

Projektarbeit Raytracing

Prof. Dr. Helmut Neemann

Programmieren II





- 1 Einführung
- 2 Kugel
- 3 Ebene
- 4 Reflexion
- 5 Beleuchtung
- 6 Aufgabe
- 7 Animationen



- Ein Verfahren aus der Computergrafik zur Berechnung photorealistischer Grafiken.
- löst das Problem der verdeckten Flächen
- kann ermitteln, wo Schatten ist, und wo nicht
- einfache Simulation von Reflexionen und Lichtbrechung
- kann Tiefenschärfe und diffuse Schatten darstellen

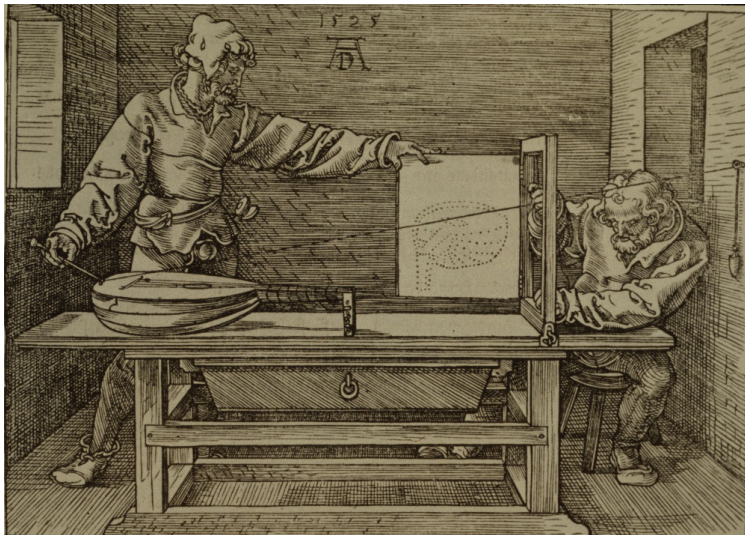


Die Idee

- Bilde ein mathematisches Modell der Szene
- Erzeuge einen virtuellen Bildschirm in der Szene
- Erzeuge einen Sehstrahl vom Augpunkt durch jedes Pixel des virtuellen Bildschirms
- Schneide diesen Strahl mit allen Objekten der Szene
- Der Schnittpunkt der den kleinsten Abstand zum Augpunkt hat, ist sichtbar.
- Berechne einen Schattenstrahl vom Schnittpunkt in Richtung Lichtquelle.
- Trifft der Schattenstrahl ein anderes Objekt, bevor er die Lichtquelle erreicht, liegt der Schnittpunkt im Schatten.
- Berechne die Farbe und Helligkeit des Schnittpunktes anhand der Winkel zwischen Flächennormale und Blickrichtung bzw. Flächennormale und Richtung zur Lichtquelle.

Raytracing

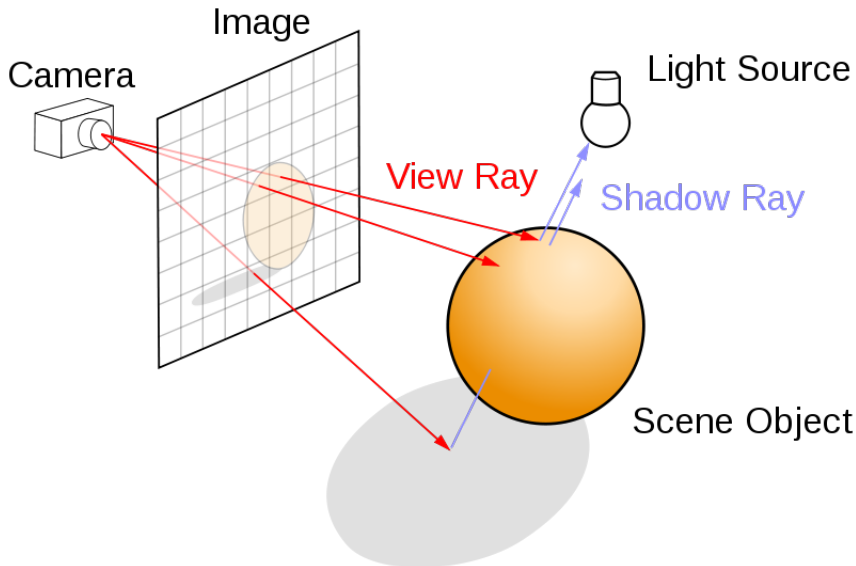
Einführung



Stich aus “Underweysung der Messung mit dem Zirckel und Richtscheyt“,
Albrecht Dürer, 1525

Raytracing

Einführung





Vorteile im Vergleich zu polygonbasierten Rasterverfahren mit z-Puffer.

