Computergrafik

Mitschrift von

Markus Vieth   David Klopp

20. Dezember 2016

Vorwort

Dieses Skript basiert auf unserer Mitschrift der Vorlesung Computergrafik und VR im WS 2016/17 an der JGU Mainz (Dozent: Prof. Dr. E. Schömer).
Es handelt sich nicht um eine offizielle Veröffentlichung der Universität.
Wir übernehmen keine Gewähr für die Fehlerfreiheit und Vollständigkeit des Skripts.
Fehler können unter Github gemeldet werden. Die aktuelle Version dieses Skriptes ist ebenfalls auf Github zu finden.

Inhaltsverzeichnis

1 VBO

	Kartesische Koord.	Farben	Textur-Koord.	Normal
v_0	x_0, y_0, z_0, w_0	r_0, g_0, b_0	s_0, t_0	u_0, v_0, w'_0
v_1				
$\vdots$				
v_{n-1}				

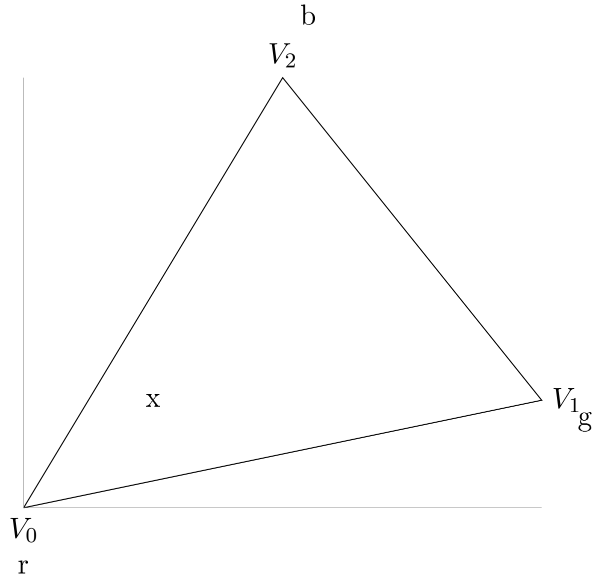


Abbildung 1.1: Beispiel Raster?

