

## 10. Grundzüge der Beleuchtungsrechnung

- *Beleuchtungsmodelle (illumination models, lighting models)* beschreiben die Faktoren, die die Farbe eines Objekts an einem Punkt bestimmen
- *Shading models* beschreiben, wann und wie ein Beleuchtungsmodell angewendet wird – und erfüllen häufig (auch) den Zweck der *visuellen Glättung* von Polygonnetzen

### Prinzipien der Materie-Licht-Interaktion

Interaktion wird bestimmt durch Eigenschaften der Lichtquelle, optische Eigenschaften des beleuchteten Materials und eventuell durch weitere Objekte (Atmosphäre: Nebel, Dunst..., Spiegelung anderer Objekte...)

Struktur, Aufbau und Bearbeitung der Oberfläche bestimmen die optischen Eigenschaften

⇒ komplexes Problem!

Ziele:

- Realitätsnähe der Bilder
- Erleichterung der Objekterkennung durch den Menschen ("shape from shading")
- schneller Bildaufbau

Physikalische Grundlagen:

- geometrische Optik (Strahlenoptik)
- Wellenoptik
- Quantenoptik

Phänomene, die eine Rolle spielen können:

*Reflektion, Transmission, Absorption, Diffraktion (Beugung), Refraktion (Brechung), Dispersion (Streuung), Interferenz, Polarisation...*

In der Computergrafik wurde beträchtlicher Aufwand in die Modellierung und Simulation dieser Effekte (insbes. der Reflektion) investiert!

## Physikalische Grundlagen zu Beleuchtungsmodellen

(nach Krömker 2001)

### *Radiometrie und Photometrie*

**Radiometrie:** Physikalische Beschreibung elektromagnetischer Energie, z.B. Betrag der Lichtenergie je Wellenlänge

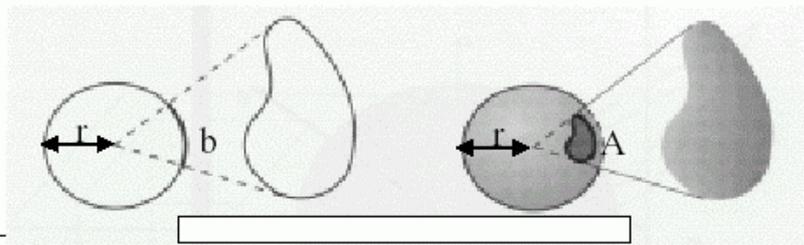
**Photometrie:** psychophysikalische Messung der "visuellen Energie", die von der elektromagnetischen Strahlung induziert wird

Menschliche Augen sind empfindlich für elektromagnetische Strahlung zwischen ca. 380 nm und 770 nm (jedoch in unterschiedlichem Maß empfindlich für verschiedene Wellenlängen); siehe Kap. 2.

### Radiometrische Grundgrößen

#### *Raumwinkel*

räumliches Winkelmaß  $\omega$ , das analog zum Bogenwinkel in der Ebene definiert wird:



Bogenmaß: Verhältnis  $b/r$  der überdeckten Strecke  $b$  zum Radius  $r$  des Kreises. Es gilt stets  $0 \leq b \leq 2\pi r$ . Die Einheit für den Winkel ist rad.

Raumwinkel: Verhältnis der durch  $A$  überdeckten Fläche zum Quadrat des Kugelradius  $r$   $\omega = \frac{A}{r^2}$ . Die Einheit des Raumwinkels ist sr.

Der volle Raumwinkel ist per Definition die Fläche einer Einheitskugel, hat also die Größe  $4\pi sr$ .

(sr = steradian, "stereo radiant")

Zwischen dem Differential  $d\omega$  des Raumwinkels und den Polarkoordinaten-Winkeln  $\theta$  (= Winkel zum Zenith) und  $\phi$  besteht der folgende Zusammenhang:

$$d\omega = \sin \theta \, d\theta \, d\phi$$

(wichtig für Integralberechnungen, wenn über räumliche Winkelfelder integriert werden soll).

*Strahlungsenergie* (radiant energy)  $Q$ : wichtigste Größe der Radiometrie. Einheit J (Joule).

*Strahlungsenergiedichte* (radiant energy density)  $U$ :  
Strahlungsenergie pro Einheitsvolumen.  $U = dQ/dV$ ; Einheit  $J/m^3$ .

*Strahlungsleistung*, auch: Strahlungsfluss (radiant flux)  $\Phi$ :  
zeitliche Ableitung der Strahlungsenergie.  $\Phi = dQ/dt$ ; Einheit  $J/s = W$  (Watt).

*Strahlstärke* oder *Intensität* (Intensity)  $I$  einer Strahlungsquelle:  
Strahlungsleistung pro Einheitsraumwinkel.  $I = d\Phi/d\omega$ ; Einheit  $W/sr$  (Watt pro Steradian).

*Bestrahlungsstärke* (Irradiance): pro Fläche auftreffende  
Strahlungsleistung bei einem bestrahlten Objekt.  $E = d\Phi/dA$ ;  
Einheit  $W/m^2$ .

Die Bestrahlungsstärke nimmt mit dem Quadrat der Entfernung zur Strahlungsquelle ab.

Die entsprechende Sendegröße (für eine Strahlungsquelle) heißt *spezifische Ausstrahlung* oder *Radiosity*.  $E = d\Phi/dA$  (für Lichtquelle); Einheit  $W/m^2$ .

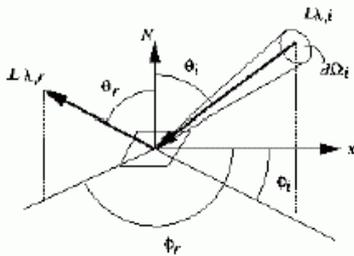
## Reflektion

Auf eine Fläche trifft Strahlung auf (mit einer bestimmten Bestrahlungsstärke  $E$ ) und wird (z.T.) wieder ausgesendet.

### BRDF

Die Reflexion von Strahlung wird durch den spektralen Reflexionsfaktor beschrieben, der das Verhältnis von **reflektierter Strahlendichte** zur **einfallenden Bestrahlungsstärke** angibt

$$\rho(\lambda, \phi_r, \theta_r, \phi_i, \theta_i) = \frac{L_{\lambda,r}(\lambda, \phi_r, \theta_r)}{E_{\lambda,i}(\lambda, \phi_i, \theta_i)} = \frac{\text{reflektierte Strahlendichte} [\text{Wsr}^{-1}\text{m}^{-2}]}{\text{einfallende Bestrahlungsstärke} [\text{Wm}^{-2}]}$$



Index i: einfallende Strahlung  
Index r: reflektierte Strahlung

In der englischen Literatur wird diese Größe „**bidirectional reflection distribution function (BRDF)**“ genannt.

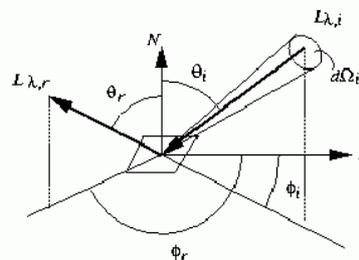
Manche Autoren definieren die BRDF als

$$\rho(\lambda, \phi_r, \theta_r, \phi_i, \theta_i) = \frac{I_{\lambda,r}(\lambda, \phi_r, \theta_r)}{\Phi_{\lambda,i}(\lambda, \phi_i, \theta_i)} = \frac{\text{reflektierte Strahlungsstärke} [\text{Wsr}^{-1}]}{\text{einfallende Strahlungsleistung} [\text{W}]}$$

Für infinitesimal kleine Flächen sind diese Definitionen identisch.

Ferner gilt:

$$E_{\lambda,i}(\lambda, \phi_i, \theta_i) = \int L_{\lambda,i}(\lambda, \phi_i, \theta_i) \cos \theta_i d\Omega_i$$



## Eigenschaften der BRDF

Wichtige Eigenschaften von (BRDF) für die Beleuchtungsmodelle der GDV sind:

1. Reziprozität:  $\rho_\lambda$  ändert sich nicht, wenn Einfallswinkel und Ausfallswinkel vertauscht werden. Hierauf beruht u.a. das Raytracing.
2.  $\rho_\lambda$  ist im allgemeinen anisotrop: Wird bei gleicher Einfallswinkel- und Ausfallrichtung die Fläche um die Normale verdreht, so ändert sich der Anteil des reflektierten Lichts. Typische Beispiele sind Stoffe oder Metalleffektlacke.
3. Superposition: Trifft in einem Punkt Licht aus mehreren Richtungen ein, so beeinflussen sich die einzelnen Reflexionen nicht, sondern können linear überlagert werden. Durch Integration über alle Einfallswinkelrichtungen erhält man:

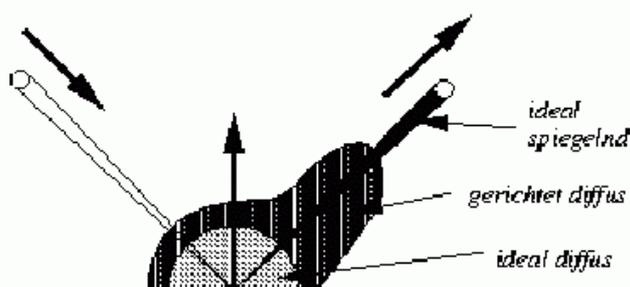
$$L_r = \int_{\Omega_i} \rho L_i \cos \theta_i d\Omega_i$$

Der Reflexionsfaktor ist wegen der Energieerhaltung immer positiv. Da er auf den Raumwinkel bezogen ist und die Dimension (Raumwinkel<sup>-1</sup>) besitzt, kann er im Extremfall den Wert  $\infty$  annehmen. Deshalb wird in der GDV, besonders bei den empirischen Beleuchtungsmodellen, stattdessen mit dem Reflexionsgrad  $r$  gearbeitet, der das Verhältnis von reflektierter zu einfallender Bestrahlungsstärke angibt und deshalb dimensionslos ist:

$$r_\lambda = \frac{E_{\lambda,r}}{E_{\lambda,i}}, \quad 0 \leq r_\lambda \leq 1$$

### Arten des Reflektionsverhaltens:

- ideal diffus
- ideal spiegelnd
- gerichtet diffus (Mischform)



### *Ideal diffuse Reflektion:*

reflektierte Leuchtdichte ist unabhängig von der Abstrahlungsrichtung. "Lambert'sche Reflektion".

Auf dieser Art der Reflektion beruht im Radiosity-Verfahren (siehe nächstes Kapitel) der modellierte Austauschmechanismus für Licht zwischen den Oberflächen der Objekte.

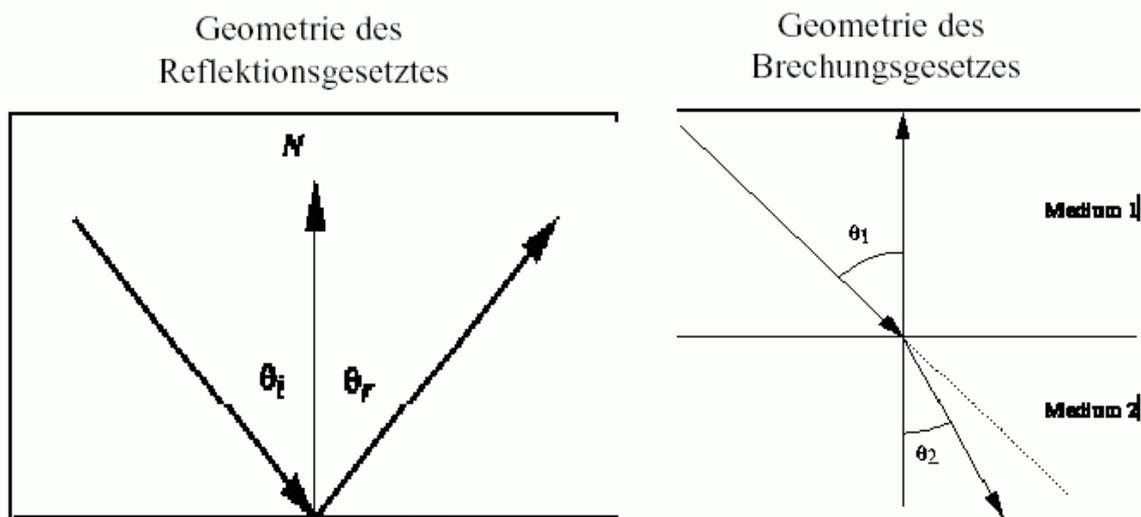
### *Ideal spiegelnde Reflektion:*

der einfallende und der reflektierte Strahl bilden mit der Normalen der reflektierenden Oberfläche gleiche Winkel. Einfallender Strahl, reflektierter Strahl und Oberflächennormale liegen in einer Ebene.

In Polarkoordinaten über der reflektierenden Fläche (Normalenrichtung = Zenith):

$$\theta_r = \theta_i \text{ und } \phi_r = \phi_i + \pi \quad (\text{Index } r: \text{reflected, } i: \text{incoming}).$$

Konventionelles Raytracing beruht auf diesem einfachen Reflexionsgesetz.



In der Natur trifft man die beiden idealen Reflektionsarten selten. Man muss also für alle Oberflächen die richtungsmäßige Verteilung des Reflektionsfaktors  $\rho$  bestimmen. Sehr häufig tritt der Fall auf, dass  $\rho$  ein deutliches Maximum in Richtung der spiegelnden Reflektion hat und kleiner wird, je weiter man sich von dieser Richtung entfernt (*gerichtet diffuse Reflektion*, auch *spekulare Reflektion*). In der Computergrafik ist es üblich, die gerichtet diffuse Reflektion in einen richtungsunabhängigen, diffusen Anteil und einen richtungsabhängigen Anteil aufzuspalten.

