

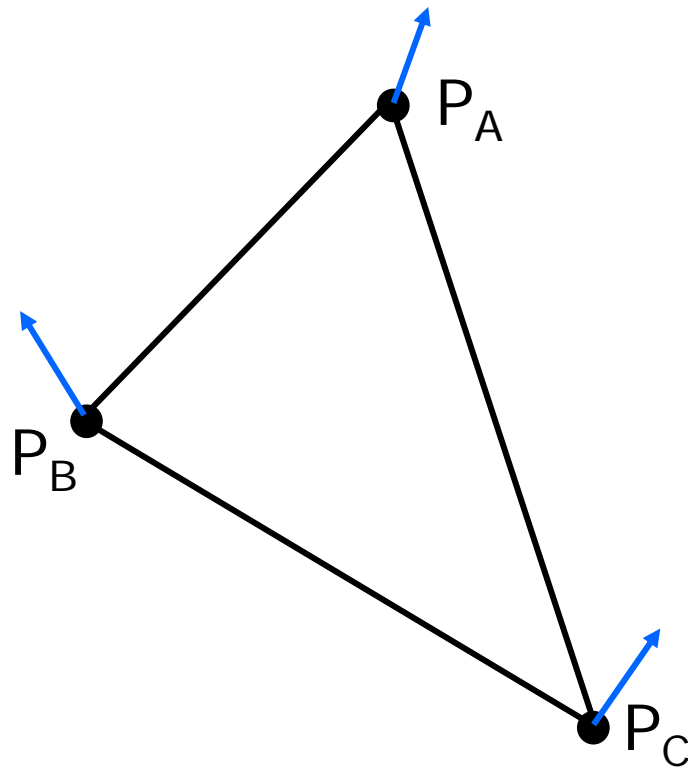
Computergrafik 2010

Oliver Vornberger

Kapitel 18:
Beleuchtung

Ausgangslage

am Ende der Viewing Pipeline liegt vor:



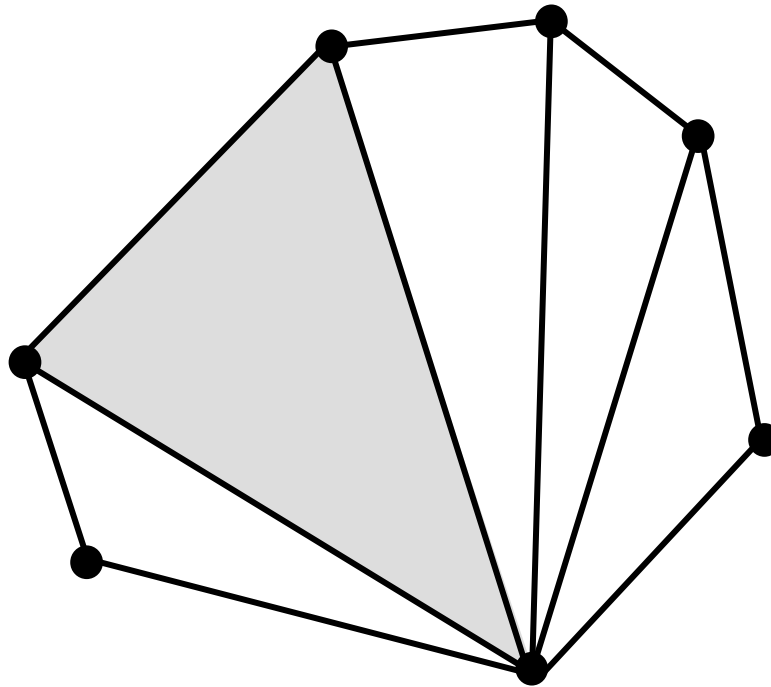
Materialeigenschaften

Beleuchtungsmodelle

- lokal:
Objekt, Lichtquellen, Augenpunkt
- global:
Objekt, Lichtquellen, Augenpunkt,
alle anderen Objekte

