

# Graphische Datenverarbeitung

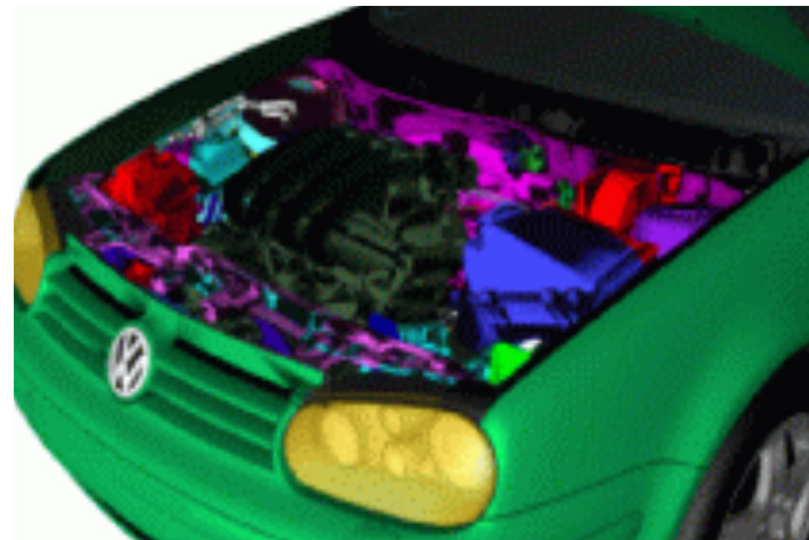
## Visualisierungstechniken

Prof. Dr. Elke Hergenröther

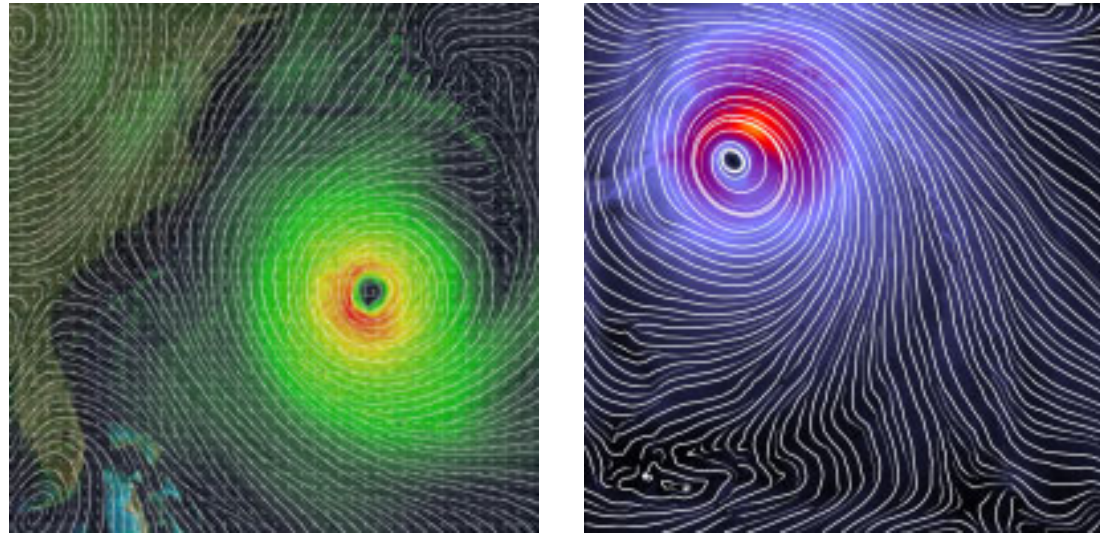
# Visualisierungstechniken

## Visualisierung:

Visualisierung bedeutet sichtbar machen, darstellen. Die Computer Graphik beschränkt sich dabei jedoch nicht auf die Abbildung real existierender Objekte sondern beschäftigt sich auch mit der Darstellung von Information und wissenschaftlich-technischen Daten.



# Strömungsvisualisierung



Beispiele für Strömungsvisualisierungs-Resultate. Der verwendete Datensatz ist eine Simulation des Hurrikan Isabel.  
(Bilder von: Technische Universität Wien, Institut für Computergraphik und Algorithmen)

# Visualisierungstechniken

## **Visualisierung:**

Visualisierung bedeutet sichtbar machen, darstellen. Die Computer Graphik beschränkt sich dabei jedoch nicht auf die Abbildung real existierender Objekte sondern beschäftigt sich auch mit der Darstellung von Information und wissenschaftlich-technischen Daten.

## **Visualisierung geometrischer Modelle**

- Visualisierung ohne Berücksichtigung von Licht
- Visualisierung mit Berücksichtigung von Licht
  - lokale Verfahren
  - globale Verfahren

# Visualisierung ohne Berücksichtigung von Licht

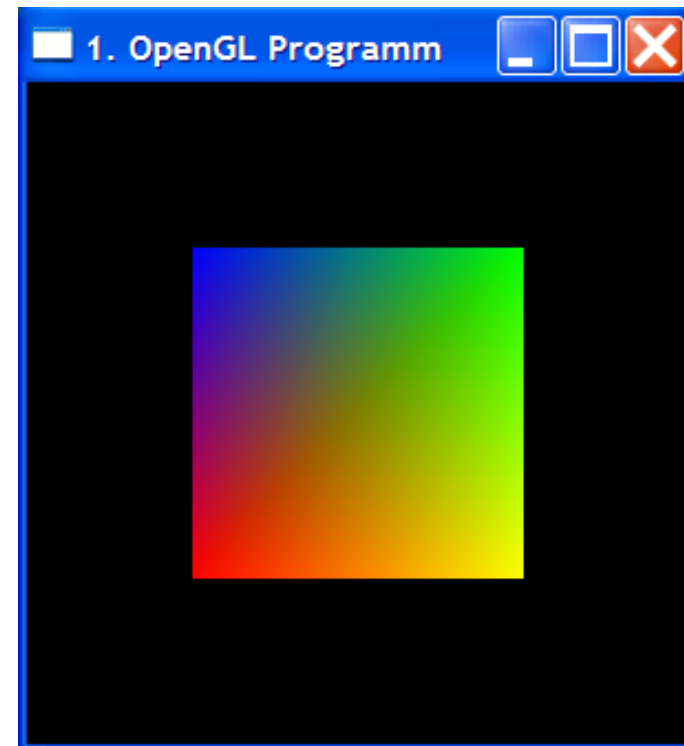
**Wie berechnet man den Farbverlauf innerhalb einer Linie oder eines Dreiecks, wenn jedem Punkt eine andere Farbe zugeordnet wurde?**

## **Lineare Interpolation:**

Interpolationsverfahren für Linien

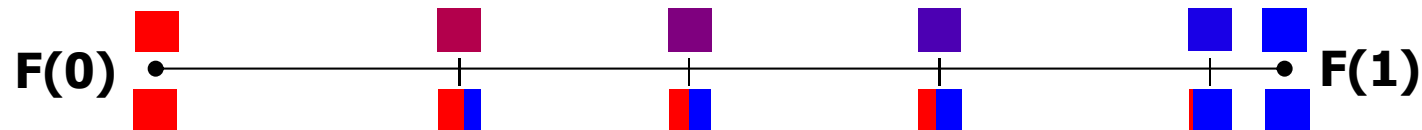
## **Bilineare Interpolation:**

Interpolationsverfahren für Flächen



# Lineare Farbinterpolation

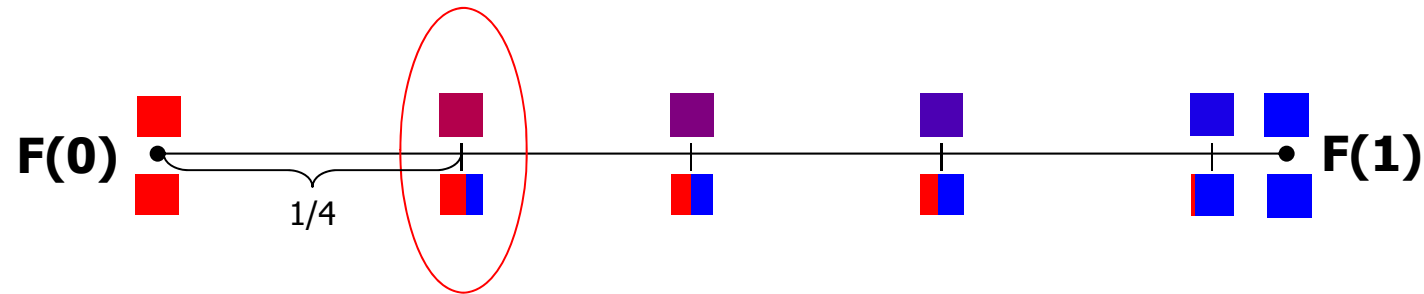
Verhältnis der „Farbanteile“ von  $F(0)$  und  $F(1)$



$$F(t) = (1 - t) \cdot F(0) + t \cdot F(1)$$

$$F(t) = (1 - t) \cdot \begin{pmatrix} R(0) \\ G(0) \\ B(0) \end{pmatrix} + t \cdot \begin{pmatrix} R(1) \\ G(1) \\ B(1) \end{pmatrix}$$

# Lineare Farbinterpolation



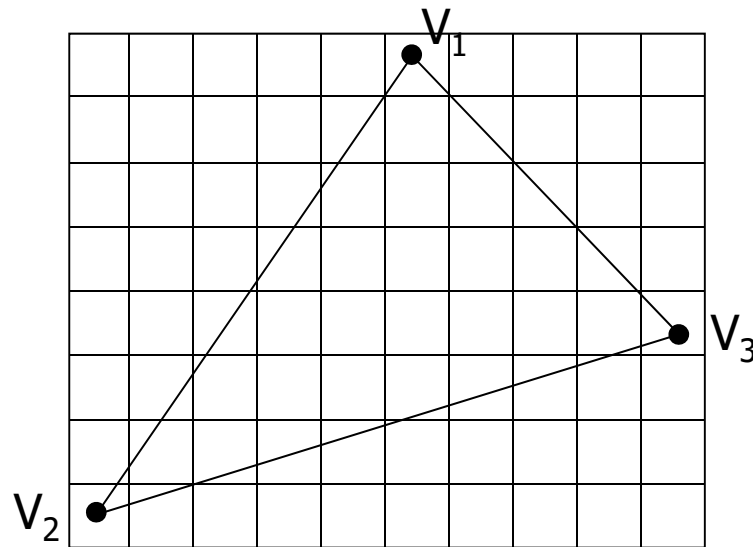
$$F(t) = (1-t) \cdot F(0) + t \cdot F(1)$$

$$F(t) = \left(1 - \frac{1}{4}\right) \cdot F(0) + \frac{1}{4} \cdot F(1)$$

$$F(t) = \frac{3}{4} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \frac{1}{4} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{4} \\ 0 \\ \frac{1}{4} \end{pmatrix}$$

# Bilineare Farbinterpolation für Dreiecke

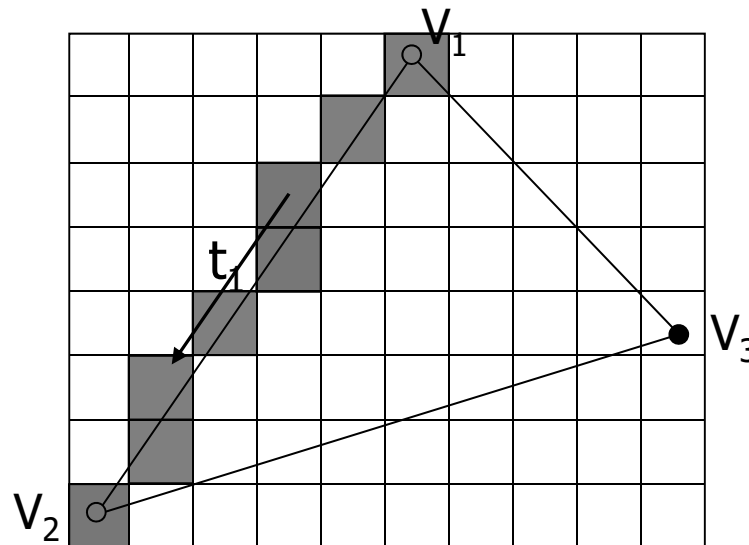
## 1. Schritt: Rasterisierung des Dreiecks





# Bilineare Farbinterpolation für Dreiecke

## 2. Schritt: Farbwerte der Pixel zwischen den Vertices interpolieren



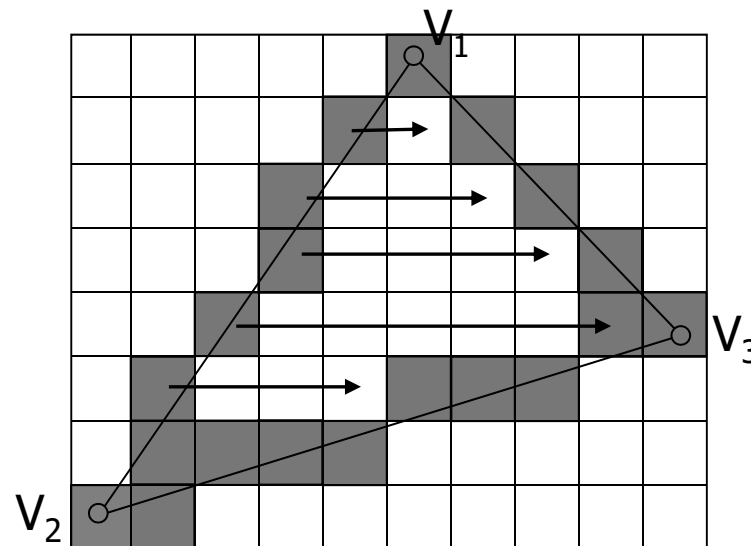
### Interpolation zwischen

#### $V_1$ und $V_2$ :

- Zunächst muss die Strecke zwischen den beiden Vertices normiert werden.
- Der Parameter  $t_1$  steht für die normierte Strecke und geht von 0 bis 1.
- Danach erfolgt eine lineare Farbinterpolation abhängig von  $t_1$ .

