Transformation dreidimensionaler Koordinaten in zweidimensionale Koordinaten (German version)

Transformation dreidimensionaler Koordinaten in zweidimensionale Koordinaten (German Version)

Edward Gerhold

September 15, 2015

Deutsche Version 0.0.98 von 0.1.0

Contents

1 Einleitung

Auf einem Blatt Papier sehen wir ein dreidimensionales Koordinatensystem in den Raum zeigen. In der Realität sind die drei Basisvektoren der Abbildung zweidimensional. Denn sie zeigen in drei Richtungen, flach auf dem Papier, und gar nicht in den reellen Raum.

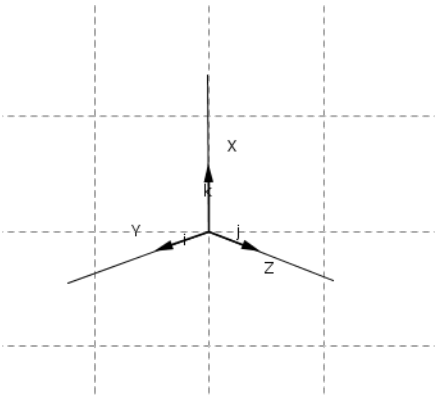


Figure 1: Bild eines rechtshändigen Koordinatensystems mit ijk-Basisvektoren auf den Achsen, in drei dimensionen zeigend. Schauen sie in (**?**) für eine Einführung.

In diesem Dokument entwerfen wir ein $\mathbb{R}^{2\times3}$ Koordinatensystem um 3-D Punkte in 2-D Punkte umzuwandeln. Mit den Kosinus- und Sinusfunktionen errechnen wir die exakten Anteile der horizontalen und vertikalen Teilverschiebungen.

Was wir in diesem Dokument machen werden.

1. Winkel auswählen für unsere Koordinatenachsen
2. Einheiten der Achsen auswählen oder sich für 1 entscheiden
3. Die drei zweidimensionalen Achsvektoren des Koordinatensystems aufschreiben
4. Die Achsen zu einer Matrix, einer Funktion, oder anstelle der Basis zusammenbauen
5. Den JavaScript Code der Transformation lesen
6. In den Folgeversionen die noch nicht eingetragenen bereits bekannten Fakten kennenlernen

2 Entwurf eines $\mathbb{R}^{2\times3}$ Koordinatensystems für unsere Koordinatentransformation vom $\mathbb{R}^3$ auf den $\mathbb{R}^2$

In der englischen Version schreibe ich alles etwas anders. Diese deutsche Fassung ist speziell neu angefangen und ich verfolge das Ziel, mich auch etwas kürzer zu fassen. Mathematik, die ich in der deutschen Fassung auslassen werde, ist aber problemlos in der englischen Version zu verfolgen. Wer das rechnen kann, kann bestimmt auch ein wenig Englisch und versteht auch deutsches Englisch.

2.1 Koordinatensystem

2.1.1 Links- und rechts

Ein linkshändiges Koordinatensystem auf der Ebene hat die dritte Achse zwischen den beiden normalen Achsen in die gleiche Richtung zeigend.

Bei einem rechtshändigen Koordinatensystem auf der Ebene zeigt die dritte Achse in die entgegengesetzte Richtung.

Figure 2: Ein linkshändiges Koordinatensystem

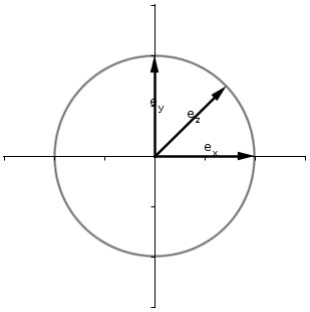
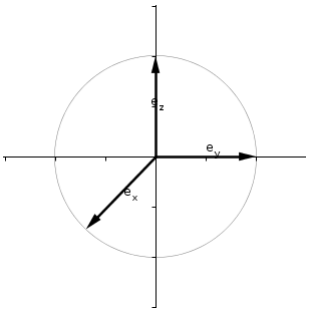


