

4. CHEMNITZER LINUX-TAG

Technische Universität Chemnitz
Fakultät für Informatik

Progressive Raytracing

Matthias Kupfer

`maku@informatik.tu-chemnitz.de`

Chemnitz, den 10. März 2002

1 Einleitung

In vielen Bereichen entsteht heutzutage der Bedarf an fotorealistischen Computergrafiken.

Raytracing ist eine der ältesten Technik und aufgrund ihres einfachen Grundprinzips und des globalen Beleuchtungsmodells wohl die am weitesten verbreitete Methode.

Die Idee des Raytracing ist es, den Verlauf der Lichtstrahlen in einer Szene zu verfolgen.

Da es nicht möglich ist alle Strahlen zu verfolgen, die von einer Lichtquelle ausgehen, wird entgegen der physikalischen Strahlrichtung vom Betrachter ausgehend gestartet. Die meisten Lichtstrahlen einer Lichtquelle erreichen den Betrachter ohnehin nicht und tragen lediglich zur diffusen Beleuchtung bei.

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen
- Beleuchtungsberechnung erfolgt *während* der Bildsynthese

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen
- Beleuchtungsberechnung erfolgt *während* der Bildsynthese
- Darstellung erfolgt (sub)pixelgenau

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen
- Beleuchtungsberechnung erfolgt *während* der Bildsynthese
- Darstellung erfolgt (sub)pixelgenau
- beliebiges Objektmodell möglich
(lediglich Schnittpunkt von Strahl und Objekt, sowie die Normale am Schnittpunkt muß berechenbar sein)

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen
- Beleuchtungsberechnung erfolgt *während* der Bildsynthese
- Darstellung erfolgt (sub)pixelgenau
- beliebiges Objektmodell möglich
(lediglich Schnittpunkt von Strahl und Objekt, sowie die Normale am Schnittpunkt muß berechenbar sein)
- Aufwand hängt von der Anzahl der Bildpunkte (Auflösung) und der Anzahl der Objekte ab

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen
- Beleuchtungsberechnung erfolgt *während* der Bildsynthese
- Darstellung erfolgt (sub)pixelgenau
- beliebiges Objektmodell möglich
(lediglich Schnittpunkt von Strahl und Objekt, sowie die Normale am Schnittpunkt muß berechenbar sein)
- Aufwand hängt von der Anzahl der Bildpunkte (Auflösung) und der Anzahl der Objekte ab
- Speicherplatzbedarf hängt von der Objektanzahl ab

2 Eigenschaften des Raytracing

- Bilderzeugung basiert auf *strahlenoptischen* Grundlagen
- Beleuchtungsberechnung erfolgt *während* der Bildsynthese
- Darstellung erfolgt (sub)pixelgenau
- beliebiges Objektmodell möglich
(lediglich Schnittpunkt von Strahl und Objekt, sowie die Normale am Schnittpunkt muß berechenbar sein)
- Aufwand hängt von der Anzahl der Bildpunkte (Auflösung) und der Anzahl der Objekte ab
- Speicherplatzbedarf hängt von der Objektanzahl ab
- Änderungen des Standortes, der Objekte und der Beleuchtungssituation erfordern komplettes Neuberechnen

3 Grundalgorithmus

Einlesen der Objekte

für $y = 1$ bis Bildhöhe

 für $x = 1$ bis Bildbreite

 berechne Strahlstartpunkt

 berechne Strahlrichtung (x,y)

 Farbwert = Aufruf Strahlverfolgung (Strahl, Tiefe)

 schreibe Farbwert in Datei

Ende

3 Grundalgorithmus

Einlesen der Objekte

für $y = 1$ bis Bildhöhe

 für $x = 1$ bis Bildbreite

 berechne Strahlstartpunkt

 berechne Strahlrichtung (x,y)

 Farbwert = Aufruf Strahlverfolgung (Strahl, Tiefe)

 schreibe Farbwert in Datei

Ende

Strahlverfolgung (Strahl, Tiefe)

wenn Tiefe > 0 dann

 für $i = 1$ bis Objektanzahl

 wenn Schnittpunkt zwischen Strahl und Objekt(i) dann

 berechne lokalen Lichteinfluß

```
wenn Objekt(i) transparent dann
    erstelle Transmissionsstrahl
    Transmissionsfarbe =
        Strahlverfolgung (Transmissionsstrahl, Tiefe-1)
sonst
    Transmissionsfarbe = schwarz
wenn Objekt(i) verspiegelt dann
    erstelle Reflexionsstrahl
    Reflexionsfarbe =
        Strahlverfolgung (Transmissionsstrahl, Tiefe-1)
sonst
    Reflexionsfarbe = schwarz
fasse die Farben zusammen
sonst
Hintergrundfarbe
```

