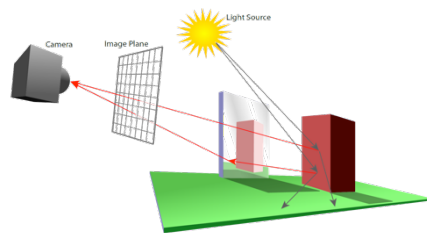




BEUTH HOCHSCHULE FÜR TECHNIK BERLIN  
University of Applied Sciences

## Computergrafik II

### Raytracing – Lokale Beleuchtung



Bachelor Medieninformatik  
Wintersemester 2011

Prof. Dr.-Ing. Hartmut Schirmacher  
<http://schirmacher.beuth-hochschule.de>  
[hschirmacher@beuth-hochschule.de](mailto:hschirmacher@beuth-hochschule.de)



#### Gliederung


- Das Phong-Beleuchtungmodell
  - Lokale vs. globale Beleuchtung
  - Überblick / Empirische Motivation
  - Diffuser Beleuchtungsterm
  - Spekularer Beleuchtungsterm
  - Ambienter Beleuchtungsterm
  - Gesamtmodell
- Implementierungshinweise

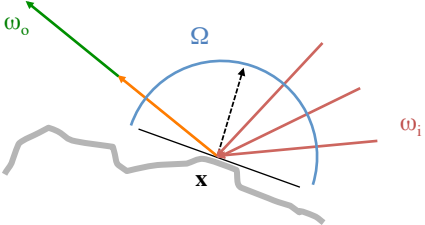


BEUTH HOCHSCHULE  
FÜR TECHNIK  
BERLIN  
University of Applied Sciences



## Rendering-Gleichung (Rendering Equation)





$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L(\mathbf{x}, \omega_i) (\omega_i \cdot \mathbf{n}) d\omega_i$$

ausfallendes  
Licht

Emission


Integral über die (Halb-) Kugel  
aller möglichen Einfallrichtungen

Abschwächung  
durch Einfallswinkel

BRDF (Verteilungsfunktion)

einfallendes Licht

## Lokale vs. globale Beleuchtung

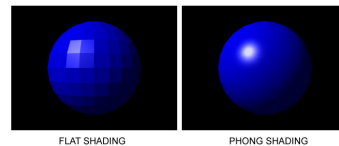


- **Lokale Beleuchtung** heutige Vorlesung
  - Interaktion zwischen Oberflächenpunkt und Lichtquelle(n)
  - Weiß nichts über andere Objekte / globale Effekte
- **Globale Beleuchtung**
  - Interaktionen zwischen allen beteiligten Objekten
    - Abschatten durch Objekte zwischen Lichtquelle und Empfänger
    - Reflexionen von Objekten auf anderen Objekten
    - Indirekte diffuse Beleuchtung
    - ...

## Das Phong-Beleuchtungsmodell (Phong Illumination)



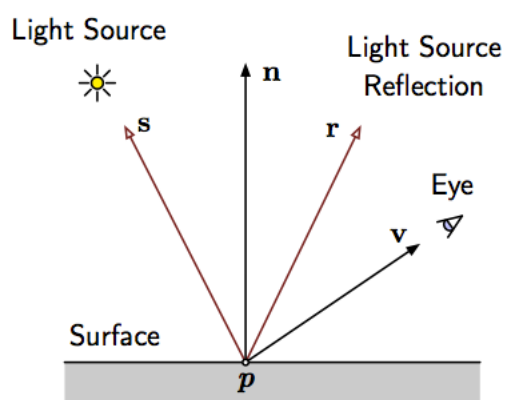
- Empirisches Beleuchtungsmodell <sup>1</sup>
  - Bui Tuong Phong 1975, durch Beobachtung glänzender Oberflächen motiviert
  - Empirisch, nicht physikalisch genau (kann zuviel Energie erzeugen)
  - Drei Komponenten
    - **ambienter** Term + **diffuser** Term + **spekularer** Term
  - Geeignet zur Beschreibung glatter, plastik-ähnlicher Materialien
  - Grundlage z.B. des OpenGL Beleuchtungsmodells (vor OpenGL 3)
- Synonyme
  - Phong-Beleuchtung, Phong Lighting, Phong Illumination
- Achtung Verwechslungsgefahr:
  - Phong *Shading*<sup>1</sup> im engeren Sinn ist ein Interpolationsverfahren



