

3D-Punkt-Transformationen und ihre Inversen:

- Translation:
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Skalierung:
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Scherung:
(entlang x)
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -q & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & q & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

evtl. beteiligte
Elemente
nach
Verkettung
mehrerer
eindimens.
Scherungen

Reihenfolge der Transformationen entscheidend – z.B.:

- Skalierung mit anschließender Translation:

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & t_x \\ 0 & s_y & 0 & t_y \\ 0 & 0 & s_z & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Translation mit anschließender Skalierung:

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & s_x t_x \\ 0 & s_y & 0 & s_y t_y \\ 0 & 0 & s_z & s_z t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

[s.o.: Das Matrizenprodukt ist nicht kommutativ!]

3D-Punkt-Transformationen

- Verwendung eines Rechts-Systems (in der CG wählbar):
Rotation positiv gegen d. Uhrzeigersinn bei Blickrichtung vom positiven Teil der Rotationsachse zum Koordinaten-Ursprung

- Herleitung der Rotationsmatrizen (s.o.):

$$A' = A \cdot \cos\theta - B \cdot \sin\theta$$

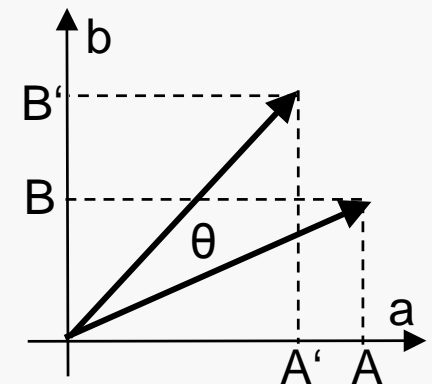
$$B' = A \cdot \sin\theta + B \cdot \cos\theta$$

- Rotation um die x-Achse:
($a \Rightarrow y$; $b \Rightarrow z$)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Rotation um die y-Achse:
($a \Rightarrow z$; $b \Rightarrow x$)

$$\begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



„Rechte-
Hand-Regel“



- Rotation um die z-Achse (vgl. 2D):
($a \Rightarrow x$ bzw. $b \Rightarrow y$)

$$\begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Beobachtungen an den Rotationsmatrizen:

- Die letzte Spalte ist gleich jener einer Einheitsmatrix (Platz für Translationswerte).
- Zeile und Spalte, die der jeweiligen Drehachse entsprechen, sind identisch mit der Zeile u. der Spalte einer Einheitsmatrix (d.h.: keine Veränderung für diese Koordinate).
- Besonders interessanter Spezialfall:
Rotation um Achse durch den Koordinaten-Ursprung und einen beliebigen Punkt:

Herleitung:

Rotation um Winkel θ um eine beliebige Raumachse durch den Koordinaten-Ursprung $[0,0,0,1]^T$ u. d. Punkt $[e_x, e_y, e_z, 1]^T$, $e_x^2 + e_y^2 + e_z^2 = 1$

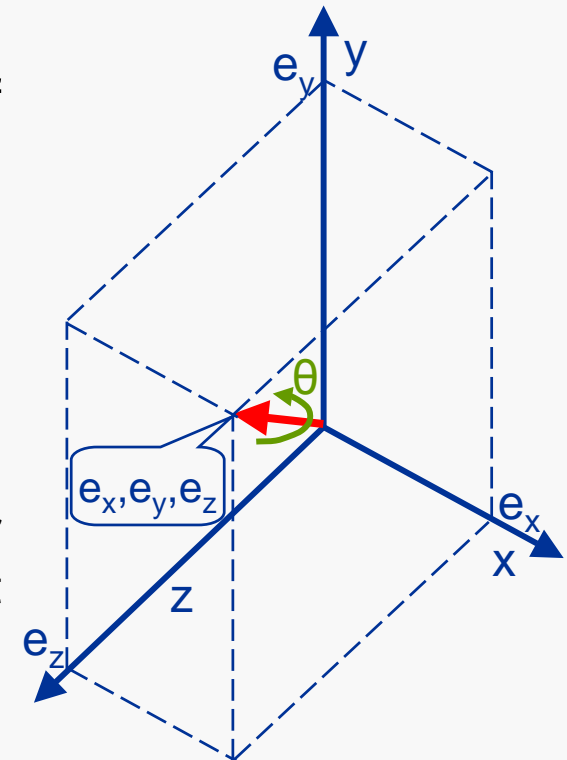
Vorgehen in 3 Schritten:

1. Transformation der gewünschten Achse auf eine der Hauptachsen
2. Rotation um die so transformierte Achse
3. Rücktransformation der Achse an ihren ursprünglichen Ort

[Herleitung hier:

- Rotation der Achse um die z-Achse bis zur y-z-Ebene, dann um die x-Achse, bis sie mit der z-Achse zusammenfällt,
- Rotation um θ um die z-Achse,
- Rücktransformation;

Ergebnis bleibt von dieser Wahl unberührt.]



