

# Aufgabe 1: Wahrnehmung, Farbe und Rasterbilder

## Teilaufgabe 1a

*Was versteht man unter Metamerie beim Farbsehen des Menschen?*

Metamerie ist das Phänomen, dass verschiedene Spektren den selben Farbeindruck erzeugen können.

## Teilaufgabe 1b

*Was versteht man unter Schwarzkörperstrahlung und Farbtemperatur?*

Ein Schwarzkörper ist eine idealisierte thermische Strahlungsquelle. Die idealisierung besteht darin, dass der Körper die komplette auftretende Strahlung vollständig absorbiert. Gleichzeitig sendet er Wärmestrahlung (Schwarzkörperstrahlung) aus, welche nur von seiner Temperatur abhängig ist.

Die Farbtemperatur ist ein Maß, um einen jeweiligen Farbeindruck einer Lichtquelle zu bestimmen.

## Teilaufgabe 1c

Siehe [martin-thoma.com/html5/graphic-filters](http://martin-thoma.com/html5/graphic-filters) zum ausprobieren.

- (A) Hervorheben von horizontalen Kanten.
- (B) Unschärfe / Weichzeichnen
- (C) Hervorheben aller Kanten
- (D) Hervorheben aller Kanten (Invertierter Laplace-Filter), entfernen vom Rest

## Teilaufgabe 1d

*Was versteht man unter einem normalisierten Filterkernel?*

Ein normalisierter Filterkernel hat als Summe der Element den Wert 1.

*Welche globale Eigenschaft eines Bildes ändert sich, wenn ein Filterkernel nicht normalisiert ist?*

Die Helligkeit des Bildes ändert sich nicht.

## Aufgabe 2: Prozedurale Modelle

### Teilaufgabe 2a

*Was sind Turbulenzfunktionen und wie können Sie aus Noise-Funktionen gebildet werden?*

Eine Turbulenzfunktion summiert  $k$  Oktaven mehrerer Noise-Funktionen  $n$  auf:

$$\text{turbulence}(x) = \sum_k \left(\frac{1}{2}\right)^k \cdot n(2^k \cdot x)$$

Einsatzgebiete:

- Natürliche Oberflächen
- Feuer

### Teilaufgabe 2b: Kontext-Freie Lindenmayer-Systeme

- (1)  $F \rightarrow F[+F][-F] \rightarrow F[+F[+F][-F]][-F[+F][-F]]$
- (2)  $F \rightarrow F[+F] \rightarrow F[+F][+F[+F]]$
- (3)  $F \rightarrow F[f - F]fF \rightarrow F[f - F]fF[f - F[f - F]fF]fF[f - F]fF$

### Teilaufgabe 2c: Turtle-Grafiken

- Die Grafik links oben ist (1)
- Die Grafik in der Mitte, unten ist (2)
- Die Grafik rechts unten ist (3)

## Aufgabe 3: Supersampling und Baryzentrische Koordinaten

### Teilaufgabe 3a

*Was ist adaptives Supersampling?*

Beim adaptiven Supersampling wird durch zwei benachbarte Pixel jeweils ein Strahl geschossen. Ist die Differenz der Pixelwerte über einem Schwellwert, so schießt man weitere Strahlen zwischen den beiden Pixeln. Dies wiederholt man so lange, bis man unter dem Schwellwert ist.

*Was ist stochastisches Supersampling mit Stratifikation?*

Beim stochastischen Supersampling wird jeder Pixel in ein Gitter unterteilt und durch jeden Gitterpunkt wird mit einer gewissen Wahrscheinlichkeit ein Strahl geschossen.

Was sind die Unterschiede zwischen adaptivem Supersampling und stochastischem Supersampling mit Stratifikation?

Adaptives Supersampling schießt nur bei Bedarf weitere Strahlen. Allerdings kann man Fälle konstruieren, wo adaptives Supersampling immer fehlschlägt.

### Teilaufgabe 3b

$$\lambda_A = \frac{A_\Delta(P, B, C)}{A_\Delta(A, B, C)} = \frac{2}{7.5} = \frac{4}{15} \quad (1)$$

$$\lambda_B = \frac{A_\Delta(P, A, C)}{A_\Delta(A, B, C)} = \frac{2.5}{7.5} = \frac{5}{15} \quad (2)$$

$$\lambda_C = \frac{A_\Delta(P, A, B)}{A_\Delta(A, B, C)} = \frac{3}{7.5} = \frac{6}{15} \quad (3)$$

## Aufgabe 4: Texturen

### Teilaufgabe 4a

Wie wird aus einer Textur eine Mip-Map-Pyramide erstellt?

TODO

Wie hoch ist der zusätzliche Speicherbedarf?

Der zusätzliche Speicherbedarf  $z$  ist  $\approx 1/3$  der ursprünglichen Texturgröße.