Ende

4 Vor- und Nachteile Raytracing

- Vorteile
 - mathematisch exakte Objektdarstellung
 - einfacher Grundalgorithmus
 - (sub)pixelgenaue Darstellung
 - sehr realistische Darstellung von Brechungs- und Spiegelungseffekten
 - durch die unabhängige Berechnung der Pixel leicht parallelisierbar

- Nachteile
 - rekursiver Algorithmus
 - dadurch bedingt keine darstellbaren Zwischenbilder
 - meist sehr lange Berechnungsdauer
 - Optimierungsverfahren schlecht für die Parallelisierung geeignet
 - keine zufriedenstellende Simulation diffuser Beleuchtung
 - schwierige Darstellung von bestimmten Naturerscheinungen (z.B. Wolken, Flammen, Nebel usw.)

5 Auswege und alternative Ansätze

Das Beheben einiger Nachteile soll durch sogenanntes Progressive Raytracing erfolgen.

Ausgangsbasis war objektorientierte Implementation eines rekursiven brute-force Raytracers für Dreiecke und Kugeln.

Dieser wird erweitert um:

5 Auswege und alternative Ansätze

Das Beheben einiger Nachteile soll durch sogenanntes Progressive Raytracing erfolgen.

Ausgangsbasis war objektorientierte Implementation eines rekursiven brute-force Raytracers für Dreiecke und Kugeln.

Dieser wird erweitert um:

Strahlwarteschlange In diese Warteschlange werden alle Strahlen eingetragen und nacheinander bearbeitet. Damit wird der rekursive Tiefensuchalgorithmus in einen iterativen Breitensuchalgorithmus geändert.

Durch eine priorisierte Warteschlange kann die Abarbeitungsreihenfolge beliebig geändert werden.

Lichtquellenstrahlen Auch Strahlen, die lediglich prüfen, ob ein Punkt sich im Schatten bezüglich einer bestimmten Lichtquelle befindet, werden als eigene Strahlen betrachtet und verfolgt. Diese haben im Mittel eine kürzere Berechnungsdauer, da das erste getroffene Objekt sofort zum Ende der Berechnung führt (sich also Schatten ergibt).

Lichtquellenstrahlen Auch Strahlen, die lediglich prüfen, ob ein Punkt sich im Schatten bezüglich einer bestimmten Lichtquelle befindet, werden als eigene Strahlen betrachtet und verfolgt. Diese haben im Mittel eine kürzere Berechnungsdauer, da das erste getroffene Objekt sofort zum Ende der Berechnung führt (sich also Schatten ergibt).

Bildarstellungsmodi Folgende Darstellungsvarianten sind möglich

Lichtquellenstrahlen Auch Strahlen, die lediglich prüfen, ob ein Punkt sich im Schatten bezüglich einer bestimmten Lichtquelle befindet, werden als eigene Strahlen betrachtet und verfolgt. Diese haben im Mittel eine kürzere Berechnungsdauer, da das erste getroffene Objekt sofort zum Ende der Berechnung führt (sich also Schatten ergibt).

Bildarstellungsmodi Folgende Darstellungsvarianten sind möglich

Sofortbild nach jedem Strahl wird das korrespondierende Pixel sofort aktualisiert. Bilderzeugung kann zu jedem Zeitpunkt abgebrochen werden. Es werden alle Ergebnisse dargestellt.

Lichtquellenstrahlen Auch Strahlen, die lediglich prüfen, ob ein Punkt sich im Schatten bezüglich einer bestimmten Lichtquelle befindet, werden als eigene Strahlen betrachtet und verfolgt. Diese haben im Mittel eine kürzere Berechnungsdauer, da das erste getroffene Objekt sofort zum Ende der Berechnung führt (sich also Schatten ergibt).

Bildarstellungsmodi Folgende Darstellungsvarianten sind möglich

Sofortbild nach jedem Strahl wird das korrespondierende Pixel sofort aktualisiert. Bilderzeugung kann zu jedem Zeitpunkt abgebrochen werden. Es werden alle Ergebnisse dargestellt.

Endbild erst nach dem die gesamte Warteschlange gebildet wurde, wird das Bild generiert. Die insgesamt kürzere Rechenzeit wird durch eine Nachlaufzeit nach der eigentlichen Strahlbearbeitung erkauft, da die Farbwerte noch zusammengefaßt werden müssen.