*Übliche Vereinfachungen* bei der Materie-Licht-Interaktion:

- **Lichtquellen** (emittieren Licht) und **Objekte** (absorbieren, reflektieren, transmittieren Licht)
- **idealisierte Reflektionen** wirken additiv: diffus, spiegelnd, ambient

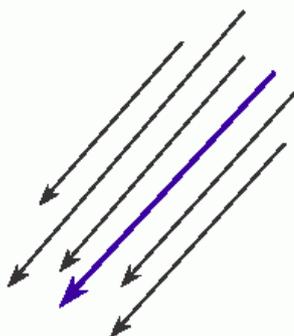
man unterscheidet:

- heuristische Verfahren <---> physikalisch basierte Verfahren
- *lokale* vs. *globale* Beleuchtungsmodelle (lokale Modelle berücksichtigen nur *eine* direkte Lichtquellen-Objekt-Interaktion)

### Lichtquellen

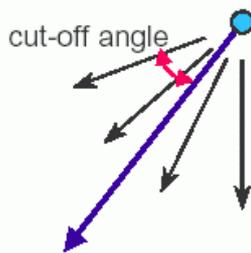
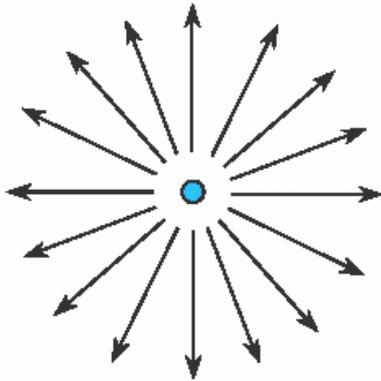
Man unterscheidet die folgenden Idealisierungen:

1.



- ◆ **Richtungslicht**  
(*directional light*)  
Position im Unendlichen  
Richtung  $\mathbf{s}_{dir}$

## 2. Positions-Lichtquellen



### ◆ Punktlichtquelle

Position  $\mathbf{s}_{pos}$

### ◆ Spot Light

Position  $\mathbf{s}_{pos}$

Richtung  $\mathbf{s}_{dir}$

Cut-off Angle  $\mathbf{s}_{cut}$

(Spot Exponent =  
Abfall der Helligkeit mit  
Abstand vom Zentrum)

## Allgemeine Lichtquellenparameter

- ◆ Intensität für verschiedene Beleuchtungsanteile in der Regel in (R,G,B)
  - *ambient intensity color*  $\mathbf{s}_{amb}$
  - *diffuse intensity color*  $\mathbf{s}_{diff}$
  - *specular intensity color*  $\mathbf{s}_{spec}$
- ◆ Positionslichtquellen haben teilweise noch Dämpfungsparameter, z.B.
  - ◆ (konstante) Dämpfung
  - ◆ Lineare Dämpfung
  - ◆ Quadratische Dämpfung

### 3. *Ambiente* Beleuchtung

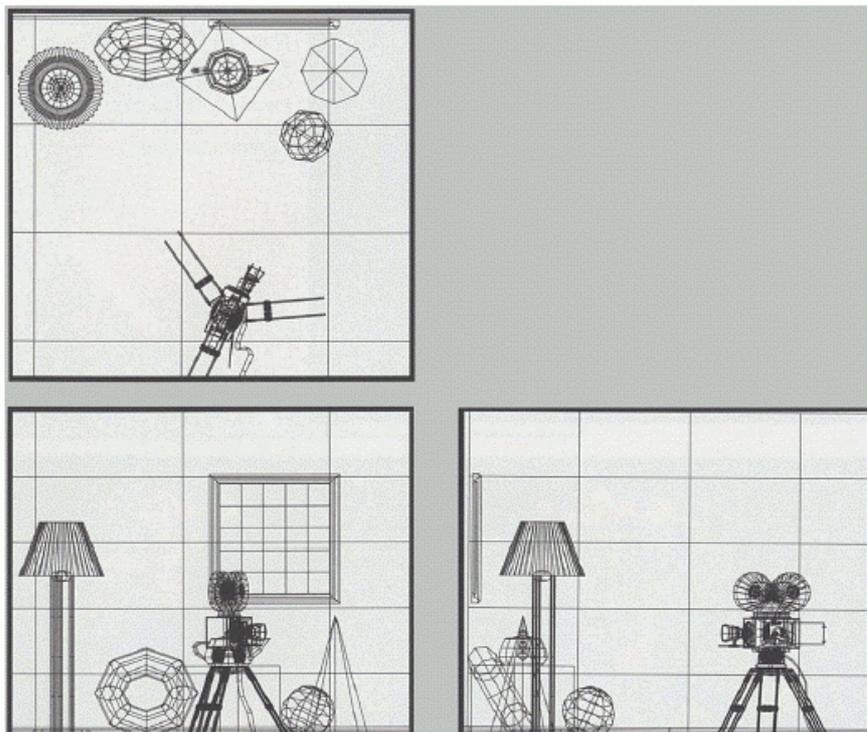
"Hintergrundlicht" – liefert einfachstes Beleuchtungsmodell

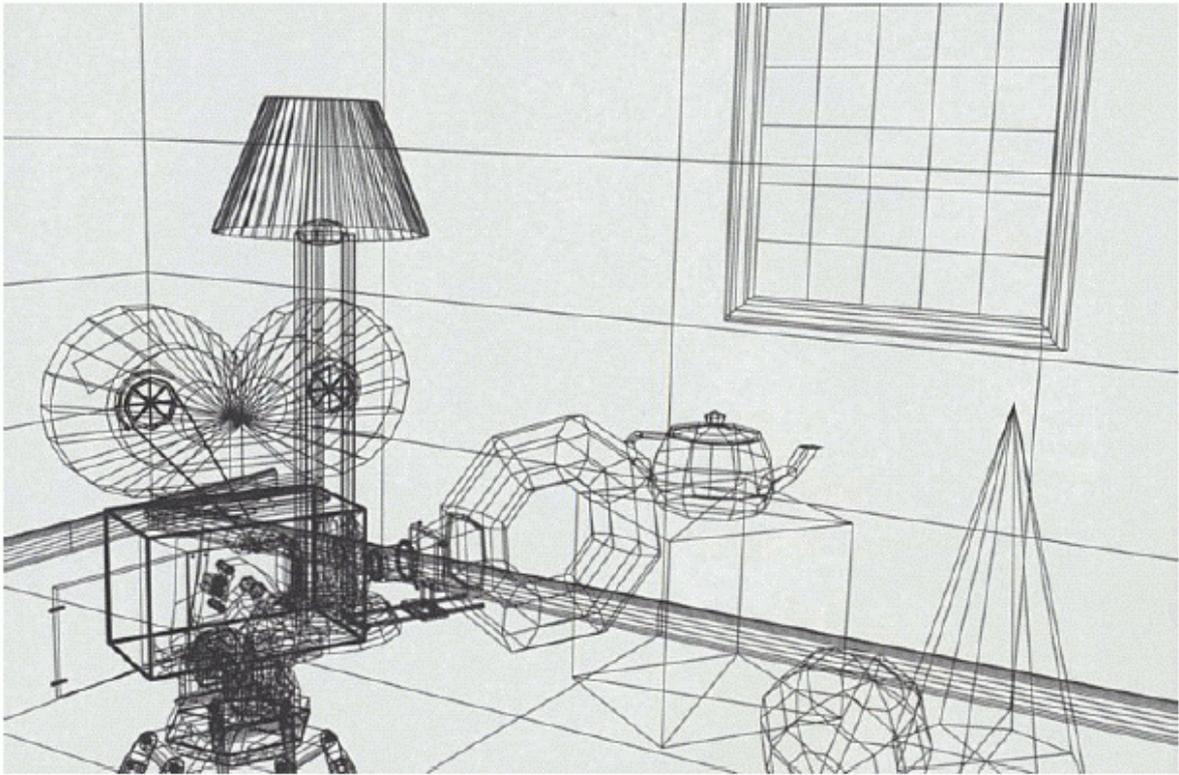
physikalische Erklärung: "ambientes Licht" als Sammelterm für Beiträge von Reflexionen durch diverse Flächen im Raum, die eine "Grundhelligkeit" erzeugen  
(VRML und OpenGL erlauben einen Wert pro Lichtquelle)

jedes Objekt hat eine ihm eigene Farbe → Objekte erscheinen als einfarbige Silhouette

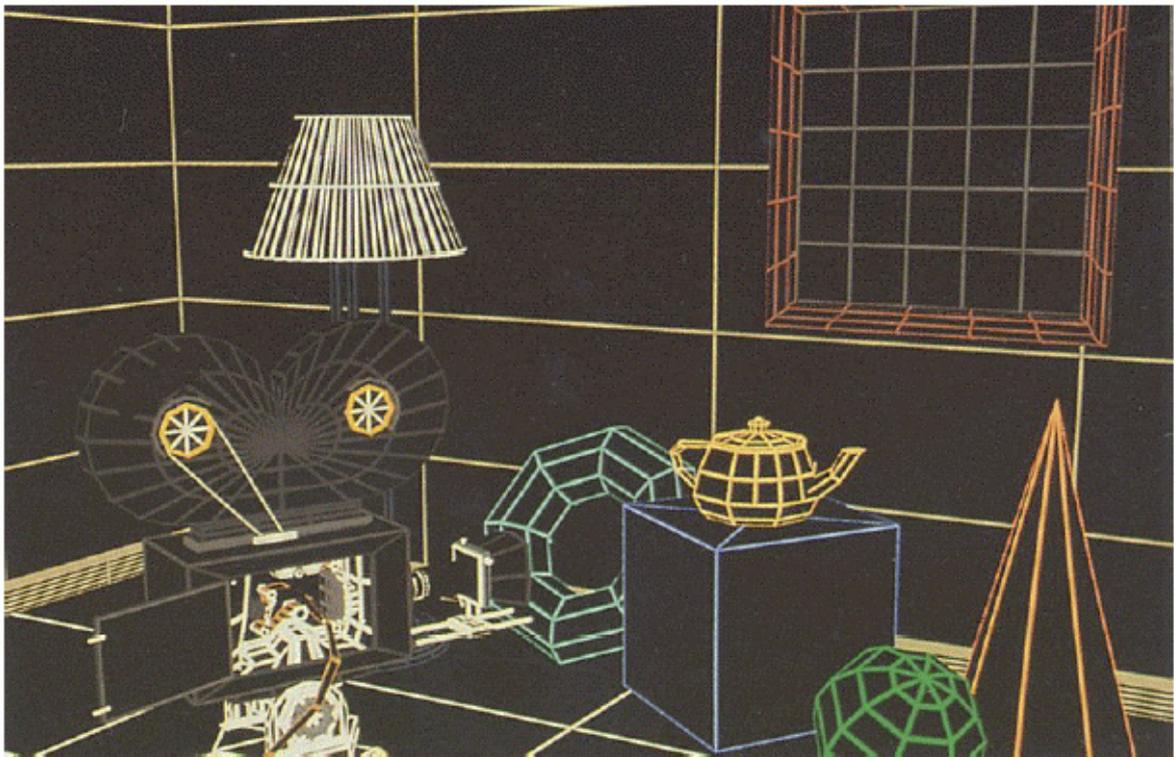
- Beleuchtungsintensität  $I$  unabhängig von einfallender Beleuchtung durch spezielle Lichtquellen oder von der Lage der Oberfläche
- eine Berechnung pro Objekt
- Beleuchtungsgleichung:  $I = I_a k_a$   
( $I_a$  ambientes Licht,  $k_a \in [0; 1]$ : ambienter Reflexionskoeffizient – Materialeigenschaft)

Begleitendes Beispiel für Beleuchtungsmodelle  
(aus Schlechtweg 2001):

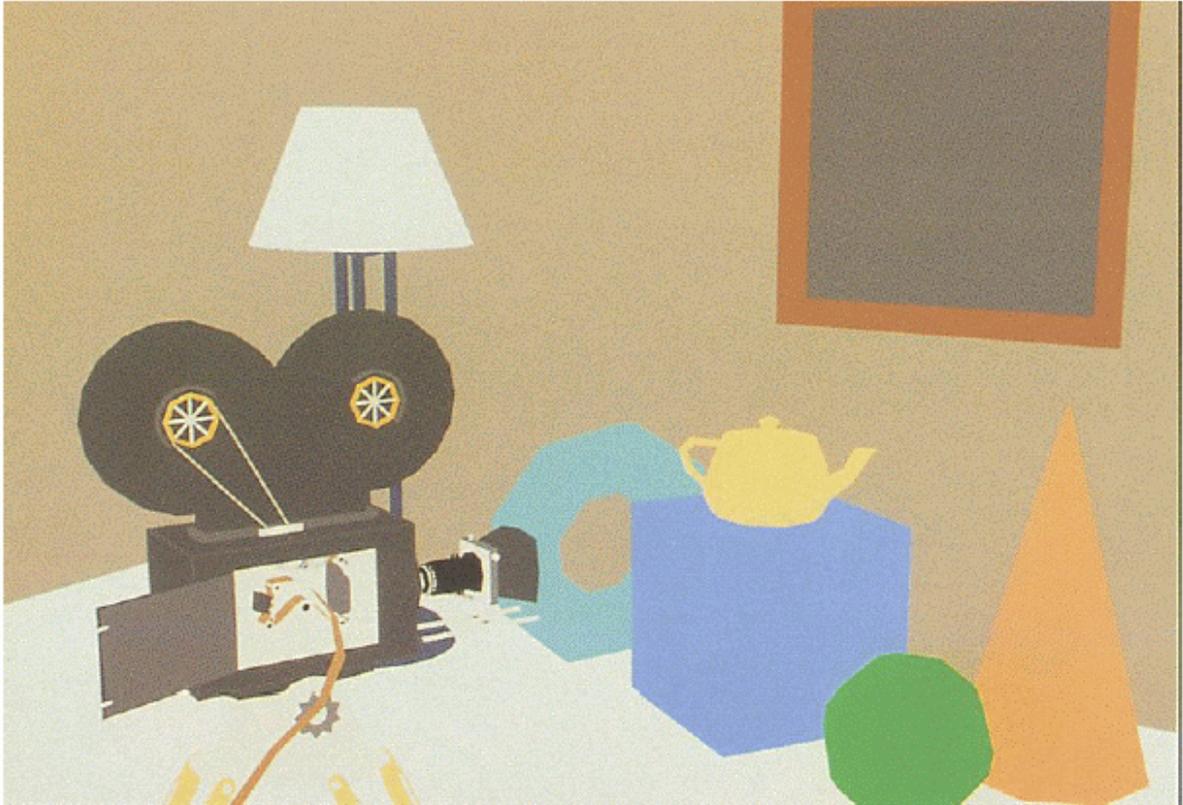




Drahtgitterdarstellung



Visible Line Determination



Ambiente Beleuchtung der sichtbaren Flächen

### Diffuse Reflektion

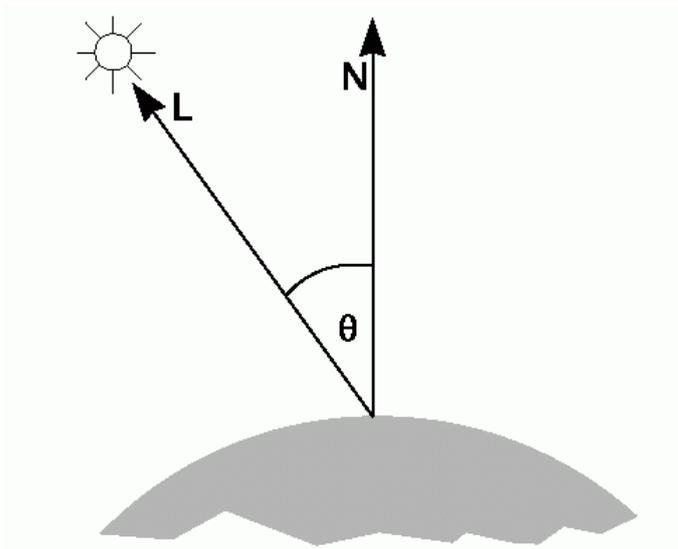
Beleuchtung durch eine Punktlichtquelle

Helligkeit variiert abhängig von

- Richtung zur Lichtquelle
- Abstand von der Lichtquelle
- atmosphärische Verhältnisse zwischen Lichtquelle und Objekt

physikalische Grundlage: Lambertsches Gesetz für Reflektion an matten Oberflächen