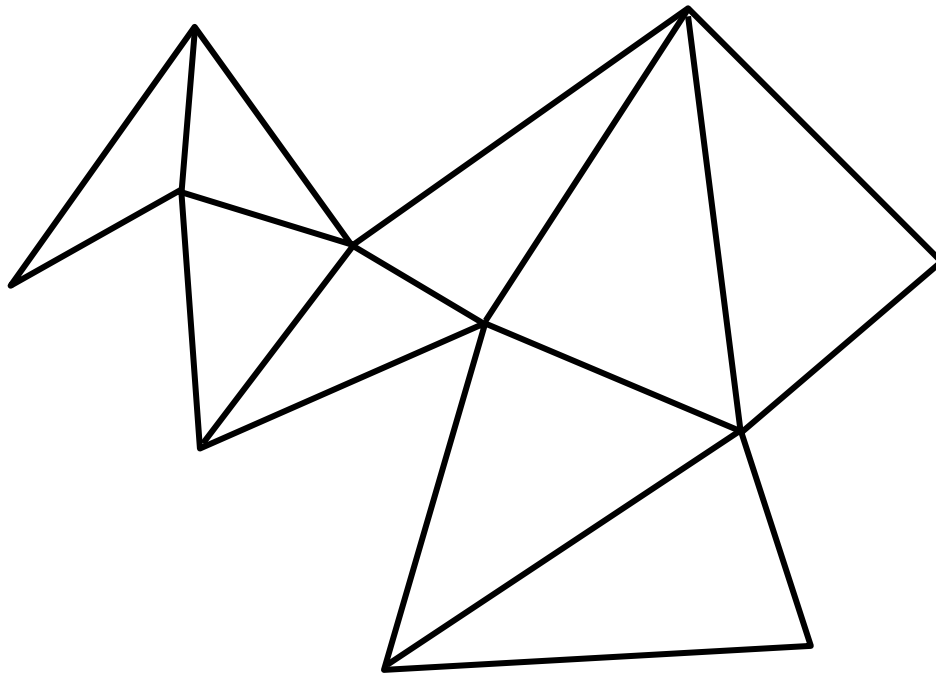
Triangulierung

konvexe Polygone triangulieren



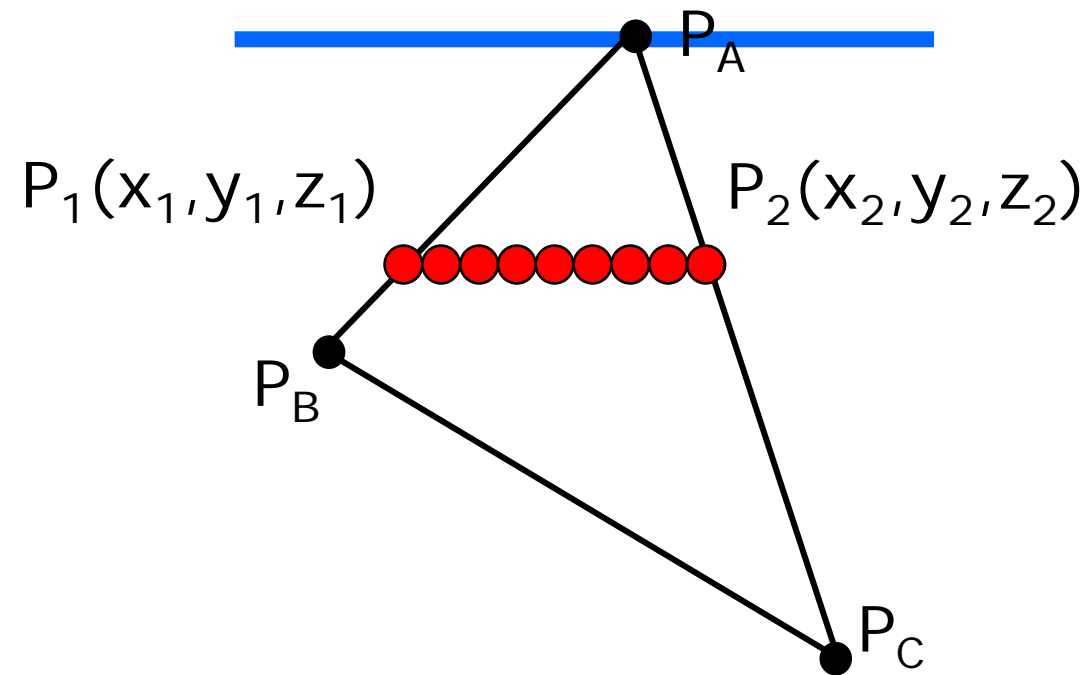
Triangulierung

konkave Polygone triangulieren



Ziel

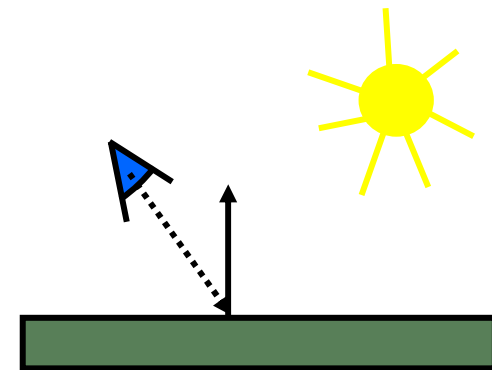
zeilenweises Rastern



Einfärbung/Beleuchtung

zur Berechnung der Farbe eines Pixels geht ein:

- Materialeigenschaften des Objekts
- Augenpunkt des Betrachters
- Normalenvektor des Objekts
- Positionierung der Lichtquellen
- Art der Lichtquellen



Lichtquellen

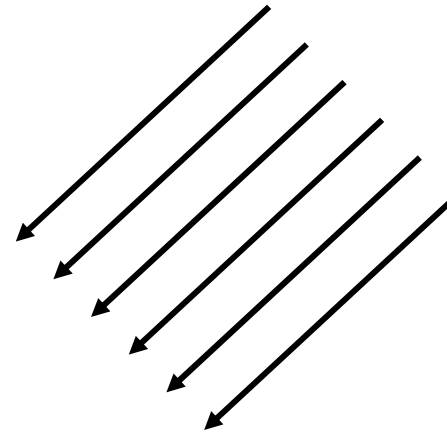
- Umgebungslicht ambient light
- gerichtetes Licht directed light
- Punktlicht point light
- Strahler spot light

Umgebungslicht

- keine Position
- keine Richtung
- Intensität I_a

Gerichtetes Licht

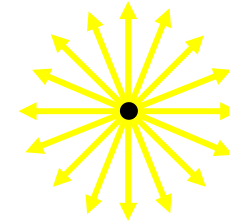
- keine Position
- Lichtrichtung L_g
- Intensität I_g



z.B. Sonnenlicht

Punktlicht

- Position P
- keine bevorzugte Richtung
- Anfangs-Intensität I_0
- Intensität nimmt mit Entfernung ab



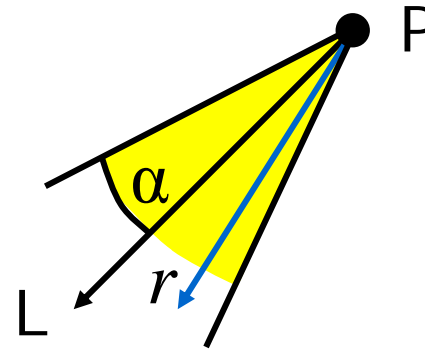
$$I(r) = \frac{I_0}{C_1 + C_2 \cdot r}$$