1) Bui Tuong Phong: *Illumination for Computer Generated Pictures*. *Communications of the ACM* 18, 6 (Jun. 1975): 311–317, [ISSN 0001-0782](https://doi.org/10.1145/321998.322007)



## Geometrie des Phong-Beleuchtungsmodells



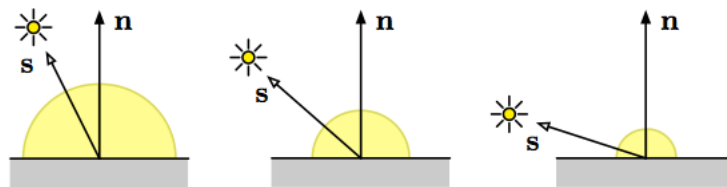
- **p**: Punkt auf der Oberfläche
- **n**: Oberflächennormale
- **s**: zur Lichtquelle (source)
- **r**: ideale Reflexionsrichtung
- **v**: zum Betrachter (view point)



## Diffuse Beleuchtung



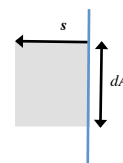
- Beispiele für diffuse Oberflächen
  - Matte Oberflächen wie Kreidetafeln, Papier, unbehandeltes Holz
- Beobachtung:
  - Einige Materialien reflektieren Licht gleichförmig in alle Richtungen
  - Die Stärke der Reflexion hängt von dem Winkel der *Lichtquelle* zur Oberfläche ab (unabhängig vom Betrachterpunkt)



## Diffuse Reflexion nach Lambert



- Einfallendes Licht aus Richtung  $s$ 
  - Senkrecht einfallendes Licht:
    - Energie trifft auf eine Fläche  $dA$

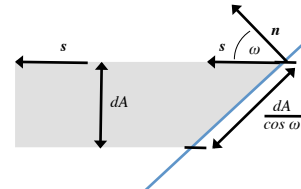


## Diffuse Reflexion nach Lambert



### ■ Einfallendes Licht aus Richtung $\mathbf{s}$

- Senkrecht einfallendes Licht:
  - Energie trifft auf eine Fläche  $dA$
- Fläche nicht senkrecht zu Einfallsrichtung:
  - Gleiche Energie trifft auf größere Fläche
  - Effektive Fläche ist  $dA / \cos \angle(\mathbf{n}, \mathbf{s})$ .
  - Eintreffendes Licht (Energie pro Fläche)  $\sim \cos \angle(\mathbf{n}, \mathbf{s})$



### ■ Reflektiertes Licht in Richtung $\mathbf{v}$

- Proportional zu einfallendem Licht
- Energie/Fläche wird in alle Richtungen gleichverteilt
- Ist somit unabhängig vom Betrachtungswinkel



## Diffuser Beleuchtungsterm des Phong-Modells



Diffus reflektiertes Licht am  
Punkt  $\mathbf{p}$  der Oberfläche

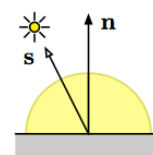
$$L_d(p) = k_d \sum_j L_j(\mathbf{n} \cdot \mathbf{s}_j)$$

Konstanter Reflexionskoeffizient

$$\mathbf{k}_d = \begin{pmatrix} r_d \\ g_d \\ b_d \end{pmatrix}, \quad r_d, g_d, b_d \in [0, 1]$$

**Annahmen:**

- Punktlicht / direktionales Licht
- $\mathbf{n}$  und  $\mathbf{s}$  normiert
- $(\mathbf{n} \cdot \mathbf{s}) > 0$  (sonst Licht auf Rückseite)



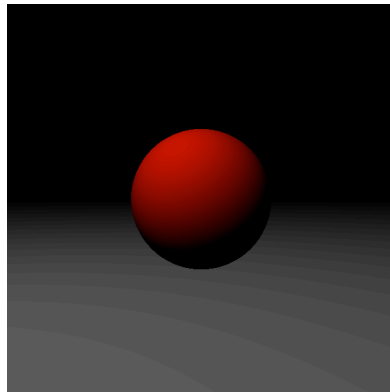
Winkel-Abschwächung des  
Lichts aus Richtung  $\mathbf{s}_j$