6 Details der iterativen Methode

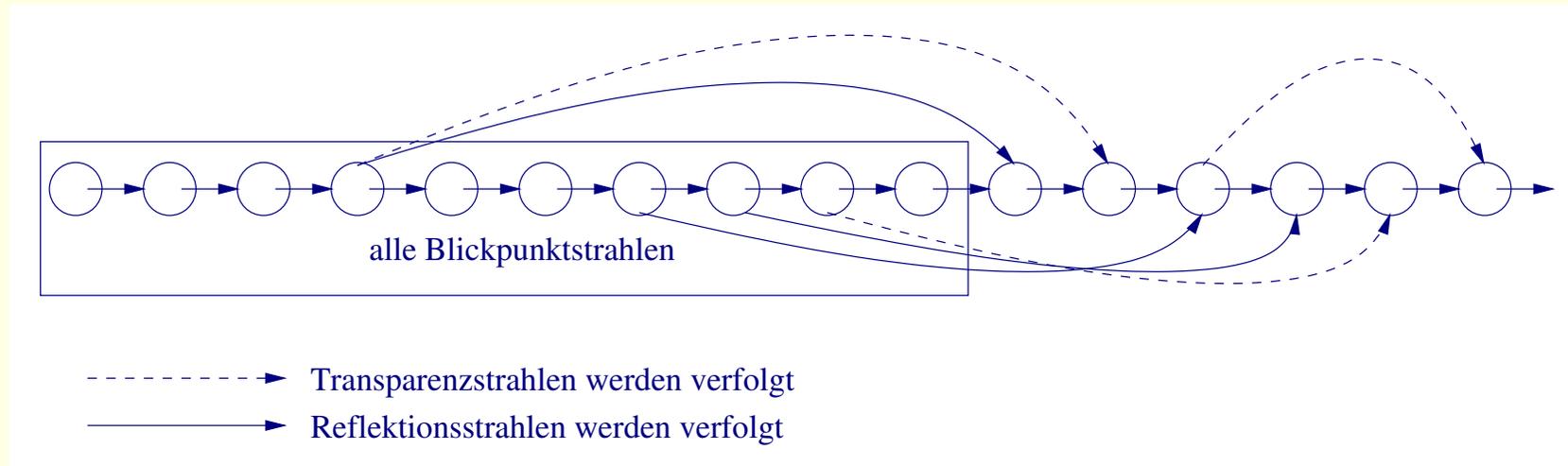


Abbildung 1: Strahlwarteschlange

6 Details der iterativen Methode

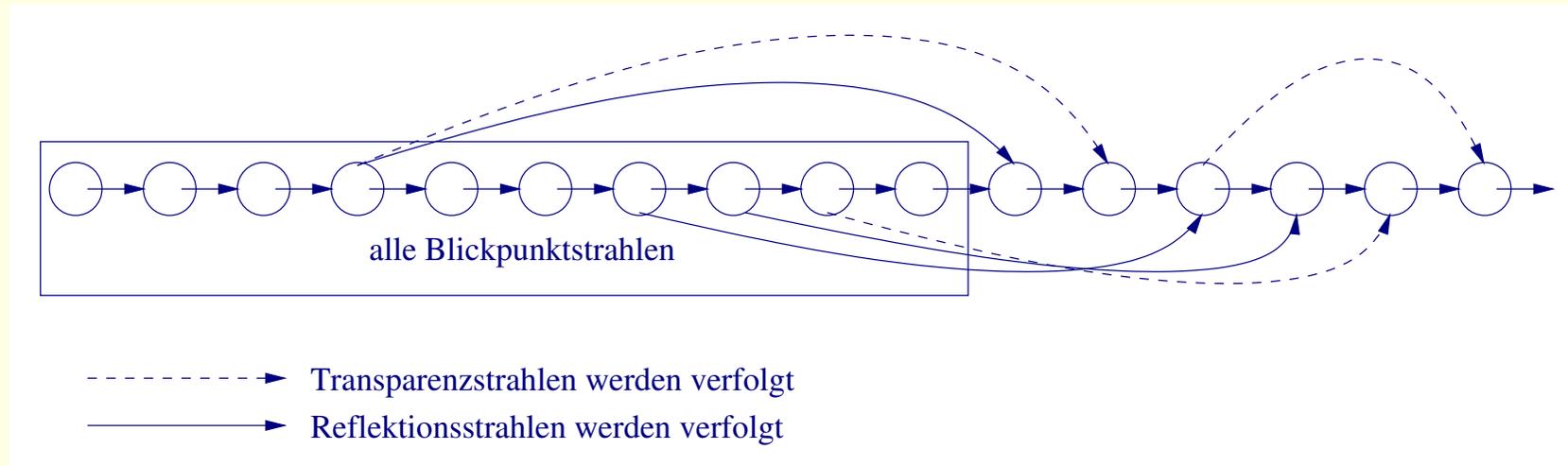


Abbildung 1: Strahlwarteschlange

- kleinste Berechnungseinheit ist nicht Pixel, sondern Strahl

6 Details der iterativen Methode

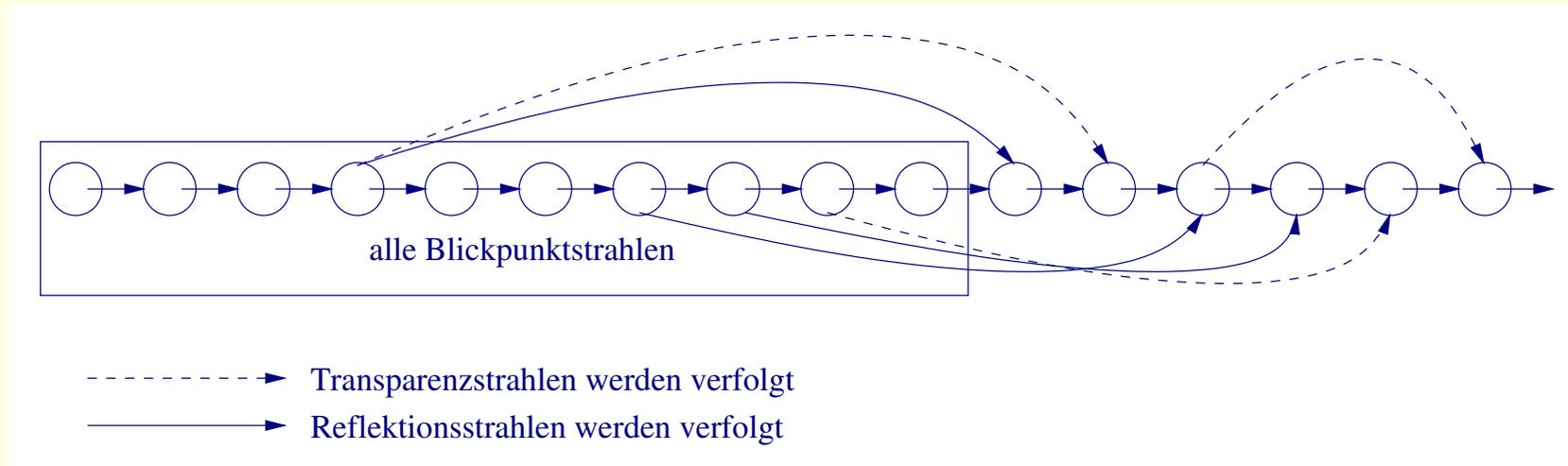


Abbildung 1: Strahlwarteschlange

- kleinste Berechnungseinheit ist nicht Pixel, sondern Strahl
- durch Breitensuche ist Vorschaubild nach jeder berechneten Tiefe vollständig darstellbar

- Änderungen in der Szene können eingegrenzt werden und z.T. in Echtzeit berechnet werden.

- Änderungen in der Szene können eingegrenzt werden und z.T. in Echtzeit berechnet werden.
- Rechenzeit ist zu Lasten der Qualität begrenzbar

- Änderungen in der Szene können eingegrenzt werden und z.T. in Echtzeit berechnet werden.
- Rechenzeit ist zu Lasten der Qualität begrenzbare
- eine einfache Erweiterung zu einem einphasigen Algorithmus für Raytracing und Radiosity (Strahlungsfeldverfahren) ist möglich

Beispiele für den Ablauf der Berechnung:

- Entstehung (sofortige Lichtstrahlberechnung)
- Entstehung (verzögerte Lichtstrahlberechnung)