# Bilineare Farbinterpolation für Dreiecke

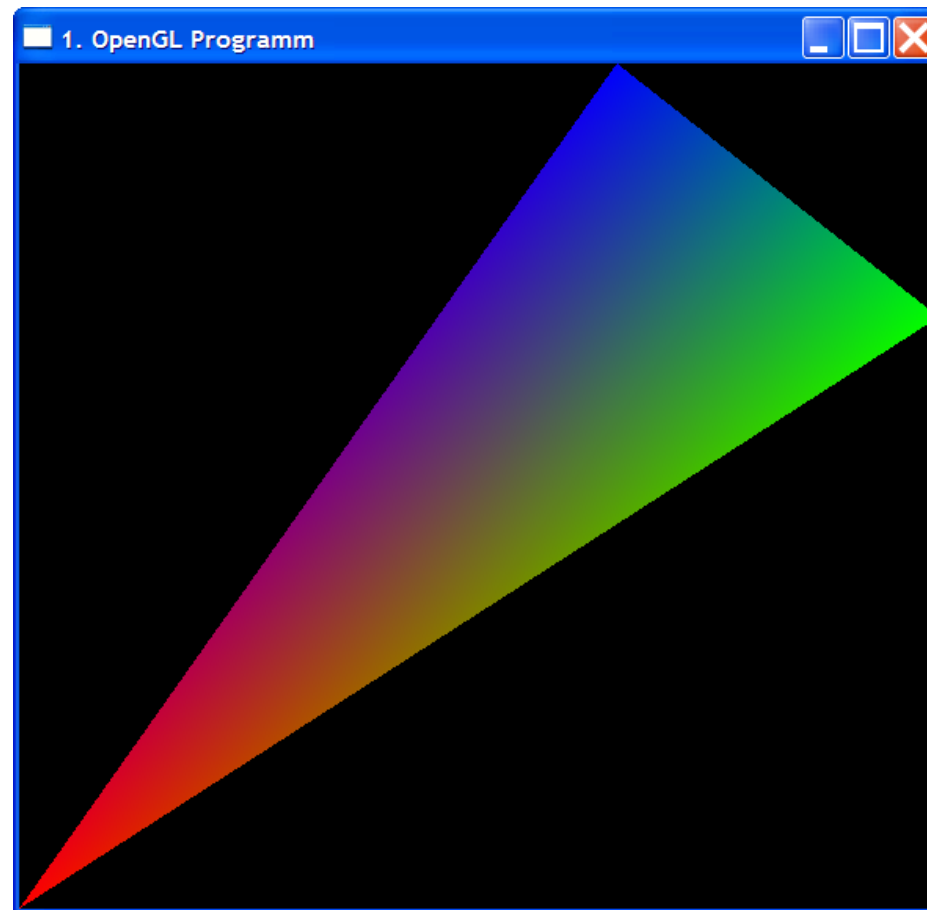
Farbwerte der Pixel zwischen den Vertices wurden linear interpoliert



### 3. Schritt:

Lineare  
Interpolation der  
Pixelwerte entlang  
der einzelnen  
Rasterzeilen

# Ergebnis einer bilinearen Farbinterpolation



# Rendering

## **Rendern:**

- Rendern bedeutet: Visualisieren einer beleuchteten Szene.
- Dabei spielen die Beleuchtungsverhältnisse und die Materialien, die auf die Beleuchtung reagieren eine wichtige Rolle.

## **Renderer:**

- Hard- oder Software, die die Visualisierung der Szene übernimmt.

## Faktoren, die das Aussehen eines Objektes bestimmen:



- Position und Orientierung des Betrachters relativ zum beleuchteten Objekt
- Position, Orientierung und Typ der Lichtquelle
- Oberflächenmaterial des Objektes

Abbildung entnommen von:

<http://forums.cgsociety.org/showthread.php?t=233460>

Zuletzt besucht: 2006-01-22.

## Faktoren, die das Aussehen eines Objektes bestimmen:

Beschaffenheit der Objektoberfläche & Verfahren zum Rendern des Objektes



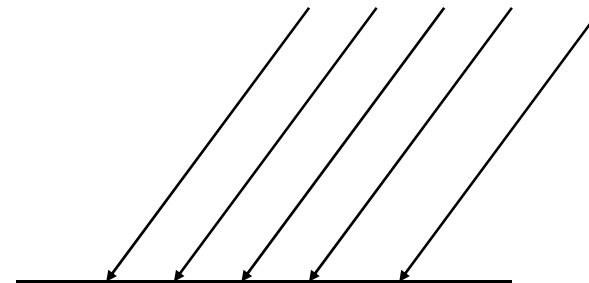
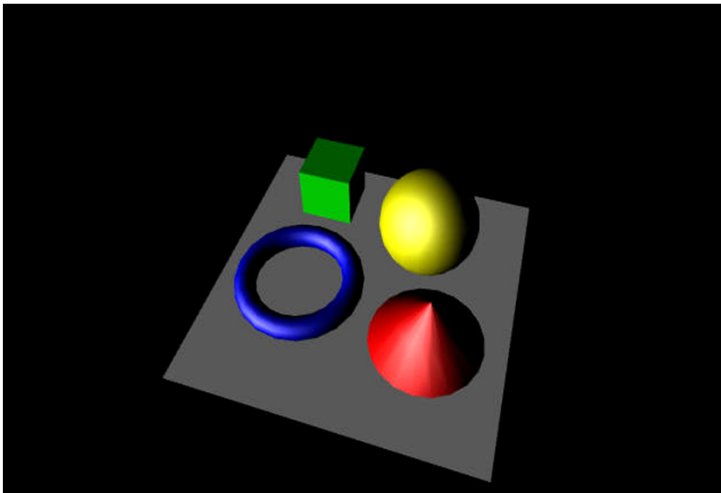
Simulation von Samt



Simulation von Satin

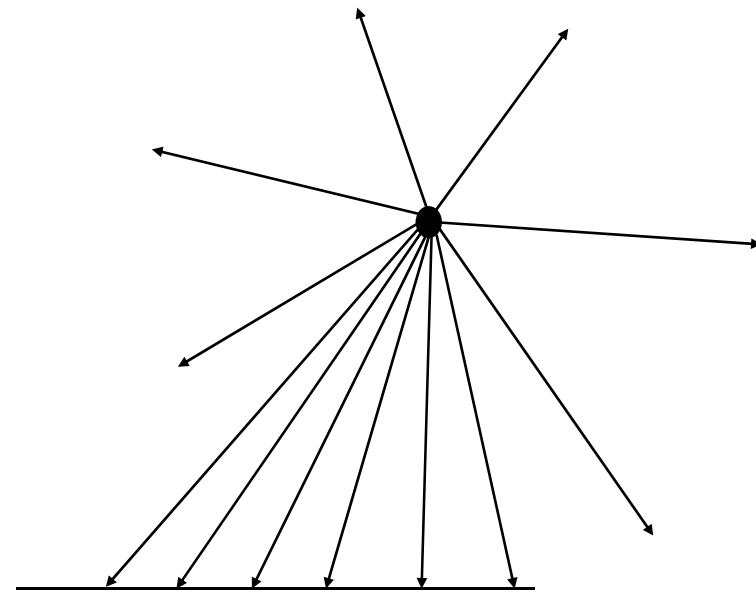
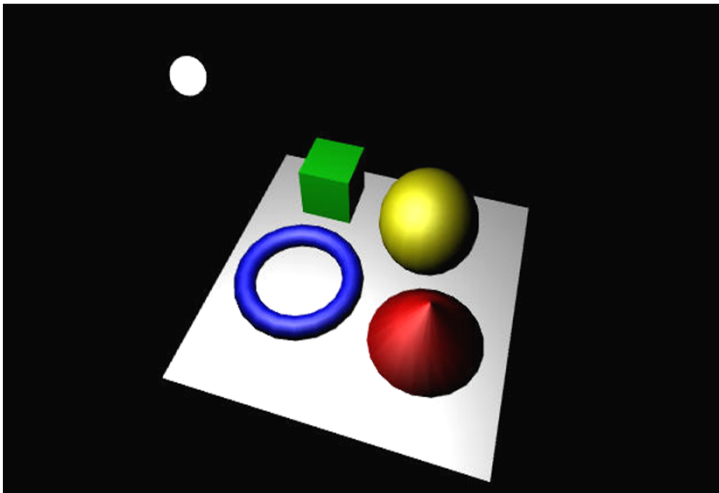
# Gerichtete Lichtquelle

- Entsprechen einer unendlich weit entfernten Lichtquelle
- parallel einfallende Strahlen.



# Punktlichtquelle

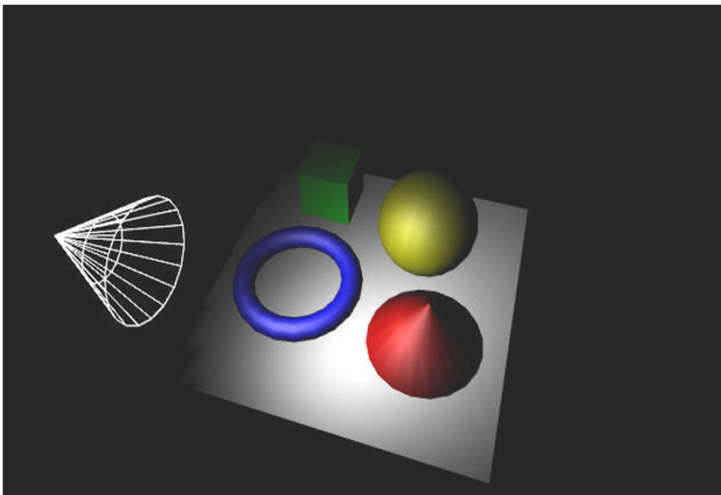
- Ein lokalisierter Punkt strahlt das Licht aus.
- unterschiedliche Winkel auf eine ebene Fläche





# Spotlichtquelle

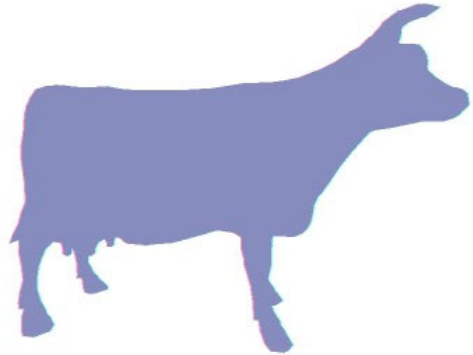
- Nur um einen bestimmten Winkel um eine angegebene Richtung wird Licht ausgestrahlt.
- je weiter von der Richtung weg desto schwächer wird das Licht



# Phong'sches Reflexionsmodell

- Nach: Bui-Tuong Phong, „Illumination for Computer Generated Pictures“, in Communications of the ACM, 1975
- Lichtquelle wird als Punktlichtquelle angenommen
- Die, in Richtung des Betrachters abgegebene Lichtintensität ergibt sich aus der Summe der drei Komponenten:
  - Diffuse Reflexion
  - Ambiente Beleuchtung
  - Spiegelnde Reflexion

## Phong'sches Reflexionsmodell:



Ambiente Reflexion



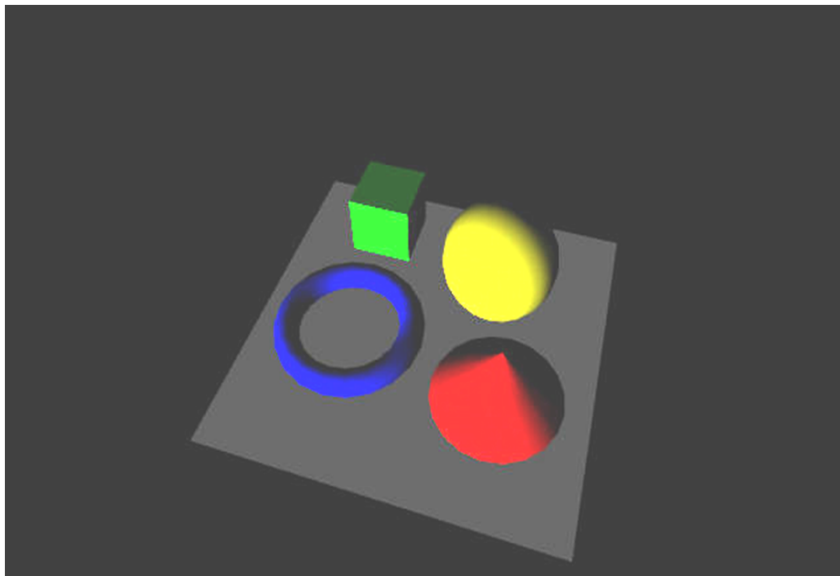
Ideal diffuse Reflexion



Ideal spiegelnde  
Reflexion

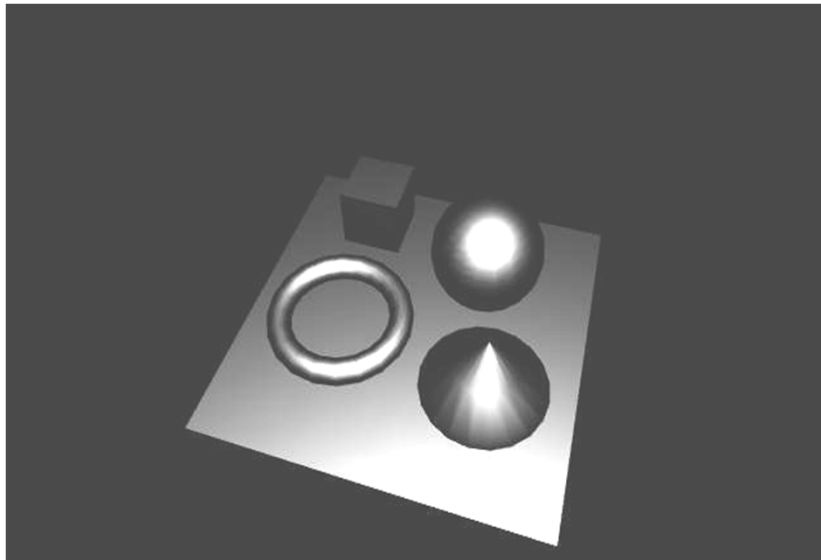


# Diffuse Reflexion



- Sehr matte Oberflächen sind perfekt diffus reflektierend.
- Diese Oberflächen haben unabhängig von der Position des Betrachters die gleiche Farbe und Helligkeit.
- Farbe der diffusen Reflexion ist Materialabhängig.

