Computergrafik 1

Übungsblatt 6

Aufgabe 1 i

- Wichtige Begriffe:
  - Richtung der Projektion (dop)
  - Projektionsebene/Sichtebene
  - Normale der Sichtebene (vpn)
  - Projektoren: Strahlen, die die Projektion charakterisieren

Aufgabe 1 i

Parallelprojektionen

Rechtwinklige Parallelprojektion

- Seitenansicht
- Vorderansicht
- Draufsicht

Axonometrische Parallelprojektion:

- isometrisch
- Dimetrisch
- Trimetrisch

Schiefwinklige Parallelprojektion

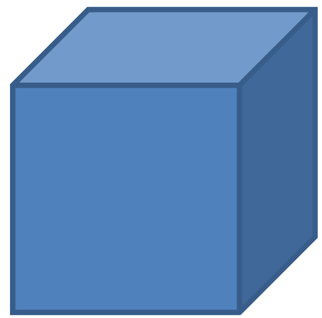
- Kavalierperspektive
- Kabinettperspektive
- …

Aufgabe 1i

- Rechtwinklige Parallelprojektion:
  - Normale der Sichtebene und Richtung der Projektion stimmen überein
- Schiefwinklige Parallelprojektion:
  - Normale der Sichtebene und Richtung der Projektion stimmen nicht überein
- Projektionszentrum: im Unendlichen –> Projektoren verlaufen parallel
- Axonometrische Parallelprojektion:
  - Normale der Sichtebene nicht parallel zu einer der Koordinatenachsen
  - Isometrisch: Winkel zwischen vpn und allen drei Koordinatenachsen gleich
  - Dimetrisch: Winkel zwischen vpn und zwei Koordinatenachsen gleich
  - Trimetrisch: Winkel zwischen vpn und allen drei Koordinatenachsen unterschiedlich

Schiefwinklige Projektionen

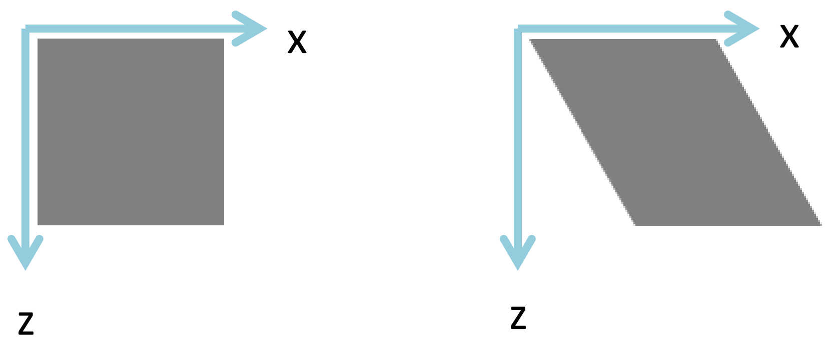
- In OpenGL ist folgendes nicht standardmäßig möglich:



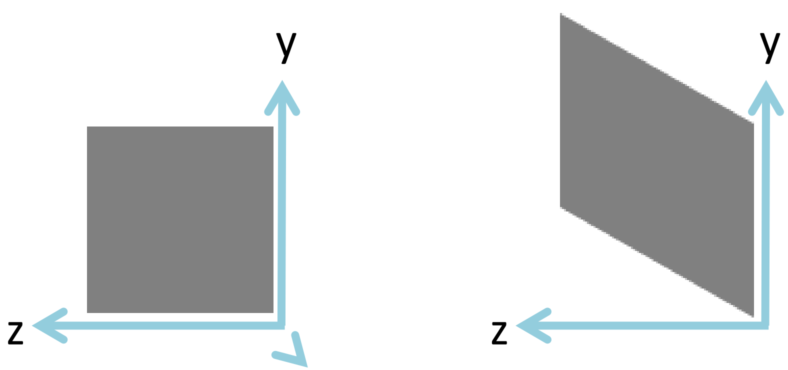
- Schiefwinklige Projektion = Scherung + orthogonale Projektion

