

4. Homogene lineare Algebra

4.1 Umformungen / Rechenoperationen

Verschiebung: $\vec{a} + \vec{b}$ (Addition)

Skalierung (Verkleinern / Vergrößern): $\alpha \cdot \vec{p}$ (Skalarmultiplikation)

Drehung: $R \cdot \vec{p}$ (Matrixmultiplikation)

Wird von 3D auf 2D vereinfacht. 2D ist einfacher, die Operationen sind aber grundsätzlich gleich. Daher wird im Folgenden nur in 2D gearbeitet (\rightarrow Euklidischer Raum)

Addition (Verschieben):

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$$

Skalarmultiplikation (Skalierung):

$$\alpha \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} \alpha \cdot p_x \\ \alpha \cdot p_y \end{pmatrix}$$

Matrixmultiplikation (Drehung):

$$\begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

Problem: Die drei Operationen werden für die häufigsten Umformungen benötigt, müssen aber separat von einander berechnet werden.

Lösung: homogene lineare Algebra

\rightarrow Änderung aller Operationen zu Matrixmultiplikationen

\rightarrow Kunstgriff: Hinzufügen einer neuen Koordinate zu allen Vektoren und Matrizen (NICHT „z“, sondern „w“ für z.B. Weltkoordinate)

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \Rightarrow \begin{pmatrix} a_x \\ a_y \\ 1 \end{pmatrix} = \vec{a}, \text{ aber homogen (die „1“ wird auch als homogene 1 bezeichnet)}$$

Skalierung:

$$\alpha \cdot \vec{p} = \alpha \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix} \Rightarrow \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = S \cdot \vec{p}$$

Die Skalierungsmatrix erlaubt NICHT-gleichmäßige Skalierung ($s_x \neq s_y$)

Was meint $s_x < 0$; $s_y < 0$; $s_x, s_y < 0$?

Setzt man die Werte für s_x , s_y oder beide Werte kleiner 0, so entstehen negative Koordinaten, die Flächen werden allerdings trotzdem entsprechend den Vorgaben skaliert. Sie sind nur nach dem Skalieren an den anderen Stellen, welche sich durch ein Spiegeln an den Achsen ergeben würden.

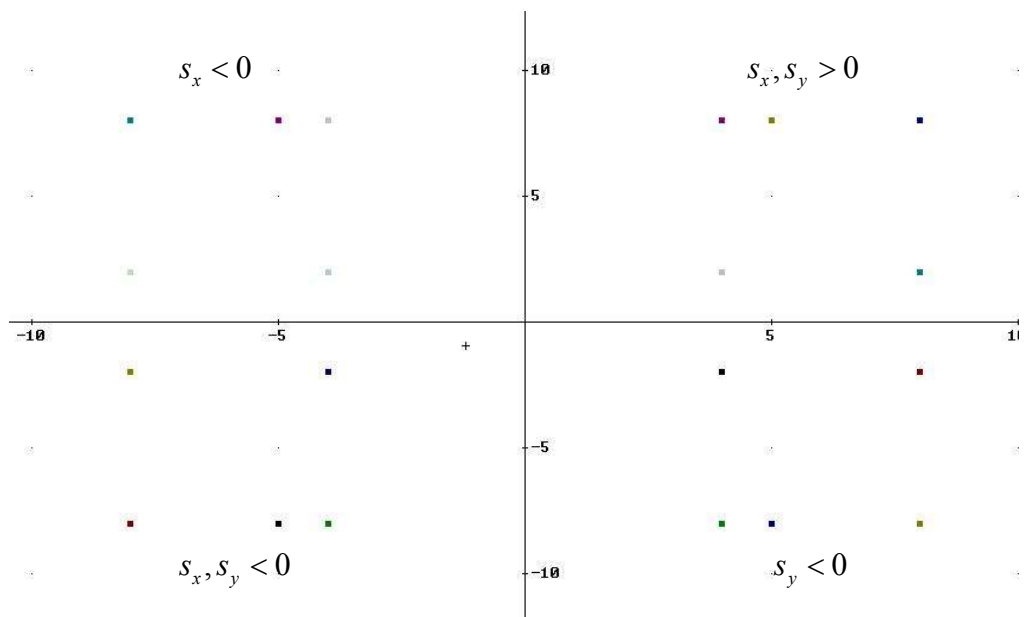
In Blender wird jenes nicht umgesetzt, das Objekt bleibt an jener Stelle, an der es skaliert wird, auch wenn man negative Werte darauf anwendet.

Setzt man nur $s_x < 0$, so wird das Objekt skaliert und an der y-Achse gespiegelt ausgegeben.

Setzt man nur $s_y < 0$, so wird das Objekt skaliert und an der x-Achse gespiegelt ausgegeben.

Setzt man beide Werte < 0 , so wird das Objekt skaliert und an beiden Achsen gespiegelt ausgegeben.

Das folgende Bild verdeutlicht diese Erläuterung:



Drehung:

$$R \cdot \vec{p} = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix} \Rightarrow \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = R \cdot \vec{p}$$

Verschiebung:

$$\vec{p} + \vec{\Delta} = \begin{pmatrix} p_x + \Delta_x \\ p_y + \Delta_y \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & \Delta_x \\ 0 & 1 & \Delta_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = T_{\Delta} \cdot \vec{p}$$

4.2 Verknüpfung der Umformungen

Beispiel: Verschiebung, Skalierung und Drehung einer Fläche

- 1) Verschieben des Dreh- und Angelpunkts (gemeinsam mit dem Objekt) in den Ursprung (T_-)
- 2) Objekt auf die neue Größe skalieren (S)
- 3) Objekt in neue Richtung drehen (R)
- 4) Verschieben des Dreh- und Angelpunktes (gemeinsam mit dem Objekt) zurück auf die ursprüngliche Position (T_+)

Verknüpfung der Schritte:

$$T_+ \cdot R \cdot S \cdot T_- = K \text{ (nicht kommutativ!!)}$$

$$K = \begin{pmatrix} 1 & 0 & \Delta_x \\ 0 & 1 & \Delta_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -\Delta_x \\ 0 & 1 & -\Delta_y \\ 0 & 0 & 1 \end{pmatrix}$$

Prinzip: Zunächst wird K berechnet, anschließend wird K auf alle Ecken des Objekts angewendet

Beispiel:

$$C = \begin{pmatrix} 10 \\ 5 \\ 1 \end{pmatrix}, C' = ? \Rightarrow C' = K \cdot C$$

C entspricht der Koordinate einer Ecke eines Objekts