Herleitung der perspektivischen Projektionsmatrix

– Peter Hofmann, Februar 2015 –

Inhaltsverzeichnis

1 Koordinatensystem, Blickrichtung, Objektpositionen

Wir verwenden ein rechtshändiges Koordinatensystem wie in Abbildung 1. Die Kamera wird im Ursprung platziert und blickt in *negative* z-Richtung, damit die positive x-Richtung weiterhin nach „rechts“ und positive y-Richtung nach „oben“ zeigt. Objekte werden daher ebenfalls mit *negativen* z-Koordinaten platziert.

2 Herleitung der Projektionsmatrix mit Kegelstumpf

Wir wollen eine Zentralprojektion erreichen mit einem Fluchtpunkt. Ähnlich der Konventionen bei OpenGL verwenden wir dafür einen Kegelstumpf, den wir in einen Einheitswürfel transformieren.

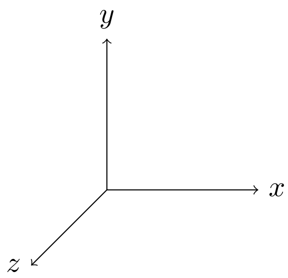

Abbildung 1: Rechtshändiges Koordinatensystem.

2.1 Grundlegende Projektionsmatrix: Zentralprojektion

In z-Richtung wird dieser Kegelstumpf durch die beiden Werte n und f für „near" und „far" begrenzt. Beide Werte werden negativ sein und es gilt $n > f$ (n ist also näher an der Null). Bei n wird unsere Bildebene liegen. Alle Objekte werden in z-Richtung zwischen n und f erwartet. Ein Beispiel für diesen Aufbau ist in Abbildung 2 zu sehen.

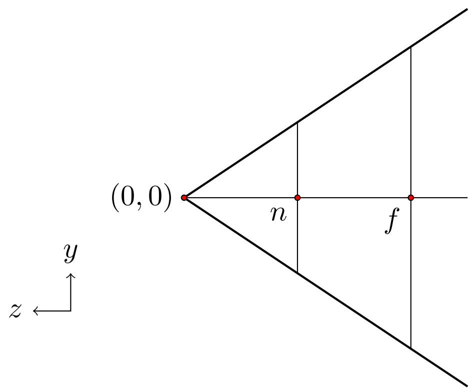

Abbildung 2: Querschnitt des Kegelstumpfs mit n und f.

Wird nun ein Punkt $\mathbf{p}$ projiziert, dann suchen wir den dazugehörigen Punkt $\mathbf{p}'$, der sich durch die Zentralprojektion ergibt. In Abbildung 3 kann man erkennen, dass sich hier der zweite Strahlensatz anwenden lässt:

$$\frac{\mathbf{p}_y}{\mathbf{p}_z} = \frac{\mathbf{p}'_y}{n}$$

$$\iff \quad \mathbf{p}'_y = \mathbf{p}_y \cdot \frac{n}{\mathbf{p}_z}$$

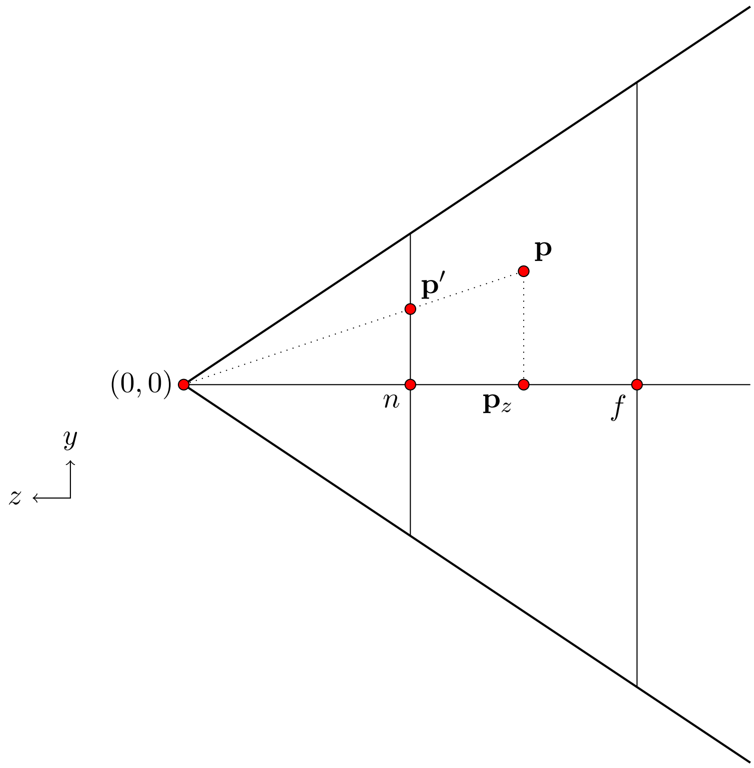

Abbildung 3: Gegebener Punkt $\mathbf{p}$ und gesuchter Punkt $\mathbf{p}'$.

Für die nicht eingezeichnete x-Richtung gilt dasselbe:

$$\frac{\mathbf{p}_x}{\mathbf{p}_z} = \frac{\mathbf{p}'_x}{n}$$

$$\iff \quad \mathbf{p}'_x = \mathbf{p}_x \cdot \frac{n}{\mathbf{p}_z}$$

Daraus ergibt sich unter Verwendung von homogenen Koordinaten vorübergehend die Matrix M_1:

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{n} & 0 \end{pmatrix}$$

Beispielhafte Multiplikation:

$$M_1 \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \frac{z}{n} \end{pmatrix} \cong \begin{pmatrix} x\frac{n}{z} \\ y\frac{n}{z} \\ n \end{pmatrix}$$

Wie gewünscht wurden x und y also jeweils mit dem Faktor $\frac{n}{z}$ multipliziert.

2.2 Transformation in den Einheitswürfel

2.2.1 Definition des Bildbereichs

Bisher wurde lediglich eine Zentralprojektion erreicht. Das heißt, dass jeder Punkt im Raum auf unsere Bildebene projiziert werden kann. Eine Ebene ist unendlich groß, unser Bildschirm aber endlich. Daher wollen wir nun auf der Bildebene ein Rechteck für den tatsächlichen Bildbereich definieren. Erst dadurch entsteht auch tatsächlich ein Kegelstumpf.

In x-Richtung wird der Bildbereich durch l und r für „left" und „right" beschränkt, in y-Richtung durch b und t für „bottom" und „top".

In den allermeisten Fällen wird dieses Rechteck im Ursprung zentriert sein. Wir betrachten jedoch zunächst den allgemeinen Fall. Für den genannten Sonderfall bietet sich ohnehin die Vereinfachung an, die im nächsten Abschnitt beschrieben wird.

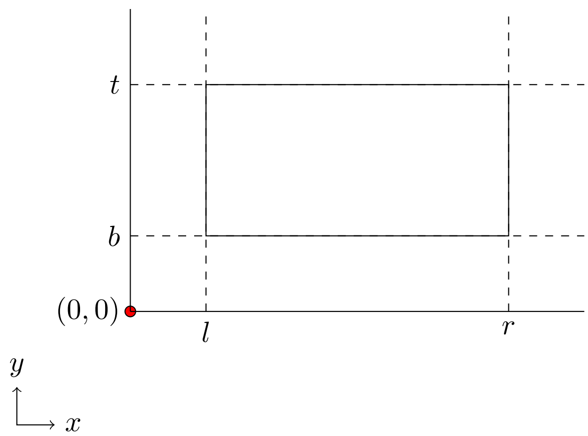

Abbildung 4: Bildbereich auf der Projektionsebene.