# Spiegelnde Reflexion



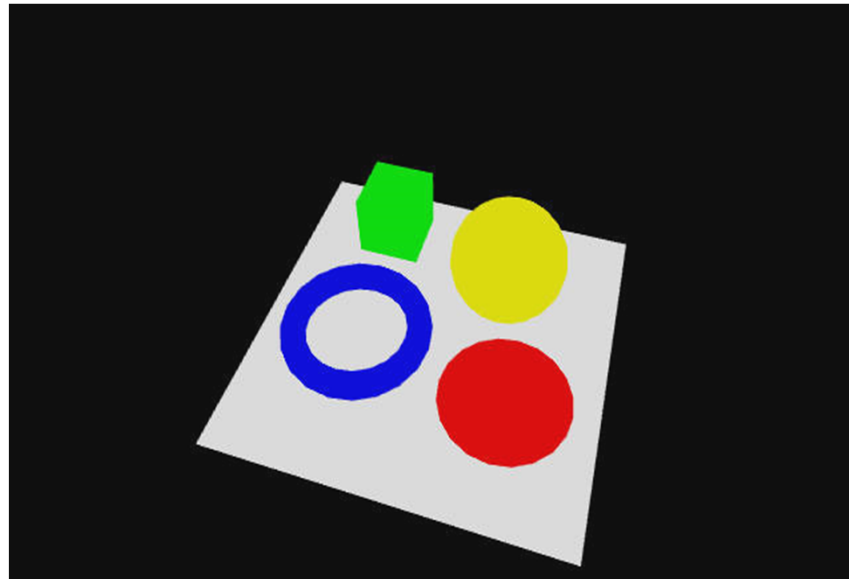
Im Gegensatz zum diffus reflektierenden Licht hat spiegelnd reflektierendes Licht nicht die Farbe der Oberfläche sondern die des Lichtes.

**Beispiel:** Ein grünes Objekt wird mit weißen Licht bestrahlt.

- Das diffus reflektierte Licht ist grün.
- Das spiegelnd reflektierte Licht ist weiß.

# Ambiente Materialbeleuchtung

- Die Ambiente Beleuchtung soll das Umgebungslicht simulieren.
- Sie ist nur von der Materialfarbe und der eingestellten Lichtintensität abhängig.



# Licht aus der Umgebung

- Oberflächen, die parallel zum einfallenden Licht liegen, haben keine diffuse Reflexion ...
- ... und erst recht keine spiegelnde Reflexion.
- D.h. sie müssten schwarz dargestellt werden.
- Realistisch ist aber, dass diese Flächen vom abgestrahlten Licht der Umgebung beleuchtet werden also auch sichtbar sind.

Simulation dieses Effektes durch Anwendung eines Umgebungslichts:

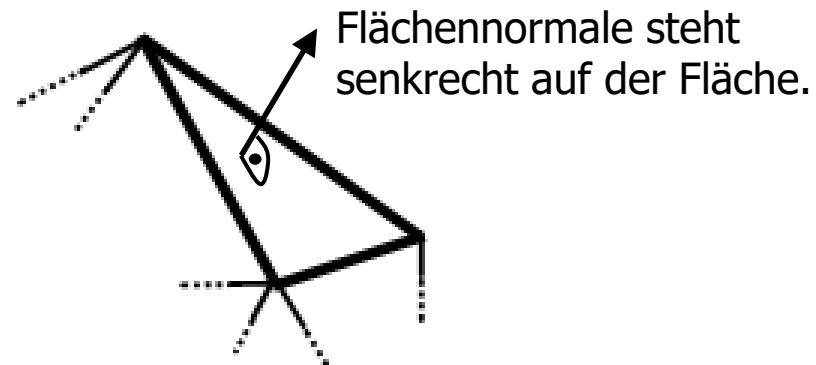
## **Der Ambienten Materialbeleuchtung**

# Ohne Normalen keine Beleuchtungsberechnung!

## 1. Variante: **Flächennormalen**



Ikosaeder



Auf welcher Stelle der Fläche „sitzt“ diese Flächennormale im Dreieck?



# Flat Shading

## Verfahren:

- Für jede Fläche gibt es nur eine Normale.
- Dadurch erhält jede Fläche (Facette) eine einheitliche Beleuchtung\*).
- D.h. jede Fläche besitzt **nur eine Farbe**.



## Nachteil:

- Da die meisten Objektoberflächen gekrümmt sind, ist die Qualität der Ergebnisse meist schlecht.

## Vorteil:

- sehr einfache und sehr schnelle Beleuchtungsberechnung!

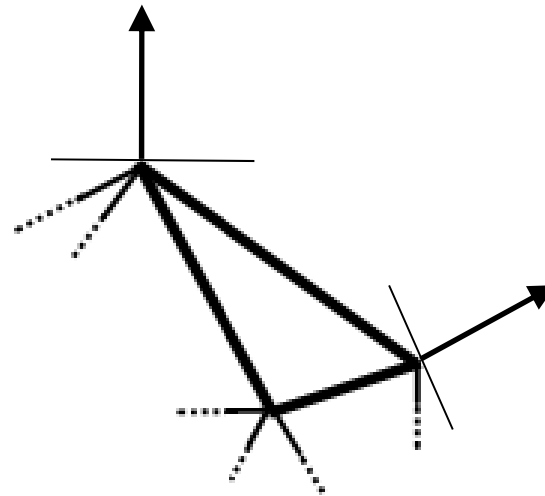
\*) Beleuchtung wird meist mit dem Phong'schen Beleuchtungsmodell berechnet.

# Verbesserung der Beleuchtungssimulation beginnt bei den Normalen!

## 2. Variante: **Eckennormalen**



Ikosaeder



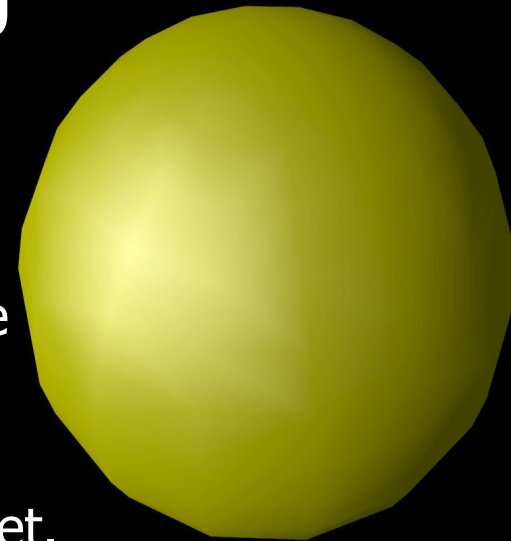
**Eckennormalen stehen senkrecht auf der Tangentialebene ...**

... und **NICHT** wie die Flächennormale senkrecht auf der Fläche!

Eckennormalen berechnet man durch die Interpolation der Flächennormalen der benachbarten Flächen.

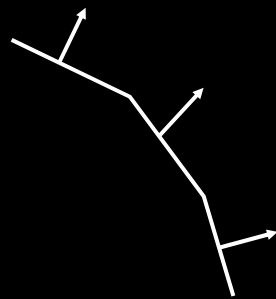
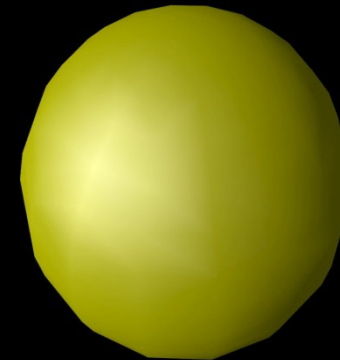
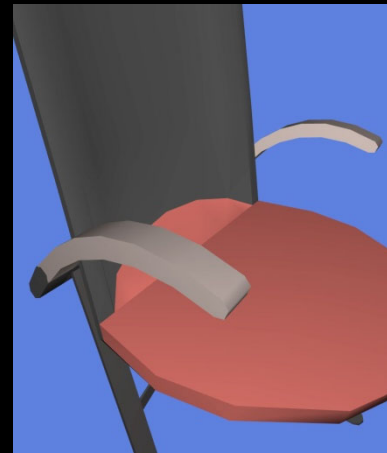
# Gouraud Shading

- Beleuchtungsberechnung \*) wird für **jeden** Eckpunkt durchgeführt.
- D.h. pro Dreieck werden drei unterschiedliche Farben (Helligkeiten) berechnet.
- Die Farben der Pixel innerhalb des Dreiecks werden durch bilineare Interpolation berechnet.
- Dadurch entsteht eine **optische Glättung** – **ohne** Änderungen an der Geometrie vornehmen zu müssen.
- Optische Glättung bewirkt ein „kaschieren“ der Kanten, die nicht zur Kontur gehören!



\*) Beleuchtung wird meist mit dem Phong'schen Beleuchtungsmodell berechnet.

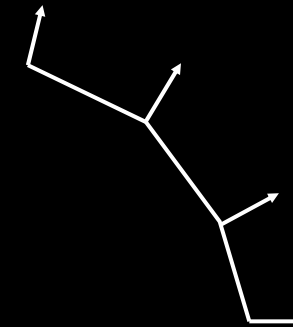
# Flat Shading versus Gouraud Shading



Ergebnis ist abhängig von  
den berechneten  
Normalen:

Links: Flächennormalen

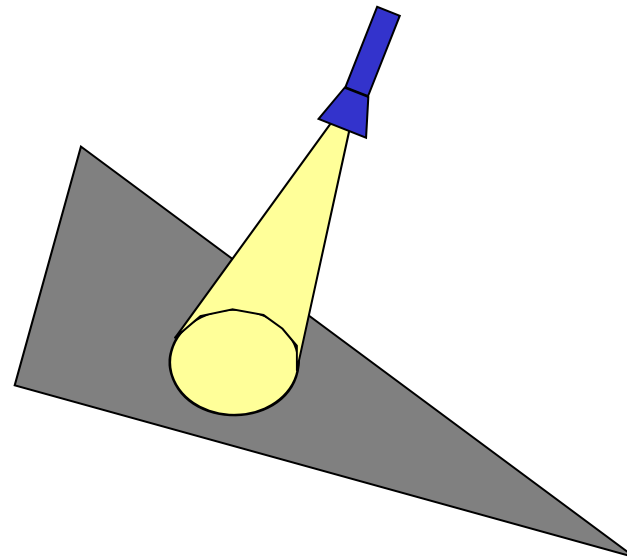
Rechts: Eckpunktnormalen



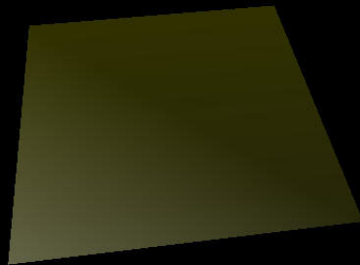
# Grenzen des Gouraud Shading Verfahrens

## Lichtquelle bestrahlt nur die Mitte eines Dreiecks:

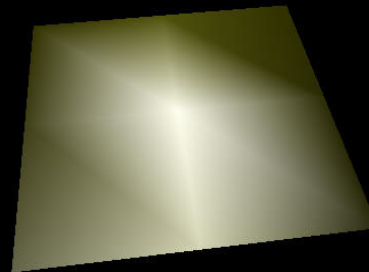
- Lichtberechnung wird nur an den Eckpunkten berechnet.
- Dadurch kann für das Dreieck keine Glanzlicht berechnet werden.
- Außer man macht folgendes ...



