

# 3D-Sicht, Projektionen

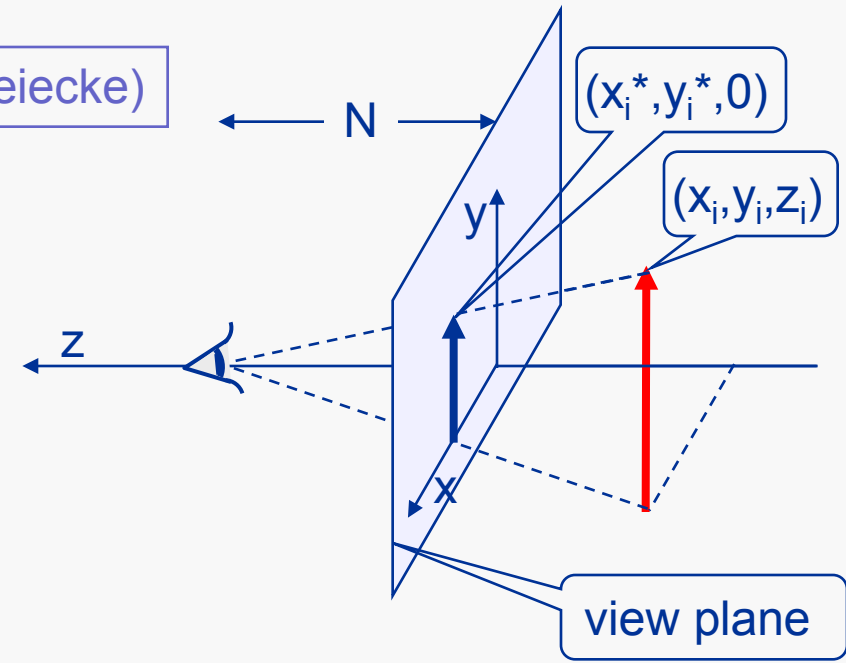
Transformationen, deren Matrix als letzte Zeile nicht die Form:  $[ 0 \dots 0 \ 1 ]$  hat, gehören zur allgemeineren Klasse der **perspektivischen Transformationen**.

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:

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





## Nebeneffekt der Homogenisierung: Gesamtskalierung

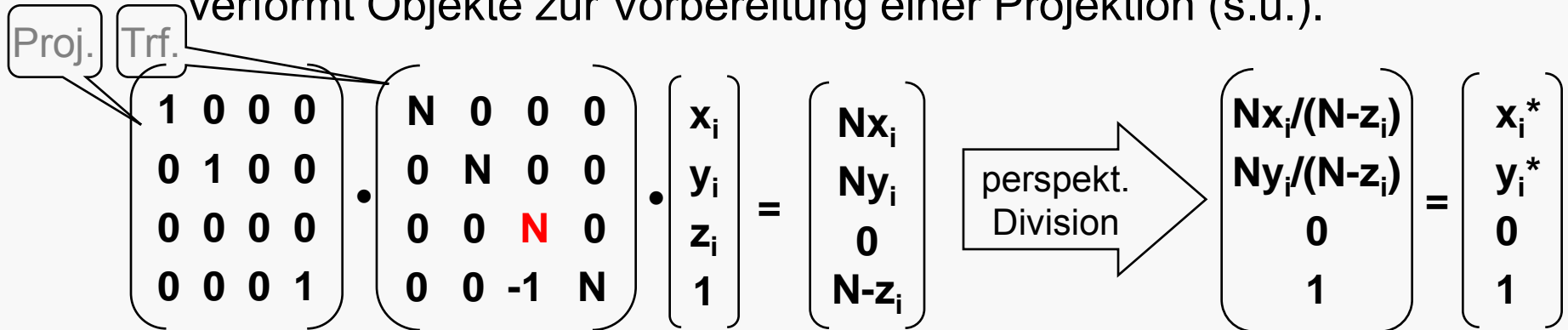
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/s \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1/s \end{pmatrix} \xrightarrow{\text{Homogenisierung}} \begin{pmatrix} sx_i \\ sy_i \\ sz_i \\ 1 \end{pmatrix}$$

