Matrix-Mathematik

Transformationen im dreidimensionalen Raum werden in der Computergrafik häufig mit Hilfe von 4×4-Matrizen dargestellt. Die drei Raumkoordinaten der Punkte werden dabei um eine vierte Dimension, die sogenannte homogene Koordinate w, erweitert. Mit diesem ‘Trick’ ist es möglich, jede affine Abbildung von $R^3 \rightarrow R^3$ durch eine Matrix-Vektor-Multiplikation im Vierdimensionalen darzustellen [1].

Eine 4×4-Matrix A bildet jeden Punkt v auf sein Bild v’ durch eine Matrix-Vektor-Multiplikation ab: v' = Av

Dabei hat eine Translationsmatrix die folgende Form:

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Wird also ein Punkt v in homogenen Koordinaten (w = 1) mittels v' = Tv transformiert, hat sein Bild die Koordinaten

$$v' = Tv = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix} = \begin{pmatrix} v_x + t_x \\ v_y + t_y \\ v_z + t_z \\ 1 \end{pmatrix}$$

Entsprechendes gilt für die Skalierungsmatrix S, die auch mit unterschiedlichen Skalierungsfaktoren für jede Koordinate zurechtkommt:

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und die Rotationen gegen den Uhrzeigersinn um die drei Koordinatenachsen für einen Rotationswinkel φ

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und

$$R_z = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotationen um beliebige Achsen können aus Rotationen um die Koordinatenachsen zusammengesetzt werden.

Die in homogenen Koordinaten vorliegenden Punkte lassen sich wie folgt in den ‘gewöhnlichen’ dreidimensionalen Raum zurückwandeln:

$$v_4(v_x, v_y, v_z, w)^T \leftrightarrow v_3(v_x/w, v_y/w, v_z/w)^T$$

Punkte mit einer homogenen Koordinate w = 0 sind als Grenzwert vereinbart:

$$v_4(v_x, v_y, v_z, 0)^T := \lim_{w \rightarrow \infty} v_4(w * v_x, w * v_y, w * v_z, 1)^T$$

Sie liegen gewissermaßen im Unendlichen.

Zur Berechnung eines Bildes der Szene durchlaufen meist viele Raumpunkte v_1 bis v_n dieselbe Kette von Drehungen, Skalierungen und Verschiebungen, die durch Matrizen M_1 bis M_m dargestellt werden:

$$v'_i = M_1 * M_2 * \ldots * M_m * v_i, \; i = 1,\ldots,n$$

Aufgrund der Assoziativität der Matrixmultiplikation lassen sich zunächst alle Transformationsmatrizen M_j zu einer Matrix T zusammenfassen:

$$T = \prod_{j=1}^{m} M_j$$

Für jeden Punkt v_i, der die Transformationskette durchlaufen soll, genügt anschließend eine einzige Matrix-Vektor-Multiplikation, um ihn auf seinen Bildpunkt v'_i abzubilden:

$$v'_i = T * v_i$$

Die State-Maschine OpenGL speichert lediglich die Matrix T. Werden mittels *glTranslate** et cetera weitere Transformationen angefordert, so berechnet OpenGL eine passende Transformationsmatrix und multipliziert diese von rechts an die Matrix T. Das Resultat wird wiederum in T gespeichert. Da die Matrixmultiplikation nicht kommutativ ist, kommt es auf die Reihenfolge der Transformationen an. OpenGL multipliziert Transformationen und Punkte stets von rechts an die gespeicherten Matrizen, daher wird die zuletzt angegebene Transformation als erste auf den zu transformierenden Punkt angewandt.