1.1 Baryzentrische Koordinaten

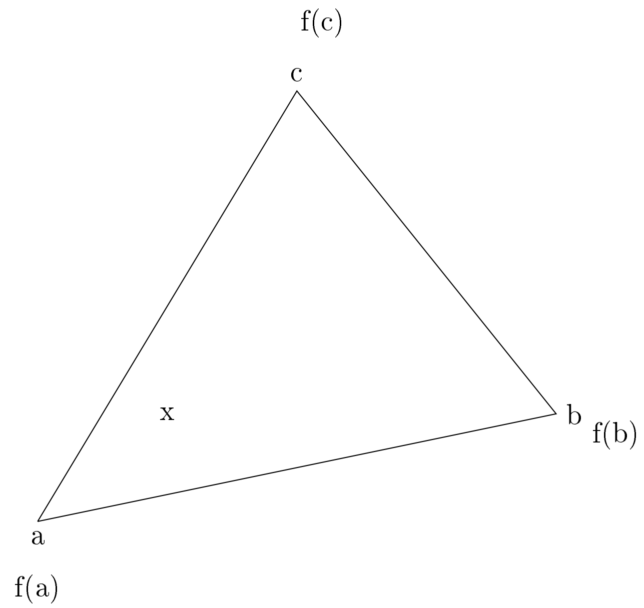


Abbildung 1.2: Baryzentrisches Koordinatensystem

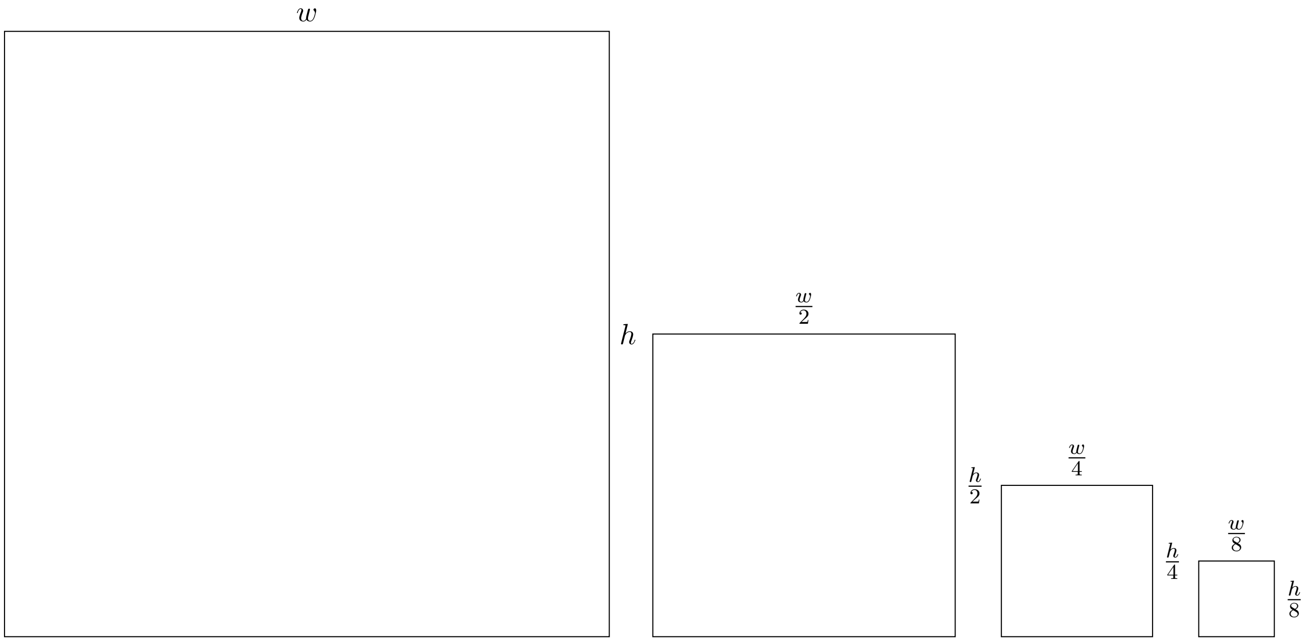
$$\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

$$x = \alpha \cdot a + \beta \cdot b + \gamma \cdot c \wedge \alpha + \beta + \gamma = 1$$

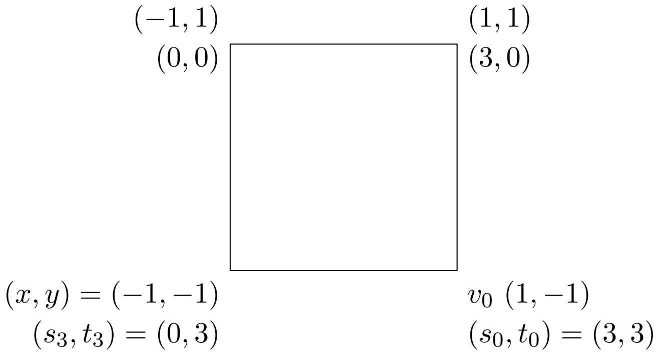
$$\Rightarrow f(x) = \alpha \cdot f(a) + \beta \cdot f(b) + \gamma \cdot f(c)$$

1.2 Texturen

1.2.1 Mipmap

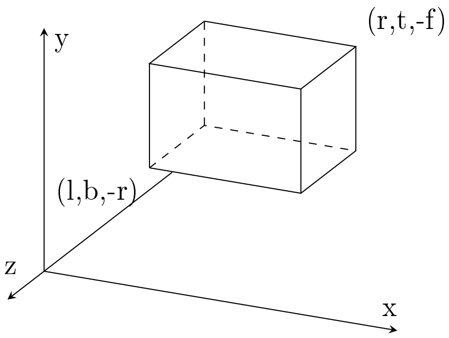


$$S = \sum_{i=0}^{\infty} (\frac{1}{4})^i = \frac{1}{1 - \frac{1}{4}} = \frac{4}{3}$$


$(-1, 1)$
$(0, 0)$
$(1, 1)$
$(3, 0)$
$(x, y) = (-1, -1)$
$(s_3, t_3) = (0, 3)$
$v_0\ (1, -1)$
$(s_0, t_0) = (3, 3)$