Figure 3: Ein rechtshändiges Koordinatensystem



2.1.2 Einheitskreis

Der Einfachheit halber kann man die drei Achsen am Einheitskreis ausrichten. Dazu bedienen wir uns gleich einer hilfreichen Formel aus dem Polarkoordinatensystem.

$$(x, y) := (r\cos\varphi, r\sin\varphi)$$

(x, y) stellen die Spitze eines Ortsvektors aus dem Ursprung des Koordinatensystems dar. Wenn wir drei Achsen haben wollen, sollten wir uns drei solche Vektoren, die von $(0, 0)$ bis $(x = r_n\cos\varphi_n, y = r_n\sin\varphi_n)$ zeigen, deren Längen fuer die Einheit 1 der jeweiligen Achse stehen, erzeugen. Das werden wir auch gleich tun.

2.2 Winkel der Achsen

Die Achsen werden von der horizontalen x-Achse des zweidimensionalen Bilds gegen den Uhrzeigersinn gezählt. Da wir drei Achsen haben, brauchen wir auch drei Winkel.

$$\varphi_n := \{\varphi_x, \varphi_y, \varphi_z | \text{ die Winkel der drei Achsen}\}$$

Die Winkel kann man selbst in Grad oder Radians festlegen. Die Programmiersprache JavaScript zum Beispiel nimmt, für die Funktionen Kosinus und Sinus die Winkel, in Radians an. Die sind mit einer einfachen Formel berechenbar.

$$\text{rad(deg)} := \frac{\pi}{180} \times \text{deg}$$

$$\text{deg(rad)} := \frac{180}{\pi} \times \text{rad}$$

2.3 Einheit der Achsen

2.3.1 Der r-Wert der Achsen

Der r-Wert, den man als Radius in den Polarkoordinaten kennt, steht für die Länge des Ortsvektors. Er steht somit auch für die Länge unserer Achsen. Im angewandten Sinne kann man damit unseren drei Achsen die Abstand der Einheiten eingestellt werden.

$$r_n := \{r_x, r_y, r_z | \text{ die Einheit jeder der drei Achsen}\}$$

Mit dem r-Wert $r_x = r_y = r_z = 1$ erhalten wir normalisierte Einheitsvektoren. In dem Fall kann r_n ganz weggelassen werden.

Von der Norm der Achsvektoren handelt der Abschnitt 2.4.1

2.3.2 Mathematische Vorsicht mit dem r-Wert

1. Der r-Wert verkompliziert die Berechnungen und Abschätzungen natürlich, weil die Koordinaten mit multipliziert werden.

2. Wenn man bewegte Bilder produzieren will, sollte der r-Wert grundsaetzlich gleich auf allen Achsen sein, weil Rotationen und Translationen sonst daneben gehen, da die Punkte dann plötzlich ihre Einheiten wechseln. Das sieht bewegt unrealistisch aus, ist aber bei Standbildern kein Problem. Es ist möglich selbige Veränderung lokal an den Objekten vorzunehmen, damit bleibt die Vergrösserung transformationsinvariant bezüglich den Drehungen.

3. Die Formel verkompliziert sich bei drei verschiedenen r-Werten natürlich und zum Rechnen sollte zuerst das vereinfachte Modell herangezogen werden, wo der r-Wert auf allen drei Achsen gleich ist, oder gleich 1 ist und komplett entfällt.

