

II.B.33

Lineare Algebra und analytische Geometrie

Matrizenrechnung – Wie bewegt sich eine Spielfigur im virtuellen Raum?

Mona Hitznauer



© gorodenkoff/istock/Getty Images Plus

Diese Unterrichtseinheit zeigt Matrizenrechnung lebensnah am Beispiel moderner Computergrafik. Die Lernenden nutzen Matrizenaddition, Skalarmultiplikation, Matrizenmultiplikation, inverse Matrizen und Rechengesetze zur Beschreibung von Translation, Rotation und kombinierten Transformationen. Sie trainieren mathematisches Argumentieren, den Umgang mit mathematischen Darstellungen und das Arbeiten mit digitalen Medien. Gruppenarbeit, Selbsttest und dynamische GeoGebra-Simulationen unterstützen die Lernziele des Lehrplans anschaulich, praxisnah und methodisch vielfältig.

KOMPETENZPROFIL

Klassenstufe:	12
Dauer:	10 Unterrichtsstunden
Kompetenzen:	mathematisch argumentieren (K1), mathematische Darstellungen verwenden (K4), mit mathematischen Objekten umgehen (K5), mit Medien mathematisch arbeiten (K7)
Inhalt:	Matrizenaddition, Produkt aus Skalar und Matrix, Matrizenmultiplikation, Inverse, Rechengesetze für Matrizen

Auf einen Blick

Einstieg

M 1 Matrizenrechnung in der Computergrafik

Erarbeitung

M 2 Translation

M 3 Rotation

M 4 Effekte, Interpolation und Recall

M 5 Kombinierte Transformationen

Benötigt: ☐ Digitale Endgeräte mit Internet

Übung

M 6 Gemischte Übungen zur Matrizenrechnung

Lernerfolgskontrolle

M 7 Selbsttest zur Matrizenrechnung

Minimalplan

Die Zeit ist knapp? Dann planen Sie die Unterrichtseinheit für 5–7 Stunden wie folgt:

M 1 Matrizenrechnung in der Computergrafik

M 2 Translation

M 3 Rotation

M 5 Kombinierte Transformationen

Erklärung zu den Symbolen



Dieses Symbol markiert differenziertes Material. Wenn nicht anders ausgewiesen, befinden sich die Materialien auf mittlerem Niveau.



einfaches Niveau



mittleres Niveau



schwieriges Niveau

Matrizenrechnung in der Computergrafik

M 1

Minecraft, Fortnite, Final Fantasy, Wii Sports – Computerspiele sind für viele aus dem Alltag nicht mehr wegzudenken. Doch wer denkt dabei an Mathematik? Abschalten, Dopamin und Spaß, während die Grafikkarte der Spielekonsole oder des Computers Höchstleistungen erbringt.

In unglaublicher Geschwindigkeit verarbeiten moderne Grafikkarten tausende von Berechnungen.

Auf eine davon ist sie besonders optimiert: die Matrizenrechnung bzw. Matrizenmultiplikation.

Die Bewegungen einer Spielfigur in der virtuellen Welt werden z. B. deshalb vorzugsweise mit Matrizen dargestellt.

Virtuelle Welt

Die virtuelle Welt wird meist als rechtshändiges, kartesisches Koordinatensystem beschrieben.

Das heißt, die x-Achse schließt mit der y-Achse (und die y-Achse mit der z-Achse) einen mathematisch **positiven, rechten** Winkel ein.

Neben diesem System gibt es das – ebenfalls rechtshändige, kartesische – *lokale Koordinatensystem* der Spielfigur.

Modell-Matrix in der 2D-Welt

Eine Modell-Matrix speichert dieses *lokale Koordinatensystem* für eine Spielfigur. Mit ihr können die aktuelle Position (ein Ankerpunkt der Figur) und ihre Orientierung bzgl. der virtuellen Welt bestimmt werden.

Diese Modell-Matrix besteht in einer 2D-Welt aus drei Spaltenvektoren:

- Die Orientierung der Figur definieren zwei Richtungsvektoren, die stets senkrecht aufeinander stehen und jeweils stets die Länge eins haben:
 - **Rechtsvektor** (right-vector) $\vec{r}$
 - **Obenvektor** (up-vector) $\vec{u}$
- Die Position der Spielfigur hält der **Positionsvektor** (position-vector) $\vec{p}$ fest.