Schiefwinklige Projektionen

Top View:



Side View:



3-dimensionale Scherung entlang der z-Achse:

$$\begin{pmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

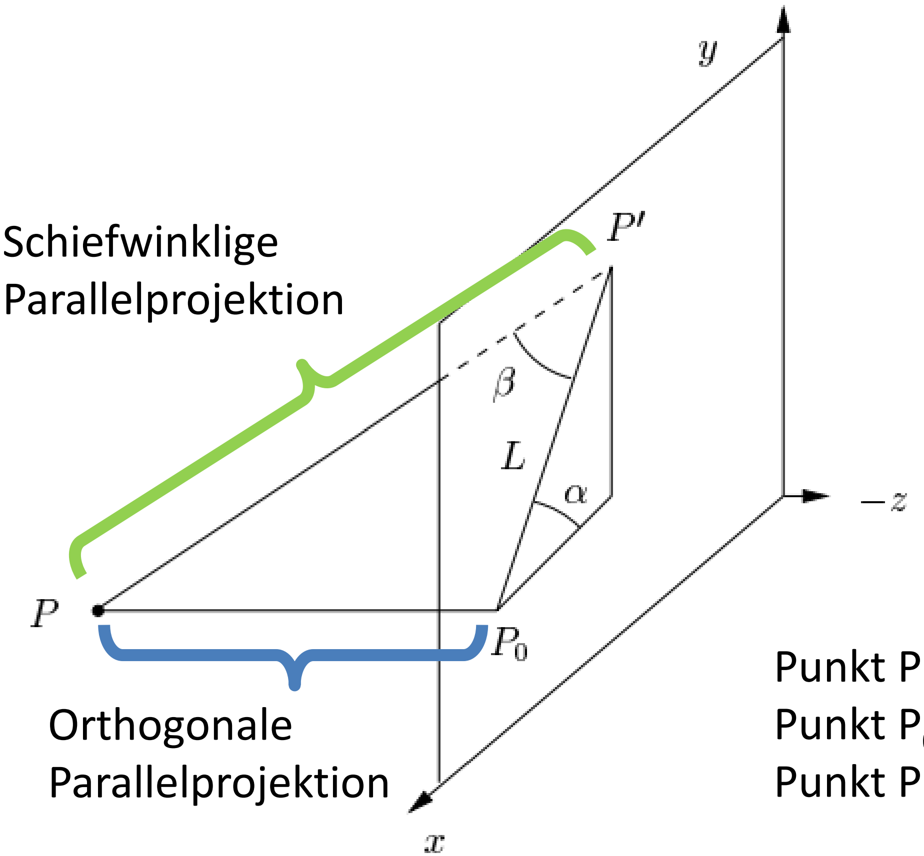
Welche Werte haben a und b?

Gesucht:

- Scherwinkel α
- Verkürzungsfaktor d

Schiefwinklige Projektionen

Abbildung: http://www-lehre.informatik.uni-osnabrueck.de/~cg/2000/Postscript/skript13-16.pdf



Punkt P = (x,y,z)
Punkt P_0 = (x,y,0)
Punkt P‘ = (x‘,y‘,0)

Scherwinkel = α, Winkel zur Horizontalen, unter dem projizierte Kanten aufgetragen werden.