- Rotation der Drehachse um den Winkel α ($\alpha \geq 0$) um die z-Achse bis zur y-z-Ebene:

$$\sin \alpha = e_x / (e_x^2 + e_y^2)^{1/2}$$

$$\cos \alpha = e_y / (e_x^2 + e_y^2)^{1/2}$$

- Rotation der Drehachse um den Winkel β ($\beta \geq 0$) um die x-Achse bis zur z-Achse:

$$\sin \beta = (e_x^2 + e_y^2)^{1/2} / (e_x^2 + e_y^2 + e_z^2)^{1/2}$$

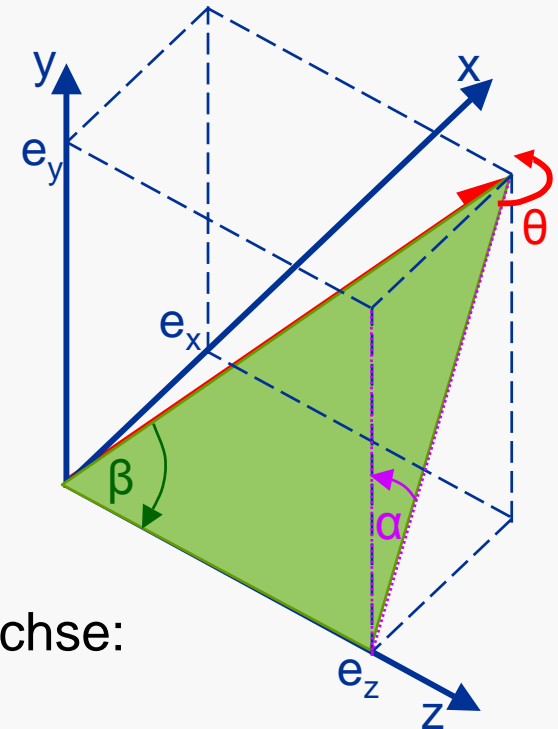
$$= (e_x^2 + e_y^2)^{1/2}$$

$$\cos \beta = e_z / (e_x^2 + e_y^2 + e_z^2)^{1/2}$$

$$= e_z$$

- Rotationsmatrix zur Drehung um die neue z-Achse:

$$\begin{aligned} \underline{R}(\theta) &= \underline{R}_z(-\alpha) \cdot \underline{R}_x(-\beta) \cdot \underline{R}_z(\theta) \cdot \underline{R}_x(\beta) \cdot \underline{R}_z(\alpha) \\ &= \underline{R}_z(\alpha)^{-1} \cdot \underline{R}_x(\beta)^{-1} \cdot \underline{R}_z(\theta) \cdot \underline{R}_x(\beta) \cdot \underline{R}_z(\alpha) \\ &= \underline{R}_z(\alpha)^T \cdot \underline{R}_x(\beta)^T \cdot \underline{R}_z(\theta) \cdot \underline{R}_x(\beta) \cdot \underline{R}_z(\alpha) \\ &= [\underline{R}_x(\beta) \cdot \underline{R}_z(\alpha)]^T \cdot \underline{R}_z(\theta) \cdot [\underline{R}_x(\beta) \cdot \underline{R}_z(\alpha)] \end{aligned}$$



Anmerkungen:

- Die Berechnung von $\underline{R}(\theta) = [\underline{R}_x(\beta) \cdot \underline{R}_z(\alpha)]^T \cdot \underline{R}_z(\theta) \cdot [\underline{R}_x(\beta) \cdot \underline{R}_z(\alpha)]$ erfordert nur drei Matrizenprodukte und eine Transposition.
- Ausmultipliziert lautet die Transformationsmatrix für die Rotation um den Winkel θ um eine beliebige Raumachse durch den Koordinaten-Ursprung und den Punkt $[e_x, e_y, e_z, 1]^T$, mit $e_x^2 + e_y^2 + e_z^2 = 1$:

$$\begin{pmatrix} e_x^2 + (1 - e_x^2) \cdot \cos\theta & e_x \cdot e_y \cdot (1 - \cos\theta) - e_z \cdot \sin\theta & e_x \cdot e_z \cdot (1 - \cos\theta) + e_y \cdot \sin\theta & 0 \\ e_x \cdot e_y \cdot (1 - \cos\theta) + e_z \cdot \sin\theta & e_y^2 + (1 - e_y^2) \cdot \cos\theta & e_y \cdot e_z \cdot (1 - \cos\theta) - e_x \cdot \sin\theta & 0 \\ e_x \cdot e_z \cdot (1 - \cos\theta) - e_y \cdot \sin\theta & e_y \cdot e_z \cdot (1 - \cos\theta) + e_x \cdot \sin\theta & e_z^2 + (1 - e_z^2) \cdot \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Hat der Richtungsvektor $[X, Y, Z]^T$ der Raumachse einen Betrag $\neq 1$, so ist er zu normieren auf $[X, Y, Z]^T \cdot 1/(X^2 + Y^2 + Z^2)^{1/2} = [e_x, e_y, e_z]^T$.
- Wird Drehung um eine Raumachse benötigt, die nicht durch den Koordinaten-Ursprung führt (vgl. Flugsimulator), so muß sie zuvor an den Ursprung verschoben werden und nach der Rotation an den ursprünglichen Ort zurücktransformiert werden:

$$\underline{R}(\theta) = \underline{I}(t_x, t_y, t_z) \cdot [\underline{R}_x(\beta) \cdot \underline{R}_z(\alpha)]^T \cdot \underline{R}_z(\theta) \cdot \underline{R}_x(\beta) \cdot \underline{R}_z(\alpha) \cdot \underline{I}(-t_x, -t_y, -t_z)$$

- Überall bisher: Letzte Zeile wie jene einer Einheitsmatrix – ein Kennzeichen affiner Transformationen.