In dieser Reihenfolge nebeneinandergesetzt in einer Matrix ergeben diese drei Spaltenvektoren die aktuelle Modell-Matrix der Spielfigur.

In einer 2D-Welt sieht die Modell-Matrix (vereinfacht) daher wie folgt aus:

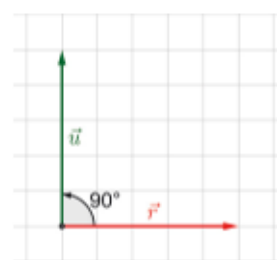
$$\begin{pmatrix} r_x & u_x & p_x \\ r_y & u_y & p_y \end{pmatrix}$$

Die Abbildung zeigt das *lokale Koordinatensystem* einer Spielfigur. Die Spielfigur selbst ist nicht abgebildet, lediglich ihr Ankerpunkt, der zugleich ihre Position beschreibt. Die lokalen Achsen werden über die Richtungsvektoren (Rechts- und Obenvektor) definiert.

Der Rechtsvektor schließt dabei immer einen mathematisch positiv definierten, rechten Winkel mit dem Obenvektor ein.

Wir definieren für die folgenden Aufgaben und Überlegungen fortan,

dass wir den Ankerpunkt einer Spielfigur in einer 2D-Welt (Ebene) betrachten und der Rechtsvektor in die Richtung zeigt, in die die Spielfigur blickt (Blickvektor).



Mona Hitznauer © Raabe

Hinweis:

Intuitiv könnte man annehmen, dass für die Darstellung eines Objekts in der 2D-Welt eigentlich zwei Vektoren reichen müssten: die Position, dargestellt durch den Positionsvektor $\vec{p}$, und die Blickrichtung, dargestellt durch den Rechtsvektor $\vec{r}$.

M 2

Translation

Die wichtigsten Transformationen einer Spielfigur sind die Veränderung ihrer Position – die Translation – und die Drehung ihrer Blickrichtung – die Rotation.

Bei der **Translation** (Verschiebung) würden wir in der linearen Algebra zum Positionsvektor einen Translationsvektor $\vec{t}$ addieren:

$$\begin{pmatrix} p'_x \\ p'_y \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} p_x + t_x \\ p_y + t_y \end{pmatrix}$$

Die Grafikkarte ist allerdings besonders auf die Matrizenmultiplikation optimiert, daher führt sie die Translation mithilfe von sogenannten *homogenen Koordinaten* durch.

Dabei werden die Spaltenvektoren einer Matrix um eine Dimension zu einer **quadratischen Matrix** ergänzt.

Die ergänzten Elemente sind die der **Einheitsmatrix**.

Die Modell-Matrix sieht deshalb in *homogenen Koordinaten* wie folgt aus:

$$M = \begin{bmatrix} r_x & u_x & p_x \\ r_y & u_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Hinweis: Diese unechten 3D-Koordinaten stehen zur Unterscheidung meist in eckigen Klammern.

Die Translation führt die Grafikkarte aus, indem sie zunächst die zugehörige **Translationsmatrix** (in homogenen Koordinaten) mit dem Translationsvektor $\vec{t}$ aufstellt:

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

Anschließend muss die Translationsmatrix T nur noch mit der Modell-Matrix M multipliziert werden. Das Ergebnis ist die neue, translatierte Modell-Matrix M_T , bei der die zwei Richtungsvektoren unverändert sind und der Positionsvektor um die entsprechenden Werte verschoben ist.

Aufgabe

Berechnen Sie allgemein das Matrizenprodukt $M_T = T \cdot M$.

Beweisen Sie damit die Aussage, dass nur der Positionsvektor entsprechend verschoben ist.

