Karlsruher Institut für Technologie

Institut für Visualisierung und Datenanalyse
Lehrstuhl für Computergrafik
Prof. Dr.-Ing. Carsten Dachsbacher

Klausur Computergrafik
SS 2023

08.08.2023

Name	
Matrikelnummer	

Beachten Sie:

- Die Klausur umfasst 24 Seiten (12 Blätter) mit 11 Aufgaben.
- Es sind folgende **Hilfsmittel zugelassen**: Lineal, Geodreieck.
- Sie haben **90 Minuten** Bearbeitungszeit.
- Schreiben Sie Ihre Matrikelnummer oben auf jedes bearbeitete Aufgabenblatt.
- Schreiben Sie Ihre Lösungen auf die Aufgabenblätter. Bei Bedarf können Sie weiteres Papier anfordern.
- Streichen Sie nicht zu bewertende Lösungen durch.
- Wir akzeptieren auch englische Antworten.

Aufgabe	1	2	3	4	5	6	7	8	9	10	11	**Gesamt**
Erreichte Punkte												
Erreichbare Punkte	16	15	11	20	11	18	28	6	24	12	19	**180**

Note	**Bonus**	

Aufgabe 1: Farbe und Perzeption (16 Punkte)

a) Für ein Videospiel soll ein Schneeeffekt dargestellt werden (Abbildung 1). Hierfür werden Pixel vor der Darstellung des Bildes mit verschieden hellen Grauwerten eingefärbt.

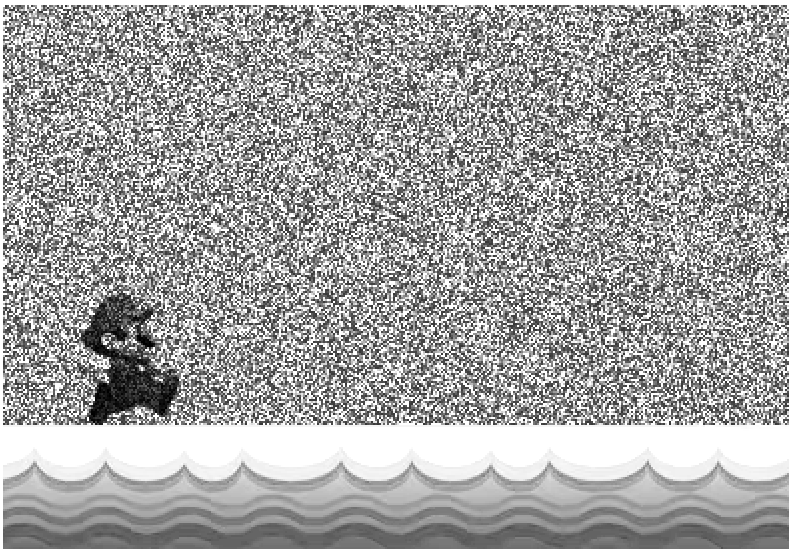


Abbildung 1: Schneeeffekt in einem Videospiel.

i) Erklären Sie *stichpunktartig*, weshalb die Gamma-Korrektur notwendig ist, um die Helligkeit des Schnees proportional zu den Grauwerten darzustellen! **(3 Punkte)**

ii) Gegeben seien ein Grauwert $0 \leq g \leq 1$ und ein Gammawert γ. Geben Sie an, wie damit der dem Monitor übergebene Farbwert $\hat{g}$ berechnet wird! Gehen Sie von einem Monitor mit idealen Display-Helligkeiten $I_{min} = 0$ und $I_{max} = 1$ aus! **(3 Punkte)**

$\hat{g} =$

iii) In einer Nachtszene des Videospiels wird ein rein schwarzer Hintergrund verwendet. Sie stellen fest, dass die Spielfigur deutlich heller erscheint. Beschreiben Sie in einem Satz, woran das liegt und benennen Sie dieses Phänomen! **(3 Punkte)**

b) Durch die Matrix $M = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$ wird mathematisch eine Bijektion zwischen dem XYZ-Farbraum und einem RGB-Farbraum hergestellt: $(X, Y, Z)^T = M \cdot (R, G, B)^T$. Die Umkehrabbildung erhält man durch die Inverse M^{-1}, welche negative Einträge enthält.

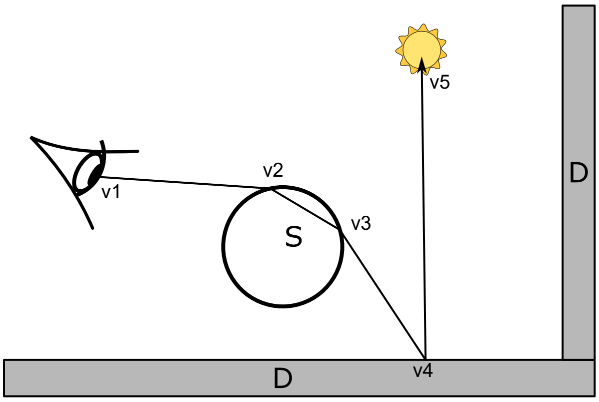
i) Was bedeuten die negativen Einträge für die Realisierung von Farbeindrücken mit RGB? Begründen Sie! **(4 Punkte)**

ii) Gegeben sei ein Vektor (R, G, B) bezüglich des RGB-Farbraums. Geben Sie an, wie die Chromatizitätswerte des xyY-Farbraums daraus berechnet werden! **(3 Punkte)**