$r \geq 0$ Abstand zur Lichtquelle

$C_2 \geq 0$ Abschwächungskoeffizient

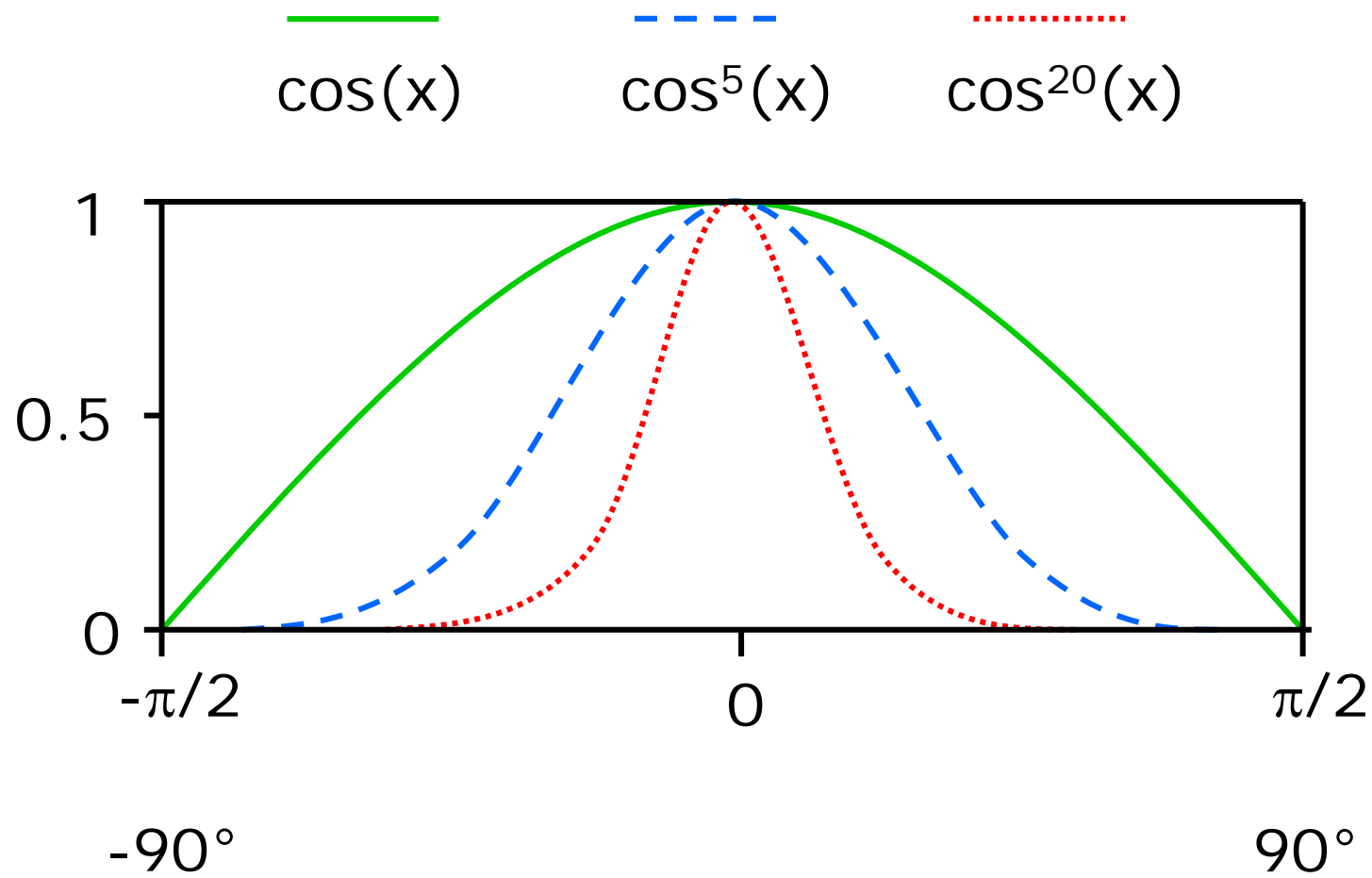
$C_1 \geq 1$ (*verhindert zu kleinen Nenner*)

Strahler



- Position P
 - Lichtrichtung L
 - Intensität I_0
 - Abschwächungskoeffizienten C_1, C_2
 - [Abstrahlwinkel α]
 - Konzentrationsexponent c
- ⇒ Intensität bei Richtung $r = \cos(r, L)^c$

Intensität im Lichtkegel



Gesamtbeleuchtung pro Pixel

- ambientes Licht
- diffus reflektiertes gerichtetes Licht
- spekulär reflektiertes gerichtetes Licht

$$\overline{C} = \overline{C}_a + \sum_{i=1}^n \overline{C}_{d_i} + \sum_{i=1}^n \overline{C}_{s_i}$$

n Zahl der Lichtquellen

\overline{C}_a ambienter Anteil

\overline{C}_{d_i} diffuser Anteil von Lichtquelle i

\overline{C}_{s_i} spekulärer Anteil von Lichtquelle i

pro Farbe einzeln berechnen !

Oberflächeneigenschaften

k_a ambienter Reflexionskoeffizient

k_d diffuser Reflexionskoeffizient

k_s spekulärer Reflexionskoeffizient

$\overline{O_d}$ diffuse Objektfarbe

$\overline{O_s}$ spekulare Objektfarbe

O_e spekulärer Exponent

ambiante Reflexion

Grundhelligkeit eines Objekts

$$\overline{C}_a = k_a \cdot I_a \cdot \overline{O}_d$$

k_a ambienter Reflexionskoeffizient

I_a Intensität des ambienten Lichts

\overline{O}_d diffuse Objektfarbe

diffuse Reflexion

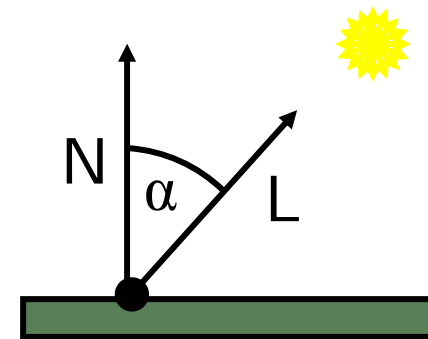
vom Objekt diffus reflektiertes Licht,
d.h. überall gleichmäßig sichtbar

$$\overline{C}_d = k_d \cdot I_e \cdot \overline{O}_d \cdot \cos(L, N)$$

k_d diffuser Reflexionskoeffizient

I_e Intensität des
einfallenden Lichts

\overline{O}_d diffuse Objektfarbe



spekulare Reflexion

vom Objekt gespiegeltes Licht,
nur in bestimmter Richtung sichtbar

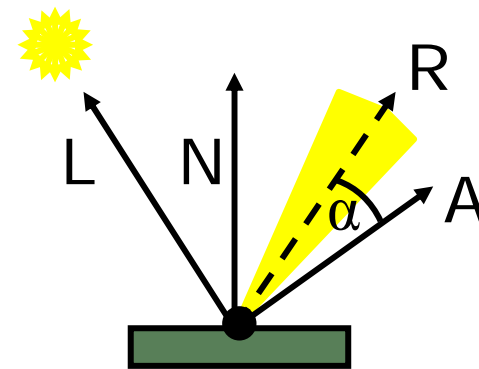
$$\overline{C}_s = k_s \cdot I_e \cdot \overline{O}_s \cdot \cos(R, A)^{O_e}$$