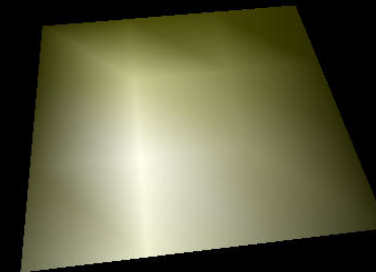
# Glanzlichter mit Gouraud Shading berechnen:



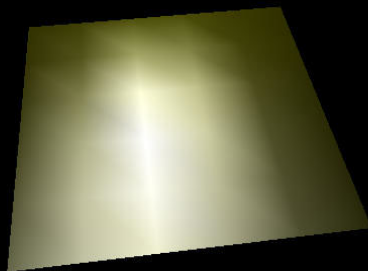
2 Dreiecke



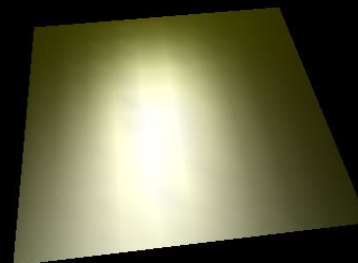
8 Dreiecke



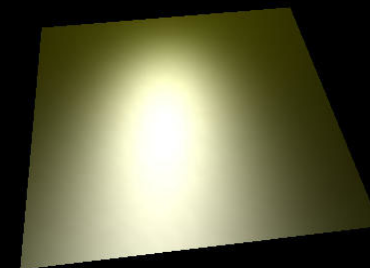
18 Dreiecke



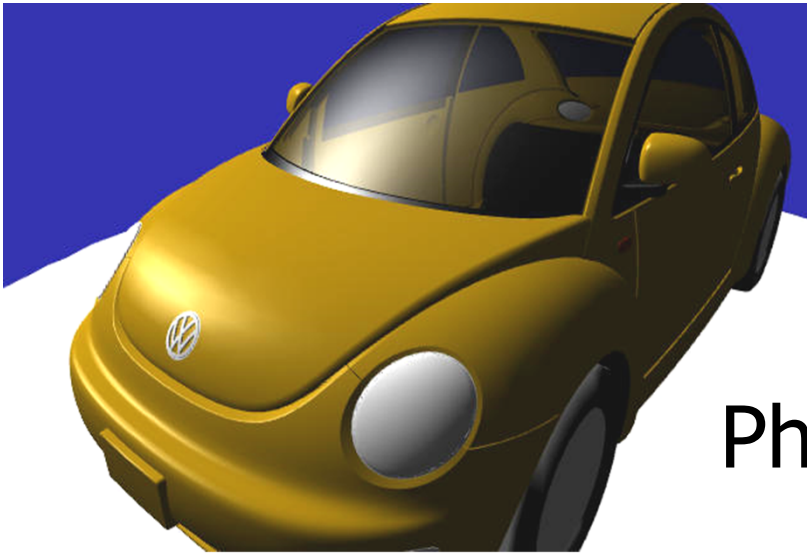
50 Dreiecke



200 Dreiecke



3200 Dreiecke



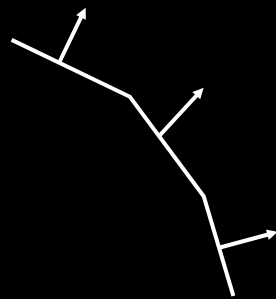
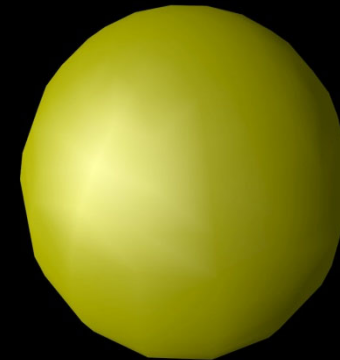
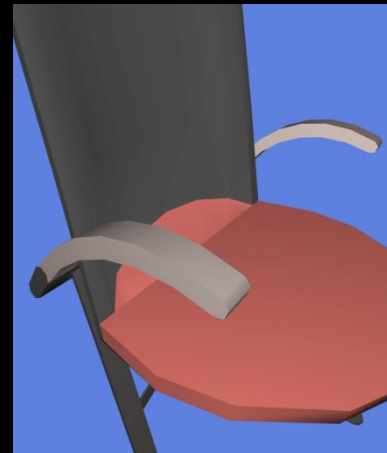
## Phong Shading

- Beleuchtungsberechnung wird für jeden einzelnen Pixel durchgeführt.
- Das bedeutet: Für jeden darzustellenden Pixel muss eine Normale berechnet werden!
- Die Normalen der Pixel werden auf Basis der Ecknormalen ermittelt oder durch eine Art Textur (Normal Map).
- Dadurch können Glanzlichter ohne zusätzliche Unterteilung (Tessilierung) des Gitters berechnet werden.

**Vorteil:** Glanzlichter & gutes visuelles Ergebnis  
Basisverfahren für Bump- und Normal Mapping

**Nachteil:** Normale für jeden Pixel interpolieren kostet viel Rechenzeit

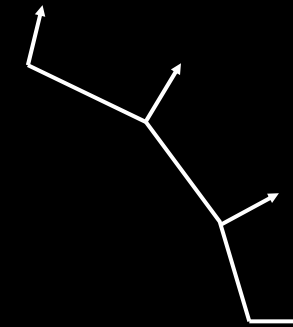
# Flat Shading versus Gouraud Shading



Ergebnis ist abhängig von  
den berechneten  
Normalen:

Links: Flächennormalen

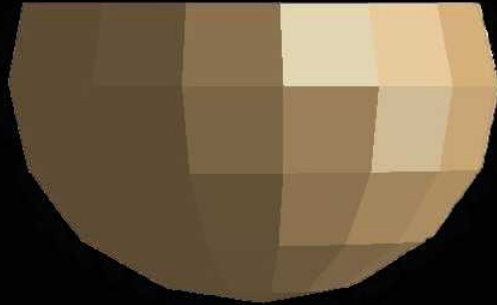
Rechts: Eckpunktnormalen



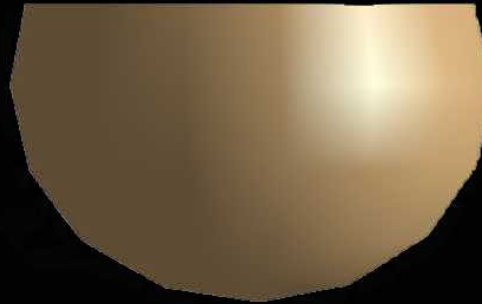


# Vergleich der Verfahren

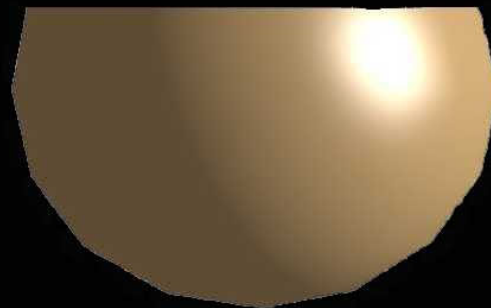
Flat Shading



Gouraud Shading



Phong Shading



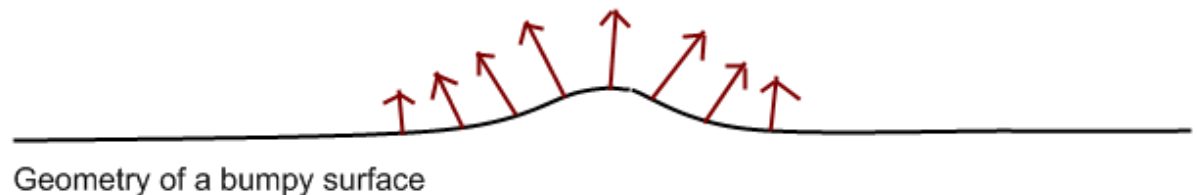
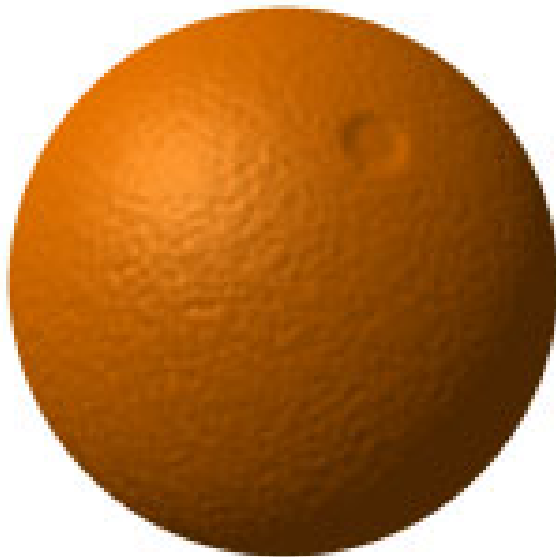


Flat-, Gouraud- und Phong-Shading

Von: [www-2.cs.cmu.edu/~ph/nyit/](http://www-2.cs.cmu.edu/~ph/nyit/)

# Phong Shading als Basisverfahren zur optischen Modellierung einer Oberfläche

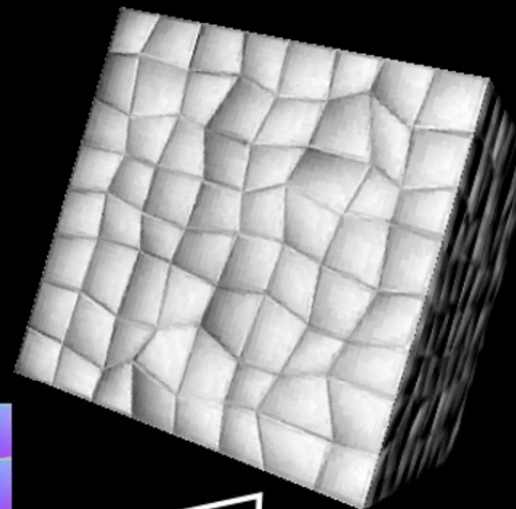
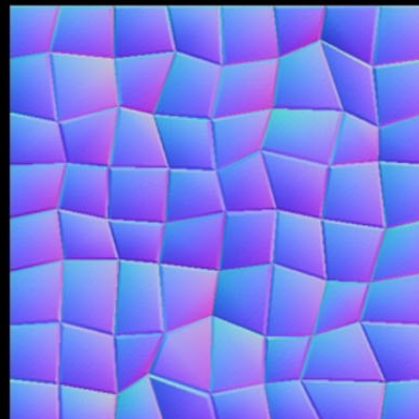
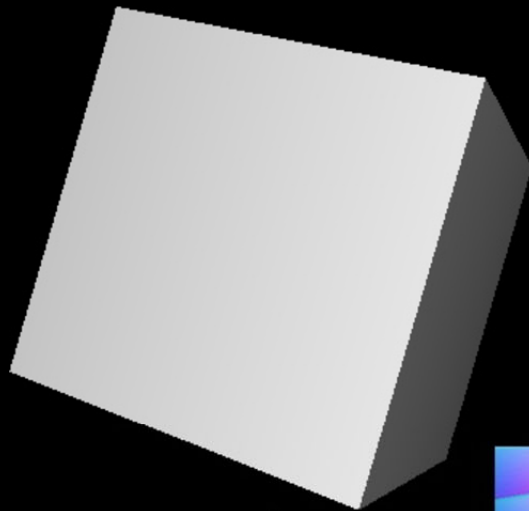
Die Geometrie der Oberfläche wird dabei nicht verändert.



\*) Bump Mapping funktioniert eigentlich ein bisschen anders – es variiert die Länge der Normalen und täuscht so Vertiefungen vor. Das hier gezeigte Verfahren heißt Normal Mapping.

Bilder aus: [http://mathforum.org/mathimages/index.php/Bump\\_Mapping](http://mathforum.org/mathimages/index.php/Bump_Mapping)

# Beispiel für Normal Mapping

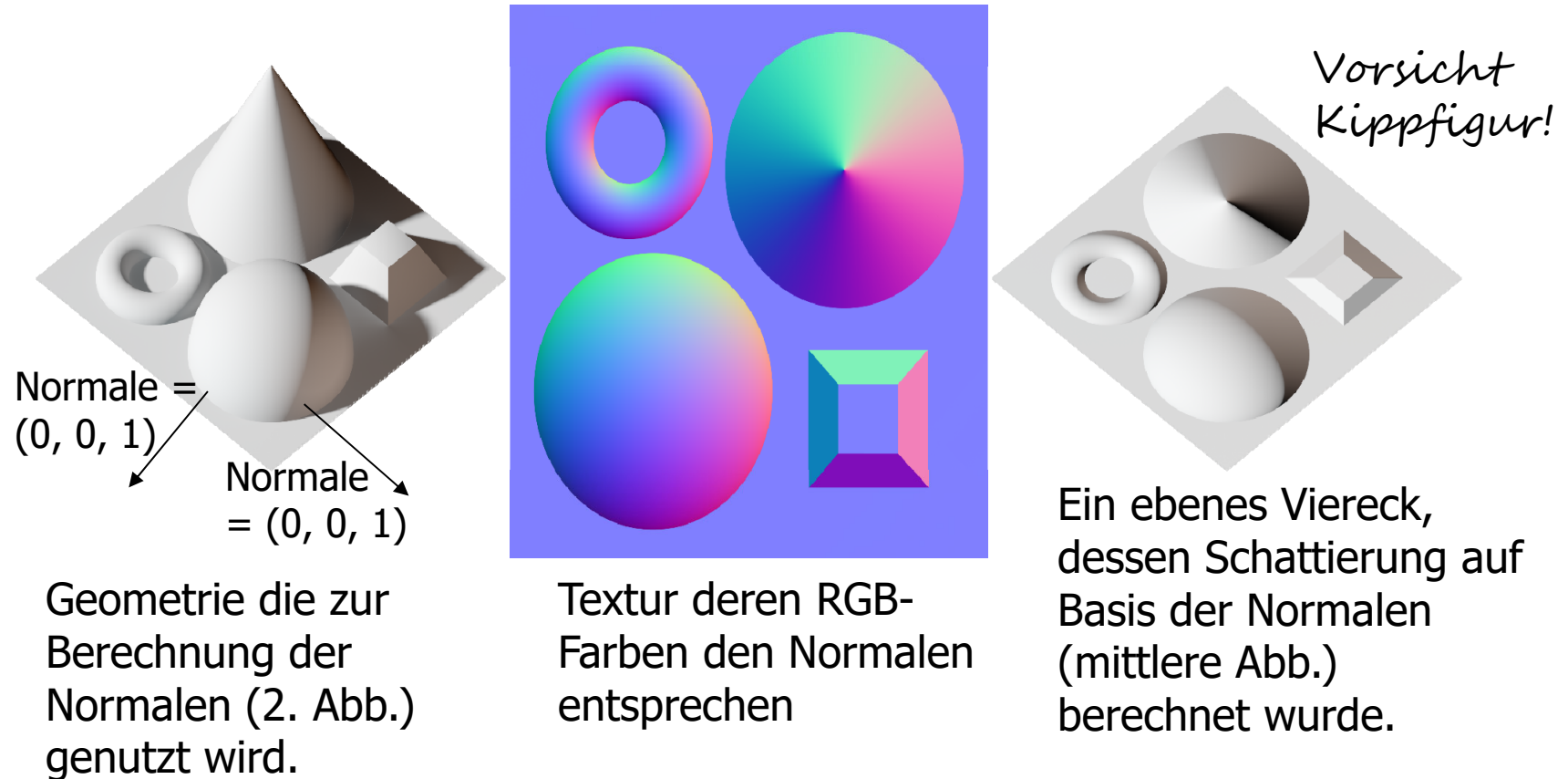


Würfel, dessen Schattierung (Beleuchtungsreflektion) auf Basis der Normalen (unten) berechnet wurde. Die Geometrie des Würfels wurde dabei nicht verändert.