M 3

Rotation

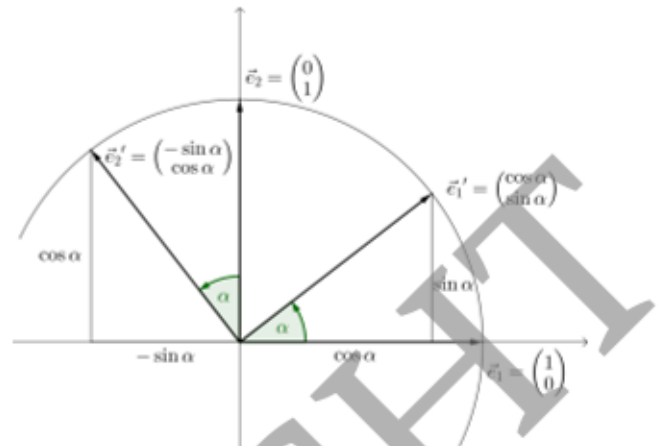
Wir wollen eine lineare Abbildung der Form

$$\vec{v}' = A \cdot \vec{v}$$

finden, die einen Vektor $\vec{v}$ in der Ebene um einen festen Ursprung um den Winkel α mithilfe einer Rotationsmatrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ rotiert.}$$

Da die Rotation eine lineare Abbildung ist, können wir die Abbildung der Einheitsvektoren bestimmen und die Ergebnisse auf alle Vektoren im $\mathbb{R}^2$ übertragen. Für die Einheitsvektoren (siehe Abbildung) muss für die gesuchte Abbildung gelten:



© Mathze/Wikimedia Commons, CC BY 4.0. <https://commons.wikimedia.org/wiki/File:Drehmatrix-Herleitung.svg>

$\vec{e}_1' = A \cdot \vec{e}_1$ $\begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $\begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} = \begin{pmatrix} a \cdot 1 + b \cdot 0 \\ c \cdot 1 + d \cdot 0 \end{pmatrix}$ $\begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix}$	$\vec{e}_2' = A \cdot \vec{e}_2$ $\begin{pmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $\begin{pmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{pmatrix} = \begin{pmatrix} a \cdot 0 + b \cdot 1 \\ c \cdot 0 + d \cdot 1 \end{pmatrix}$ $\begin{pmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{pmatrix} = \begin{pmatrix} b \\ d \end{pmatrix}$
---	--

Die Rotationsmatrix sieht daher wie folgt aus:

$$A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Die Grafikkarte rotiert die Spielfigur um den Winkel (gemessen an der positiven x-Achse) um die Position der Spielfigur daher mit folgender Rotationsmatrix (in *homogenen Koordinaten*):

$$R = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Das Ergebnis des Matrizenprodukts aus Modell-Matrix M und Rotationsmatrix R ist die neue, rotierte Modell-Matrix: $M_R = M \cdot R$

Die Richtungsvektoren (Rechtsvektor, Obenvektor) der neuen Modell-Matrix sind entsprechend rotiert und der Positionsvektor bleibt unverändert erhalten.

M 4

Effekte, Interpolation und Recall

Matrizenaddition

Bei komplexeren Effekten, etwa physischen Kräften, die mit in die Bewegung einberechnet werden (wenn z. B. innerhalb der Spielwelt Wind die Bewegung beeinflusst), wird auch die Matrizenaddition angewendet. Dabei werden die einzelnen Elemente der Matrizen entsprechend ihrem Platz in der Matrix addiert.

Beispiel:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \quad A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} \end{pmatrix}$$

Multiplikation mit einem Skalar

Bei der Interpolation von zwei Animationen berechnet die Grafikkarte oft das Produkt zweier gewichteter Matrizen. Die Gewichtung einer Matrix M wird durch die Multiplikation mit einem Skalar s (einer reellen Zahl) erreicht:

$$A_s = s \cdot A = s \cdot \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{pmatrix} s \cdot a & s \cdot b & s \cdot c \\ s \cdot d & s \cdot e & s \cdot f \\ s \cdot g & s \cdot h & s \cdot i \end{pmatrix}$$

Inverse

Bei einigen Spielen ist es nötig, eine vorangegangene Modell-Matrix wiederherzustellen. Etwa bei einem Recall-Effekt:

Die Spielfigur „beamt“ sich mit Tastendruck auf die exakte Position (und Orientierung) zurück, die sie vor ein paar Sekunden hatte.

