3. FS - Transformationen

2D-Transformationen

Translation (Verschiebung)

Das Verschieben eines Punktes (x, y) um den Vektor (t_x, t_y) liefert den transformierten Punkt:

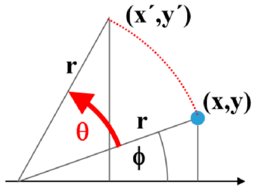
$$(x', y') = (x + t_x, y + t_y)$$

Rotation (Drehung)

Durch das Drehen eines Objektes mit dem Winkel θ um den Koordinatenursprung kommt der Punkt (x, y) wegen $x = r \cdot cos\theta$ und $y = r \cdot sin\theta \Rightarrow x' = r \cdot cos(\phi + \theta) = r \cdot cos\phi \cdot cos\theta - r \cdot sin\phi \cdot sin\theta = x \cdot cos\theta - y \cdot sin\theta$ (und y' analog) auf

$$(x', y') = (x \cdot cos\theta - y \cdot sin\theta, x \cdot sin\theta + y \cdot cos\theta)$$

zu liegen.



Skalierung (Vergrößerung oder Verkleinerung)

Beim Skalieren eines Objektes um den Faktor s um den Ursprung $(0, 0)$ wird ein Punkt (x, y) auf

$$(x', y') = (s \cdot x, s \cdot y)$$

abgebildet. Wenn in x- und y-Richtung unterschiedliche Skalierungsfaktoren s_x und s_y verwendet werden, dann erhält man

$$(x', y') = (s_x \cdot x, s_y \cdot y).$$

Reflexion (Spiegelung)

Die Spiegelung an einer Koordinatenachse ist ein Sonderfall der Skalierung mit $s_x = -1$ oder $s_y = -1$.

Alle anderen Transformationen können durch Hintereinanderausführen der beschriebenen einfachen Transformationen erreicht werden. Diese Abbildungen (mit Ausnahme der Translation) lassen sich auch durch Transformationsmatrizen darstellen. Dabei werden die Punkte als Vektoren dargestellt, um mit ihnen die Matrixoperationen durchführen zu können:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

(a) Skalierung

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

(b) Rotation (gegen Uhrzeigersinn)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

(c) Spiegelung um x-Achse

$$(x', y') = (x + t_x, y + t_y)$$
...?

(d) Verschiebung

3. Transformationen > Einfache 2D-Transformationen

Homogene Koordinaten:

- $x = \frac{x'}{h}$
- $y = \frac{y'}{h}$

3. Transformationen > Homogene Koordinaten

Transformationen als Matrizen

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(a) 2D-Skalierung

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(b) 2D-Rotation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(c) 2D-Translation

3. Transformationen > Grundlegende Transformationen

Kurznotation der Transformationen

- $T(tx, ty)$ = Translation um Vektor (tx, ty)
- $R(\theta)$ = Rotation um Winkel θ
- $S(s_x, s_y)$ = Skalierung in x- und y-Richtung

Inverse Transformationen

- $T^{-1}(tx, ty) = T(-tx, -ty)$
- $R^{-1}(\theta) = R(-\theta)$
- **$S^{-1}(sx, sy) = S(1/sx, 1/sy)$

3. Transformationen > **Kurznotation für Transformationen**

3D Transformationen

3D Transformationen

Alle 2D-Konzepte lassen sich leicht auf 3D erweitern. Man benötigt wieder eine homogene Komponente, so dass 4x4-Matrizen auf 4-dimensionalen Vektoren operieren. Später werden wir sehen, dass man auch Projektionen auf diese Art formulieren kann.

Hier sind einmal die wichtigsten 3D-Transformationen:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

(a) 3D-Skalierung

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

(b) 3D-Translation

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c) Spiegelung um yz-/xz-/xy-Ebene

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

(d) 3D-Rotation um x-Achse

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

(e) 3D-Rotation um y-Achse

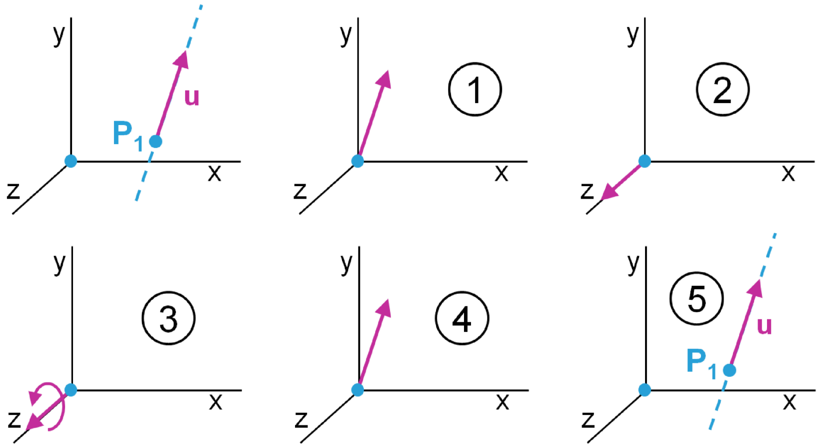
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

