




# Computer-Graphik 1

## Lighting & Shading



G. Zachmann  
Clausthal University, Germany  
[cg.in.tu-clausthal.de](http://cg.in.tu-clausthal.de)

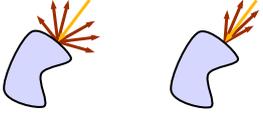



## Beleuchtungsmodelle (*lighting models*)

- Definition "Beleuchtungsmodell": Vorschrift zur Berechnung der **Farb- und Helligkeitswerte** von Punkten auf der Oberfläche von Objekten
  - Grundlage sind physikalische Gesetze
  - Modelliert werden Einflüsse von:
    1. **Lichtquellen** (Position, Intensität, Farbe, etc.)



- 2. **Objektoberfläche** (Geometrie, Reflexionseigenschaften)

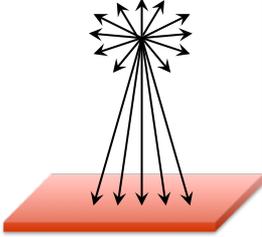
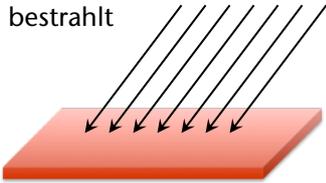


- Für Echtzeitanwendungen verwendet man sehr einfache Modelle

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 2

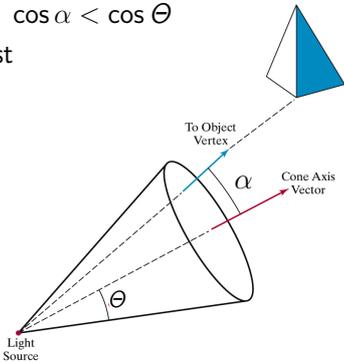
## Lichtquellen

- **Punktlichtquelle (*point light*):** strahlt in alle Richtungen gleichmäßig ab
  - Wird eindeutig charakterisiert durch
    1. Position &
    2.  $I(\lambda)$  = abgestrahltes Spektrum  
= Intensität abhängig von der Wellenlänge
- **Richtungslichtquelle (*directional light*):** jeder Punkt im Raum wird aus derselben Richtung bestrahlt
  - Charakterisiert durch Richtung &  $I(\lambda)$
  - Beispiel: Sonne

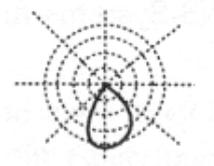
G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 3

- **Strahler (*spot light*):** Lichtausbreitung wird auf einen bestimmten Raumwinkel (Lichtkegel) beschränkt. Der Abfall der Lichtstärke von der Kegelachse zum Rand wird durch folgendes Gesetz bestimmt:
 
$$I(\lambda) = \begin{cases} 0 & \text{falls } \cos \alpha < \cos \Theta \\ I_0(\lambda) \cos^n \alpha & \text{sonst} \end{cases}$$
- Charakterisierung durch:
  - Position, Richtung (Kegelachse),
  - Exponent (Öffnungswinkel),  $I(\lambda)$



G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 4

- **Goniometrische Lichtquelle:** Abstrahlcharakteristik wird per Tabelle beschrieben. Zur Ermittlung von  $I(\lambda)$  muß evtl. zwischen Tabelleneinträgen interpoliert werden