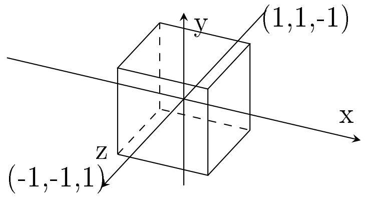
2 3D-Objekte

2.1 Orthogonalprojektion



$$x \in [l, r]$$
$$y \in [b, t]$$
$$z \in [-f, -n]$$

Sichtquader $\rightarrow$ Einheitsquader



$$x' \in [-1, 1]$$
$$y' \in [-1, 1]$$
$$z' \in [-1, 1]$$

$$x' = a\alpha \cdot x + \beta$$
$$l \mapsto -1, \; r \mapsto 1$$

$$\begin{array}{rr}
(1) & -1 = \alpha \cdot l + \beta \\
(2) & 1 = \alpha \cdot r + \beta \\
\hline
(2) - (1) & 2 = \alpha \cdot r - \alpha \cdot l \Rightarrow \alpha = \dfrac{2}{r - l} \\
 & 1 = \dfrac{2 \cdot r}{r - l} + \beta \\
 & \beta = 1 - \dfrac{2r}{r - l} = \dfrac{r - l - 2r}{r - e} = -\dfrac{r + l}{r - l}
\end{array}$$

$$\underbrace{\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}}_{\text{NDS}} = \underbrace{\begin{pmatrix} \frac{2}{r-l} & 0 & 0 & \frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & \frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{Q} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

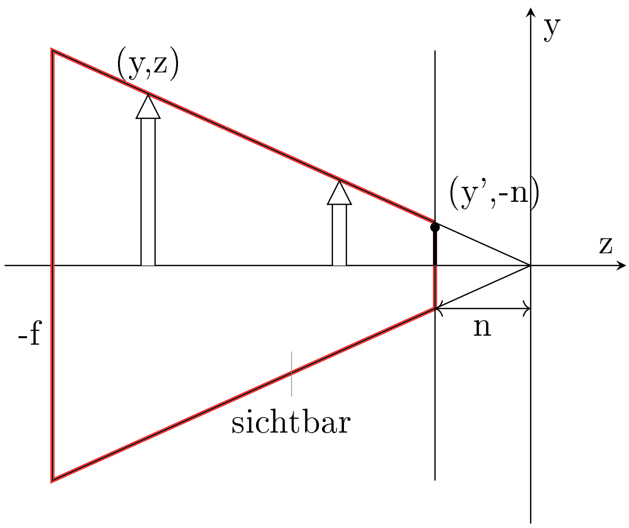
$$z' = -\frac{2}{f-n}z - \frac{f+n}{f-n}$$