- Licht wird gleichmäßig in alle Richtungen reflektiert
- Helligkeit abhängig vom Winkel  $\theta$  zwischen Vektor  $L$  zur Lichtquelle und Oberflächennormalenvektor  $N$   
(großes  $\theta \Rightarrow$  flacher Lichteinfall  $\Rightarrow$  kleinere Lichtmenge pro Flächeneinheit  $\Rightarrow$  geringere Intensität der diffusen Reflektion)



Warum ist die Intensität unabhängig vom Betrachterstandpunkt?

- Gesetz von Lambert: Menge an Licht, die von einem infinitesimal kleinen Flächenstück zum Betrachter ausgesandt wird, ist direkt proportional zum Kosinus des Winkels zwischen Betrachtungsrichtung und Oberflächennormale
- Aber: Größe der sichtbaren Fläche *invers* proportional zum selben Kosinuswert  
 $\Rightarrow$  beide Faktoren gleichen sich aus  
 $\Rightarrow$  für Lambertsche Oberflächen ist Reflektion unabhängig vom Betrachterstandpunkt

## Diffuse Reflektion

- Beleuchtungsgleichung:

$$I = I_i k_d \cos \theta$$

mit  $I_i$  als Intensität der Punktlichtquelle

- $k_d \in [0, 1]$ : *diffuser Reflexionskoeffizient*
- Winkel  $\theta$  zwischen  $0^\circ$  und  $90^\circ$   
(sicherstellen mit  $I = I_i k_d \max(\cos \theta, 0)$ )
- wenn sowohl  $N$  als auch  $L$  normalisiert sind, dann:

$$I = I_i k_d (N \cdot L)$$

Die Beleuchtungsberechnung muss in Weltkoordinaten (d.h. im Objektraum) durchgeführt werden, denn bei der Transformation in den Bildraum (Normalisierungstransformation und perspektivische Abbildung) wird der Winkel  $\theta$  verändert!

- Kombination mit ambienter Beleuchtung von oben ergibt dann

$$I = I_a k_a + I_i k_d (N \cdot L)$$

- Energie von einer Punktlichtquelle nimmt mit der Entfernung ab – *light source attenuation*

$$I = I_a k_a + f_{\text{att}} I_i k_d (N \cdot L)$$

- normalerweise quadratische Abnahme aber dann kein realistisches Verhalten (Es gibt nur sehr wenige Punktlichtquellen)
- Berechnung:  $f_{\text{att}} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1\right)$  mit  $c_1$ ,  $c_2$  und  $c_3$  entsprechend gewählte Konstanten

## Farbiges Licht und Material:

- spezielle Gleichungen für die Farbkomponenten
- diffuse Farbe des Objektes: Faktoren  $O_{dR}$ ,  $O_{dG}$  und  $O_{dB}$
- Licht der Lichtquelle aus drei Komponenten:  $I_{iR}$ ,  $I_{iG}$  und  $I_{iB}$

$$I_R = I_{aR}k_a O_{dR} + f_{\text{att}}I_{iR}k_d O_{dR}(N \cdot L)$$

$$I_G = I_{aG}k_a O_{dG} + f_{\text{att}}I_{iG}k_d O_{dG}(N \cdot L)$$

$$I_B = I_{aB}k_a O_{dB} + f_{\text{att}}I_{iB}k_d O_{dB}(N \cdot L)$$

- Koeffizienten  $k_a$  und  $k_d$  zum Mischen der Anteile
- allgemeiner: Abhängigkeit von der Wellenlänge  $\lambda$

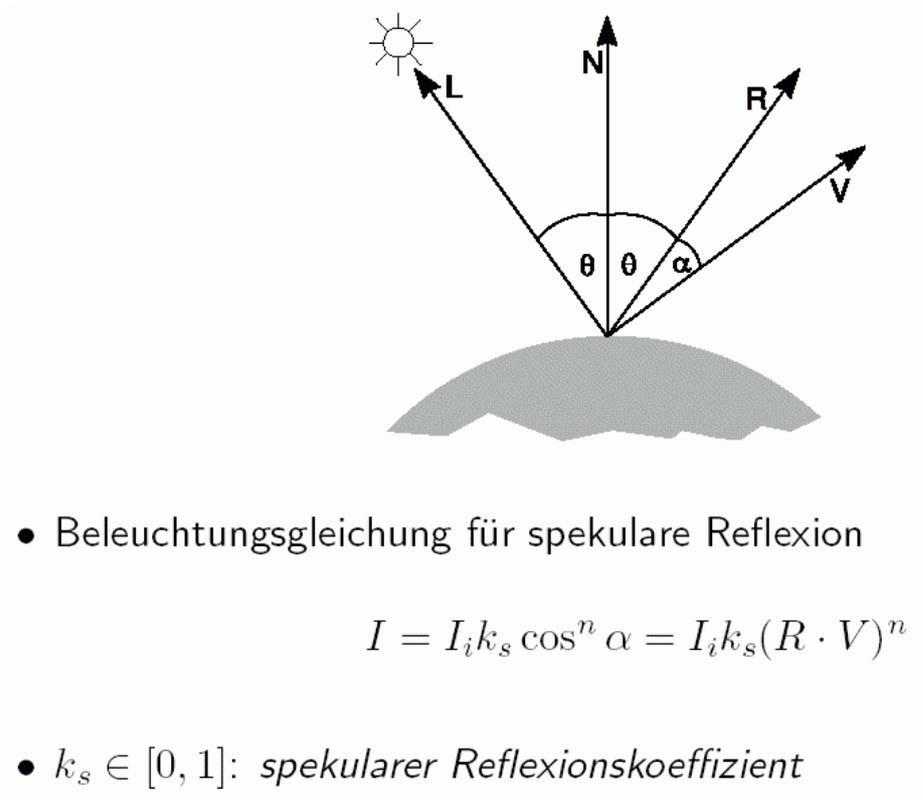
$$I_\lambda = I_{a\lambda}k_a O_{d\lambda} + f_{\text{att}}I_{i\lambda}k_d O_{d\lambda}(N \cdot L)$$

## Spekulare (spiegelnde) Reflektion

- beobachtbar an allen glänzenden Oberflächen: es entstehen "Highlights".
- spekulär reflektiertes Licht hat eher die Farbe der Lichtquelle als die des Objektes (abhängig vom Material).
- glänzende Flächen reflektieren Licht *nicht* gleichmäßig in alle Richtungen  $\Rightarrow$  Abhängigkeit vom Betrachterstandpunkt.
- Bei perfekten Spiegeln wird Licht *nur* in die Reflektionsrichtung  $R$  abgestrahlt.

## Spiegelnde Reflektion nach Phong

(nur die Richtung  $R$  ist physikalisch basiert, sonst rein empirisch-heuristisches Modell, physikalisch nicht erklärbar! – Erzeugt die gewünschten Highlights):



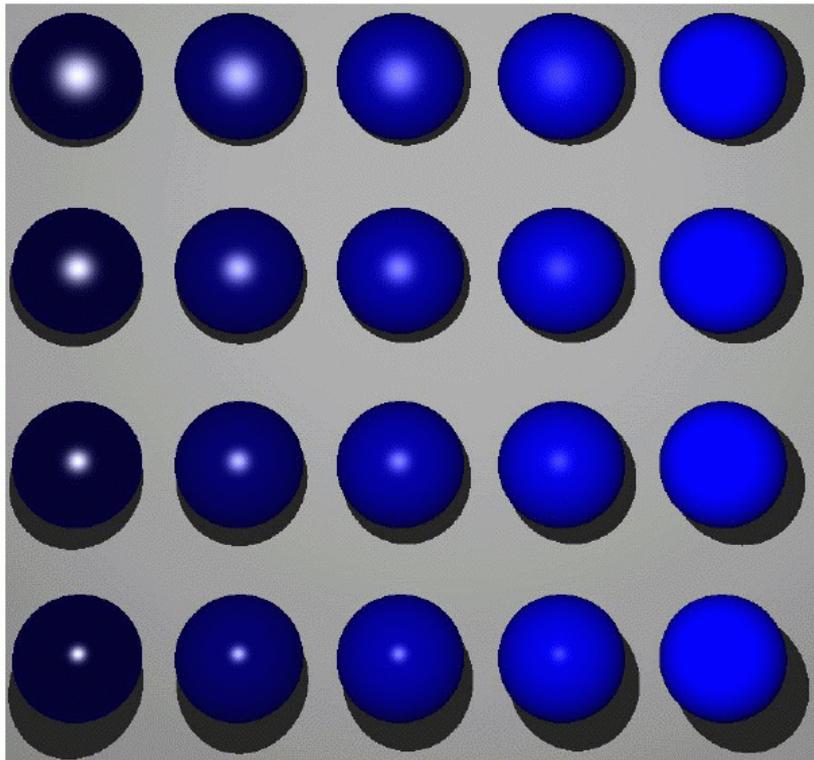
- der Exponent  $n$  heißt *shininess*

- in der Richtung  $R$  (Ausfallswinkel = Einfallswinkel) ist die Abstrahlungsintensität maximal (dann ist das Skalarprodukt bzw. der Kosinus = 1)

- $R$  lässt sich vektoriell berechnen als

$$R = 2(N \cdot L)N - L.$$

- Farbe der spekularen Komponente ist *nicht* abhängig von der Materialfarbe
- maximale spekulare Reflexion bei  $\alpha = 0$
- fällt ab mit steigendem  $\alpha$ , Stärke des Abfallens bestimmt durch  $n$  – spekularer Reflexionsexponent
- für  $n = 1$  breites, weiches Abfallen, großes Highlight mit sanftem Rand
- größere Werte für  $n \rightarrow$  schärfer abgegrenzte, kleinere Highlights
- perfekter Spiegel:  $n = \infty$



$k_s$  von links nach rechts: 1.0 – 0.75 – 0.5 – 0.25 – 0.0.  $n$  von oben nach unten: 5 – 10 – 20 – 40.

## Beleuchtungsmodell nach Phong

- Summe der drei Komponenten:

$$I = I_a k_a + f_{\text{att}} I_i k_d (N \cdot L) + f_{\text{att}} I_i k_s (R \cdot V)^n$$

- Abhängigkeit von Wellenlänge und Material:

$$I_\lambda = I_a k_a O_{d\lambda} + f_{\text{att}} I_{i\lambda} (k_d O_{d\lambda} (N \cdot L) + k_{s\lambda} O_{s\lambda} (R \cdot V)^n)$$

- Bui-Tuong PHONG: „Illumination for Computer Generated Pictures“. In: *Communication of the ACM*, Juni 1975, S. 311–317

## Mehrere Lichtquellen

- Summe über die Terme für jede Lichtquelle:

$$I_\lambda = I_a k_a O_{d\lambda} + \sum_{j=1}^m f_{\text{att}_j} I_{i\lambda_j} (k_d O_{d\lambda} (N \cdot L_j) + k_{s\lambda} O_{s\lambda} (R_j \cdot V)^n)$$

- Problem:  $I_\lambda$  kann größer als 1 werden
- Lösungen:
  - wenn  $I_\lambda$  größer 1  $\rightarrow I_\lambda := 1$
  - Division durch den höchsten Wert von  $I_\lambda$  im gesamten Bild
  - wenn Berechnung von  $I_\lambda$  vor der Anzeige  $\rightarrow$  Bildverarbeitungsoperatoren

## Shading-Modelle

Durch das Rastern können wir bis jetzt nur einfarbige Polygone darstellen

*Problem:* Bei Polygonnetzen, die gekrümmte Oberflächen approximieren, sichtbare Kanten!

