestIntersection( Intersection * hit ) = 0;
virtual bool anyIntersection( const Ray & ray, float max_t,
                             Intersection * hit ) = 0;

// Normale am Schnittpunkt
virtual void calcNormal( Intersection * hit ) = 0;

// Material des Objekts
int getMaterialIndex() const;
```

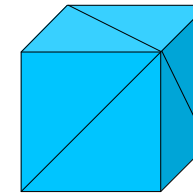
- Camera:
 - Alle Eigenschaften der Kamera, z.B. from, at, up, angle
 - Generiert Primärstrahlen durch alle Pixel
- Scene:
 - Speichert alle Daten der Szene
 - Liste aller Objekte
 - Liste aller Materialien
 - Liste aller Lichtquellen
 - Kamera

Das OBJ-File-Format

```


vertices {
v -1 -1 -1
v 1 -1 -1
v -1 1 -1
v 1 1 -1
v -1 -1 1
v 1 -1 1
v -1 1 1
v 1 1 1
}

triangles {
f 1 3 4
f 1 4 2
f 5 6 8
f 5 8 7
f 1 2 6
f 1 6 5
f 3 7 8
f 3 8 4
f 1 5 7
f 1 7 3
f 2 4 8
f 2 8 6
}
  
```



Aliasing

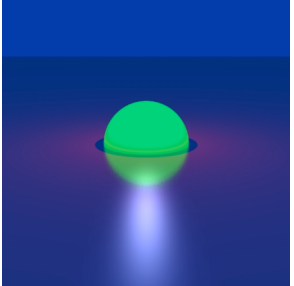
- Ein Strahl pro Pixel → typ. Aliasing-Artefakte:
 - Treppeneffekte
 - Moiré- Effekt



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 107

Distribution Ray Tracing

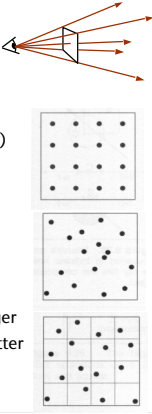
- Einfache Modifikationen des rekursiven Ray Tracings für
 - Antialiasing
 - Weiche Schatten
 - Tiefenschärfe
 - Spekulare Reflexion
- Bewegungsunschärfe
- Anderer Name früher:
 - „Distributed Ray Tracing“
 - ist aber sehr unglücklich ("distributed" = verteilt)



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 108

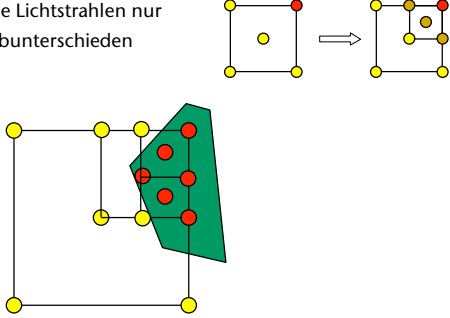
Anti-Aliasing beim Ray-Tracing

- Anstatt pro Pixel nur einen Strahl zu verfolgen werden mehrere Strahlen verfolgt und die resultierende Farbe gemittelt
- Methoden zur Auswahl der Punkte:
 - Regelmäßige Abtastung (Problem der Moire Muster)
 - Zufällige Abtastung (Problem des Rauschens)
 - Stratifikation, d.h. eine Kombination von regelmäßiger und zufälliger Abtastung, indem ein reglmäßiges Gitter zufällig gestört wird.



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 109

Adaptives Supersampling

- Idee: verschieße Lichtstrahlen nur bei großen Farbunterschieden
- Beispiel:
 
