

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

# Raytracing

Tobias G. Pfeiffer

tobias.pfeiffer@math.fu-berlin.de

Freie Universität Berlin  
Fachbereich Mathematik und Informatik

29. November 2007



- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens
- 3 Schnittpunktbestimmung
- 4 Farbbestimmung
- 5 Stochastisches/Verteiltes Raytracing
- 6 Laufzeitanalyse, Optimierung

# Gliederung

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

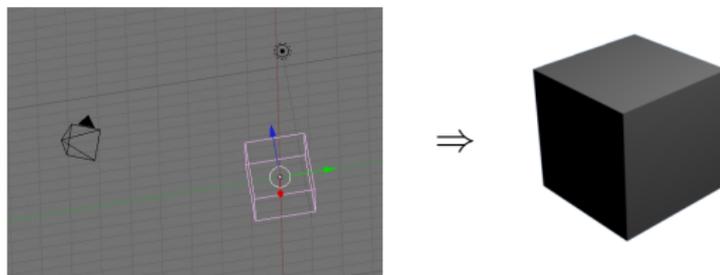
- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens
- 3 Schnittpunktbestimmung
- 4 Farbbestimmung
- 5 Stochastisches/Verteiltes Raytracing
- 6 Laufzeitanalyse, Optimierung

# Problemstellung, Motivation

Raytracing

Tobias  
Pfeiffer

- Erzeugung eines 2D-Bildes aus einer 3D-Szene (*Rendering*):



- gegeben:
  - geometrische Primitive (Polygone, Kugeln etc.) und deren Positionen
  - Lichtquellen (Farbe, Intensität, Typ)
  - Blickpunkt, Blickrichtung
- gesucht:
  - Rastergrafik mit Farbwert pro Pixel

# Entstehung eines Bildes im Auge

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

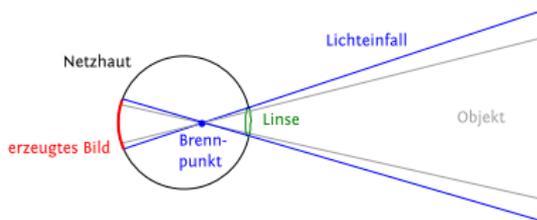
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Rendern bedeutet Nachempfinden der Ereignisse im menschlichen Auge
- Licht fällt durch Pupille ins Auge, trifft auf die Netzhaut:



- Farbrezeptoren nehmen Zusammensetzung der Wellenlängen des Lichts (*Spektrum*) wahr
- Gehirn erstellt aus empfangenen Impulsen ein Bild

# Umsetzung in der Computergrafik

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

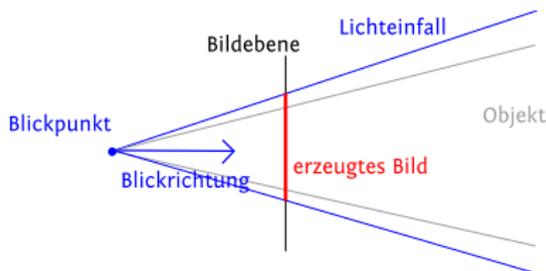
## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

- *Blickpunkt* und *Bildebene* im  $\mathbb{R}^3$  treten an Stelle von Brennpunkt und Netzhaut:



- rechteckige Teilmenge der Bildebene ist das zu berechnende Bild
- Aufgabe: berechne für jeden Pixel des Bildes die Farbe des eintreffenden Lichts → Ausbreitung des Lichts in der Szene muss berechnet werden

# Gliederung

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens**
- 3 Schnittpunktbestimmung
- 4 Farbbestimmung
- 5 Stochastisches/Verteiltes Raytracing
- 6 Laufzeitanalyse, Optimierung

# Raytracing als globales Beleuchtungsmodell

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

- Raytracing berechnet die Lichtausbreitung in der Szene (*globales Beleuchtungsmodell*)
- Idee: Verfolgung der „Lichtstrahlen“
- intuitiver Ansatz: *Forward-Raytracing*
  - schicke von jeder Lichtquelle Strahlen aus
  - verfolge den Weg des Strahls während des Wegs über die platzierten Objekte (z. B. Spiegelungen)
  - beende, wenn Lichtstrahl die Szene verlässt oder auf die Bildebene trifft
  - wird aus Performance-Gründen praktisch nicht verwendet