2.4 (Basis)vektoren der Linearkombination

Wir benötigen für das Koordinatensystem drei Achsen. Jede Achse bekommt einen kanonischen Einheitsvektor $\begin{pmatrix}\cos\varphi_n \\ \sin\varphi_n\end{pmatrix}$, wie er aus Unterricht und Büchern auch Nichtmathematikern bekannt ist. Allerdings behalte ich mir vor, das Koordinatensystem flexibel zu definieren, warum jeder Achsvektor mit dem r-Wert konfigurierbar ist.

$$\vec{e}_n := \{\vec{e}_x, \vec{e}_y, \vec{e}_z | \vec{e}_n = \begin{pmatrix} r_n \cos\varphi_n \\ r_n \sin\varphi_n \end{pmatrix}, \text{ die drei Achsen des Koordinatensystems } \}$$

Wenn der r-Wert gleich 1 ist, haben diese Vektoren gleich normalisierte Einheitslänge im Sinne der Orthonormalbasis. Der Unterschied zur Orthonormalbasis ist, dass wir mindestens eine linear abhängige Achse haben. Je nach Arrangement um den Kreis können dabei bis zu drei linear abhängige Achsen entstehen, in Bezug zur 2-D Standardbasis $\begin{pmatrix}1 & 0 \\ 0 & 1\end{pmatrix}$ auf die das Ergebnis abgebildet wird.

2.4.1 Norm der Achsvektoren

Wenn man den r-Wert, die Einheit der Achse nicht einsetzt, ergibt $\|\vec{e}_n\| = 1$

Beweis:

$$\begin{aligned}\|\vec{e}_n\| =&(\sum_{i=1}^{2} |\vec{e}_{ni}|^2)^{\frac{1}{2}} \\ =&(\cos^2\varphi_n + \sin^2\varphi_n)^{\frac{1}{2}} \\ =&\sqrt{1} \\ =&1\end{aligned}$$

Wenn man den r-Wert setzt, ist die Norm des Vektors $\|\vec{e}_n\| = r_n$.

Beweis:

$$\begin{aligned}\|\vec{e}_n\| =&(\sum_{i=1}^{2} |\vec{e}_{ni}|^2)^{\frac{1}{2}} \\ =&(r_n^2\cos^2\varphi_n + r_n^2\sin^2\varphi_n)^{\frac{1}{2}} = (r_n^2(\cos^2\varphi_n + \sin^2\varphi_n))^{\frac{1}{2}} \\ =&(r_n^2(1))^{\frac{1}{2}} \\ =&r_n\end{aligned}$$

Um mathematisch mit vielen Sätzen zu harmonieren, ist die normalisierte Länge der Achsen natürlich zu bevorzugen.

Bemerkung. Möchte man mit dem Computer einfach Graphen plotten, und mal eine Achse vergrössern, ist der r-Wert optimal.

3 Transformationswerkzeuge

In diesem Kapitel stelle ich dann vor, wie man die Transformation durchführt. Im Prinzip bleibt eins überall gleich. Wir geben den 3-D Punkt ein, und erhalten einen 2-D Punkt.

3.1 Anstelle der Basis

Wir definieren mit $x\vec{i} + y\vec{j} + z\vec{k}$ in der Regel einen Vektor mit einer 3x3 Basis. Man kann die 2x3 Basis anstelle der 3x3 Basis einsetzen und erhält eine saubere Orthogonalprojektion. Wir geben drei Koordinaten an und erhalten zwei Koordinaten. Den richtigen Punkt.

$$x\vec{e}_x + y\vec{e}_y + z\vec{e}_z = \vec{w}$$

Beweis:

$$x\vec{e}_x + y\vec{e}_y + z\vec{e}_z = \sum_{i=1}^{3} \vec{v}_i\vec{e}_i = \vec{w}$$

Bemerkung. Hierbei ist zu verstehen, das $\vec{e}_i$ ein ganzer Vektor ist, und keine Komponente.