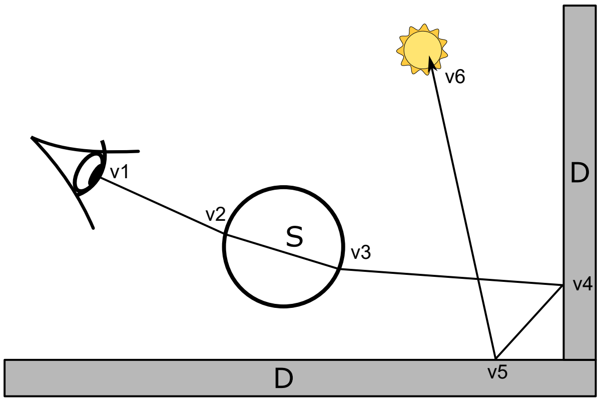
$x =$

$y =$

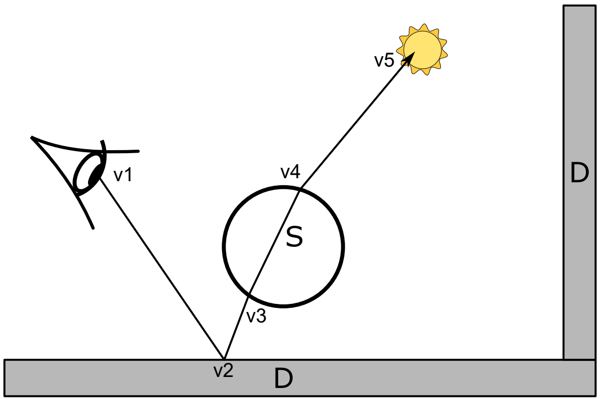
Aufgabe 2: Whitted-Style Raytracing (15 Punkte)



(a)



(b)



(c)

In den obigen Abbildungen sind drei Lichttransportpfade (a–c) eingezeichnet, die die Kamera und die Punktlichtquelle verbinden. Die Kugel S ist eine Glaskugel (nur perfekte Transmission). Die Wände D sind diffus.

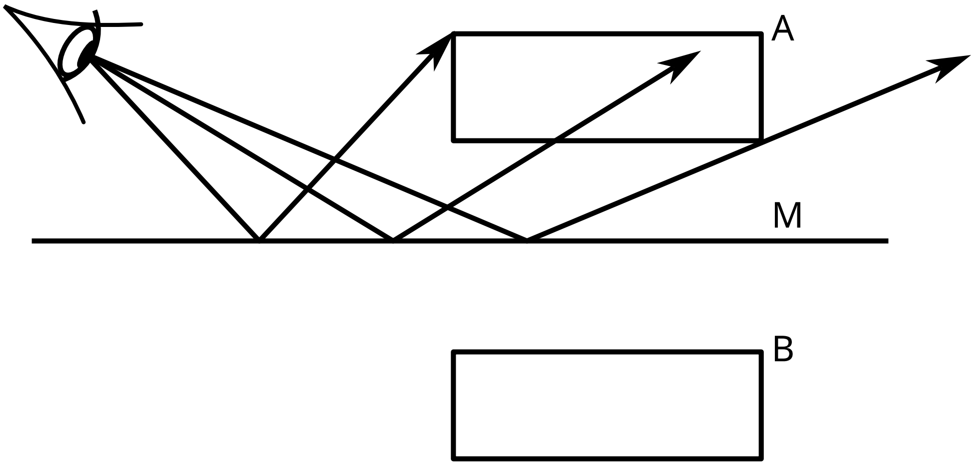
a) Geben Sie für jeden der drei Pfade (a–c) an, ob dieser mit Whitted-style Raytracing erzeugt werden kann und begründen Sie, falls nicht! Bei der Begründung können Sie die jeweiligen Bezeichnungen der Schnittpunkte vi verwenden. **(5 Punkte)**

(a)

(b)

(c)

Matrikelnummer: ____________________



In der obigen Skizze sind eine Wasserfläche M und zwei Quader A und B abgebildet. Die Spiegelung des Quaders A im Wasser soll deckungsgleich über dem Bild des Quaders B in der Kamera zu sehen sein. Dazu wurde Quader B unter A platziert. Die Skizze zeigt ausserdem exemplarisch drei Primärstrahlen, die an der Wasserfläche gespiegelt werden. Das Bild wird mittels Whitted-style Raytracing dargestellt, das Wasser hat einen Brechungsindex von 1.3.

b) Warum sind die beiden Quader im Bild nicht deckungsgleich? Begründen Sie! **(5 Punkte)**

c) In welche Richtung muss der Quader B bei gleichbleibender Größe bewegt werden, damit das gewünschte Resultat entsteht? Begründen Sie! **(5 Punkte)**

Aufgabe 3: Schattierung und Beleuchtungsmodell (11 Punkte)

a) Eine Oberfläche mit Normale $\mathbf{N}$ wird mit dem Phong Beleuchtungsmodell schattiert. Geben Sie an, wie die spekulare Komponente I_s berechnet wird! Bestimmen Sie in Abhängigkeit von $\mathbf{L}$ und $\mathbf{N}$ jene Betrachterrichtung $\mathbf{V}_{\mathrm{max}}$, die die spekulare Komponente maximiert! Begründen Sie Ihre Antwort! **(5 Punkte)**

$I_s =$

$\mathbf{V}_{\mathrm{max}} =$

b) Ein rasterisiertes Dreieck deckt 20 Pixel ab. Wie oft muss das Phong Beleuchungsmodell mit Gouraud und Phong Shading jeweils ausgewertet werden? **(2 Punkte)**

Gouraud Shading:

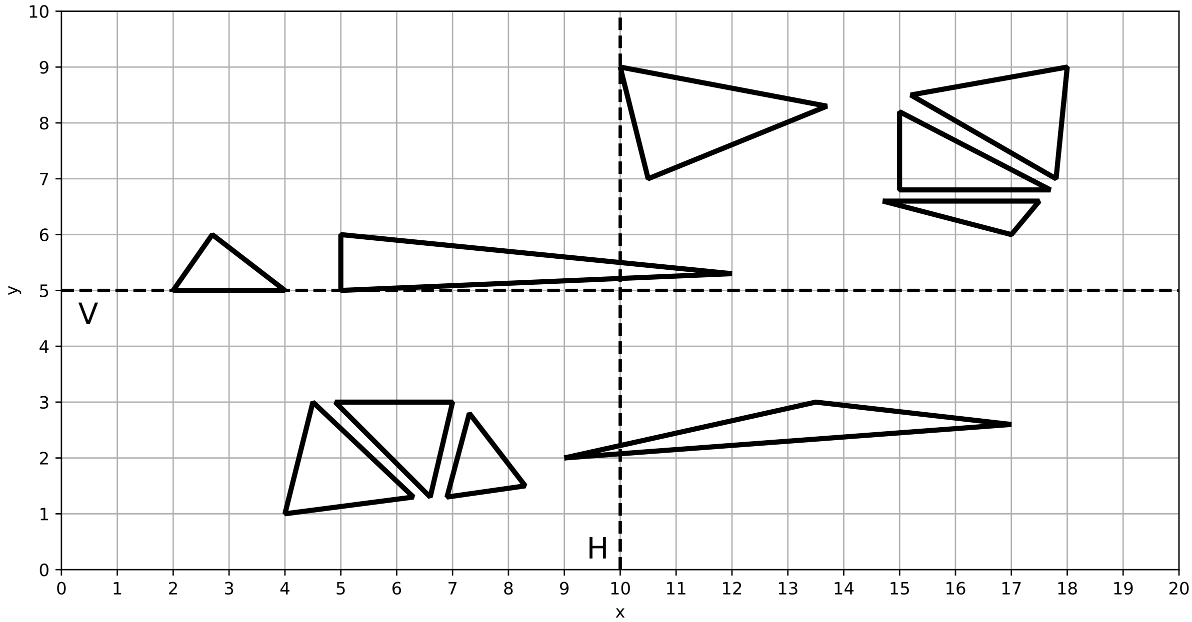
Phong Shading:

c) Wann sind weniger Artefakte beim Rendering mit Gouraud Shading zu erwarten: Bei Verwendung eines rein diffusen oder eines rein spekularen Beleuchtungsmodells? Begründen Sie Ihre Antwort! **(4 Punkte)**

Matrikelnummer: ____________

Aufgabe 4: Räumliche Datenstrukturen (20 Punkte)

a) Bei der Konstruktion einer binären Hüllkörperhierarchie mit achsenparallelen Boxen, von der Wurzel beginnend, soll die Menge der unten abgebildeten Dreiecke weiter aufgeteilt werden. Dazu werden die beiden möglichen Trennebenen H ($x = 10$) und V ($y = 5$) betrachtet. Mithilfe der Surface Area Heuristic (SAH) soll die Aufteilung ausgewählt werden, die für Raytracing besser geeignet ist.
Hinweis: Dreiecke werden anhand ihrer Mittelpunkte einem Halbraum zugeordnet.



i) Warum führt die Auswahl nach der SAH zu einer guten Aufteilung für Raytracing? **(3 Punkte)**

ii) Die SAH wird für das 2D-Beispiel angepasst, indem Oberfläche durch Umfang ersetzt wird. Der SAH-Wert für die Unterteilung nach V ist gegeben durch $\text{SAH}(V) = C_T + \frac{360}{48} \cdot C_i$. Berechnen Sie $\text{SAH}(H)$ und geben Sie an, ob demnach H oder V besser zur Aufteilung geeignet ist! **(8 Punkte)**
Hinweis: Der Umfang des Hüllkörpers für alle Dreiecke beträgt 48.

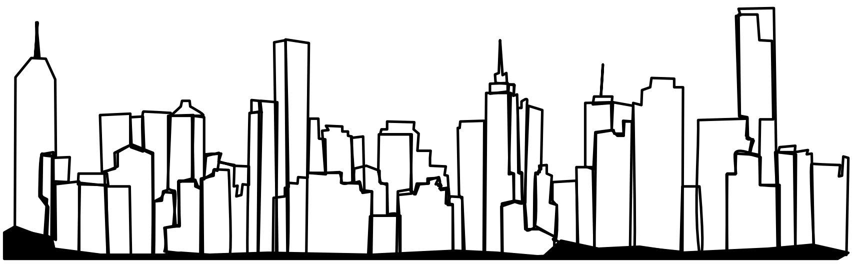
b) Nennen Sie die Eigenschaft, die eine räumliche Datenstruktur haben muss, damit Mailboxing sinnvoll eingesetzt werden kann und begründen Sie! Nennen Sie *je ein* Beispiel für eine Datenstruktur, bei der Mailboxing sinnvoll sein kann, und eine bei der dies nicht der Fall ist! **(4 Punkte)**

Eigenschaft:

Mailboxing sinnvoll:

Mailboxing nicht sinnvoll:

c) Eine Stadt (unten schematisch abgebildet) soll mit Raytracing und einer räumlichen Datenstruktur zur Beschleunigung dargestellt werden. Nachdem die Szene versehentlich mit einer globalen affinen Transformation verändert wurde, ist das Raytracing deutlich langsamer geworden.



Nennen Sie eine Transformation und eine Beschleunigungsstruktur, bei denen dieser Effekt auftreten kann und begründen Sie! **(5 Punkte)**

Matrikelnummer: ______________

Aufgabe 5: Transformationen (11 Punkte)

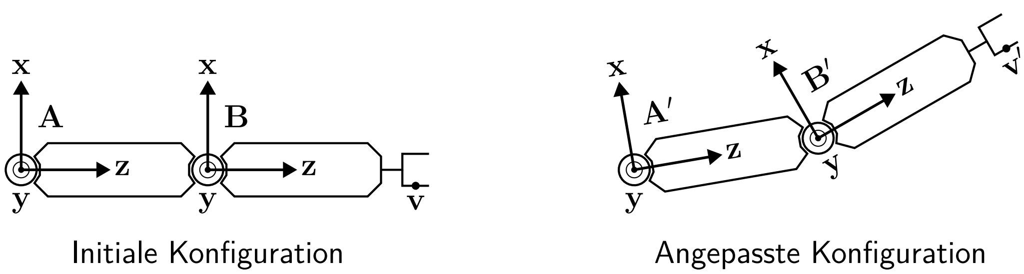


Abbildung 2: Skeletale Animation eines Roboterarms.

Mittels skeletaler Animationen können Modelle mit vielen Primitiven über Gelenke manipuliert werden. Zunächst werden die Gelenke anhand der initialen Konfiguration des Modells (Abb. 2, links) ausgerichtet. Die Gelenke bilden eine Hierarchie von Transformationen. **B** transformiert in das Koordinatensystem von **A** und **A** in Modellkoordinaten. Jeder Vertex wird einem Gelenk zugewiesen: Die Bewegung von Vertex **v** folgt dem Gelenk **B** bzw. **B′** (Abb. 2, rechts).