Ansatz für Freiformflächen:

"brute force": Berechnen der Oberflächennormale für jeden Punkt der Oberfläche, bestimmen der Farbe / Beleuchtungsinteraktion an jedem Punkt

*Problem:* extrem aufwändig

⇒ Vorgehensweise:

- Verwendung polygonaler Modelle
- Bestimmung der Farbe so, dass gleichzeitig Glättung der polygonalen Struktur erfolgt

3 wichtige Verfahren:

- Flat Shading
- Gouraud-Shading
- Phong-Shading

### *Flat Shading*

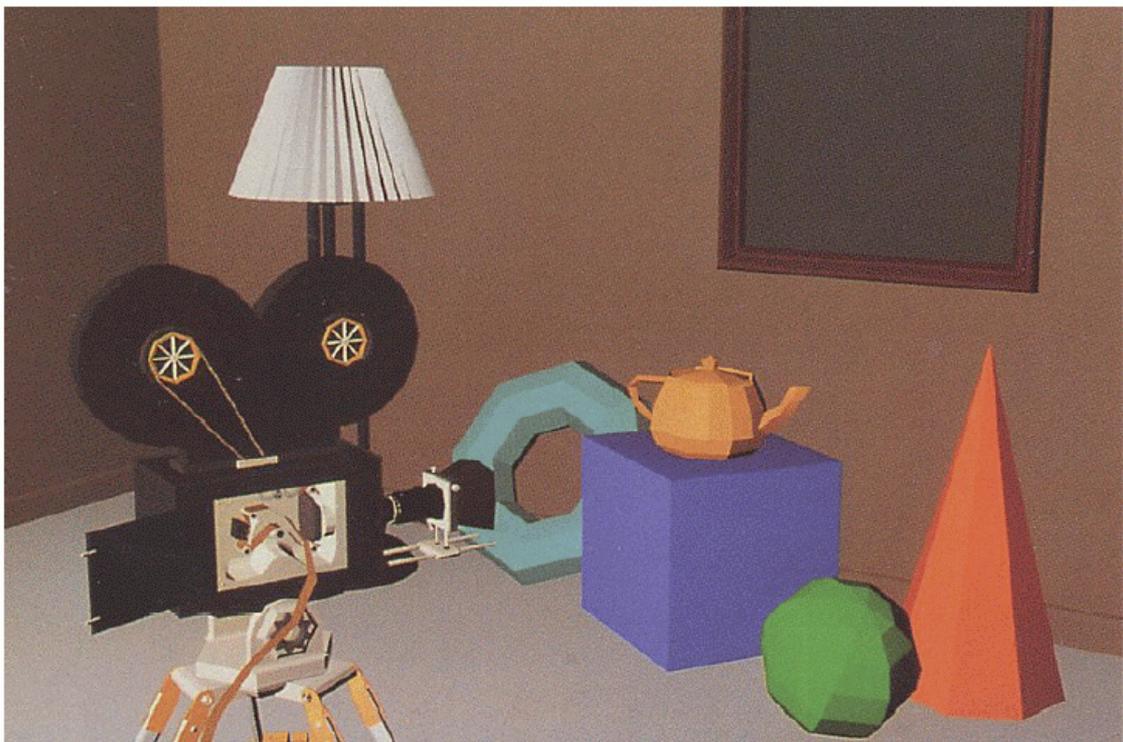
auch: *constant shading*

- Anwendung des Beleuchtungsmodells auf einen Punkt eines Polygons und Darstellen des gesamten Polygons mit der berechneten Farbe / Intensität

Glättungsproblem wird dadurch nicht gelöst

korrekt anwendbar, wenn:

- Lichtquellen im Unendlichen, so dass der Winkel zwischen Oberflächennormale und Beleuchtungsrichtung konstant ist für das Polygon
- Betrachter im Unendlichen, so dass der Winkel zwischen Haupt-Abstrahlrichtung und Sichtlinie konstant ist für das Polygon
- Polygon repräsentiert die wirkliche Objektoberfläche, keine Approximation einer gekrümmten Fläche



Flat Shading (ambiente und diffuse Komponente)

## Interpolierende Shading-Verfahren

- Wylie, Romney, Evans & Erdahl 1967: Beleuchtungswerte sind an den Ecken eines Dreiecks gegeben, werden über die Dreiecksfläche interpoliert.
- Gouraud 1971: Verallgemeinerung für beliebige Polygone.
- besonders gut nutzbar in Scanlinien-Algorithmen, die sowieso schon z-Werte interpolieren müssen
- interpolierende Shading-Verfahren sind *nicht* physikalisch korrekt!
- wichtiges Qualitätskriterium: Artefakte, die durch Approximation gekrümmter Flächen durch Polygonnetze entstehen, sollen vermindert werden.

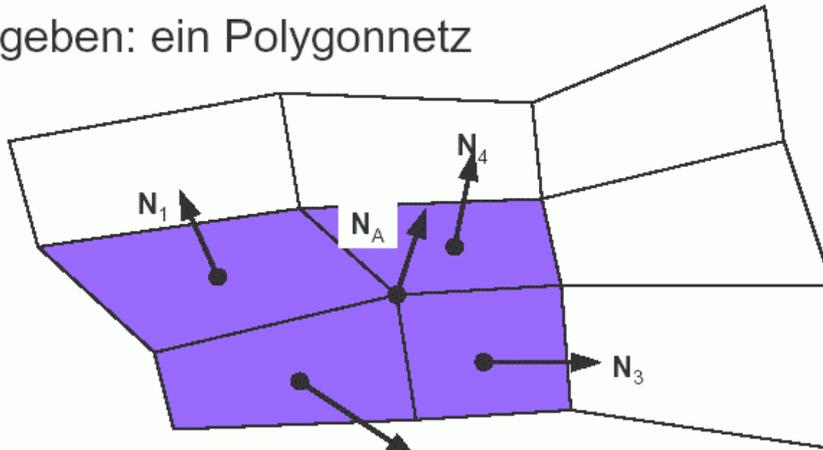
### Gouraud-Shading (1971):

- Shading durch Intensitätsinterpolation oder Farbinterpolation
- Intensitäts-/Farbwerte an Eckpunkten des Polygongitters berechnen
- Normale  $N_v$  an diesen Eckpunkten muß bekannt sein
  - übernehmen von einer analytischen Oberflächenbeschreibung (wenn Polygongitter eine Approximation ist)
  - berechnen aus den Flächennormalen  $N_i, 1 \leq i \leq n$  der angrenzenden  $n$  Flächen

$$N_v = \frac{\sum_{i=1}^n N_i}{\left\| \sum_{i=1}^n N_i \right\|}$$

- Berechnen der Intensität mit diesen Normalen

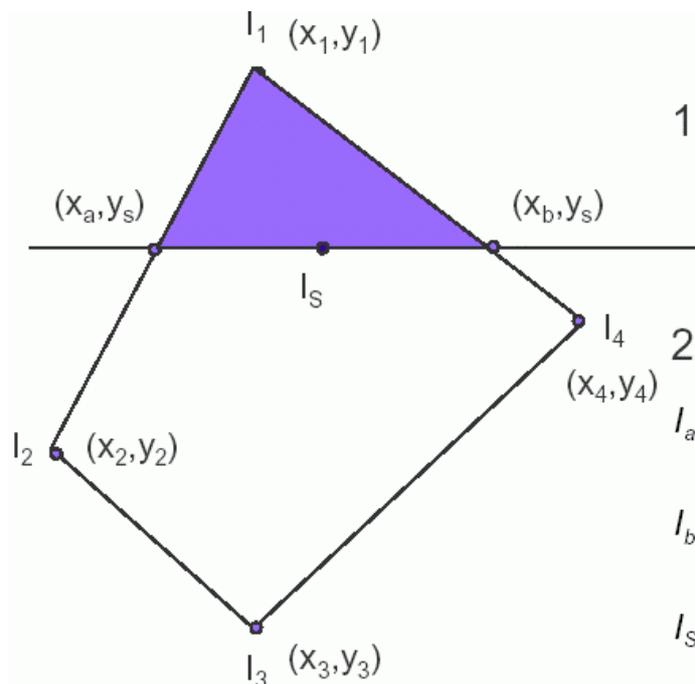
◆ Gegeben: ein Polygonnetz



◆ Berechnung der Vertex-Normale

$$\mathbf{N}_A = \frac{\mathbf{N}_1 + \mathbf{N}_2 + \mathbf{N}_3 + \mathbf{N}_4}{4}$$

Das eigentliche Shading erfolgt durch lineare Interpolation der Intensitäten zuerst entlang der Kanten, dann entlang der Scanlinien (zwischen den Kanten).



1. Schritt: Berechne Intensitäten  $I$  (RGB) an den Eckpunkten

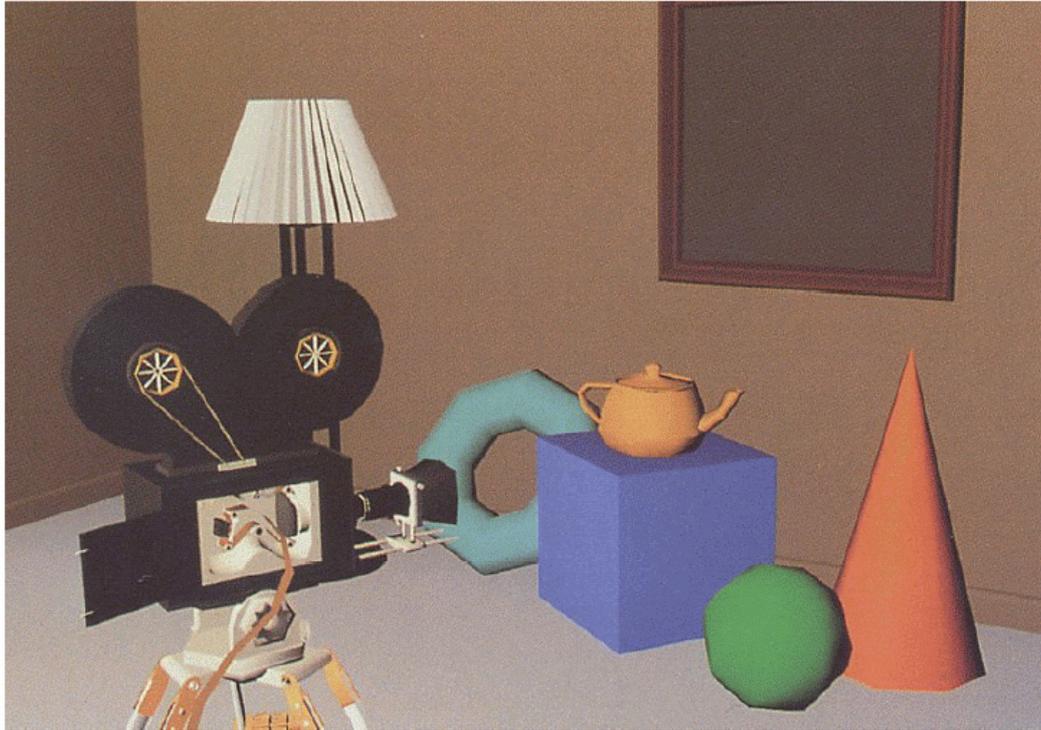
2. Interpoliere

$$I_a = \frac{1}{y_1 - y_2} [I_1(y_S - y_2) + I_2(y_1 - y_S)]$$

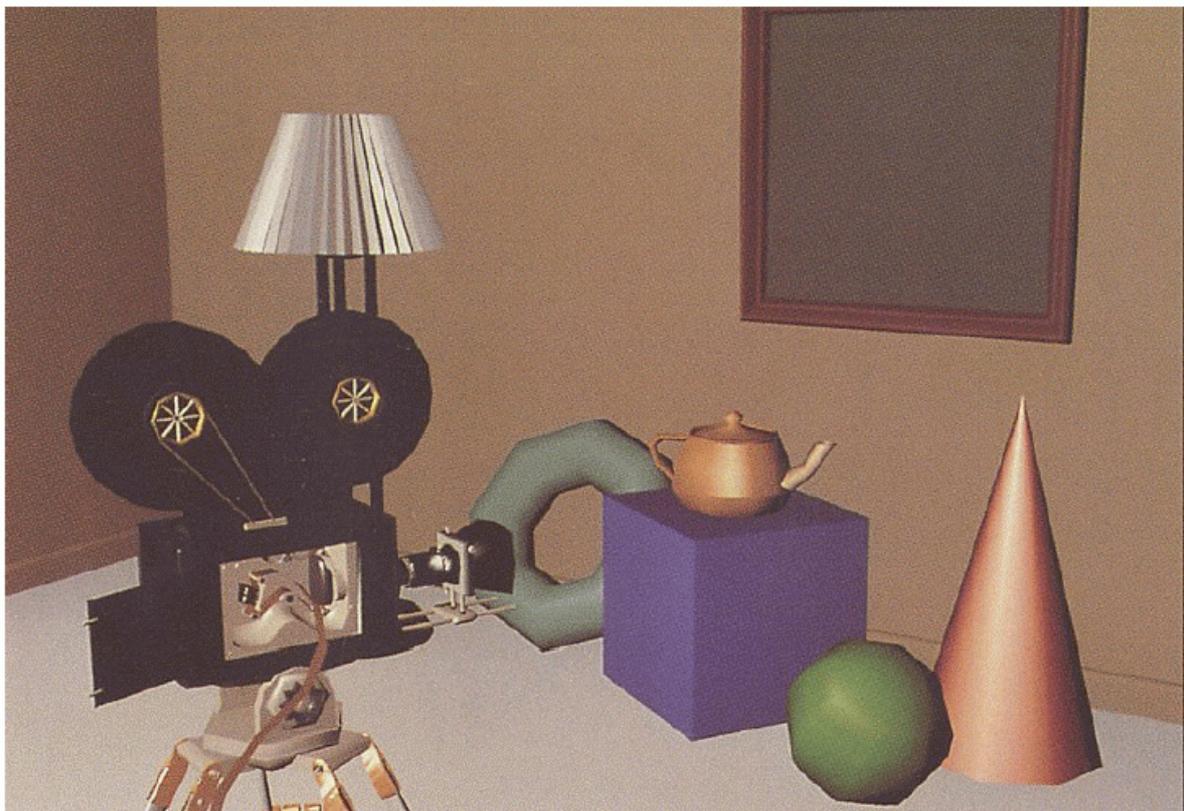
$$I_b = \frac{1}{y_1 - y_4} [I_1(y_S - y_4) + I_4(y_1 - y_S)]$$

$$I_S = \frac{1}{y_b - y_a} [I_a(x_b - x_S) + I_b(x_S - x_a)]$$

Korrektur: In der untersten Formel muss im Nenner " $x_b - x_a$ " stehen!