Textur in der die Normalen pro Pixel als RGB-Farben gespeichert sind

Aus: <http://www.independentdeveloper.com/archive/tag/normal-mapping>

# Normal-Mapping am Beispiel einer komplett ebenen Fläche

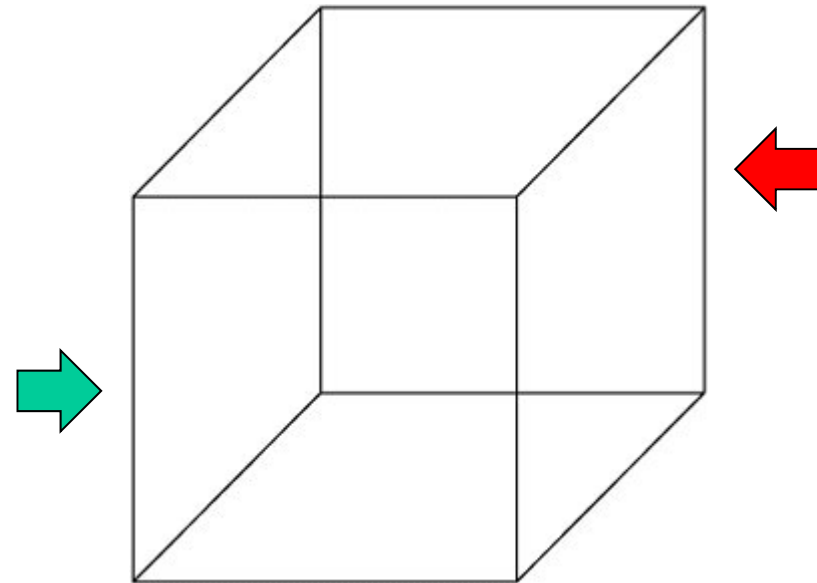


Aus Wikipedia: Normal Mapping

## Weitere Beispiele für bekannte Kippfiguren ...



Junges Mädchen  
oder alte Frau?



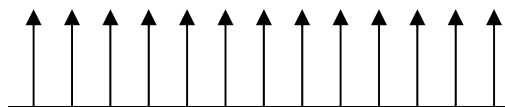
### **Neckerwürfel**

Einmal bildet das vordere Viereck die Oberfläche des Würfels (neben grüner Pfeil) und einmal das hintere Viereck (neben rotem Pfeil).

# Übungsaufgabe (Klausur WS 16/17)

## Phong Shading

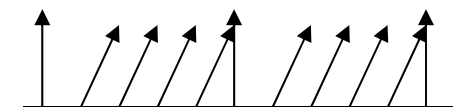
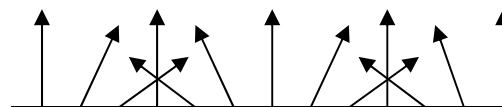
In den drei Abbildungen unten sehen Sie jeweils einen Querschnitt durch eine Oberfläche. Die Pfeile symbolisieren die Flächennormalen pro Pixel. Durch die anschließende Beleuchtungsberechnung bspw. durch Phong Shading wird die Oberfläche optisch modelliert. Zeichnen und beschreiben Sie den visuellen Eindruck (ebenfalls im Querschnitt), den man nach der Beleuchtungsberechnung aufgrund der eingezeichneten Flächennormalen von der Geometrie haben wird.



Beispiel:

\_\_\_\_\_

*Geometrie erscheint eben.*



# Vergleich der Verfahren



Aus: Watt A., Policarpo F.: The Computer Image, Addison-Wesley 1998

Nicht berechnet werden kann mit den hier vorgestellten Shading Verfahren:

- Schattenwurf,
- transparente Oberflächen,
- sowie spiegelnde Oberflächen

} ...das können nur globale Verfahren



# Verfahren zum Rendering

## Lokale Verfahren:

- Berücksichtigt bei der Berechnung des von der Objektoberfläche reflektierten Lichts nur das von der definierten Lichtquelle einfallende Licht (primäre Lichtquellen).
  - Die Beleuchtung der Objekte wird nicht beeinflusst durch:
    - Spiegelung,
    - Lichtreflexionen von anderen Objekten,
    - und Schattenwurf
- } sekundäre Lichtquellen

# Was sind primäre und was sind sekundäre Lichtquellen?



# Verfahren zum Rendering

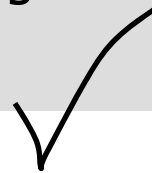
## **Globale Verfahren:**

- Objekte der Szene beeinflussen sich gegenseitig.
- In die Beleuchtungsberechnung gehen neben den primären Lichtquellen auch die sekundären mit ein.
- D.h. in der Beleuchtungsberechnung werden berücksichtigt:
  - Spiegelung,
  - Schattenwurf und
  - Lichtreflexion anderer Objekte.

# Verfahren zum Rendering

## Lokale Verfahren

- Wenig rechenaufwendig - schnelle Berechnung
- Beispiele: Flat Shading  
Gouraud Shading  
Phong Shading



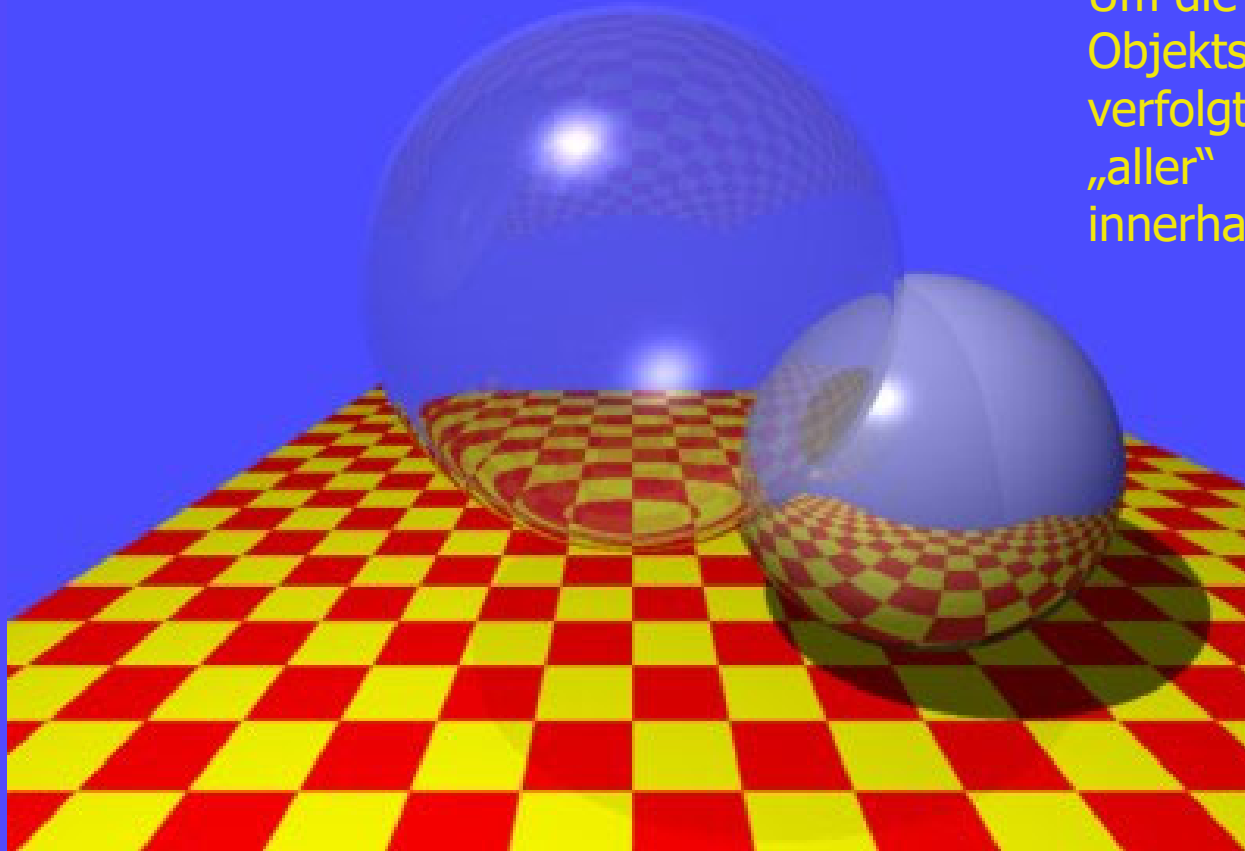
## Globale Verfahren

- Sehr rechenaufwendig
- Beispiele: Ray Tracing  
Radiosity

# Globales Verfahren: Ray Tracing

## Idee beim Ray Tracing:

Um die Beleuchtung eines Objekts zu ermitteln, verfolgt man den Weg „aller“ Lichtstrahlen innerhalb der Szene



Watt A., Policarpo F.  
The Computer Image  
Addison-Wesley 1998

# Verfolgung der Lichtstrahlen:

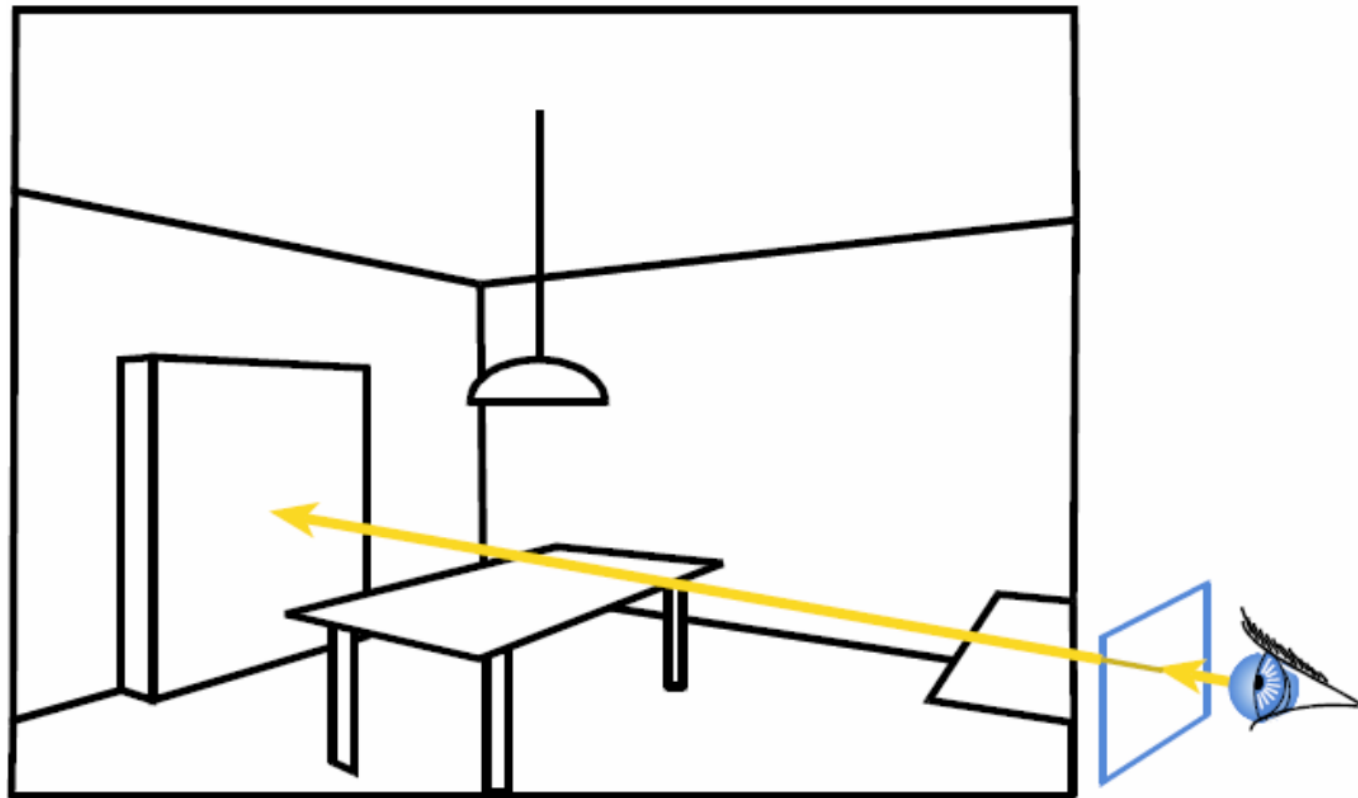
## Forward Ray Tracing:

- Von der Lichtquelle ausgehend werden die Lichtstrahlen bis ins „Auge des Betrachters“ (COP = Center of Projection) verfolgt.
- **Zu aufwändig:** da die meisten Strahlen das COP nicht erreichen.