- photorealistische Darstellung der Scene
- Schatten, Spiegelungen, Lichtbrechung einfach zu berechnen.

Nachteile

- Sehr rechenintensiv
- Bei 800x600 Pixeln Auflösung müssen schon etwa eine halbe Million Sehstrahlen mit der Szene geschnitten werden.
- Zusätzlich sind noch die Schattenstrahlen zu berechnen.
- Bei spiegelnden Oberflächen muss der Sehstrahl rekursiv weiterverfolgt werden.
- Aktuelle Grafikhardware kommt jedoch an eine Echtzeitberechnung heran.
- In Zukunft daher auch in Computerspielen zu erwarten



- ① Einführung
- ② **Kugel**
- ③ Ebene
- ④ Reflexion
- ⑤ Beleuchtung
- ⑥ Aufgabe
- ⑦ Animationen

Die Kugel

Kugel



- Die Schnittberechnung ist die wesentliche Aufgabe beim Raytracing.
- Der Schnitt zwischen einer Geraden und einer Kugel lässt sich besonders einfach berechnen.
- Daher finden Sie oft spiegelnde Kugeln in Raytracing-Demos

Die Gleichung einer Geraden:

$$\vec{r} = \vec{a} + t\vec{b}$$

mit \vec{a} als Ausgangspunkt, und \vec{b} als Blickrichtung durch eines der Pixel

Die Gleichung einer Kugel:

$$|\vec{r} - \vec{k}| = R$$

mit \vec{k} als Kugelmittelpunkt und R als Radius der Kugel

Die Kugel

Kugel



$$\vec{r} = \vec{a} + t\vec{b}$$

$$|\vec{r} - \vec{k}| = R$$

$$(\vec{r} - \vec{k})^2 = R^2$$

$$(\vec{a} + t\vec{b} - \vec{k})^2 = R^2$$

$$((\vec{a} - \vec{k}) + t\vec{b})^2 = R^2$$

$$(\vec{d} + t\vec{b})^2 = R^2 \quad \text{mit} \quad \vec{d} = \vec{a} - \vec{k}$$

$$\vec{d}^2 + t^2\vec{b}^2 + 2t\vec{d}\vec{b} = R^2$$

$$t^2\vec{b}^2 + 2t\vec{d}\vec{b} + \vec{d}^2 - R^2 = 0$$

Quadratische Gleichung in t .



$$t^2 \vec{b}^2 + 2t \vec{d}\vec{b} + \vec{d}^2 - R^2 = 0$$

Quadratische Gleichung in t .

$$\alpha t^2 + \beta t + \gamma = 0$$

$$\begin{aligned} t_{1,2} &= \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \\ &= \frac{-2\vec{d}\vec{b} \pm \sqrt{4(\vec{d}\vec{b})^2 - 4\vec{b}^2(\vec{d}^2 - R^2)}}{2\vec{b}^2} \\ &= \frac{-\vec{d}\vec{b} \pm \sqrt{(\vec{d}\vec{b})^2 - \vec{b}^2(\vec{d}^2 - R^2)}}{\vec{b}^2} \end{aligned}$$

Die Kugel

Kugel



$$t_{1,2} = \frac{-\vec{d}\vec{b} \pm \sqrt{(\vec{d}\vec{b})^2 - \vec{b}^2(\vec{d}^2 - R^2)}}{\vec{b}^2}$$

Annahme: \vec{b} sei normiert! Also $|\vec{b}| = 1$ und $\vec{b}^2 = 1$

$$\begin{aligned} t_{1,2} &= -\vec{d}\vec{b} \pm \sqrt{(\vec{d}\vec{b})^2 - (\vec{d}^2 - R^2)} \\ &= -\vec{d}\vec{b} \pm \sqrt{(\vec{d}\vec{b})^2 + R^2 - \vec{d}^2} \end{aligned}$$

- Da \vec{b} normiert ist, ist $t_{1,2}$ der Abstand zwischen Augpunkt \vec{a} und Kugeloberfläche.
- Ist t negativ, liegt der Schnittpunkt hinter dem Betrachter und kann ignoriert werden.
- Ist der Term unter der Wurzel negativ, trifft der Sehstrahl die Kugel nicht!



- ① Einführung
- ② Kugel
- ③ Ebene**
- ④ Reflexion
- ⑤ Beleuchtung
- ⑥ Aufgabe
- ⑦ Animationen

Schnitt mit einer Ebene

Ebene



Eine Ebene im Raum ist definiert durch einen Punkt \vec{E} auf der Ebene und die Normale \vec{N} .

Die Normale \vec{N} ist ein Vektor der Länge 1, welcher senkrecht auf der Ebene steht.

