

## Aufgabe 1

### Teilaufgabe 1a

Der Bereich des vom Menschen sichtbaren Lichts liegt zwischen 380 und 700 nm. Dort hat Spektrum  $B(\lambda)$  die höhere Energie und wird daher heller wahrgenommen.

### Teilaufgabe 1b

$$\begin{aligned}x &= \frac{X}{X + Y + Z} = 0.5 \\y &= \frac{Y}{X + Y + Z} = 0.3 \\z &= 1 - x - y = 0.2\end{aligned}$$

### Teilaufgabe 1c

Die Kontrastsensitivität des Menschen ist für Luminanzunterschiede wesentlich stärker ausgeprägt wie für Farbunterschiede. Die Chrominanzinformation kann daher verlustbehafteter komprimiert werden, ohne dass dies wahrgenommen wird.

## Aufgabe 2

### Teilaufgabe 2a

TODO

### Teilaufgabe 2b

$$\mathbf{d}' = \mathbf{d}$$

### Teilaufgabe 2c

Bei einem Whitted-Style Raytracer ist ...	Notwendig	Optional
... Primärstrahlen erzeugen	<input checked="" type="checkbox"/>	<input type="checkbox"/>
... Strahlschnitte berechnen	<input checked="" type="checkbox"/>	<input type="checkbox"/>
... den Fresnel-Term auswerten	<input type="checkbox"/>	<input checked="" type="checkbox"/>
... Reflexionsstrahlen rekursiv weiterverfolgen	<input checked="" type="checkbox"/>	<input type="checkbox"/>
... Mip-Maps erstellen	<input type="checkbox"/>	<input checked="" type="checkbox"/>
... Beschleunigungsstrukturen verwenden	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## Aufgabe 3

### Teilaufgabe 3a

$$M_1 = V_1 \cdot T_A \cdot T_R$$

### Teilaufgabe 3b

$$\mathbf{A}' = T_S^{-1} \cdot T_A \cdot \mathbf{A}$$

### Teilaufgabe 3c

$$V_2 = T_A^{-1} \cdot T_F^{-1}$$
$$M_2 = T_F^{-1} \cdot T_R$$

### Teilaufgabe 3d

Aussage	Wahr	Falsch
1 Eine perspektivische Projektion ist immer eine affine Abbildung.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2 Zur Transformation von Punkten und dazugehörigen Normalen muss immer die gleiche Transformationsmatrix verwendet werden.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3 Homogene Transformationsmatrizen können Translationen darstellen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 Wenn eine allgemeine Transformation durch eine Matrix $M$ dargestellt wird, so führt $M^T$ die inverse Transformation durch.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5 Bei der homogenen Transformation eines Punktes mit einer Matrix $M$ beschreiben alle $\lambda M$ mit $\lambda \in \mathbb{R}, \lambda \neq 0$ dieselbe Abbildung.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

## Aufgabe 4

### Teilaufgabe 4a

$$I = k_a \cdot I_L + k_d \cdot I_L \cdot (\mathbf{n} \cdot \mathbf{l}) + k_s \cdot I_L \cdot (\mathbf{r}_1 \cdot \mathbf{v})^n$$

$k_a, k_d, k_s$ : Anteil der ambienten/diffusen/spekularen Komponente

$I_L$ : Intensität der Lichtquelle

$n$ : Phong-Exponent

### Teilaufgabe 4b

$$\mathbf{r}_1 = 2 \cdot \mathbf{n} \cdot (\mathbf{n} \cdot \mathbf{l}) - \mathbf{l}$$

### Teilaufgabe 4c

**Flat Shading** Berechnung der Flächennormale, Auswertung des Beleuchtungsmodells an einem Punkt für das ganze Dreieck

**Gouraud Shading** Auswertung des Beleuchtungsmodells an den Eckpunkten, dann Interpolation der Ergebnisse

**Phong Shading** Interpolation der Normale, dann Auswertung des Beleuchtungsmodells pro Punkt

**Phong-Beleuchtungsmodell** für alle Shading-Verfahren einsetzbar

## Aufgabe 5

### Teilaufgabe 5a

TODO

### Teilaufgabe 5b

f, g, a, d

### Teilaufgabe 5c

Aussage	BVH	kD-Baum	Gitter	keine
Die Datenstruktur kann nicht zur Beschleunigung von Nachbarschaftssuchen verwendet werden.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Der Aufbau der Datenstruktur ist <i>linear</i> in der Anzahl der Primitive.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Mehrfache Schnitttests mit demselben Primitive müssten explizit vermieden werden.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Die Datenstruktur eignet sich insbesondere für Szenen mit viel freiem Raum.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Teilaufgabe 5d

