Visual Computing

Fachbereich Informatik

Studiengang Bachelor Informatik / Bachelor Informatik dual

h_da hochschule darmstadt

member of EUT+ EUROPEAN UNIVERSITY OF TECHNOLOGY

Wintersemester 2025/2026

Björn Frömmer und Elke Hergenröther

Vorname: _____

Nachname: _____

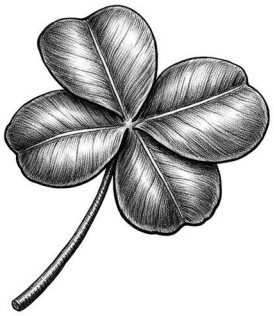
Matrikelnummer: _____

Punkte _____ **+ Zusatzpunkte** _____ **= Summe** _____

Note: _____

Hinweise zur Klausur: Zur Klausur zugelassen sind ein nicht programmierbarer Taschenrechner und ein beschriebenes Blatt DinA4-Papier. Weitere Hilfsmittel sowie Schmierpapier sind nicht erlaubt.

Falls Sie Schwierigkeiten mit der deutschen Sprache haben:
Es gibt eine Übersetzungshilfe für die Klausur ins Englische. Prüfungsrelevant sind aber die deutschen Aufgabenstellungen.

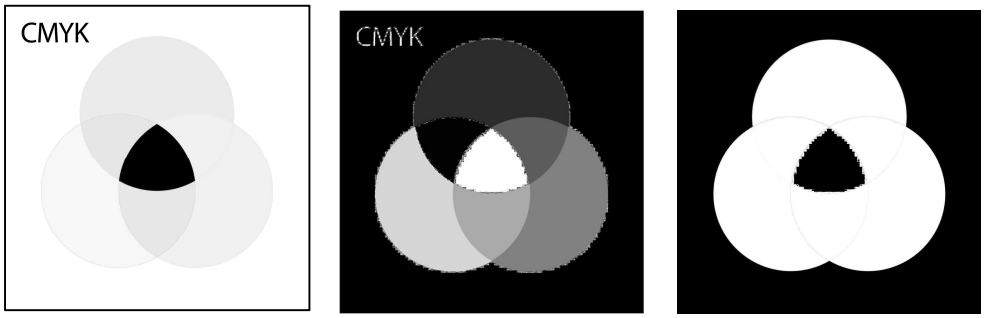


Wir wünschen Ihnen viel Glück und Erfolg

1. Aufgabe: RGB und HSV Farbmodelle

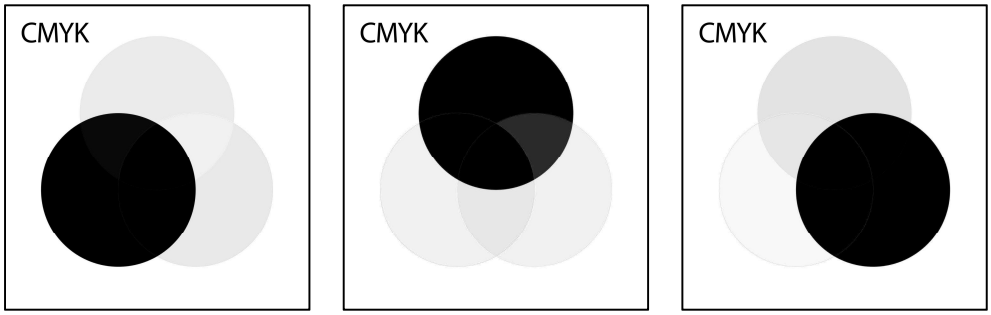
Sie sehen hier ein Bild mit drei sich überlappenden Farbkreisen (oben Gelb, unten links Magenta, unten rechts Cyan) auf weißem Hintergrund, einmal in den RGB und einmal in den HSV-Farbraum zerlegt. Dabei gehören jeweils drei Grauwertbilder zusammen. **Tipp:** Weiß = Maximalwert, Schwarz = Minimalwert.

Reihe A:



a.) b.) c.)

Reihe B:



Punkte (/ 4%)

Teil a: Entscheiden Sie begründet welche Bildreihe zu welchem Farbraum passt.