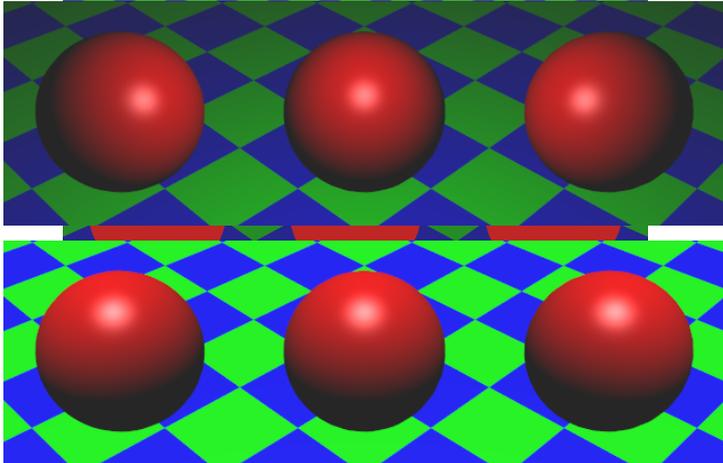


- **Area light source**

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 5

Unterschiedlicher Effekt zwischen *point light* und *directional light*

- Wie erkennt man anhand der Beleuchtung einer Kugel, von welcher Art die Lichtquelle ist? (point oder directional?)



point light source

directional light source

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 6

## Lokale Beleuchtungsmodelle

- Vereinfachung: berücksichtige bei der Berechnung der Beleuchtung eines Punktes **keine sekundären Effekte** (Strahlungsaustausch zwischen Objekten), **nur primären** Austausch zwischen Lichtquelle und Objekt → **lokales Beleuchtungsmodell**
- **Superpositionsprinzip**: betrachte Licht als Teilchen →

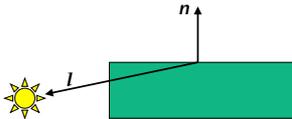
$$I(\lambda) = \sum_j I_j(\lambda)$$

- Vereinfachung der Notation: wir lassen im Folgenden  $\lambda$  überall weg, und merken uns, daß alle photometrischen Größen eigtl. von  $\lambda$  abhängen!

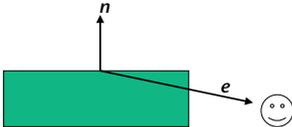
G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 7

## Konvention bzgl. negativer Skalarprodukte

- Falls  $\mathbf{n} \cdot \mathbf{l} < 0$ , dann befindet sich das Licht hinter der Fläche



- Falls  $\mathbf{n} \cdot \mathbf{e} < 0$ , so befindet sich der Betrachter auf der Rückseite der Fläche



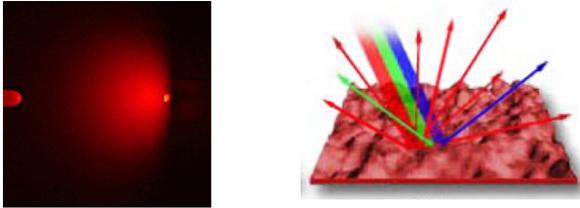
- Wir definieren im Folgenden (der Einfachheit halber) prinzipiell:

$$\mathbf{n} \cdot \mathbf{l} := \max(0, \mathbf{n} \cdot \mathbf{l})$$

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 9

## Diffuse Reflexion

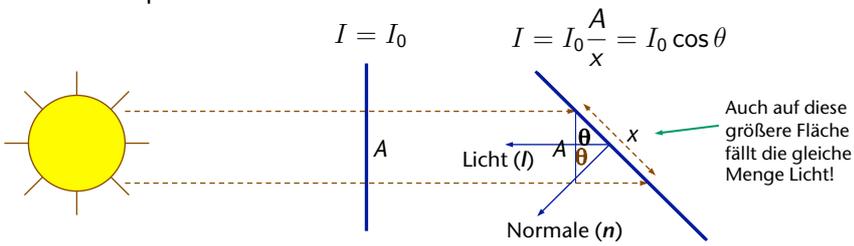
- Licht wird von der Objektoberfläche gleichmäßig in alle Richtungen reflektiert
- Folge: Helligkeit ist unabhängig vom Viewpoint!
- Beispiele: Stück Papier, Tafel, unbearbeitetes Holz
- Diffuse/Matte Objekte werden auch Lambert'sche Objekte bezeichnet



G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 10

## Lambert'sche Oberflächen

- Das Lambert'schen Kosinus-Gesetz:  
Die Intensität  $I$  einer Oberfläche ist proportional zum Kosinus des Winkels zwischen der Oberflächennormalen und der Richtung zur Lichtquelle