- Affine Transformationen (AT_n) sind Punkt-Transformationen, die Linearkombinationen der kartesischen Koordinaten von Punkten bilden; sie sind die am häufigsten angewandten Transformationen in der CG.
- Translation, Rotation, Skalierung und Scherung sind AT_n .
- AT_n erhalten affine (und konvexe) Punkt-Kombinationen.
- AT_n erhalten gerade Linien und Ebenen.
- AT_n erhalten Parallelität von Linien und Ebenen; Translation und Rotation erhalten darüber hinaus auch die Längen von Linien und die Winkel zwischen ihnen.
- AT_n erhalten Teilungsverhältnisse von Strecken. (Spezialfall: mittlerer Schnittpunkt von Quadrat-/ Würfel-Diagonalen bildet sich ab auf mittigen Schnittpunkt der Parallelogramm-/Parallelepipiped-Diagonalen)

- 2D-ATn multiplizieren den Flächeninhalt transformierter 2D-Objekte mit der Determinanten ihrer Matrix; 3D-ATn verändern entsprechend das Objekt-Volumen mit $|\det M|$
- Multiplikation einer bel. Anzahl der Matrizen mehrerer ATn ergibt die Matrix einer wirkungsgleichen AT.
- Die Transformation, die eine AT rückgängig macht, ist ebenfalls eine AT; ihre Matrix-Darstellung ist die inverse Matrix der ursprünglichen AT.
- Jede AT läßt sich aus einer (bel.) vorgegebenen Folge v. Elementar-ATn zusammensetzen –z.B.: (\Leftarrow Leserichtung!)
 $M = (\text{Scherung}) \cdot (\text{Skalierung}) \cdot (\text{Rotation}) \cdot (\text{Translation})$
- Die Spalten der AT-Matrix enthalten das transformierte CF (\equiv transformierte Einheitsmatrix!).
- ATn lassen sich immer in Matrixform darstellen; ihre letzte Zeile hat immer die Form: $[0 \dots 0 \ 1]$.

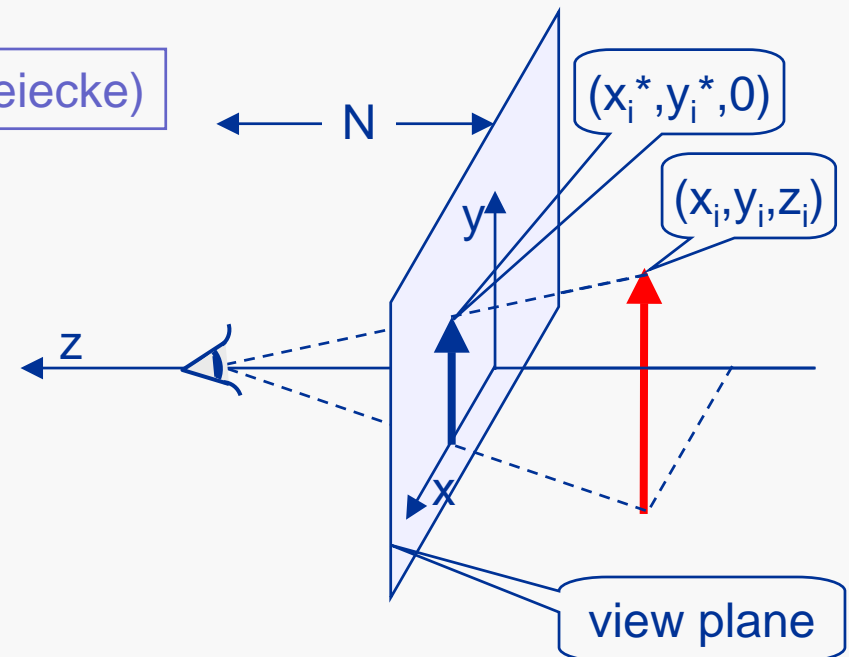
Transformationen, deren Matrix als letzte Zeile nicht die Form: $[0 \dots 0 \ 1]$ hat, gehören zur allgemeineren Klasse der **perspektivischen Transformationen**.

Perspektivische Projektion von Punkten (x_i, y_i, z_i) auf $(x_i^*, y_i^*, 0)$ in der Projektionsebene $z=0$ mit Proj.zentrum („Augenpunkt“) bei $z=N$ ($N>0$) in einem Rechts(koordinaten)system:

$$x_i^*/x_i = y_i^*/y_i = N/(N-z_i) \quad (\text{ähnliche Dreiecke})$$

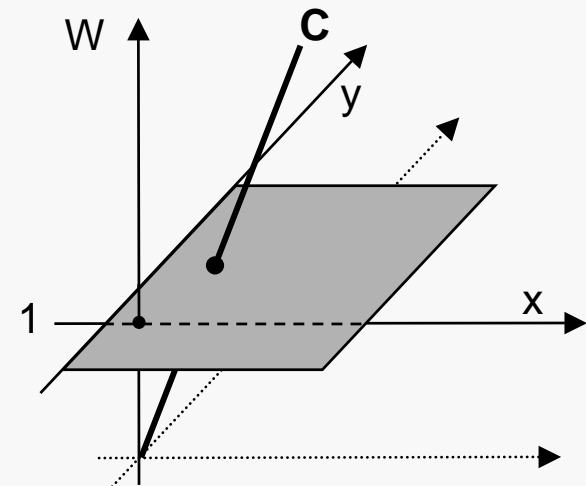
Versuch der Bildung eines Matrizenprodukts:

$$\begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & N \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} Nx_i \\ Ny_i \\ 0 \\ N-z_i \end{pmatrix} = \begin{pmatrix} (N-z_i) \cdot x_i^* \\ (N-z_i) \cdot y_i^* \\ (N-z_i) \cdot 0 \\ (N-z_i) \cdot 1 \end{pmatrix}$$