➤ Also: Erstmal gucken, was beim Betrachter überhaupt ankommt ;-)

# Ray Tracing

Statt der Lichtstrahlen werden die Sehstrahlen verfolgt:



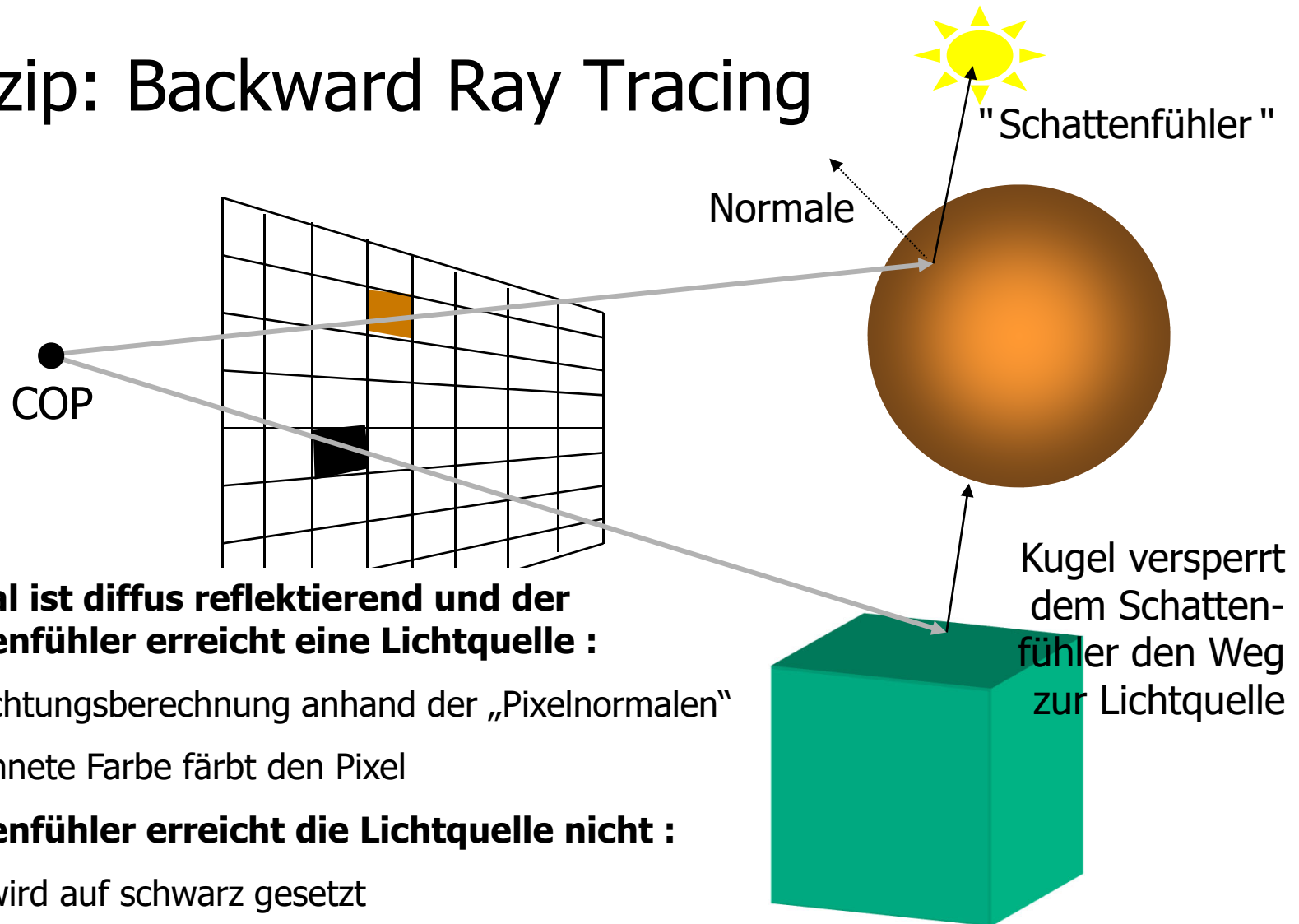
MIT EECS 6.837, Cutler and Durand

# Ray Tracing

- **Backward Ray Tracing wird allgemein als Ray Tracing bezeichnet!**
- Nur für die Stellen, die der Betrachter tatsächlich sieht wird eine Beleuchtungsberechnung durchgeführt:
- Statt Lichtstrahlen verfolgt man „Sehstrahlen“; im Prinzip eine Rückwärtsverfolgung der Lichtstrahlen
- Also vom Auge des Betrachters (COP) werden durch die Projektionsebene hin zum Objekt „Sehstrahlen“ verschickt.



# Prinzip: Backward Ray Tracing



**Material ist diffus reflektierend und der Schattenfühler erreicht eine Lichtquelle :**

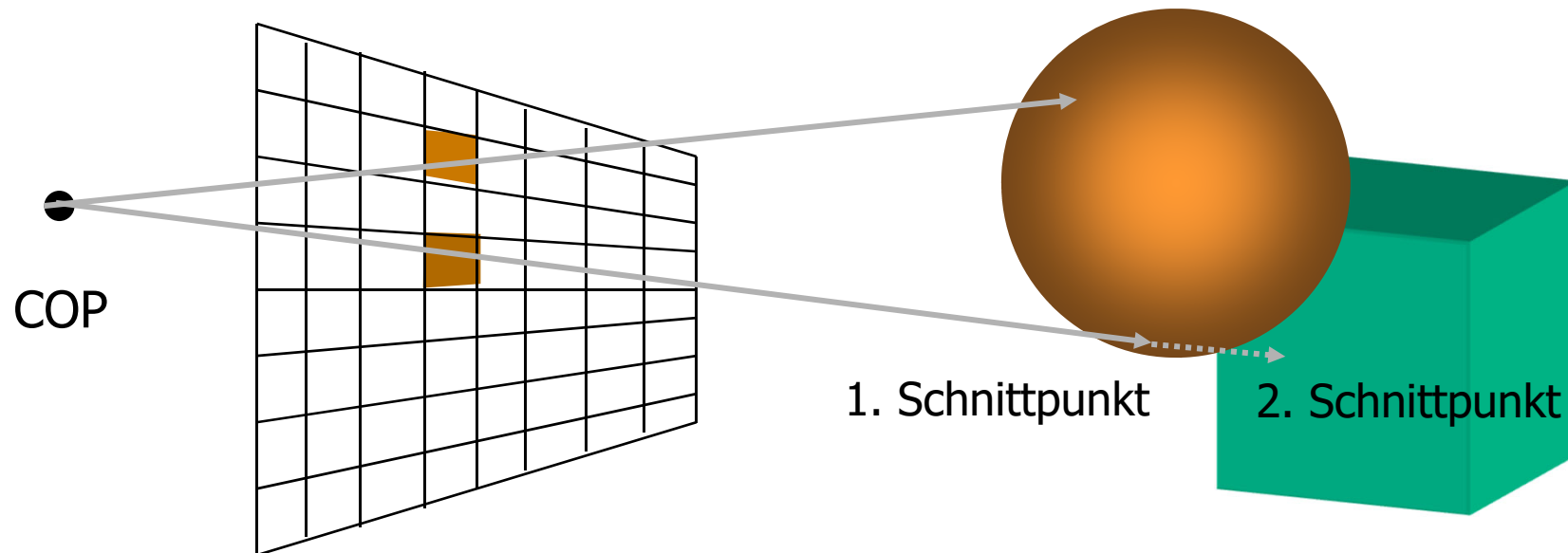
- Beleuchtungsberechnung anhand der „Pixelnormalen“
- Berechnete Farbe färbt den Pixel

**Schattenfühler erreicht die Lichtquelle nicht :**

- Pixel wird auf schwarz gesetzt

# Prinzip: Backward Ray Tracing

Die Verdeckungsrechnung erfolgt also nebenbei, ganz „automatisch“



Bisher beeinflussen sich die Objekte noch nicht gegenseitig.

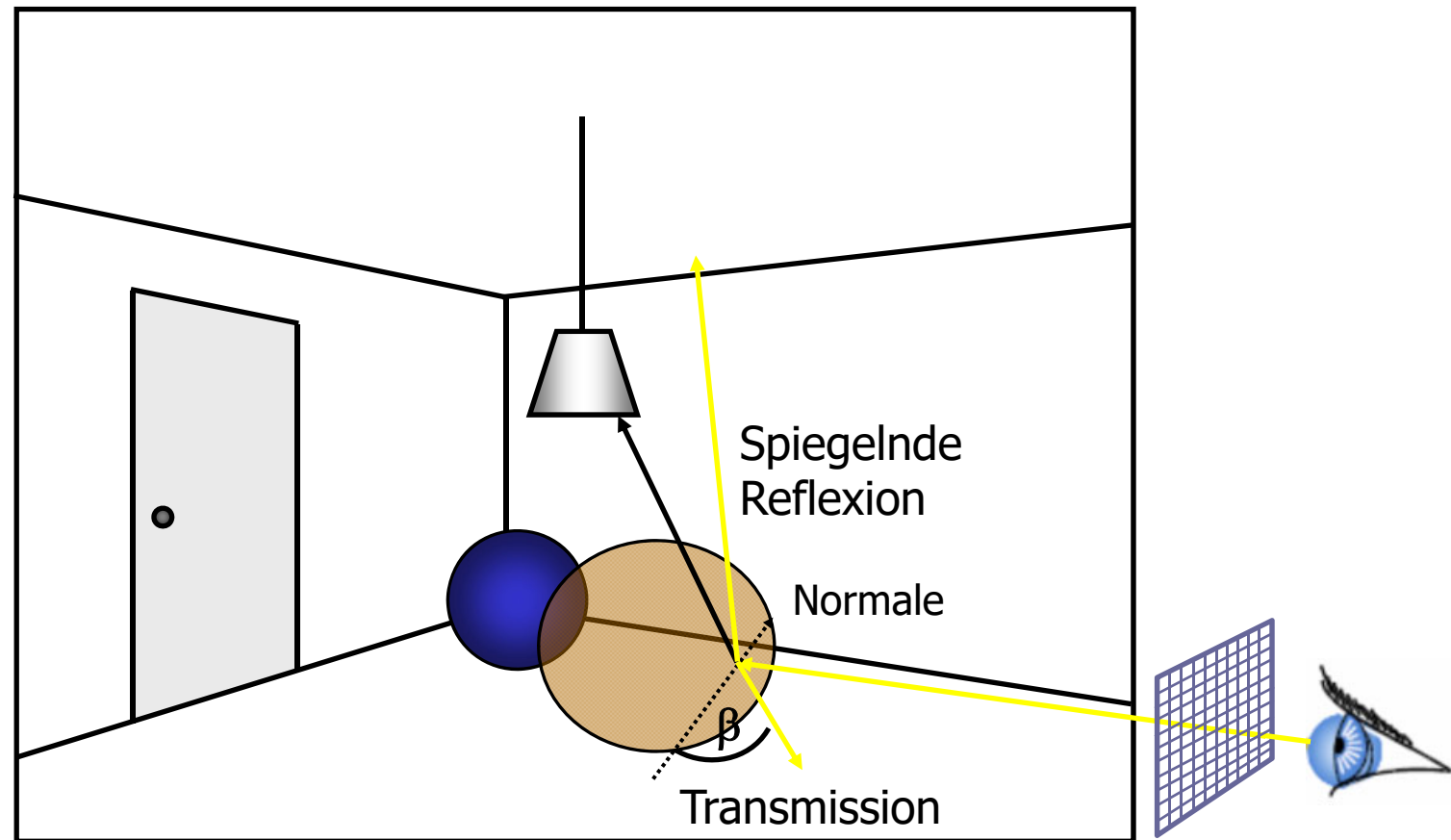
# Ray Tracing Prinzip

Wenn der getroffene Schnittpunkt nur diffus reflektierend ist, wird die Sehstrahlverfolgung (nach dem Schattentest & der Beleuchtungsberechnung) abgebrochen.

## **Was passiert aber wenn...**

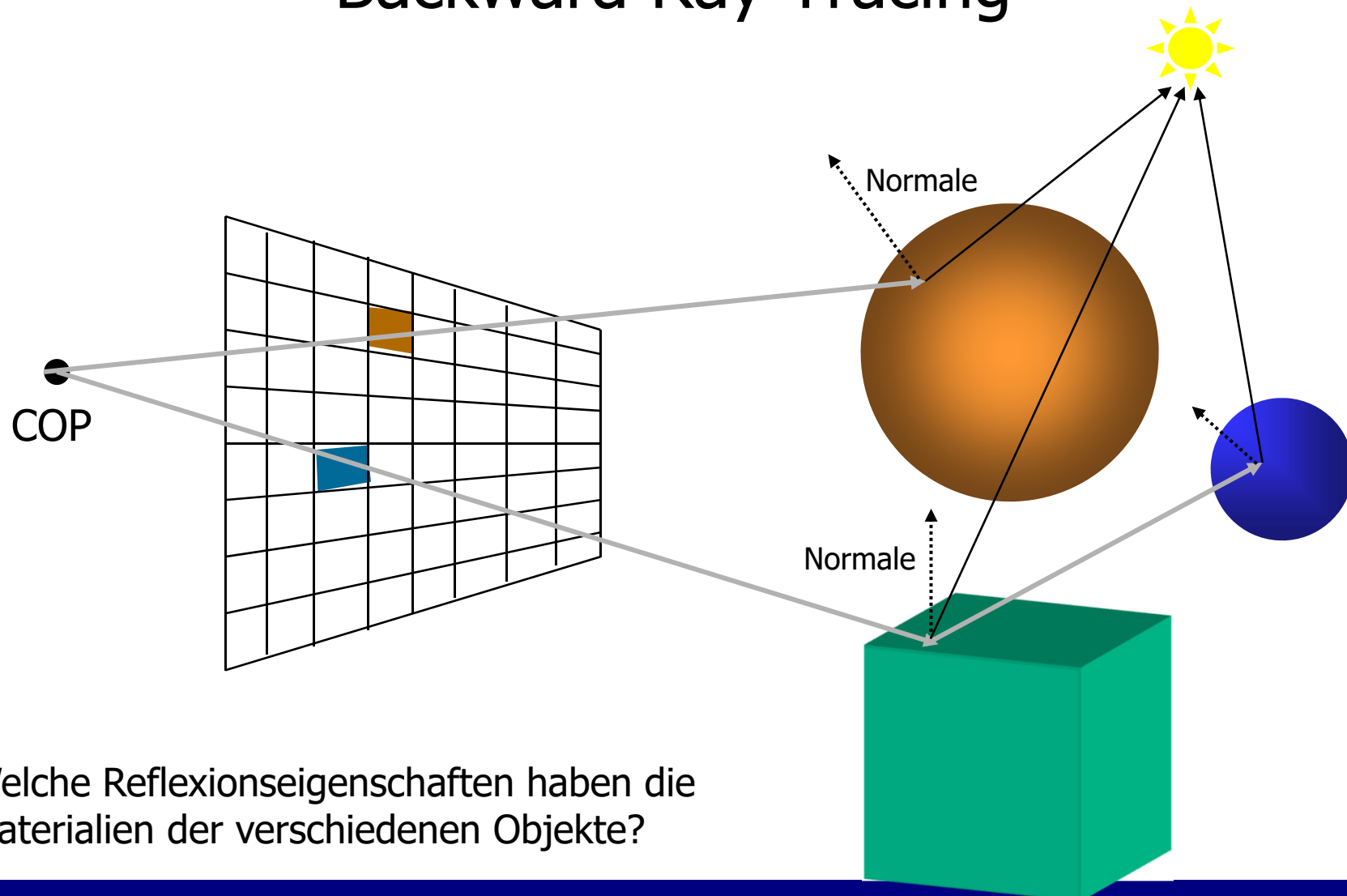
- ... der getroffener Punkt auf einer spiegelnd reflektierenden Oberfläche liegt ...
- ... oder die Oberfläche transparent ist?

