

Lokale Beleuchtungsmodelle

Farbschattierung der Oberflächen abhängig von:

- Position, Orientierung und Charakteristik der Oberfläche
- Lichtquelle■

Vorgehensweise:

1. **Modell** (Objektbeschreibung, Szenenbeschreibung)■
2. **Projektion** (Bildraum, Kameramodell, Koordinatensystem)■
3. **Bestimmung Beleuchtung** (oft in Schattierung integriert)■
4. **Bestimmung Sichtbarkeit** (z.B. Z-Buffer, Raycasting)■
5. **Schattierung** (Ausrechnen der Pixelwerte, Beleuchtungsmodell)

Ein einfaches Beleuchtungsmodell

→ Punktlichtquellen werden vorausgesetzt

Oberflächen reflektieren Licht als

1. diffuse Reflexion
2. spekulare Reflexion (gerichtet)

keine:

- Spiegelungen
- echten Transmissionen

1. Term: Ambientes Licht

→ einfachstes Beleuchtungsmodell

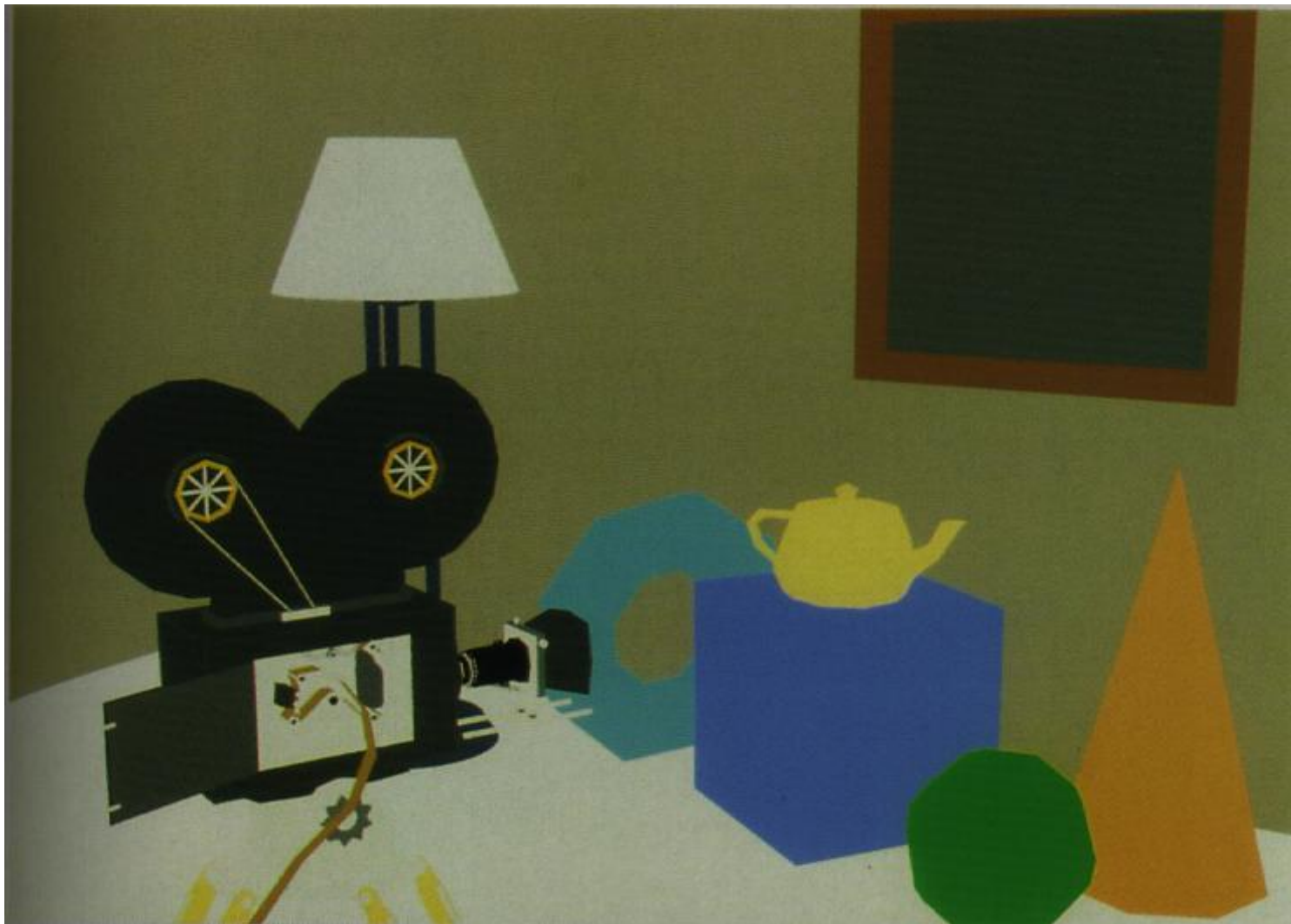
→ verwende diffuse, ungerichtete Lichtquelle als Summe aller Reflexionen:

$$I = k_a I_a$$

I_a : Intensität des ambienten Lichtes (konstant für alle Objekte)

k_a : materialabhängiger ambienter Reflexionskoeffizient ($k_a \in [0, 1]$)