Konzept-Erweiterung: Homogene Koordinaten $[x_i, y_i, z_i, 1]^T$ als Darstellung v. „Punkt-Familien“ $[w \cdot x_i, w \cdot y_i, w \cdot z_i, w]^T$, $w \neq 0$:

- Wechsel von kartesischen zu homogenen Koordinaten durch Anhängen einer 1 (oder einer anderen Zahl $\neq 0$, mit der zuvor alle Punkt-Koordinaten multipliziert wurden).
- Wechsel von homogenen zu kartesischen Koordinaten durch Division durch die letzte Komponente („perspektivische Division“, auch: „Homogenisieren“, engl. *homogenize*) und Weglassen dieser letzten Komponente.



Geometrische Deutung 2D-Fall:
Punkt $C \in (x, y)$ wird in homog.
Koord. als Gerade $C \in (x_h, y_h, w_h)$
mit $x_h = w_h \cdot x$, $y_h = w_h \cdot y$ dargestellt.

Nebeneffekt der Homogenisierung: Gesamtskalierung

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/s \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1/s \end{pmatrix} \xrightarrow{\text{Homogenisierung}} \begin{pmatrix} sx_i \\ sy_i \\ sz_i \\ 1 \end{pmatrix}$$

– aufgrund d. Konvention zur Homogenisierung wirkungsgleich mit der Verwendung separater (aber einheitlicher) Skalierungsfaktoren

$$\begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} sx_i \\ sy_i \\ sz_i \\ 1 \end{pmatrix}$$

Begriffliche und rechnerische Trennung:

Die (hier: perspektivische) Projektion reduziert die Anzahl von Objekt-Dimensionen; die perspektivische Transformation verformt Objekte zur Vorbereitung einer Projektion (s.u.).

Proj. Trf.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & -1 & N \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} Nx_i \\ Ny_i \\ 0 \\ N-z_i \end{pmatrix} \xrightarrow{\text{perspekt. Division}} \begin{pmatrix} Nx_i/(N-z_i) \\ Ny_i/(N-z_i) \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x_i^* \\ y_i^* \\ 0 \\ 1 \end{pmatrix}$$

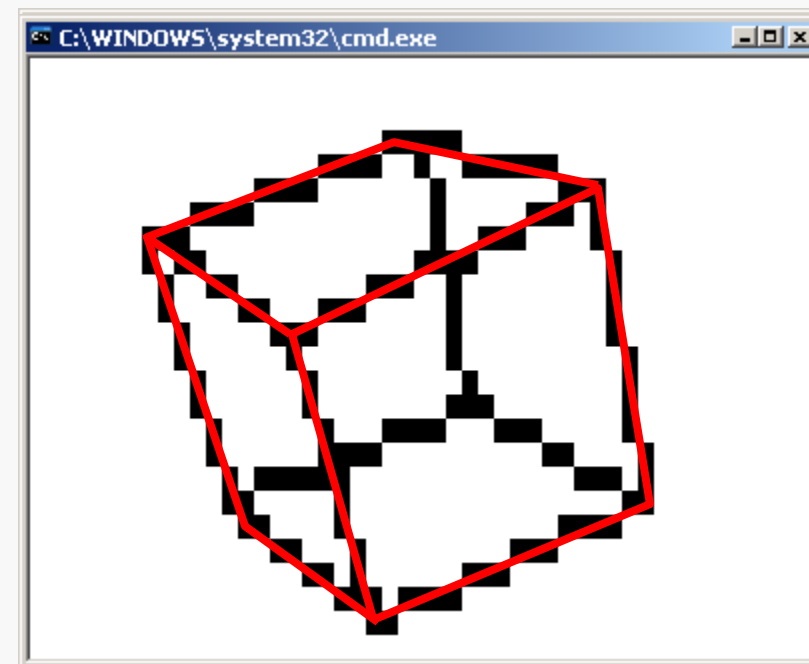
Namensgebung: Koordinaten „homogen“, denn sie ermöglichen auch Perspektive als Matrizen-Multiplikation i.d. Grafik-Pipeline.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/N & 1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ 0 \\ 1-z_i/N \end{pmatrix} \xrightarrow{\text{perspektivische Division}} \begin{pmatrix} Nx_i/(N-z_i) \\ Ny_i/(N-z_i) \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x_i^* \\ y_i^* \\ 0 \\ 1 \end{pmatrix}$$

Übung:

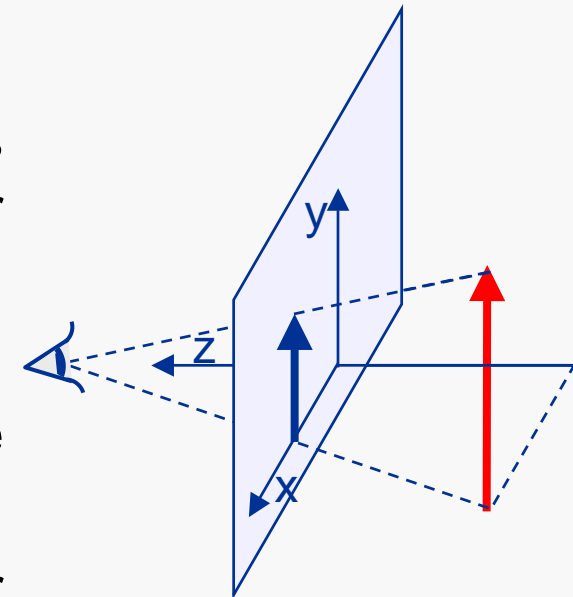
Im Konsole-Fenster sollen mit ASCII-Zeichen geladene 3D-Drahtmodelle (*wire frame*) dargestellt werden; sie sollen um die x-, y- und z-Achsen drehbar und bei veränderlichem Projektionszentrum darstellbar sein.