(f) 3D-Rotation um z-Achse

Die Namen für diese einfachen Transformationsmatrizen sehen nun so aus:

- $\mathbf{T(t_x, t_y, t_z)}$ = Translation um den Vektor (t_x, t_y, t_z)
- $\mathbf{R_x}(\theta)$ = Rotation um den Winkel θ um die x-Achse; y- und z-Achse analog
- $\mathbf{S(s_x, s_y, s_z)}$ = Skalierung um die Faktoren s_x, s_y und s_z.

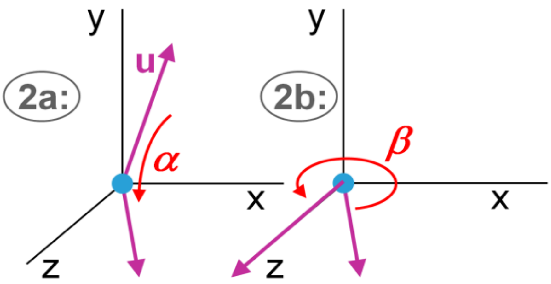
Als Beispiel für eine komplexere Transformation wollen wir eine
Drehung um den Winkel θ um eine beliebige Achse im Raum
herleiten. Die Achse sei durch einen Punkt $P_1(x_1, y_1, z_1)$ und einen Richtungsvektor u gegeben.



1. Schritt = Punkt P_1 in den Koordinatenursprung verschieben: $T(-x_1, -y_1, -z_1)$
2. Schritt = Vektor u in die z-Achse drehen
   (a) Vektor u um die x-Achse in die xz-Ebene drehen: $R_x(\alpha)$
   Sei $u = (a, b, c)$, dann ist $u' = (0, b, c)$ die Projektion von u auf die yz-Ebene. Der Drehungswinkel α um die x-Achse ergibt sich aus $cos\alpha = c/d$ mit $d = \sqrt{(b^2 + c^2)}$
   (b) Vektor u um die y-Achse in die z-Achse drehen: $R_y(\beta)$
   Der Drehungswinkel β um die y-Achse ergibt sich aus $cos\beta = d$ (bzw. $sin\beta = -a$)
3. Schritt = Drehung um θ um die z-Achse ausführen: $R_z(\theta)$
4. Schritt = Vektor u in die ursprüngliche Richtung zurückdrehen: zuerst $R_y(-\beta)$, dann $R_x(-\alpha)$
5. Schritt = Punkt P_1 an die ursprüngliche Position zurückverschieben: $T(x_1, y_1, z_1)$

Die resultierende Matrix berechnet sich also so:

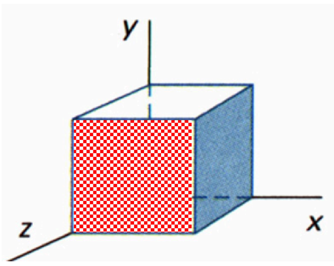
$$R(\theta) = T^{-1}(-x_1, -y_1, -z_1) \cdot R_x - 1(\alpha) \cdot R_y - 1(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T(-x_1, -y_1, -z_1) =$$
$$= T(x_1, y_1, z_1) \cdot R_x(-\alpha) \cdot R_y(-\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T(-x_1, -y_1, -z_1)$$

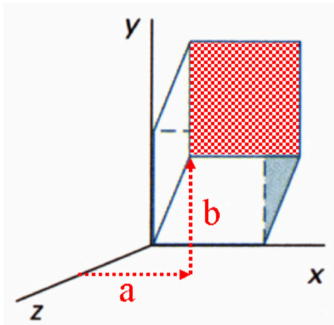


Die **Scherung in 3D** ist ebenfalls einfach darstellbar:

Eine Scherung parallel zur xy-Ebene um den Wert a in x-Richtung und den Wert b in y- Richtung erreicht man mittels

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





Eine Scherung mit einer anderen Fixebene als einer der Koordinatenhauptebenen kann man sich leicht ableiten.

3. Transformationen > 3D Transformationen