Von Quelle  $j$  in Punkt  $\mathbf{p}$   
eintreffende Lichtmenge

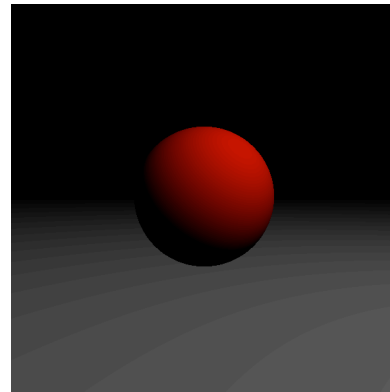
Summe über die Beiträge  
aller Lichtquellen  $j$



### Beispiel diffuse Beleuchtung



Lichtquelle links oben



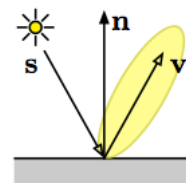
Lichtquelle rechts oben

$$\mathbf{k}_d = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



### Glänzende Reflexion (Glossy Reflection)

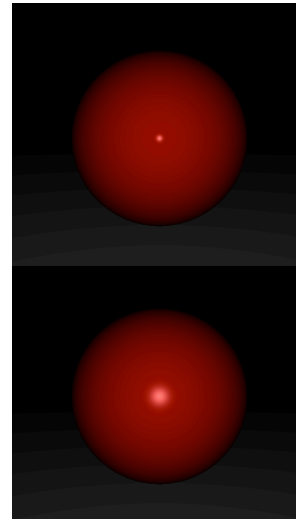
- Beobachtung:
  - Einige Materialien reflektieren Licht bevorzugt in bestimmte Richtungen
  - Die Stärke der Reflexion hängt ab von:
    - dem Winkel der Fläche zur *Lichtquelle* **und**
    - dem Winkel der Fläche zum Betrachter
  - Die Reflexion der Lichtquelle kann als Glanzlicht (Highlight) auf der Oberfläche beobachtet werden
- Beispiele
  - Glänzende Oberflächen wie Glas oder glattes Plastik
  - Glänzend beschichtete / versiegelte Oberflächen



## Glanzlichter (Highlights) nach Phong



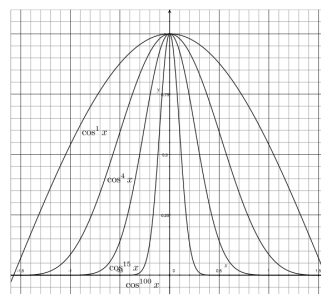
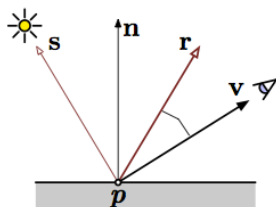
- Wie modelliere ich ein Glanzlicht?
  - Es gibt eine Hauptrichtung, aus der man das Glanzlicht beobachten kann
    - Hauptrichtung = Richtung perfekter Reflexion der Lichtquelle
  - Helligkeit läßt „nach außen“ nach
    - Je größer der Winkel zwischen Reflexionsrichtung und Blickrichtung, desto weniger Reflexion
  - Je glänzender das Material, desto „enger“ das Glanzlicht
    - Steilerer / schnellerer Abfall der Helligkeit mit dem Winkel



## Glanzlichter (Highlights) nach Phong



- Phong-Exponent
  - Richtung des Betrachters  $\mathbf{v}$
  - Ideale Reflexionsrichtung  $\mathbf{r}$
  - Reflektiertes Licht proportional zu  $(\mathbf{r} \cdot \mathbf{v})^a$
  - $\mathbf{r}, \mathbf{v}$  normiert  $\rightarrow (\mathbf{r} \cdot \mathbf{v}) = \cos(\mathbf{r}, \mathbf{v})$
  - $a$  = „Phong-Exponent“, „shininess“

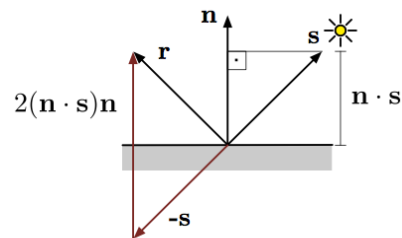
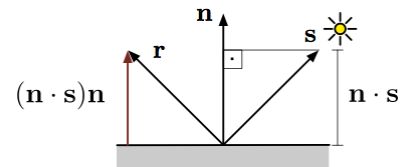


