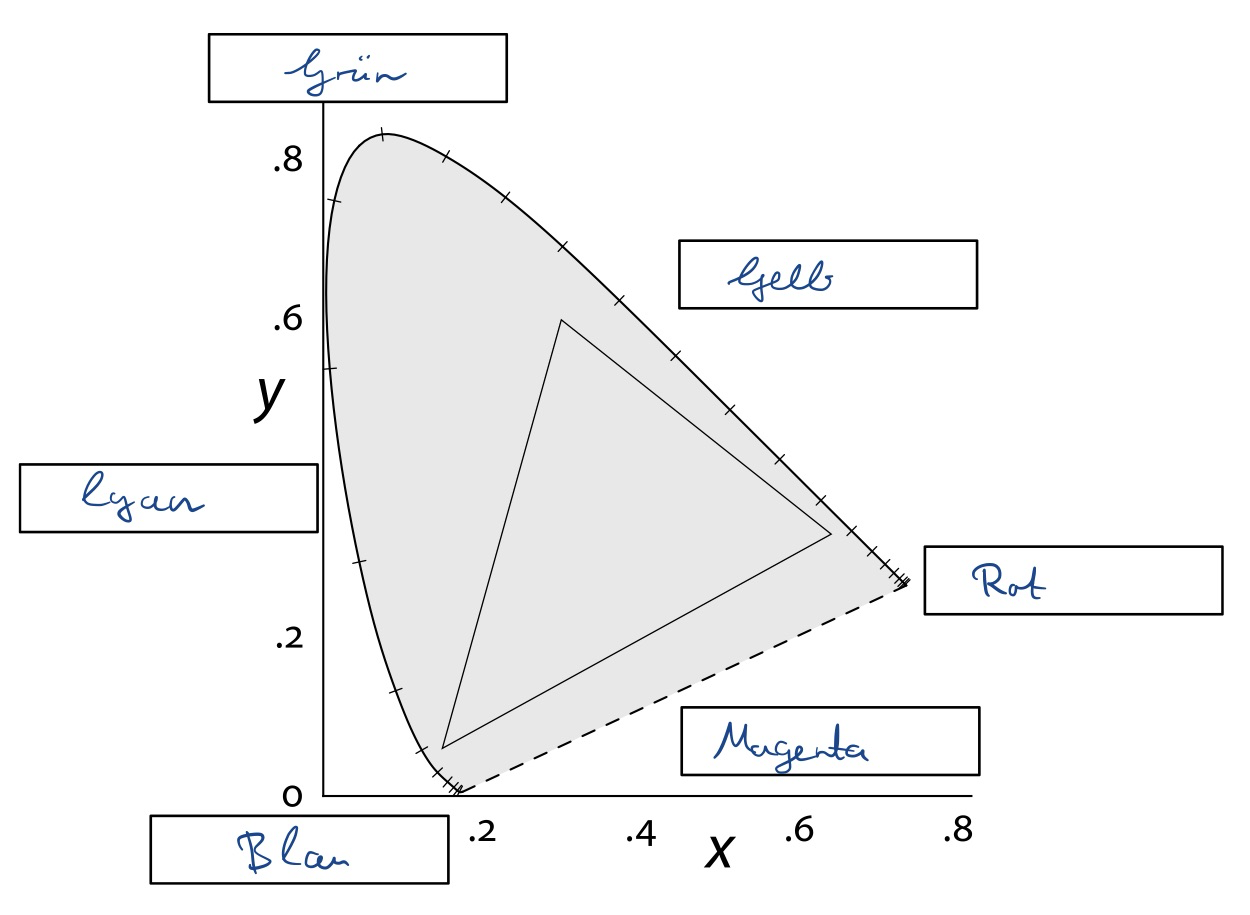
Relativ neue Lösung. Keine Garantie auf Richtigkeit. Bitte Fehler korrigieren oder kommentieren!

Aufgabe 1

a)



b)

Das Dreieck stellt den RGB-Farbraum dar.