# Backward-Raytracing

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

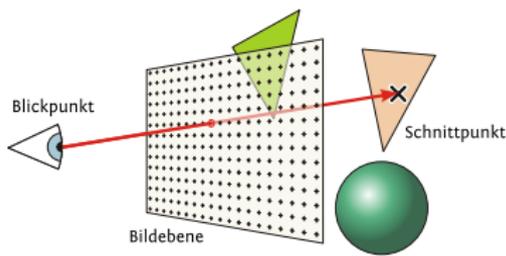
## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

- *Backward-Raytracing*: bestimme, *woher* das Licht kam
- schieße vom Blickpunkt „Sichtstrahlen“ durch jeden Pixel der Bildebene
- bestimme Objekt, das der Sichtstrahl trifft
- berechne Farbe des Lichts, das in Richtung des Betrachters ausgesandt wird durch *lokales Beleuchtungsmodell* (→ *Shading*)



Quelle: <http://de.wikipedia.org/wiki/Raytracing>

# Gliederung

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens
- 3 Schnittpunktbestimmung**
- 4 Farbbestimmung
- 5 Stochastisches/Verteiltes Raytracing
- 6 Laufzeitanalyse, Optimierung

# Strahlaussendung vom Blickpunkt

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

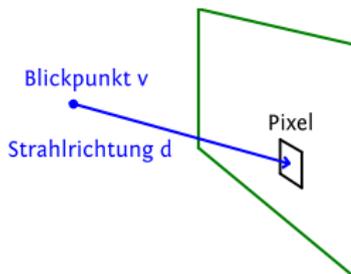
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

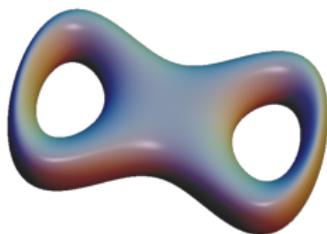
- Welches Objekt wird auf einen bestimmten Bildpixel abgebildet?  $\Leftrightarrow$  Welches Objekt ist vom Blickpunkt aus in Richtung eines bestimmten Pixels sichtbar?
- Strahl (Halbgerade) gegeben durch:  
 $s : \mathbb{R}_+ \rightarrow \mathbb{R}^3, \lambda \mapsto \vec{v} + \lambda \cdot \vec{d}$
- $v$  Blickpunkt,  $d$  Richtung des Strahls:



- berechne für alle Objekte Schnittpunkt mit dem Strahl

# Mögliche Primitivtypen

- Raytracing erlaubt alle Primitivtypen, die
  - Schnittpunktberechnung mit einem Strahl erlauben (z. B. Kugeln, Polygone, implizit gegebene Flächen) und
  - überall eine Oberflächennormale haben
- Vorteil: runde Körper müssen nicht durch Polygone approximiert werden



Quelle: <http://virtualmathmuseum.org/Surface/bretzel2/bretzel2.html>

$$(x^2(1-x^2) - y^2)^2 + \frac{1}{2}z^2 - f \cdot (1 + b \cdot (x^2 + y^2 + z^2)) = 0$$

- Nachteil: für jeden Primitivtyp eigenes Lösungsverfahren zur Schnittpunktbestimmung (oder numerische Lösung)

# Welches Objekt ist sichtbar?

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

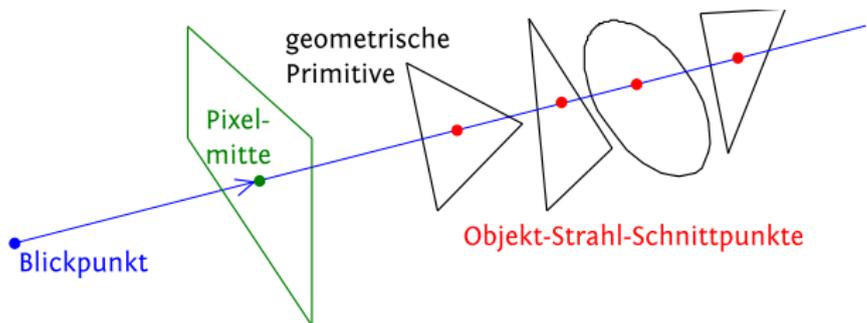
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Schnittpunkte mit Objekt  $O_1, O_2, \dots$  identifiziert durch Strahlparameter  $\lambda_1, \lambda_2, \dots$
- wähle das kleinste  $\lambda_i$  aus; das entsprechende Objekt ist vom Blickpunkt aus sichtbar:



- nenne den entsprechenden Schnittpunkt  $P$  und die Normale in diesem Punkt  $\vec{n}$

# Gliederung

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens
- 3 Schnittpunktbestimmung
- 4 Farbbestimmung**
- 5 Stochastisches/Verteiltes Raytracing
- 6 Laufzeitanalyse, Optimierung

# Bestimmung des ausgesandten Lichts

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Erinnerung: wir wollen die Farbe des zum Blickpunkt ausgesandten Lichts bestimmen
- einfachste Methode: verwende Texturfarbe



Quelle: <http://de.wikipedia.org/wiki/Raytracing>

- Verdeckung wird korrekt berechnet (*Raycasting*)
- qualitativ nicht ausreichend

# Arten des einfallenden Lichts

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

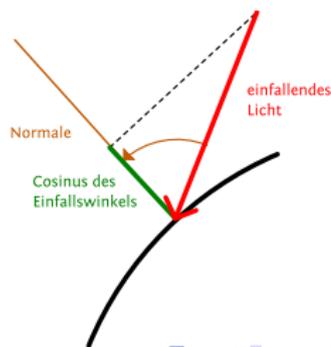
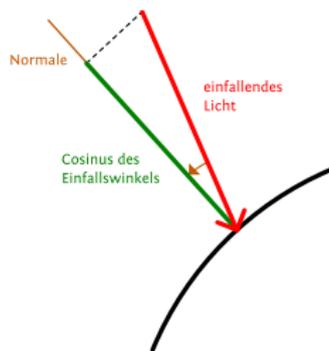
Welches Licht kann von  $P$  in Richtung des Blickpunktes ausgesandt werden?

- gerichtet reflektiertes Licht (z. B. Spiegel)
- gerichtet transmittiertes Licht (z. B. Glaskugel)
- diffus reflektiertes Licht (alle matten Oberflächen)
- diffus transmittiertes Licht (z. B. Milchglas)
- Sonderfall der diffusen Reflexion: von einer platzierten Lichtquelle diffus reflektiertes Licht

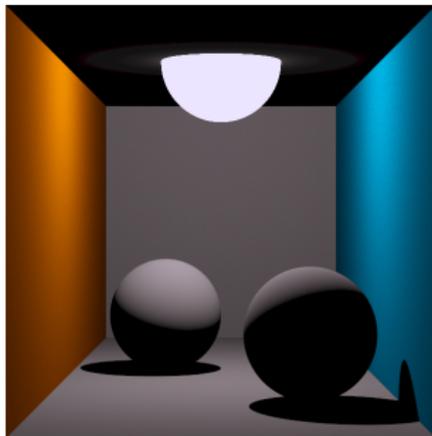
# Direkter Lichteinfall

Von welcher Lichtquelle kommt Licht in  $P$  an?

- sende *Schattenstrahl* von  $P$  zu jeder Lichtquelle
- prüfe, ob es ein Objekt gibt, das sich zwischen  $P$  und der Lichtquelle  $L$  befindet
  - ja:  $P$  liegt im Schatten bezgl.  $L$
  - nein:  $P$  bekommt direktes Licht von  $L$
- Intensität des ausgesandten Lichts ist proportional zum Cosinus des Einfallswinkels des Lichts:



Beispiel:



Quelle: <http://de.wikipedia.org/wiki/Raytracing>

- dreidimensionale Gestalt erkennbar
- hart berandete Schatten
- Einfluss des Einfallswinkels erkennbar