Bemerkung. Im englischen Original stelle ich bereits sehr deutlich fest, dass die 2x3 Basis der Fläche mit drei Koordinaten im $\mathbb{R}^2$ nicht linear unabhängig ist. Allerdings drängt sich mir eine 2x3 Theorie auf, die ich in den kommenden Fassungen vorstellen werde. Die Basis, die eine oder mehrere linear kombinierte Achsen im $\mathbb{R}^2$ besitzt, wird für die $\mathbb{R}^{2\times3}$ Projektion, im Einheitskreis arrangiert. Im $\mathbb{R}^2$ nutzt man zwei statt drei der Einheitsvektoren, die man ebenso im Kreis dreht, allerdings stehen sie sicher im neunzig Grad Winkel zueinander, was mit drei Achsen nicht mehr einwandfrei möglich ist, hier aber *richtig*.

3.2 Funktional

Das Funktional nimmt einen Vektor mit drei Koordinaten an und gibt einen Vektor mit zwei Koordinaten zurück. Den richtigen Punkt auf der Ebene entsprechend den Achsen, wie wir sie konfiguriert haben.

$$\vec{f}(\vec{x}) := x \begin{pmatrix} r_x \cos\varphi_x \\ r_x \sin\varphi_x \end{pmatrix} + y \begin{pmatrix} r_y \cos\varphi_y \\ r_y \sin\varphi_y \end{pmatrix} + z \begin{pmatrix} r_z \cos\varphi_z \\ r_z \sin\varphi_z \end{pmatrix}$$

Die folgenden Bedingungen linearer Funktionale werden erfüllt

$$f(x+y) = f(x) + f(y) \qquad \lambda f(x) = f(\lambda x)$$

Das Funktional ist partiell differenzierbar. Die ersten partiellen Ableitungen ergeben abgeleitet nach x die x-Komponente des Achsvektors und abgeleitet nach y die y-Komponente des Achsvektors.

Die zweiten partiellen Ableitungen existieren nicht mehr, da das Funktional linear ist, und keine Krümmung enthäelt. Die partiellen Ableitungen sind nicht stetig im herkoemmlichen Sinne, da eine gerade Linie nur eine Steigung und keine Tangente zur Kurve hat. Dennoch ist das Ergebnis zuverlässig und nützlich.

$$\partial_x \vec{f} = \vec{e}_x \qquad \partial_y \vec{f} = \vec{e}_y \qquad \partial_z \vec{f} = \vec{e}_z$$

$$\nabla \vec{f} = \begin{pmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{pmatrix} \nabla \vec{f} \cdot \vec{v} = \qquad \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Diese Ableitungen sind nicht stetig. Alle Koordinaten ausser dem Nullvektor enden bei Anwendung der Ableitungen auf der Summe der Vektoren.

$$f'(\vec{x}) = \sum_{i=1}^{3} \frac{\partial f}{\partial x}(\vec{x}_i) = \sum_{i=1}^{3} \vec{e}_n$$

Die Ableitungen haben eine nette Eigenschaft. Zu der Jacobi-Matrix zusammengefasst, erhalten wir die Matrix, die in 3.3 besprochen wird.

$$\boldsymbol{J} := \begin{pmatrix} \partial_x f_1 & \partial_y f_1 & \partial_z f_1 \\ \partial_x f_2 & \partial_y f_2 & \partial_z f_2 \end{pmatrix} = \begin{pmatrix} r_x \cos\varphi_x & r_y \cos\varphi_y & r_z \cos\varphi_z \\ r_x \sin\varphi_x & r_y \sin\varphi_y & r_z \sin\varphi_z \end{pmatrix}$$

Jetzt folgt noch ein Beispiel. Eine Schreibweise, die von Mannigfaltigkeiten und Differentialgeometrie bekannt ist, kann man auch für diese Funktion benutzen.

$$\sum_{i=0}^{3} x^i \frac{\partial f}{\partial x^i} = \begin{pmatrix} x \\ y \end{pmatrix}$$

Hierbei bedeuten die Superskripte keine Potenz, sondern sind die Indizes der Koordinaten. In der Differentialgeometrie gibt es eine Konvention, Superskripte zu verwenden. Der Superskript im Nenner gilt übrigens als Subskript, da er im Nenner steht.