Das Gamut eines Farbraums

c)

Die gestrichelte Linie wird auch Purpurlinie genannt und erhält im Gegensatz zur durchgezogenen Rand kein monochromatisches Licht, d. h. keine Spektralfarben, die aus einer bestimmten Wellenlänge bestehen.

d)

Die subjektiv wahrgenommene Intensität ist proportional zum Logarithmus der physikalischen Intensität.

e)

L = Luminanz

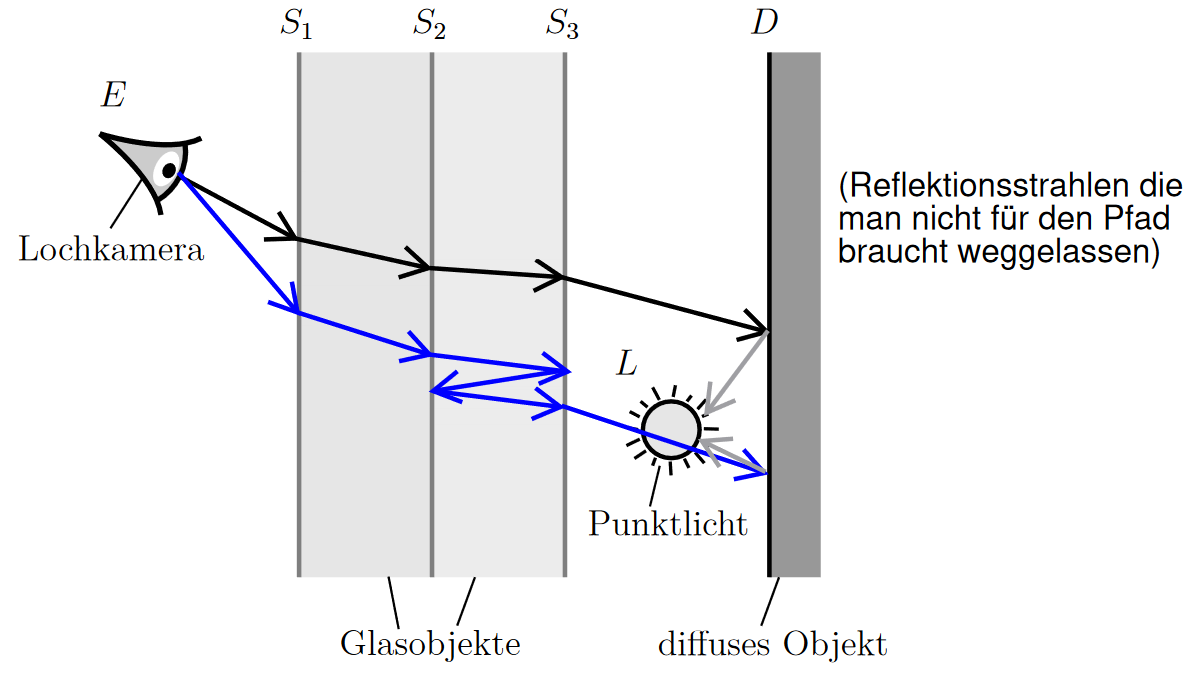
a = rot-grün Anteil

b = blau-gelb Anteil

f)

Euklidische Abstände entsprechen dem wahrgenommenen Farbunterschieden.

Aufgabe 2

a & b)  


Schwarz = Primärstrahl, Grau = Schattenstrahl

Blau = b)

c)

Gar nicht, da Whitted-Style Raytracing schattenstrahlen bei Transmission abbrechen.

d)

So ein Pfad besteht immer aus Primärstrahl (1)

Zwischen S\_1 und S\_2 immer ein Transmissionsstrahl (und für i Reflexionen zwischen

S\_1 und S\_2 betrachten, erhalten wir 2i Reflexionsstrahlen)

Zwischen S\_2 und S\_3 analog

Auch wenn wir an S\_3 reflektieren und in das erste Glasobjekt eintreten treten Reflexions-

bzw. Transmissionsstrahlen immer paarweise auf

Nach S\_3 Transmissionsstrahl und Schattenstrahl (2)

Also insgesamt 5 Strahlen (Primärstrahl, drei Transmissionsstrahlen und ein

Schattenstrahl) und paarweise Transmissions- bzw. Reflexionsstrahlen, also ungerade

Aufgabe 3

Anmerkung a.) und b.): Ich glaube, alle Vektoren müssen normiert sein. Bin mir aber nicht sicher. Sonst wäre eine doppelt so weit entfernte Lichtquelle mit gleicher Intensität doppelt so hell. (L wäre doppelt so lang)

a)

b)

0,75 \* 500 \* <(0, 1, 0), L\_p - (0, 0, 0)> = 0,75 \* 500 \* (0 \* 3 + 1 \* 4 + 0 \* 0) = 1500

L Normiert: 300

c)

Blinn-Phong-Beleuchtungsmodell (Abwandlung des Phong-Modells):