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

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

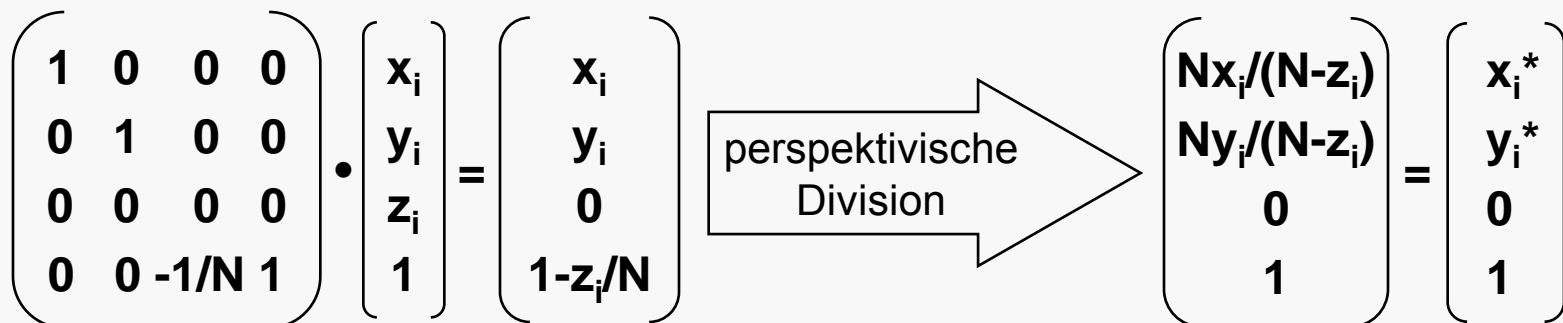
## Begriffliche und rechnerische Trennung:

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



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

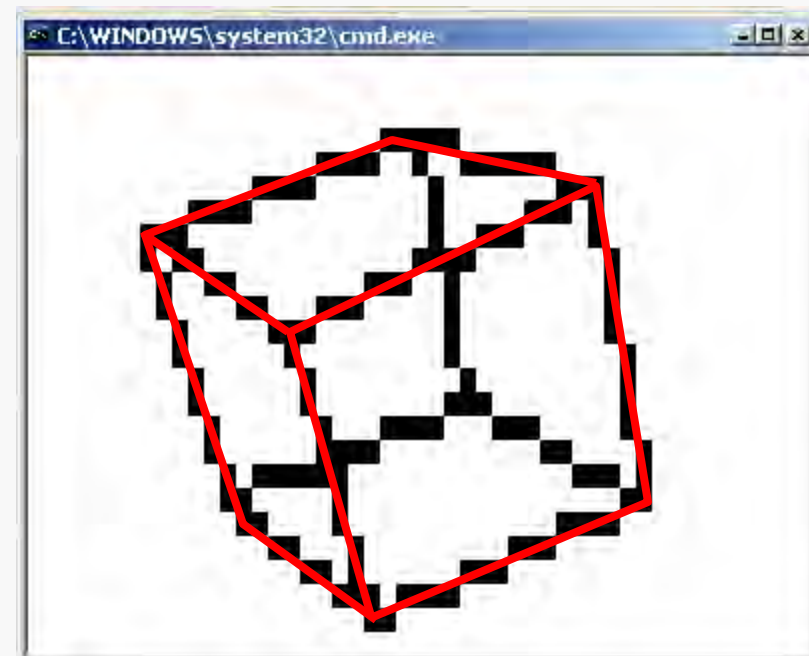
**Namensgebung:** Koordinaten „homogen“, denn sie ermöglichen auch Perspektive als Matrizen-Multiplikation i.d. Grafik-Pipeline.