- Resultierende Farbe = Durchschnittsfarbe aller Samples, gewichtet mit dem Flächenanteil des Pixels, den das Sample "überdeckt"

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 110

Beispiel

	Aliasing	mit Anti-Aliasing
Punktlichtquelle		
Ausgedehnte Lichtquelle		

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 111

Weiche Schatten, Halbschatten

- Regionen:
 - "Vollschatten" (*umbra*)
 - Halbschatten (*penumbra*)
 - voll beleuchtet

XVI. Léonard de Vinci (1452-1519). Lumière d'une fenêtre sur une sphère ombreuse avec (en partant du haut) ombre intermédiaire, primitive, dérivée et (sur la surface, en bas) portée. Plume et lavis sur pointe de métal sur papier, 24 x 38 cm. Paris, Bibliothèque de l'Institut de France (ms. 2185; B.N. 2038. F° 14 r°).

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 112

In der Realität ...

<http://www.davidfay.com/index.php>

http://3media.initialized.org/photos/2000-10-18/index_gall.htm

<http://www.pa.uky.edu/~sciworks/light/preview/bulb2.htm>

klare Glühbirne matte Glühbirne

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 113

... und im Ray-Tracing

- Bisher: 1 Schattenstrahl

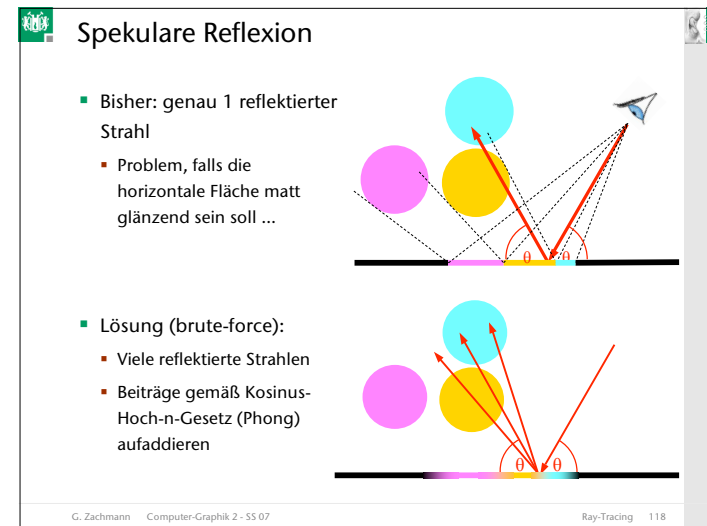
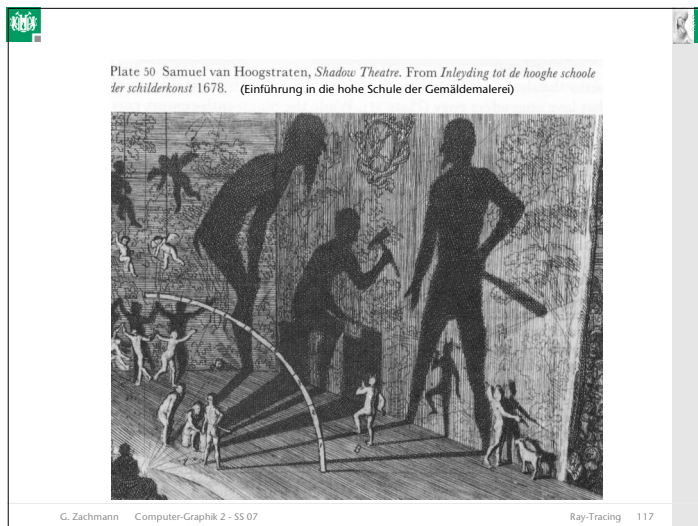
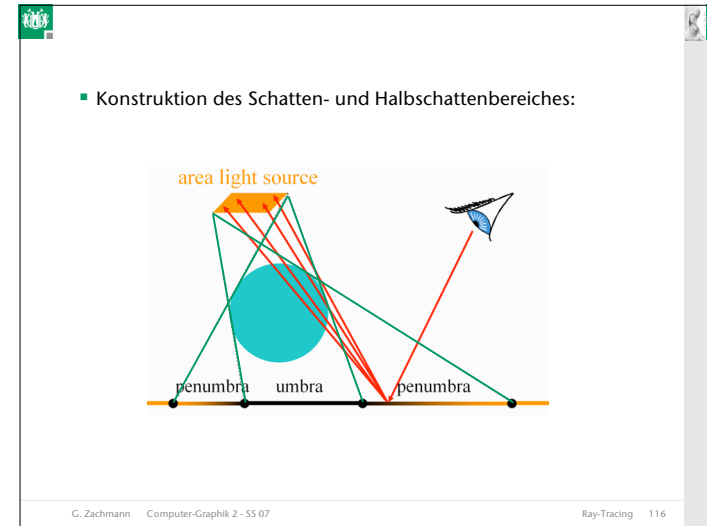
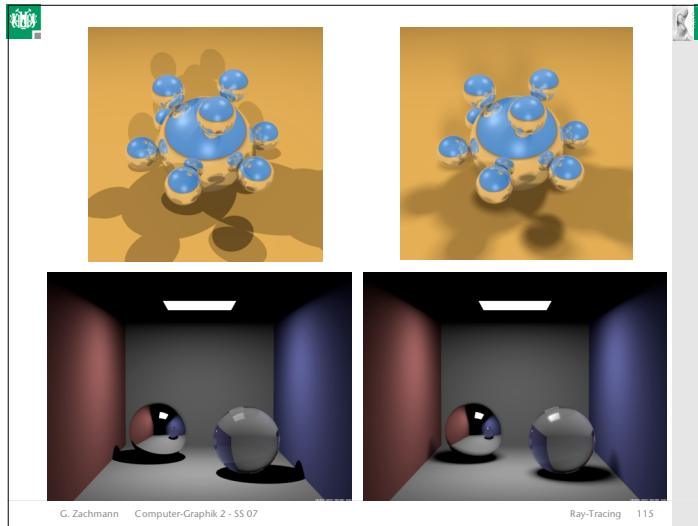
$$s_i = \begin{cases} 1, & \text{Lichtquelle sichtbar} \\ 0, & \text{nicht sichtbar} \end{cases}$$

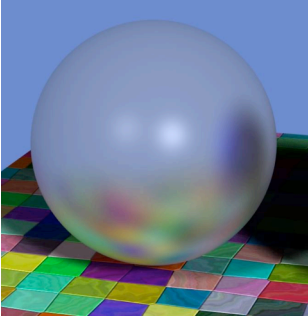
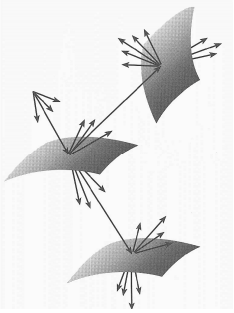
- Jetzt: mehrere Schattenstrahlen

$$s_i = \frac{\text{Anzahl sichtbarer Samples}}{\text{Anzahl Schattenstrahlen}}$$

- Drei Arten von Sampling der Lichtquelle:
 - Regelmäßige Abtastung der Lichtquelle
 - Zufällige Abtastung der Lichtquellen
 - Stratifizierte Abtastung

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 114

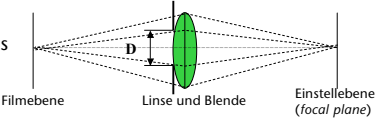
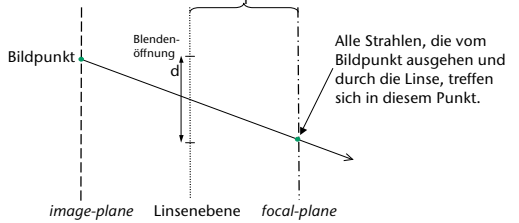


- Beispiel: 
- Strahlbaum: 

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 119

Tiefen(un-)schärfe

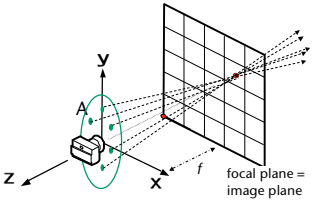