2.2.2 Bestimmung der Einträge für x und y

Sei zur Steigerung der Lesbarkeit nun y die y-Koordinate des Ausgangspunkts $\mathbf{p}$ und y' die y-Koordinate des auf die Bildebene projizierten Punkts $\mathbf{p}'$. Wir möchten diejenigen Punkte in das Intervall $[-1, 1]$ überführen, die innerhalb des rechteckigen Bildbereichs liegen, also suchen wir das skalierte y'':

$$y' \in [b, t] \quad \to \quad y'' \in [-1, 1]$$

Wir wissen, dass es eine lineare Abbildung ist und dass b auf -1 abgebildet wird und t auf 1. Daher können wir das folgende LGS aufstellen:

$$-1 = \alpha \cdot b + \beta \ , \tag{1}$$

$$1 = \alpha \cdot t + \beta \tag{2}$$

Aus (2) folgt direkt:

$$\beta = 1 - \alpha \cdot t \tag{3}$$

Setze (3) in (1) ein:

$$\begin{aligned} -1 &= \alpha \cdot b + 1 - \alpha \cdot t \\ &= \alpha \cdot (b - t) + 1 \\ \Rightarrow \quad \alpha &= \frac{-2}{b - t} \end{aligned} \tag{4}$$

Dann (4) in (3):

$$\begin{aligned} \beta &= 1 - \frac{-2}{b-t} \cdot t \\ &= \frac{b-t}{b-t} - \frac{-2 \cdot t}{b-t} \\ &= \frac{b - t + 2 \cdot t}{b-t} \\ &= \frac{b+t}{b-t} \end{aligned}$$

Das LGS ist gelöst und insgesamt ergibt sich:

$$y'' = \frac{-2}{b-t} \cdot y' + \frac{b+t}{b-t}$$

Der letzte Schritt ist, $y' = y \cdot \frac{n}{z}$ einzusetzen und die Gleichung dann in eine für die Matrix geeignete Form zu bringen. Das heißt, es muss eine Summe aus x, y und z sein, wobei diese Werte noch einfache Koeffizienten haben dürfen, und die Summe kann am Ende eventuell noch durch einen Wert geteilt werden (aufgrund der homogenen Koordinaten).[1]

[1] Es gibt hier häufig die Konvention, dass durch $-z$ geteilt werden soll. Das machen wir an dieser Stelle *nicht*.

Das ist Schritt für Schritt:

$$\begin{aligned}
y'' &= \frac{-2}{b-t} \cdot \left(y \cdot \frac{n}{z}\right) + \frac{b+t}{b-t} \\
&= \frac{-2}{b-t} \cdot y \cdot \frac{n}{z} + \frac{b+t}{b-t} \cdot \frac{z}{z} \\
&= \left(\frac{-2 \cdot n}{b-t} \cdot y\right) \cdot \frac{1}{z} + \left(\frac{b+t}{b-t} \cdot z\right) \cdot \frac{1}{z} \\
&= \left(\frac{-2 \cdot n}{b-t} \cdot y + \frac{b+t}{b-t} \cdot z\right) \cdot \frac{1}{z} \qquad (5)
\end{aligned}$$

Gleichung (5) genügt unseren Anforderungen.

Dieser Prozess lässt sich analog auch für x' beziehungsweise x'' durchführen. Hier ist die Forderung:

$$x' \in [l, r] \quad \rightarrow \quad x'' \in [-1, 1]$$

Das Ergebnis ist genau dasselbe, nur l statt b und r statt t.

Damit ist unsere Matrix vorläufig diese:

$$M_2 = \begin{pmatrix} \frac{-2n}{l-r} & 0 & \frac{l+r}{l-r} & 0 \\ 0 & \frac{-2n}{b-t} & \frac{b+t}{b-t} & 0 \\ ? & ? & ? & ? \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

2.2.3 Bestimmung der Einträge für z

Die dritte Zeile ist bisher unbestimmt, weil wir uns noch nicht darum gekümmert haben, wie z abgebildet werden soll. Das werden wir nun nachholen.

z wird nicht projiziert, also ist $z' = z$. Das einzige, was hier passieren soll, ist eine Transformation von $z' \in [n, f]$ nach $z'' \in [-1, 1]$, damit wir in der Gesamtheit einen Einheitswürfel erreichen. Wir wissen, dass dafür weder x noch y relevant sind, also sind $m_{3,1} = m_{3,2} = 0$. Die beiden anderen Einträge sind noch offen.

