

## 0.1 Globale Beleuchtung

### 0.1.1 Rendering-Gleichung

Die **Rendering-Gleichung** bildet die tatsächlichen Verhältnisse bei der Beleuchtung am genauesten ab.

#### Variablen

- $p, p', p'', p''' \dots$  Punkte auf den Oberflächen der Objekte
- $\overline{pp'}$  ... Sehstrahlen.
- $f(p', p'')$  ... Leuchtdichte auf dem Strahl  $p'p''$

#### Gegeben

- $\rho(p', p'', p''')$  ... Welchen Anteil des Lichtes auf  $p'p''$  wird in Richtung  $p''p'''$  zurückgeworfen?

**Beispiel** perfekter Spiegel:

$$\rho(p', p'', p''') = \begin{cases} 1, & \text{wenn die Winkehalbierende von } p'p''p''' \text{ senkrecht auf der Fläche steht} \\ 0, & \text{sonst} \end{cases}$$

#### Rendering-Gleichung

$$f(p'', p''') = \int_{p': p'p'' \text{ ist sichtbar}} f(p', p'') \rho(p', p'', p''') \underbrace{dp'}_{\text{Flächenintegral}} + \underbrace{e(p'', p''')}_{\substack{\text{Wass wird von} \\ p'' \text{ als Lichtquelle} \\ \text{in Richtung } p''' \\ \text{ausgestrahlt?}}}$$

Integralgleichung.  $f$  ist eine Funktion von 4 Variablen

$$f(p'', p''', \lambda) = \int \dots \rho(p', p'', p''', \lambda) \dots, \lambda \in \mathbb{R} \text{ bzw. } \lambda \in \{R, G, B\}$$

$\Rightarrow$  In wirklichkeit auch von  $\lambda$  abhängig.

### 0.1.2 Radiosity-Verfahren

Vereinfachung der Rendering-Gleichung

- Betrachtet alle Wechselwirkungen zwischen Flächen
- nur diffuse Flächen
- Die Flächen werden in kleine Flächenstücke zerlegt; auf die Art wird das Problem diskreditiert  
 $\rightarrow$  großes lineares Gleichungssystem

**Formfaktor  $f_{ij}$ :** Welcher Anteil des Lichtstroms, der von Fläche  $A_i$  ausgesendet wird, kommt bei der Fläche  $A_j$  an?

#### Variablen

- $b_i \dots$  Beleuchtungsstärke der Fläche  $A_i$ . Wieviel Lichtstrom wird pro Flächeneinheit ausgeschickt?
- $\rho_i \dots$  Reflektionskoeffizient der Fläche  $A_i$
- $e_i \dots$  Eigenstrahlungslichtstärke von  $A_i$ , wenn  $A_i$  eine Lichtquelle ist.

Summe des eintreffenden Lichtes auf  $A_i$ :

$$\sum_j b_j \cdot |A_j| \cdot f_{ji}$$

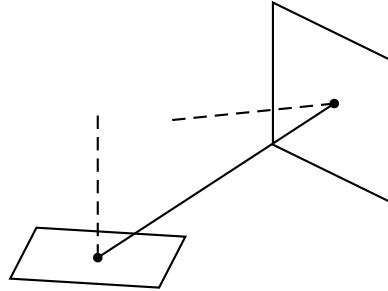
Ausgeschicktes Licht auf  $A_i$ :

$$b_i \cdot |A_i| = \rho_i \cdot \sum_j b_j \cdot |A_j| \cdot f_{ji} + e_i \cdot |A_i|$$

Dies muss man für  $R, G, B$  bzw. für jede Wellenlänge  $\lambda$  separat lösen.

### Formfaktoren

Hängen von der Geometrie ab.



$p \in A_i, q \in A_j$

$\alpha, \beta$  Winkel zu den Flächennormalen

$r = \|p - q\|$

**Annahme**  $A_i$  und  $A_j$  relativ klein, Sicht ist nicht unterbrochen ...  $r, \alpha, \beta$  schwanken nur wenig.

**Näherungsformel** (Fehler gering wenn  $p$  und  $q$  weit aus einander)

$$f_{ij} \approx \cos \alpha \cdot \cos \beta \cdot \frac{1}{r^2} \cdot |A_j| \cdot \frac{1}{\pi}$$