## Berechnung des Reflexionsvektors $\mathbf{r}$



$$\mathbf{r} = 2(\mathbf{n} \cdot \mathbf{s})\mathbf{n} - \mathbf{s}$$

- $\mathbf{r}$  ist die Spiegelung von  $\mathbf{s}$  an  $\mathbf{n}$ 
  - $\mathbf{s}, \mathbf{n}$  normiert
  - $(\mathbf{n} \cdot \mathbf{s})$  ist die Projektion von  $\mathbf{s}$  auf  $\mathbf{n}$  („Höhe“ von  $\mathbf{s}$ )
  - Von  $-\mathbf{s}$  gehe  $2(\mathbf{n} \cdot \mathbf{s})$  in Richtung  $\mathbf{n}$



## Spekularer Beleuchtungsterm des Phong-Modells



Spekular reflektiertes Licht am  
Punkt  $\mathbf{p}$  in Richtung  $\mathbf{v}$

Phong-Exponent  
 $a \in [1, \infty]$

$$L_s(\mathbf{p}, \mathbf{v}) = k_s \sum_j L_j(\mathbf{v} \cdot \mathbf{r}_j)^a$$

Konstanter Reflexionskoeffizient

$$\mathbf{k}_s = \begin{pmatrix} r_s \\ g_s \\ b_s \end{pmatrix}, \quad r_s, g_s, b_s \in [0, 1]$$

Winkel zwischen Betrachter  $\mathbf{v}$  und  
idealer Reflexionsrichtung  $\mathbf{r}_j$

Von Quelle  $j$  in Punkt  $\mathbf{p}$   
eintreffende Lichtmenge

Summe über die Beiträge  
aller Lichtquellen  $j$


**Annahmen:**

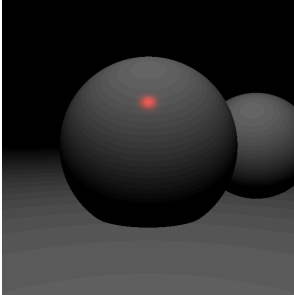
- Punktlicht / direktionales Licht
- $\mathbf{r}$  und  $\mathbf{v}$  normiert
- $(\mathbf{r} \cdot \mathbf{v}) > 0$  bzw.  $(\mathbf{n} \cdot \mathbf{s}) > 0$  (sonst Licht auf Rückseite)





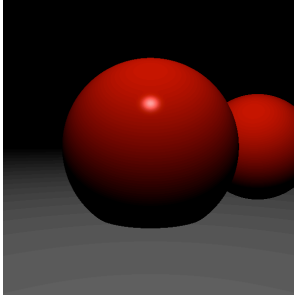
### Addition von diffusem + spekularem Anteil





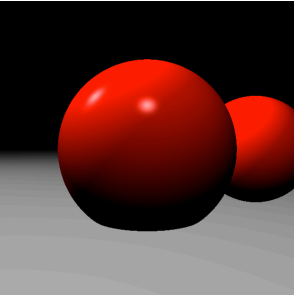
Rotes Highlight  
auf grauem Grund

$k_d = (0.5, 0.5, 0.5)$   
 $k_s = (1, 0, 0)$



Weißes Highlight  
auf rotem Grund


$k_d = (1, 0, 0)$   
 $k_s = (0.8, 0.8, 0.8)$

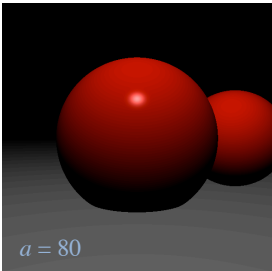


Zwei Lichtquellen

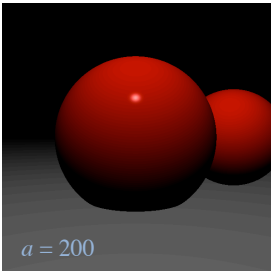
$k_d = (1, 0, 0)$   
 $k_s = (0.8, 0.8, 0.8)$