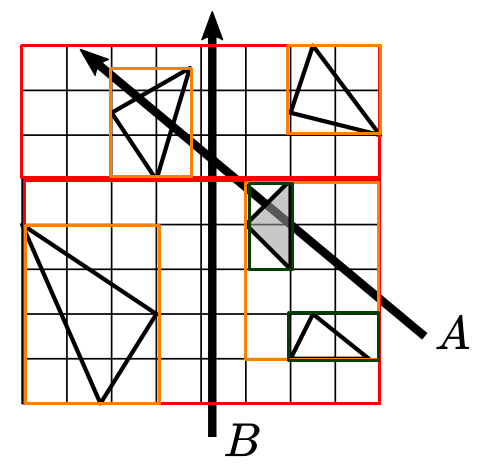
Ändert spekularen Anteil:

mit , der sogenannte Halfway-Vektor.

d)

Skalarprodukt des diffusen Terms wäre negativ. Wir sind aber nur an Richtungen interessiert, die positiv sind -> Strenggenommen <N, L>^+ = max(0, <N, L>) => diffuser Term fällt weg, ambienter und spekularer Term bleiben gleich.

Aufgabe 4

a)  


b)

5 (Wurzelknoten, unterer roter Teil, rechter orangener Teil, unteres Dreieck: kein Treffer, oberes Dreieck: Treffer).

c)

Weil das graue Dreieck ein Blattknoten ist.

d)

3 (Wurzelknoten, unterer Teil, oberer Teil)

e)

* Reguläre Gitter
* Verschachtelte Gitter
* Oktalbaum /Octree
* BSP-Tree
* kD-Baum

f)

Gitter, Octree, k\_d und BSP profitieren allesamt davon. Mailboxing erlaubt es mehrere Schnitttests mit Objekten zu vermeiden, indem geschnittene Objekte als getestet markiert werden.

Aufgabe 5

a)

Der Vektor wird um Achse y im 90° Winkel gedreht.

b)

A^T \* A =

(1 0 0,

0 1 0,

0 0 1)

= Einheitsmatrix.

c)

B =

(1 0 0 0)

(0 1 0 4)

(0 0 1 0)

(0 0 0 1)

d)

C =

(2 0 0 0)

(0 2 0 0)

(0 0 2 0)

(0 0 0 1)

e)

D = B \* C, denn Dx = (B \* C) \* x = B (C x).

f)

B^(-1) =

(1 0 0 0)

(0 1 0 -4)

(0 0 1 0)

(0 0 0 1)

C^(-1) =

(½ 0 0 0)

(0 ½ 0 0)

(0 0 ½ 0)

(0 0 0 1)

D^(-1) = (B \* C)^(-1) = C^(-1) \* B^(-1) =

(½ 0 0 0)

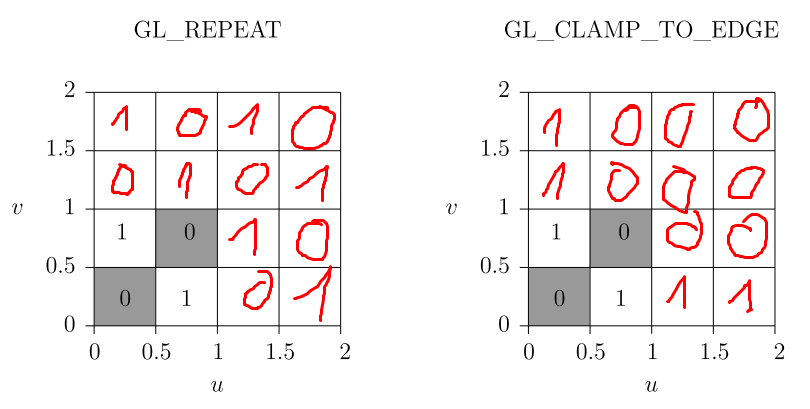
(0 ½ 0 -2)

(0 0 ½ 0)

(0 0 0 1)

Aufgabe 6

a)



b)

Trilineare Interpolation wird bei Mip-Mapping verwendet:

Dabei wird auf den zwei passenden Mip-Map Stufen bilinear interpoliert und anschließend zwischen den Ergebnissen nochmal linear interpoliert.

c)

Nur die Richtung r des reflektierenden Strahls wird beim Environment Mapping verwendet, der Ausgangspunkt x wird ignoriert.  
Vereinfachung: Umgebung ist unendlich weit entfernt

d)

A), denn bei B) wird die Textur im hinteren Bereich versucht sehr exakt abzutasten, wodurch Alias-Effekte auftreten. Bei A gibt es keine Abtastfehler, da die Textur nur eine Farbe hat.

e)

P = λ\_A \* (0, 0) + λ\_B \* (4, 0) + λ\_C \* (2, 3)

Es gilt λ\_A = 1 - λ\_B - λ\_C und damit bleibt ein LGS mit 2 Gleichungen und 2 Unbekannten.