Gouraud-Shading (ambiente und diffuse Komponente)

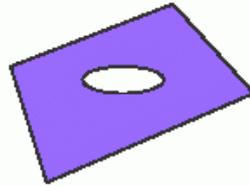


Gouraud-Shading (ambiente, diffuse und spekulare Komponente)

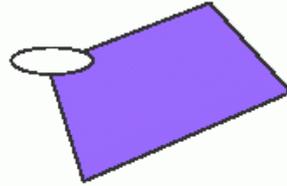
## Probleme des Gouraud Shading

**Highlight (Glanzlicht)**

wird nicht errechnet

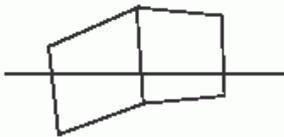


wird errechnet



Nur für diffuse  
Reflektion brauchbar

## Machbanding



Effekt: trotz Interpolation sind die  
Kanten manchmal deutlich zu  
erkennen

Ursache: Entlang einer Scanline  
ist die 1. Ableitung **nicht**  
konstant

Auge erkennt diese Diskontinuität  
2. Ordnung → Machbands

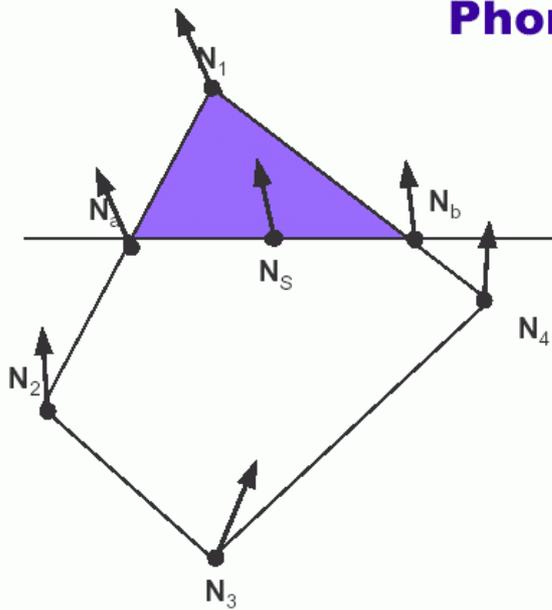
(vgl. 2. Kapitel, Wahrnehmungs-Besonderheiten)

Abhilfe:

Interpolation der Normalenvektoren anstatt der Intensitätswerte  
Intensitätswerte werden dann berechnet mit den normalisierten  
interpolierten Normalenvektoren

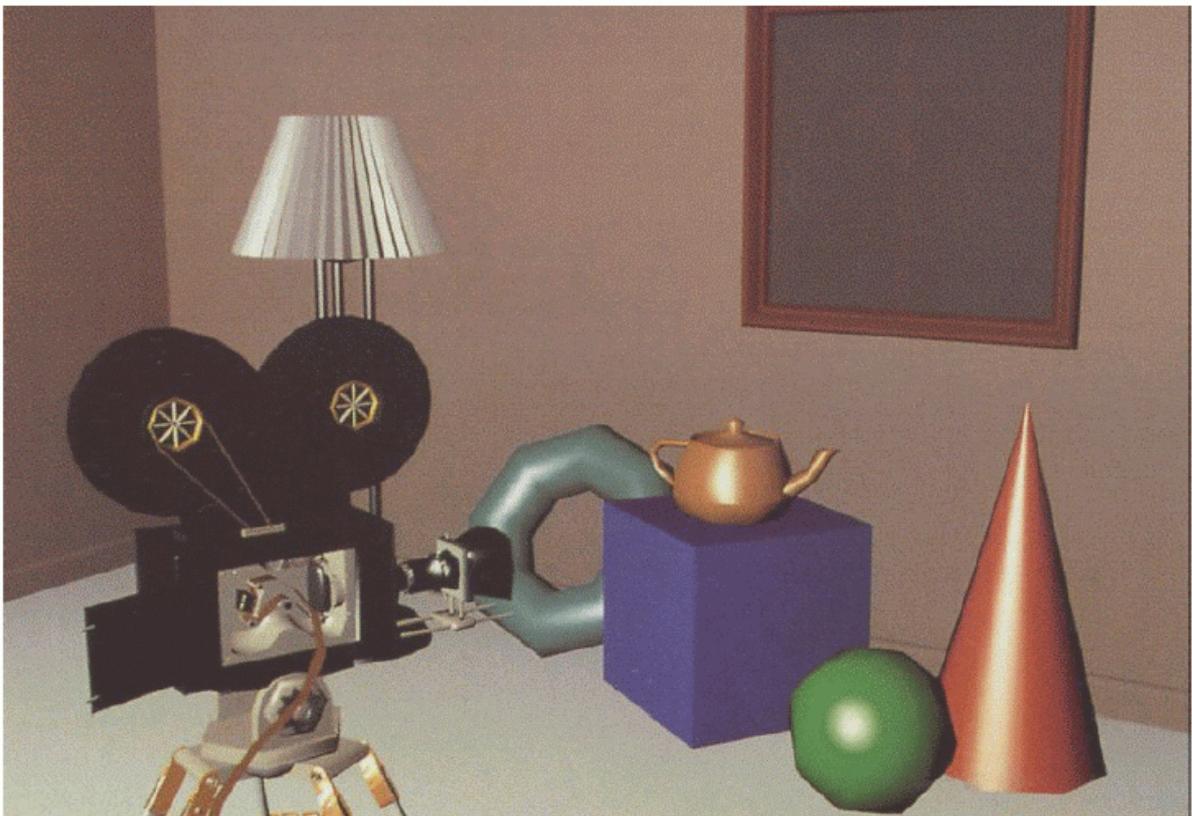
→ *Phong-Shading*

## Phong Shading (1975)



- ◆ Löst das Highlight-Problem
- ◆ Reduziert (stark) das Machbanding-Problem
- ◆ Interpoliert **Normalen**
- Vektoren müssen interpoliert werden
- Beleuchtungsgleichung muß für jedes Pixel gerechnet werden

- Ergebnisse besser als bei Gouraud-Shading
- aber: wesentlich aufwändiger (Interpolation von jeweils 3 Komponenten; sehr häufige Intensitätsberechnung)



Phong-Shading

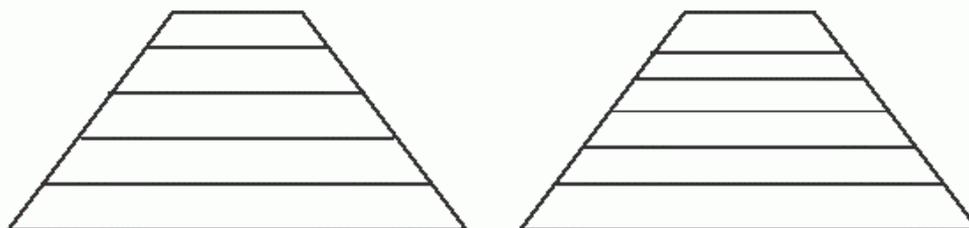
## Phong Shading (Speed-up Techniken)

- ◆ Beleuchtungsgleichung nur für jedes 2. Pixel rechnen → linear interpolieren
- ◆ Numerische Optimierungen:
  - Duff (79) Interpolation und Beleuchtung in einen Ausdruck bringen, dann Forwärts-Differenzen nutzen
  - Bishop (86) ähnlich, aber nutzt Taylorreihe
- ◆ Geometrische Optimierungen  
Versuchen abzuschätzen, ob ein Highlight auftreten kann  
Bergman 86 und Harrison 88

Generelle Probleme der Interpolationsverfahren (Gouraud, Phong):

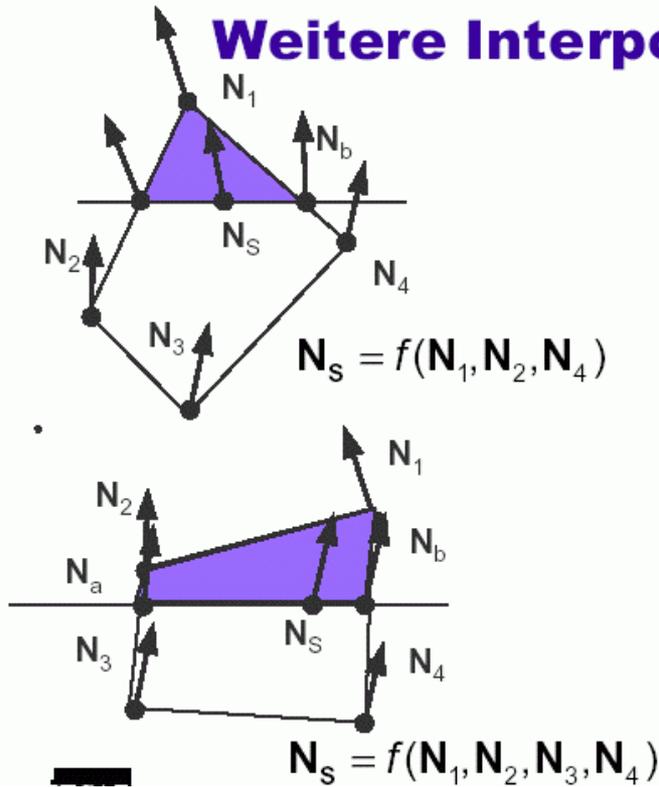
### Offene Probleme

- ◆ kantige Silhouetten der Objekte
- ◆ Interpolationsprobleme  
Interpolation in Bildschirmkoordinaten entspricht nicht der perspektivischen Verkürzung



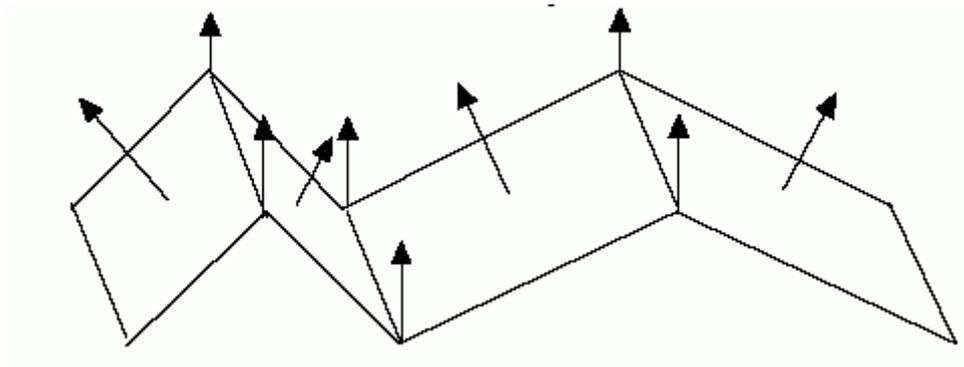
Lösungsmöglichkeit: Unterteilung in kleinere Polygone (mit allen dabei auftretenden Problemen!)

## Weitere Interpolationsprobleme



Der gleiche Polygonpunkt  $N_S$  wird, obwohl das Polygon nur in seiner Ebene gedreht wurde, von unterschiedlichen Eckpunkten beeinflusst

- nicht repräsentative Normalen an den Eckpunkten: die aus den Flächennormalen gemittelten Normalenvektoren an den Eckpunkten können die Geometrie falsch repräsentieren



Fazit:

- Glättung der polygonalen Modelle möglich, aber nur mit vielen Interpolationsproblemen
- trotzdem sind Gouraud- und Phong-Shading die Standardverfahren

## Zusammenfassung zu Shading-Modellen

Die Schattierung (*Shading*) umfasst sowohl die Auswahl eines geeigneten *Beleuchtungsmodells* als auch die Entscheidung, in welchen Bildpunkten das jeweilige Modell explizit zur Anwendung gelangen soll und in welchen Punkten *Interpolationsmethoden* (billiger) eingesetzt werden. Letztere sind erforderlich, da die explizite Berechnung des aus dem Beleuchtungsmodell resultierenden Intensitäts- bzw. Farbwertes für jedes Pixel insbes. bei interaktiven Echtzeitanwendungen zu aufwändig wäre.

- Varianten des Phong-Shading sind die "Arbeitspferde" in der hochwertigen Computergrafik
- heute häufig in Hardware realisiert!

nicht verwechseln:

- *Shading*: verbunden mit geometrischer Glättung (Interpolation)
  - Gouraud: Farbwerte, Phong: Normalen
- Reflektion, Beleuchtungsgleichung: Berechnung der Farbintensitäten
  - Gouraud: i.allg. nur diffuser + ambienter Anteil;
  - Phong: spiegelnder + diffuser + ambienter Anteil.

## Transparenz

Manche Objekte sind (teilweise) durchsichtig

⇒ spezielle Probleme (Lichtbrechung, -abschwächung...)

Einfache Ansätze ohne Berücksichtigung der Brechung:

### 1. *Screen-Door-Transparenz*

Das transparente Polygon wird mit einem Schachbrettmuster überzogen; ein Pixel wird voll transparent, das nächste wieder undurchsichtig (opak) gerendert.

Vorteile:

- einfach
- keine Tiefensortierung notwendig

Nachteile:

- Transparenzgrad 50 % (schwierig variierbar)
- Artefakte bei mehreren transparenten Objekten hintereinander

### 2. *Interpolierte Transparenz (Alpha-Blending)*

Beim Rasterisieren, wird jedem Pixel nicht nur ein Farb- und z-Wert, sondern auch ein  $\alpha$ -Wert zugeordnet:  $0 \leq \alpha \leq 1$ .

$\alpha = 1$  entspricht vollständig opakem Objekt

$\alpha = 0$  vollständig transparentes Objekt

Jedes Pixel, das von einem transparenten Objekt bedeckt wird, wird als Farbe

$$\mathbf{c}_{new} = \alpha \cdot \mathbf{c}_{ren} + (1 - \alpha) \cdot \mathbf{c}_{old}$$

zugeordnet, wobei  $c_{ren}$  die Farbe des transparenten Objekts und  $c_{old}$  die Pixelfarbe vor dem Blending ist.

Das transparente Objekt wird *über* die schon gerenderten Objekte geschrieben.

Alpha-Blending erfordert Tiefensortierung der transparenten Objekte (back-to-front-Sortierung).