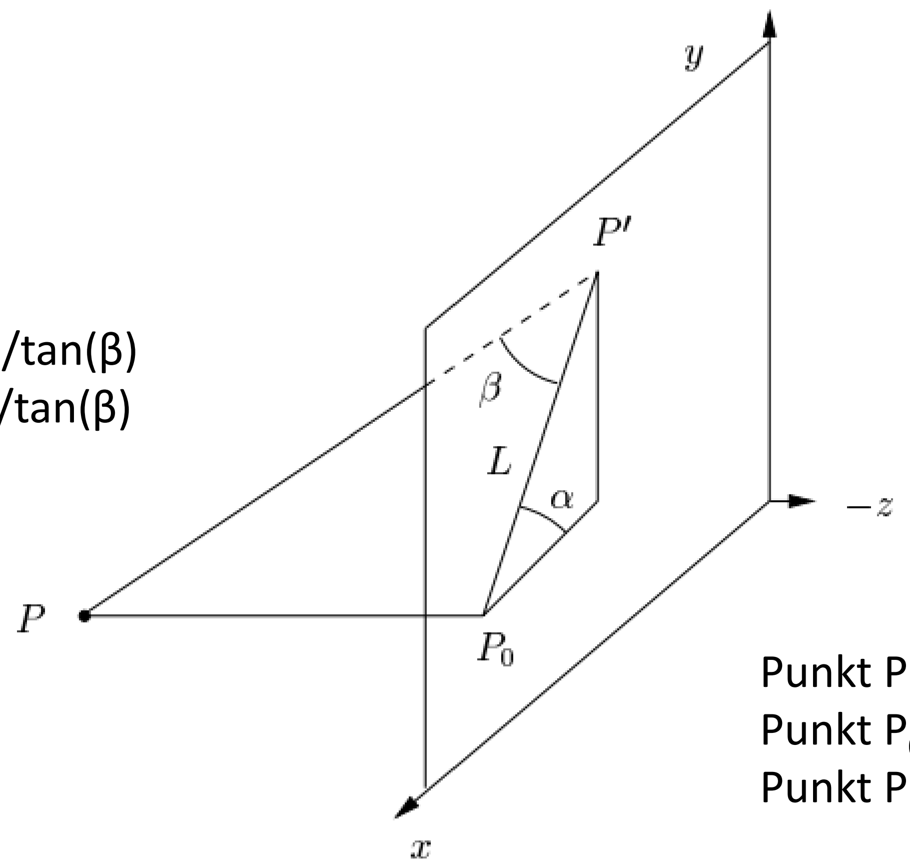
Schiefwinklige Projektionen

Trigonometrie:
x‘ = x – L * cos(α)
y‘ = y + L * sin(α)

tan(β) = z/L

x‘ = x – z * cos(α)/tan(β)
y‘ = y + z * sin(α)/tan(β)

Abbildung: http://www-lehre.informatik.uni-osnabrueck.de/~cg/2000/Postscript/skript13-16.pdf



Punkt P = (x,y,z)
Punkt P_0 = (x,y,0)
Punkt P‘ = (x‘,y‘,0)

Schiefwinklige Projektionen

Abstände für projizierte Punkte P1‘ und P2‘, die sich nur in ihren z-Koordinaten unterscheiden:

|x1‘ – x2‘| = |(z1 – z2) * cos(α) / tan (β)|
|y1‘ – y2‘| = |(z1 – z2) * sin(α) / tan (β)|

|P1‘ – P2‘| = $\sqrt{|x1' - x2'|2 + |y1' - y2'|2}$

= $\sqrt{\frac{(z1-z2)^2}{tan^2(\beta)} * (cos^2(\alpha) + sin^2(\alpha))}$

= $\frac{z1-z2}{tan(\beta)}$ wegen $\cos^2(\alpha) + \sin^2(\alpha) = 1$.