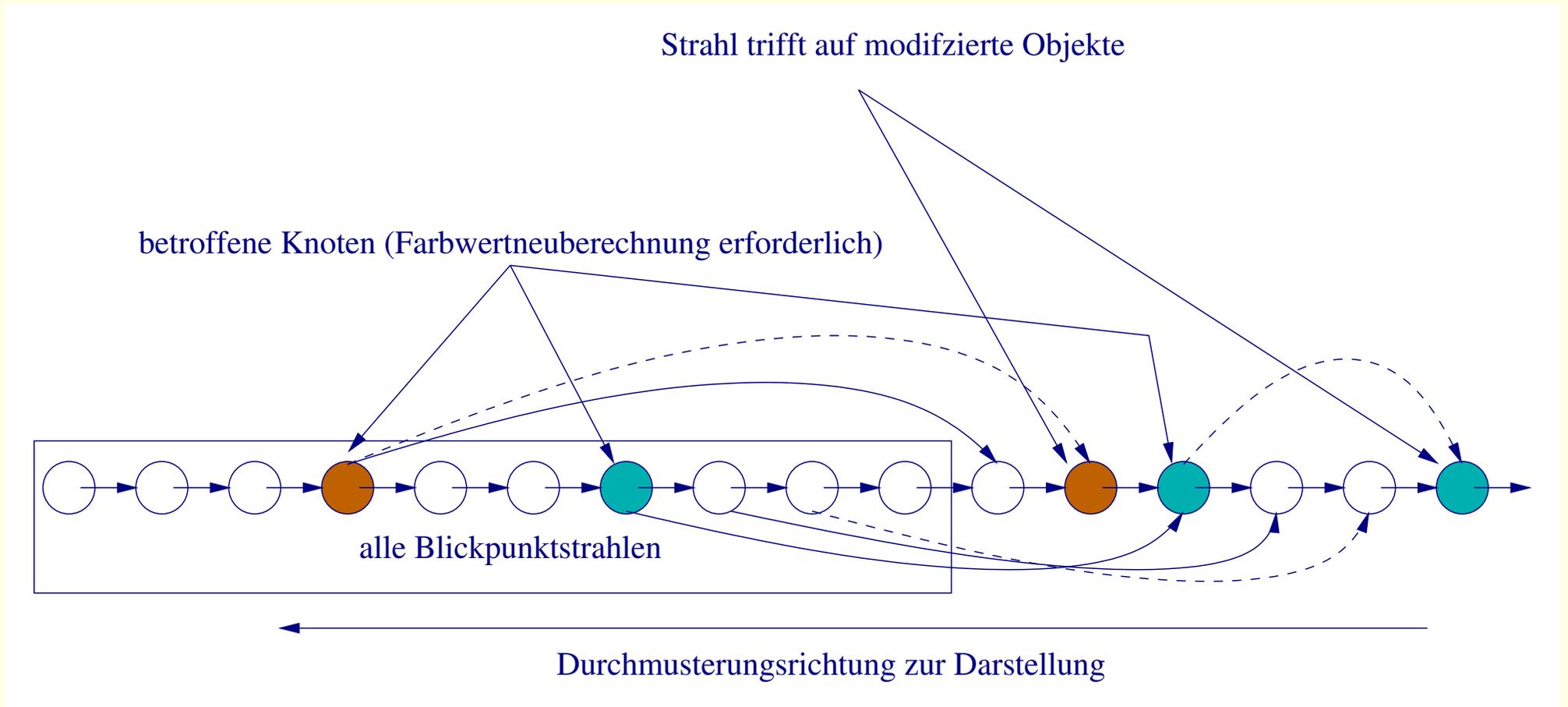


Abbildung 2: Bearbeitung der Änderungen

7 Beschränkungen und Grenzen

Nachteil des iterativen Verfahren ist der immense Speicherplatzbedarf S zur Laufzeit, der bisher aus

$$S = O + n(R + L) \quad (1)$$

ergab. Nun steigt dieser auf

$$S = O + [(2 + m)^n XY \max(R, L)] \quad (2)$$

an. Dabei entsprechen

O den Objektdaten, n der Maximaltiefe der Strahlverfolgung,

R dem Speicherbedarf der Strahlverfolgungsfunktion,

L den Daten für die lokale Beleuchtung,

m der Anzahl der Lichtquellen und X, Y der Bildgröße

8 Beleuchtungsmodell

Wesentlichen Anteil für den Realismus hat das verwendete Beleuchtungsmodell. Es wurde hier ein relativ einfaches nach Phong implementiert.

$$\begin{aligned} L_{gesamt} = & k_{ambient} \cdot I_{basis} + \\ & \sum_{i=1}^m (f(r) \cdot I_{Lq_i} \cdot (k_{diffus} \cdot (\mathbf{N} \cdot \mathbf{L}_i) + k_{specular} \cdot (\mathbf{L}_i \cdot \mathbf{R})^{highlight})) + \\ & k_{transparent} \cdot I_t + k_{reflexion} \cdot I_r \end{aligned} \quad (3)$$

Um eine gleichmäßige Lastverteilung zu realisieren müssen alle Strahlen einzeln bearbeitet werden. Dies gilt auch für Strahlen, die zur Lichtquelle gehen sollen, und lediglich Abschattungen testen.