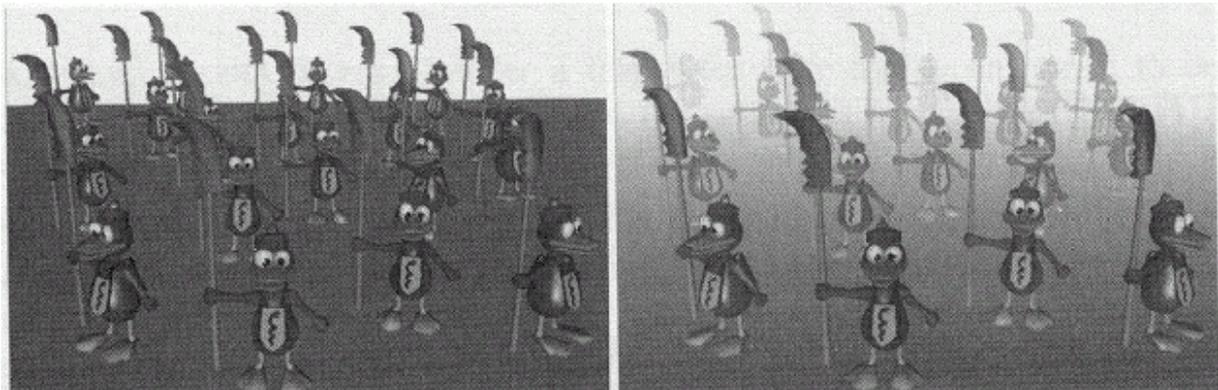
## Nebel

Nebel (oder Dunst) kann relativ einfach simuliert werden, auch in Echtzeitsystemen

Er kann für viele Zwecke sinnvoll eingesetzt werden:

- Realismus von Landschaftsszenen (*outdoor scenes*) erhöhen
- Verbesserung der Tiefenschätzung / des 3D-Effekts
- Nebel verdeckt Artefakte in der Nähe der hinteren Clipping-Ebene (*far clipping plane*)

Nebel wird inzwischen oft von der Grafik-Hardware unterstützt.



(aus Krömker 2001)

Ähnlich wie bei der Transparenz, werden Nebeleffekte auch beim Rasterisieren realisiert, indem die Farbe  $c_{ren}$  eines Pixel mit der Farbe  $c_{fog}$  des Nebels gemischt wird:

$$\mathbf{c}_{new} = f\mathbf{c}_{ren} + (1-f)\mathbf{c}_{fog}$$

wobei  $f$  eine Funktion des Abstands vom Beobachter mit Werten in  $[0, 1]$  ist.

Einfach gesagt: Tiefe im Bild ändert die Farbe von Pixeln.

## Arten von Nebel:

- **Linearer Nebel:**

$$f(z) = \frac{z_{\max} - z}{z_{\max} - z_{\min}}$$

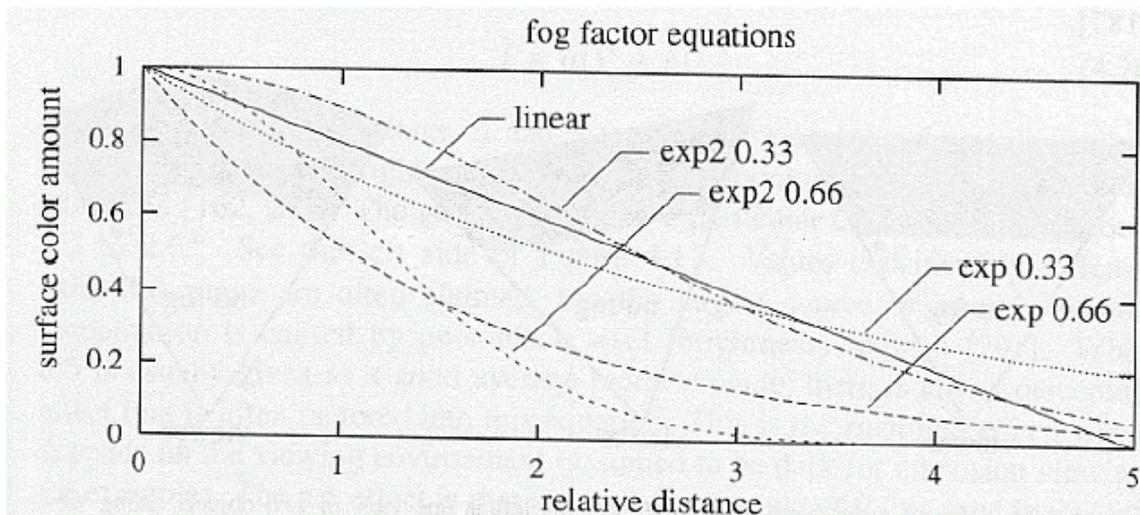
- **Exponentieller Nebel:**

wobei  $d_f$  die Dichte des Nebels bestimmt.

$$f(z) = e^{-d_f z}$$

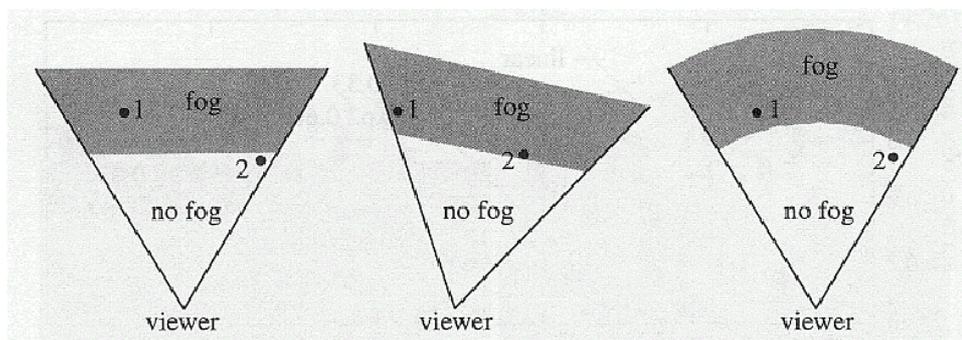
- **Quadrierter exponentieller Nebel:**

$$f(z) = e^{-(d_f z)^2}$$



## Implementierungsprobleme beim Nebel:

- die  $z$ -Werte verhalten sich nach der perspektivischen Transformation (in den kanonischen Sichtkörper) nicht mehr linear  $\Rightarrow$  Nebel-effekt müsste eigentlich im Objektraum berechnet werden
- Häufig wird der Abstand nicht für jedes Pixel ermittelt, sondern es wird einfach der Abstand entlang der zentralen Sichtgeraden verwendet. Das führt zu Artefakten. *Radialer Nebel* (Euclidean Distance Fog) vermeidet dieses Problem.



- Nebel auf Pixel-Ebene wird zur Zeit von Grafik-Hardware i.allg. nicht unterstützt, stattdessen auf Vertex-Ebene, d.h. nur an den Polygon-Eckpunkten berechnet und dann interpoliert.

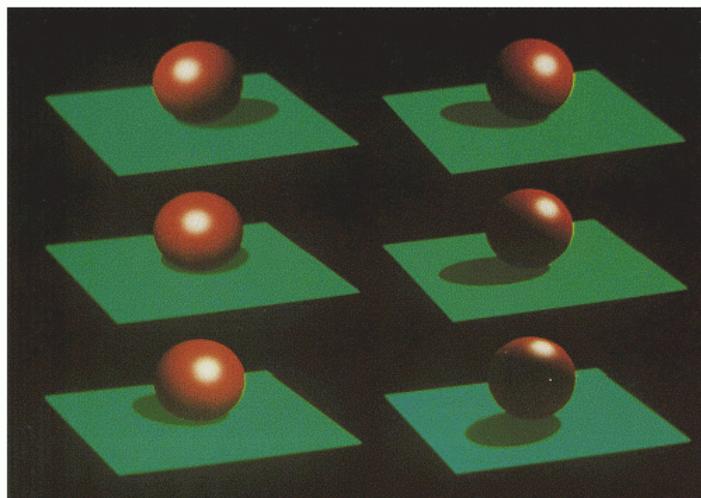
### Schattenberechnung

hier geht es um Beschattung von Objekten durch andere  
– zu unterscheiden von *shading* (Schattierung)!

wichtiges Element des Fotorealismus!

### Funktionen des Schattens

- „verankert“ Objekte in der Szene (keine „fliegenden“ Objekte)
- hebt die Richtung der Beleuchtung hervor



- Steigerung des Realismus des Bildes
- Schatten geben Hinweise zum Aufbau der Szene
- Verbesserung der Tiefenwahrnehmung
- verwendbar für Simulationen (Energie-, Wärmeverteilung)

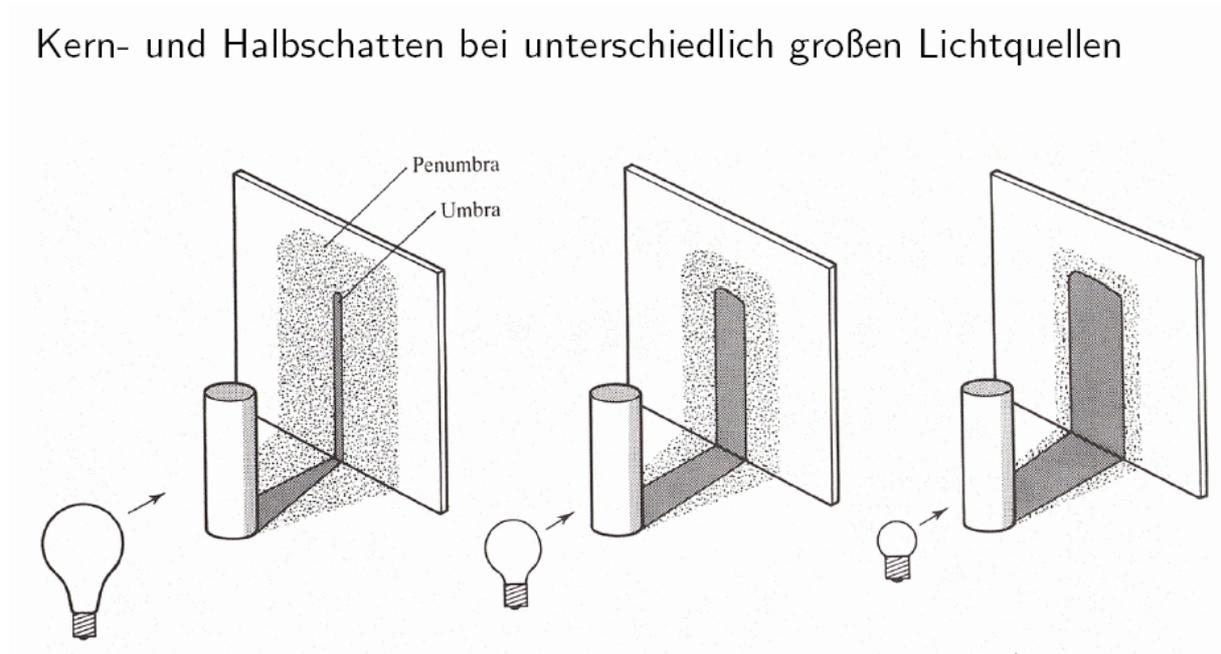
## Kern- und Halbschatten

Schatten bestehen i.allg. aus 2 Teilen: Kernschatten (*umbra*) und Halbschatten (*penumbra*)

- Kernschatten: zentraler, scharf abgegrenzter dunkler Teil; hier kommt kein Licht der betrachteten Lichtquelle an, die Lichtquelle ist von hier aus nicht sichtbar.
- Halbschatten: helleres Gebiet, das den Kernschatten umgibt; von hier aus ist die Lichtquelle *partiell* sichtbar.

- Punktlichtquellen generieren nur Kernschatten
- flächige Lichtquellen erzeugen beides

Halbschatten aufwändig zu berechnen; da in der Computergrafik i.d. Regel Punktlichtquellen eingesetzt werden, *häufig nur Kernschattenberechnung*.



## Position der Lichtquelle

- Komplexität der Schattenberechnung abhängig von Position der Lichtquelle
- keine Schatten, wenn (einzige) Lichtquelle im Betrachterstandpunkt
- Lichtquelle im Unendlichen: orthographische Projektion zur Bestimmung des Schattens → einfachster Fall
- Lichtquelle an endlicher Position außerhalb des Gesichtsfeldes: perspektivische Projektion zur Bestimmung des Schattens → etwas komplizierter
- Lichtquelle innerhalb des Gesichtsfeldes: Unterteilung der Szene in Sektoren, Schattenberechnung separat für die Sektoren → kompliziertester Fall
- Schatten ändern sich bei Animation

## Schatten und Beleuchtung

- HSR/VSD-Algorithmen bestimmen, welche Flächen vom Betrachterstandpunkt aus sichtbar sind.
- Schattenberechnung bedeutet, die Flächen zu bestimmen, die von der Position der Lichtquelle aus nicht sichtbar sind.
- → VSD/HSR-Algorithmen können genutzt werden
- Sichtbarkeit aus Sicht der Lichtquelle: entweder sichtbar oder nicht
- wenn Oberflächenpunkt von Lichtquelle aus nicht sichtbar
  - Punkt liegt im Schatten
  - Beleuchtungsberechnung muß entsprechend angepaßt werden

- Faktor  $S_j$  bestimmt, ob Licht von Lichtquelle  $j$  ein Objekt an einem gegebenen Punkt erreicht:

$$S_j = \begin{cases} 0, & \text{wenn Lichtquelle } j \text{ blockiert ist} \\ 1, & \text{wenn Lichtquelle } j \text{ nicht blockiert ist} \end{cases}$$

- Hinzufügen zur Beleuchtungsberechnung:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + \sum_{j=1}^m S_j f_{\text{att}_j} I_{i\lambda_j} (k_d O_{d\lambda} (N \cdot L_j) + k_{s\lambda} O_{s\lambda} (R_j \cdot V)^n)$$

- Gebiete im Schatten immer noch durch das ambiente Licht beleuchtet

### *Übersicht: Verfahren der Schattenberechnung*

- Schatten auf einer einzelnen Grundebene (Spezialfall)
- Algorithmen zur Schattenberechnung im allgemeinen Fall:
  - Scanlinien-Schatten-Berechnung
  - Schattenvolumen-Ansatz
  - Hidden Surface Removal (HSR) aus Sicht der Lichtquelle
  - Schatten-z-Buffer
  - Raytracing
  - Radiosity

## Das Raytracing-Verfahren

"globale" Verfahren:

bei Beleuchtungsrechnung für einen Punkt wird (potenziell) die gesamte Szene mit einbezogen