# Gerichtet reflektiertes Licht

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

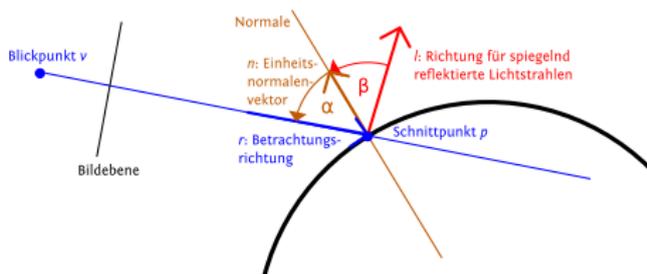
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- bei spiegelnden Oberflächen kann das ausgesandte Licht aus genau einer Richtung kommen
- Einfallswinkel = Ausfallswinkel  $\rightarrow \vec{l} = \vec{r} - 2 \langle \vec{n}, \vec{r} \rangle \cdot \vec{n}$



- Aufgabe: bestimme das Licht, das aus der berechneten Richtung  $\vec{l}$  kommt

# Gerichtet transmittiertes Licht

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

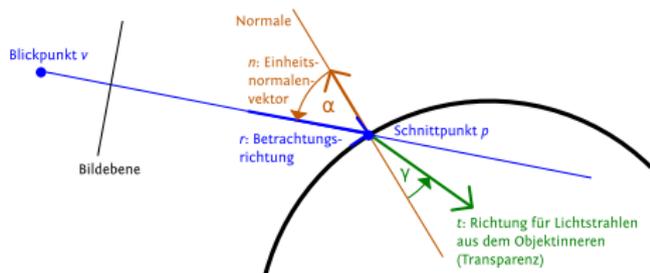
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- bei durchsichtigen Objekten kann das ausgesandte Licht auch nur aus genau einer Richtung kommen (*Transparenz*)
- *Lichtbrechung*: Winkel abhängig vom *Brechungsindex* der beiden Materialien:  $\frac{\sin \alpha}{\sin \gamma} = \frac{\eta_r}{\eta_t}$  (*Snells Gesetz*); unter Umständen *Totalreflexion*



- $\vec{t} = \frac{\eta_r}{\eta_t} \vec{r} + \left( \frac{\eta_r}{\eta_t} \cos \alpha - \sqrt{1 + \left(\frac{\eta_r}{\eta_t}\right)^2 \cdot (\cos \alpha - 1)} \right) \vec{n}$
- Aufgabe: bestimme das Licht, das aus der berechneten Richtung  $\vec{t}$  kommt

# Rekursives Raytracing

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

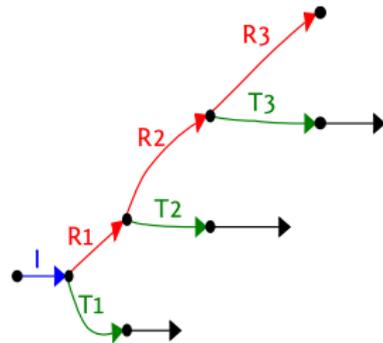
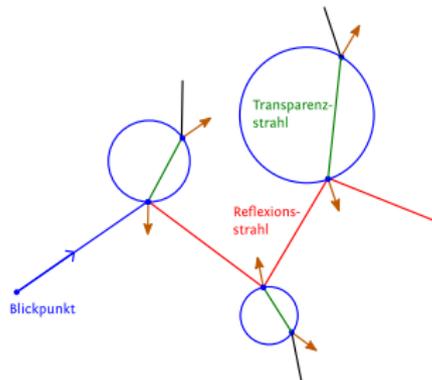
Schnittpunkte

Farben

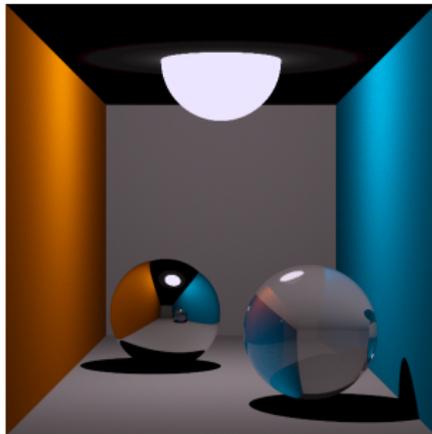
Verteiltes  
Raytracing

Optimierung

- benutze rekursiven Ansatz für gerichtete Reflexion und Transmission
- schicke von  $P$  aus Strahlen in berechnete Richtungen, bestimme ersten Schnittpunkt, bestimme Farbe, ...
- Rekursionsanker: Verlassen der Szene oder festgelegte Tiefe des *Strahlenbaums*