Es folgt:

Verkürzungsfaktor d = $\frac{1}{tan(\beta)}$

Resultierende Transformationsmatrix:

$$\begin{pmatrix} 1 & 0 & -d * \cos(\alpha) & 0 \\ 0 & 1 & d * \sin(\alpha) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aufgabe 1 ii

- **Kavalierperspektive:**
  - Projektionsebene parallel zu einer von zwei Koordinatenachsen aufgespannten Ebene
  - Projektion der dritten Koordinatenachse verläuft unter 45°, d.h. **d = 1 / tan(45) = 1**
  - Somit unverkürzte Abbildung von auf der Bildebene normal stehenden Strecken

$$\begin{pmatrix} 1 & 0 & -\cos(\alpha) & 0 \\ 0 & 1 & \sin(\alpha) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aufgabe 1 ii

- **Kabinettperspektive:**
  - Projektionsebene parallel zu einer von zwei Koordinatenachsen aufgespannten Ebene
  - Projektion der dritten Koordinatenachse verläuft unter 63.43° , d.h. **d = 1/tan(63.43)** $\approx$ **0.5**
  - Auf der Bildebene normal stehende Strecken werden um die Hälfte verkürzt abgebildet

$$\begin{pmatrix} 1 & 0 & -0.5 * \cos(\alpha) & 0 \\ 0 & 1 & 0.5 * \sin(\alpha) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aufgabe 1 iii

Matrizen als Arrays anlegen:

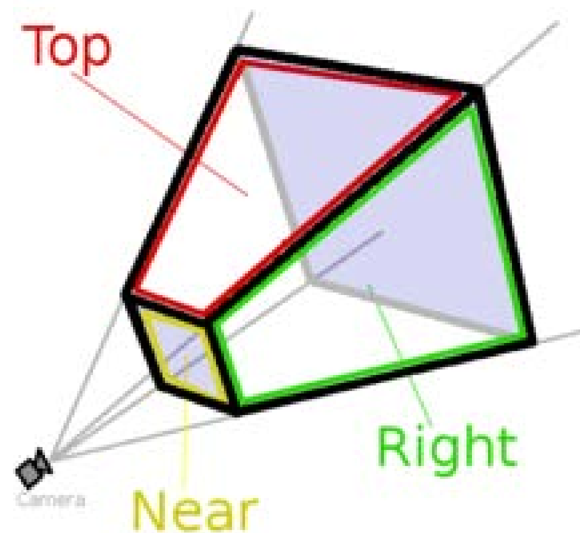
```
private double[] cavalier = {1,0,0,0,0,1,0,0,-Math.cos(45),Math.sin(45),0,0,0,0,0,1};
private double[] cabinet = {1,0,0,0,0,1,0,0,-0.5*Math.cos(50),0.5*Math.sin(50),0,0,0,0,0,1};
```

Projektionsmatrix entsprechend anpassen:

```
gl.glMatrixMode(GL2.GL_PROJECTION); //select the projection matrix
gl.glLoadIdentity(); //reset the projection matrix
if(cabinetPerspective)
        gl.glMultMatrixd(DoubleBuffer.wrap(cabinet));
if(cavalierPerspective)
        gl.glMultMatrixd(DoubleBuffer.wrap(cavalier));
```

Aufgabe 2-i

```
glFrustum(
        Left,
        Right,
        Bottom,
        Top,
        Near,
        Far
)
```