WireCullFill(1).exe



Anmerkungen zur Perspektive:

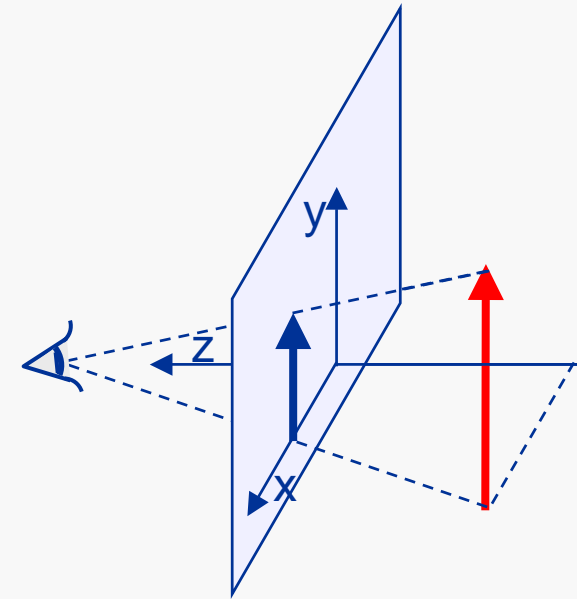
- Die perspektivische Division bewirkt, daß weiter entfernte Objekte (z_i groß) in der Projektion kleiner erscheinen.
- Verschiebung d. Projektionszentrums (N) verändert die Abbildungs-Unterschiede zwischen nah und fern.
- Verschiebung der Proj.ebene entlang der z-Achse (auf $z \neq 0$) verändert nur den Abb.-Maßstab. (Dreiecke bleiben ähnlich.)
- Auslassen der perspektiv. Transformation erzeugt Parallelprojektion (orthografische P., engl. *orthographic p.*); das entspricht einer Abb. mit Proj.zentr. im Unendlichen.



$$\begin{pmatrix} Nx_i/(N-z_i) \\ Ny_i/(N-z_i) \\ 1 \end{pmatrix} = \begin{pmatrix} x_i^* \\ y_i^* \\ 1 \end{pmatrix}$$

Weitere Anmerkungen zur Perspektive:

- Gerade Linien und ebene Flächen werden als solche abgebildet: Punkt-Kollinearität und -Komplanarität bleiben erhalten; Teilungsverhältnisse von Strecken und Flächen bleiben dagegen nicht erhalten.
- Parallele Linien, die auch parallel zur Projektionsebene liegen, werden als Parallelen abgebildet; sonst laufen sie in einem Punkt zusammen, dem jeweil. Fluchtpunkt (engl. *vanishing point*).
- Geraden, die das Projektionszentrum enthalten, werden auf Punkte projiziert.
- Ebenen, die d. Proj.zentrum enthalten, werden auf Geraden projiziert.



3D-Sicht, Projektionen

In Praxis und Literatur meist verbreitetes Paradigma:

Perspektivische Projektion v. Punkten (x_i, y_i, z_i) auf $(x_i^*, y_i^*, -N)$ i.d. Projektionsebene $z = -N$ ($N > 0$) mit Projektionszentrum am Koordinaten-Ursprung eines Rechts(koordinaten)systems:

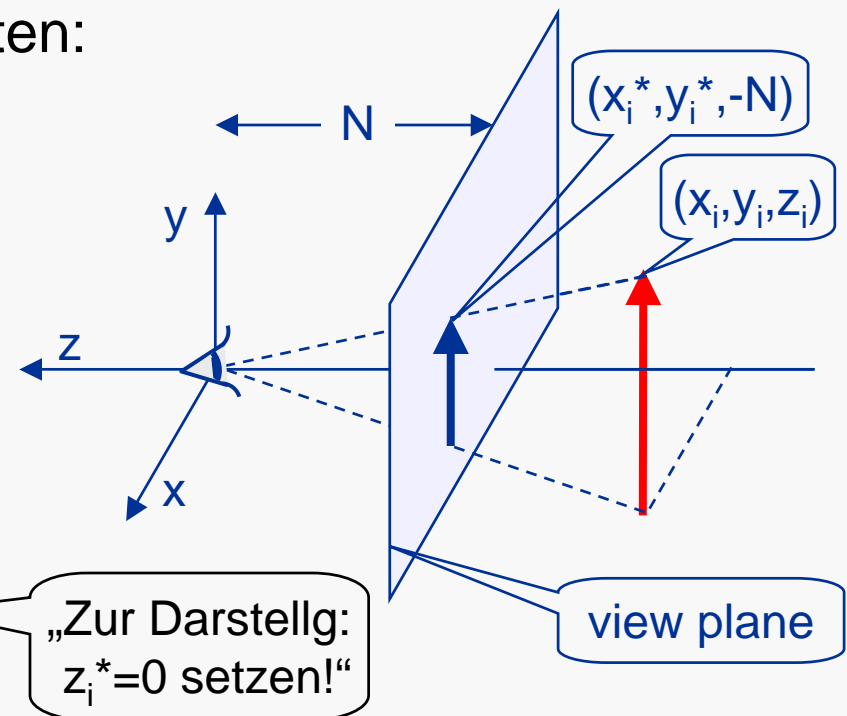
$$x_i^* / x_i = y_i^* / y_i = N / (-z_i)$$