Punkte (/ 9%)

Teil b: Ordnen Sie den einzelnen Bildern der Reihen A und B und Spalten a.) bis c.) dem entsprechenden Farbkanal zu. Bitte begründen Sie auch diese Entscheidungen!

insg. ____ / 13 % erreicht

2. Aufgabe: OpenGL Shader

Unten sehen Sie einen Vertex- und einen Fragment-Shader. Beantworten Sie dazu die Fragen in Teil a. bis Teil d.

Vertex Shader

```
#version 330
layout(location = 0) in vec3 vertex;
layout(location = 1) in vec3 vertex_normal;
layout(location = 2) in vec3 vertex_color;
uniform mat4 modelMatrix;
uniform mat4 viewMatrix;
uniform mat4 projMatrix;
out vec3 colorVS;

void main() {
    gl_Position = projMatrix * viewMatrix * modelMatrix * vec4(vertex, 1.0);
    colorVS = vertex_color;
}
```

Fragment Shader

```
#version 330
in vec3 colorVS;
out vec4 color;

void main() {
    color = vec4(colorVS, 1.0);
}
```

Punkte (/ 3%)

Teil a: Worauf verweisen die *layout(location = index)* Einträge im Vertex Shader?

Punkte (/ 3%)

Teil b: Was ist eine *uniform* Variable und wozu werden diese verwendet?

insg. ____ / 6 % erreicht

Punkte (/ 3%)

Teil c: Wozu wird der Variablentyp *out* verwendet?

Punkte (/ 10%)

Teil d: Füllen Sie den Lückentext korrekt aus:

Der Vertex-Shader bekommt die Vertex-Daten (Position, Normale und Farbe) übertragen und bekommt zusätzlich drei _____-dimensionale Matrizen übermittelt, die die _____ Transformationen sowie die _____- und die _____ -Matrix enthalten. Der Vertex-Shader berechnet mit Hilfe dieser 3 Matrizen die finale Vertex-_____ und gibt außerdem die _____ des Vertex weiter an den Fragment Shader.

Der Fragment Shader wandelt einen Farbwert von einem _____-dimensionalen in einen _____-dimensionalen Vektor um, indem der Farbe ein _____-Wert hinzugefügt wird. Am Ende gibt der Fragment-Shader die finale _____ des Fragments weiter.

Punkte (/ 3%)

Teil e: Wofür stehen die Parameter in folgender OpenGL-Funktion. Reihenfolge der Parameter wird nicht gewertet: *gluLookAt(* $a_x, a_y, a_z,\ b_x, b_y, b_z,\ c_x, c_y, c_z$ *);*

a =

b =

c =

Punkte (/ 3%)

Teil f: Beschreiben Sie möglichst genau, was die einzelnen OpenGL-Funktionsaufrufe bewirken:

1) *glEnable(GL_DEPTH_TEST);*
2) *glDepthFunc(GL_LESS);*
3) *glClearDepth(1.0);*

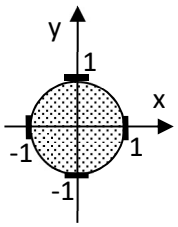
insg. ____ / 19 % erreicht

3. Aufgabe: Affine Transformationen

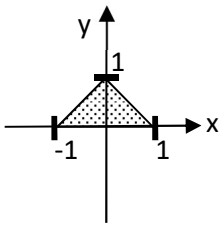
Punkte (/ 10%)

Teil a: Szenegraphen

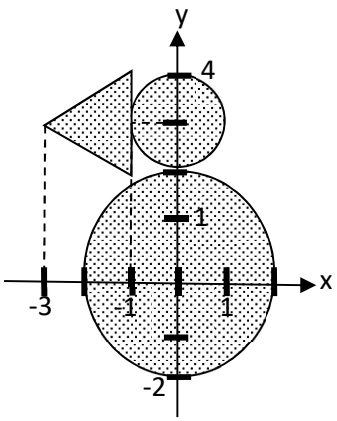
Aus den 2D-Elementen `Kreis` und `Dreieck` (s. u. links) soll mit Hilfe des ganz unten gezeigten Szenegraphen die unten rechts gezeigte Szene modelliert werden. Alle Koordinatensysteme sind rechtshändig.



