Aufgabe 1: Das Phong-Beleuchtungsmodell

Teilaufgabe 1a

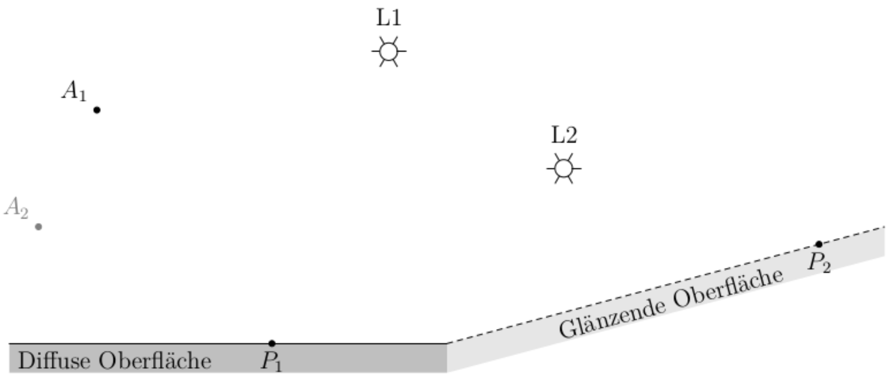

Abbildung 1: Skizze zu Aufgabe 1

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 $S(\lambda)$?

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

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.

$\Rightarrow$ Richtig (Hue (Farbton), Saturation (Sättigung), Value (Hellwert)).

2. Der Farbeindruck einer additiv gemischten Farbe hängt nicht vom Farbeindruck der Ausgangsfarben ab.

$\Rightarrow$ TODO

3. Farbige Flächen werden unabhängig von ihrer Umgebung vom menschlichen Auge immer gleich wahrgenommen.

$\Rightarrow$ Falsch. (TODO: Welche Folie?)

4. Der Machsche Bandeffekt ist vor allem bei Phong-Shading ein Problem.

$\Rightarrow$ 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* $\mathbf{b}({}^{1}\!/_{3})$*!*

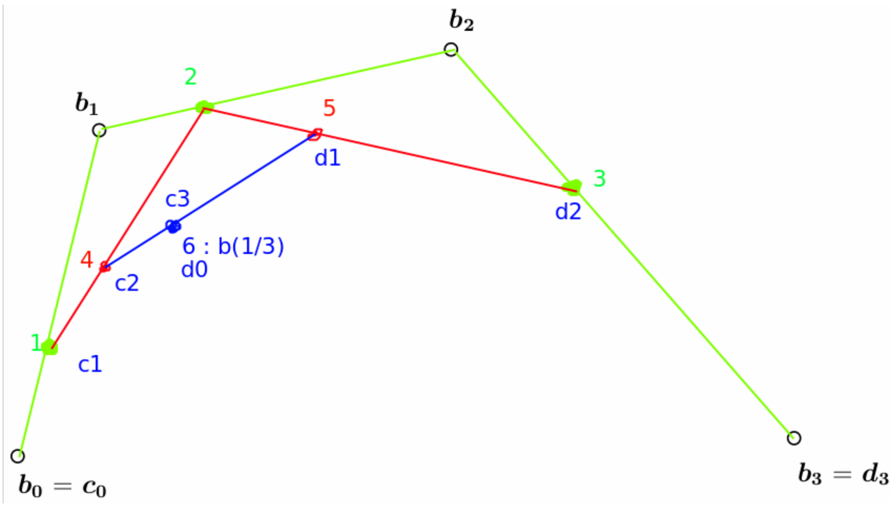

Abbildung 2: Skizze zu Aufgabe 4a und 4b

Teilaufgabe 4a (II)

vgl Abbildung 2 (da bin ich mir aber unsicher, ob das stimmt).

Teilaufgabe 4b

Siehe Nachklausur 2015, Aufgabe 11b für eine detailierte Erklärung.

1. Nein, da die Kontrollpunkte auf den Ecken eines Rechtecks liegen, aber die Kurve nicht symmetrisch ist.
2. Nein, da die Kurve nicht in der konvexen Hülle der Kontrollpunkte liegt.
3. Ja
4. Nein, da die Kurve nicht tangential an $b_0 b_1$ ist.