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

## Übung:

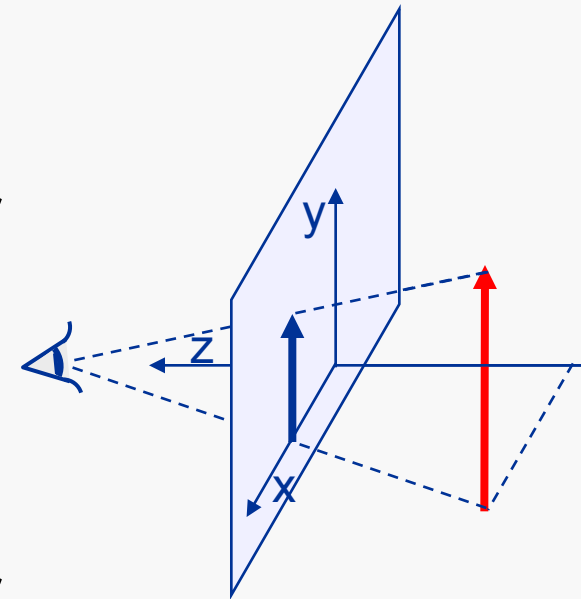
Im Konsole-Fenster sollen mit ASCII-Zeichen geladene 3D-Drahtmodelle (*wire frame*) dargestellt werden; sie sollen um die x-, y- und z-Achsen drehbar und bei veränderlichem Projektionszentrum darstellbar sein.



WireCullFill(1).exe

## Anmerkungen zur Perspektive:

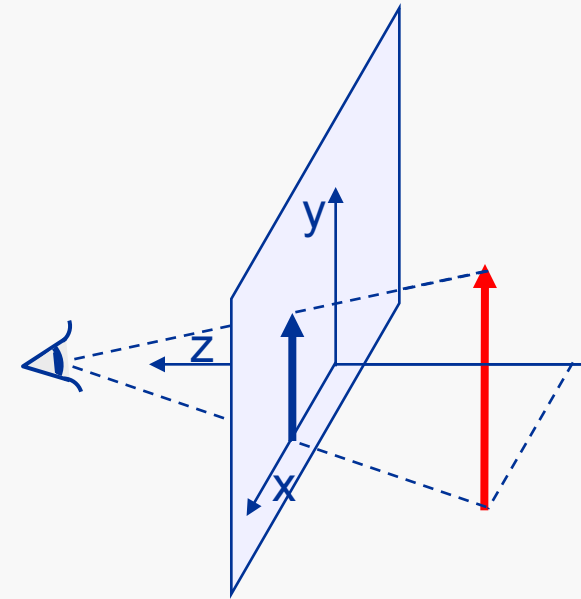
- Die perspektivische Division bewirkt, daß weiter entfernte Objekte ( $z_i$  groß) in der Projektion kleiner erscheinen.
- Verschiebung d. Projektionszentrums (N) verändert die Abbildungs-Unterschiede zwischen nah und fern.
- Verschiebung der Proj.ebene entlang der z-Achse (auf  $z \neq 0$ ) verändert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