# Ray Tracing Prinzip



$\beta$  = Brechungswinkel

# Backward Ray Tracing




Welche Reflexionseigenschaften haben die Materialien der verschiedenen Objekte?



Beispiel: Backward Ray Tracing

# Beleuchtungsberechnung beim Ray Tracing

## Farbe des Pixels setzt sich zusammen aus:

- ambienten Umgebungslicht (globaler Wert)
  - diffuser Reflektion (Primärstrahl)
  - spiegelnder Reflektionen
  - Transmissionen
- } Sekundärstrahlen
- 

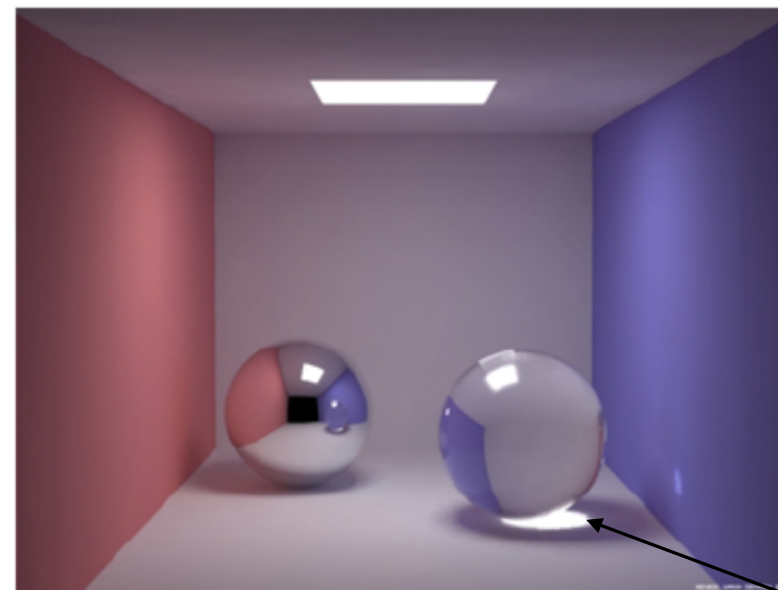
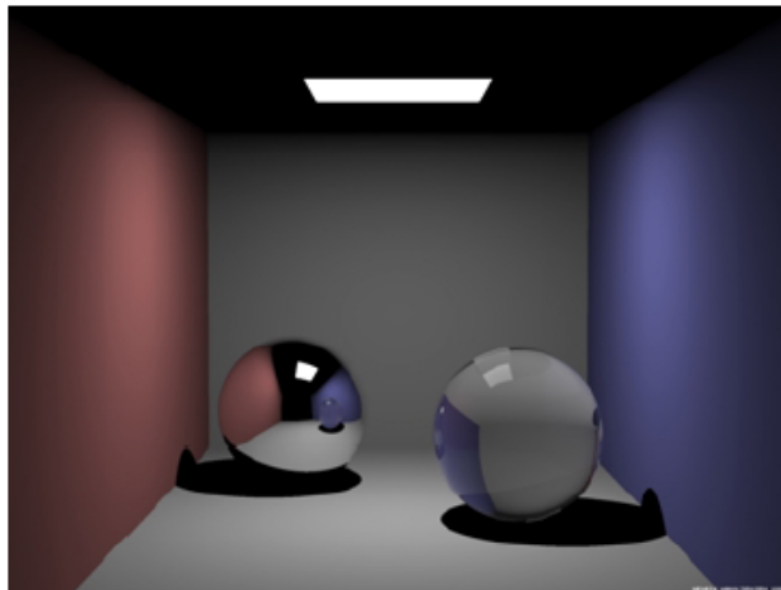
Anmerkung: Material ist meist nicht nur spiegelnd oder nur diffus reflektierend!

## Nicht berücksichtigt werden:

- Diffuse indirekte Beleuchtung die von benachbarten Objekten ausgeht!

## Einschränkungen des Ray Tracing Verfahrens

- Keine "weichen" Schatten
- Keine indirekte diffuse Beleuchtung der Objekte durch andere Objekte der Szene
- Keine "Caustics"



Images courtesy of  
Henrik Wann Jensen  
University of California  
San Diego

© 2004 Tomas Akenine-Möller

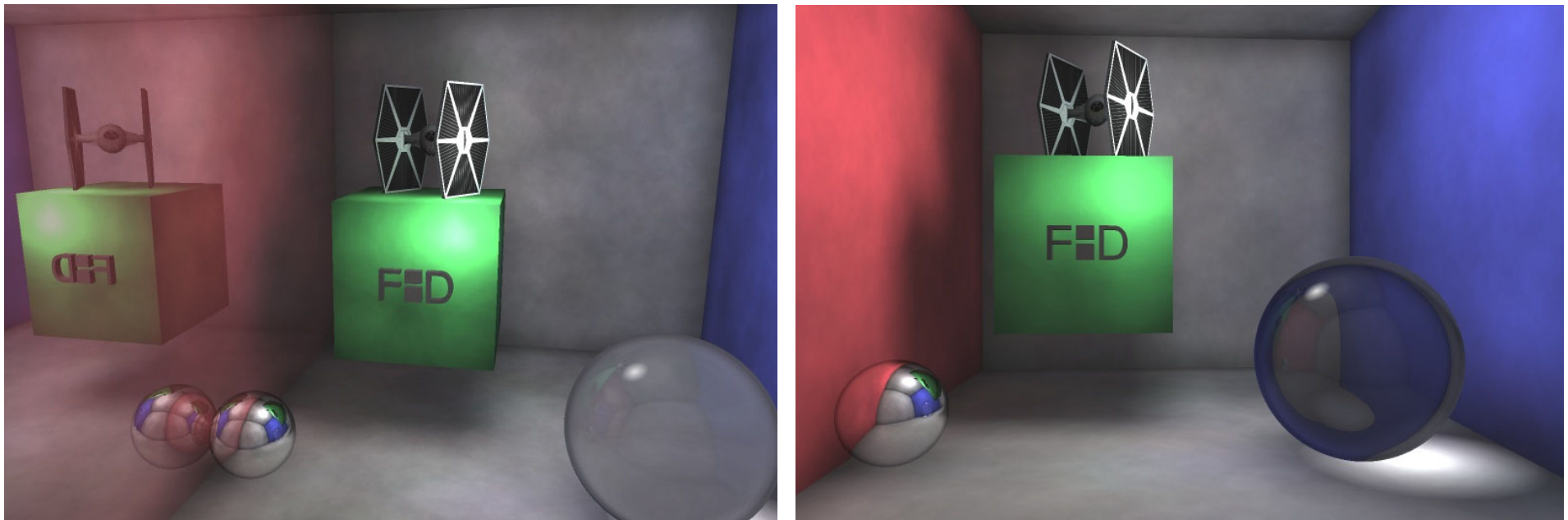
Caustics



Erweiterung des Ray-Tracing Verfahrens zur  
Berechnung von Caustics und diffuser Reflexion:

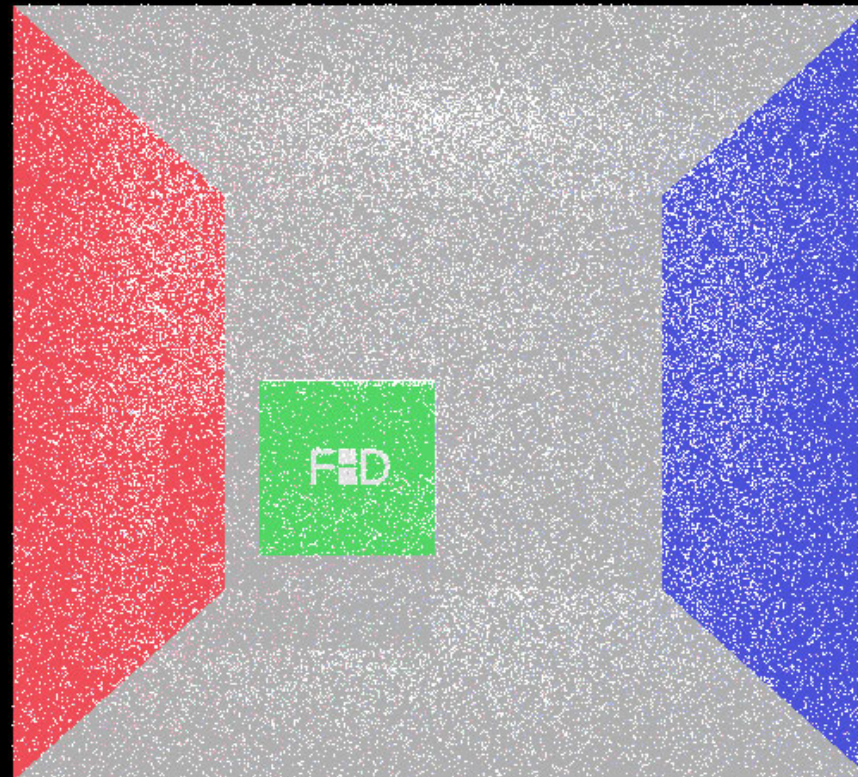
## „Photon Mapping“ (H. W. Jensen)

Ergebnisse der Masterarbeit von Tobias Geis (an der FHD)



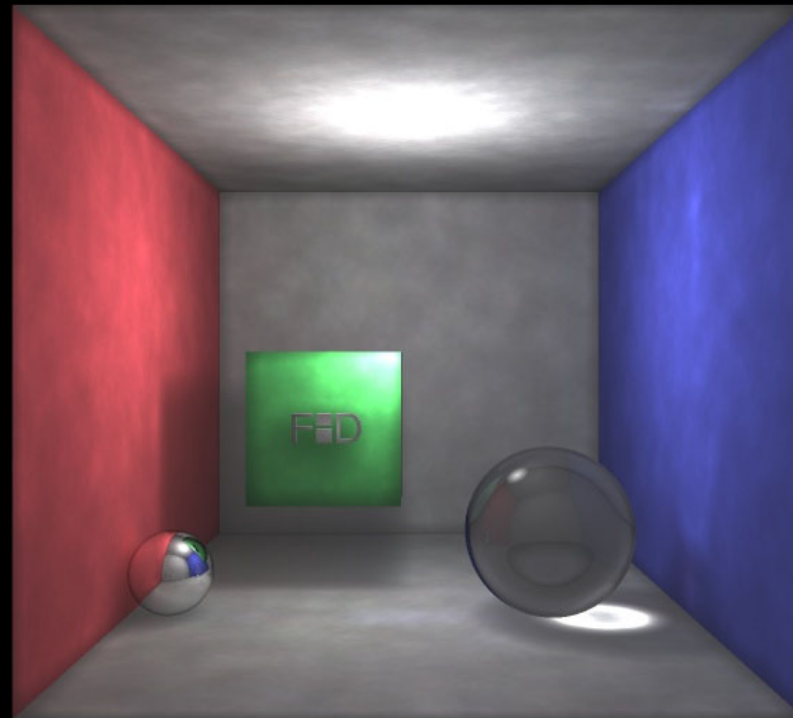
Diffuse Beleuchtung und Caustic-Berechnung werden durch die Verwendung von „Photonen“ ins Ray-Tracing integriert

Photonmap



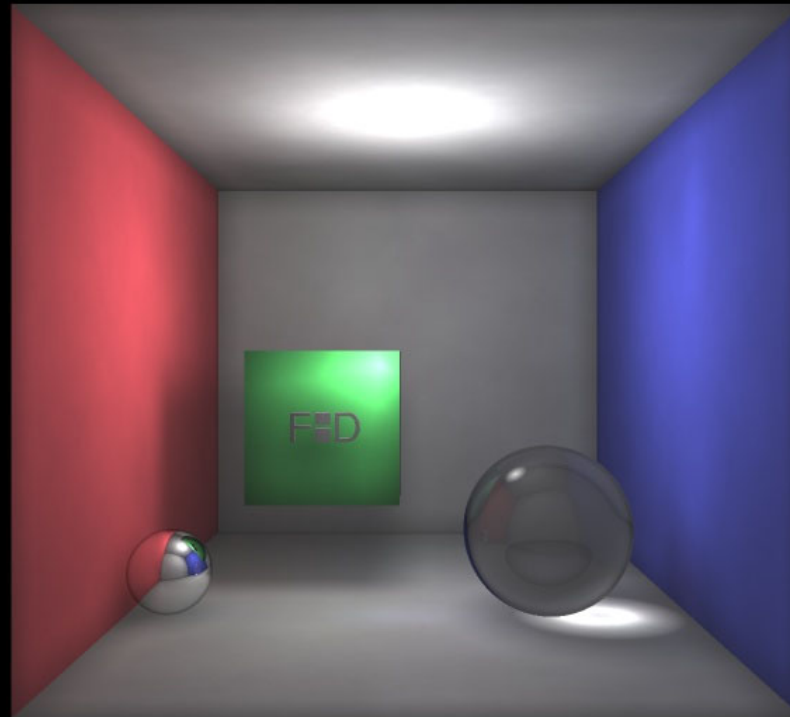
# Ergebnisse der Masterarbeit von Tobias Geis (an der FHD)