Die Surface Area Heuristic ...	Wahr	Falsch
... wird beim Traversieren einer Datenstruktur eingesetzt.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
... kann die Traversierung eines kD-Baumes beschleunigen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
... eignet sich besonders für den Aufbau von Oktalbäumen (Octrees).	<input type="checkbox"/>	<input checked="" type="checkbox"/>
... schätzt die Wahrscheinlichkeit, dass genau $N$ Primitive in einem Teilbaum liegen.	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## Aufgabe 6

### Teilaufgabe 6a

Aussage	Wahr	Falsch
1 Texturkoordinaten müssen immer im Bereich $[0, 1]$ liegen.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2 Beim Textur-Wrapping kann die Adressierung für jede Dimension separat gewählt werden.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 Anisotrope Texturfilterung sorgt im Allgemeinen dafür, dass Texturen im Vergleich zu isotroper Texturfilterung schärfer erscheinen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 RIP-Maps können als eine Verallgemeinerung von Mip-Maps angesehen werden.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 Der größte Vorteil des Sphere-Mapping gegenüber Latitude/Longitude-Maps besteht darin, dass die Abbildung keine Singularität(en) beinhaltet.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6 Durch Mip-Mapping wird der Speicheraufwand für Texturen um den Faktor $\sqrt{2}$ erhöht.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7 Bei der Verkleinerung (Minification) werden mehrere Bildschirm-Pixel auf den selben Texel abgebildet.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8 Mittels Summed Area Tables kann man über zwei Texturzugriffe die Summe über eine rechteckige Region in einer Textur berechnen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9 Für die bilineare Filterung muss der Abdruck (Footprint) eines Pixels im Texturraum bestimmt werden.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### Teilaufgabe 6b

**Vorteil:** Abbildung auf Oberflächenmodell einfach (keine Verzerrung etc.)

**Nachteil:** hoher Speicherbedarf

### Teilaufgabe 6c

**Normal-Mapping** Veränderung der Normalen eines Oberflächenpunkts für die Beleuchtungsberechnung

**Displacement-Mapping** Verschiebung eines Oberflächenpunkts

## Aufgabe 7

```
shader7.frag
1 uniform samplerCube tEnv;           // Environment-Map
2 uniform samplerCube tEnvFiltered;   // vorgefilterte Environment-Map
3 uniform sampler2D tRC;              // RGB-Textur, die r_c enthält
4 uniform sampler2D tKD;              // RGB-Textur, die k_d enthält
```

```

5
6 in vec3 v; // interpolierter View-Vektor (zur Kamera hin) in Weltkoordinaten
7 in vec3 n; // interpolierte Normale in Weltkoordinaten
8 in vec2 tc; // interpolierte Texturkoordinate
9
10 out vec3 color;
11
12 void main()
13 {
14     vec3 kd = vec3(texture(tKD, tc));
15     vec3 rc = vec3(texture(tRC, tc));
16
17     vec3 F = rc + (1 - rc) * pow(1 - dot(normalize(n), normalize(v)), 5);
18     color = mix(vec3(texture(tEnvFiltered, n)) * kd,
19                vec3(texture(tEnv, n)), F);
20 }