- Fazit:  
$$I = I_0 \cos \theta = I_0 \cdot \mathbf{n} \cdot \mathbf{l}$$
 Annahme:  $\mathbf{n}$  und  $\mathbf{l}$  seien Einheitsvektoren

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 11

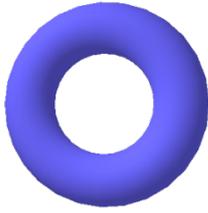
## Ambientes Licht

- Das Lambert'sche Modell erzeugt schwarze Farbe für Oberflächen, die nicht zur Lichtquelle zeigen
- In der Realität trifft Licht aus allen Richtungen ein (dieses wurde von anderen Objekten, evtl. mehrfach, reflektiert)
- Füge für alle Objekte einen **ambienten Beleuchtungsterm** ein:

$$I = I_a + I_0 \cdot \mathbf{n} \cdot \mathbf{l}$$



Nur ambiente Beleuchtung



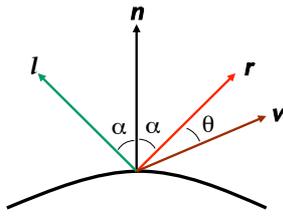
Diffuse + ambiente Beleuchtung

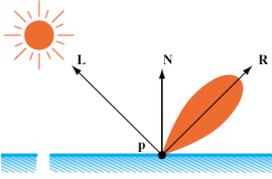
G. Zachmann Computer-Graphik 1 - WS 09/10
Lighting & Shading 12

## Spiegelnde Reflexion

- Stellt Glanzpunkte auf glänzenden Oberflächen dar
- Oberflächenreflexion ist abhängig von
  - Richtung der Lichtquelle,  $\mathbf{l}$
  - Oberflächennormale,  $\mathbf{n}$
  - Richtung zum Betrachter,  $\mathbf{v}$
- Bei idealer spiegelnder Reflexion sieht man nur dann Licht von der Lichtquelle, wenn  $\mathbf{r} = \mathbf{v}$
- Bei glänzenden Oberflächen sieht man auch "nahe" bei  $\mathbf{r}$  ein Highlight; das erreicht man z.B. mit

$$I = I_0 (\cos \theta)^p$$





G. Zachmann Computer-Graphik 1 - WS 09/10
Lighting & Shading 13

## Gerichtet diffuse Reflexion

- Ideal diffuse und ideal spiegelnde Reflexion sind in der Realität selten; meist eine Mischung
- **Gerichtet diffuse Reflexion**: die abgestrahlte Intensität hat (oft) ein Maximum in Richtung der idealen Reflexion

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 15

## Das Phong-Beleuchtungsmodell [1975]

- Zusammensetzung:

$$I = I_{\text{amb}} + I_{\text{diff}} + I_{\text{spec}}$$

- Aufgrund des Superpositionsprinzips erhält man für  $n$  Lichtquellen:

$$I = r_d \cdot I_a + \sum_{j=1}^n (r_d \cos \phi_j + r_s \cos^p \theta_j) \cdot I_j$$

$r_d = r_d(\lambda)$  = diffuser Reflexionskoeffizient (spiegelnde Materialfarbe)  
 $r_s = r_s(\lambda)$  = spekularer Reflexionskoeffizient (diffuse Materialfarbe)  
 $p$  = "Glanzzahl" (*shininess*), hat keine Einheit (hat keine physikalische Bedeutung)

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 16

ambient + diffus + spekulär = Phong

Phong	$\rho_{\text{ambient}}$	$\rho_{\text{diffuse}}$	$\rho_{\text{specular}}$	$\rho_{\text{total}}$
$\phi_i = 60^\circ$				
$\phi_i = 25^\circ$				
$\phi_i = 0^\circ$				

G. Zachmann Computer-Graphik 1 - WS 09/10
Lighting & Shading 17

### Demo

Ambient: 0.160 0.080 0.000  
 Diffuse: 0.443 0.221 0.000  
 Specular: 0.000 0.000 0.000

(Quelle: <http://www.avl.iu.edu/%7Eewernert/gviz/phong/>)

G. Zachmann Computer-Graphik 1 - WS 09/10
Lighting & Shading 18

### Effekt der Parameter im Phong-Modell

Der Exponent  $p$  steuert die "Schärfe" des Highlights:

in Polarkoord.