$$z = n \quad z* = \frac{2n-(f+n)}{f-n} = \frac{n-f}{f-n} = -1$$

$$-n \mapsto -1, \quad -f \mapsto 1$$

```
Qmatrix4x4.ortho(l,n,b,t,n,f);
```

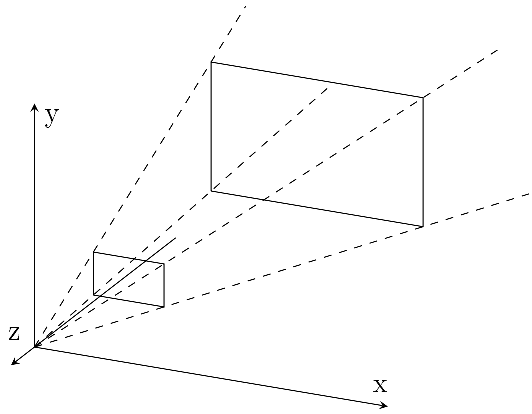
2.2 Perspektivische Projektion



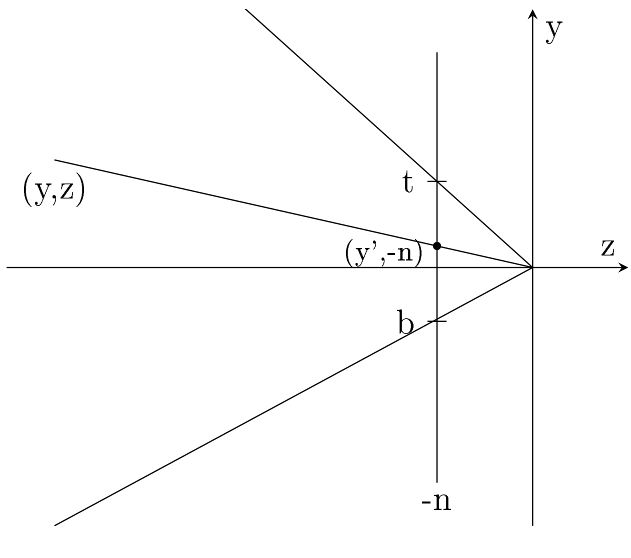
$$\frac{y'}{-n} = \frac{y}{z}$$

$$y' = -\frac{n \cdot y}{z}$$

Sichtpyramide → Einheitswürfel



$$\begin{aligned} y' &= -\frac{n \cdot y}{z} \\ &\Cap \\ [b,t] &\mapsto [-1,1] \\ y'' &= \alpha \cdot y' + \beta \\ y'' &= \frac{2}{t \cdot b} \cdot y' - \frac{t+b}{t-b} \\ &\boxed{y'' = \frac{-2n}{t-b} \cdot \frac{y}{z} - \frac{t+b}{t-b}} \end{aligned}$$



$$\underset{\text{homogene Koord.}}{\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}} \xrightarrow[\text{isierung}]{\text{Dehomogen-}} \underset{\text{Kartesiche koord.}}{\begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \end{pmatrix}}$$

$$\begin{pmatrix} x'' \\ y'' \\ z'' \\ w'' \end{pmatrix} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$y'' = \frac{2n}{t-b} \cdot y + \frac{t+b}{t-b} \cdot z$$

$$w'' = -z$$

$$\frac{y''}{w''} = \frac{2n}{t-b} \frac{y}{(-z)} + \frac{t+b}{t-b} \frac{z}{(-z)}$$

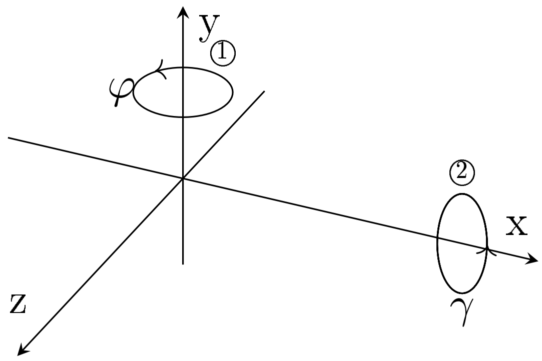
$$z''' = \frac{z''}{w''} = \frac{\alpha \cdot z + \beta}{-z} = -\alpha - \frac{\beta}{z}$$

$$-n \mapsto -1, \quad -f \mapsto 15$$

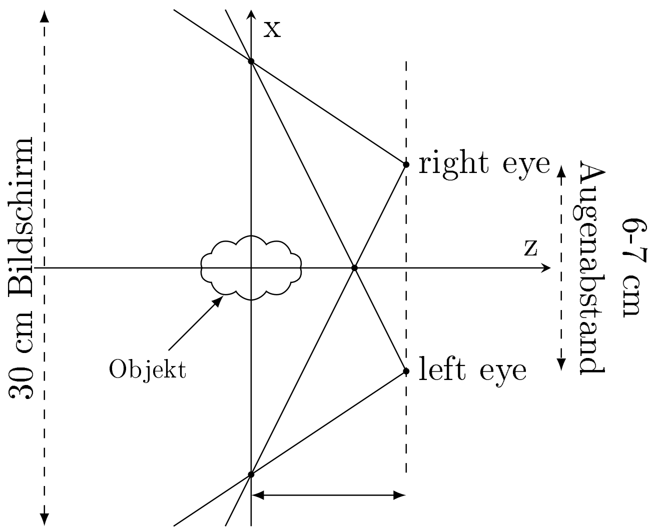
$$\begin{aligned} -\alpha - \frac{\beta}{-n} &= -1 \\ -\alpha - \frac{\beta}{-f} &= 1 \\ -\alpha + \frac{\beta}{n} &= -1(1) \\ -\alpha + \frac{\beta}{f} &= 1(2) \end{aligned}$$

$$\begin{aligned} \frac{\beta}{f} - \frac{\beta}{n} &= 2(2) - (1) \\ \beta \left(\frac{1}{f} - \frac{1}{n} \right) &= 2 \\ \beta \left(\frac{n-f}{fn} \right) & \\ \beta &= \frac{-2nf}{f-n} \\ \alpha = \frac{\beta}{f} - 1 = -\frac{2n-(f-n)}{f-n} &= \frac{f+n}{f-n} \end{aligned}$$

$$p' = \underset{\substack{② \\ \uparrow \\ \text{Drehung um} \\ \text{die Welt-x-Achse}}}{R_{\vartheta,x}} \cdot \underset{\substack{① \\ \uparrow \\ \text{Drehung um} \\ \text{die Welt-y-Achse}}}{R_{\varphi,y}} \cdot p$$