Im Folgenden werden Sie die notwendigen Transformationen zur Animation bestimmen.

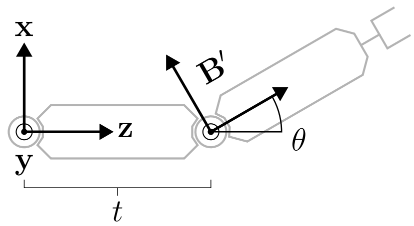
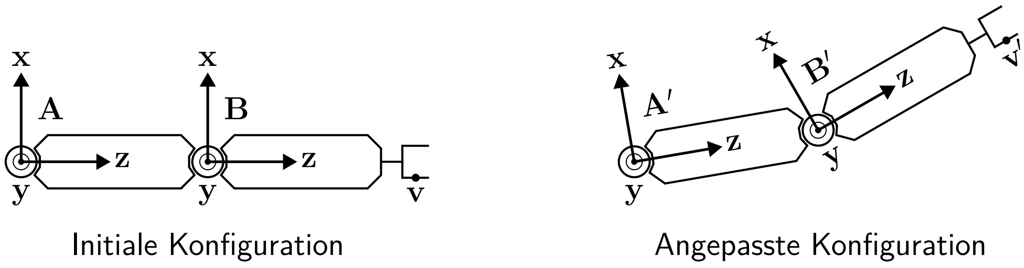


Abbildung 3: Transformation eines Armsegments.

a) Die Transformation von Gelenk $\mathbf{B}'$ wird beschrieben durch eine Rotation um die **y**-Achse mit dem Winkel θ, sowie einer Translation in **z**-Richtung um t Einheiten (Abb. 3). Bestimmen Sie die zugehörige Transformationsmatrix $\mathbf{B}'$ als Produkt einer Translation $\mathbf{T}$ und der Rotation $\mathbf{R}_y(\theta)$! $\mathbf{R}_y$ müssen Sie nicht angeben. **(5 Punkte)**

$\mathbf{T} =$

$\mathbf{B}' =$



Duplikat von Abbildung 2.

b) Die Vertices sind alle in Modellkoordinaten gegeben. Um eine Manipulation der Gelenke anzuwenden muss ein Vertex zunächst in das lokale Koordinatensystem eines Gelenks in initialer Konfiguration transformiert werden. Bestimmen Sie die Transformation $\mathbf{I_v}$ für den Vertex $\mathbf{v}$, die von Modellkoordinaten in das lokale Koordinatensystem des rechten Armsegments führt! **(3 Punkte)**

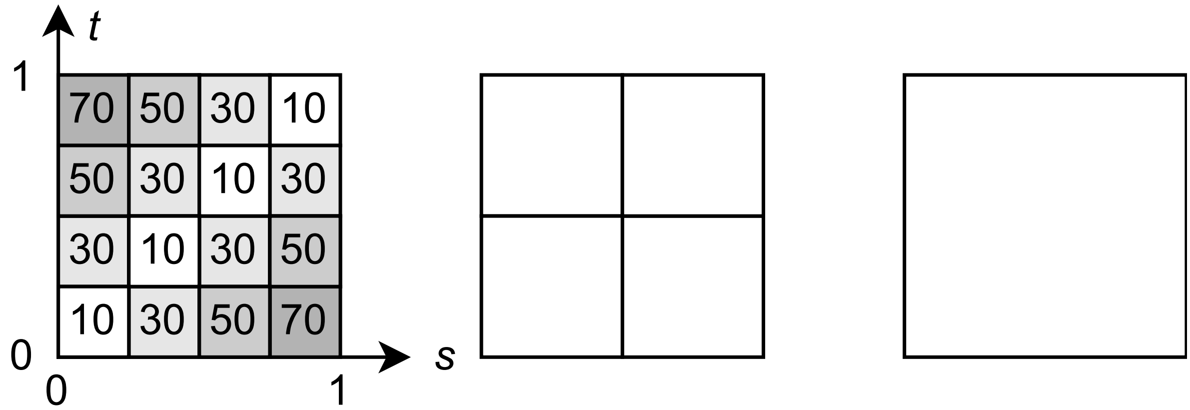
$\mathbf{I_v} =$

c) Mit $\mathbf{I_v}$ aus der vorherigen Teilaufgabe kann der Vertex $\mathbf{v}$ während der Animation wieder in Modellkoordinaten transformiert werden (Abb. 2, rechts). Geben Sie die Gesamttransformation $\mathbf{G_v}$ an, die $\mathbf{v}$ in $\mathbf{v}'$ transformiert! **(3 Punkte)**

$\mathbf{G_v} =$

Matrikelnummer: ____________________

Aufgabe 6: 2D-Texturen (18 Punkte)



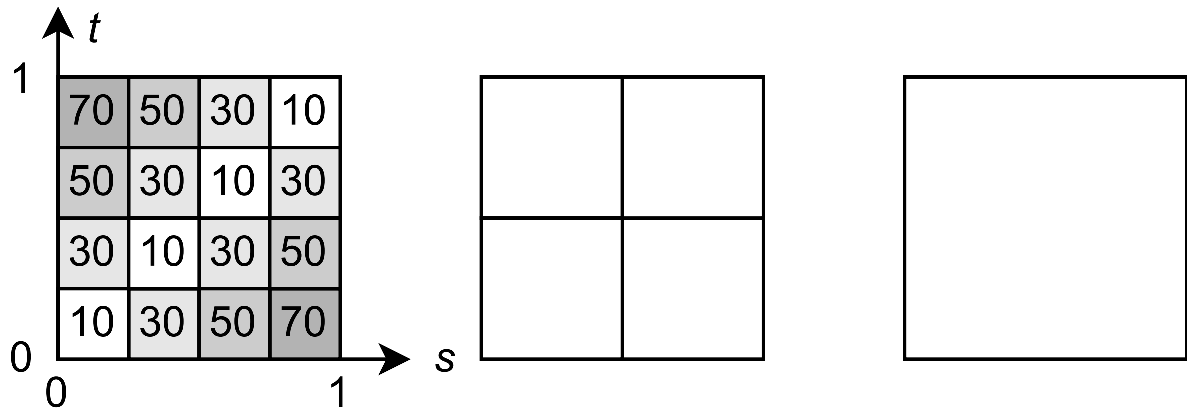
a) Geben Sie die Ergebnisse der folgenden Texturzugriffe mit den angegebenen Adressierungsmodi unter der Verwendung von Nearest Neighbor-Filterung an! **(4 Punkte)**