Beweis:

Es gilt:

$$z = \sum_{i=1}^{\infty} \left(\frac{1}{4}\right)^i$$

Dies ist eine geometrische Reihe. Die Summe der ersten  $n$  Terme ist

$$\frac{1 - (1/4)^n}{1 - 1/4}$$

daher gilt:

$$\lim_{n \rightarrow \infty} = 1 - \frac{1}{1 - 1/4} = 1/3$$

## Teilaufgabe 4b

*Welche Probleme bei der Texturfilterung im Fall der Verkleinerung (Texture Minification) löst Mip-Mapping?*

Mip-Mapping verringert Aliasing-Effekte.

*Welche Probleme bei der Texturfilterung im Fall der Verkleinerung löst Mip-Mapping nicht?*

Verwaschenes aussehen bei länglichem Footprint, da Mip-Map isotrop (TODO: erklären, ausformulieren).

## Teilaufgabe 4c

*Wofür verwendet man Environment Mapping?*

TODO

*Was speichert eine Environment Map?*

TODO

*Welche vereinfachenden Annahmen werden bei der Anwendung getroffen?*

Der Betrachter ist sehr weit von der Umgebung entfernt, sodass die Position keine Rolle spielt und ausschließlich die Blickrichtung wichtig ist.

## Aufgabe 5: $\alpha$ -Clipping

### Teilaufgabe 5a

$$\text{WEC}_{x_{\min}}(P_0) = -3 - 0 = -3 \qquad \text{WEC}_{x_{\min}}(P_1) = 6 - 0 = 6 \qquad (4)$$

$$\text{WEC}_{x_{\max}}(P_0) = 10 + 3 = 13 \qquad \text{WEC}_{x_{\max}}(P_1) = 6 - 6 = 0 \qquad (5)$$

$$\text{WEC}_{y_{\min}}(P_0) = 4 - 0 = 4 \qquad \text{WEC}_{y_{\min}}(P_1) = -2 - 0 = -2 \qquad (6)$$

$$\text{WEC}_{y_{\max}}(P_0) = 6 - 4 = 2 \qquad \text{WEC}_{y_{\max}}(P_1) = 6 + 2 = 8 \qquad (7)$$

### Teilaufgabe 5b

*Wenn Sie alleine die WEC betrachten, welche Kanten des Viewports werden dann potenziell geschnitten? Begründen Sie Ihre Antwort mit Hilfe der Outcodes!*

Da für  $\text{WEC}_{x_{\min}}(P_0) < 0$  und  $\text{WEC}_{y_{\min}}(P_1) < 0$  ist der Outcode  $x_{\min}(P_0) = 1$  und Outcode  $y_{\min}(P_1) = 1$ . Daraus folgt, dass  $x_{\min}$  und  $y_{\min}$  auf Schnitt getestet werden müssen. (TODO: Stimmt das?)

### Teilaufgabe 5c

$$a_{max} = 1, \quad a_{min} = 0 \quad (8)$$

$$a_{min} = \max(a_{min}, a_s) = 1/3 \quad (9)$$

$$a_s = \frac{\text{WEC}_{y_{\min}}(P_1)}{\text{WEC}_{y_{\min}}(P_0) - \text{WEC}_{y_{\min}}(P_1)} = 4/6 = 2/3 \quad (10)$$

$$a_{max} = \min(a_{max}, a_s) = 2/3 \quad (11)$$

$$(12)$$

Das neue Liniensegment ist somit  $(P_0 + 1/3 \cdot (P_1 - P_0), P_1 - 2/3 \cdot (P_1 - P_0))$ .

## Aufgabe 6

### Teilaufgabe 6a

*Was ist der Unterschied zwischen Gouraud- und Phong-Shading bei der Schattierung eines Dreiecks mit einem Rasterisierungsverfahren?*

TODO

### Teilaufgabe 6b

*Was versteht man unter einem Accumulation-Buffer, wie Sie ihn von OpenGL kennen?*

Der Accumulation-Buffer ist ein Zwischenspeicher zur Kombination mehrerer Redinering-schritte.

*Nennen Sie zwei Beispiele für Effekte, die sich mit einem Accumulation-Buffer erreichen lassen!*

- Bewegungsunschärfe
- TODO

## Teilaufgabe 6c

Aussage	Wahr	Falsch	Begründung
T-Vertices können bei Phong-Shading Artefakte verursachen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TODO: Wirklich?
Z-Fighting kann durch die Repräsentation der Tiefenwerte mit beschränkter Genauigkeit im Tiefenpuffer entstehen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TODO: Wirklich?
Je feiner eine Oberfläche tesselliert wird, umso geringer werden die Unterschiede zwischen Gouraud- und Phong-Shading.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TODO: Wirklich?
Bei der Rasterisierung ist eine perspektivisch korrekte Abbildung der Textur aufwendiger als eine affine, da pro Pixel eine zusätzliche Division benötigt wird.	<input type="checkbox"/>	<input type="checkbox"/>	TODO
Der Scissor-Test dient dazu, durchsichtige Teile einer Oberfläche gemäß einer Textur wegzuschneiden.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dient dazu Teile außerhalb des Rechtecks wegzuschneiden (vgl. OpenGL Teil 2, Folie 103)
Screendoor-Transparency stellt transparente Objekte mittels Blending dar.	<input type="checkbox"/>	<input type="checkbox"/>	TODO