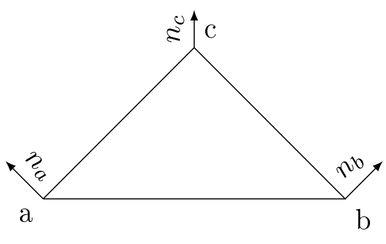

y
①
φ
②
x
z
γ

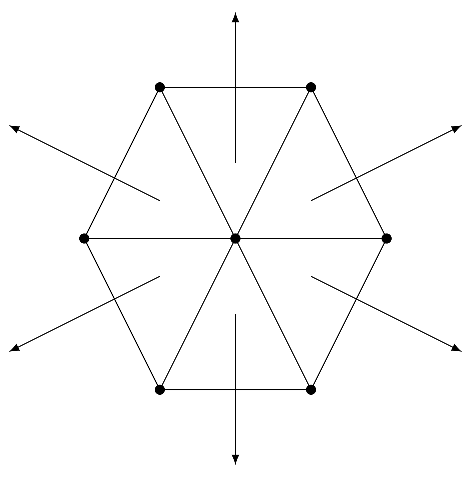
3D-Brille


x
30 cm Bildschirm
right eye
6-7 cm
Augenabstand
z
Objekt
left eye

3 Beleuchtung

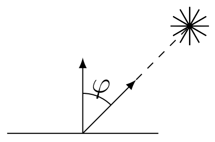
3.0.1 Smoothing



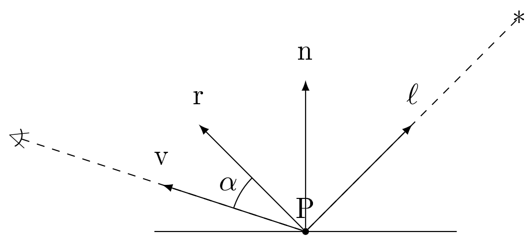


Lambert

$$\mathrm{I}_D = \mathrm{I}_L \cdot \left(n^T \cdot \ell\right)$$



3.1 Phong Lichtmodell

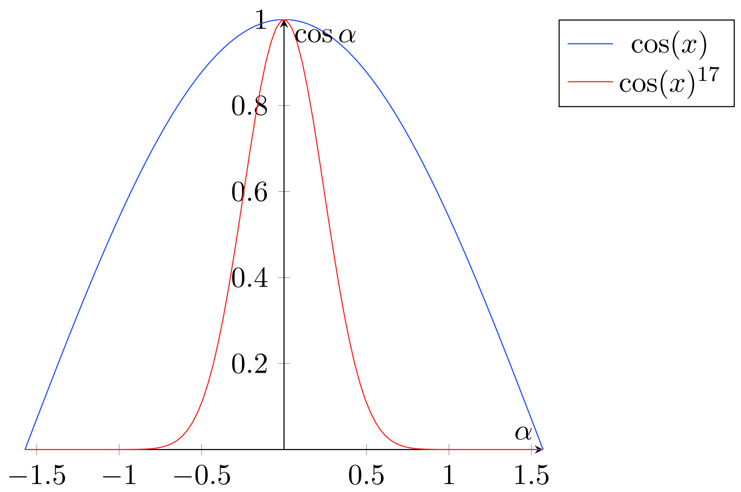


$$|n| = |\ell| = |r| = |v| = 1$$

$$r = 2n(n^T\ell) - \ell$$

S = Shininess

$$I_S = I_L(\cos\alpha)^S = I_L\left(r^T v\right)^S, \quad I_D = I_L\left(n^T \ell\right)$$