Verwendung homogener Koordinaten:

$$\begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} Nx_i \\ Ny_i \\ Nz_i \\ -z_i \end{pmatrix}$$

perspektivische
Division

$$\begin{pmatrix} Nx_i / (-z_i) \\ Ny_i / (-z_i) \\ Nz_i / (-z_i) \\ -z_i / (-z_i) \end{pmatrix} = \begin{pmatrix} x_i^* \\ y_i^* \\ -N \\ 1 \end{pmatrix}$$

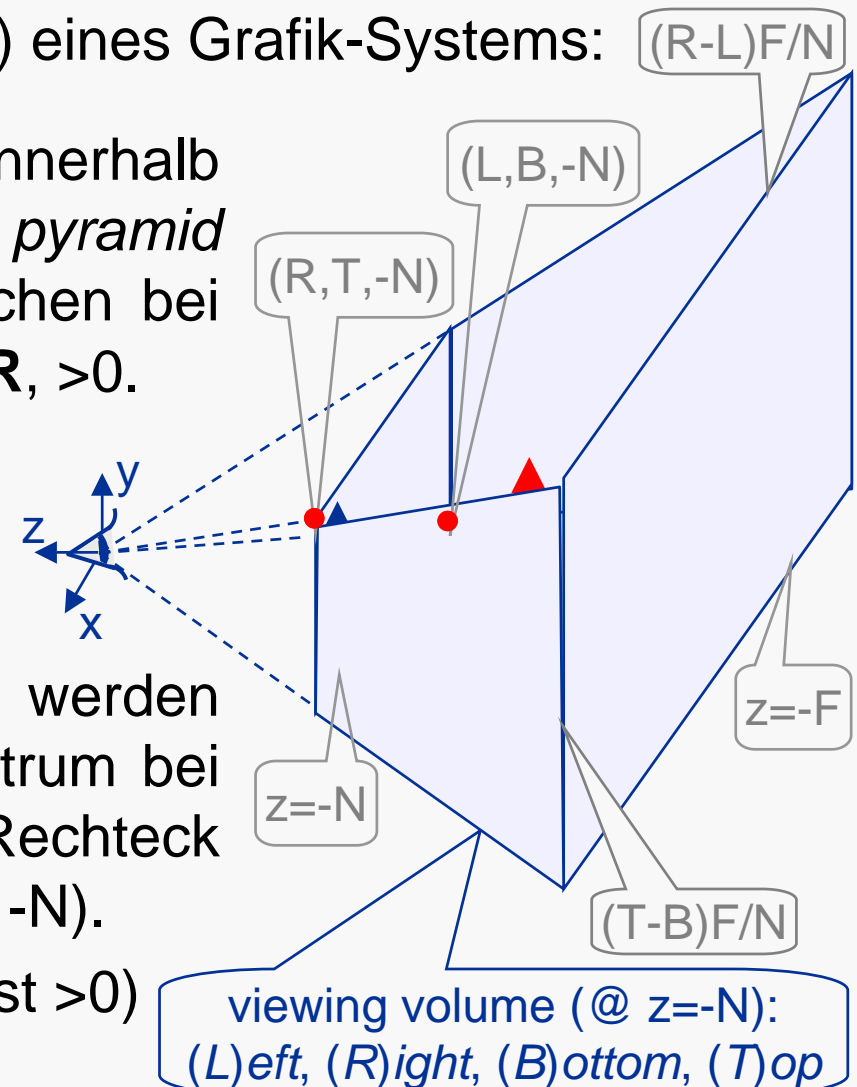


Sichtvolumen (engl. *view volume*) eines Grafik-Systems:

Darstellung nur für Objektpunkte innerhalb eines Pyramidenstumpfs (engl. *pyramid frustum*) mit Deck- und Grundflächen bei $z=-N$ (*near*) und $z=-F$ (*far*); $N, F \in \mathbf{R}, >0$.

Trapezförmige Seitenflächen werden bestimmt durch d. Projektionszentrum bei $(0,0,0)$ u. (OpenGL:) durch das Rechteck mit Ecken bei $(L, B, -N)$ und $(R, T, -N)$.

$(L, B \in \mathbf{R}, \text{meist } <0; R, T \in \mathbf{R}, \text{meist } >0)$



Nutzung der perspekt. Trf. für den Zuschnitt von Objekten an den Sichtvolumen-Grenzen eines Grafik-Systems (*Clipping*):
Einführung einer „Pseudo-Tiefe“ (engl. *pseudodepth*) für Objekt-Punkte mit Werten zwischen -1 ($z_i = -N$) und +1 ($z_i = -F$).

$$\begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & \mathbf{a} & \mathbf{b} \\ 0 & 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} Nx_i \\ Ny_i \\ \mathbf{az_i+b} \\ -z_i \end{pmatrix} \xrightarrow{\begin{matrix} (-aN+b)/N=-1 \\ (-aF+b)/F=+1 \end{matrix}} \begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & \frac{-(F+N)}{F-N} & \frac{-2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

„Pseudo-Tiefe“ wächst proportional zur eingestellten Nah-Grenze N und reziprok zu z_i (für entfernte Objekte ungenau):

$$F \gg N \Rightarrow F+N \approx F-N \approx F \Rightarrow (az_i+b)/(-z_i) = -a + b/(-z_i) = \frac{(F+N)/(F-N) + 2FN/[z_i(F-N)]}{1} \approx 1 + 2N / z_i$$

d.h.: „kurzsichtige“ Systeme (N klein) verdecken evtl. falsch!