Integrale zu den partiellen Ableitungen habe ich auch definiert.

Version 1:

Um negative Koordinaten gleichsam korrekt zu integrieren muss das Vorzeichen entsprechend beachtet werden.

$$I_n(x) := \begin{cases} -\int_x^0 \vec{e}_n dx & \forall x < 0 \\ \int_0^x \vec{e}_n dx & \forall x \geq 0 \end{cases}$$

Die Integrale der partiellen Ableitungen setzen wir zu einer Integralfunktion zusammen.

$$I(\vec{v}) := I_x(\vec{v}_1) + I_y(\vec{v}_2) + I_z(\vec{v}_3)$$

Version 2:

Die $sign(x) = \pm 1$ mit $sign(0) = 0$ Funktion gibt das Vorzeichen als positiven oder negativen Einserfaktor zurück.

$$sign(x) := \begin{cases} -1 & \forall x < 0 \\ 0 & x = 0 \\ 1 & \forall x > 0 \end{cases}$$

Die Absolutwertfunktion $(x) := |x|$ gibt den positiven Betrag zurück. Mit $|-x| = x$ und $|x| = x$.

$$abs(x) := \begin{cases} -x & \forall x < 0 \\ \\ x & \forall x \geq 0 \end{cases}$$

Die Integrationsgrenzen muessen zwar nicht mehr getauscht werden und auch das Vorzeichen nicht gesetzt werden, dafür ist die sign Funktion ebenso zu rufen, wie der Absolutwert einzusetzen.

$$\begin{aligned} \hat{I}(x,y,z) := sign(x) \int_0^{|x|} \vec{e}_x dx + sign(y) \int_0^{|y|} \vec{e}_y dy + sign(z) \int_0^{|z|} \vec{e}_z dz \\ = \pm x\vec{e}_x \pm y\vec{e}_y \pm z\vec{e}_z \end{aligned}$$

Bemerkung. Ich nenne $\vec{f}(\vec{x})$ es (vektorwertiges) Funktional, da es mit der Theorie linearer Operatoren ebenso harmoniert, wie in der Form des Skalarprodukts (mehrzeilig) geschrieben zu sein. Die eindeutige Integraldarstellung $\int_a^b f(x)g(x)dx$ mit $f \in \mathbb{L}^p$ und $g \in \mathbb{L}^q$ mit $\frac{1}{p} + \frac{1}{q} = 1$ habe ich allerdings noch nicht untersucht.

3.3 Matrix

Eine $m \times n$ matrix ist ein Rechteck oder ein Quadrat aus Zahlen.

$$\boldsymbol{A} = (a_{ij})_{i,j \in \mathbb{N}^+} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

Eine Matrix mit einem Vektor multipliziert man so

$$\boldsymbol{A}\vec{v} = (\sum_{j=1}^{n} a_{ij}\vec{v}_j)_{i=1..m} = \begin{pmatrix} a_{11}v_1 + a_{12}v_2 + \dots + a_{1n}v_n \\ \vdots \\ a_{m1}v_1 + a_{m2}v_2 + \dots + a_{mn}v_n \end{pmatrix} = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} = \vec{w}$$

Die Multiplikation unserer Achsvektoren als Spaltenvektoren einer Matrix mit einem dreidimensionalen Vektor gibt einen zweidimensionalen Vektor zurück. Den richtigen Punkt.

$$\boldsymbol{A} := \begin{pmatrix} r_x \cos\varphi_x & r_y \cos\varphi_y & r_z \cos\varphi_z \\ r_x \sin\varphi_x & r_y \sin\varphi_y & r_z \sin\varphi_z \end{pmatrix}$$

Beweis:

$$\boldsymbol{A} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = (\sum_{j=1}^{3} a_{ij}\vec{x}_j)_{i=1..2} = \begin{pmatrix} r_x \cos(\varphi_x)x + r_y \cos(\varphi_y)y + r_z \cos(\varphi_z)z \\ r_x \sin(\varphi_x)x + r_y \sin(\varphi_y)y + r_z \sin(\varphi_z)z \end{pmatrix}$$

$$= x\vec{e}_x + y\vec{e}_y + z\vec{e}_z = \sum_n \vec{x}_n\vec{e}_n = \begin{pmatrix} x \\ y \end{pmatrix}$$

Bemerkung. Über die drei Werkzeuge sind mir bereits mehr Fakten bekannt und ich werde sie in der kommenden Version berücksichtigen und freilassen.