```

---

## Aufgabe 8

```

shader8.frag
1 uniform sampler2D tCarColor; // RGBA-Farbtextur der Autos
2 uniform sampler2D tTreeColor; // RGBA-Farbtextur der Bäume
3 uniform sampler2D tCarDepth; // Tiefentextur der Autos in NDC
4 uniform sampler2D tTreeDepth; // Tiefentextur der Bäume in NDC
5
6 in vec2 tc; // interpolierte Texturkoordinate
7 out vec4 color; // Ausgabefarbe des Fragments
8
9 void main()
10 {
11     // Tiefe und Farbe der Autos.
12     float dCar = texture(tCarDepth, tc).r;
13     vec4 cCar = texture(tCarColor, tc);
14
15     // Tiefe und Farbe der Bäume.
16     float dTree = texture(tTreeDepth, tc).r;
17     vec4 cTree = texture(tTreeColor, tc);
18
19     vec4 c1, c2;
20     if (cCar < cTree) {
21         c1 = cCar;
22         c2 = cTree;

```

```

23     } else {
24         c1 = cTree;
25         c2 = cCar;
26     }
27
28     color = vec4(c1.rgb, 1.0) * c1.a * (1 - c2.a) +
29             vec4(c2.rgb, 1.0) * c2.a;
30 }

```

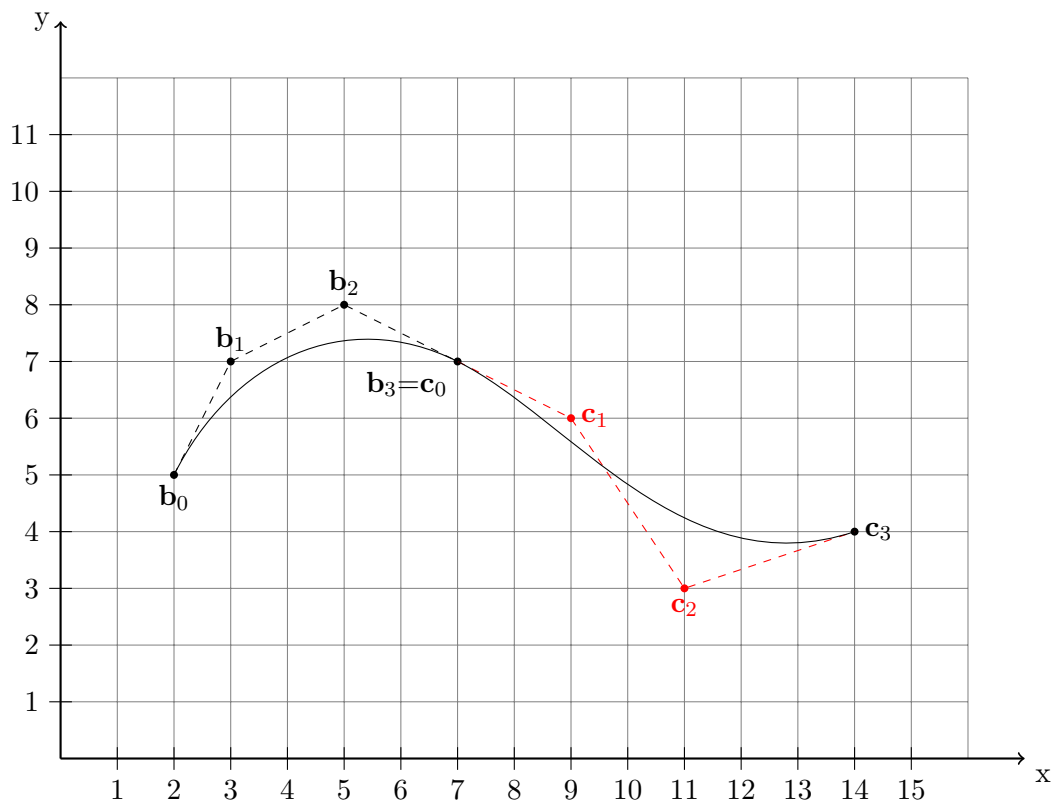
---

## Aufgabe 9

TODO

## Aufgabe 10

### Teilaufgabe 10a



### Teilaufgabe 10b

$$\mathbf{c}_0 = \mathbf{b}_3$$

$$\mathbf{c}_1 = \mathbf{c}_0 + (\mathbf{b}_3 - \mathbf{b}_2)$$

$$\mathbf{c}_2 = 2\mathbf{c}_1 + \mathbf{b}_1 - 2\mathbf{b}_2$$

### Teilaufgabe 10c

TODO