Es wird letztlich für jeden Bildpunkt der gesamte Strahlbaum gebildet.

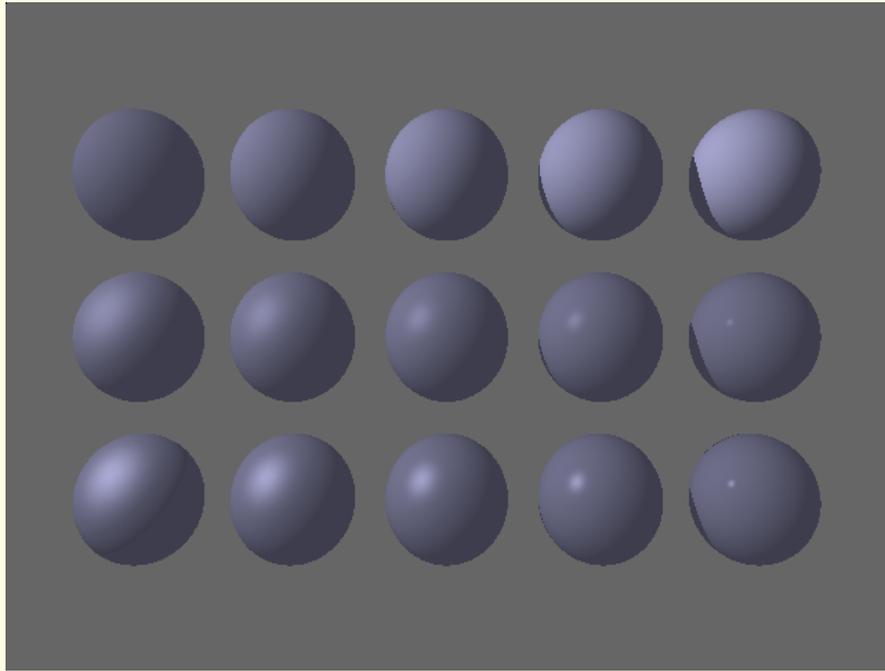


Abbildung 3: versch. Oberflächen

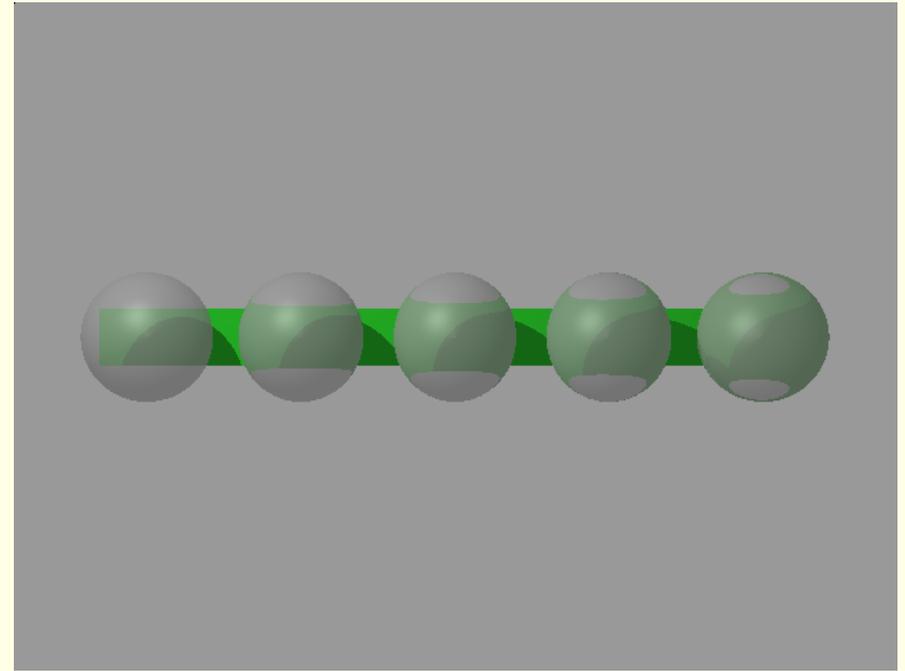


Abbildung 4: Brechungseffekt

9 Konkretes Beispielbild

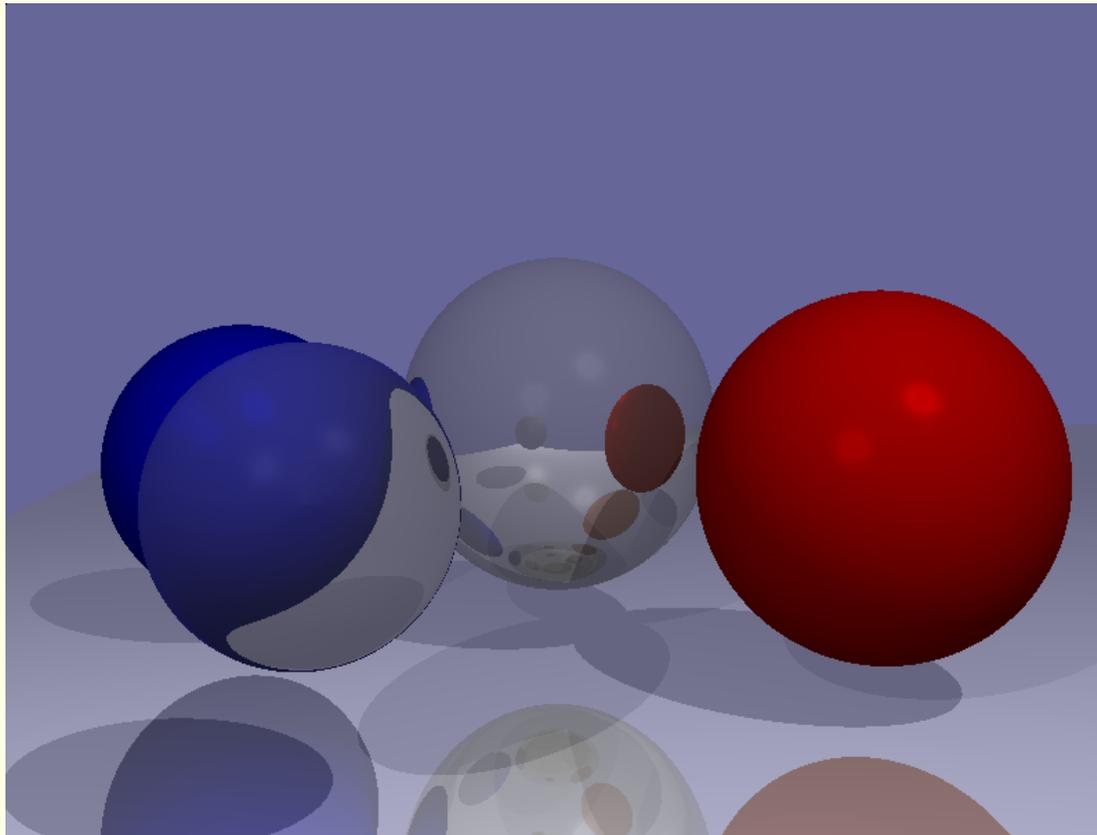


Abbildung 5: Beispielbild

statistische Daten des Bildes:

Auflösung: 640 x 480 Bildpunkte

Strahltyp	Anzahl
Blickpunktstrahlen	307200
Reflexionsstrahlen	154426
Transmissionsstrahlen	56998
Lichtquellenstrahlen	593776
Summe	1112400

Bei einer Größe von 292 Bytes je Warteschlangenelement entspricht dies einem Gesamtbedarf von ca. 310 MByte.

Das Datenvolumen verteilt sich allerdings bei Parallelisierung gleichmäßig auf zu rechnende Knoten.

Optimierungen sind durch Look-Up-Tabellen möglich.