- Bisher: ideales Lochkameramodell
- Für Tiefenunschärfe muß man reale Kameras modellieren

Alle Strahlen, die vom Bildpunkt ausgehen und durch die Linse, treffen sich in diesem Punkt.

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 120

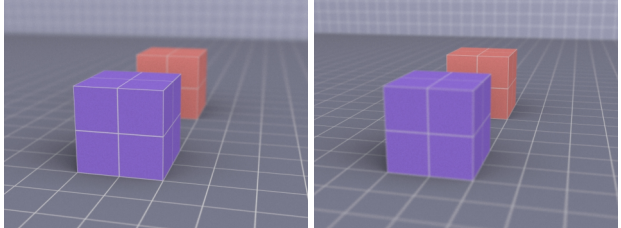
- Eine Klasse `LensCamera` würde die Strahlen also ungefähr so erzeugen:
 - Verteile Strahlen auf die gesamte Blendenöffnung und mitte



- Bemerkungen:
 - Sample die Scheibe (=Linse) stratifiziert
 - Achtung bei Kombination mit Anti-Aliasing

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 121

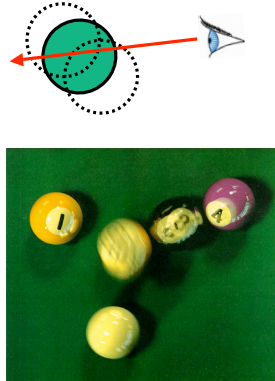
Beispiele



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 122

Motion Blur (Bewegungsunschärfe)

- Schieße viele Strahlen pro Pixel
- Wähle für jeden Strahl einen zufälligen Zeitpunkt t , $t_0 < t < t_1$
- Betrachte während des Schnitttests mit diesem Strahl alle Objekte an ihren Positionen $P = P(t)$ zu diesem Zeitpunkt t
- Middle Pixelwerte



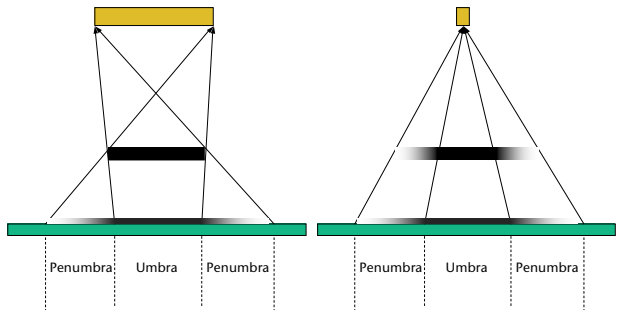
G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 123

Effiziente, visuell plausible Soft Shadows [1998]

- Soft Shadows mit Distribution Ray Tracing sind **sehr** teuer
 - Selbst bei 16 Schattenstrahlen hat man nur 16 Graustufen → Rauschen oder Machbänder im Halbschatten
- Ziel: weniger Schattenstrahlen & weicher Schattenverlauf
 - Non-Goal: physikalisch korrekter Schatten
 - Kann der Mensch i.A. sowieso nicht unterscheiden
- Zentrale Idee: ersetze **ausgedehnte Lichtquelle** und Objekte mit **harten Kanten** durch **Punktlichtquelle** und Objekte mit **weichen Kanten**

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 124

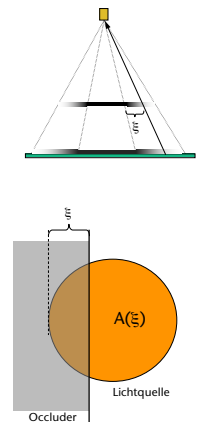
Veranschaulichung der zentralen Idee:



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 125

Details

- Annahme: wir haben schon eine Variable ξ , $0 < \xi < 1$, die den Abstand von der "harten" Kante des Objektes angibt
- Wie sieht die Dämpfungsfunktion $s(\xi)$ aus?
 - Klar: $0 < s(\xi) < 1$
- Übersetzt in die Situation mit flächigen Lichtquellen:
 - Annahme: kreisförmige Lichtquelle & lange, gerade, harte Kante
 - Wie sieht die Funktion der Fläche $A(\xi)$ aus?

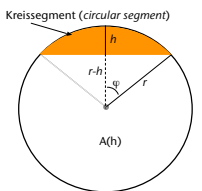


G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 126

- Zu berechnen: Fläche A des weißen Teils des Kreises
- Gegeben: h und r
- Annahme zunächst: $h < r$

$$A = \pi r^2 - (A_S - A_D)$$

A_S = Fläche des Kreissektors
 A_D = Fläche des Dreiecks



Kreissegment (circular segment)

A(h)

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 127

- Fläche des Dreiecks:

$$A_D = \frac{1}{2}s(r-h)$$

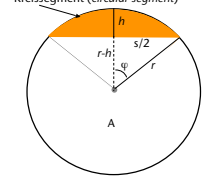
$$\left(\frac{s}{2}\right)^2 = r^2 - (r-h)^2$$

$$A_D = (r-h)\sqrt{h(2r-h)}$$

- Fläche des Kreissektors:

$$A_S = \varphi \cdot r^2$$

$$\varphi = \cos^{-1}\left(\frac{r-h}{r}\right)$$

$$A_S = r^2 \cos^{-1}\left(\frac{r-h}{r}\right)$$


Kreissegment (circular segment)

A

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 128

- Zusammen:

$$A = \pi r^2 - r^2 \cos^{-1}\left(\frac{r-h}{r}\right) + (r-h)\sqrt{h(2r-h)}$$

- Wir wollen eine Funktion, die von 0...1 geht:

$$A' := \frac{A}{\pi r^2} = 1 - \frac{1}{\pi} \left(\cos^{-1}\left(\frac{r-h}{r}\right) + \frac{r-h}{r} \sqrt{\frac{h(2r-h)}{r^2}} \right)$$

- Substituiere $\tau := \frac{r-h}{r}$, um einen Parameter in $[0,1]$ zu erhalten:

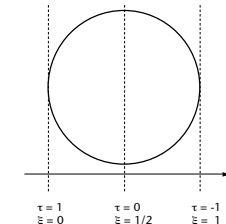
$$A'(\tau) = 1 - \frac{1}{\pi} \left(\cos^{-1}(\tau) + \tau \sqrt{(1-\tau)(1+\tau)} \right)$$

$$= 1 - \frac{1}{\pi} \left(\cos^{-1}(\tau) + \tau \sqrt{1-\tau^2} \right)$$

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 129

- Substituiere nochmals, da wir einen Parameter benötigen, der von 0...1 einmal über den Kreis streicht:

$$\xi = 1 - \frac{1}{2}(\tau + 1)$$

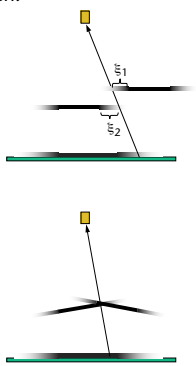
$$\tau = 1 - 2\xi$$


- Eine mögliche Approximation:

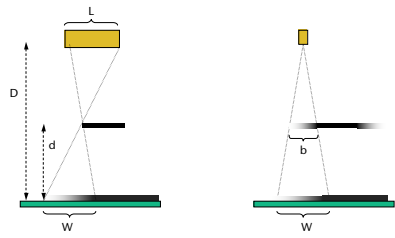
$$s(\xi) = 1 - (3\xi^2 - 2\xi^3)$$

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 130

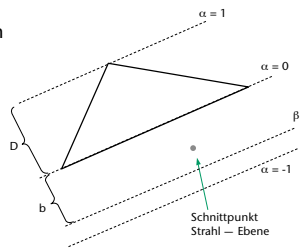
- Schattenstrahlen, die mehrere Objekte treffen:
 - Liefert mehrere ξ
 - Gesucht $\xi = f(\xi_1, \xi_2)$, so daß
 - $\xi_1 = 0 \rightarrow \xi = 0$
 - $\xi_1 = 1 \rightarrow \xi = \xi_2$
 - Möglichkeiten:
 - $s = 1 - ((1 - s_1) + (1 - s_2))$
 - $s = s_1 s_2$
 - $s = \min(s_1, s_2)$
 - Letztere scheint am besten zu wirken



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 131

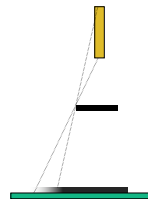
- Wie breit muß der "weiche" Rand eines Objektes sein?
 
 - Vom Strahltest haben wir d & D , von der Lichtquelle L
 - Zweimal Strahlensatz $\rightarrow b = \frac{d}{D}L$

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 132

- Wie berechnet man ξ :
- Idee: baryzentrische Koordinaten
 - Falls alle baryz. Koord. $> 0 \rightarrow$ Vollschatten
 - Sonst: bestimme die kleinste (am weitesten neg.) Koord.
 - Rechne b um baryz. Koord. β , wobei b auf die Dicke des Dreiecks entlang dieser Koord. bezogen wird
 - Rechne β um in ξ

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 133

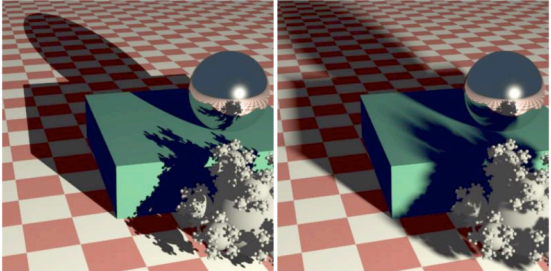
Bemerkungen

- Beschleunigungsdatenstruktur (demnächst) muß die Ausdehnung dieser "dickeren" Objekte verwenden (z.B. bei Bbox-Berechnung)
 - Macht Ray-Tracing etwas langsamer
 - Insbesondere für Strahltest, die **nicht** die weichen Kanten benötigen (alles außer Schattenstrahlen)
 - Zwei Beschleunigungsdatenstrukturen?
- Berücksichtige evtl. Winkel der Lichtquelle:
 
 - Der "gefakte" Vollschatten ist i.A. größer als der reale Vollschatten
 - Läßt sich vermutlich reparieren

Diplomarbeit ...

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 134

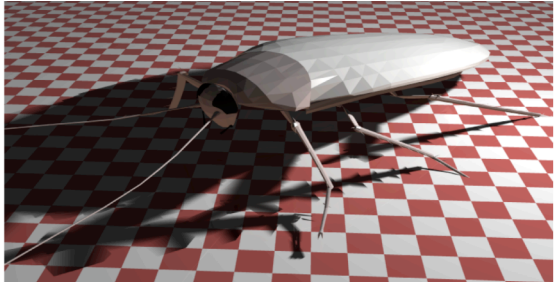
Beispiele



1 Schattenstrahl pro Pixel —
konventionelle harte Schatten

1 Schattenstrahl pro Pixel —
visuell plausible weiche Schatten

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 135



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 136

Software

- POV-Ray — Persistence of Vision Raytracer
 - open source
 - www.povray.org
- Radiance:
 - open source vom LBNL
 - physikalisch korrekt
 - <http://radsite.lbl.gov/radiance/>
 - Workshops