Ähnliche Rechnung: Einführung eines „kanonischen Sichtvolumens“ (engl. *canonical view volume*) mit Werten zwischen -1 u. +1 in allen Dimensionen u. mit der Trf.-Matrix:

$$\begin{pmatrix} \frac{2N}{R-L} & 0 & \frac{R+L}{R-L} & 0 \\ 0 & \frac{2N}{T-B} & \frac{T+B}{T-B} & 0 \\ 0 & 0 & \frac{-(F+N)}{F-N} & \frac{-2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

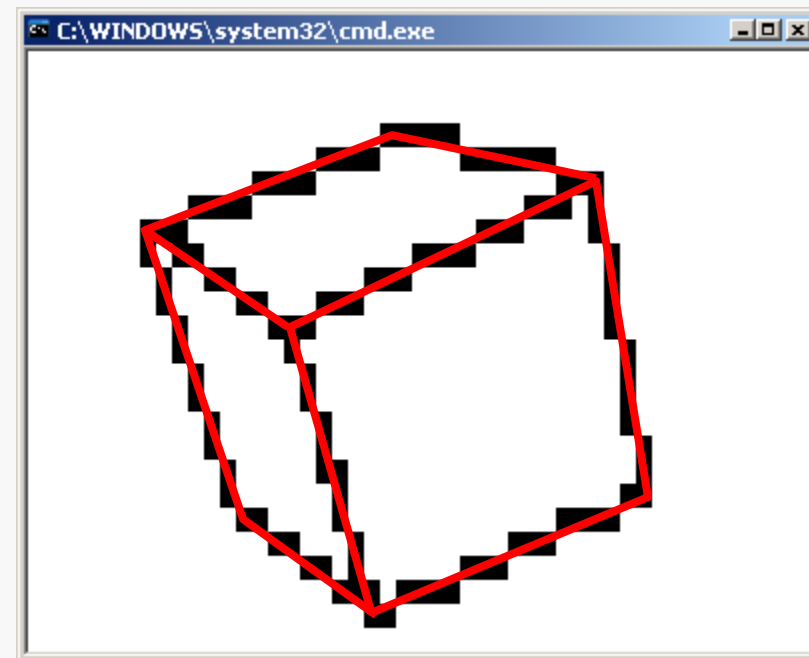
Hintergrund: Da Ebenen, die das Proj.zentrum enthalten, auf Geraden projiziert werden, genügt (nach der Trf.) ein Vergleich der transform. Koordinaten mit den Geraden, die das transf. Sichtvolumen begrenzen. Vorteil des kanon. Sichtvolumens: Koord. transformierter Objektpunkte werden geprüft auf $-1 < (x_i^*, y_i^*, z_i^*) < 1$.

Trf. des Sicht-Pyramidenstumpfs in einen Würfel; das ist eine Verformg., die bei d. Window-Viewport-Trf. aufgehoben wird.

Übung (Forts.):

Erweiterung des Programms zur Konsole-Darstellung eines 3D-Drahtmodells um die wahlweise Ausblendung abgewandter Objektflächen (Flächenmodell – *solid model*).

WireCullFill(2).exe



3D-Sicht, Projektionen

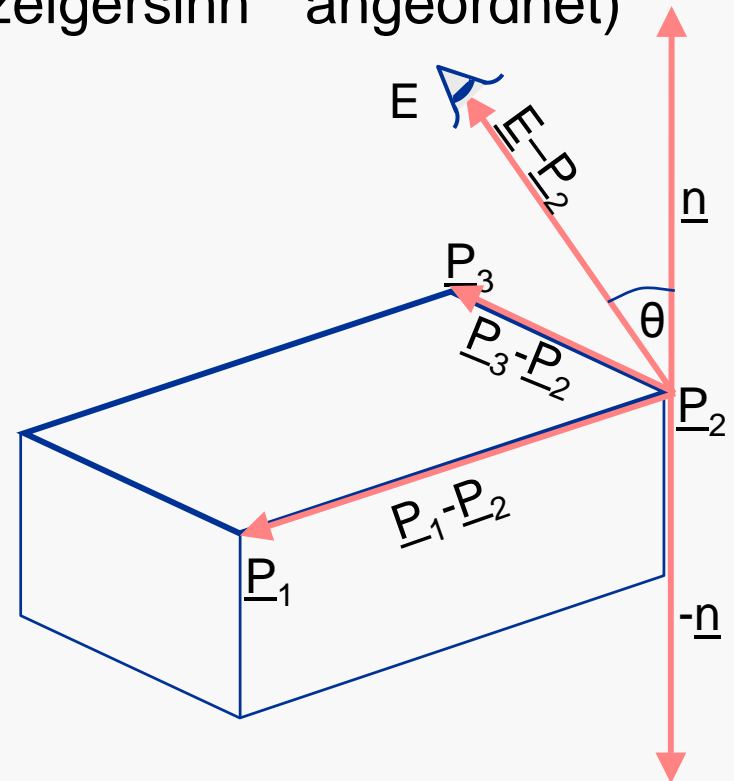
Orientierung einer Objektfläche mit den Eckpunkten $\underline{P}_1, \underline{P}_2, \underline{P}_3$ (bei Draufsicht: gegen den Uhrzeigersinn angeordnet) gegenüber dem Augenpunkt E:

Nach außen gerichtete Normale \underline{n} :

$$\underline{n} = (\underline{P}_3 - \underline{P}_2) \times (\underline{P}_1 - \underline{P}_2)$$

Winkel zwischen der Normalen und dem Verbindungsvektor vom Eckpunkt \underline{P}_2 zum Augenpunkt E:

$$\cos \theta = \underline{n} \cdot (\underline{E} - \underline{P}_2) / (|\underline{n}| \cdot |\underline{E} - \underline{P}_2|)$$



$\underline{n} \cdot (\underline{E} - \underline{P}_2) \geq 0 \Leftrightarrow -90^\circ \leq \theta \leq 90^\circ \Leftrightarrow$ sichtbare Fläche

$\underline{n} \cdot (\underline{E} - \underline{P}_2) < 0 \Leftrightarrow 90^\circ < \theta < 270^\circ \Leftrightarrow$ nicht sichtbare (Rück-)Fläche

3D-Sicht, Projektionen

Zur Erinnerung: Verwendung des Skalarprodukts zweier Vektoren zur Berechnung des Winkels θ zwischen ihnen:

$$x_{B'} = R \cdot \cos\varphi$$

$$y_{B'} = R \cdot \sin\varphi$$

$$x_{B''} = r \cdot \cos(\varphi+\theta)$$

$$y_{B''} = r \cdot \sin(\varphi+\theta)$$

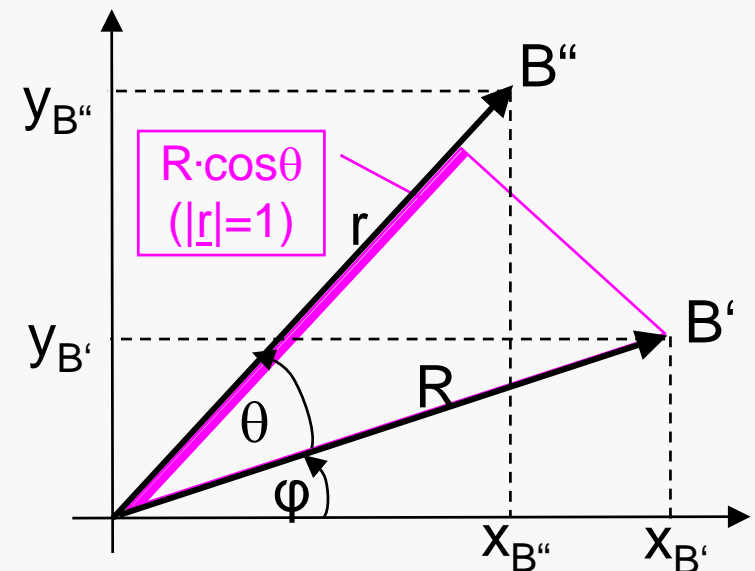
$$\begin{pmatrix} x_{B'} \\ y_{B'} \end{pmatrix} \cdot \begin{pmatrix} x_{B''} \\ y_{B''} \end{pmatrix} = x_{B'} \cdot x_{B''} + y_{B'} \cdot y_{B''}$$

$$= R \cdot \cos\varphi \cdot r \cdot \cos(\varphi+\theta) + R \cdot \sin\varphi \cdot r \cdot \sin(\varphi+\theta)$$

$$= r \cdot R \cdot \cos(-\theta) = r \cdot R \cdot \cos\theta$$

$$\Rightarrow \cos\theta = \frac{[x_{B'} \ y_{B'}] \cdot [x_{B''} \ y_{B''}]^T}{r \cdot R}$$

Das Skalarprodukt $r \cdot R \cdot \cos\theta$ eines Einheitsvektors \underline{r} ($|\underline{r}|=1$) mit einem bel. Vektor \underline{R} ist gleich der Länge der Projektion des Vektors \underline{R} auf die Achse des Einheitsvektors \underline{r} .



$$\cos(\alpha \pm \beta) = \cos\alpha \cos\beta \mp \sin\alpha \sin\beta$$

$$\sin(\alpha \pm \beta) = \sin\alpha \cos\beta \pm \cos\alpha \sin\beta$$

Beispiel zur Sichtbarkeit einer Objekfläche:

Eckpunkte $\underline{P}_1 = [1, 1, 0]^T$, $\underline{P}_2 = [0, 0, 0]^T$ und $\underline{P}_3 = [1, 0, 0]^T$

Augenpunkt $\underline{E} = [0, 0, 1]^T$:

$$\underline{n} = (\underline{P}_3 - \underline{P}_2) \times (\underline{P}_1 - \underline{P}_2) = \begin{pmatrix} 1 - 0 \\ 0 - 0 \\ 0 - 0 \end{pmatrix} \times \begin{pmatrix} 1 - 0 \\ 1 - 0 \\ 0 - 0 \end{pmatrix} = \begin{pmatrix} 0 - 0 \\ 0 - 0 \\ 1 - 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\underline{n} \cdot (\underline{E} - \underline{P}_2) = [0, 0, 1] \cdot \begin{pmatrix} 0 - 0 \\ 0 - 0 \\ 1 - 0 \end{pmatrix} = 1 \geq 0$$

⇒ Die Fläche ist sichtbar!

(Gegenprobe mit $\underline{E} = [0, 0, -1]^T$!)