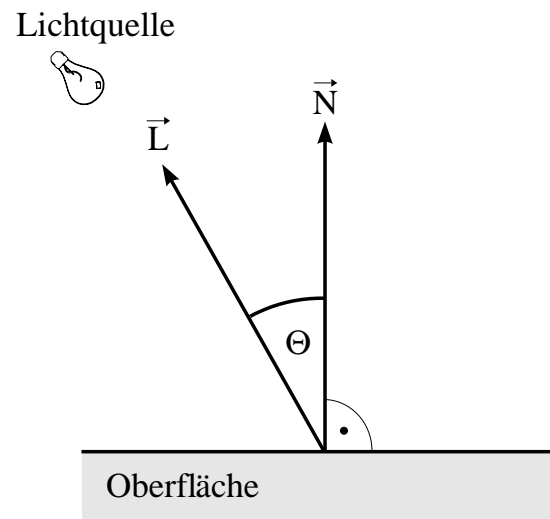
Verwendung: Grundhelligkeit an nicht direkt beleuchteten Oberflächen



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles and Practice

2. Term: Diffuse Reflexion

- matte Oberflächen reflektieren Licht einer Lichtquelle gleichmäßig diffus in alle Richtungen.■
- Intensität der Reflexion hängt vom Winkel zwischen Oberfläche und Lichtstrahl ab



Gesetz von Lambert

$$I = k_d I_{Lq} \cos(\theta)$$

I_{Lq} : Intensität der Lichtquelle, θ : Einfallswinkel ($\theta \in [0^\circ, 90^\circ]$),
 k_d : diffuser Reflektionskoeffizient ($k_d \in [0, 1]$) ■

→ sind \vec{N} und \vec{L} normiert: Cosinus durch Skalarprodukt:

$$I = k_d I_{Lq} \max(0, (\vec{N} \cdot \vec{L})) \quad \blacksquare$$

Nachteile:

- alle nicht direkt beleuchteten Objekte sind schwarz
- Objekte in unterschiedlicher Entfernung zur Lichtquelle besitzen gleiche Intensität

Lösung

1. Kombination ambientes und diffuses Modell
2. Addition eines entfernungsabhängiger Parameters

$$I = k_a I_a + f(d) k_d I_{Lq} (\vec{N} \cdot \vec{L}) \quad f(d) : \text{Abstandsfunktion}$$



→ weitere Modifikation: mehrere Lichtquellen, RGB-Komponenten

$$I^p = k_a^p I_a^p + \sum_i f(d_i) k_d^p I_{Lq_i}^p (\vec{N} \cdot \vec{L}_i) \quad p \in \text{rot, grün, blau}$$



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles and Practice

3. Term: Spekulare Reflexion

→ bei glänzenden Oberflächen: Glanzlichter von Lichtquellen

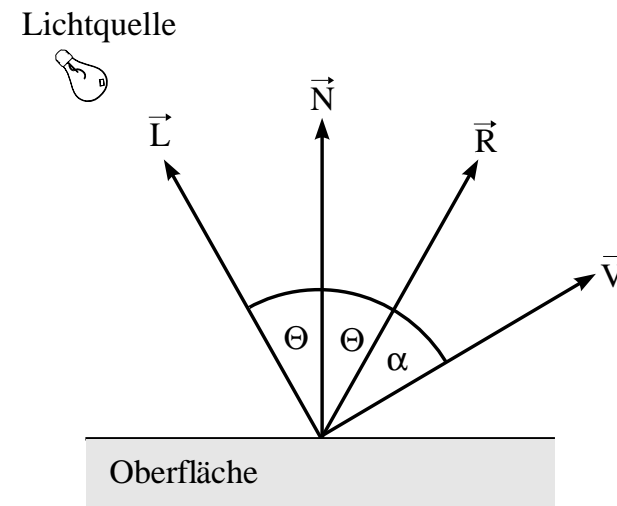
→ perfekt spiegelnde Oberfläche (Totalreflexion):

Licht wird nur in Richtung des Reflexionsvektors gespiegelt:

$$\vec{R} = 2 \vec{N} (\vec{N} \cdot \vec{L}) - \vec{L}$$

\vec{N} : Oberflächennormale

\vec{L} : einfallender Strahl



→ die wenigsten Oberflächen sind perfekte Spiegel ■

→ daher: Mischung diffuser und gerichteter Reflexion:

$$I = I_{ambient} + f(d) (I_{diffus} + I_{spiegelnd})$$

Phongsches Beleuchtungsmodell

Phong modelliert gerichteten Anteil über

$$I_{\text{spiegelnd}} = k_s I_{Lq} \cos^n(\alpha)$$

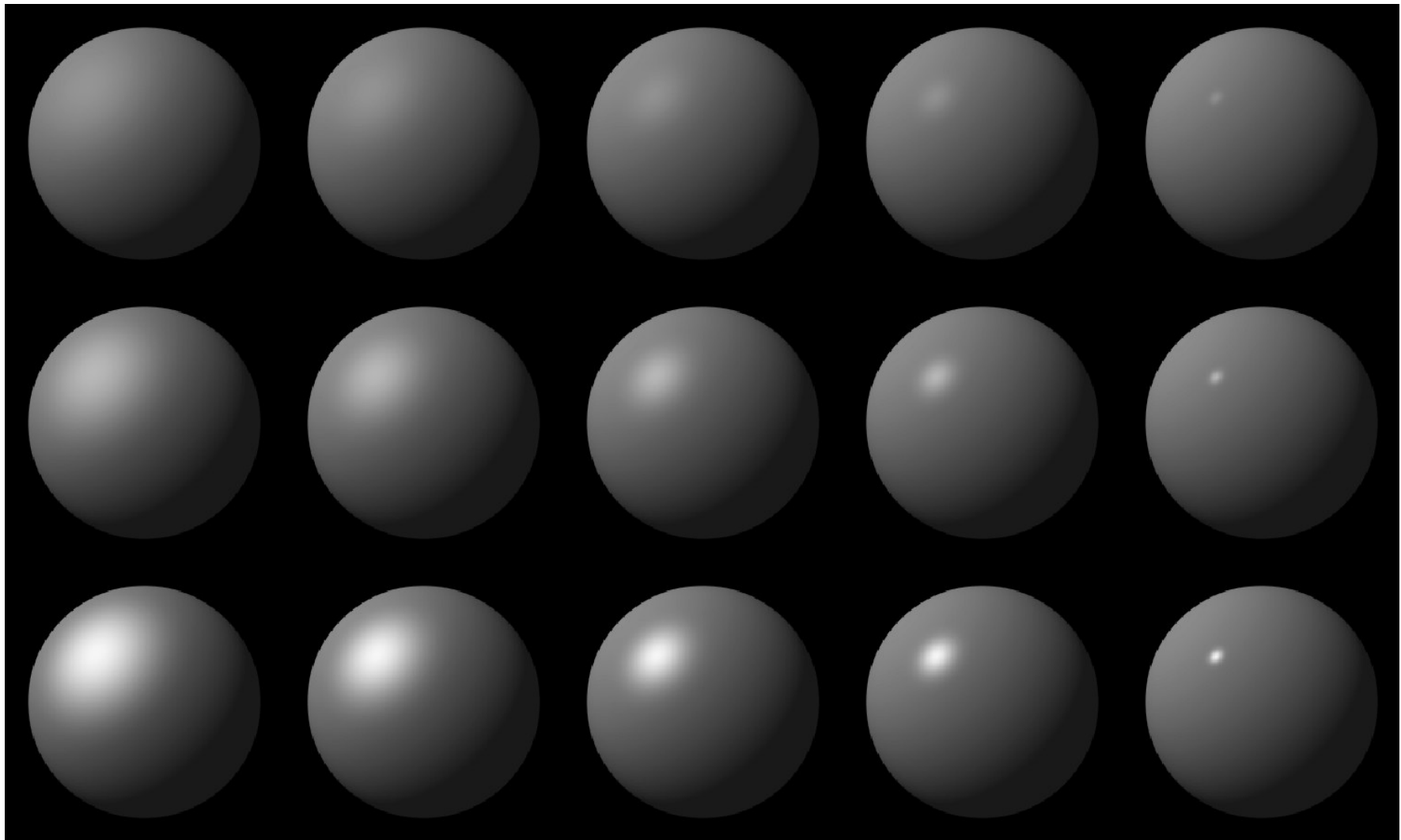
k_s : spiegelnder Reflexionskoeffizient

n : Materialkoeffizient (Maß für die Schärfe des Glanzlichtes)■

→ auch hier: sind Vektoren normiert, kann der Kosinus durch Skalarprodukt $(\vec{V} \cdot \vec{R}_i)$ ausgedrückt werden■

Vollständiges Beleuchtungsmodell (Phong):

$$I = k_a I_a + \sum_i f(d_i) I_{Lq_i} [k_d (\vec{N} \cdot \vec{L}_i) + k_s (\vec{V} \cdot \vec{R}_i)^n]$$



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Schattenberechnung

Schattenterm wird in Beleuchtungsformel eingebaut:

$$S_i = \begin{cases} 0 & , \text{ wenn Licht } i \text{ an diesem Punkt blockiert} \\ 1 & , \text{ sonst} \end{cases}$$

■

$$I = k_a I_a + \sum_i S_i f(d_i) I_{Lq_i} \left[k_d (\vec{N} \cdot \vec{L}_i) + k_s (\vec{V} \cdot \vec{R}_i)^n \right]$$

■

→ Objekt im Schatten, wenn es von Lichtquelle aus nicht sichtbar

Implementierung:

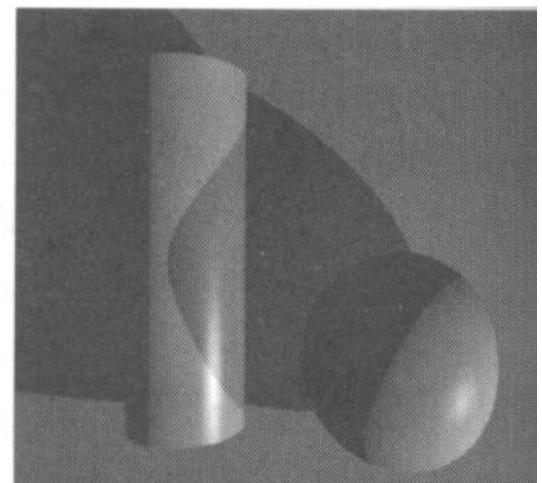
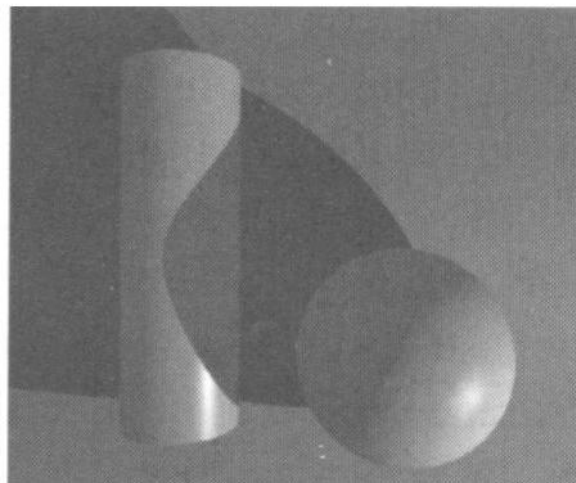
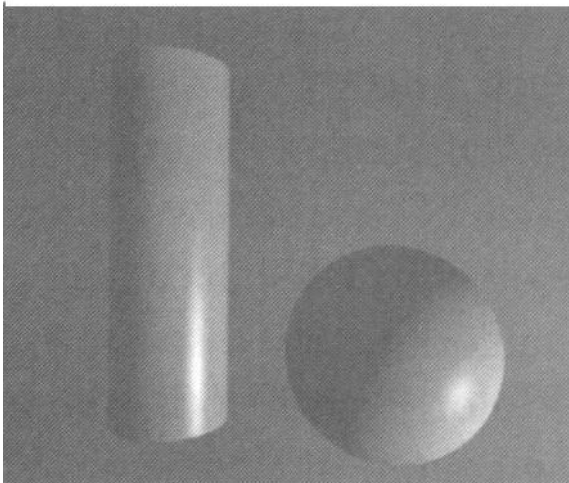
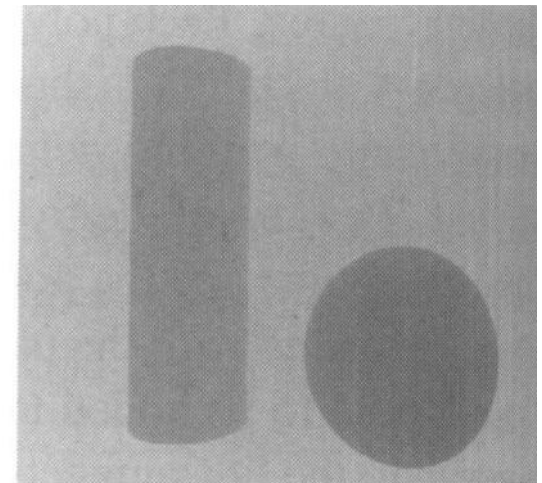
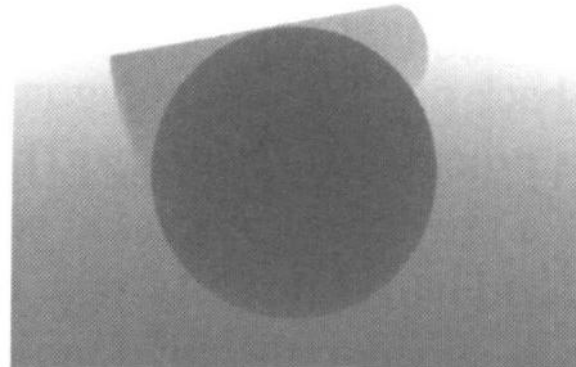
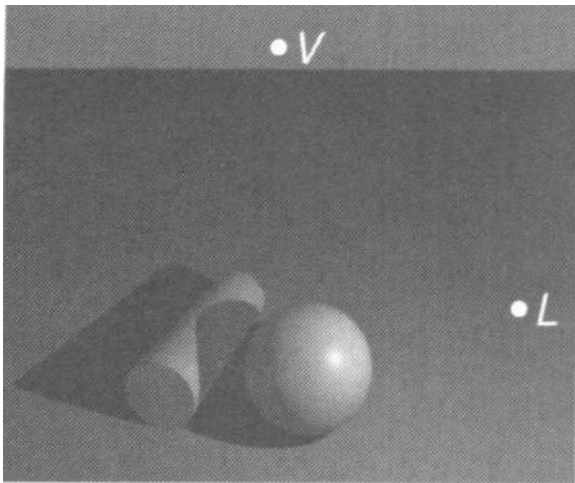
1. Verwende analytischen Sichtbarkeitsalgorithmus ■
2. Pixelbasiert (Shadow-Map) ■
 - Berechne z-Buffer von Betrachter und von Lichtquelle aus ■

Algorithmus:

Für alle Pixel des Bildes

Bestimme Koordinaten in der Lichtquelle (x,y,z) ■

*Wenn Tiefe größer als Wert im z-Buffer der Lichtquelle
blockiere Punkt → Schatten*



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Schattierungsverfahren

Wann und wie wird ein Beleuchtungsverfahren angewendet.