```

#include "colors.inc"
#include "stdcam.inc"
#include "textures.inc"

text{tff "arial.ttf"
  "hello world", 0.4, 0
  translate <-3, 0, -2>
  pigment (Red_Marble)
}

sphere { <0, 2, 0>, 2
  texture(
    Silver_Texture)
}

```

```

case polygon case.17.1
5
0.37 0 0.065
0.37 0

## stand
case cylinder stand
7
0.185 0.18 0
0.185 0.18 0.15
0.185

## generate curved screen
! generate auf 'c_x(s)' 'c_y(s)' 'h(t)'
20 1 -s -e 'c_x(s) = ((s-0.5)*0.36)'
[ ] | xform -t .18 3.4 .15 -t 0 .005
.07

```

G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 137

"But is it real-time?"

- Ray Tracing in der Vergangenheit war sehr langsam
- Inzwischen Echtzeit-Fähigkeit für einige Szenen
- OpenRT-Projekt: Real-Time Ray Tracing
 - Siehe <http://www.openrt.de>
- Special-Purpose-Hardware, PC-Cluster
- Nur eine Frage der Zeit, bis Commodity-Graphics-Hardware es kann



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 138



Ray tracing in Egoshooters


Example: Quake3 Demo

<http://graphics.cs.uni-sb.de/~sidapohl/egoshooter/>



Quake 3 mit Ray-Tracing. Plattform: Cluster mit 20 AMD XP1800
<http://graphics.cs.uni-sb.de/~sidapohl/egoshooter/>

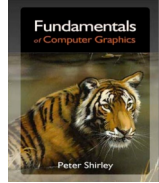
G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 139



Eine Anmerkung zu Typos

- Typos passieren auch auf den Folien
 - Keine Angst haben zu fragen!
 - Bitte teilen Sie mir Fehler mit

- Typos passieren sogar in Lehrbüchern
 - Ich selbst habe 2 nicht-triviale Fehler im Shirley-Buch, 2-te Auflage gefunden [WS 05/06]
 - Fazit: mitdenken, nicht einfach direkt kopieren



G. Zachmann Computer-Graphik 2 - SS 07 Ray-Tracing 140