Wikipedia

Aufgabe 2-i

gluPerspective(

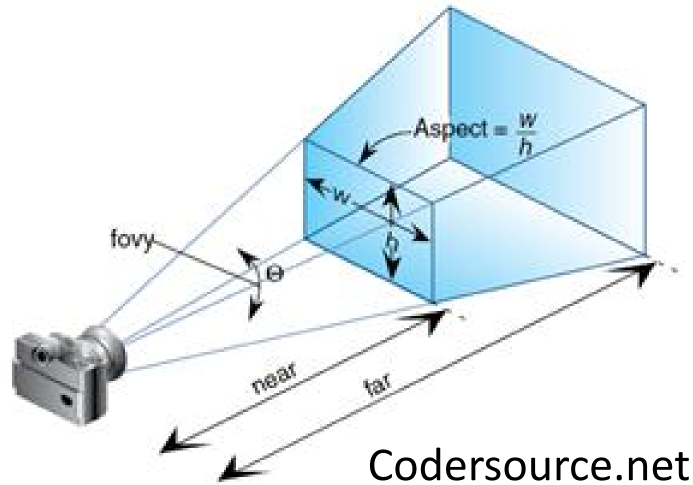
  Fovy,

  Aspect,

  Near,

  Far

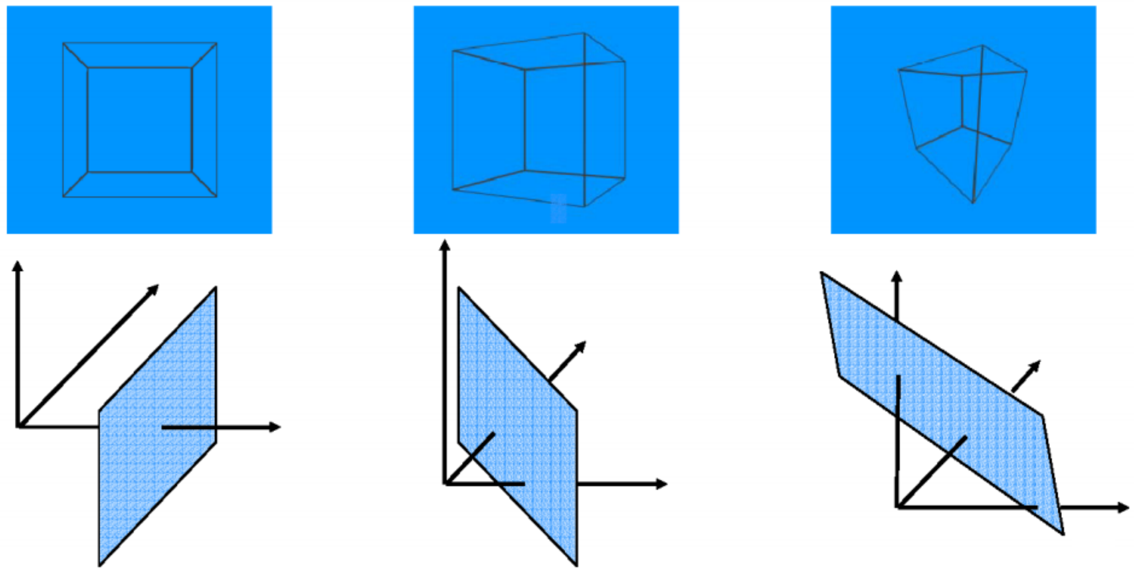
)



Codersource.net

Aufgabe 2-ii

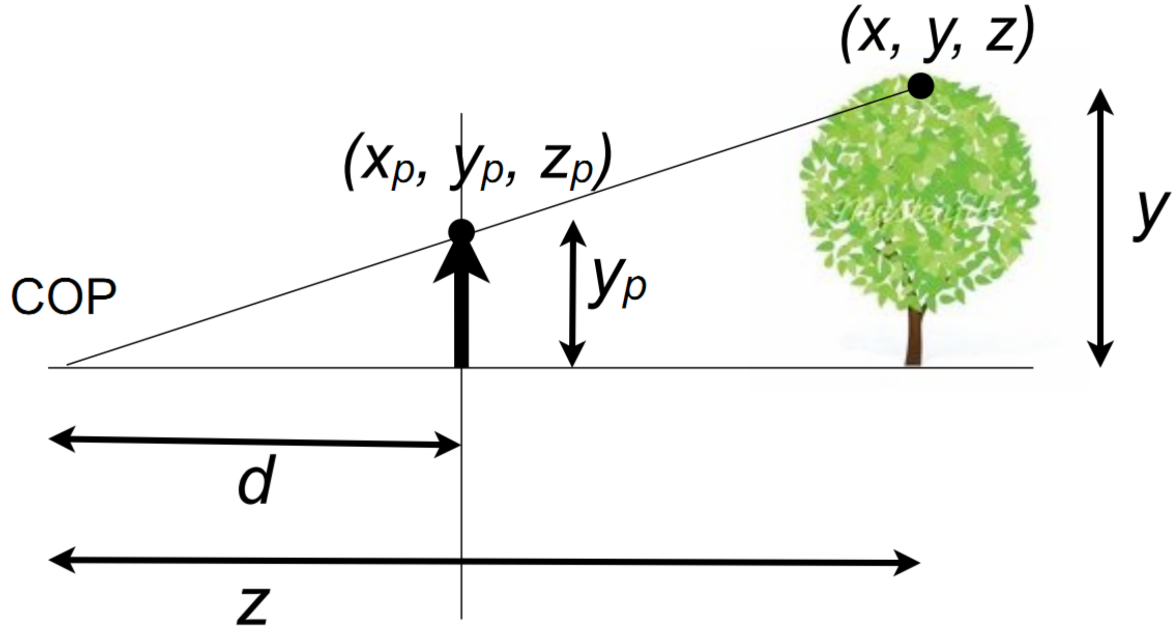
Ausschlaggebend: Schnittpunkte von Projektionsebene und Koordinatenachsen