Zur Erinnerung: Das Skalarprodukt zweier Vektoren ist Null, wenn diese senkrecht aufeinander stehen!

Liegt ein Punkt \vec{r} in der Ebene, dann gilt

$$\begin{aligned}(\vec{r} - \vec{E})\vec{N} &= 0 \\(\vec{a} + t\vec{b} - \vec{E})\vec{N} &= 0 \\(\vec{a} - \vec{E})\vec{N} + t\vec{b}\vec{N} &= 0 \\t\vec{b}\vec{N} &= (\vec{E} - \vec{a})\vec{N} \\t &= \frac{(\vec{E} - \vec{a})\vec{N}}{\vec{b}\vec{N}}\end{aligned}$$



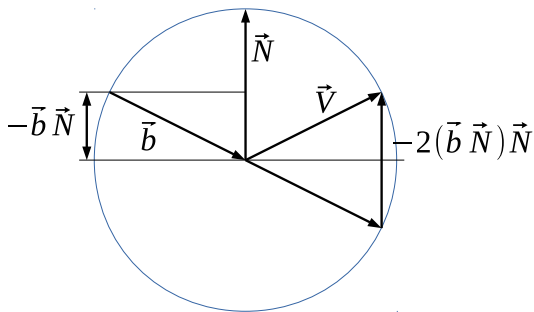
- 1 Einführung
- 2 Kugel
- 3 Ebene
- 4 Reflexion**
- 5 Beleuchtung
- 6 Aufgabe
- 7 Animationen

Reflexion an einer Ebene

Reflexion



Der Strahl in Richtung \vec{b} trifft auf eine Fläche mit der Normalen \vec{N} .
In welche Richtung \vec{V} wird er reflektiert?



Der Richtungsvektor \vec{V} des reflektierten Strahls berechnet sich wie folgt:

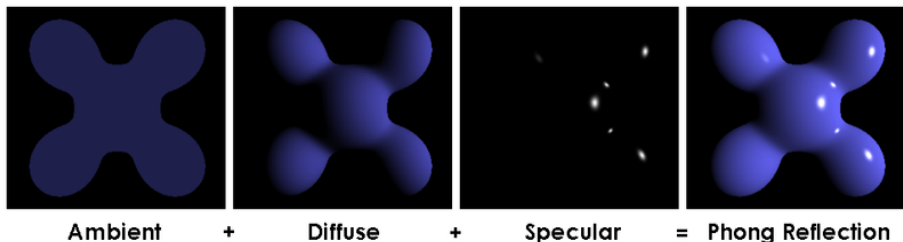
$$\vec{V} = \vec{b} - 2(\vec{b}\vec{N})\vec{N}$$



- ① Einführung
- ② Kugel
- ③ Ebene
- ④ Reflexion
- ⑤ **Beleuchtung**
- ⑥ Aufgabe
- ⑦ Animationen

Das Phong-Beleuchtungsmodell

Beleuchtung



Im Phong-Modell setzt sich die Helligkeit eines Pixels aus drei Komponenten zusammen:

- ① Es gibt einen konstanten ambienten Anteil I_A . Er ist eine Materialkonstante.
- ② Der diffuse Anteil I_D modelliert einen Lambert-Strahler: Die Helligkeit hängt nicht vom Blickwinkel ab, sondern nur vom Winkel φ zwischen Lichtquelle und Flächennormale (z.B. mattes Papier).
- ③ Bei der Komponente I_S hängt die Helligkeit vom Winkel Θ zwischen Betrachter und dem reflektierten Lichtstrahl ab (z.B. Plastik).

Das Phong-Beleuchtungsmodell

Beleuchtung



Die Helligkeit ergibt sich dann zu:

$$I = I_A + I_D + I_S$$

wobei

$$I_A = I_{\text{ln}} k_A$$

$$\begin{aligned} I_D &= I_{\text{ln}} k_D \cos \varphi \\ &= I_{\text{ln}} k_D \vec{L} \vec{N} \end{aligned}$$

$$\begin{aligned} I_S &= I_{\text{ln}} k_S (\cos \Theta)^n \\ &= I_{\text{ln}} k_S (-\vec{b} \vec{V})^n \end{aligned}$$

Mit \vec{L} als Einheitsvektor in Richtung der Lichtquelle und \vec{V} als Einheitsvektor in die Richtung in die der Lichtstrahl der Lichtquelle reflektiert wird.