### Wertebereich des Phong-Exponenten $a$

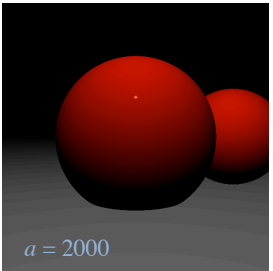




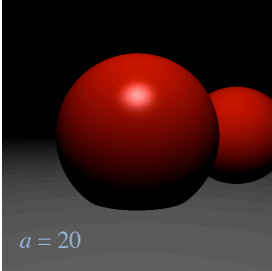
$a = 80$



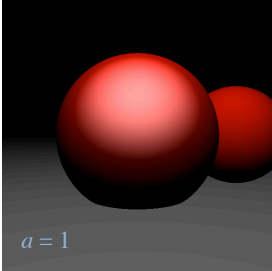
$a = 200$



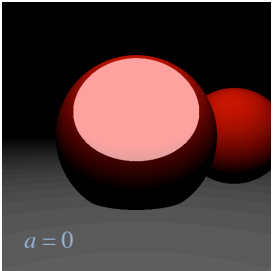
$a = 2000$



$a = 20$



$a = 1$

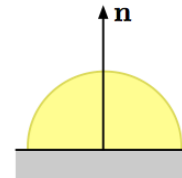


$a = 0$

## Ambiente Beleuchtung



- Modelliert Hintergrund-Beleuchtung
  - Konstante, globale, indirekte Beleuchtung
  - Wird als unabhängig von den bisher eingeführten Lichtquellen modelliert
  - Konstante ambiente Beleuchtung  $L^A$
  - Materialkoeffizient  $k_a$  gibt an, welcher Anteil davon reflektiert wird

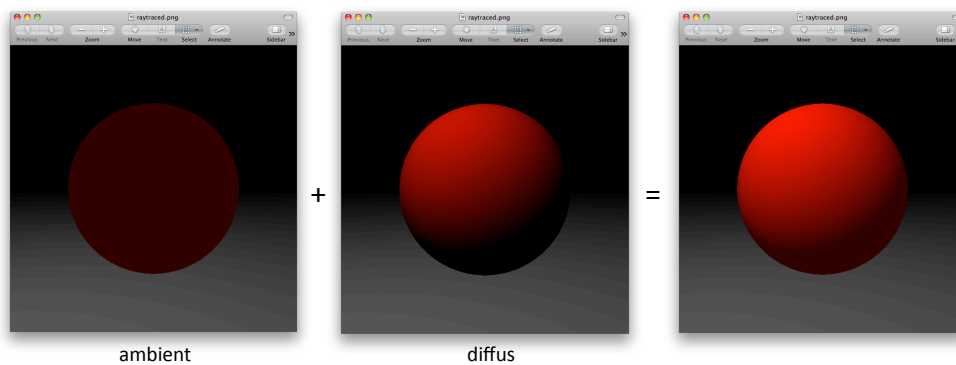


$$L_a = k_a L^A$$

$$k_a = \begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix}, \quad r_a, g_a, b_a \in [0, 1]$$



## Auswirkung des ambienten Anteils



- Der ambiente Anteil macht vor allem in Kombination mit diffuser und spekularer Reflexion Sinn, um die Schattenbereiche der Objekte aufzuhellen

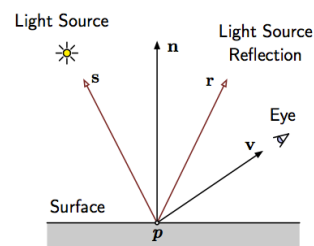


## Vollständige Phong-Beleuchtungsformel



$$L(\mathbf{p}, \mathbf{v}) = \underbrace{k_a L^A}_{\text{ambient}} + \underbrace{k_d \sum_j L_j (\mathbf{n} \cdot \mathbf{s}_j)}_{\text{diffus}} + \underbrace{k_s \sum_j L_j (\mathbf{v} \cdot \mathbf{r}_j)^a}_{\text{spekular}}$$

- $k_a, k_d, k_s$  : ambiente, diffuse und spekulare Reflektivitäts-Koeffizienten, (R,G,B)-Vektoren
- $a$  : Phong-Exponent, „Shininess“ (  $a \geq 1$  )
- $L^A$  : ambientes Umgebungslicht
- $L_j$  : von Lichtquelle  $j$  eintreffendes Licht
- $\mathbf{n}, \mathbf{v}$  : Oberflächennormale, Richtung zum Betrachter
- $\mathbf{s}_j, \mathbf{r}_j$  : Richtung zur Lichtquelle  $j$ , deren Spiegelung an der Normalen  $\mathbf{n}$
- $|\mathbf{n}| = |\mathbf{v}| = |\mathbf{s}_j| = |\mathbf{r}_j| = 1$
- $(\mathbf{n} \cdot \mathbf{s}_j) > 0$  und  $(\mathbf{v} \cdot \mathbf{r}_j) > 0$