Quelle: M . Haller, FH Hagenberg, (2002)

Aufgabe 2-iii

Recap:



Aus dem Strahlensatz folgt: $y/z = y_p/d$ und somit $y_p = y/(z/d)$

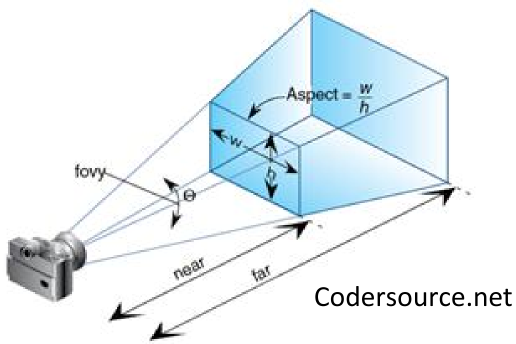
Aufgabe 2-iii

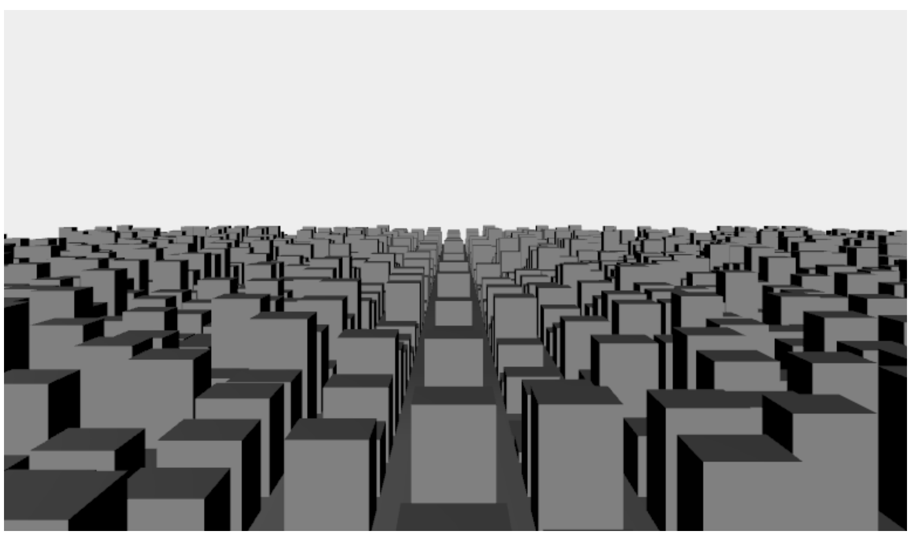
d = -1, da Projektionsebene parallel zur XY-Ebene bei z = -1