Beispiel:



Quelle: <http://de.wikipedia.org/wiki/Raytracing>

- Reflexion in hinterer Kugel
- Reflexion und Transmission in vorderer Kugel
- Schatten und Wände wie vorher

# Diffus reflektiertes / transmittiertes Licht

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- diffuse Reflexion: einfallendes Licht wird durch raue Oberfläche in alle Richtungen ausgesandt
- diffuse Transmission: Licht wird innerhalb des Objekts durch kleine Partikel ständig umgelenkt (*Transluzenz*)
- Problem: keine eindeutige Richtung, in die weiter verfolgt werden kann
- mögliche Lösung: stochastisches Raytracing, Path Tracing, Photon Mapping
- wichtig: Information über eintreffendes Licht muss durch ein *lokales Beleuchtungsmodell* verarbeitet werden

# Gliederung

## Raytracing

Tobias  
Pfeiffer

## Rendering- Grundlagen

## Prinzip des Raytracings

## Schnittpunkte

## Farben

## Verteiltes Raytracing

## Optimierung

- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens
- 3 Schnittpunktbestimmung
- 4 Farbbestimmung
- 5 Stochastisches/Verteiltes Raytracing**
- 6 Laufzeitanalyse, Optimierung

# Grundprinzip

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

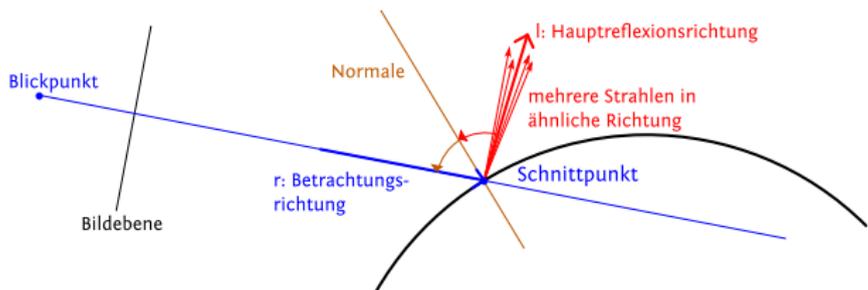
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Idee: schieße in bestimmten Situationen mehrere Strahlen in eine grobe Richtung
- Ziel: realistischere Effekte, Qualitätssteigerung
- nimm Werte nicht aus genau einer Richtung, sondern von mehreren Strahlen aus *ungefähr* dieser Richtung
- entweder nach bestimmtem Muster oder zufällig ausgewählt



# Beispiel: weiche Schatten

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

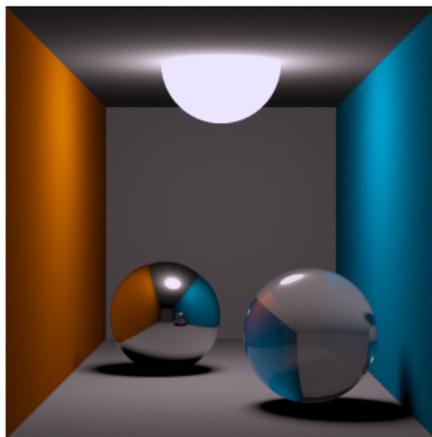
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- modelliere Lichtquelle als Fläche, nicht punktförmig
- schieße mehrere Schattenstrahlen in Richtung der „Lichtfläche“ und mittele Ergebnisse
- Ergebnis: Schatten bekommen weiche Ränder