### Exakte Formel

$$f_{ij} = \int_{p \in A_i} \int_{q \in A_j} \frac{\cos \alpha(p, q) \cdot \cos \beta(p, q) \cdot s(p, q)}{\|p - q\|^2} dq dp \cdot \frac{1}{|A_i|} \cdot \frac{1}{\pi}$$

$$s(p, q) = \begin{cases} 1, & p \text{ und } q \text{ sehen sich} \\ 0, & \text{sonst} \end{cases}$$

$$f_{ij} \cdot |A_i| = f_{ji} \cdot |A_j|$$

$$b_i \cdot |A_i| = \rho_i \cdot \sum_j b_j \cdot |A_j| \cdot f_{ji} + e_i \cdot |A_i|$$

$$b_i \cdot \cancel{|A_i|} = \rho_i \cdot \sum_j b_j \cdot \cancel{|A_j|} \cdot f_{ij} + e_i \cdot \cancel{|A_i|}$$

$$\Rightarrow \boxed{b_i = \rho_i \sum_j b_j \cdot f_{ij} + e_i} \quad \text{Radiosity-Gleichungssystem}$$

lineares Gleichungssystem in  $n$  Variablen  $b_1, \dots, b_n$  ( $n = \# \text{Flächenstücke}$ ),  $O(n^2)$  Koeffizienten  $f_{ij}$   $0 \leq \rho_i \leq 1$

### Geometrische „Berechnung“ von $f_{ij}$

$A_i$  ist ein kleines Flächenstück

Projektion auf die obere Einheitskugel um  $A_i$

$\hat{=}$  Multiplikation mit  $\cos \beta \cdot \frac{1}{r^2}$

$\hat{=}$  scheinbare Größe von  $A_j$  aus Sicht von  $A_i$

Anschließende Projektion senkrecht auf die Ebene durch  $A_i \triangleq$  Multiplikation mit  $\cos \alpha$   
 gesamter Fläche des Kreises =  $\pi$

$$\sum_i f_{ij} \stackrel{!}{=} 1$$

Wahl des Konstanten Faktors  $\frac{1}{\pi}$  wird durch die physikalische Forderung  $\sum_j = 1$  erzwungen.

Das Gleichungssystem löst man am besten iterativ:

- Beginne mit einer beliebigen Ausgangslösung  $\vec{b}^{(0)}$  z. B. durch Phong-Beleuchtung berechnet. oder

$$\vec{b}^{(0)} = \begin{pmatrix} b_1^{(0)} \\ \vdots \\ b_n^{(0)} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

- Setze  $b^{(k)}$  rechts ein und erhalte die nächste Näherungslösung

$$b_i^{(k+1)} = \sum_j b_j^{(k)} \cdot f_{ij} + e_i, \quad i = 1, \dots, n$$

Das konvergiert umso schneller, je kleiner die  $\rho_i$ -Werte sind.

$$x = Ax \qquad A \in \mathbb{R}^{n \times n} \qquad b \in \mathbb{R}^n \qquad x \in \mathbb{R}^n$$

$$x^{(0)}, x^{(1)} = Ax^{(0)} + b, x^{(2)} = Ax^{(1)} + b$$

GAUSS-SEIDEL-Verfahren (iteratives Verfahren zur Lösung des Gleichungssystems). Konvergiert unter gewissen Bedingungen. z. B. wenn die Summe der Absolutbeträge in jeder Zeile von  $A < 1$  ist.

Bei der Radiosity-Gleichung ist dies fast der Fall:  $\leq 1$ .

Falls alle  $\rho_i < 1$  sind:  $< 1$

### Zusammenfassung

1. Flächen in kleine Stücke zerlegen

a) gleichförmiges Gitter

b) angepasst an den erwarteten Lichtaustausch.

Wo Flächen nah an anderen Flächen sind wird die Zerlegung feiner.

c) adaptiv.

Man rechnet zuerst mit einer gröberen Zerlegung und verfeinert sie dort, wo das Ergebnis stark schwankt

2. Formfaktoren bestimmen (am zeitaufwändigsten)

Können auch mit Hilfe von Grafikpuffern berechnet werden.

$$\int_{p \in A_i} \int_{q \in A_j} \dots$$

Für ein festes  $p$  ähnelt die Berechnung des Integranden der Grafikpufferberechnung mit Kamera  $p$ . Diese Berechnung wird von Grafik-Hardware unterstützt

3. Gleichungssystem lösen

4. mit den so bestimmten (diffusen) Helligkeiten kann man dann die Szene darstellen

## 0.2 Transparenz

teildurchsichtige Medien. z. B. Glas mit einem Bild, mit Staub, Nebel, Wolken, Rauch.