k_s spekulärer Reflexionskoeffizient

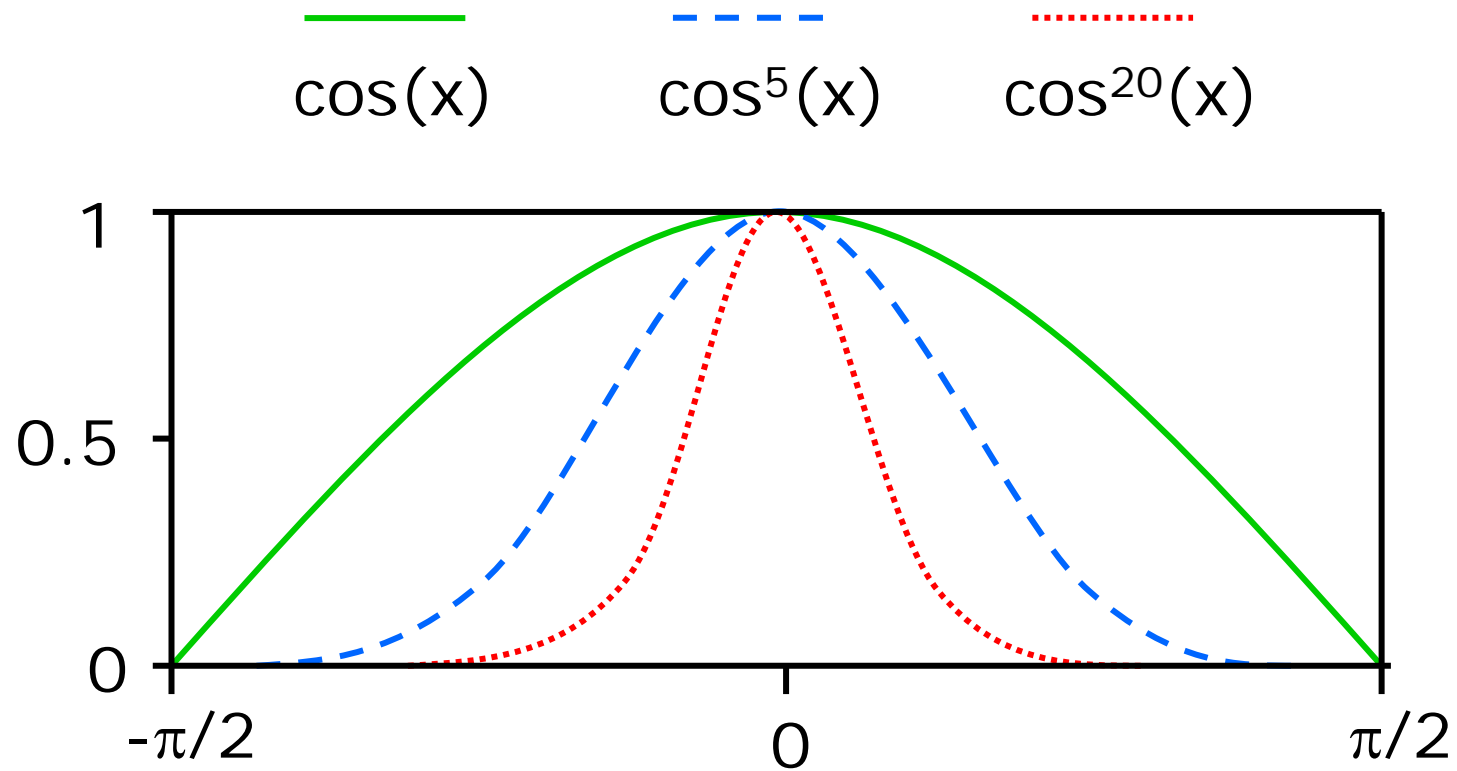
I_e Intensität des einfallenden Lichts

\overline{O}_s spekulare Objektfarbe

O_e spekulärer Exponent,
regelt Streukegel



Intensität im Streukegel



Materialeigenschaften

nicht mehr abstrahlen
als empfangen:

$$0 \leq k_a, k_d, k_s \leq 1$$

$$k_a + k_d + k_s \leq 1$$

kontrastarm

$$k_a \gg k_d, k_s$$

matt

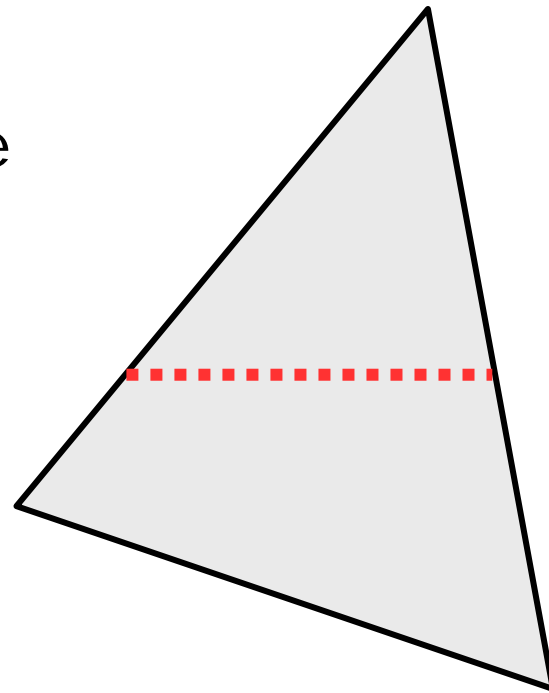
$$k_d \gg k_s$$

spiegelnd

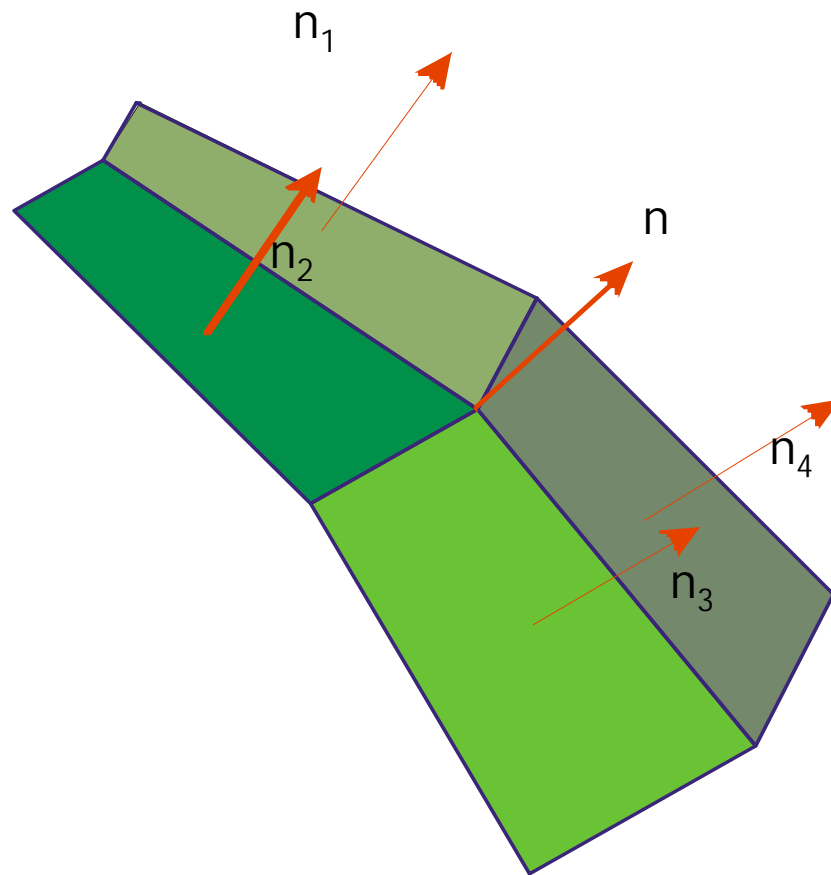
$$k_s > k_d$$

Schattierungsalgorithmen

- Flat Shading:
pro Dreieck eine Farbe
- Gouraud Shading:
Interpolation der Farbwerte
- Phong Shading:
Interpolation der Normalen