Quelle: <http://de.wikipedia.org/wiki/Raytracing>

# Beispiel: Tiefenunschärfe

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

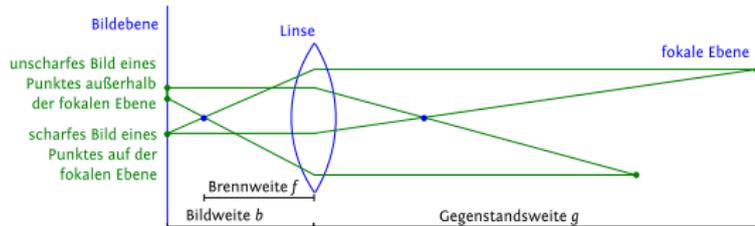
Schnittpunkte

Farben

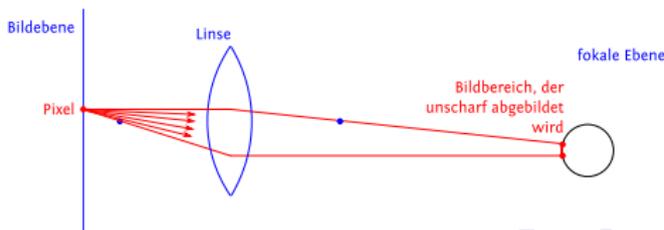
Verteiltes  
Raytracing

Optimierung

- Problem: beim Raytracing gibt es keine Tiefenunschärfe
- tatsächlich sehen wir jedoch Objekte unscharf, die nicht im Fokus der Linse sind:  $\frac{1}{f} = \frac{1}{b} + \frac{1}{g}$



- Lösung: mehrere Strahlen über die Linsenoberfläche verteilen



# Beispiel: Tiefenunschärfe

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung



Quelle: <http://www-viz.tamu.edu/students/flannery/raytracing.html>

- Gegenstände vor und hinter der fokalen Ebene sind verschwommen

# Beispiel: Subsampling

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

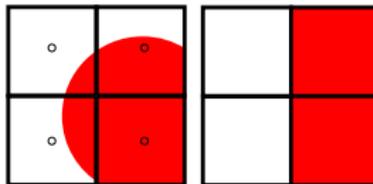
Schnittpunkte

Farben

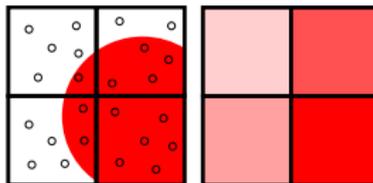
Verteiltes  
Raytracing

Optimierung

- unter Umständen Treppcheneffekte (*Aliasing*), wenn benachbarte Pixel unterschiedliche Objekte treffen



- Idee: mehrere Strahlen pro Pixel (ggf. adaptiv), zufällig verteilt oder nach gewissem Muster, Ergebnisse mitteln



# Gliederung

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- 1 Grundlagen des Renderings
- 2 Prinzip des Raytracing-Verfahrens
- 3 Schnittpunktbestimmung
- 4 Farbbestimmung
- 5 Stochastisches/Verteiltes Raytracing
- 6 Laufzeitanalyse, Optimierung

- hochqualitative Bilder möglich, aber hoher Aufwand
- jeder Strahl muss gegen alle  $n$  Objekte getestet werden
- pro Pixel ein Primärstrahl und  $O(2^d)$  Sekundärstrahlen ( $d$  maximale Tiefe des Strahlenbaums)
- Gesamtlaufzeit  $O(x \cdot y \cdot 2^d \cdot n)$
- Beispiel:  $1024 \times 768$  Pixel,  $d = 3$ , 1000 Objekte:  
7 077 888 000 Schnittpunkttests
- Optimierung vor allem bei Reduzierung der Anzahl der Strahl-Objekt-Schnittpunktberechnungen