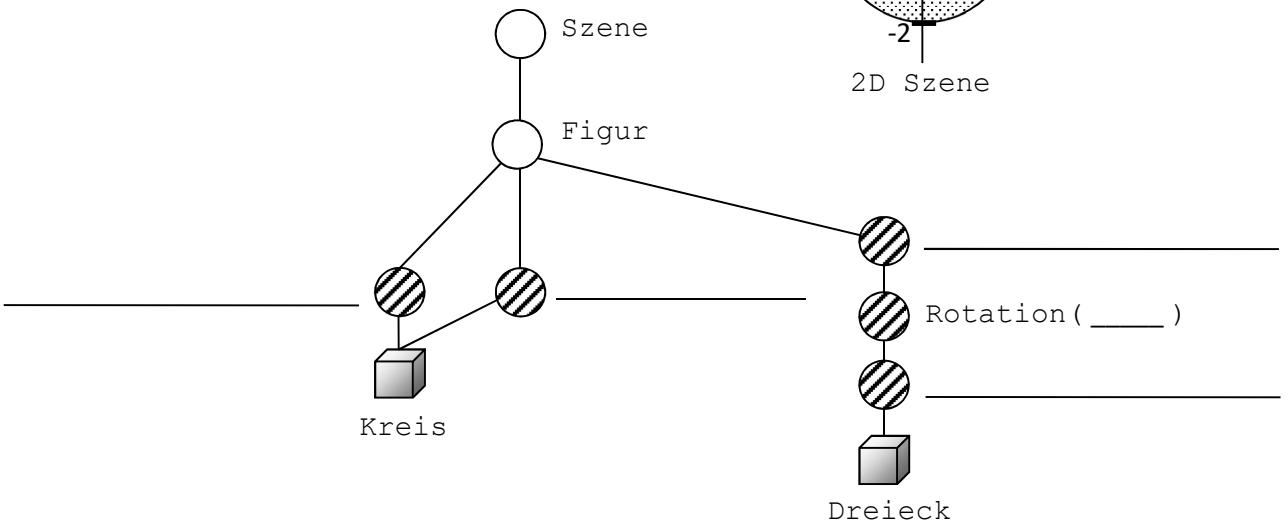
`Kreis`



`Dreieck`



`2D Szene`



Bitte tragen Sie auf die eingezeichneten Striche neben dem Szenengraphen den jeweiligen Namen der notwendigen Transformation und die entsprechenden Parameterwerte ein. Es gilt: auf jedem Strich muss eine Transformation eingetragen werden, die eine Auswirkung hat: Skalierung (1., 1.) ist also keine zulässige Transformation.

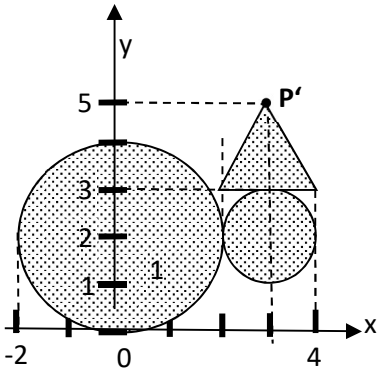
Tipp: Achten Sie auf die Reihenfolge, in der die Transformationen im Szenengraph ausgewertet werden.

insg. ____ / 10 % erreicht

Punkte (/ 6%)

Teil b: Homogene Transformationen

Wie lautet die Transformation M in homogenen Koordinaten, die sich im Szenengraph in Teil a zwischen dem Gruppierungsknoten `Szene` und `Figur` befindet und zu der transformierten Szene führt, die Sie im Koordinatensystem (rechts) sehen. Beschreiben oder Berechnen Sie die Herleitung der homogonen Matrix und berechnen Sie eine Probe für P‘ (3, 5).



Rotierte 2D Szene

4. Aufgabe: Beleuchtung nach Phong

Auf dem Bild unten sehen Sie eine Punkt-Lichtquelle, eine **diffus** & **spiegelnd** reflektierende Oberfläche, einen Punkt **P** auf der Oberfläche und eine Kamera.