Punkt in Weltkoordinaten	Verkürzungsfaktor t = z/d	Projektion des Punktes
A (2,1,-1)	-1/-1 = **1**	A_p (2,1)
B (2,1,-10)	-10/-1 = **10**	B_p (0.2,0.1)
C (2,1,-100)	-100/-1 = **100**	C_p (0.02, 0.01)
D (2, -5, -100)	-100/-1 = **100**	D_p (0.02, -0.05)
E (-50, -50, -100)	-100/-1 = **100**	E_p (-0.5, -0.5)

Fluchtpunkt: lim(z-> -inf) (z/d) = inf
Daher: x_p = x/t = x/inf = 0, y_p = 0 –> **Fluchtpunkt ist (0,0)**

Aufgabe 3 ii



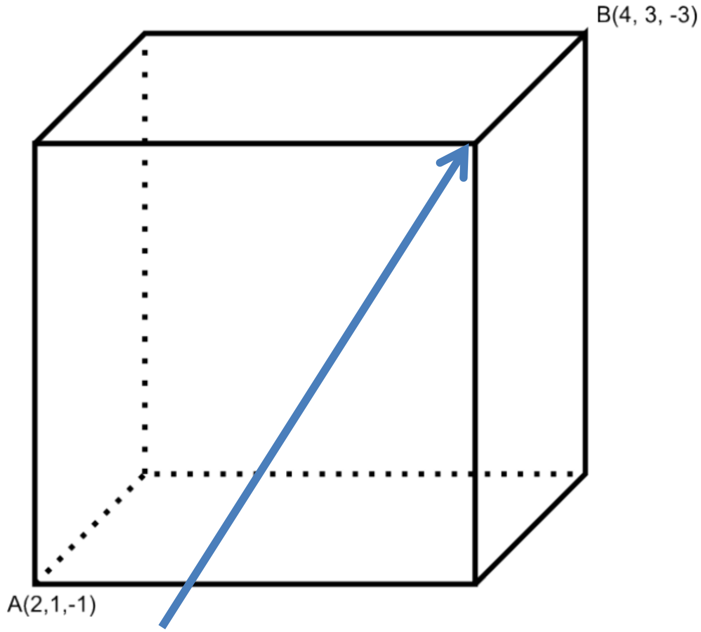


Frustum Culling:
Objekte, die nicht innerhalb des Sichtvolumens sind, werden nicht gerendert.

Occlusion Culling:
Objekte, die von anderen Objekten verdeckt sind, werden nicht gerendert.

Backface Culling:
Polygone auf der Rückseite von Objekten müssen nicht gezeichnet werden.

Aufgabe 3 iii



Standpunkt v = (3,2,2)

Wir könnten z.B. hierher blicken: e = (4,3,-1)
$\overrightarrow{ve}$ = (1,1,-1)

Normalenvektoren des Würfels:
Vorderseite: $\overrightarrow{vf}$ = (0,0,1)
Rückseite: $\overrightarrow{vb}$ = (0,0,-1)
Links: $\overrightarrow{vl}$ = (-1,0,0)
Rechts: $\overrightarrow{vr}$ = (1,0,0)
Oben: $\overrightarrow{vu}$ = (0,1,0)
Unten: $\overrightarrow{vd}$ = (0,-1,0)

Aufgabe 3iii

Wiederholung:

Winkel zwischen zwei Vektoren:

cos ϒ = $\frac{\vec{a} * \vec{b}}{\left|\vec{a}\right| * \left|\vec{b}\right|}$

Winkel zwischen $\overrightarrow{ve}$ und

$\overrightarrow{vf} = 131{,}1°$ -> sichtbar
$\overrightarrow{vb} = 57{,}3°$ -> nicht sichtbar
$\overrightarrow{vl} = 131{,}1°$-> sichtbar
$\overrightarrow{vr} = 57{,}3°$ -> nicht sichtbar
$\overrightarrow{vu} = 57{,}3°$ -> nicht sichtbar
$\overrightarrow{vd} = 131{,}1°$ -> sichtbar