i) Texturkoordinaten: (s, t) = (0.2, 0.6), Texture Wrapping: *clamp*

ii) Texturkoordinaten: (s, t) = (1.2, 1.6), Texture Wrapping: *repeat*

iii) Texturkoordinaten: (s, t) = (1.4, 0.1), Texture Wrapping: *clamp*

b) Um Artefakte bei *Verkleinerung* (*Minification*) zu reduzieren, werden Texturen mithilfe von Mip-Maps vorgefiltert. Berechnen Sie die Mip-Map-Stufen der Textur in der obigen Abbildung! Sie können zur Korrektur die folgende Abbildung benutzen. *Kennzeichen Sie eindeutig, welche Lösung bewertet werden soll!* **(3 Punkte)**



c) Gegeben sei der Pixelfootprint aus Abbildung 4. Geben Sie an, welche Mip-Map Stufe für die bilineare Interpolation ausgewählt wird und begründen Sie kurz! Dabei entspricht Stufe 0 der Originaltextur. **(3 Punkte)**

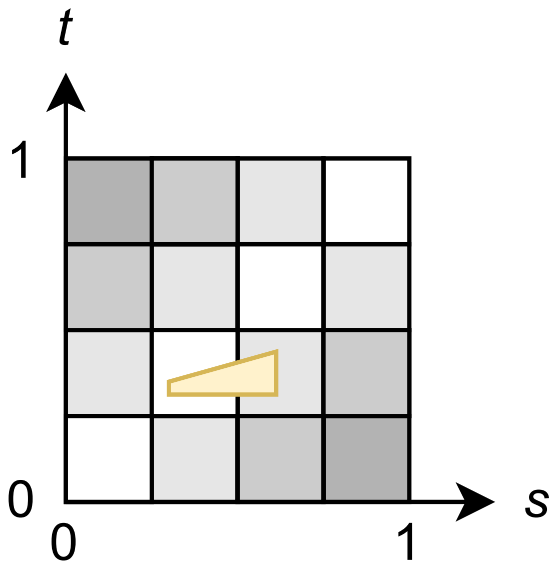


Abbildung 4: Pixelfootprint im Texturraum.

d) Der endgültige Farbwert wird nun durch trilineare Interpolation bestimmt. Nennen Sie, welche Interpolationsschritte dazu durchgeführt werden! Wie viele Texlwerte müssen dazu ausgelesen werden? **(4 Punkte)**

e) Gegeben sei ein Dreieck mit Texturkoordinaten $T_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $T_2 = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$, $T_3 = \begin{pmatrix} 1 \\ 0.25 \end{pmatrix}$ der Eckpunkte. Berechnen Sie die Texturkoordinate $T = \begin{pmatrix} s \\ t \end{pmatrix}$ für die baryzentrischen Koordinaten $\lambda_1 = 0$, $\lambda_2 = 0.5$, $\lambda_3 = 0.5$! **(4 Punkte)**

Aufgabe 7: OpenGL Pipeline (28 Punkte)

a) Geben Sie für die folgenden Aufgaben an, in welcher Pipeline-Stufe in modernem OpenGL diese üblicherweise durchgeführt werden und ob diese frei programmierbar ist oder nicht! **(9 Punkte)**

Hinweis: Manche Aufgaben lassen mehrere Antworten zu.

i) Der Tiefentest wird durchgeführt.

Stufe: Programmierbar:

ii) Das Beleuchtungsmodell wird ausgewertet.

Stufe: Programmierbar:

iii) Alle Vertices eines Dreiecksnetzes werden in eine Richtung verschoben.

Stufe: Programmierbar:

iv) Ein grob aufgelöstes Dreiecksnetz wird in kleinere Dreiecke unterteilt.

Stufe: Programmierbar:

v) Die Transformation zur Berechnung der Clip Space-Koordinaten wird angewandt.

Stufe: Programmierbar:

vi) Zur Darstellung eines spiegelnden Objekts wird auf eine Environment Map zugegriffen.

Stufe: Programmierbar:

b) Betrachten Sie den folgenden Geometry Shader:

```
1  layout (triangles) in;
2  layout (line_strip, max_vertices = 6) out;
3
4  void main() {
5      for(int i = 0; i < 3; i++) {
6          int j = (i + 1) % 3;
7
8          gl_Position = gl_in[ i ].gl_Position;
9          EmitVertex();
10         gl_Position = gl_in[ j ].gl_Position;
11         EmitVertex();
12
13         EndPrimitive();
14     }
15 }
```

i) Beschreiben Sie, welche Ausgabe der Shader aus dem Eingabestrom erzeugt und welche Funktionalität die Zeilen 8 bis 13 dafür umsetzen! **(6 Punkte)**

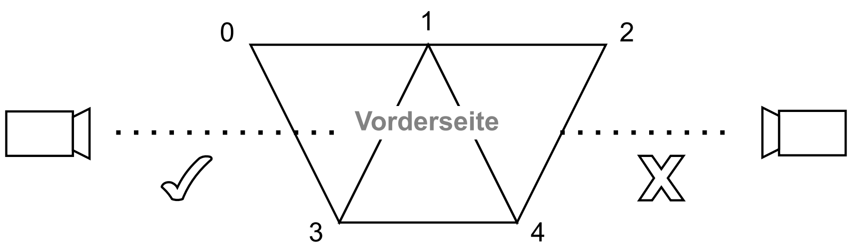
ii) Erklären Sie stichpunktartig, warum die Ausgabe des obigen Geometry-Shaders nicht ideal ist! **(4 Punkte)**

c) Eine Dreiecks-Fläche soll so rasterisiert werden, dass diese nur von der Vorderseite aus betrachtet sichtbar ist. Für eine Kamera, die von der Rückseite auf die Fläche blickt, soll das Objekt unsichtbar sein.

i) Welche Funktionalität der Pipeline kann genutzt werden, um dies umzusetzen? Geben Sie zusätzlich entweder in Worten oder in Form von zwei OpenGL-Befehlen an, wie diese konfiguriert wird! **(3 Punkte)**

ii) Für das Rendering wird ein Index-Buffer verwendet, um auf die Vertices zuzugreifen. In der unteren Abbildung ist eine Fläche aus drei Dreiecken gegeben. Die Indizes der fünf Vertices sind jeweils eingetragen. Befüllen Sie die untenstehenden Index-Buffer so, dass sich die dargestellten Dreiecke ergeben! **(6 Punkte)**

Hinweis: Gehen Sie davon aus, dass die Pipeline nach Ihren vorherigen Anweisungen konfiguriert ist. Wenn Sie die vorherige Teilaufgabe nicht beantwortet haben, gehen Sie vom Standardverhalten aus.