4 Transformationsverhalten

Die Transformation ist eine lineare Transformation und erfüllt damit die grundsätzlichen Bedingungen der linearen Funktionen.

$$f(x+y) = f(x) + f(y) \qquad \lambda f(x) = f(\lambda x)$$

4.1 Der Ursprung wird auf den Ursprung abgebildet

Der Nullvektor aus dem dreidimensionalen Raum $\vec{0} \in \mathbb{R}^3$ wird auf den Nullvektor im zweidimensionalen Raum abgebildet $\vec{0} \in \mathbb{R}^2$.

Beweis:

$$\boldsymbol{A} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0+0+0 \\ 0+0+0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Bemerkung. Der Kern von A, ker(A) ist nicht leer, d.h. er enthält nicht nur $\vec{0}$ sondern eine der Projektion entsprechende Gerade durch den Ursprung normal ($\perp$) zur Projektionsebene.

4.2 Punkte auf einer Achse

Punkte, die nur auf einer Achse liegen, sind ein Vielfaches des jeweiligen Achsvektors.

Beweis:

$$\boldsymbol{A} \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} ar_x \cos\varphi_x + 0 + 0 \\ ar_x \sin\varphi_x + 0 + 0 \end{pmatrix} = a\vec{e}_x$$

$$\boldsymbol{A} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 + r_y \cos\varphi_y + 0 \\ 0 + r_y \sin\varphi_y + 0 \end{pmatrix} = \vec{e}_y$$

$$\boldsymbol{A} \begin{pmatrix} 0 \\ 0 \\ -b \end{pmatrix} = \begin{pmatrix} 0 + 0 - br_z \cos\varphi_z \\ 0 + 0 - br_z \sin\varphi_z \end{pmatrix} = -b\vec{e}_z$$

4.3 Skalare Multiplikation

Es ist einfach zu zeigen, dass $\boldsymbol{A}(\lambda\vec{x}) = \lambda\boldsymbol{A}\vec{x}$. Man kann vor der Transformation oder nach der Transformation mit dem Skalar multiplizieren. Das Resultat ist identisch.

Beweis:

$$\begin{aligned}
\boldsymbol{A}(\lambda\vec{x}) &= \boldsymbol{A}\begin{pmatrix} \lambda x \\ \lambda y \\ \lambda z \end{pmatrix} \\
&= \begin{pmatrix} \lambda x r_x\cos(\varphi_x) + \lambda y r_y\cos(\varphi_y) + \lambda z r_z\cos(\varphi_z) \\ \lambda x r_x\sin(\varphi_x) + \lambda y r_y\sin(\varphi_y) + \lambda z r_z\sin(\varphi_z) \end{pmatrix} \\
&= \lambda\begin{pmatrix} x r_x\cos(\varphi_x) + y r_y\cos(\varphi_y) + z r_z\cos(\varphi_z) \\ x r_x\sin(\varphi_x) + y r_y\sin(\varphi_y) + z r_z\sin(\varphi_z) \end{pmatrix} \\
&= \lambda\begin{pmatrix} x' \\ y' \end{pmatrix} \\
&= \lambda\boldsymbol{A}\vec{x}
\end{aligned}$$

4.4 Addition und Subtraktion

Einfach zu zeigen ist auch, dass $\boldsymbol{A}(\vec{v} + \vec{w}) = \boldsymbol{A}\vec{v} + \boldsymbol{A}\vec{w}$. Man kann auch hier die Eingabevektoren vor der Umwandlung addieren, oder die Ausgabevektoren nach der Umwandlung. Die resultierende Summe ist identisch.