Aufgabe 5: Transformationen

Der Basiswechsel ist eine Verschiebung um $\begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$ und dann eine Rotation um 180° um die x-Achse. Daher:

$$M = \overbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 180° & -\sin 180° & 0 \\ 0 & \sin 180° & \cos 180° & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\text{Rotation}} \cdot \overbrace{\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\text{Translation}} \tag{1}$$

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

$$M = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

Aufgabe 6: Texturierung

Teilaufgabe 6a

$$\lambda_A = \frac{A_\Delta(P,B,C)}{A_\Delta(A,B,C)} = \frac{3}{6} \qquad \lambda_B = \frac{A_\Delta(P,A,C)}{A_\Delta(A,B,C)} = \frac{1}{6} \qquad \lambda_C = \frac{A_\Delta(P,A,B)}{A_\Delta(A,B,C)} = \frac{2}{6} \tag{4}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \lambda_A \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} + \lambda_B \cdot \begin{pmatrix} 12 \\ 6 \end{pmatrix} + \lambda_C \cdot \begin{pmatrix} 0 \\ 12 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} \tag{5}$$

Teilaufgabe 6b

Siehe Kapitel 4, Folie 56

TODO

Teilaufgabe 6c

Welchen Vorteil haben Summed-Area-Tables gegenüber Mipmaps bei der Texturfilterung? TODO

Teilaufgabe 6d

Das Interpolationsschema heißt *Bilineare Interpolation* und funktionert wie folgt:

$$t_{12} = (1-a) \cdot t_1 + a \cdot t_2 \tag{6}$$
$$t_{23} = (1-a) \cdot t_3 + a \cdot t_4 \tag{7}$$
$$t = (1-b) \cdot t_{12} + b \cdot t_{23} \tag{8}$$

Aufgabe 7: Cube-Maps und Environment-Mapping

Teilaufgabe 7a

Teilaufgabe 7a (I)

Wie wird die Cube-Map-Seite bestimmt, auf die zugegriffen wird? Welche ist es für $\mathbf{r}$*?* Es wird die betragsmäßig größte Komponente gewählt. Diese bestimmt ob es oben/unten oder links/rechts oder vorne/hinten wird. Das Vorzeichen bestimmt dann per Konvention die konkrete Fläche.

Im vorliegenden Fall ist $|r_z|$ am größten, also ist es Fläche 1.

Teilaufgabe 7a (II)

Berechnen Sie die Texturkoordinaten des Zugriffs auf der für $\mathbf{r}$ *ausgewählten Cube-Map-Seite!*

$$s = {}^1\!/_2 + \frac{r_x}{2 \cdot r_z} = \frac{1}{4} \tag{9}$$
$$t = {}^1\!/_2 + \frac{r_y}{2 \cdot r_z} = \frac{3}{4} \tag{10}$$

Teilaufgabe 7a (III)

Nennen Sie einen Vorteil von Cube-Maps gegenüber Sphere-Maps!

TODO

Teilaufgabe 7b

Teilaufgabe 7b (I)

Was wird in einer Environment-Map gespeichert?

Ein Bild der Umgebung in einer Textur.

Teilaufgabe 7b (II)

Nennen Sie ein Anwendungsbeispiel für Environment-Maps! TODO

Teilaufgabe 7b (III)

Welche grundlegende Annahme wird bei Environment-Mapping gemacht? Das Environment ist unendlich weit weg (also: nur die Richtung r wird verwendet, nicht jedoch der Ausgangspunkt P).

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

shader.vert

```
uniform mat4 matN; // Normalentransformation (Objekt -> Kamera)
uniform mat4 matM; // Modelltransformation
uniform mat4 matV; // Kameratransformation
uniform mat4 matP; // Projektionstransformation
uniform mat4 matMV; // Model-View-Matrix
uniform mat4 matMVP; // Model-View-Projection-Matrix

in vec3 P; // Eingabe-Vertex in Objektkoordinaten
in vec3 n; // Eingabenormale in Objektkoordinaten

out vec3 P_k; // Vertex-Position in Kamerakoordinaten
out vec3 n_k; // Vertex-Normale in Kamerakoordinaten

```

```
void main() {
    // P_k = TODO;
    // n_k = TODO;
    // gl_Position = TODO;
}
```