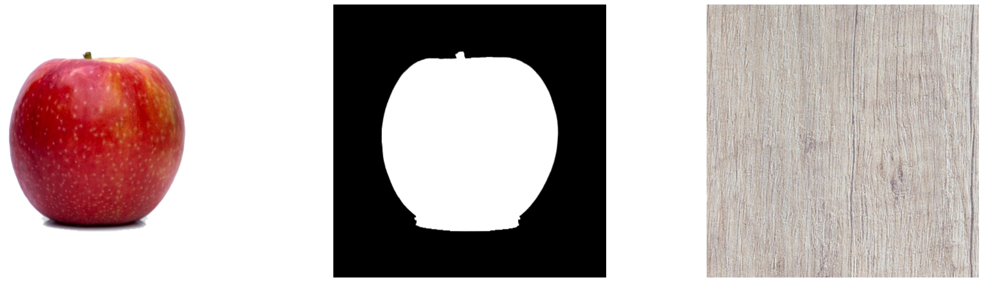
Index-Buffer, wenn `GL_TRIANGLES` als Topologie gezeichnet wird:

Index-Buffer, wenn `GL_TRIANGLE_FAN` als Topologie gezeichnet wird:

Index-Buffer, wenn `GL_TRIANGLE_STRIP` als Topologie gezeichnet wird:

Matrikelnummer: ____________________

Aufgabe 8: Blending-Operationen (6 Punkte)



(a) Source RGB (b) Source Alpha (c) Frame Buffer

Abbildung 5

a) Blending kombiniert die Farbwerte $F_s = (R_s, G_s, B_s, A_s)$ aus dem Fragment Shader mit den Farbwerten $F_d = (R_d, G_d, B_d, A_d)$ im Frame Buffer. Geben Sie für die folgenden Konfigurationen die Berechnungsvorschrift der Farbe F_b beim Blending an! Ordnen Sie außerdem jedem Ergebnis das entsprechende Bild aus Abb. 6 zu! Die Eingaben stehen in Abb. 5. **(6 Punkte)**

Hinweis: Geben Sie die Lösung wie in diesem Beispiel an: $F_b = A_d * F_s - 0 * F_d$.

(a) (b) (c)

Abbildung 6

i) glBlendFunc(GL_ZERO, GL_ONE);
glBlendEquation(GL_FUNC_ADD);

$F_b =$

Abbildung:

ii) glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glBlendEquation(GL_FUNC_ADD);

$F_b =$

Abbildung:

iii) glBlendFunc(GL_ZERO, GL_ONE_MINUS_SRC_ALPHA);
glBlendEquation(GL_FUNC_ADD);

$F_b =$

Abbildung:

Aufgabe 9: OpenGL-Shader (24 Punkte)

In dieser Aufgabe sollen Sie eine Szene durch OpenGL-Shader zeichnen. Die Szene enthält texturierte Dreiecke und einige Punktlichter. Die Beleuchtungsberechnung soll perfekt-diffuse Reflexion im Fragment Shader in Weltkoordinaten berechnen.

a) Vervollständigen Sie den folgenden Vertex Shader: **(12 Punkte)**

   i) Berechnen Sie die Clip-Koordinaten `gl_Position` aus den gegebenen Transformationen!

   ii) Ergänzen Sie `normalTransform(Mat, n)` um die Transformation eines Normalenvektors `n` für eine gegebene Transformation `Mat` zu berechnen!

   iii) Definieren und implementieren Sie die weiteren Ausgaben des Shaders, sodass die Beleuchtungsberechnung im Fragment Shader durchgeführt werden kann!

```
uniform mat4 matM; // Transformationsmatrix von Modell- zu Weltkoordinaten
uniform mat4 matV; // Transformationsmatrix von Welt- zu Kamerakoordinaten
uniform mat4 matP; // Projektionsmatrix von Kamera- zu Clip-Koordinaten

in vec4 P;  // Vertex-Position in homogenen Modellkoordinaten, w = 1
in vec3 N;  // Vertex-Normale in Modellkoordinaten
in vec2 t;  // Vertex-Texturkoordinate

// Definieren Sie hier Ihre Shader-Ausgaben:

out







// Geben Sie den transformierten Normalenvektor zurück:
vec3 normalTransform(mat4 Mat, vec3 n)
{







}

void main()
{









    gl_Position =
}
```

b) Alle Materialien in der Szene sind rein diffus und der Reflexionskoeffizient `kd` soll aus der Textur `textur` ausgelesen werden. Die Positionen und Intensitäten der Punktlichter werden in den Arrays `LPs` und `LIs` übergeben. Die Methode `lambert` berechnet die diffuse Beleuchtung für `kd`, Lichtintensität `I`, sowie Licht- und Normalenvektoren `L` und `N`. Vervollständigen Sie den folgenden Fragment Shader, sodass er die Beleuchtungsberechnung durchführt! **(12 Punkte)**

Hinweis: Die Beleuchtung durch ein Punktlicht nimmt quadratisch mit dem Abstand ab.