$p = 1$

$p = 2$

$p = 4$

$p = 8$

$p = 16$

$p = 32$

$p = 64$

$p = 128$

$p = 256$

$$I_{\text{spec}} = r_s I_0 \cos^p \Theta = r_s I_0 (\mathbf{r} \cdot \mathbf{e})^p$$

G. Zachmann Computer-Graphik 1 - WS 09/10
Lighting & Shading 19

$r_a$

$r_d$

$r_s$

$p$

→ "Oberflächenkörnung" ( $p$ )

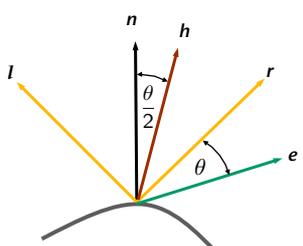
G. Zachmann Computer-Graphik 1 - WS 09/10
Lighting & Shading 20

- Zusätzliche Freiheit: diffuse und spekulare (= spiegelnde) Materialfarbe können verschieden sein.
- Problem: Werte > 1 können entstehen!
  - Abhilfe: Clamping
  - Besser wäre: Erhalten von Farbton und Sättigung

Lighting & Shading 21

## Das Blinn-Phong Modell [1978]

- Problem des Phong-Modells: man muß für jeden Punkt den Reflexionsvektor bestimmen
- Idee: verwende Winkelhalbierende  $\mathbf{h}$  ("half-vector") und  $\mathbf{n}$ , statt  $\mathbf{r}$  und  $\mathbf{e}$ :
 
$$\mathbf{h} = \frac{\mathbf{l} + \mathbf{e}}{|\mathbf{l} + \mathbf{e}|}$$
  - Setze:
 
$$I'_{\text{spec}} = r_s I_0 \cos^q \frac{\theta}{2} = r_s I_0 (\mathbf{h} \cdot \mathbf{n})^q$$

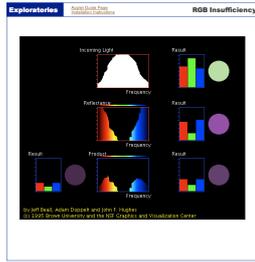


- Es gilt:  $I'$  ist max  $\Leftrightarrow I$  ist max
- Frage: ist es dasselbe Modell?  $\rightarrow$  fast
- Vorteil dieser Methode: wenn Auge und Lichtquelle unendlich weit entfernt sind, dann ist  $\mathbf{h}$  (für eine bestimmte Lichtquelle) konstant! (Kann man also am Beginn eines Frames vorberechnen)

Lighting & Shading 22

## Spectral Lighting vs RGB-Lighting

- Erinnerung: alle photometrischen Größen sind eigtl. **Funktionen in  $\lambda$ !** beschreiben also ein Spektrum ...
- In der Praxis (z.B. OpenGL): führe alle Berechnungen jeweils für die 3 Primärvalenzen durch
- Aber: dadurch erhält man nicht 100% korrekte Bilder!
- Denn:
 
$$\text{RGB}(r(\lambda) \cdot I(\lambda)) \neq \text{RGB}(r(\lambda)) \cdot \text{RGB}(I(\lambda))$$



The screenshot shows a web browser window titled 'Exploratories' with a sub-page 'RGB Insufficiency'. It displays three spectral plots: 'Intensity' (a white curve), 'Reflectance' (a blue curve), and 'Product' (a red curve). Each plot has a corresponding 'Pixel' image to its right, showing how the combination of spectral components affects the resulting color. The text at the bottom of the browser window reads: '© 1998-2001, Albert S. Gotsch and James F. Foley. All rights reserved. See the README file for details and the full license.' Below the browser window is the URL <http://www.cs.brown.edu/exploratories/>.