# Objekthierarchien

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

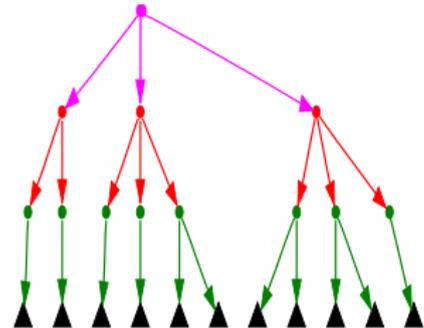
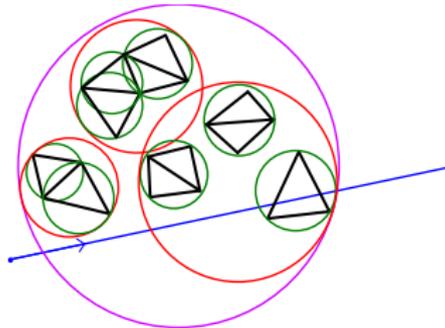
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Idee: verschachtele die Objekte in Container-Objekte, die einfacher auf Schnitt zu testen sind
- konstruiere Hierarchie aus Containern:



- bei der Schnittpunktbestimmung nur in Unterbaum absteigen, wenn es einen Schnitt mit Wurzelobjekt gibt

# Objekthierarchien

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Wahl der Container-Objekte: Kompromiss zwischen möglichst guter Approximation und möglichst schnellem Schnittpunkttest
- weitere Optimierung: behandle Baum mit DFS und speichere  $\lambda$ -Intervall mit Strahl
- verkleinere oberes Ende des Intervalls, wenn Schnitt gefunden  $\rightarrow$  Objekte *hinter* bereits gefundenen Schnittpunkten werden übersprungen

# Raumaufteilung

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- alternative Idee: partitioniere den Raum und speichere zu jedem Bereich die dort vorhandenen Objekte
- bei der Schnittpunktbestimmung nur gegen Objekte testen, die sich im Bereich um den Strahl herum befinden
- Beispiel: Binary Space Partitioning,  $k$ - $d$ -Bäume, Octrees, uniforme Würfelgitter

# Binary Space Partitioning

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

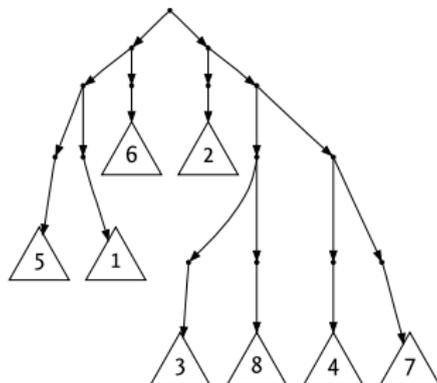
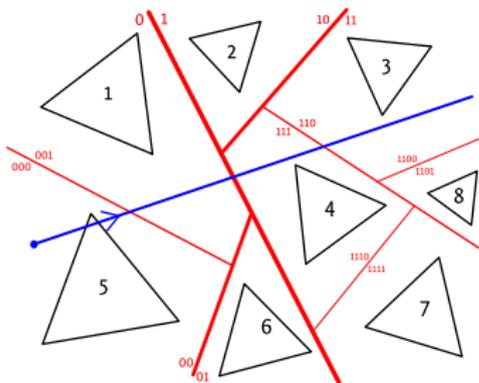
Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- Idee: teile den Raum in Hälften, bis nur noch kleine Anzahl Objekte in Bereich übrig:



- steige zur Schnittpunktbestimmung immer zuerst in den Unterbaum ab, der näher am Blickpunkt liegt
- mit Erweiterungen kann man die Anzahl der Schnittpunkttests auf  $O(\log n)$  bringen

# Zusammenfassung, Ausblick

## Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung

- für Photorealismus (insbes. diffuse Reflexionen) benötigt man weitergehende Techniken (z. B. Photon Mapping)
- noch Spielraum und Notwendigkeit für Geschwindigkeitsoptimierungen, im Hinblick auf Realtime-Raytracing
- vertiefende Informationen:
  - Andrew S. Glassner: An Introduction to Ray Tracing
  - Heiko Duin, Günter Symanzik, Ute Claussen: Beleuchtungsalgorithmen in der Computergrafik
  - <http://www.povray.org>: POV-Ray, ein freier Raytracer

# Vielen Dank!

Raytracing

Tobias  
Pfeiffer

Rendering-  
Grundlagen

Prinzip des  
Raytracings

Schnittpunkte

Farben

Verteiltes  
Raytracing

Optimierung



Quelle: <http://hof.povray.org/glasses.html>