Betrachten wir noch einmal die Multiplikation mit einem homogenen Vektor und legen den Fokus auf das Endergebnis der z-Komponente nach Dehomogenisierung:

$$M_2 \cdot \mathbf{p} = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cdots \\ \cdots \\ z \\ 1 \end{pmatrix},$$

$$\text{also ist:} \quad z'' = (\alpha \cdot z + \beta \cdot 1) \cdot \frac{1}{z}$$

$$= \alpha + \frac{\beta}{z}$$

Wir gehen nun wie vorhin den Weg über ein LGS. Wir möchten n auf -1 abbilden und f auf 1. Also:

$$-1 = \alpha + \frac{\beta}{n}\ , \tag{6}$$

$$1 = \alpha + \frac{\beta}{f} \tag{7}$$

Aus (7) ergibt sich direkt:

$$\alpha = 1 - \frac{\beta}{f} \tag{8}$$

Setze nun (8) in (6) ein:

$$-1 = 1 - \frac{\beta}{f} + \frac{\beta}{n}$$

$$= 1 - \frac{n \cdot \beta}{n \cdot f} + \frac{f \cdot \beta}{f \cdot n}$$

$$= 1 + \frac{\beta \cdot (f - n)}{n \cdot f}$$

$$\iff \quad -2 = \beta \cdot \frac{f - n}{n \cdot f}$$

$$\iff \quad \beta = \frac{-2 \cdot n \cdot f}{f - n} \tag{9}$$

Setze letztendlich (9) in (8) ein:

$$\begin{aligned}\alpha &= 1 - \frac{2 \cdot n \cdot f}{f - n} \cdot \frac{1}{f} \\ &= 1 + \frac{2 \cdot n}{f - n} \\ &= \frac{f - n}{f - n} + \frac{2 \cdot n}{f - n} \\ &= \frac{f - n + 2 \cdot n}{f - n} \\ &= \frac{f + n}{f - n}\end{aligned}$$

Damit haben wir final M_3 bestimmt:

$$M_3 = \begin{pmatrix} \frac{-2n}{l-r} & 0 & \frac{l+r}{l-r} & 0 \\ 0 & \frac{-2n}{b-t} & \frac{b+t}{b-t} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & \frac{-2nf}{f-n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

3 Vereinfachung: Field of View ϑ und Aspect Ratio a

Sehr häufig genügt es, den Bildbereich in der Projektionsebene zu zentrieren. Damit blickt die Kamera genau entlang der negativen z-Achse, ebenso herrscht Symmetrie und es ist immer $b = -t$ und $l = -r$. Man muss die Kamera dann auch nicht mehr über l, r, b und t parametrisieren, sondern kann einfach einen Öffnungswinkel ϑ angeben. Möchte man keinen quadratischen Bildbereich, dann kann man das Seitenverhältnis a zusätzlich angeben.

n und f werden weiterhin angegeben.

3.1 Field of View ϑ

Wir projizieren initial wie gehabt $y' = y \cdot \frac{n}{z}$. Da die Division zu einem anderen „Zeitpunkt“ passiert als die Multiplikation mit n (nämlich erst bei der Dehomogenisierung und nicht direkt als Koeffizient für y), schreiben wir explizit:

$$y' = (y \cdot n) \cdot \frac{1}{z}$$

Dann betrachten wir Abbildung 5: t ist nicht gegeben, wir können es aber über den Öffnungswinkel bestimmen. Relevant ist dazu nur das obere Dreieck mit dem eingezeichneten rechten Winkel.

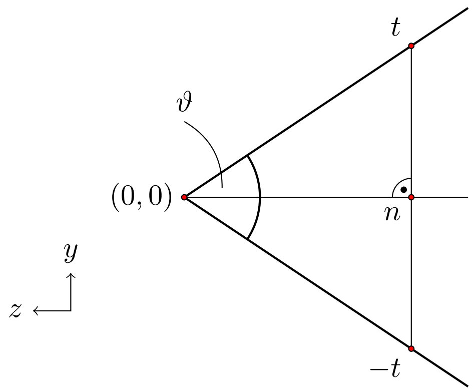

Abbildung 5: Kamera mit n und Öffnungswinkel ϑ. Die zweite Ebene bei f befindet sich weiterhin rechts von n und ist nicht eingezeichnet.

