

# Aufgabe 1: Das Phong-Beleuchtungsmodell

## Teilaufgabe 1a

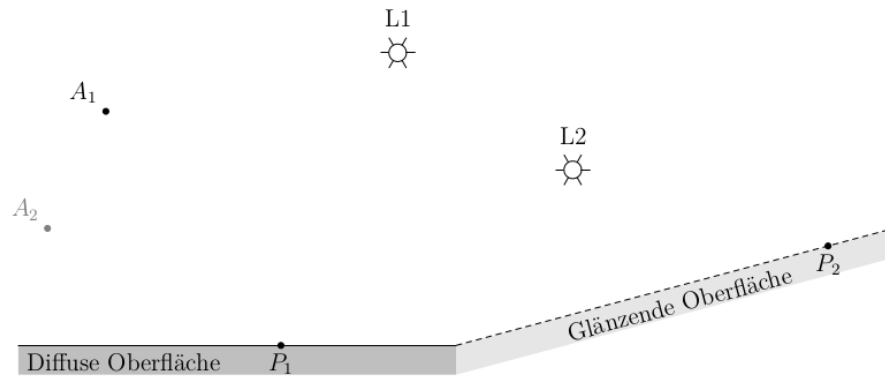


Abbildung 1: Skizze zu Aufgabe 1

## Teilaufgabe 1b

TODO

## Teilaufgabe 1c

TODO

## Teilaufgabe 1d

TODO

## Teilaufgabe 1e

TODO

## Teilaufgabe 1f

TODO

## Aufgabe 2: Raytracing

### Teilaufgabe 2a

- Anstelle einen Punkt für einen Pixel abzutasten, tastet man  $k^2$  mal in äquidistanten Intervallen ab.
- Aliasing wird dadurch verringert.

### Teilaufgabe 2b

- Maximale Rekursionstiefe erreicht
- Rekursion bis der Beitrag zur Farbe vernachlässigbar wird

### Teilaufgabe 2c

*Was ist der Unterschied zwischen Distributed Raytracing und Whitted-Style Raytracing?* TODO

*Welchen Lichttransport kann man durch Distributed Raytracing berechnen, den Whitted-Style Raytracing nicht erfassen kann?* TODO

### Teilaufgabe 2d

*Nennen Sie kurz und stichpunktartig die zwei Schritte, die zur Berechnung von Vertex-Normalen bei einem Dreiecksnetz notwendig sind! Gehen Sie dabei davon aus, dass nur die Vertex-Positionen und die Topologie des Netzes gegeben sind!* TODO

## Aufgabe 3: Farben und Farbwahrnehmung

### Teilaufgabe 3a

#### Teilaufgabe 3a (I)

Wie berechnet man die Sensorantwort  $a$  für ein Spektrum  $S(\lambda)$ ?

$$a(S(\lambda)) = \int_{\lambda} E(\lambda) \cdot S(\lambda) d\lambda$$

#### Teilaufgabe 3a (II)

Unter einem *Metamerismus* versteht man das Phänomen, das unterschiedliche Spektren den selben Farbeindruck vermitteln können. Es muss also

$$a_1 = a_2$$

gelten, damit  $S_1(\lambda)$  und  $S_2(\lambda)$  bzgl. der gegebenen Kamera Metamere sind.

### Teilaufgabe 3b

1. Das HSV-Farbmodell trennt Farbton von Helligkeit.  
⇒ Richtig (*Hue* (Farbton), *Saturation* (Sättigung), *Value* (Hellwert)).
2. Der Farbeindruck einer additiv gemischten Farbe hängt nicht vom Farbeindruck der Ausgangsfarben ab.  
⇒ TODO
3. Farbige Flächen werden unabhängig von ihrer Umgebung vom menschlichen Auge immer gleich wahrgenommen.  
⇒ Falsch. (TODO: Welche Folie?)
4. Der Machsche Bandeffekt ist vor allem bei Phong-Shading ein Problem.  
⇒ TODO

## Aufgabe 4: Bézier-Kurven

### Teilaufgabe 4a

Gegeben sei die Bézier-Kurve  $\mathbf{b}(u) = \sum_{i=0}^3 \mathbf{b}_i B_i^3(u)$  mit den Kontrollpunkten  $\mathbf{b}_i$ , wobei  $u \in [0, 1]$  und  $B_i^3$  das  $i$ -te Bernstein-Polynom vom Grad 3 ist.

#### Teilaufgabe 4a (I)

Werten Sie die Bézier-Kurve zeichnerisch mit dem de-Casteljau-Algorithmus an der Stelle  $u = 1/3$  aus! Markieren Sie den Punkt  $\mathbf{b}(1/3)$ !

TODO

#### Teilaufgabe 4a (II)

TODO

### Teilaufgabe 4b

TODO

## Aufgabe 5: Transformationen

TODO

## Aufgabe 6: Texturierung

### Teilaufgabe 6a

TODO

### Teilaufgabe 6b

TODO

### **Teilaufgabe 6c**

TODO

### **Teilaufgabe 6d**

TODO

## **Aufgabe 7: Cube-Maps und Environment-Mapping**

### **Teilaufgabe 7a**

TODO

### **Teilaufgabe 7b**

TODO

## **Aufgabe 8: Hierarchische Datenstrukturen**

### **Teilaufgabe 8a**

TODO

### **Teilaufgabe 8b**

TODO

### **Teilaufgabe 8c**

TODO

### **Teilaufgabe 8d**

TODO

## Aufgabe 9: Rasterisierung und OpenGL

TODO

## Aufgabe 10: Tiefenpuffer und Transparenz

### Teilaufgabe 10a

TODO

### Teilaufgabe 10b

TODO

### Teilaufgabe 10c

TODO

## Aufgabe 11: Phong-Shading und Phong-Beleuchtungsmodell

---

```
1  uniform mat4 matN; // Normalentransformation (Objekt -> Kamera)
2  uniform mat4 matM; // Modelltransformation
3  uniform mat4 matV; // Kameratransformation
4  uniform mat4 matP; // Projektionstransformation
5  uniform mat4 matMV; // Model-View-Matrix
6  uniform mat4 matMVP; // Model-View-Projection-Matrix
7
8  in vec3 P; // Eingabe-Vertex in Objektkoordinaten
9  in vec3 n; // Eingabenormale in Objektkoordinaten
10
11 out vec3 P_k; // Vertex-Position in Kamerakoordinaten
12 out vec3 n_k; // Vertex-Normale in Kamerakoordinaten
13
14 void main() {
15     // P_k = TODO;
16     // n_k = TODO;
17     // gl_Position = TODO;
18 }
```

---