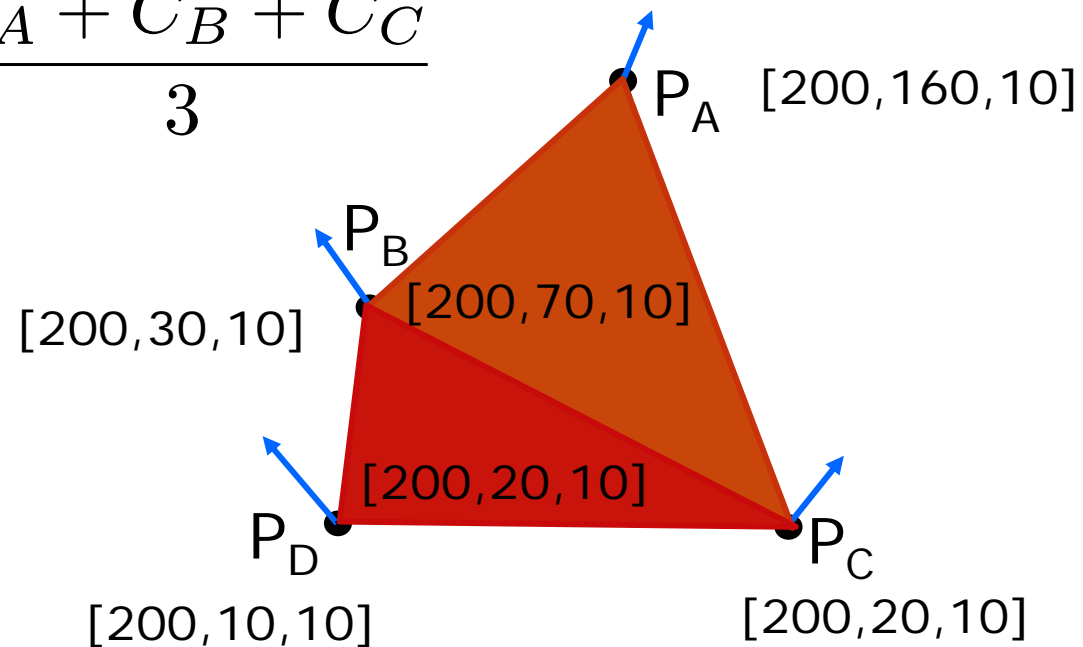
Normalen angleichen



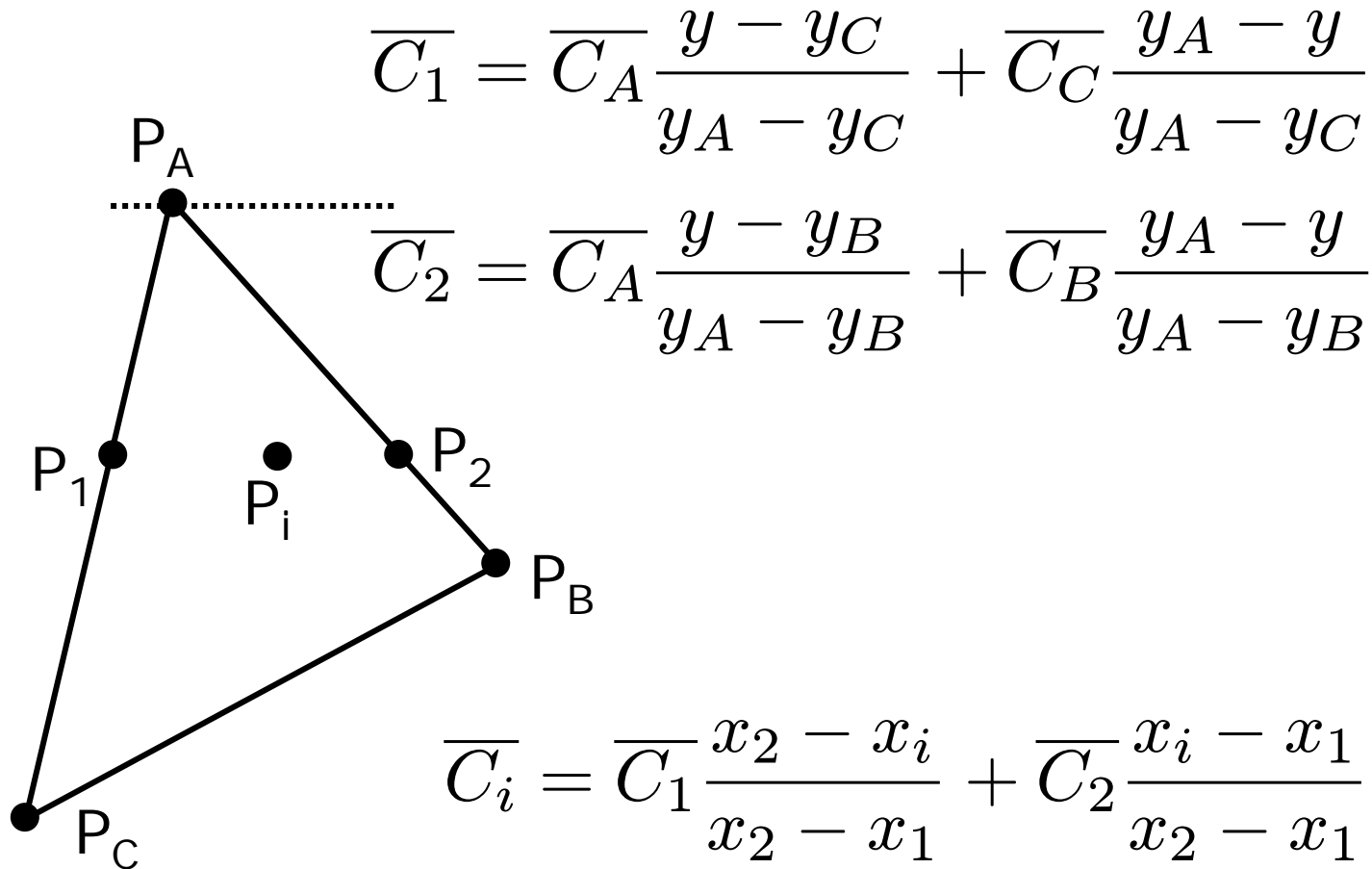
Flat Shading

- Eckpunkte im WC beleuchten
- Mittelwert für alle Pixel

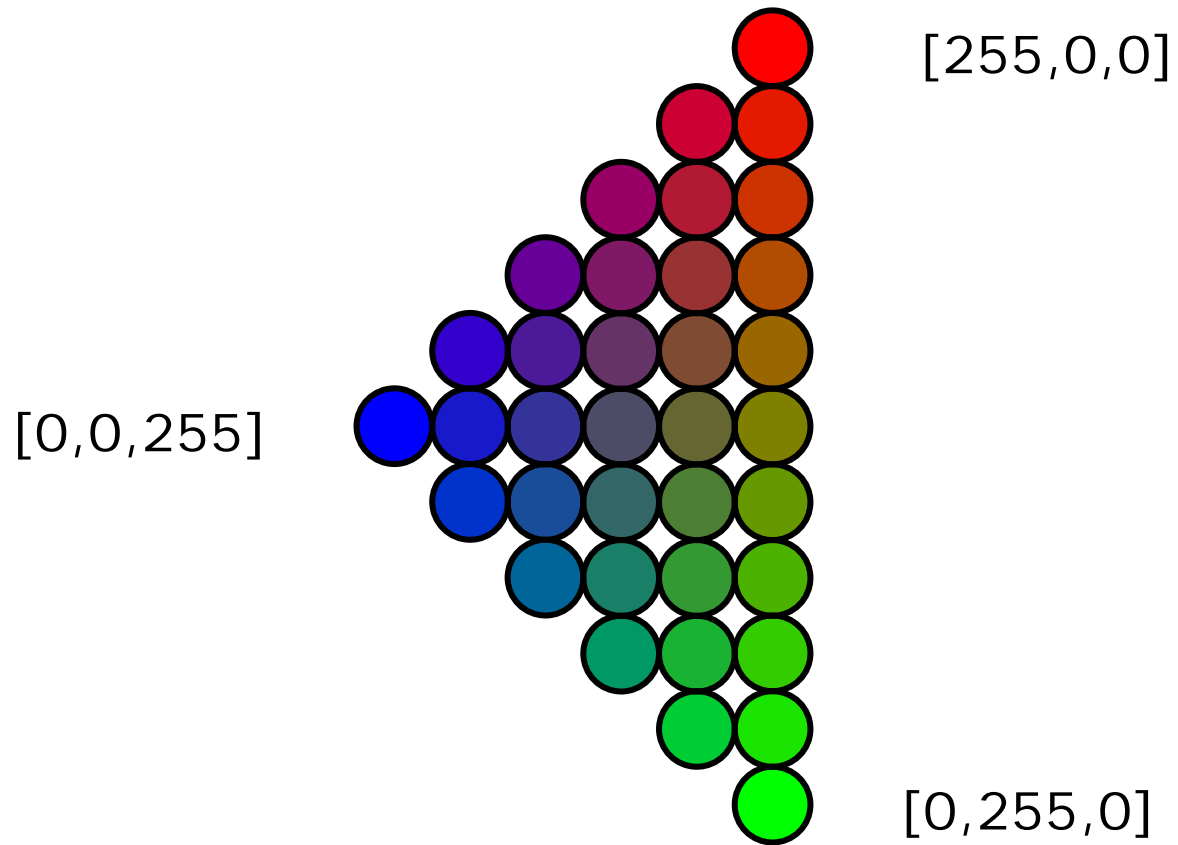
$$\overline{C}_i = \frac{\overline{C}_A + \overline{C}_B + \overline{C}_C}{3}$$



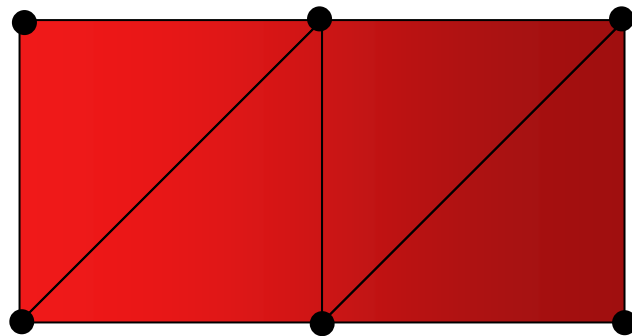
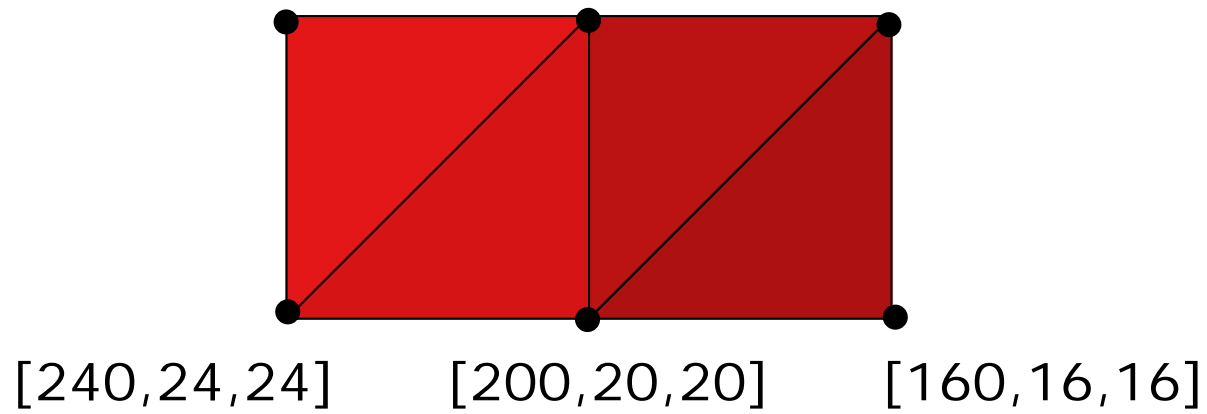
Gouraud Shading