Implementierung



## Implementierungshinweise – Shading



- Jedem Objekt wird anstelle einer einfachen Farbe ein *Material* zugewiesen
  - Die Materialbeschreibung ist stark abhängig vom Beleuchtungsmodell, also handelt es sich quasi um ein „Phong-Material“
  - Anstatt das Shape nach seiner Farbe zu fragen, frage nach seinem Material
  - Das Material wird durch die drei Reflexions-Koeffizienten und den Phong-Exponenten vollständig beschrieben
  - Das Material kann dann die Farbe an einem bestimmten Punkt auf der Oberfläche ausrechnen (*Shading-Operation*, *Shader*)

```

Shape {abstrakt}
emissiveColor: Color
material: Material

intersect(Ray): Hit {abstrakt}
getMaterial(): Material
    
```

```

Material

kAmbient: Color
kDiffuse: Color
kSpecular: Color
phongExponent: float

shade(point: Vector, ...): Color
    
```



## Implementierungshinweise – Lichtquellen



- Die Szene benötigt nun neben den geometrischen Objekten (Shapes)...
- ... eine Menge von Punktlichtquellen und eine ambiente Lichtquelle
  - Eine Punktlichtquelle wird vollständig durch eine Position und eine konstante Farbe / Strahldichte L beschrieben
  - Alle Lichtquellen müssen irgendwie an den Shader übergeben werden
    - Alternativ: eine Referenz auf die Szene übergeben, und der Shader fragt die Szene nach den Lichtquellen.

```

Scene

shapes: ArrayList<Shape>
lights: ArrayList<LightSource>
ambientLight: Color

intersect(Ray): Hit
getLightSources(): Collection<LightSource>
getAmbientLight(): Color
    
```

```

LightSource

position: Vector
color: Color

getPosition(): Vector
getColor(): Color
    
```



## Implementierungshinweise – Normalen



- Für die Beleuchtungsberechnung benötigt der Shader auch die Normale in dem Oberflächenpunkt
  - Diese Normale kann z.B. ebenfalls im Hit-Objekt gespeichert werden, oder direkt über das getroffene Objekt abgefragt werden
  - Jedes geometrische Objekt sollte in der Lage sein, für der Trefferpunkt auch eine Normale zu berechnen
    - Ebene: Normale ist immer die gleiche
    - Kugel: Normalenrichtung ist die Richtung vom Mittelpunkt zum Oberflächenpunkt
    - Box: (siehe Ebene)



## Implementierungshinweise – Shader-Funktion



- Signatur der Methode `Material.Shade()`
  - Strahl, Richtung zum Auge
  - Position des Treffers
  - Getroffenes Objekt
    - Oberflächennormale am Trefferpunkt
    - Material des getroffenen Objekts
  - Liste von Punktlichtquellen
  - Ambientes Licht

} Hit  
} via Hit  
} Scene  
(oder direkt)



## Implementierungshinweise – Berechnungen



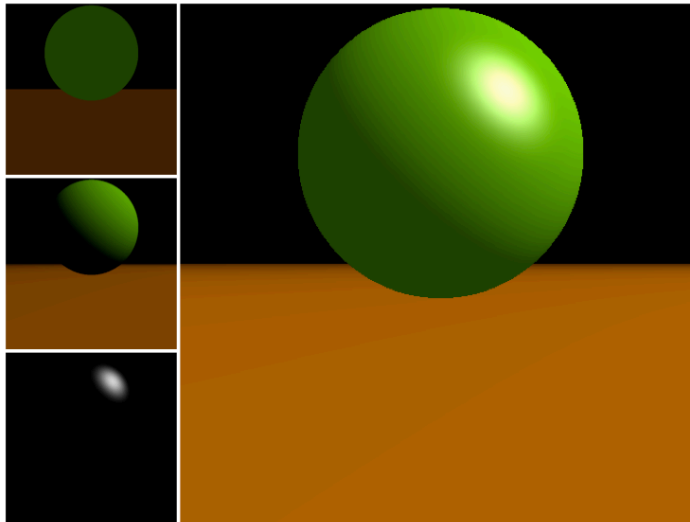
- **Beleuchtungsberechnungen**
  - Achtung: testen, ob Lichtquelle vor oder hinter der Fläche ist!
    - Wenn hinter der Fläche, kann sie nichts beitragen
  - Testen auf sehr kleine Werte (Epsilon), sonst unerwünschte Effekte (Berechnung numerisch nicht sehr robust)
  - Die meisten Formeln gehen von normierten Richtungsvektoren aus (im Zweifelsfall also normieren)



... und weiter?



Warum sieht das noch nicht so realistisch aus?



... Schatten und Interreflexionen fehlen!