P =>

λ\_C \* (2, 3) = 1 => λ\_C = ⅓   
λ\_B \* (4, 0) + ⅓ \* (2, 3) = 2 => λ\_B \* (4, 0) = 4/3 => λ\_B = ⅓

λ\_A + λ\_B + λ\_C = 1 => λ\_A = 1 - ⅓ - ⅓ = ⅓

Aufgabe 7

a)

Gouraud: Berechnung der Beleuchtung pro Vertex; Interpolation der Farbe (wird im Vertex-Shader implementiert)

Phong: Interpolation der Normale; Berechnung der Beleuchtung pro Pixel (wird im Fragment-Shader implementiert)

b)

Backface Culling: Flächen auf deren Rückseite man schaut, können entfernt werden.

Vorteil: Speedup, da Objekte entfernt werden und in den nachfolgenden Schritten nicht mehr beachtet werden müssen.

Unterschied: Standard-Festlegung ist, dass ein Polygon von vorne gesehen wird, wenn die Vertices am Bildschirm entgegen des Uhrzeigersinns verlaufen. Es kann aber explizit festgelegt werden, was als Vorderseite angesehen wird.

Shader Stufen: Vertex, ?Geometrie?

Aufgabe 8

a)

1 \* c\_s + 1 \* c\_d

b)

In diesem Fall nicht, da glBlendFunc(GL\_ONE, GL\_ONE)

im Fall glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA) wäre der Vorgang nicht mehr kommutativ.

c)

Alpha-Test

Aufgabe 9

a)

void main() {

gl\_Position = MVP \* vec4(position, 1.0);

shading\_normal\_ws =

vec3(transpose(inverse(MV)) \* vec4(normal, 1.0));

position\_ws = vec3(MV \* vec4(position, 1.0));

tex\_uv = uv;

}

b)

void main() {

mat3 TBN = transpose(mat3(Tx, Ty, n));

vec3 mapped\_normal = texture(normalmap, tex\_uv).xyz;

vec3 N = normalize(mapped\_normal);

vec4 kd = texture(diffuse, tex\_uv);

vec3 L = normalize(TBN \* normalize(Lpos - position\_ws));

float length = length(Lpos - position\_ws);

out\_color = vec4(Lcol, 1.0) \* kd \* dot(N, L) \* pow(length, -2);

}

Aufgabe 10

a)

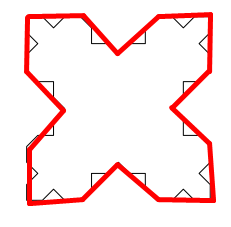
S = F++F++F++F

P = F -> F+F- -F+F

θ = 45

n =1: F+F- -F+F ++ F+F- -F+F ++ F+F- -F+F ++ F+F- -F+F

n =2: F+F- -F+F + F+F- -F+F- - F+F- -F+F + F+F- -F+F ++

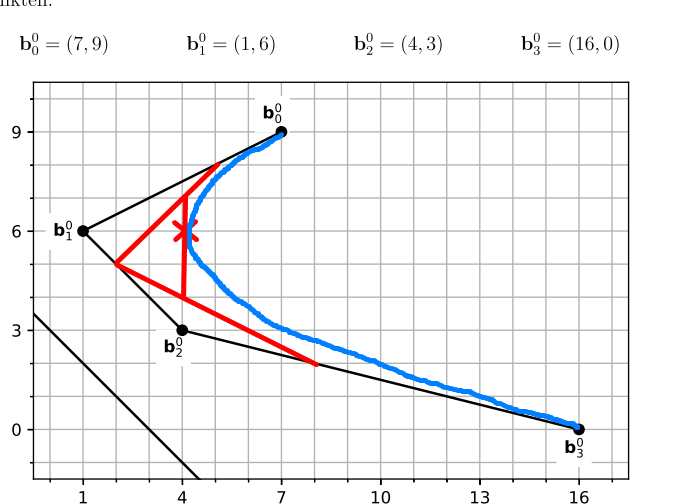
b)  


Aufgabe 11

a)

Nein, da die Bezier-Kurve sich innerhalb der konvexen Hülle der Kontrollpunkte bewegt und die Gerade außerhalb ist.

b)



Schwarz = b^0

Rot = b^1 & b^2

Blau = Bézier-Kurve