Gouraud Shading



Flat versus Gouraud



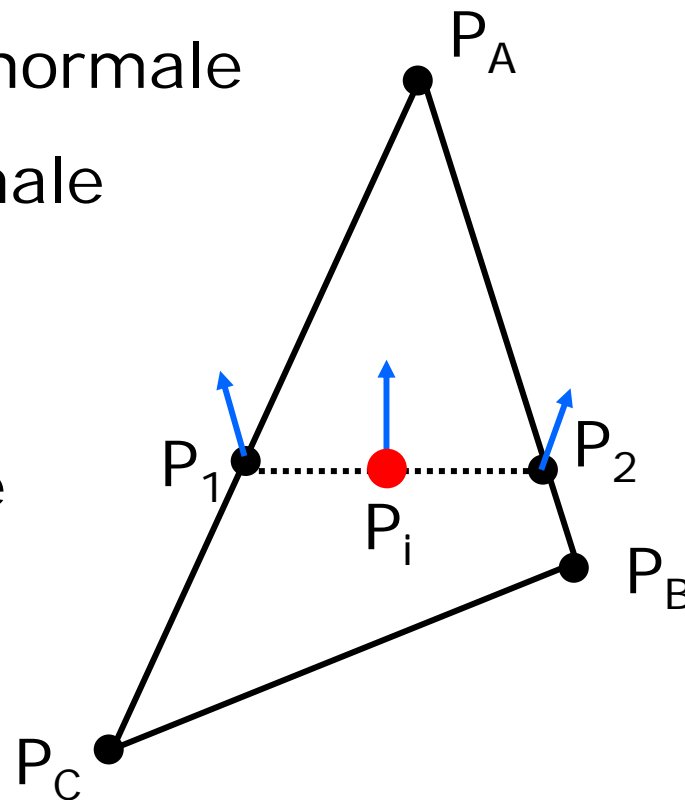
Phong Shading

pro Scanline:

- interpoliere Anfangsnormale
- interpoliere Endnormale

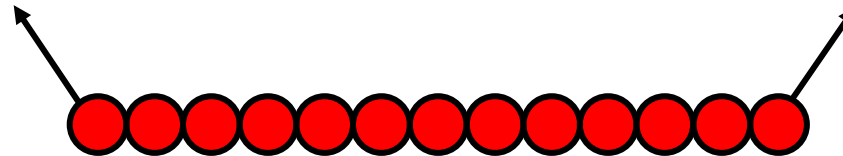
pro Pixel:

- interpoliere Normale
- berechne Farbwert

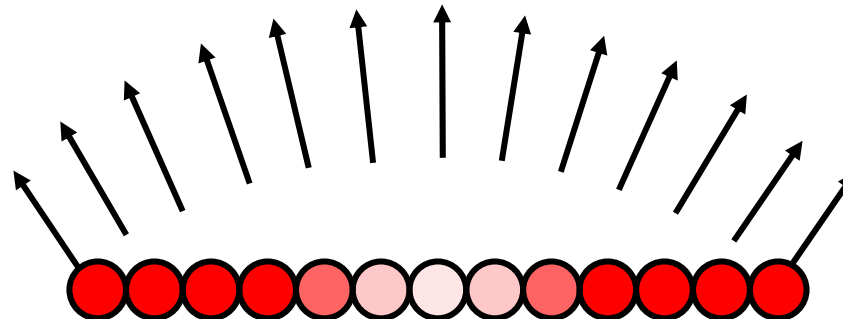


Gouraud versus Phong

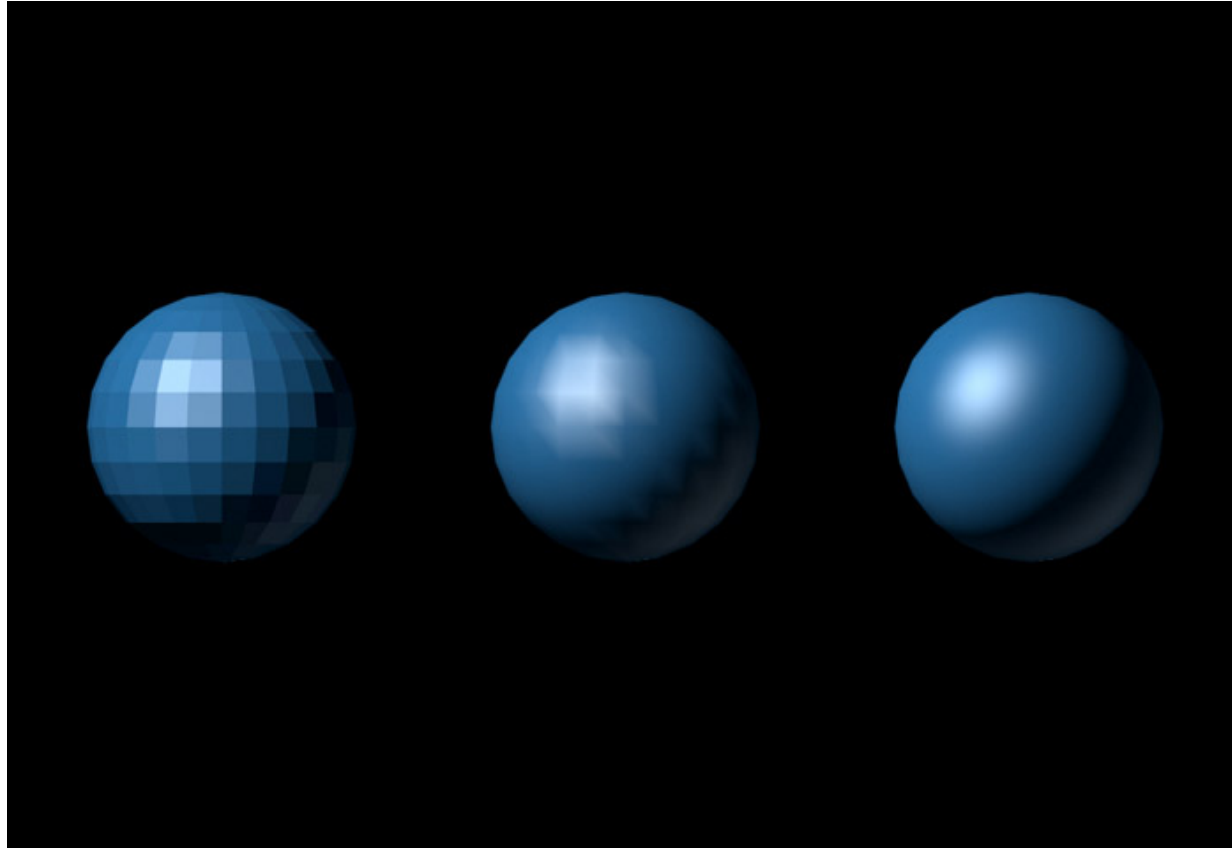
Farbwerte interpolieren



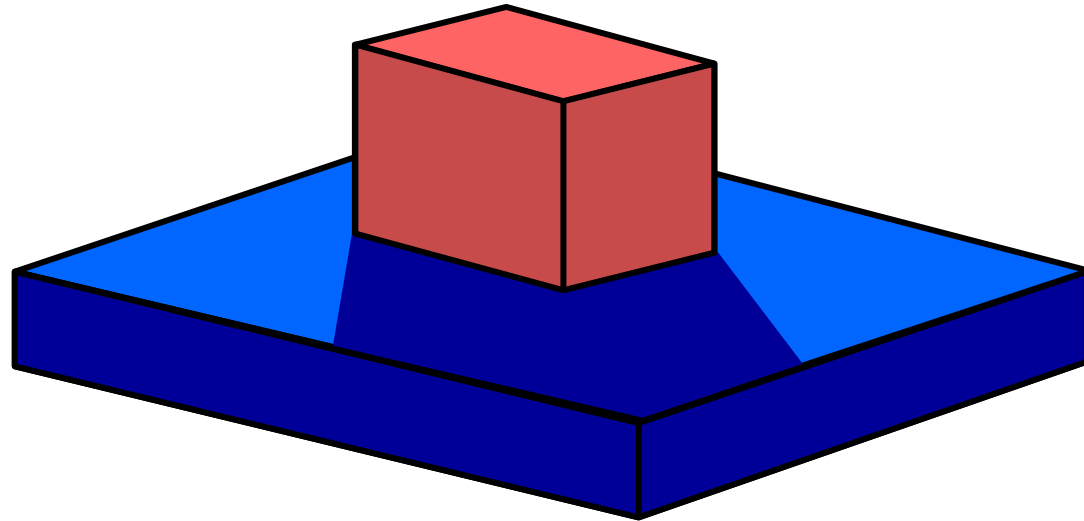
Normalen interpolieren
Farbwerte ausrechnen



Flat versus Gouraud versus Phong



Schatten



von der Lichtquelle nicht sichtbare Pixel

Berechnung von Schatten

geeignet: Hidden-Surface-Removal-Algorithmen

Phase 1: rendere Bild aus Position der Lichtquelle L
in einen Schattentiefenpuffer $s_tiefe[][]$

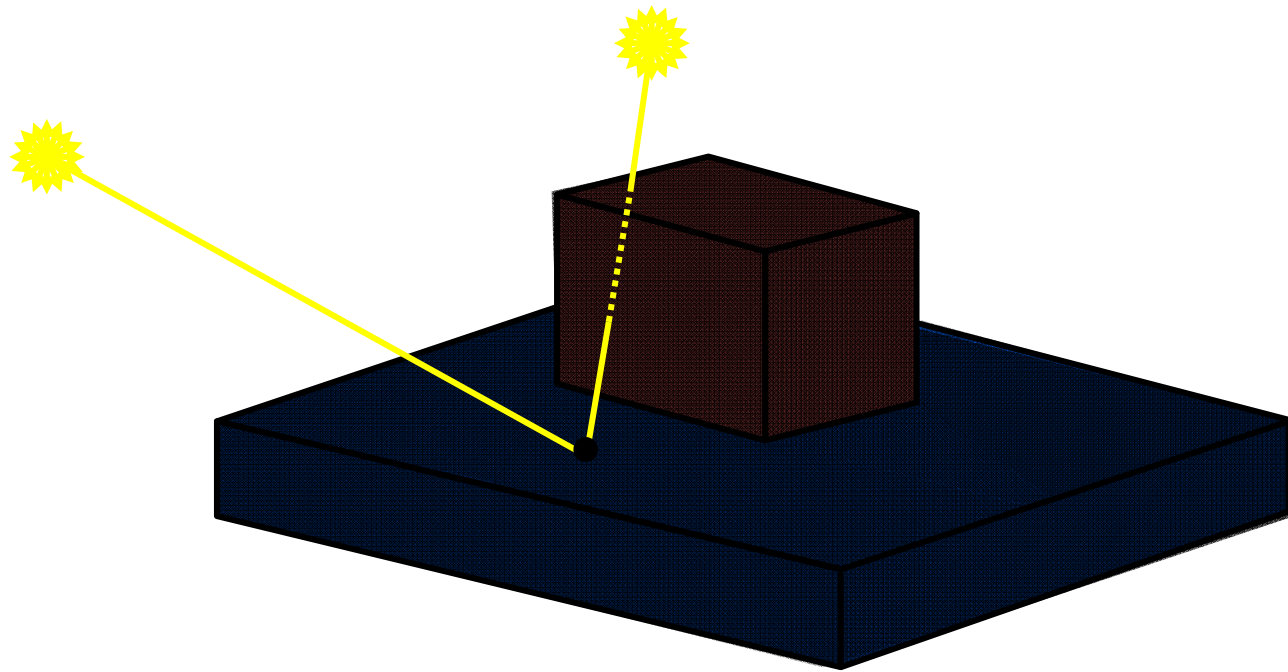
Phase 2: rendere Bild aus Position des Betrachters mit
modifiziertem Tiefenpuffer-Algorithmus:

falls Pixel P (x,y,z) sichtbar, transformiere P in den
Koordinatenraum von Phase 1 zu $P'(x',y',z')$.

Falls $z' < s_tiefe[x',y']$ dann P im Schatten von L
 \Rightarrow P ohne L beleuchten

Falls $z' \geq s_tiefe[x',y']$ dann P nicht im Schatten von L
 \Rightarrow P mit L beleuchten

Berechnung von Schatten



Kernschatten & Halbschatten