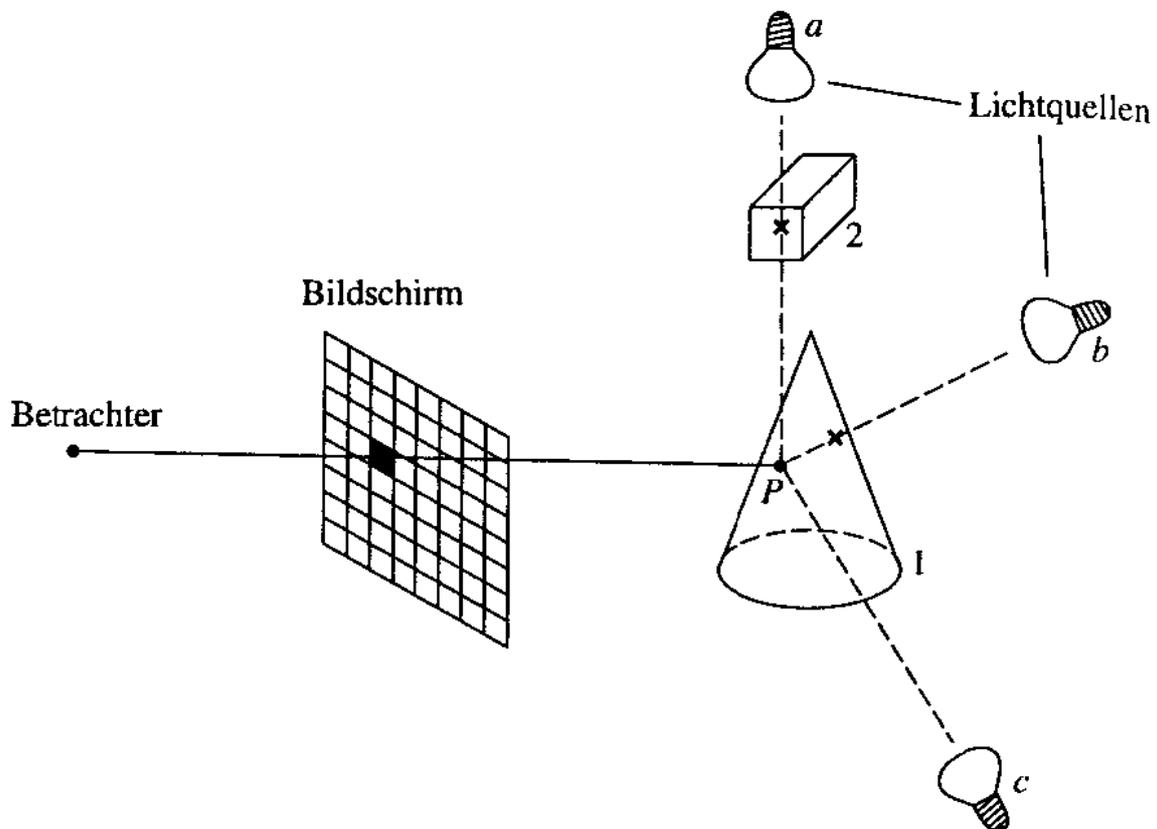
- Einsatz bei angestrebtem hohem Grad an Fotorealismus
- angezeigt bei Szenen mit nicht vernachlässigbarem Anteil an Mehrfachreflexionen
- Versuch, die physikalischen Vorgänge zu modellieren

## Raytracing

"Strahlverfolgung"; genauer eigentlich: *visibility tracing*

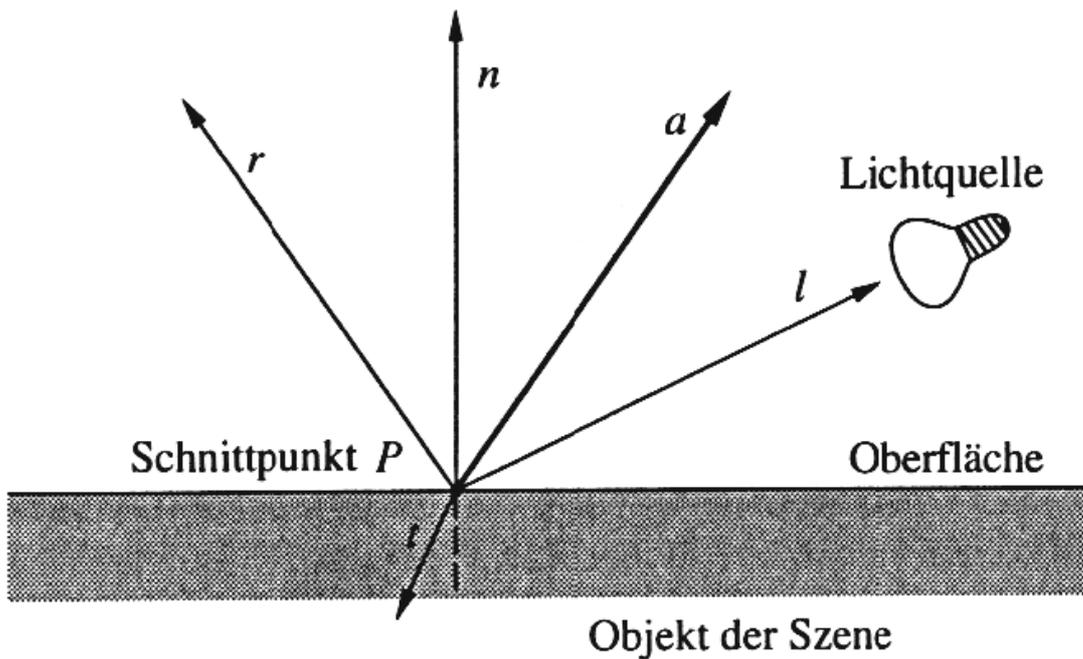
- simuliert den Prozess der Lichtausbreitung und arbeitet dabei nach den strahlenoptischen Gesetzen für ideale Spiegelung und Lichtbrechung
- vor allem für Szenen mit hohem Anteil spiegelnder und transparenter Flächen
- *Grundidee*: Lichtstrahlen auf Weg von der Quelle bis zum Auge / zur virtuellen Kamera verfolgen
- Da nur wenige Strahlen das Auge erreichen, *kehrt man das Verfahren um* (Ausnutzung der Reziprozität der Reflektion) und sendet durch jedes Pixel des Bildschirms einen vom Augpunkt ausgehenden Strahl in die Szene
- Vereinfachung: beim konventionellen Raytracing werden *nur ideal reflektierte und ideal gebrochene* Strahlen weiterverfolgt.
- Trifft der Sehstrahl auf ein Objekt, so wird dort das lokale Beleuchtungsmodell berechnet. Anschließend werden zwei neue Strahlen erzeugt, nämlich der reflektierte und der gebrochene (transmittierte) Sehstrahl. Der Leuchtdichtebeitrag dieser beiden Strahlen wird rekursiv berechnet.

- Dieser Prozess bricht ab, wenn
  - eine Lichtquelle getroffen wird,
  - die auf dem Strahl transportierte Energie zu gering wird,
  - der Sehstrahl die Szene verlässt.
- Aus praktischen Gründen setzt man a priori eine Obergrenze für die Rekursionstiefe.
- Das Verfahren wird zugleich auch für die *Schattenberechnung* benutzt: Dazu sendet man von den Auftreffpunkten des Sehstrahls sogenannte *Schattenstrahlen* zu den Lichtquellen der Szene. Nur wenn kein undurchsichtiges Objekt zwischen einer Lichtquelle und dem Auftreffpunkt liegt, trägt sie zur (direkten) Beleuchtung des betreffenden Punktes bei.
- Das Verfahren erledigt auch die Sichtbarkeitsrechnung!



*Schattenberechnung beim Raytracing* (aus Bungartz et al. 1996). Der Schattenstrahl von P zu a schneidet Objekt 2, der von P zu b schneidet Objekt 1 ein weiteres Mal (Austritt), der Schattenstrahl von P zu c erreicht diese Lichtquelle dagegen ungehindert.

Wir haben also 3 (oder mehr) *sekundäre Strahlen*, die in jedem Auftreffpunkt erzeugt werden:



Primärstrahl  $a$  vom Betrachter (bzw. hier – als Vektor – zum Betrachter hin gerichtet)

Sekundärstrahl  $l$  zur Lichtquelle (Schattenstrahl)

Sekundärstrahl  $r$  (perfekt reflektierter Strahl)

Sekundärstrahl  $t$  (gebrochener Strahl)

Entsprechende Erweiterung des Phong-Beleuchtungsmodells durch Whitted (1980):

zum lokalen Beleuchtungsanteil nach Phong kommen noch der ideal spiegelnde und der ideal transmittierte Anteil hinzu.

nun:

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

wobei:

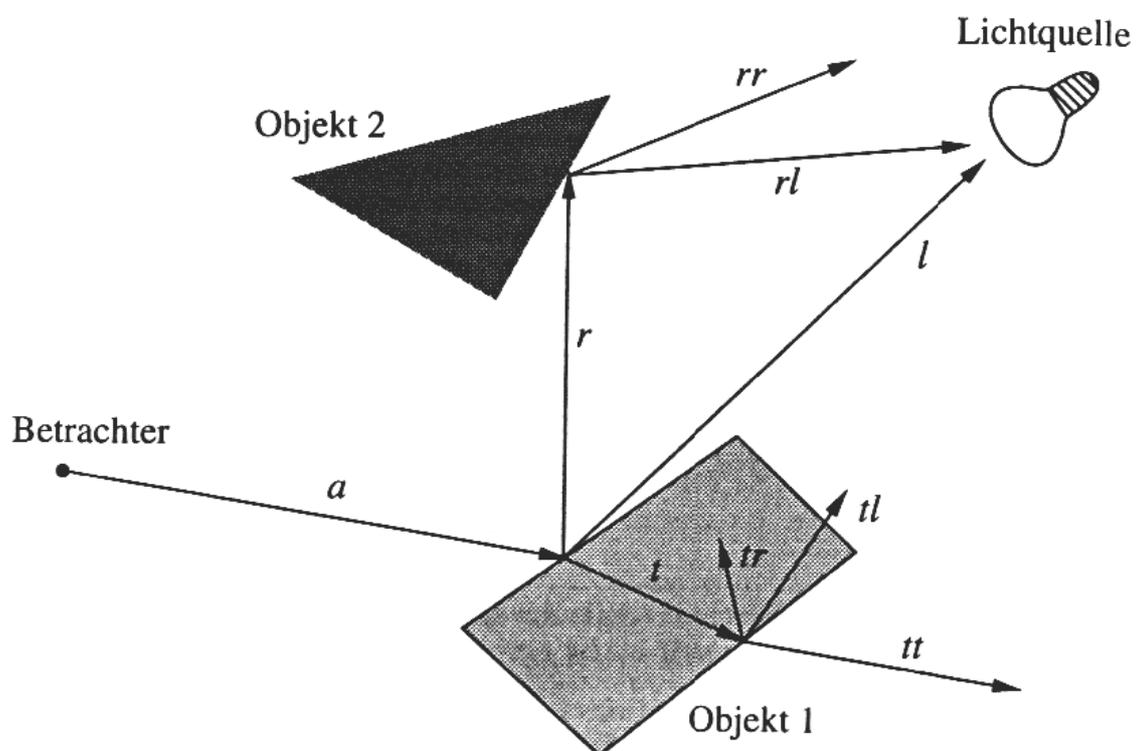
$I_t$  Intensität durch Transmission

$I_r$  Intensität durch Reflektion

Berechnung der Richtungen der Sekundärstrahlen:

- für reflektierten Strahl nach idealem Reflexionsgesetz
- für transmittierten Strahl nach Brechungsgesetz von Snellius

Beispiel der rekursiven Strahlverfolgung, Rekursionstiefe 2:  
Objekt 1 wird als transparent angenommen, Objekt 2 ist lichtundurchlässig



( $a$ : Primärstrahl,  $r$ : reflektierter Strahl,  $t$ : transmittierter Strahl,  $l$ : Schattenstrahl)

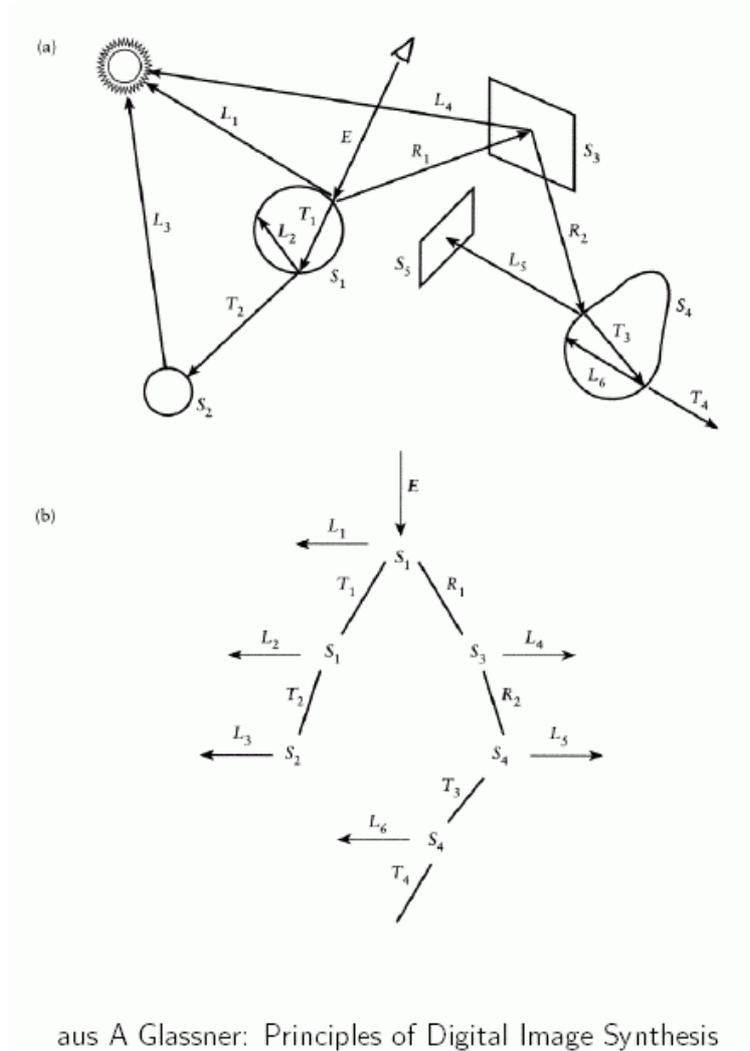
grobe Struktur des rekursiven Algorithmus:

für jedes Pixel {

1. bestimme nächstliegenden Schnittpunkt des entspr. Sehstrahls mit einem Objekt der Szene
2. berechne ideal reflektierten Lichtstrahl
3. berechne die Leuchtdichte aus dieser Richtung
4. berechne ideal gebrochenen Lichtstrahl
5. berechne die Leuchtdichte aus dieser Richtung
6. berechne Schattenstrahl(en) zu den Lichtquellen
7. werte das Phong-Beleuchtungsmodell an dieser Stelle aus und addiere die gewichteten Leuchtdichten des reflektierten und des gebrochenen Sekundärstrahls

}

Die Schnittpunkte bilden einen Baum:



Der Algorithmus wird manchmal zweistufig durchgeführt:  
in der 1. Phase wird der Baum mit Hilfe der  
Schnittpunktberechnungen aufgebaut, in der zweiten Phase  
wird der Baum traversiert, wobei die Intensitäten berechnet  
werden.