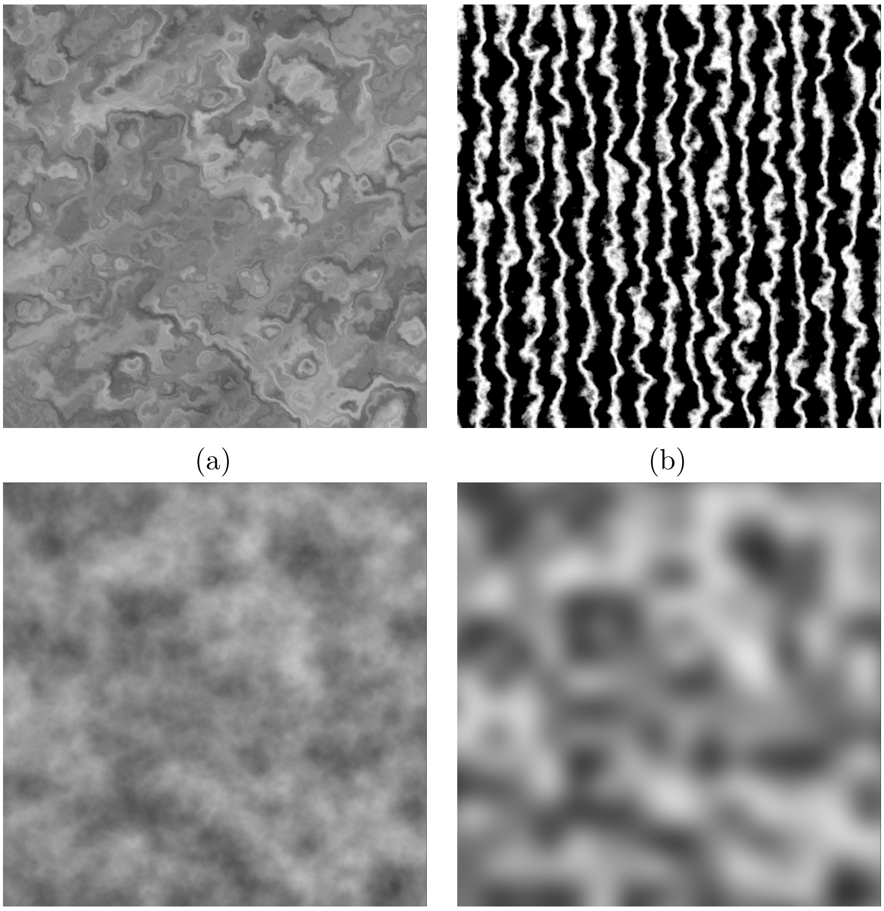
i) Definieren Sie die nötigen Shader-Eingaben!

ii) Lesen Sie `kd` aus der Textur aus!

iii) Akkumulieren Sie die Beleuchtung über alle Lichtquellen in `out_color`!

```
uniform sampler2D textur; // Textur

uniform vec3 LPs[];  // Positionen der Lichtquellen in Weltkoordinaten
uniform vec3 LIs[];  // Intensitäten der Lichtquellen
uniform int Lcount;  // Anzahl der Lichtquellen

out vec4 out_color;

// perfekt diffuses Lambert Beleuchtungsmodell
vec3 lambert(vec3 I, vec3 L, vec3 N, vec3 kd);

// Definieren Sie hier Ihre Shader-Eingaben:

in





void main()
{
    // Lesen Sie hier die Textur aus:

    vec3 kd =

    // Berechnen Sie hier die Farbe des Fragments:










    out_color =
}
```

Aufgabe 10: Prozedurale Modellierung (12 Punkte)



(a) (b) (c) (d)

a) Gegeben sei eine Rauschfunktion $n(\mathbf{x})$, sowie eine Turbulenzfunktion $T(\mathbf{x}) = \sum_{k=0}^{4} \frac{1}{2^k} n(2^k \mathbf{x})$.
Ordnen Sie jede der obigen Texturen ihrer zugehörigen Funktion unten zu! **(4 Punkte)**

$f(\mathbf{x}) = n(\mathbf{x})$:

$f(\mathbf{x}) = T(\mathbf{x})$:

$f(\mathbf{x}) = T(\mathbf{x} + T(\mathbf{x}))$:

$f(\mathbf{x}) = \max(0, \cos(\mathbf{x}_x + T(\mathbf{x})))$:

b) Wie kann Aliasing beim Auswerten einer Turbulenzfunktion einfach vermieden werden? Begründen Sie Ihre Antwort! **(2 Punkte)**

c) Bestimmen Sie die vorzeichenbehaftete 2D-Distanzfunktion $D(\mathbf{x}, r)$ eines im Ursprung zentrierten Kreises mit Radius r! **(2 Punkte)**

$D(\mathbf{x}, r) =$

d) Nutzen Sie $D(\mathbf{x}, r)$, um die vorzeichenbehaftete Distanzfunktion $H(\mathbf{x})$ eines Kreisrings mit versetzter Aussparung wie in Abb. 8 zu modellieren! **(4 Punkte)**

Hinweis: Bestimmen Sie zuerst die Distanzfunktionen der einzelnen Kreise.

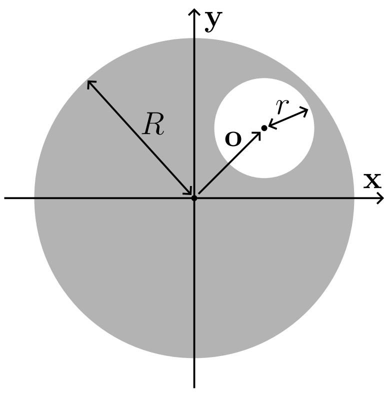


Abbildung 8: Kreisring mit versetzter Aussparung.