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

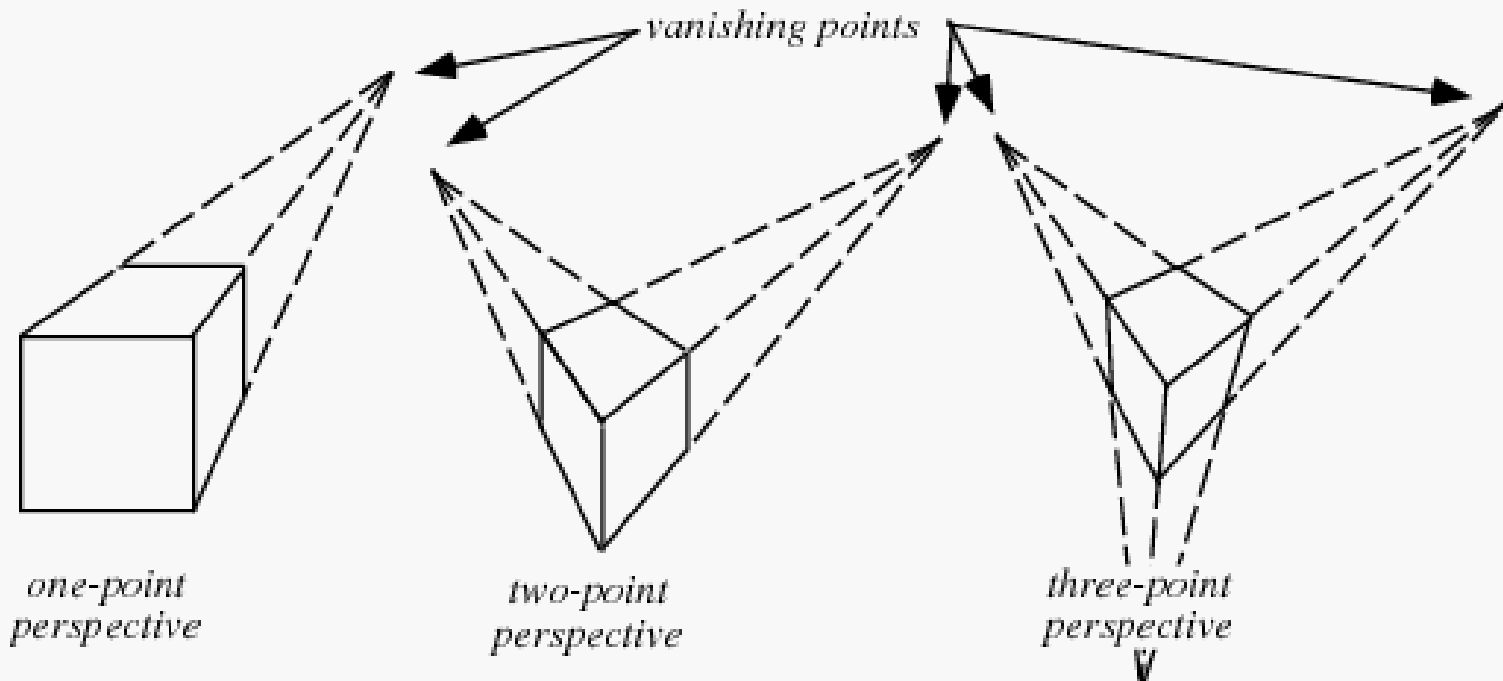
## Weitere Anmerkungen zur Perspektive:

- 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

Häufige Anwendungen in künstlerischen und technischen Darstellungen: Ein- / Zwei- / Dreipunktperspektive  
(meist abweichend von der exakten Perspektive)





# 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:

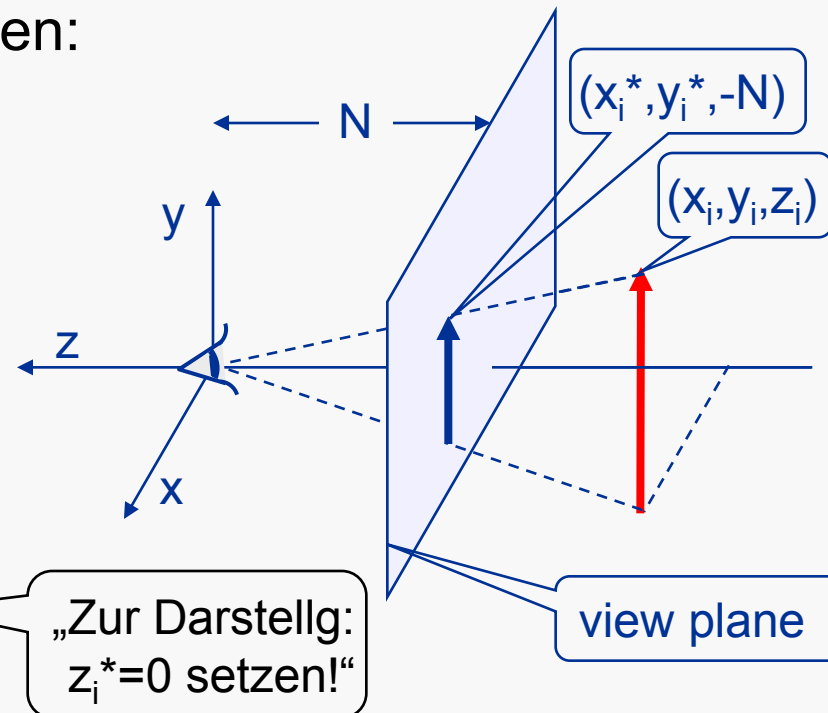
$$x_i^* / x_i = y_i^* / y_i = N / (-z_i)$$