<http://www.cs.brown.edu/exploratories/>

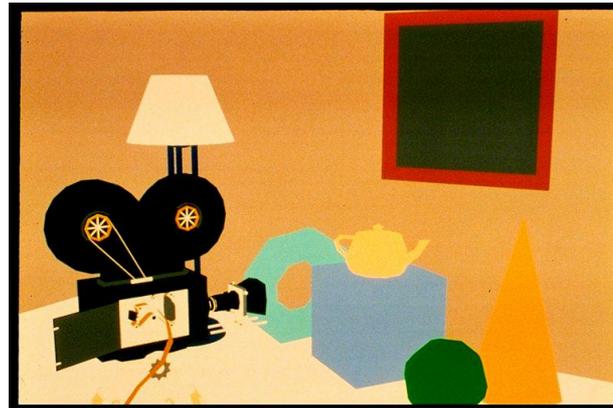
G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 23

## Shading-Algorithmen

- Achtung: unterscheide zwischen **Beleuchtungsmodell (lighting model)** und **Beleuchtungsalgorithmus (shading algorithm)**!
  - Beleuchtungsmodell** beschreibt Zusammenhang zwischen Lichtquellen und Oberflächen zur Berechnung der Intensität in jedem Punkt.
  - Beleuchtungsalgorithmus** berechnet aus der Intensität/Farbe einiger Punkte die Farbe **aller** Bildpunkte.
  - Leider: große Begriffsverwirrung! ;-(
    - "lighting algorithm", "shading model", ...
- 3 Möglichkeiten, das Beleuchtungsmodell auszuwerten:
  - 1x pro Polygon
  - 1x pro Vertex
  - 1x pro Pixel

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 24

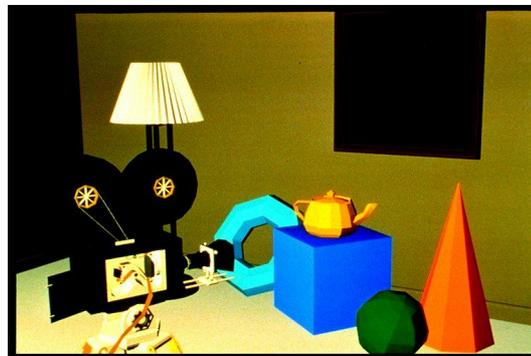
- Die Szene ganz ohne Shading ...



Pixar "Shutterbug"

## Flat Shading (Konstante Beleuchtung)

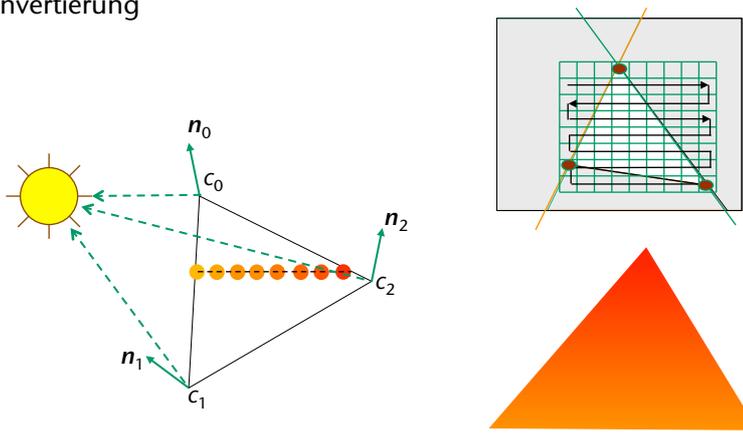
- Simplester Shading-Algo: jedes Polygon erhält einen einheitlichen Farbwert
  - Werte dazu (im Prinzip irgend ein) Beleuchtungsmodell an irgend einer Ecke des Polygons aus



Pixar "Shutterbug"

## Gouraud-Shading [1971]

- Werte das Beleuchtungsmodell an allen 3 Ecken des Dreiecks aus, interpoliere linear dazwischen während der Scanline-Konvertierung