Als Faustregel für die maximale Rekursionstiefe gilt 5.

kritischer Teil des Algorithmus: **Schnittpunktberechnung**  
Strahl mit Objekt

Whitted stellte nach Laufzeitmessungen fest: 75 % der Zeit  
werden für die Schnittpunktberechnungen verwendet und 12 %  
für die Berechnung des Beleuchtungsmodells

typische Schnittpunktprobleme:

*Strahl – Kugel:*

Strahl in (Zweipunkte-) Parameterform (Parameter  $t$ ) gegeben  
Kugel durch Mittelpunkt  $M = (x_M, y_M, z_M)$  und Radius  $r$ :

$$(x - x_M)^2 + (y - y_M)^2 + (z - z_M)^2 = r^2$$

Einsetzen von  $x, y, z$  aus Parametergleichung des Strahls liefert  
quadratische Gleichung für  $t$ , Lösung ergibt 2 Werte

wähle kleineren der beiden Werte, dieser muss  $> 0$  sein

Einsetzen in Parametergleichung des Strahls liefert

Schnittpunktposition  $P$

Normalenvektor am Schnittpunkt:  $N = (P - M)/r$

*Strahl – Dreieck*

- bilde Ebene durch Dreieck
- schneide Strahl mit Ebene
- teste, ob Schnittpunkt innerhalb des Dreiecks liegt  
(bei Verwendung zweier Seitenvektoren als Richtungs-  
vektoren für 2-Parameterform der Ebenengleichung muss für  
die Parameter  $s, t$  gelten:  $s, t, s+t \in [0; 1]$ )

*Objekte aus mehreren Dreiecken:*

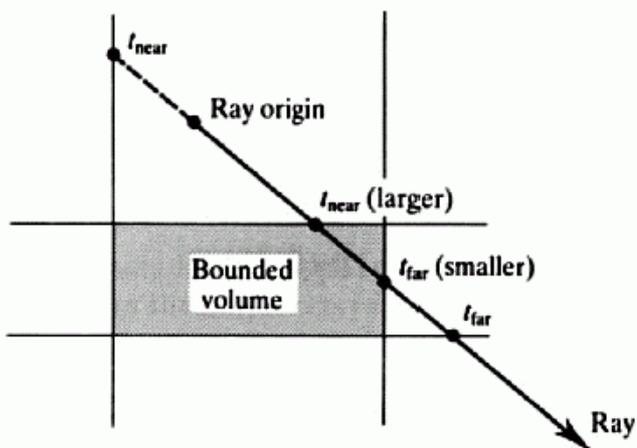
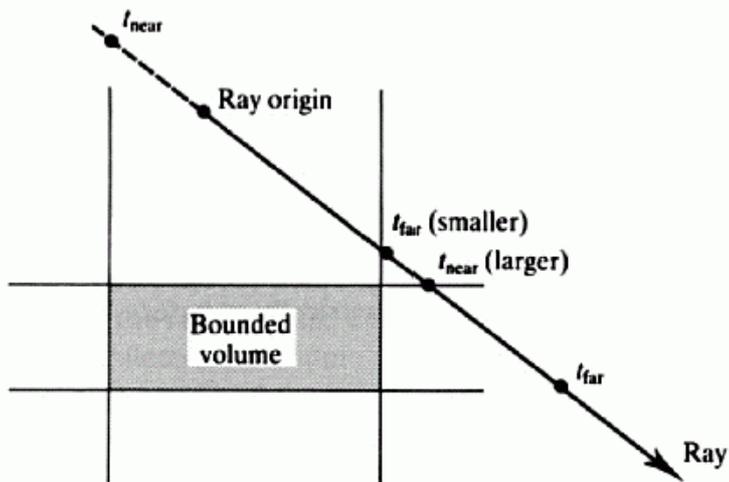
mit allen Dreiecken schneiden (level-of-detail-Ansatz sinnvoll)

## Polygonnetze: Triangulation

### Strahl – Quader

(wichtig, weil effizient berechenbar)  
insbes.: achsenparallele Quader (Boxen)  
einfacher Schnittalgorithmus:

- Bearbeite die jeweils parallelen Ebenenpaare und bestimme  $t_{near}$  und  $t_{far}$  (Abstand zu Ebene auf Strahl)
- ist der größere Wert von  $t_{near}$  größer als der kleinere von  $t_{far}$   $\Rightarrow$  kein Schnitt



aus: Watt, Watt: Advanced animation and rendering techniques

Anzahl der Schnittpunktberechnungen:

- proportional zur Anzahl der Strahlen und zur Anzahl der (Primitiv-) Objekte
- wächst exponentiell mit der Anzahl der Rekursionsstufen

zwei Verbesserungsansätze:

- effizientes Bestimmen, ob ein Objekt überhaupt geschnitten wird
- effiziente Schnittpunktbestimmung

Lösungsansätze für die Effizienzverbesserung:

- bounding-Objekte (Quader, Kugeln etc.)
- Raumaufteilung (Zellen, Octrees, BSP-Trees etc.)

Weitere Beschleunigungstechniken:

### *Adaptive Tiefenkontrolle*

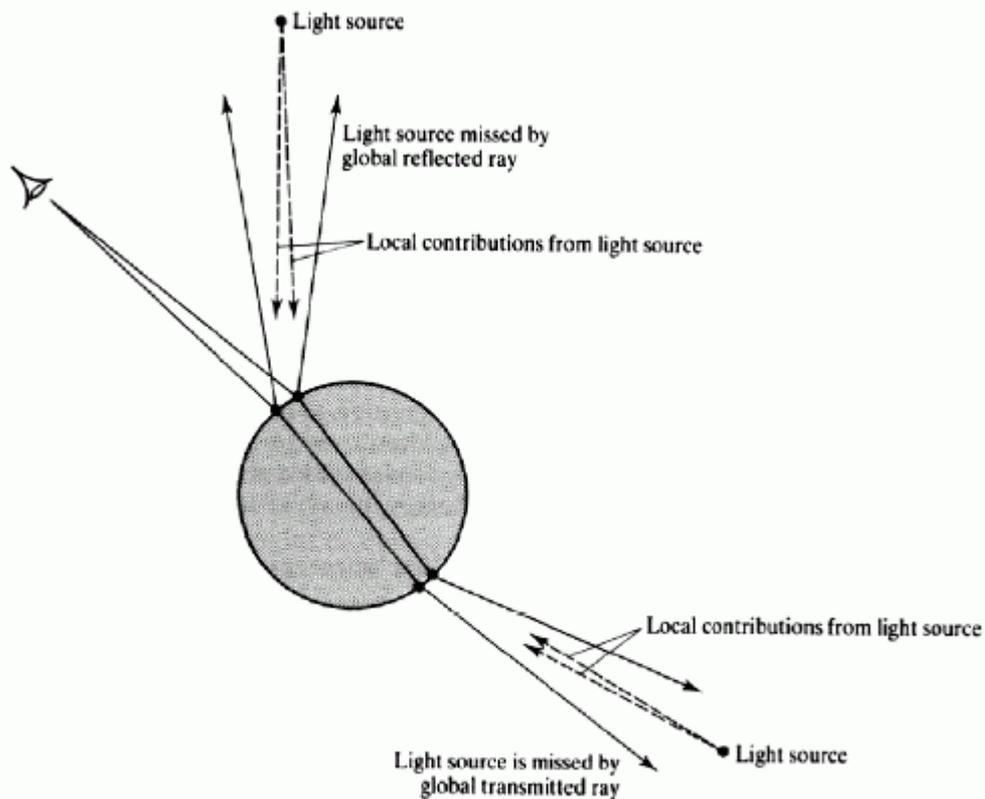
Veränderung der Rekursionstiefe in Abhängigkeit vom Material – wenn zu erwartender Beitrag eines Sekundärstrahls zur Gesamthelligkeit eines Pixels zu klein wird, wird abgebrochen

### *Mischung von Standard- (Phong-) Shading mit Raytracing*

- Raytracing nur bei spiegelnden und / oder durchsichtigen Flächen
- Spiegelungen werden als Texturen (siehe nächstes Kap.) auf die Objekte projiziert, dadurch "Zwischenspeicherung" der Ergebnisse des (unvollst.) Raytracing

## Probleme beim Standard-Raytracing

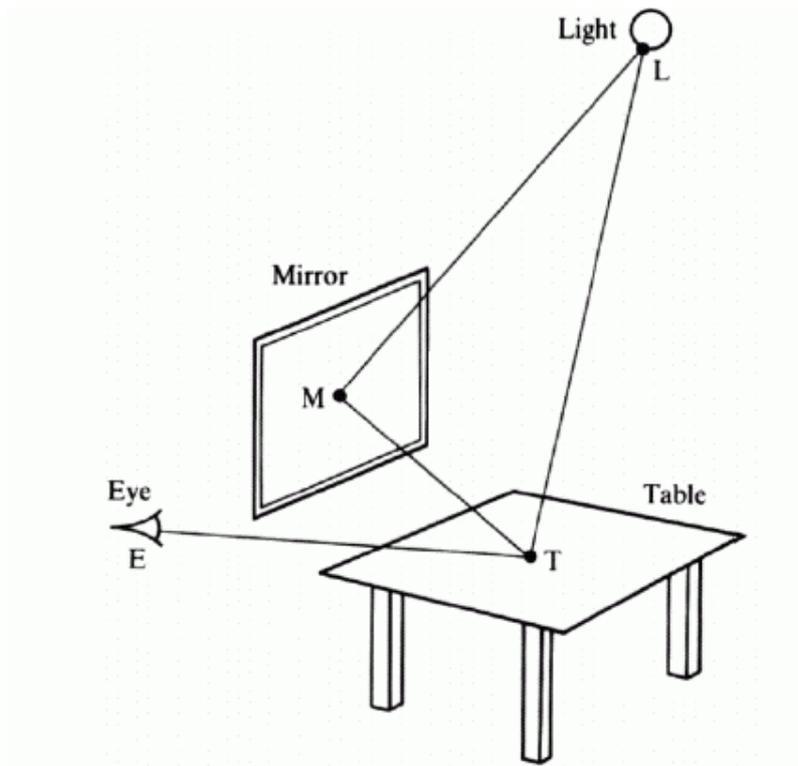
- keine Halbschatten darstellbar  
Schattenwürfe wirken unnatürlich scharf
- Reflexionen / Transmissionen sind "zu perfekt" – in der Realität gibt es in der Regel Abweichungen vom perfekten Spiegel
- mit Strahlen ist eine Punktlichtquelle oft nicht zu finden:



aus: Watt, Watt: Advanced animation and rendering techniques

- spiegelnd-diffuse Interaktionen werden nicht korrekt abgebildet

Beispiel: Raytracing erkennt den Lichtweg Lichtquelle-Spiegel-Tisch nicht, wenn der Tisch nur diffuses Material besitzt



aus: Watt, Watt: Advanced animation and rendering techniques

zur Behebung dieser Mängel wurden verschiedene Erweiterungen und Modifikationen des Standard-Raytracing vorgeschlagen

## *Bewertung des Raytracing (Fazit)*

### Vorteile:

- Szene kann beliebig komplexe Objekte enthalten – einzige Bedingung: Objektnormalen und Schnittpunkte mit Strahlen müssen berechnet werden können. Keine Notwendigkeit, alle Objekte durch Polygone zu approximieren
- Hidden Surface Removal, Schattenberechnung, Berechnung von Reflexionen und Transparenzen werden alle mit erledigt
- Explizite perspektivische Transformationen der Objekte und Clipping-Berechnungen sind nicht notwendig
- Objekte dürfen sich gegenseitig durchdringen, Schnitte zwischen Objekten brauchen nicht berechnet zu werden
- das Beleuchtungsmodell muss nur in sichtbaren Punkten ausgewertet werden

### Nachteile:

- Abtastung der Szene mit einem Strahl pro Pixel erzeugt in der Regel Aliasing (kann mit Supersampling oder stochastischem Abtasten gemildert werden)
- großer **Rechenaufwand**, da für die (zahlreichen) Schnittpunktberechnungen in der Regel Floating-Point-Arithmetik erforderlich ist; u.U. müssen viele Millionen Strahlen verfolgt werden
- Schatten haben stets scharfe Grenzen – weiche Halbschatten nur durch Erweiterungen des Standardverfahrens
- Schatten müssen bei jeder Änderung der Kameraparameter neu berechnet werden, obwohl sie nur von den Lichtquellen und den Objekten der Szene abhängen
- keine Berücksichtigung von Mehrfachreflexionen an diffusen Oberflächen (diese bildet aber signifikanten Anteil der Beleuchtung in Innenräumen)
- Szenen werden bzgl. eines Blickpunktes berechnet – ein Durchwandern von Räumen in Echtzeit ist daher (außer bei ganz einfachen Szenen) momentan noch kaum mit dieser Technik zu realisieren