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

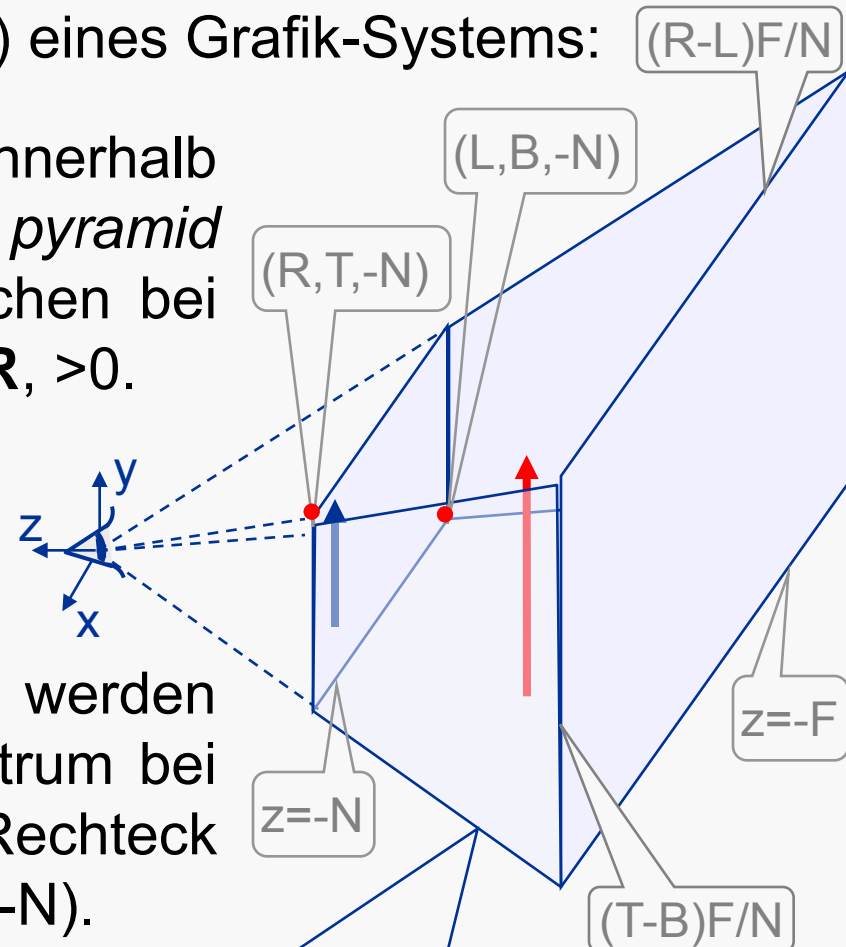
$$\begin{pmatrix} Nx_i / (-z_i) \\ Ny_i / (-z_i) \\ Nz_i / (-z_i) \\ -z_i / (-z_i) \end{pmatrix} = \begin{pmatrix} x_i^* \\ y_i^* \\ -N \\ 1 \end{pmatrix}$$



# 3D-Sicht, Projektionen

**Sichtvolumen** (engl. *view volume*) eines Grafik-Systems: (R-L)F/N

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  $(R, T, -N)$ .

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

viewing volume (@  $z=-N$ ):  
(L)eft, (R)ight, (B)ottom, (T)op

# 3D-Sicht, Projektionen

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 ( $z_i = -F$ ).

$$\begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & \mathbf{a} & \mathbf{b} \\ 0 & 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} Nx_i \\ Ny_i \\ \mathbf{az}_i + \mathbf{b} \\ -z_i \end{pmatrix} \rightarrow \begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & \frac{-(F+N)}{F-N} & \frac{-2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