Beweis:

$$\begin{aligned}
\boldsymbol{A}\begin{pmatrix} x+u \\ y+v \\ z+w \end{pmatrix} &= \begin{pmatrix} (x+u)r_x\cos(\varphi_x) + (y+v)r_y\cos(\varphi_y) + (z+w)r_z\cos(\varphi_z) \\ (x+u)r_x\sin(\varphi_x) + (y+v)r_y\sin(\varphi_y) + (z+w)r_z\sin(\varphi_z) \end{pmatrix} \\
&= \begin{pmatrix} x r_x\cos(\varphi_x) + y r_y\cos(\varphi_y) + z r_z\cos(\varphi_z) \\ x r_x\sin(\varphi_x) + y r_y\sin(\varphi_y) + z r_z\sin(\varphi_z) \end{pmatrix} + \begin{pmatrix} u r_x\cos(\varphi_x) + v r_y\cos(\varphi_y) + w r_z\cos(\varphi_z) \\ u r_x\sin(\varphi_x) + v r_y\sin(\varphi_y) + w r_z\sin(\varphi_z) \end{pmatrix} \\
&= \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} u' \\ v' \end{pmatrix} \\
&= \boldsymbol{A}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \boldsymbol{A}\begin{pmatrix} u \\ v \\ w \end{pmatrix}
\end{aligned}$$

4.5 Linearität

Durch die letzten zwei Beweise ist es offensichtlich, dass die Transformation die Regeln der Linearität befolgt.

$$\boldsymbol{A}(\lambda\vec{v} + \kappa\vec{w}) = \lambda\boldsymbol{A}\vec{v} + \kappa\boldsymbol{A}\vec{w} = \lambda\begin{pmatrix} x' \\ y' \end{pmatrix} + \kappa\begin{pmatrix} u' \\ v' \end{pmatrix}$$

Auf den Beweis der Kombination beider Abschnitte verzichte ich.

5 Computer Implementierung

Getestet habe ich das Koordinatensystem auf dem PC unter Linux im Webbrowser auf dem Canvas2DRenderingContext. Ich habe dreidimensionale Punkte erzeugt und mit der Formel in zweidimensionale Punkte umgewandelt. Ich habe die Punkte so gewählt, dass sie in der richtigen Reihenfolge vorliegen, dass man sie nur mit einem lineTo(x,y) von Punkt A zu Punkt B verbinden muss.

Mit der Formel ist es zum Beispiel möglich die dreidimensionalen Kurven einer Funktion zweier Variablen $z = f(x, y)$ zu zeichnen. Man berechnet die Punkte auf dem normalen Wege. Man transformiert sie dann von drei Komponenten auf zwei Komponenten, indem man sie mit dem Koordinatensystem multipliziert und summiert, wie beschrieben, beziehungsweise, wie ich es im weiteren Text weiter beschreiben werde.

5.1 Generischer Code

Das folgende Beispiel ist der Pseudocode für alle Computersysteme.

```
x_ = x*r*cos(alpha) + y*r*cos(beta) + z*r*cos(gamma)
y_ = x*r*sin(alpha) + y*r*sin(beta) + z*r*sin(gamma)
```

Das sind die einzigen beiden Zeilen Code die man braucht.

