Transformation dreidimensionaler Koordinaten in zweidimensionale Koordinaten

Transformation dreidimensionaler Koordinaten in zweidimensionale Koordinaten (German Version)

Edward Gerhold

August 26, 2015

German Version 0.0.1 of 0.3.24

Bemerkung. Das ist eine Entwicklungsversion. Und hat chaotische Stellen. Und (nur temporär einige logische mathematische Fehler, welche in jedem Fall von mir korrigiert werden.) Diese Datei enthält Typos, ungewollte logische Fehler und Verrechnungen, und vielleicht versehentliche Buchstaben, die von einem pltzlich erscheinenden Doppelcursor im Editor kommen. Das ist mein erstes LaTeX, aber nicht mein letztes. Genaugenommen, ist es die deutsche Übersetzung meines ersten. Falls sie das jemals drucken sollten, dann 4 auf 1. Das spart Papier. Das Dokument ist unfertig und nicht komplett geformt.

Contents

1 Einleitung

Auf einem Blatt Papier sehen sie drei Koordinatenachsen in drei Richtungen in den Raum zeigen. In der Realität sind diese Vektoren zweidimensional. Denn sie zeigen in drei Richtungen auf dem Papier, und nicht in den reellen Raum.

In diesem Dokument entwerfen wir eine $\mathbb{R}^{2\times 3}$ Basis für die Koordinatentransformation. Eine Basis multipliziert man mit den Werten der Koordinaten, um sich für jede Komponente ein Stück zu bewegen, um die Bewegung auf dem korrekten neuen Punkt zu beenden. Im Fall der Kosinus und Sinus Funktionen, bewegen wir uns nach links und rechts, und nach oben und unten, um ihnen direkt mitzuteilen, was passiert, wenn wir die Koordinaten mit der Matrix multiplizieren.

Was wir in diesem Dokument machen werden.

1. Winkel auswählen für unserer Koordinatenachsen rund um
2. Choose angles for our coordinate axes around the unit circle to lay out three axes.
3. Write down the basis vectors for each coordinate axis
4. Assemble a matrix with the vector basis for a point by point transformation.
5. Read the example source code for a computer function, which is exactly two lines long. One for the new x and one for the new y.

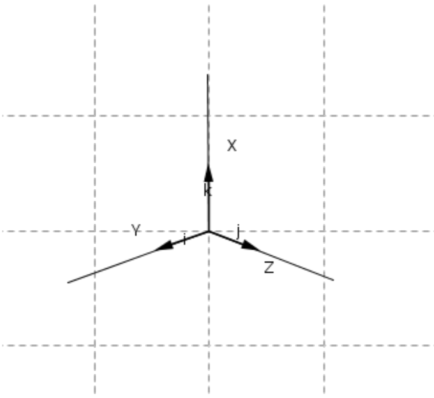

Figure 1: Bild eines rechtshändigen Koordinatensystems mit ijk-Basisvektoren auf den Achsen, in drei dimensionen zeigend. Schauen sie in (1) für eine Einführung.

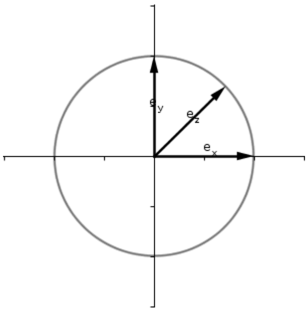

Figure 2: Ein linkshändiges Koordinatensystem

6. Read other versions of the transformation, with functions, for example.
7. Derive the generic case of transforming coordinate systems down to the plane.

2 Entwurf eines $\mathbb{R}^{2\times 3}$ 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.

3 Definitionen

3.1 Koordinatensystem

3.1.1 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.

3.1.2 Einheitskreis

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

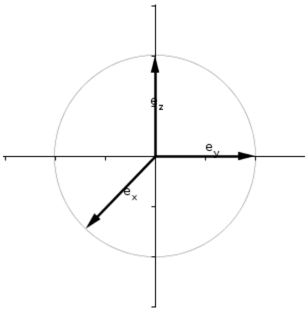

Figure 3: Ein rechtshändiges Koordinatensystem

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

(x, y) stellen die Spitze eines Ortsvektors aus Zentrum des Koordinatensystems dar. Wenn wir drei Achsen haben wollen, sollten wir uns drei solche Vektoren erzeugen. Das werden wir auch gleich tun.

3.2 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.

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

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

3.3 Einheit der Achsen

3.3.1 Der r-Wert der Achsen

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

3.3.2 Mathematische Vorsicht

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

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 unrealistisch aus, ist aber bei Standbildern kein Problem.

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.

3.4 (Basis)vektoren der Linearkombination

Wir benötigen für das Koordinatensystem drei Achsen. Jede Achse bekommt einen kanonischen Einheitsvektor.

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

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.

4 Transformationswerkzeuge

In diesem Kapitel stelle ich dann vor, wie man die Transformation durchführt.

4.1 Matrix

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

4.2 Funktional

() := x

5 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.

6 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

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

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

Gilbert Strang, MIT, Linear Algebra and its Applications. Fourth Edition.

Gilbert Strang, MIT, Calculus. MIT OpenCourseWare Supplemental Resources. http://ocw.mit.edu

, , Lecture Script, Topology (english),

, Lecture Script, Functional Analysis (english),

TODO, Lecture Script,

TODO, Lecture Script,

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

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

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

Michael Corral, Schoolcraft College, Latex Mini Tutorial, http://mecmath.net

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

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