G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 27

## Vergleiche



Gouraud-Shading mit rein diffusem Beleuchtungsmodell

Gouraud-Shading mit Phong-Lighting

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 28

The slide compares two shading techniques. On the left, a scene with a green cube, a yellow sphere, a red sphere, and a blue ring on a white surface is shown. The top image, labeled 'Flat-Shading', shows sharp, distinct color boundaries between the objects. The bottom image, labeled 'Gouraud-Shading mit Phong-Lighting', shows smooth color transitions and realistic shading on the surfaces. On the right, a yellow Volkswagen Beetle is shown. The top image, labeled 'Flat-Shading', shows the car with flat, uniform colors and no shading. The bottom image, labeled 'Gouraud-Shading mit Phong-Lighting', shows the car with smooth shading and realistic lighting effects.

Flat-Shading

Gouraud-Shading mit Phong-Lighting

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 29

### Einfluß der Triangulierung

- Orientierung:**
- Auflösung:**

Vergrößerung

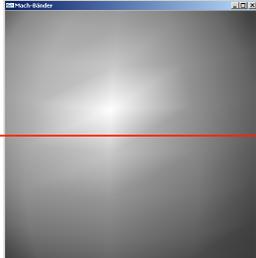
1 x 1 Flächen 2 x 2 Flächen 3 x 3 Flächen

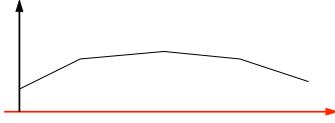
5 x 5 Flächen 10 x 10 Flächen 40 x 40 Flächen
- Frage: Woher kommen die „Streifen“?**

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 30

## Mach-Bänder

- Problem: die lineare Interpolation der "Lichtwerte"
  - Diese Interpolation ist  $C^0$ -stetig, aber nicht  $C^1$ -stetig!
- Das menschliche Auge hat einen eingebauten "Kantendetektor" (genau diese Knicke)
  - Es gibt Neuronen, die die Ableitung bilden (jew. für ein Retina-"Pixel")
  - Wahrgenommene Intensität = physik. Intensität + Ableitung
  - Resultat: Mach-Bänder bei linearer Interpolation
- Abhilfe: Hardware müsste höherwertig interpolieren...

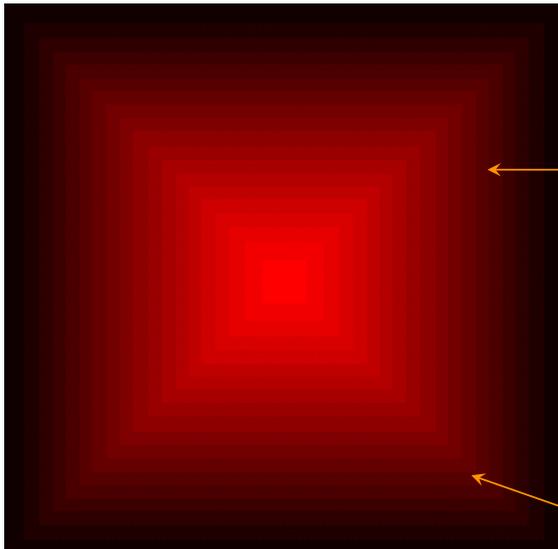






G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 31

## Extremes Beispiel



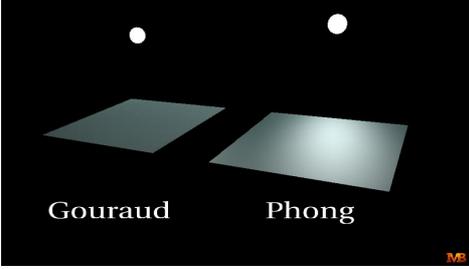
Die Intensität innerhalb eines Quadrates ist konstant!  
Ein Farbverlauf von innen nach außen ist eine Illusion.

Die hellen Linien bei  $45^\circ$  (und  $135^\circ$ ) sind eine Illusion!