3.1.1 Phong

$$I_{\text{Color}} = I_{\text{Ambient,Color}} + I_{\text{Diffuse,Color}} + I_{\text{Specular, Color}}$$

$$\text{Color} \in \{\text{Red}, \text{Green}, \text{Blue}\}$$

```
void main() {
    vec3 normal = normalize(vNormal);
    vec3 lightDir = normalize(lighPos - vPos);
    vec3 reflectDie = reflect(lightDir, normal);
    vec3 viewDir = normalize(-vPos);

    float lambertian = max(dot(loghtDir, normal), 0.0.);
    float specular = 0.0;

    if ( lambertian > 0.0) {
        float specAngle = max(dot(reflectDir, viewDir), 0.0);
        specular = pow(specAngle, uShininess);
    }
    gl_FragColor = vec4(uAmbient + lambertian * uDiffuse + specular * uSpecular, 1.0);
}
```

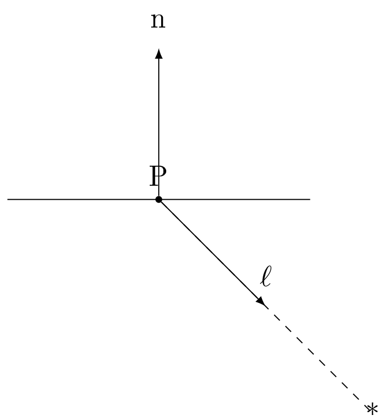
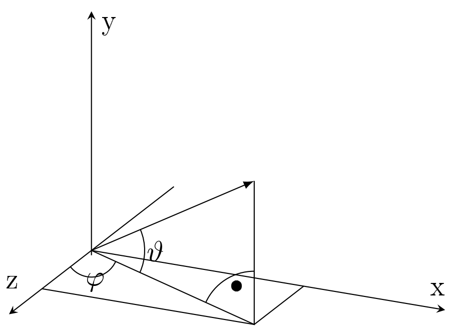


Abbildung 3.2: Zu ignorierende Lichtquelle

4 Oberflächen

4.1 Texturen



$$(\varphi, \vartheta) \mapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\vartheta \cdot \sin\varphi \\ \sin\vartheta \\ \cos\vartheta \cdot \cos\varphi \end{pmatrix}$$

$$0 \leq \varphi \leq 2\pi$$

$$-\frac{\pi}{2} \leq \vartheta \leq \frac{\pi}{2}$$

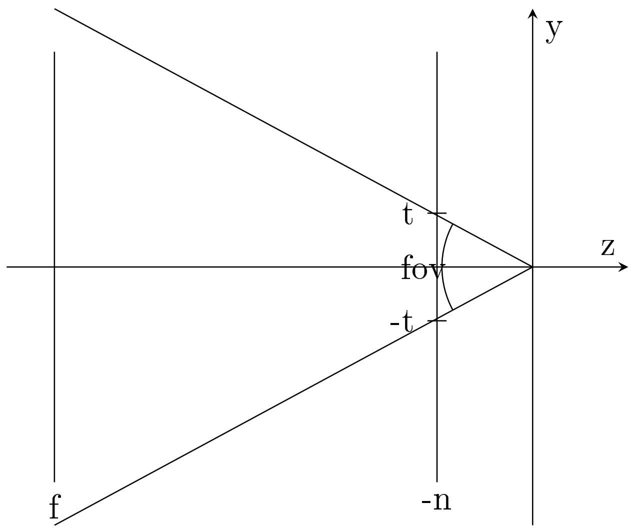
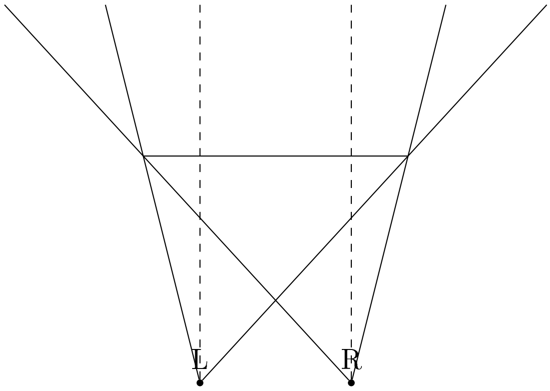
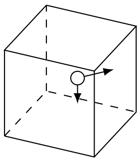