Punkte (/ 6%)

Teil a: Zeichnen Sie **alle** zur Berechnung des Phong’schen Reflexionsmodells nötigen Vektoren sowie die Winkel **α** und **β** für die Reflexion am Punkt **P** ein.

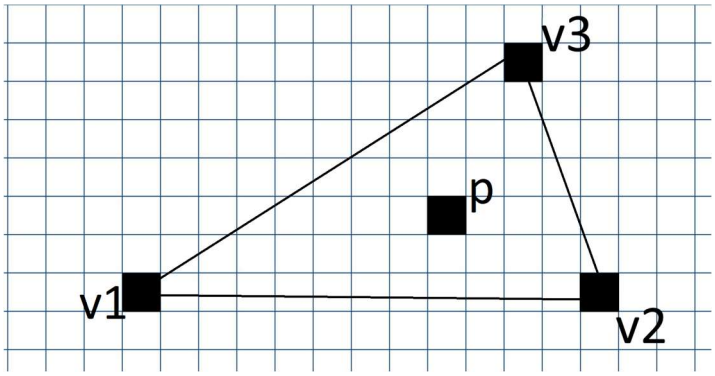
Punkte (/ 4%)

Teil b: Stellen Sie die Gleichung zur Berechnung Helligkeit der diffusen und der spiegelnde (spekulare) Reflexion an dem Punkt P auf einer Skala zwischen 0 und 1 auf. Mit **α = 35°** (für die diffuse Reflexion) und **β = 20°** (für die spiegelnde Reflexion). Der Exponent ***k*** für die spiegelnde Reflexion soll **15** betragen.

insg. ____ / 16 % erreicht

4. Aufgabe: Bilineare Interpolation

Auf dem Bild unten ist ein Dreieck zu sehen. Die 3 Vertices (diese sind zum besseren Verständnis bereits als Pixel eingezeichnet) enthalten RGB-Farbinformationen. Die Farben der Vertices des Dreiecks sind **v1 = Rot = [1, 0, 0]**, **v2 = Grün = [0, 1, 0]** und **v3 = Blau = [0, 0, 1]**.



Punkte (/ 4%)

<u>Beschreiben</u> Sie kurz Ihr Vorgehen, um den RGB-Farbwert des Punktes **p** mittels bilinearer Interpolation zu berechnen.

Punkte (/ 6%)

<u>Berechnen</u> Sie nachvollziehbar (mit Rechenweg) den RGB-Farbwert des Punktes **p** mittels bilinearer Interpolation.

insg. ____ / 10 % erreicht

5. Aufgabe: Bildverarbeitung, Lokale Operatoren:

Punkte (/ 4%)

Teil a: Warum muss die Größe von Faltungskerneln aus einer Kombination ungerader Zahlen (3x3, 5x5, 7x7, …) bestehen?

Punkte (/ 2%)

Teil b: Was bewirkt der Faltungsoperator F1?

F1 =

0	0	1
0	-1	0
0	0	0

Punkte (/ 4%)

Teil c: Was bewirkt ein Mittelwertoperator bei dessen Anwendung auf eine Rastergrafik? Schreiben Sie außerdem einen möglichen Mittelwertoperator auf, der sich **nur** auf horizontale Kanten auswirkt, und nach dessen Anwendung **keine** weitere Grauwerttransformation notwendig ist.

Punkte (/ 4%)

Teil d: Was unterscheidet das Ergebnis der Faltung mit einem Mittelwertoperator von der Anwendung des Median Rangfolgeoperators?

insg. ____ / 14 % erreicht

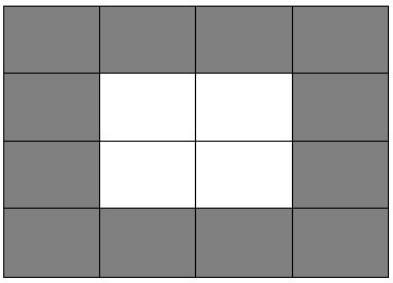
Punkte (/ 6%)

Teil d: Wenden Sie den Laplace-Operator auf die untenstehende Rastergrafik an und schreiben Sie die entstehende Matrix auf. Starten Sie mit der Faltung bei Pixel [1,1], so dass die Ergebnismatrix eine Zeile und Spalte kleiner wird. Geben Sie den Rechenweg an.