5.2 JavaScript Implementierung

Das ist ein komplettes EcmaScript 6 snippet mit allen notwendigen Informationen.

```
let rad = (deg) => Math.PI/180*deg;
let r_x = 1, r_y = 1, r_z = 1;
let phi_x = rad(220), phi_y = rad(330), phi_z = rad(90);
let xAxisCos = r_x*Math.cos(phi_x),
    yAxisCos = r_y*Math.cos(phi_y),
    zAxisCos = r_z*Math.cos(phi_z),
    xAxisSin = r_x*Math.sin(phi_x),
    yAxisSin = r_y*Math.sin(phi_y),
    zAxisSin = r_z*Math.sin(phi_z);
let transform2d = ([x,y,z]) => [
    x*xAxisCos+ y*yAxisCos+ z*zAxisCos,
    x*xAxisSin+ y*yAxisSin+ z*zAxisSin];
let transform2dAll = (P) => P.map(transform2d);

let beispielPunkte = transform2dAll([[1,2,3], [3,4,5], [14,24,15]]);
```

Das ist die realistische Menge an Code für die komplette Transformation von 3-D nach 2-D.

6 Selbständigkeitserklärung

Ich habe dieses Koordinatensystem selbst entwickelt. Es ist keine Formel aus einem Buch oder einer Lehrveranstaltung. Ob es irgendwo eine identische Formel oder eine vergleichbare Definition gibt, ist mir nicht bekannt.

Ich habe den Inhalt des Dokuments aus eigenem Ermessen zusammengestellt. Ich habe mir Gedanken zum Thema gemacht und ausserdem Rechnungen mit Stift und Papier angefertigt. Ausserdem habe ich in Lehrbüchern und Veranstaltungen geblättert, um das Koordinatensystem und die definierten Variabeln und Operationen möglichst gut in die reelle Mathematik einzuordnen. Mir mögen Fehler unterlaufen sein, und auch Details entgangen sein. Für beides möchte ich mich entschuldigen.

7 Lizenz

Der produzierte Source Code, um das Koordinatensystem und einige Abbildungen zu zeigen, ist frei für alle, wie auch das Koordinatensystem selbst und die dazugehörigen Definitionen, die ich selbst angefertigt habe. Es ist erlaubt, mir dafür Anerkennung zu gewähren, es ist aber nicht zwingend nötig, mich dafür im eigenen Projekt zu nennen. Allerdings mag auch ich keine Menschen, die diese Arbeit für ihre eigene ausgeben.

References

Corral1 *Michael Corral, Schoolcraft College*, Vector Calculus, GNU Free Documentation License, http://mecmath.net

[1] [1] *Michael Corral, Schoolcraft College*, Trigonometry, GNU Free Documentation License, http://mecmath.net

[2] *Gilbert Strang, MIT*, Linear Algebra and its Applications. Fourth Edition.

[3] *Gilbert Strang, MIT*, Calculus. MIT OpenCourseWare Supplemental Resources. http://ocw.mit.edu

[4] *John Rogues, Lecture Notes on Topology, following J.R.Munkres Textbook, for MAT3500/4500*, Lecture Script, Topology (english), http://

[5] *Roman Vershynin. Lectures in Functional Analysis. Department of Mathematics, University of Michigan*, Lecture Script, http://,

[6] *Dirk Ferus, TU-Berlin, em.*, Lecture Script, Lineare Algebra 1+2, 2000, http://page.math.tu-berlin/ ferus/skripten.html

[7] *Franziska Kühn, Technische Universität Dresden*, Lecture Script, Lineare Algebra und analytische Geometrie I+II, http://fkuehn.de/download/LAAG.pdf

[8] *Petra Wittbold, TU-berlin*, Lecture Script, Funktionalanalysis I, http://www3.math.tu-berlin.de/Vorlesungen/SS09/FA1/Doc/Funkana1-SS06-08.06.09.pd

[9] *Michael Corral, Schoolcraft College*, Latex Mini Tutorial, http://mecmath.net

[10] *Manuela Jürgens, Thomas Feuerstack, Fernuniversität Hagen*, LaTeX, eine Einführung und ein bisschen mehr..., a026_latex_einf.pdf

[11] *Dr.Jan Rudl, Technische Universität Dresden, Fachbereich Mathematik*, Einführung in LaTeX, LaTeX-Kurs.pdf