→ **nicht verwechseln mit Schattenberechnung !**

Beispiele:

- Phong-Shading: bestimmt Beleuchtung explizit an jedem Pixel
- Gouraud-Schattierung: nur an Eckpunkten von Dreiecken
- Raytracing: Strahlverfolgung durch jedes Pixel

Kategorisierung von Schattierungsverfahren

lokale Verfahren (empirische Beleuchtungsmodelle)

Flat-, Gouraud- und Phong-Shading■

globale Verfahren (physikalische Beleuchtungsmodelle)■

- Strahlverfolgungs- oder Raytracing-Algorithmen■
- Energiegleichgewichts- oder Radiosity-Methoden

Flat Shading

- einfaches Schattierungsverfahren■
- keine entfernungsabhängige Lichtabnahme■
- keine Schatten■
- Betrachter befindet sich in unendlicher Entfernung: $\Rightarrow (\vec{N} \cdot \vec{L})$ konstant■
- nur Hintergrundbeleuchtung und diffuse Reflexion■

$$I = k_a I_a + k_d I_{Lq} (\vec{N} \cdot \vec{L})$$

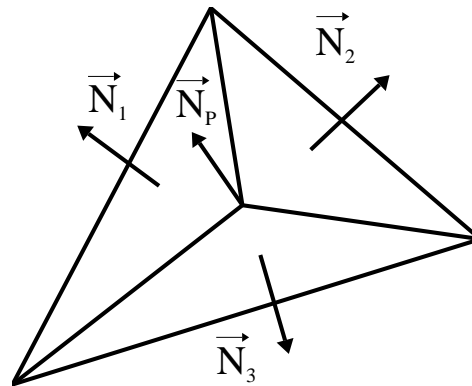
- funktioniert nicht für gekrümmte Flächen■
- Mach-Band-Effekt (unregelmäßige Helligkeitsabstufung durch Kantenverstärkung im Auge)



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Gouraud Shading

- Intensitätsbestimmung für Punkte auf Flächen durch Interpolation der Helligkeiten an den Ecken■
- benötigt Normalen in den Flächenendpunkten■



- für Dreiecksnetze (n_p Nachbarn) Berechnung durch:

$$\vec{N}_p = \frac{1}{n_p} \sum_i^{n_p} \vec{N}_i$$

- nun Bestimmung der Intensitäten in den Eckpunkten über Normalen und entsprechendem Beleuchtungsmodell■
- danach lineare Interpolation der übrigen Flächenpunkte

Scan-line-orientierter Algorithmus

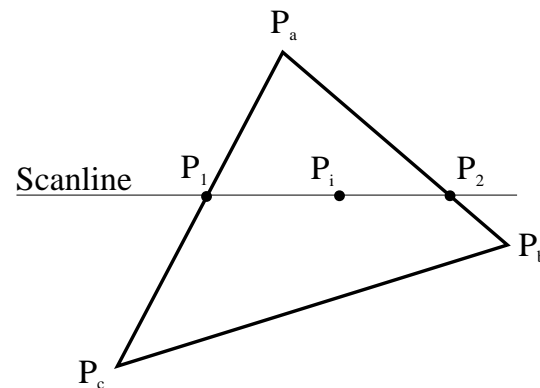
→ zunächst Berechnung der Intensitäten der Punkte

$P_1(x_1, y)$ und $P_2(x_2, y)$

$$I_1 = I_a \frac{y - y_c}{y_a - y_c} + I_c \frac{y_a - y}{y_a - y_c} \quad \text{und} \quad I_2 = I_a \frac{y - y_b}{y_a - y_b} + I_b \frac{y_a - y}{y_a - y_b}$$

→ dann Intensität für $P_i(x_i, y)$ über P_1 und P_2

$$I_i = I_1 \frac{x_2 - x_i}{x_2 - x_1} + I_2 \frac{x_i - x_1}{x_2 - x_1}$$



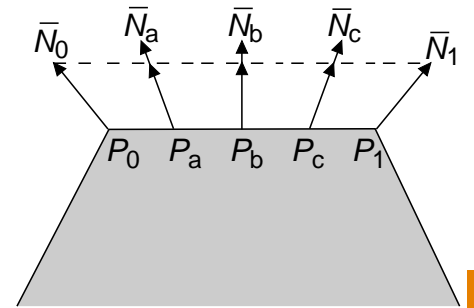
→ kontinuierliche Intensitätsabstufung innerhalb der Flächen
(und auch über Ränder hinaus)■

→ Nachteile:

- höherer Berechnungsaufwand
- Mach-Band-Effekt teilweise noch sichtbar
- unrealistische Darstellung von Spiegelungsflächen

Phong-Shading

- Interpolation der Normale (nicht der Intensität) ■
- für jedes Pixel wird aus Normale die Intensität berechnet ■
- Folgen: genauer, langsamer



- Unterschied Gourand zu Phong:



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Spekular mit Gourand



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Spekular mit Phong



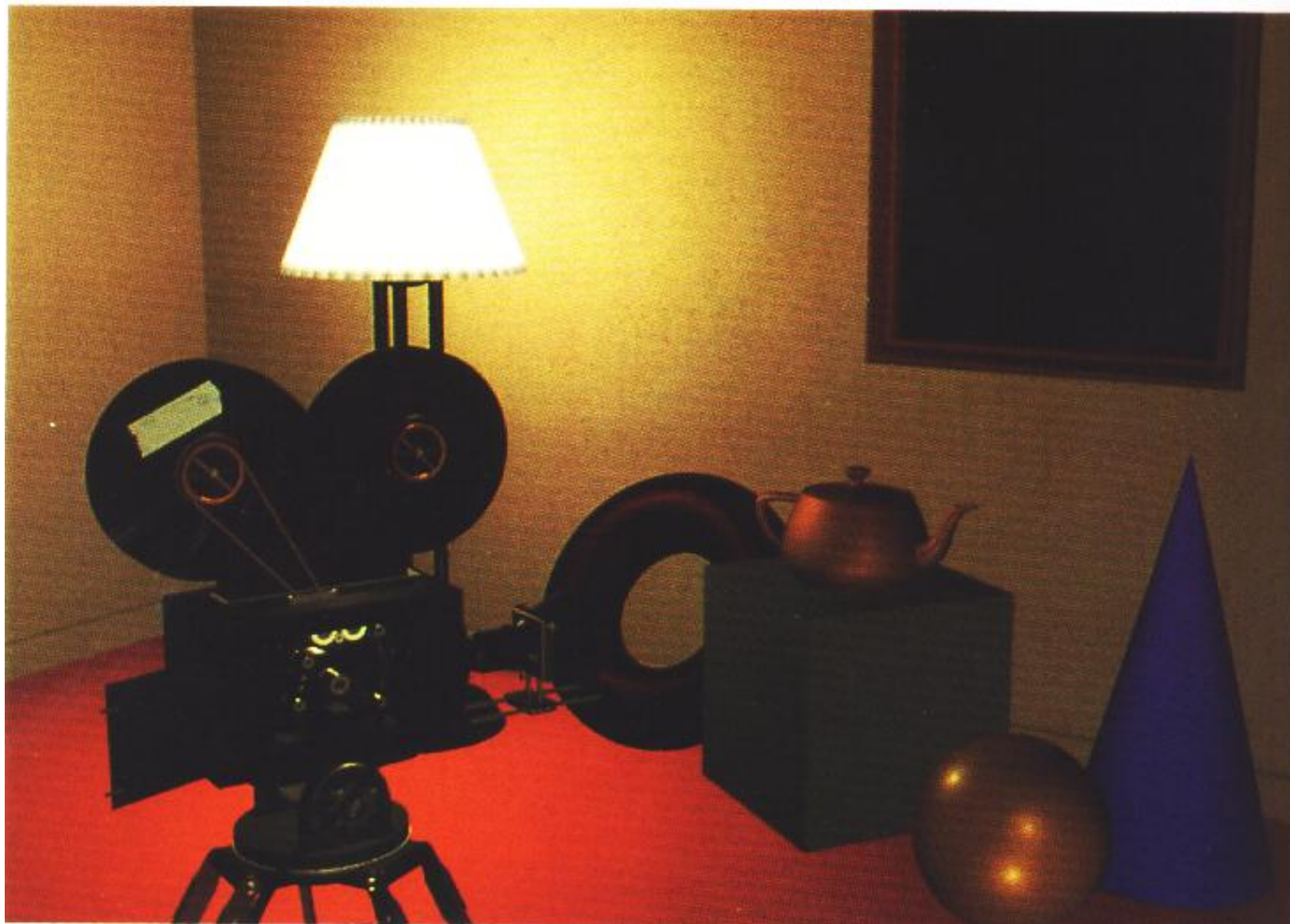
aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Gekrümmte Flächen