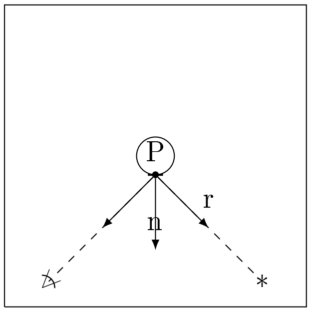
Abbildung 4.1: Field of view

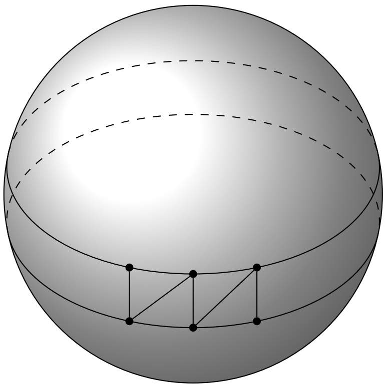
```
perspective(fov, aspectratio, n, f);
```



4.2 Cube-Mapping

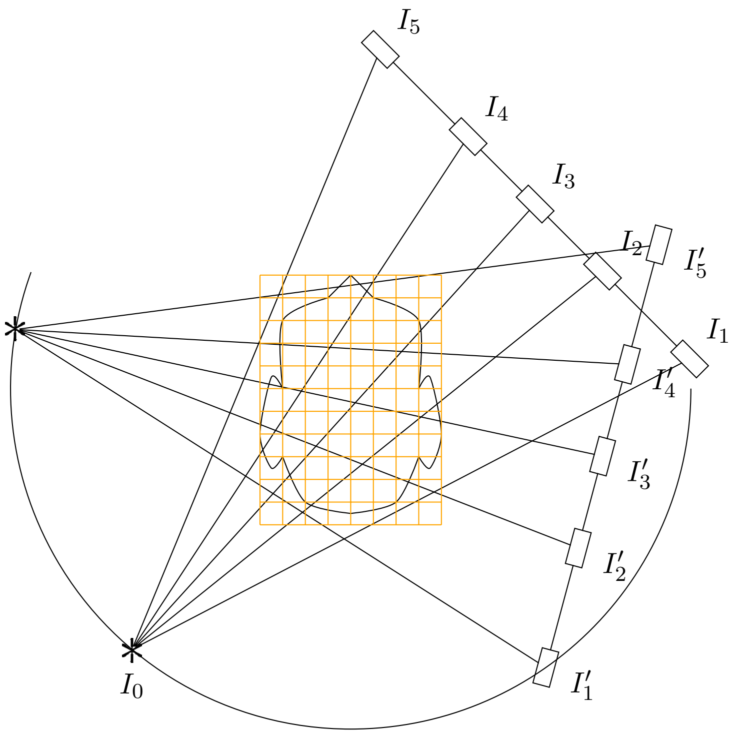


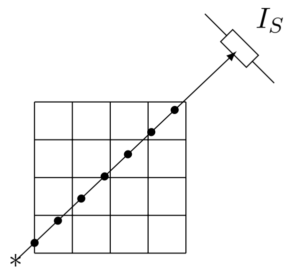




5 Volume Rendering mit 3d-Texturen

5.1 DVT





5.1.1 Lambert-Beer-Gesetz

$$I^{\text{out}} = I^{\text{in}} \cdot e^{-\mu \cdot d}$$

Schwächungskoeffizienten

$$I_S = I_0 \cdot e^{\sum_{\square(i,j)\cap S\neq\emptyset} \mu_{ij}\cdot d_{ij}}$$

$$\ln \frac{I_0}{I_S} = \sum_{\square(i,j)\cap S\neq\emptyset}^{\mu_{ij}\cdot d_{ij}}$$

$$I = \underset{\mathbb{R}^{360.000.000\times 128.000.000}}{\underset{\Cap}{D}} \cdot \mu$$

Inverses Problem

Gegeben: Gemessene Intensitäten I_S für alle Strahlen, die die Bildebene treffen für hinreichend viele Aufnahmerichtungen.

Gesucht: Schwächungskoeffizienten für alle Voxel des zu rekonstruierenden Volumens.

$$c^{\text{out}} = c^{\text{in}} \cdot (1 - \alpha_i) + c_i \alpha_i$$

$\boldsymbol{\alpha_i} =$ Deckkraft der Farbe c_i

$\mathbf{1} - \boldsymbol{\alpha_i} \hat{=}$ Transparenz

$$c^{\text{out}} = \cdot \sum_{i=1}^{n} c_i \alpha_i \prod_{j=i+1}^{n} (1 - \alpha_j)$$