50	127	127	200
50	150	150	200
50	150	150	200
50	127	127	200

Rastergrafik

Laplace-Operator

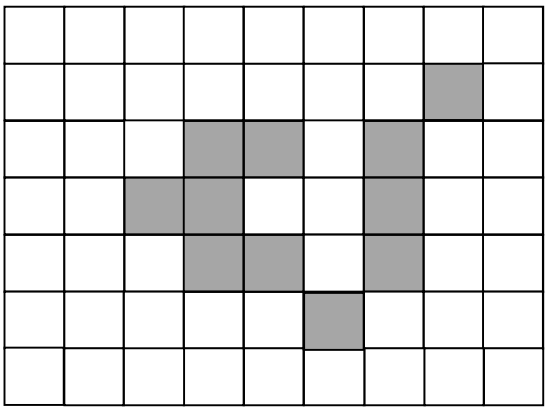


Ergebnis der Faltung

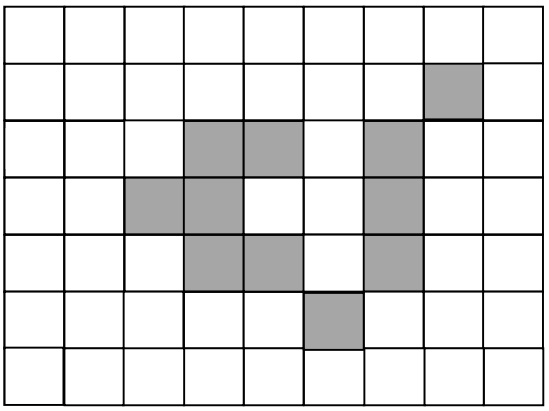
Punkte (/ 6%)

Teil e: Bilden Sie die ZHKs mittels rekursiven Flutens (Flood Fill Algorithm) und nummerieren Sie die Pixel der ZHKs. Im linken Bild sollen die ZHKs mit einer N4 und im rechten Bild mit einer N8 Nachbarschaft gebildet werden.

N4-ZHK



N8-ZHK



schwarz

weiß

insg. ____ / 12 % erreicht

Zusatzaufgaben:

6. Aufgabe:

Nach einer Faltung liegen die Pixelwerte eines Bildes im Intervall von -2 bis 10. Gesucht sind zwei lineare Transformationen (g'(i,j) = e(i,j) * mult + add), die die Pixelwerte in darstellbare Grauwerte überführen:

Punkte (/ 4%)

Teil a: Die erste gesuchte lineare Grauwerttransformation bildet das gegebene Bild auf Grauwerte im Intervall von 0 bis 36 ab. Bestimmen Sie die entsprechende Grauwerttransformation:

Beschreiben Sie, wie das Histogramm des in Teil a.) transformierten Bildes aussehen wird.

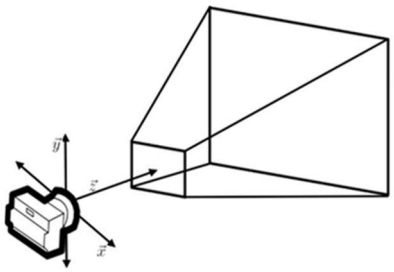
Punkte (/ 4%)

Teil b: Die zweite gesuchte linearen Transformation bildet das gefaltete Bild, dessen Pixelwerte im Intervall zwischen -2 und 10 liegen, auf Grauwerte im Intervall von 3 bis 9 ab. Bestimmen Sie die entsprechende lineare Transformation.

Wenn man die unter b) entwickelte lineare Transformation auf ein beliebiges Grauwertbild anwendet, ist dann das Ergebnisbild dieser lineare Transformation kontrastreicher oder kontrastärmer als das Ausgangsbild? Begründen Sie Ihre Aussage.

Punkte (/ 2%)

Teil c: Geben Sie die Parameter für die OpenGL-Funktion zur Berechnung der perspektivischen Kameratransformation an (Reihenfolge der Parameter spielt keine Rolle):
*glm::perspective(**a, b, c, d**)*



a =

b =

c =

d =

insg. ____ / 10 % erreicht