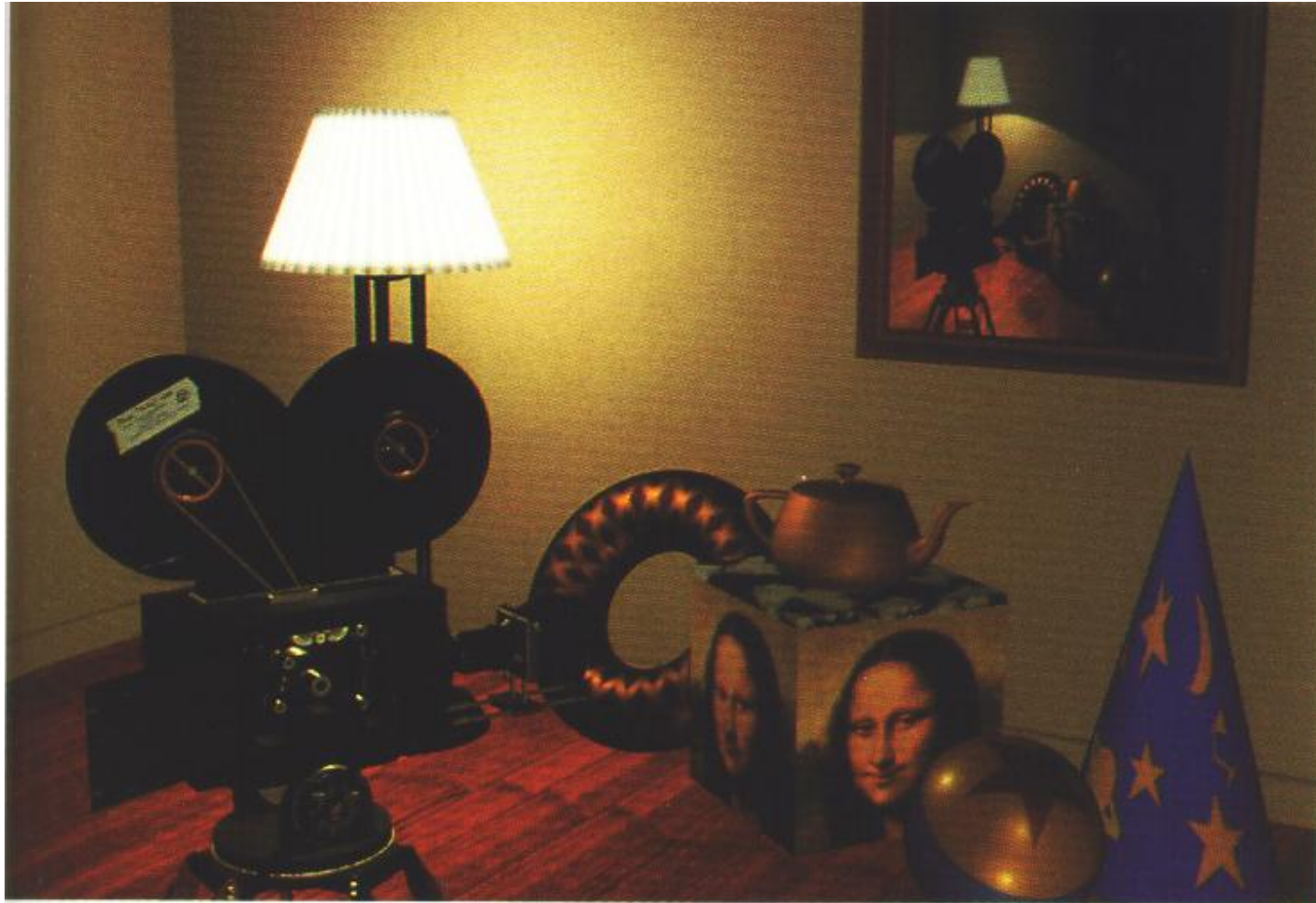
aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Mehrere Lichter



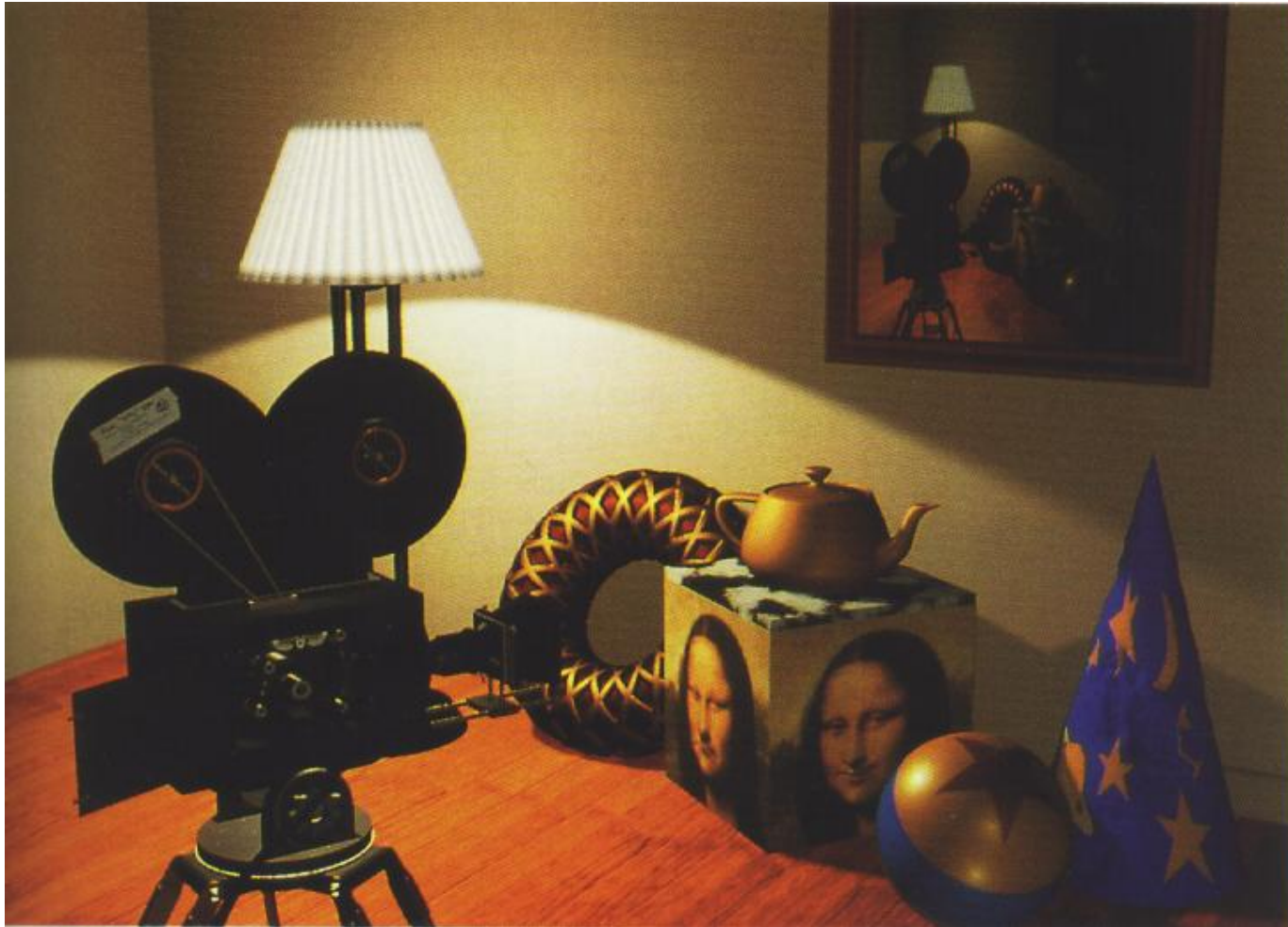
aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Texture Mapping