$H(\mathbf{x}) =$

Aufgabe 11: Bézier-Kurven (19 Punkte)

Gegeben sei die Bézier-Kurve $F(u) = \sum_{i=0}^{3} \mathbf{b}_i B_i^3(u)$ vom Grad 3 mit $u \in [0,1]$ und

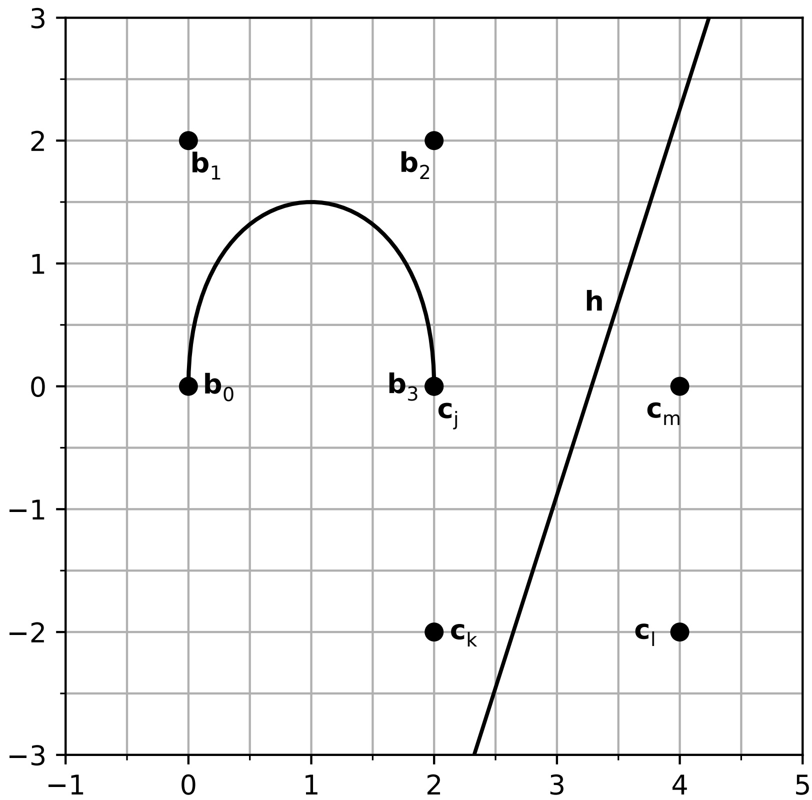
$$\mathbf{b}_0 = (0,0), \qquad \mathbf{b}_1 = (0,2), \qquad \mathbf{b}_2 = (2,2), \qquad \mathbf{b}_3 = (2,0).$$

sowie eine weitere Bézier-Kurve $G(u) = \sum_{i=0}^{3} \mathbf{c}_i B_i^3(u)$ vom Grad 3 mit $u \in [0,1]$ und den Kontrollpunkten

$$\mathbf{c}_j = (2,0), \qquad \mathbf{c}_k = (2,-2), \qquad \mathbf{c}_l = (4,-2), \qquad \mathbf{c}_m = (4,0),$$

d.h. die Reihenfolge der Kontrollpunkte c_i ist nocht nicht festgelegt.

a) Gegeben sei zusätzlich die Gerade h, siehe Zeichnung. Wie oft schneiden die möglichen $G(u)$ die Gerade maximal ohne eine aus der Vorlesung bekannte Eigenschaft von Bézier-Kurven zu verletzen? Begründen Sie unter Angabe der Eigenschaft und erläutern Sie diese kurz! **(6 Punkte)**



Maximale Anzahl der Schnitte: _____

Name der Eigenschaft der Bézier-Kurve: ______________________________________

Erläuterung: ______________________________________

____

____

____

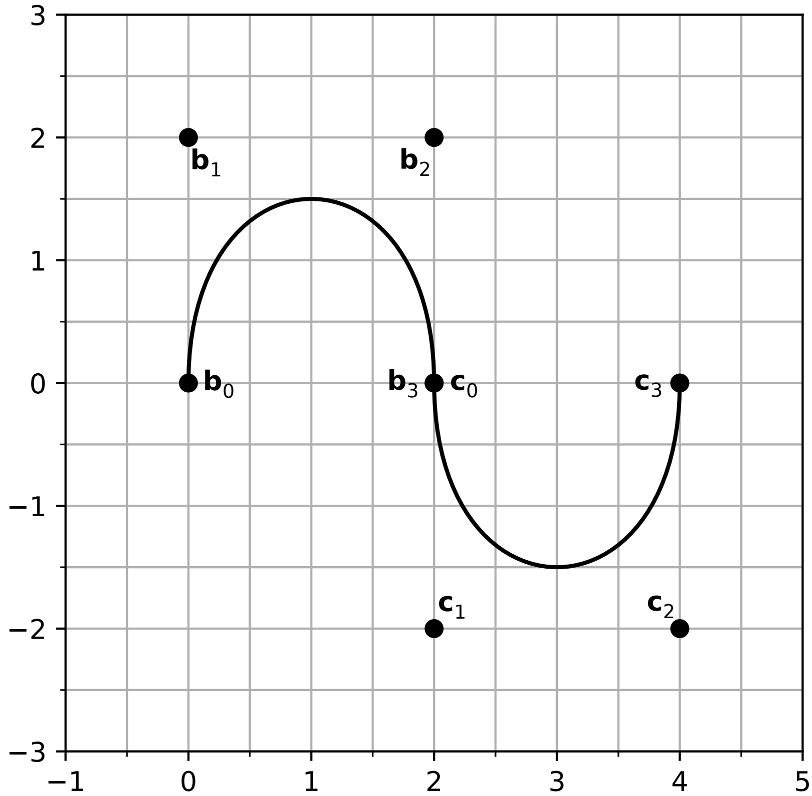
b) Geben Sie je eine mögliche Reihenfolge der Kontrollpunkte von G an, so dass der Übergang von F nach G

   i) C_{-1} stetig ist: $j =$ ___, $k =$ ___, $l =$ ___, $m =$ ___

   ii) C_0, aber nicht C_1 stetig ist: $j =$ ___, $k =$ ___, $l =$ ___, $m =$ ___

   (4 Punkte)

c) Sei die Reihenfolge der c_i wie in der Zeichnung gezeigt als $j = 0, k = 1, l = 2, m = 3$ festgelegt. Ist der Übergang zwischen F und G C^2-stetig? Begründen Sie! **(5 Punkte)**



d) Unterteilen Sie die Kurve $F(u)$ *zeichnerisch* für $u = 0.25$. Machen Sie dabei die Kanten und Punkte der Kontrollpolygone der Teilkurven kenntlich! Zur Korrektur können Sie die zweite Zeichnung darunter verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnung bewertet werden soll!* **(4 Punkte)**

Hinweis: Alle Punkte der Kontrollpolygone befinden sich auf dem gegebenen Gitter.