Inverse Matrix

Die Inverse einer Matrix A wird A^{-1} geschrieben.

Das Produkt dieser Matrizen ist die Einheitsmatrix: $E = A \cdot A^{-1}$

Die vergangene Modell-Matrix M berechnet die Grafikkarte mit der sogenannten **Inversen** der angewandten Transformationsmatrix F . Dazu bestimmt sie das Produkt aus inverser Transformationsmatrix F^{-1} und der aktuellen (transformierten) Modell-Matrix M_F :

$$M = M_F \cdot F^{-1}$$

Allerdings kann nicht jede denkbare Matrizenoperation rückgängig gemacht werden. Die Determinante einer Matrix bestimmt, ob sie invertierbar ist: Ist sie ungleich null, so ist sie invertierbar.

Determinante einer Matrix

Die Determinante einer Matrix A wird z. B. $\det(A)$ geschrieben.

Für eine Matrix ist $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ ist $\det(A) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$.

Gemischte Übungen zur Matrizenrechnung

M 6

Aufgabe 1: Modell-Matrix

- a) **Bestimmen** Sie, wo die Spielfigur steht, und einen Punkt, auf den sie direkt sieht, wenn die Modell-Matrix wie folgt lautet:

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

- b) **Bestimmen** Sie die Modell-Matrix, wenn die Spielfigur auf dem Punkt $P(0 | 3)$ steht und in Richtung des Punktes $B(-2 | 4)$ blickt.
- c) **Bestimmen** Sie die Modell-Matrix, wenn die Spielfigur auf dem Punkt $P(-2 | 3)$ steht und ihre Blickrichtung um 30° gegen die positive x-Achse (Weltachse) geneigt ist.

Aufgabe 2: Translation

Eine Spielfigur steht auf dem Punkt $P(1 | 2)$ und blickt in Richtung der positiven x-Achse. Sie soll nun drei Einheiten in x-Richtung und vier Einheiten in y-Richtung verschoben werden. **Bestimmen** Sie Ihre Modell-Matrix nach der Translation mithilfe der Matrixmultiplikation.

Aufgabe 3: Rotation

- a) **Bestimmen** Sie die Modell-Matrix, wenn die Spielfigur auf dem Punkt $P(1 | -1)$ steht und in Richtung des Punktes $B(-2 | 4)$ blickt, mithilfe der Rotationsmatrix.
- b) **Bestimmen** Sie, wo die Spielfigur steht, und mithilfe der Rotationsmatrix den Winkel zwischen Rechtsvektor und positiver x-Achse, wenn die Modell-Matrix wie folgt lautet:

$$M = \frac{1}{2} \cdot \begin{pmatrix} -1 & -\sqrt{3} & 6 \\ \sqrt{3} & -1 & 8 \\ 0 & 0 & 2 \end{pmatrix}$$

- c) Eine Spielfigur steht auf dem Punkt $P(0 | 0,5)$ und blickt parallel zur Winkelhalbierenden im ersten Quadranten. Die Spielfigur soll 90° gegen den Uhrzeigersinn gedreht werden.

Berechnen Sie ihre neue, rotierte Modell-Matrix M_R .

Aufgabe 4: Inverse

Eine Spielfigur hat jetzt die Modell-Matrix (in nicht-homogenen Koordinaten)

$$M_F = \begin{pmatrix} 0,64 & -0,77 \\ 0,77 & 0,64 \end{pmatrix}, \text{ weil sie mit}$$

$$R = \begin{pmatrix} 0,94 & -0,34 \\ 0,34 & 0,94 \end{pmatrix} \text{ rotiert wurde.}$$

Bestimmen Sie die Modell-Matrix vor der Rotation in nicht-homogenen Koordinaten.

Aufgabe 5: Kombinierte Transformation

Eine Spielfigur steht auf dem Punkt $P(1 | -1)$ und blickt in Richtung des Punktes $B(5 | 3)$. Sie geht zwei Einheiten in ihre Blickrichtung und dreht sich anschließend um 45° im Uhrzeigersinn um sich selbst. **Bestimmen** Sie mithilfe eines CAS die Modell-Matrix nach diesen Transformationen.