## Aufgabe 7: OpenGL

### Teilaufgabe 7a

#### Teilaufgabe 7a (1)

$$c' = (1 - 0.5) \cdot (0, 0, 1, 0.5) + 0.5 \cdot (0, 0, 0, 0) = (0, 0, 0.5, 0.25) \quad (13)$$

$$c = (1 - 0.5) \cdot (1, 0, 0, 0.5) + 0.5 \cdot (0, 0, 0.5, 0.25) = (0.5, 0, 0.25, 3/8) \quad (14)$$

#### Teilaufgabe 7a (2)

$$c' = 1 \cdot (0, 0, 1, 0.5) + 0.5 \cdot (0, 0, 0, 0) = (0, 0, 1, 0.5) \quad (15)$$

$$c = 1 \cdot (1, 0, 0, 0.5) + 0.5 \cdot (0, 0, 1, 0.25) = (1, 0, 0.5, 3/4) \quad (16)$$

### Teilaufgabe 7b

1. Model-View-Transformation anwenden
2. Projektionstransformation anwenden
3. Tiefentest
4. Texturierung
5. Beleuchtungsberechnung

### Teilaufgabe 7c

Zwischen 2 und 3, da die Rasterisierung nur für Dreiecke erfolgen soll, die im Viewfrustum liegen.

## Aufgabe 8

### Teilaufgabe 8a

Operation	Vertex-Shader	Fragment-Shader	Weder noch
Model-View-Transformation anwenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> TO-DO
Tiefentest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> TO-DO
Texturierung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> TO-DO
Projektionstransformation anwenden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> TO-DO
Beleuchtungsberechnung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> TO-DO
Clipping	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> TO-DO

### Teilaufgabe 8b

- **attribute:** Attribut eines Vertex; nur für Vertex-Shader; z.B. Farbe oder Normale
- **uniform:** Bei jedem Shader-Aufruf gleich (also insbesondere für jeden Vertex gleich); read-only; z.B. Transformationsmatrix
- **varying:** weitergegebene/interpolierte Werte (schreiben in einem Shader, lesen im darauffolgenden Shader)

## Teilaufgabe 8c

```
----- shader.vert -----
1 void main() {
2     gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
3
4     // TODO;
5 }
-----
----- shader.frag -----
1 void main() {
2     gl_FragColor = // TODO;
3 }
```

## Aufgabe 9: Bézier-Kurven

### Teilaufgabe 9a

$$b_0 \cdot (1 - 2u + u^2) + b_1(2u - 2u^2) + b_2 \cdot u^2 \quad (17)$$

$$= b_0 - 2b_0u + b_0u^2 + 2b_1u - 2b_1u^2 + b_2u^2 \quad (18)$$

$$= \underbrace{(b_0 - 2b_1 - b_2)}_{=a_2} u^2 + \underbrace{(2b_1 - 2b_0)}_{=a_1} u + \underbrace{b_0}_{=a_0} \quad (19)$$

### Teilaufgabe 9b

Was versteht man unter affiner Invarianz der Bézier-Repräsentation?

Für jede Bézierkurve  $F(u)$  und jede affine Abbildung  $\varphi(x) = Ax + t$  gilt:

$$\varphi(F(u)) = \sum_{i=0}^n B_i^n(u) \varphi(\mathbf{b}_i)$$

Das heißt es genügt bei Transformationen die Kontrollpunkte  $\mathbf{b}_i$  zu transformieren.

### Teilaufgabe 9c

Für  $C^0$ -Stetigkeit muss  $c_0 = b_3 = (4, 1)$  gelten.

Für  $C_1$ -Stetigkeit muss  $b_2 - b_3 = c_0 - c_1 \Leftrightarrow (3, 1) - (4, 1) = (4, 1) = \underbrace{(5, 1)}_{=c_1}$  gelten.



Für  $C_2$ -Stetigkeit muss

$$b_2 + (b_2 - b_1) = c_1 + (c_1 - c_2) \quad (20)$$

$$\Leftrightarrow (3, 1) + ((3, 1) - (1, 2)) = (5, 1) + ((5, 1) - c_2) \quad (21)$$

$$\Leftrightarrow (5, 0) = (10, 2) - c_2 \quad (22)$$

$$\Leftrightarrow c_2 = (5, 2) \quad (23)$$

$$(24)$$

TODO: Wirklich? Vielleicht eher:

$$b_2 + (b_1 - b_0) = c_1 + (c_2 - c_3) \quad (25)$$

$$\Leftrightarrow (3, 1) + ((1, 2) - (0, 0)) = (5, 1) + (c_2 - (7, 0)) \quad (26)$$

$$\Leftrightarrow (4, 3) = (-2, 1) + c_2 \quad (27)$$

$$\Leftrightarrow c_2 = (6, 2) \quad (28)$$

$$(29)$$