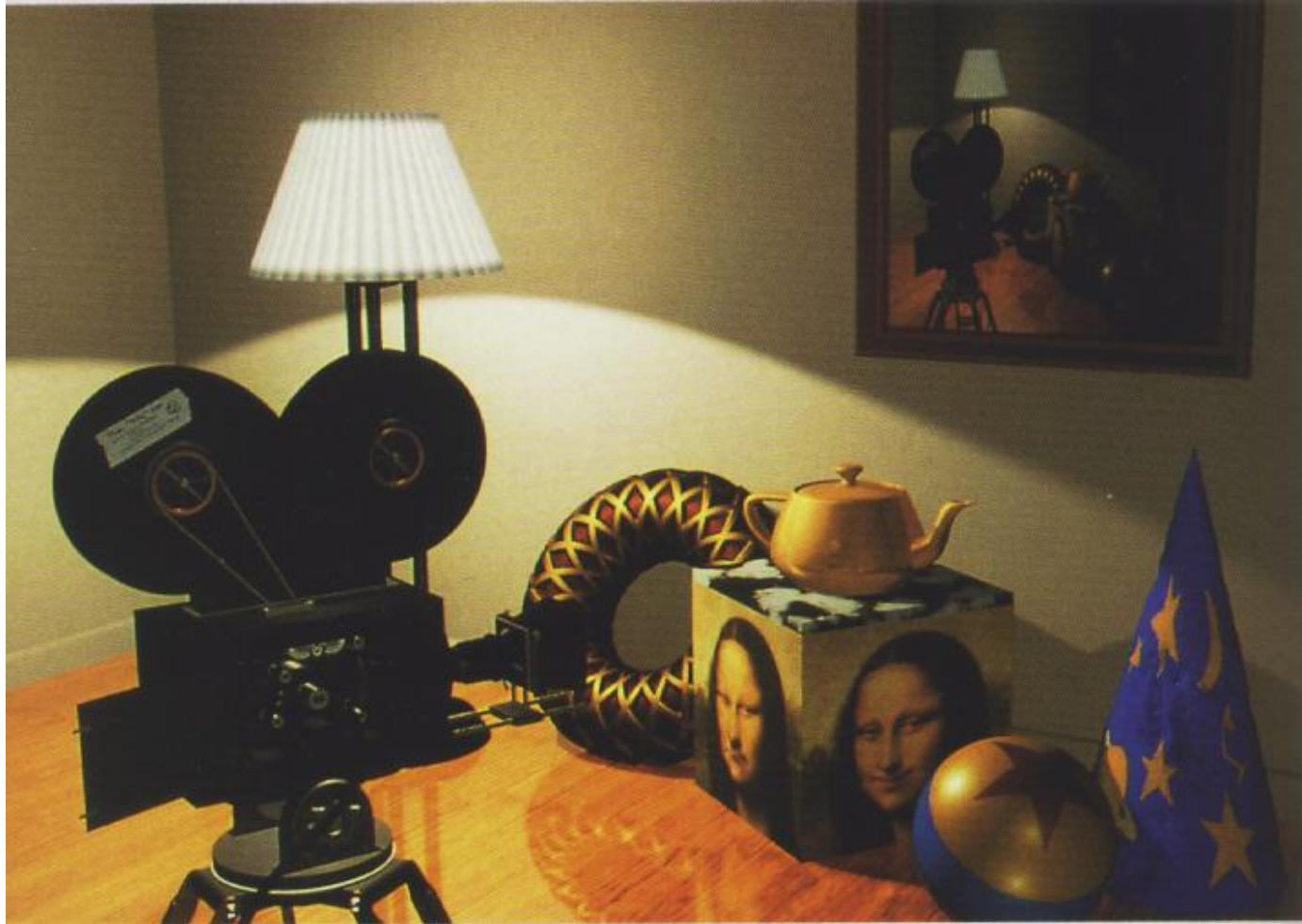
aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Displacement Mapping



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles

Reflection Mapping



aus: Foley, van Dam, Feiner, Hughes: Computer Graphics, Principles