Zur „Integration“ wurden **500 000** Photonen verwendet



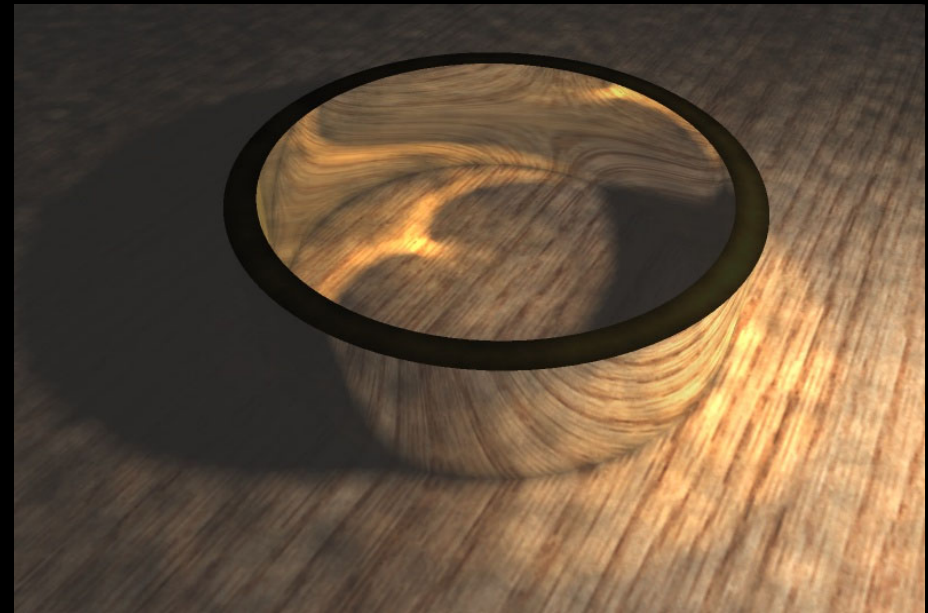
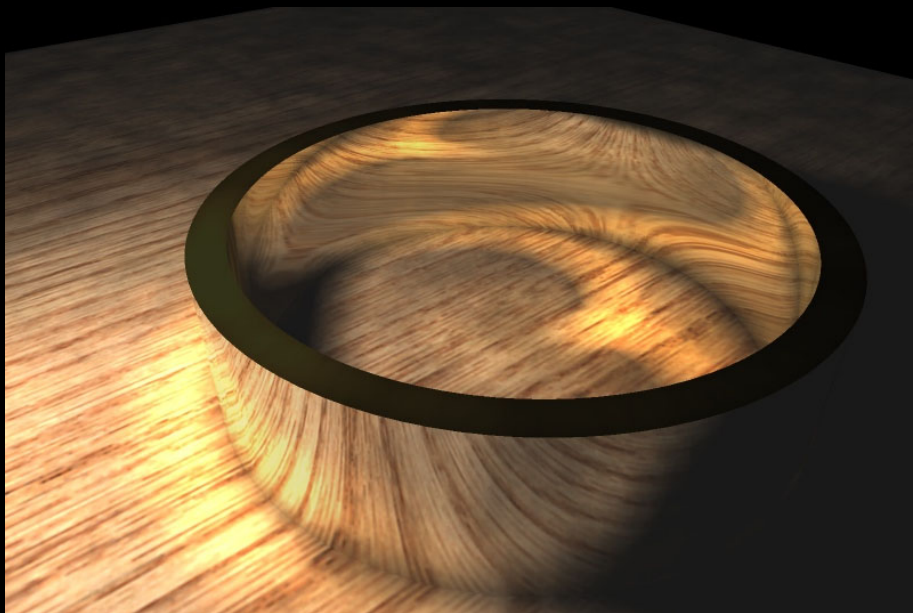
# Ergebnisse der Masterarbeit von Tobias Geis (an der FHD)

Nach der Optimierung der Photonmap (500 000 Photonen):



# Ergebnisse der Masterarbeit von Tobias Geis (an der FHD)

.... einer von zwei Preisträgern des Fachbereichspreis im WS 04/05



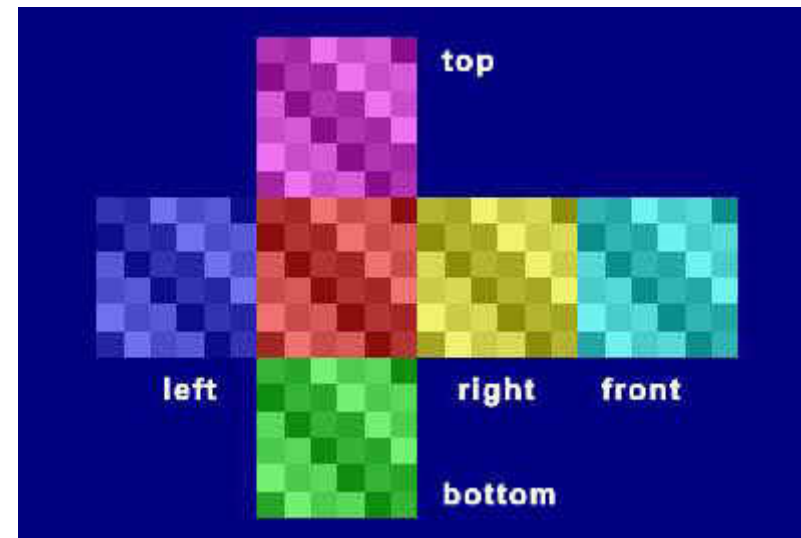
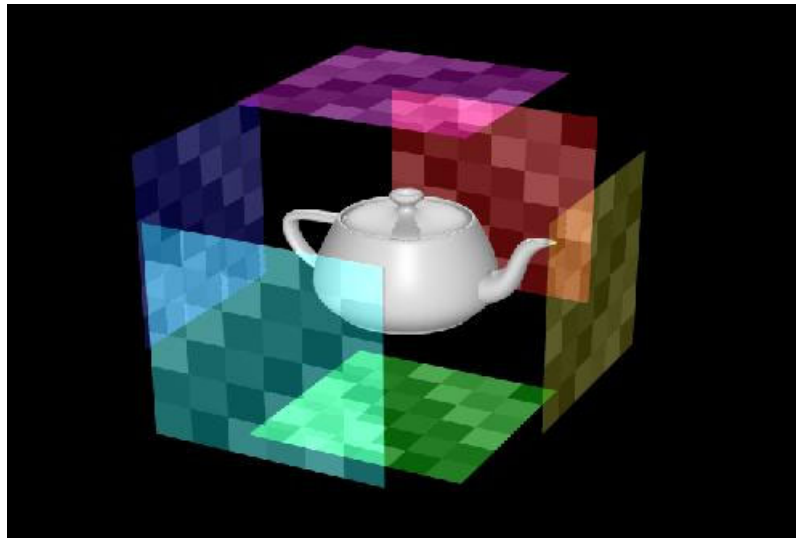


Ray Tracer aus Computer Graphics WS 17/18 von Christian Gottwein

Welche Methoden nutzte man  
alternativ zur Echtzeitberechnung  
von Spiegelungen?

# Cubic Environment Map

- Umgebung wird auf eine Würfel abgebildet (Environment Map).
- Spiegelnde Reflexion wird anhand des „Texel“ (=Texturpixel) ermittelt auf den der Reflexionsvektor zeigt.
- Was ist der Reflexionsvektor?

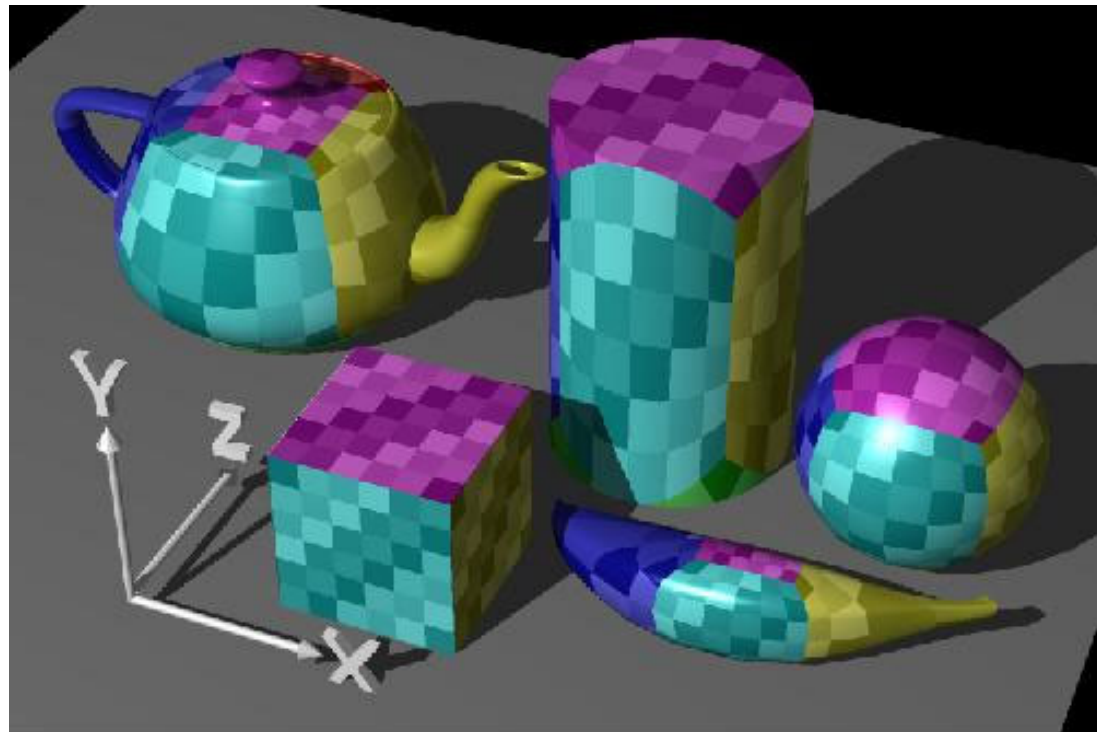
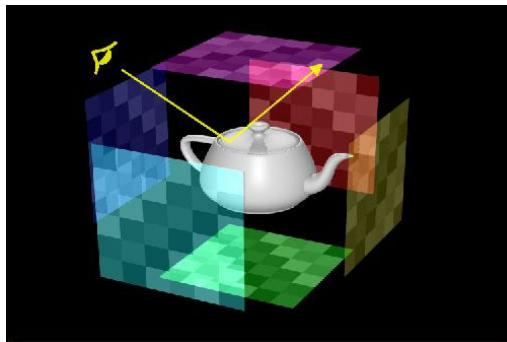


Alle Bilder aus [2]



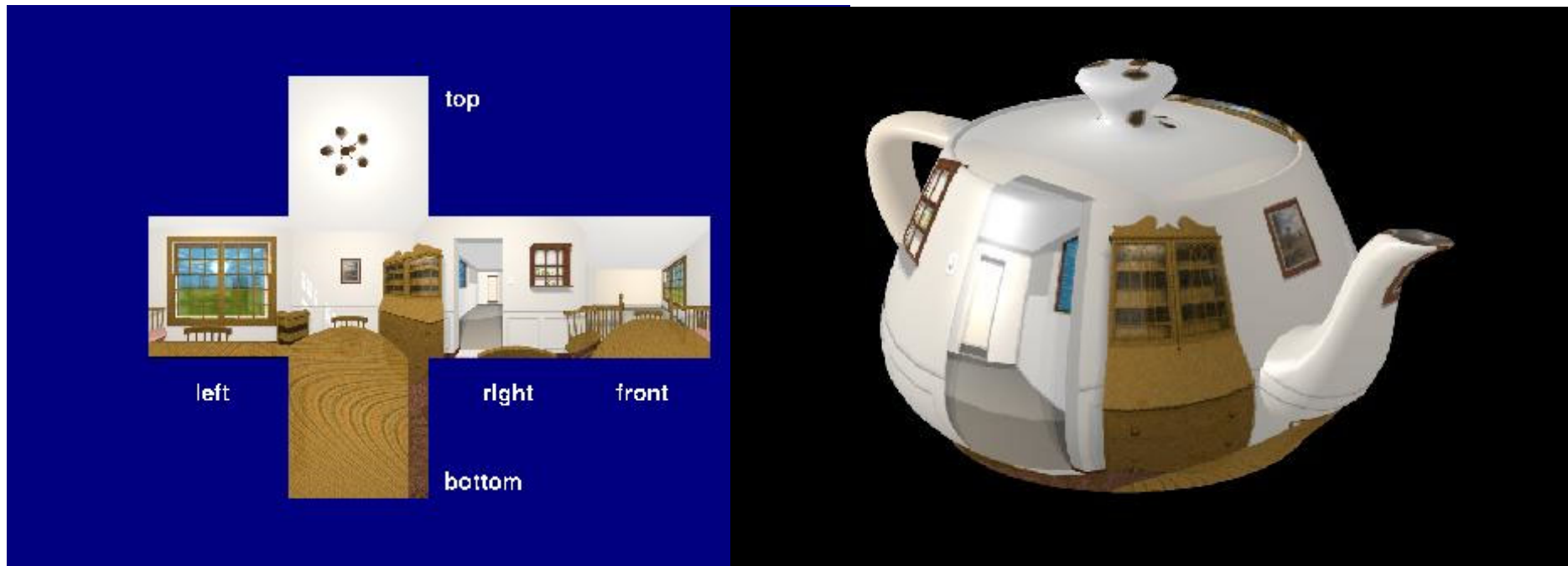
# Cubic Environment Map

Cubic Environment Map um verschiedene Objekte



Alle Bilder aus [2]

# Cubic Environment Map



Alle Bilder aus [2]

# Sphere Mapping

- Environment Map ist im Regelfall eine Photographie aufgenommen mit extrem niedriger Brennweite.
- Man nennt sie auch „Light Probe“ (siehe Abbildung)



# Sphere Mapping

Verschiedene Sphere Maps (Light Probes):



Beachte: Sphere Maps sind Blickpunkt abhängig!

# Cubic und Sphere Mapping im Vergleich