10 Erweiterungen (Teil 1)

- Prioritäten erlauben die gesteuerte Berechnung von Bildern

10 Erweiterungen (Teil 1)

- Prioritäten erlauben die gesteuerte Berechnung von Bildern
- Tiefen können geändert werden (während und sogar nach der Berechnung)

10 Erweiterungen (Teil 1)

- Prioritäten erlauben die gesteuerte Berechnung von Bildern
- Tiefen können geändert werden (während und sogar nach der Berechnung)
- Objekteigenschaften können geändert und sofort dargestellt werden

10 Erweiterungen (Teil 1)

- Prioritäten erlauben die gesteuerte Berechnung von Bildern
- Tiefen können geändert werden (während und sogar nach der Berechnung)
- Objekteigenschaften können geändert und sofort dargestellt werden
- Durch Zusammenfassen von Strahlen ist der Speicherplatz reduzierbar

11 Erweiterungen (Teil 2)

- modifizierte Optimierungsverfahren

11 Erweiterungen (Teil 2)

- modifizierte Optimierungsverfahren
 - Octree-Strukturen
 - kd-Tree
 - BSP
- Erweiterte Modell für die Behandlung von Anti-Alias-Problemen und speziellen Effekten wie Tiefenunschärfe

11 Erweiterungen (Teil 2)

- modifizierte Optimierungsverfahren
 - Octree-Strukturen
 - kd-Tree
 - BSP
- Erweiterte Modell für die Behandlung von Anti-Alias-Problemen und speziellen Effekten wie Tiefenunschärfe
- Integration eines Strahlungsfeldverfahren

12 Integration des Strahlungsfeldverfahrens

Das Strahlungsfeldverfahren berechnet sehr gut die diffuse Beleuchtung mit Hilfe eines strahlungsphysikalischen Modells.

12 Integration des Strahlungsfeldverfahrens

Das Strahlungsfeldverfahren berechnet sehr gut die diffuse Beleuchtung mit Hilfe eines strahlungsphysikalischen Modells.

Das Grundprinzip leitet sich ursprünglich aus den Hauptsätzen der Thermodynamik ab. Die Energie ist in einem geschlossenen System konstant.

12 Integration des Strahlungsfeldverfahrens

Das Strahlungsfeldverfahren berechnet sehr gut die diffuse Beleuchtung mit Hilfe eines strahlungsphysikalischen Modells.

Das Grundprinzip leitet sich ursprünglich aus den Hauptsätzen der Thermodynamik ab. Die Energie ist in einem geschlossenen System konstant.

Ausgehend von den Lichtquellen (perfekte Lambertsche Strahler) wird also die Energie auf die Flächen der Szene solange verteilt, bis die unverteilte Energiemenge vernachlässigbar klein ist. Dies bedeutet, daß jedes Polygon von (theoretisch) jedem Polygon Energie empfangen und auch an jedes Energie abgeben kann.

Bisher war die Kombination beider Verfahren nur als zweiphasiges Verfahren möglich.

Bisher war die Kombination beider Verfahren nur als zweiphasiges Verfahren möglich.

Zuerst Radiosity, dann Raytracing.

Bisher war die Kombination beider Verfahren nur als zweiphasiges Verfahren möglich.

Zuerst Radiosity, dann Raytracing.

Szenenänderungen oder Verfeinerung des Patches für den Energieaustausch machte eine komplette Neuberechnung gemäß beider Methoden erforderlich.

Dies hat natürlich einen hohen Zeitbedarf für die Berechnung zur Folge.

Bisher war die Kombination beider Verfahren nur als zweiphasiges Verfahren möglich.

Zuerst Radiosity, dann Raytracing.

Szenenänderungen oder Verfeinerung des Patches für den Energieaustausch machte eine komplette Neuberechnung gemäß beider Methoden erforderlich.

Dies hat natürlich einen hohen Zeitbedarf für die Berechnung zur Folge.

Jetzt liegen für beide Algorithmen iterative und progressive Methoden vor.

Daraus leitet sich ab, daß sie völlig parallel und gleichwertig behandelt werden können.

Verfeinerung des Polygonnetzes und damit auch die genauere Berechnung erfordert kein erneutes Raytracing.

Nur die diffusen Komponenten aller in der Strahlwarteschlange befindlichen Elemente müssen in einem Durchlauf aktualisiert werden.

Dadurch wird das Verfahren schneller und die Bildqualität praktisch beliebig verbesserbar.

Verfeinerung des Polygonnetzes und damit auch die genauere Berechnung erfordert kein erneutes Raytracing.

Nur die diffusen Komponenten aller in der Strahlwarteschlange befindlichen Elemente müssen in einem Durchlauf aktualisiert werden.

Dadurch wird das Verfahren schneller und die Bildqualität praktisch beliebig verbesserbar.

13 Warum Linux?

- Die Parallelisierung soll auf dem Chemnitzer Linux Cluster erfolgen.
- Außerdem ist Linux ein sehr leistungsfähiges (und kostenloses) Betriebssystem.

Vielen Dank für Ihre Aufmerksamkeit!

Fragen, Hinweise, Anregungen oder Kritik?