## 0.2.1 RGBA-Modell bzw. $(r, g, b, \alpha)$

Ein halbdurchsichtiges Bild wird durch eine vierte Größe  $\alpha$  (zusätzlich zu  $R, G, B$ ) für jeden Bildpunkt

|                |  |
|----------------|--|
| $\alpha = 1$   | vollständig undurchsichtig   |
| $\alpha = 0$   | vollständig durchsichtig   |
| $\alpha = 0,3$ | 30% wird von diesem Bild beigesteuert,<br>70% kommt von dem was dahinter liegt |

z. B. Bild eines Mauszeigers (idealisiert)

in Wirklichkeit natürlich „pixelig“, Übergänge zwischen den kanten mit Antialiasing geglättet:

$$\left. \begin{matrix} (r, g, b, \alpha) \\ (\bar{r}, \bar{g}, \bar{b}) \text{ (Hintergrund)} \end{matrix} \right\} \text{Ergebnis} = (\alpha \cdot r + (1-\alpha) \cdot \bar{r}, \alpha \cdot g + (1-\alpha) \cdot \bar{g}, \alpha \cdot b + (1-\alpha) \cdot \bar{b})$$

$$\text{Ergebnis} = \alpha^1 rgb^1 + (1 - \alpha^1) \alpha^2 rgb^2 + (1 - \alpha^1) (1 - \alpha^2) \alpha^3 rgb^3 + (1 - \alpha^1) (1 - \alpha^2) (1 - \alpha^3) \underbrace{rgb}_{\text{Vektor aus 3 Komponenten}}$$

← mehrere Transparente Schichten

$$\left( \underbrace{\frac{\alpha^1 rgb^1 + (1 - \alpha^1) \alpha^2 rgb^2}{\alpha^1 + (1 - \alpha^1) \alpha^2}}_{rgb^{neu}}, \underbrace{\alpha^1 + (1 - \alpha^1) \alpha^2}_{\alpha^{neu}} \right)$$

Effekt auf eine dahinter liegende Fläche  $\overline{rgb}$

$$\alpha^{neu} rgb^{neu} + (1 - \alpha^{neu}) \overline{rgb}$$

Wichtig: durchlässige Medien in der richtigen Reihenfolge kombinieren (z. B. vorne nach hinten, oder hinten nach vorne)

## 0.2.2 Kombination mit Tiefenpuffer

$(rgb^1, z^1, \alpha^1)$   $(rgb^2, z^2, \alpha^2)$  einer dieser Werte ist im Tiefenpuffer an einer bestimmten Stelle gespeichert, der andere soll dort hingeschrieben werden

$$\begin{aligned} z^1 &< z^2 \text{ (o. B. d. A.)} \\ rgb^{neu} &:= \frac{\alpha^1 rgb^1 + (1 - \alpha^1) \alpha^2 rgb^2}{\alpha^1 + (1 - \alpha^1) \alpha^2} \\ \alpha^{neu} &:= \alpha^1 + (1 - \alpha^1) \alpha^2 \\ z^{neu} &:= z^1 \end{aligned}$$

(eigentlich unzureichend, aber notwendig, um wenigstens den undurchsichtigen Fall richtig zu behandeln)

Diese Methode funktioniert, wenn man höchstens *ein* halbdurchlässiges Pixel ( $0 < \alpha < 1$ ) zeichnen möchte oder wenn *alle* Objekte von vorne nach hinten oder von hinten nach vorne eingefügt werden.

# 1 Spline-Kurven und -Flächen

z. B. Kreis

| implizite Darstellung | explizite Darstellung                         | Parameterdarstellung                                   |
|-----------------------|---|--|
| $x^2 + y^2 = 1$       | $y = \sqrt{1 - x^2}$<br>$y = -\sqrt{1 - x^2}$ | $y = \cos \alpha$<br>$x = \sin \alpha$                 |
| <hr/>                 |   |  |
|                       | <i>Spezialfall</i>                            | <i>Alternative (rationale Darstellung)</i>             |
|                       | $x = t$                                       | $x = \frac{2t}{1+t^2} \quad t = \tan \frac{\alpha}{2}$ |
|                       | $y = \sqrt{1 - t^2}$                          | $y = \frac{1-t^2}{1+t^2} \quad -\infty < t < \infty$   |

- Parameterdarstellung ist gut für das Nachfahren der Kurve; Darstellung als Folge von Punkten
- implizite Darstellung ist gut für Raytracing

**Definition Spline-Kurven** sind parametrische Kurven, wo sich Parameterfunktionen Polynome sind (es gibt auch rationale Splines). Die Kurven sind durch Leitpunkte (Kontrollpunkte) festgelegt. Dadurch sind sie einigermaßen intuitiv manipulierbar.

## 1. Bézier-Splines

$k + 1$  Kontrollpunkte definieren einen Bézier-Spline der Ordnung  $k$  (Polynome vom Grad  $\leq k$ )

## 2. B-Splines

Ordnung  $k$ : stückweise Polynome der Ordnung  $k$ , Kontrollpunkte  $< \infty$

## 3. Hermite-Splines

interpolieren zwischen 2 Endpunkten mit vorgegebenen Tangentenrichtungen und Geschwindigkeiten  
... kubische Splines