$(-aN+b)/N = -1$   
 $(-aF+b)/F = +1$

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

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

„z-fighting“

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

# 3D-Sicht, Projektionen

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

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

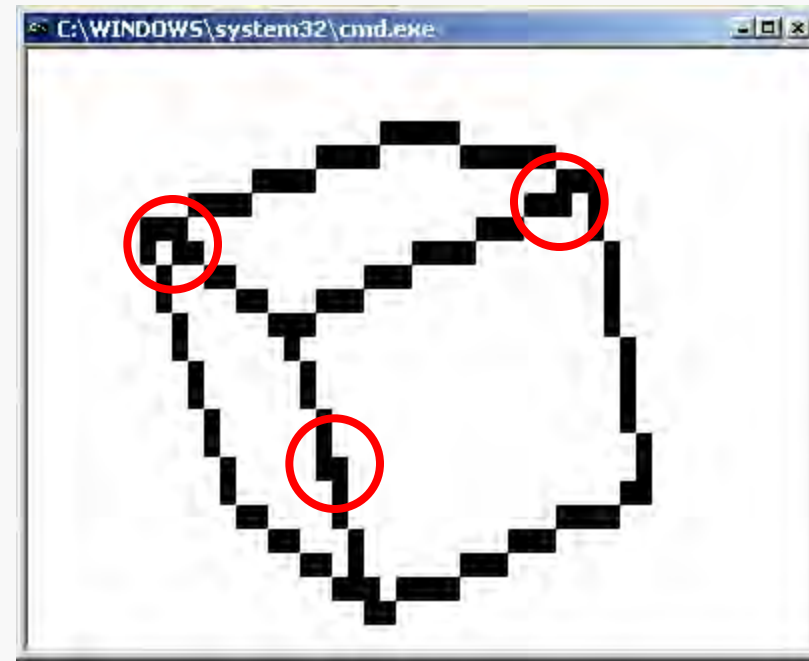
Hintergrund: Da Ebenen, die das Proj.zentrum enthalten, auf Geraden projiziert werden, genügt (nach der Trf.) ein Vergleich der transform. Koordinaten mit den Geraden, die das transf. Sichtvolumen begrenzen. Vorteil des kanon. Sichtvolumens: Koord. transformierter Objektpunkte werden geprüft auf  $-1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

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

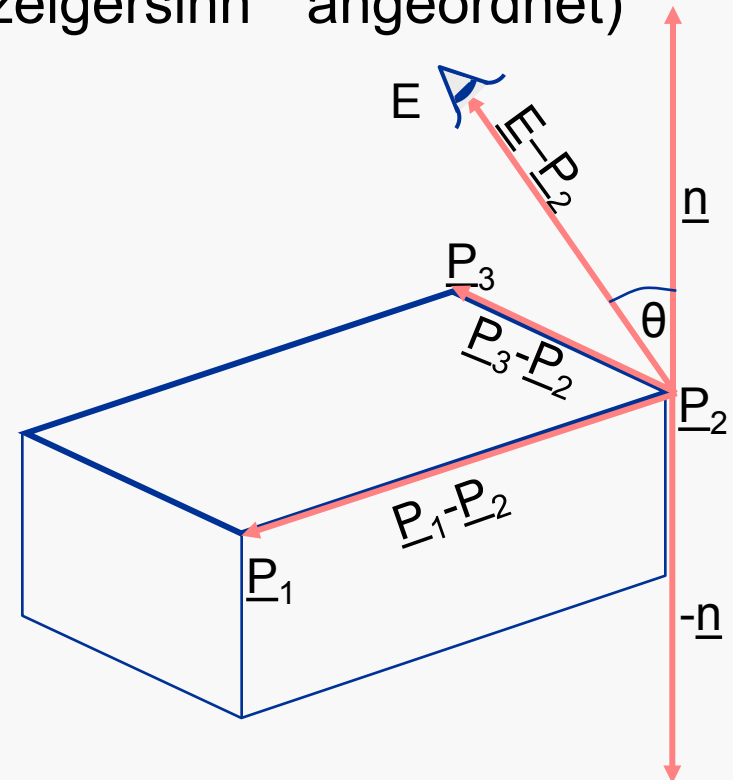
Orientierung einer Objektfläche mit den Eckpunkten  $\underline{P}_1, \underline{P}_2, \underline{P}_3$  (bei Draufsicht: gegen den Uhrzeigersinn angeordnet) gegenüber dem Augenpunkt E:

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  $P_2$  zum Augenpunkt E:

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



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

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

# 3D-Sicht, Projektionen

Zur Erinnerung: Verwendung des Skalarprodukts zweier Vektoren zur Berechnung des Winkels  $\theta$  zwischen ihnen:

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

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

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

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

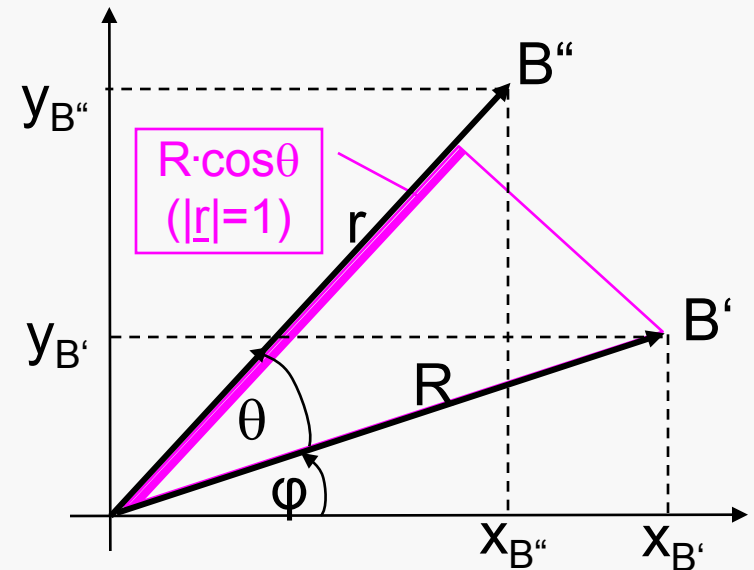
$$\begin{pmatrix} x_{B'} \\ y_{B'} \end{pmatrix} \cdot \begin{pmatrix} x_{B''} \\ y_{B''} \end{pmatrix} = x_{B'} \cdot x_{B''} + y_{B'} \cdot y_{B''}$$

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

$$= r \cdot R \cdot \cos\theta$$

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

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 Objektfläche:

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

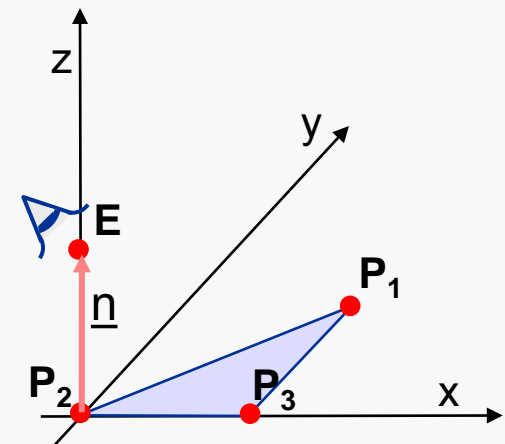
Augenpunkt  $\underline{E} = [0, 0, 1]^T$  :

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

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

⇒ Die Fläche ist sichtbar!

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





Notiz zur Sichtbarkeit:

- Benötigte Transformation der Normalen  $\underline{n}$ , die senkrecht auf dem Vektor  $\underline{p}$  standen, bis dieser mit der Matrix  $\underline{M}$  zu:

$$\underline{p}' = \underline{M} \cdot \underline{p}$$

transformiert wurde.

( $\underline{n}'$  soll mit  $\underline{p}'$  ein verschwindendes Skalarprodukt bilden)

$$0 = \underline{n}^T \cdot \underline{p} = \underline{n}^T \cdot \underline{M}^{-1} \cdot \underline{M} \cdot \underline{p} = (\underline{M}^{-1T} \cdot \underline{n})^T \cdot \underline{M} \cdot \underline{p} = \underline{n}'^T \cdot \underline{p}'$$

D.h.: Wird ein grafisches Objekt mit der Matrix  $\underline{M}$  transformiert, so müssen die dazugehörigen Normalen mit der transponierten Inverse  $\underline{M}^{-1T}$  multipliziert werden, damit sie weiterhin senkrecht zum angenommenen Verlauf stehen; die neue Ausrichtung der Normalen ist:

$$\underline{n}' = (\underline{M}^{-1})^T \cdot \underline{n}$$