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 32

## Weiteres Problem beim Gouraud-Shading

- Evtl. "verpasst" man Highlights im Inneren eines Polygons:

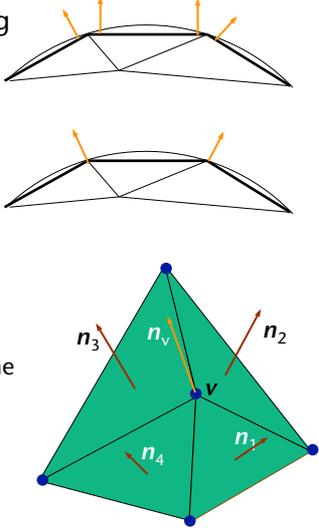


Gouraud      Phong

G. Zachmann    Computer-Graphik 1 - WS 09/10      Lighting & Shading    33

## Berechnen der Normalen der Eckpunkte

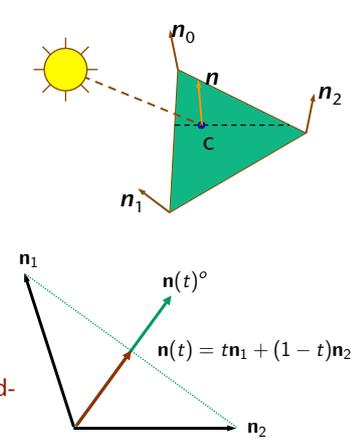
- Die Dreiecke bilden nur eine Annäherung an die wirkliche Oberfläche eines Objektes
- An den Vertices hätte man gerne die Normale der Fläche, nicht der Dreiecke!
- Algorithmus:
  - Zu Beginn berechne eine Normale für jedes Polygon
  - Bestimme für jeden Vertex, welche Polygone diesen enthalten
  - Bestimme den "Mittelwert" der Normalen dieser angrenzenden Polygone
    - Einfach aufsummieren, dann normieren



G. Zachmann    Computer-Graphik 1 - WS 09/10      Lighting & Shading    34

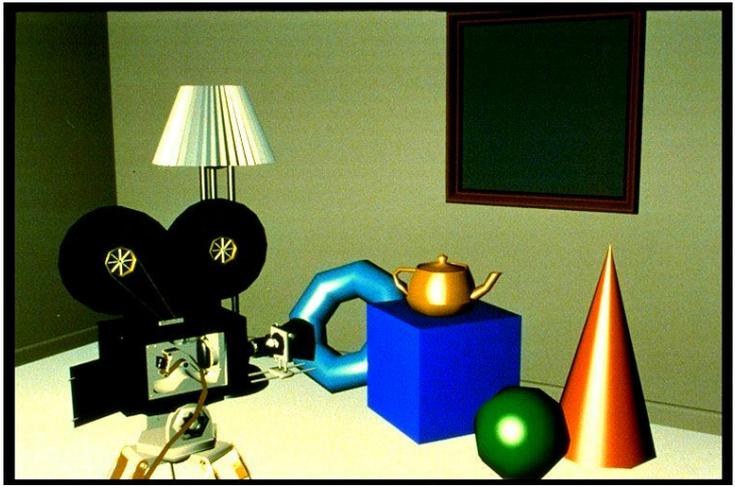
## Phong-Shading

- Idee: interpoliere die **Normale** während der Scanline-Konvertierung (und weitere Parameter) und werte das Beleuchtungsmodell in **jedem Pixel** aus
- Wie interpoliert man Normalen?
  - Typischerweise: linear mit anschließender Normierung
  - Achtung: ohne Normierung** bekäme man nur (sehr umständliches) **Gouraud-Shading!**
  - War früher sehr teuer, daher wurden viele Alternativen vorgeschlagen
    - Inkrementell, Taylor-Reihe + LUT, ...



$n(t) = tn_1 + (1-t)n_2$

G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 35



G. Zachmann Computer-Graphik 1 - WS 09/10 Lighting & Shading 36