In diesem Dreieck gilt:

$$\tan\frac{\vartheta}{2} = \frac{t}{n}$$

$$\Rightarrow \quad t = n \cdot \tan\frac{\vartheta}{2}$$

Ähnlich wie bei der allgemeinen Projektion haben wir damit die Forderung:

$$y' \in [-n \cdot \tan\frac{\vartheta}{2}, n \cdot \tan\frac{\vartheta}{2}] \quad \rightarrow \quad y'' \in [-1,1]$$

Wir können jetzt ausnutzen, dass der Bildbereich symmetrisch ist. Dadurch müssen wir lediglich durch die Intervallgrenzen dividieren. Also:

$$\begin{aligned} y'' &= y' \cdot \frac{1}{n \cdot \tan\frac{\vartheta}{2}} \\ &= y \cdot n \cdot \frac{1}{n \cdot \tan\frac{\vartheta}{2}} \cdot \frac{1}{z} \\ &= y \cdot \frac{1}{\tan\frac{\vartheta}{2}} \cdot \frac{1}{z} \end{aligned}$$

Man beachte, dass dies ein reiner Koeffizient für y ist und nichts addiert wird. Das wird die Matrix weiter vereinfachen.

3.2 Aspect Ratio a

Die x-Richtung lässt sich prinzipiell analog zur y-Richtung bestimmen. Unter Umständen will man aber keinen quadratischen Bildbereich darstellen. Es genügt hier, das Seitenverhältnis von Breite zu Höhe zu betrachten:

$$a = \frac{w}{h}$$

Um zu einem rechteckigen Bildbereich zu gelangen, werden wir y'' nicht weiter ändern, sondern nur x''. Das heißt, der Öffnungswinkel ϑ steuert nur die Öffnung der Kamera in y-Richtung, wohingegen der Winkel in x-Richtung größer oder kleiner sein kann.

Wie verwerten wir nun a? Ganz anschaulich: Wir befinden uns mit einem positiven $x'' = k \cdot r$ irgendwo auf der Strecke von 0 bis r oder darüberhinaus.[2] Für y ist klar, dass bei $k = 1$ das Ende des Bildbereichs erreicht ist. Ist der Bildbereich nun aber beispielsweise doppelt so breit wie hoch, dann kann x'' auch doppelt so weit laufen. Um es auch in den Bereich $[-1, 1]$ zu skalieren, muss demnach durch a dividiert werden.

3.3 Endergebnis der vereinfachten Projektionsmatrix

x und y haben wir wie eben dargestellt abgeändert, z bleibt gleich, ebenso wie die letzte Zeile der Matrix. Damit ergibt sich insgesamt:

$$M_4 = \begin{pmatrix} \frac{1}{\tan\frac{\vartheta}{2}}\frac{1}{a} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\frac{\vartheta}{2}} & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & \frac{-2nf}{f-n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

4 Hintergrund: Warum spielt z'' nach der Projektion überhaupt eine Rolle?

Hätte man einzig und alleine eine Matrix für die Kamera, dann wäre z'' nicht weiter wichtig. Man könnte damit sogar immer noch einen Depth Buffer umsetzen, indem man sich die z-Koordinaten der projizierten Punkte ansieht. In der Regel gibt es jedoch neben der Kameramatrix noch viele weitere Transformationsmatrizen. Diese Matrizen werden

[2] r sei wie t bestimmt und hat auch denselben Wert.

in der Regel auch nicht einzeln gespeichert, sondern miteinander multipliziert, um eine gemeinsame Matrix zu erhalten, mit der alle Punkte multipliziert werden können – für die Umrechnung eines Punktes von Weltkoordinaten in Bildkoordinaten bedarf es dann nur einer einzigen Matrix-Vektor-Multiplikation. In dieser Matrix ist auch die Kameramatrix impliziert enthalten. Um jetzt noch einen Depth Buffer umsetzen zu können, muss das final projizierte z'' noch Aussagekraft besitzen können: Wo in z-Richtung lag der projizierte Punkt ursprünglich?