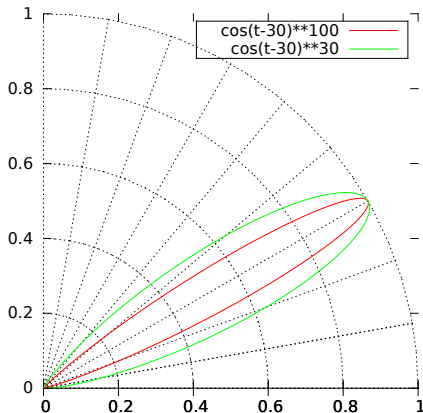
Wobei k_A , k_D , k_S und n Materialkonstanten sind.

Das Phong-Beleuchtungsmodell

Beleuchtung



Bei rauen Oberflächen gilt $n \approx 30$, ein perfekter Spiegel ergibt sich für $n = \infty$:



Hier die Reflexionskeule in Polarkoordinatendarstellung.



- ① Einführung
- ② Kugel
- ③ Ebene
- ④ Reflexion
- ⑤ Beleuchtung
- ⑥ **Aufgabe**
- ⑦ Animationen

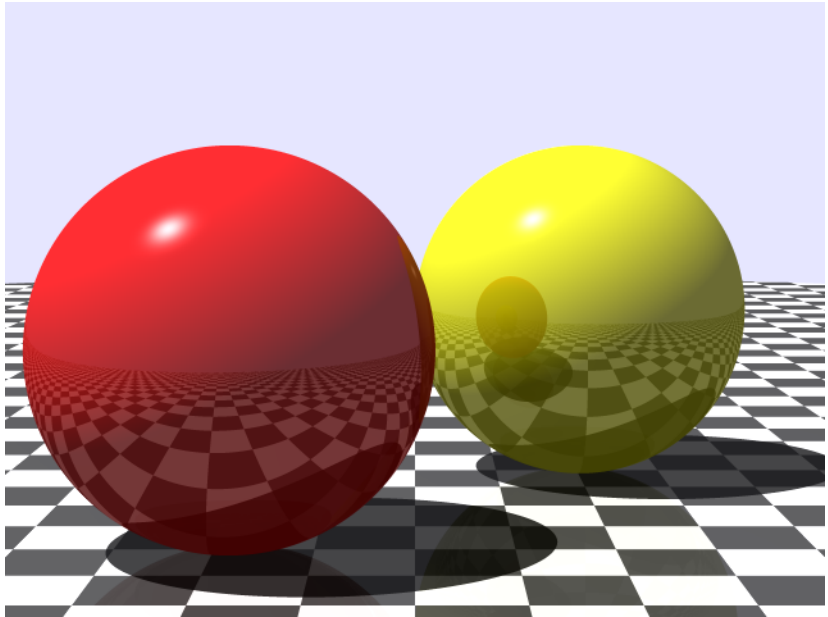


Es soll ein minimaler Raytracing-Algorithmus implementiert werden. Die Anwendung soll in der Lage sein, eine beispielhafte Szene zu berechnen, welche mindestens folgendes beinhaltet:

- ① Zwei Kugeln unterschiedlicher Farbe.
- ② Die Kugeln sollen auf einer Ebene liegen.
- ③ Es soll eine punktförmige Lichtquelle verwendet werden, die außerhalb der sichtbaren Szene platziert ist.
- ④ Die Kugeln sollen einen Schatten werfen.
- ⑤ Es soll das Phong-Beleuchtungsmodell für alle Oberflächen verwendet werden.
- ⑥ Alle sichtbaren Oberflächen sollen leicht spiegelnd sein.
- ⑦ Die Kugeln sollen so platziert sein, dass sie als Spiegelung deutlich in der jeweils anderen Kugel zu sehen sind.

Beispielszene

Aufgabe





- ① Einführung
- ② Kugel
- ③ Ebene
- ④ Reflexion
- ⑤ Beleuchtung
- ⑥ Aufgabe
- ⑦ Animationen



Eine Animation können Sie sehr leicht erstellen:

- Modifizieren Sie die Szenenbeschreibung so, dass z.B. der Augpunkt von einem Parameter abhängt.
- Lassen Sie Ihren Renderer in einer Schleife laufen und erzeugen Sie für jeden Parameterwert eine PNG-Grafik.
- Anschließend fügen Sie die Einzelbilder mit einem Tool Ihrer Wahl zu einer Animation (z.B. mp4) zusammen.

MPG-Datei erstellen

Animationen



Wenn Sie z.B Grafiken mit folgender Benennung erzeugen:

```
goRayTest_000.png  
goRayTest_001.png  
goRayTest_002.png  
...  
goRayTest_120.png
```

können Sie diese mit dem Tool `avconv` mit der Anweisung:

```
avconv -framerate 20 -i goRayTest_%03d.png -c:v libx264 -r 20  
-pix_fmt yuv420p goRayTest.mp4
```

zu einer mp4-Datei zusammenfügen.