

Computergrafik

Die Mathematik hinter OpenGL, DirectX, etc.

Die Szene im NDC-Würfel



Control (Remedy Entertainment)

Inhalt

1	Einleitung ▲	2
2	Arten der Projektion	3
3	Perspektivische Projektion.....	5
4	Beispiel (Fortsetzung)	13
5	Die [MVP]-Matrix (Model-View-Projection-Matrix)	15

1 Einleitung

IM VIERTEN SCHRITT DER PIPELINE WIRD DIE PROJEKTION AUF DEN BILDSCHIRM VORBEREITET.
(Clipping, Art der Projektion, Normierung)

CLIPPING

Der **VIEW-SPACE** umfasst noch die gesamte virtuelle Szene. Also auch die Objekte, die sich z.B. hinter der Kamera befinden.

Nur das, was im Moment auf den Bildschirm erscheinen soll, soll weiterverarbeitet werden. Der Rest wird „weggeschnitten“. Dieser Vorgang heißt „Clipping“.

Dazu gehört, dass man ganz Nahes ebenso wenig berücksichtigen muss, wie allzu Fernes, was in Nebel oder Dunst verschwindet. Man kann auch in der Realität nicht beliebig weit sehen.

Ebenso gibt es von links nach rechts, sowie von unten nach oben einen eingeschränkten Sichtbereich, den „Field of View“ (FOV).

Der so begrenzte Raum ist ein **Quader** oder eine **Frustum** (Pyramidenstumpf). Durch die Projektion der Objekte in die „Nah-Ebene (Near-Clipping-Plane)“ entsteht danach eine 2D-Ansicht des 3D- **VIEW-SPACE**.

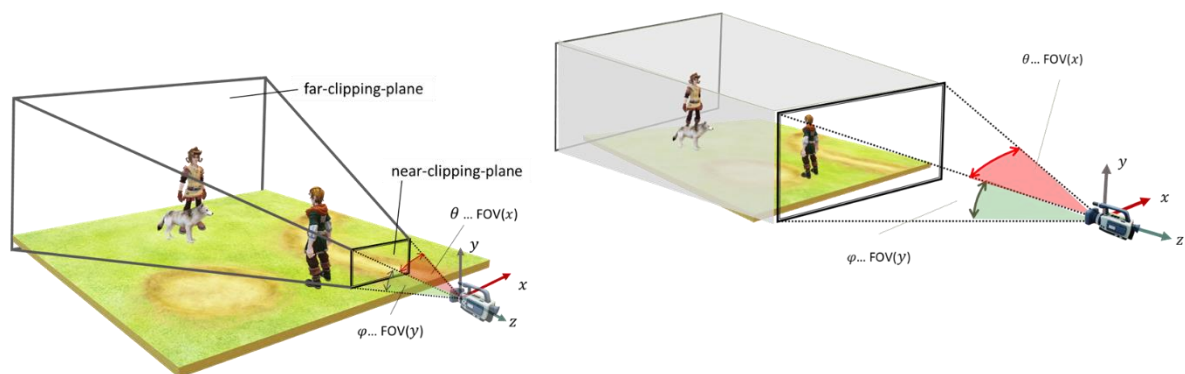


Abb. 1: Projektionsfrustrum bzw. -quader

PERSPEKTIVE

Für eine realitätsnahe Ansicht bedarf es einer perspektivischen Darstellung. Die Perspektive lässt weiter entfernte Gegenstände kleiner erscheinen. Sie entsteht erst durch die Projektion des **VIEW-SPACE** auf die „Nah-Ebene“.

NORMIERUNG

Die anschließende Normierung komprimiert die Szene in einen kleinen $2 \times 2 \times 2$ großen Würfel, dem sogenannten **PROJECTION-SPACE** in NDC-Koordinaten. NDC steht für „*normalized digital coordinates*“. Aus dem **PROJECTION-SPACE** heraus kann dann für jedes Display individuell die 3D-Szene auf den 2D-Screen des Displays skaliert und abgebildet werden.

2 Arten der Projektion

Eine Projektion ist eine Abbildung von einem höherdimensionalen Raum in einen Raum niedrigerer Dimension. In unserem Fall von 3D nach 2D.

Es gibt verschiedene Arten von Projektionen. Im Wesentlichen können zwei Klassen unterschieden werden:

Parallelprojektion (orthographische Projektion)

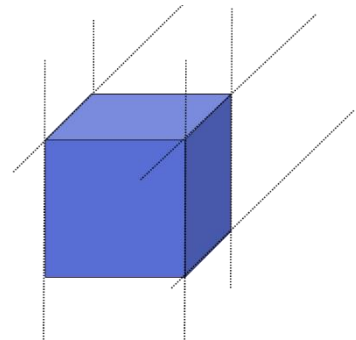


Abb. 2: Orthographische Projektion

Die Orthographische Projektion vermittelt einen räumlichen Eindruck, ignoriert jedoch die Eigenschaft unseres Auges, entferntere Dinge auf der Netzhaut kleiner abzubilden. Ihr Clipping-Raum ist der *Sicht-Quader*, „Far-Clipping-Plane“ und „Near-Clipping-Plane“ sind gleich groß.

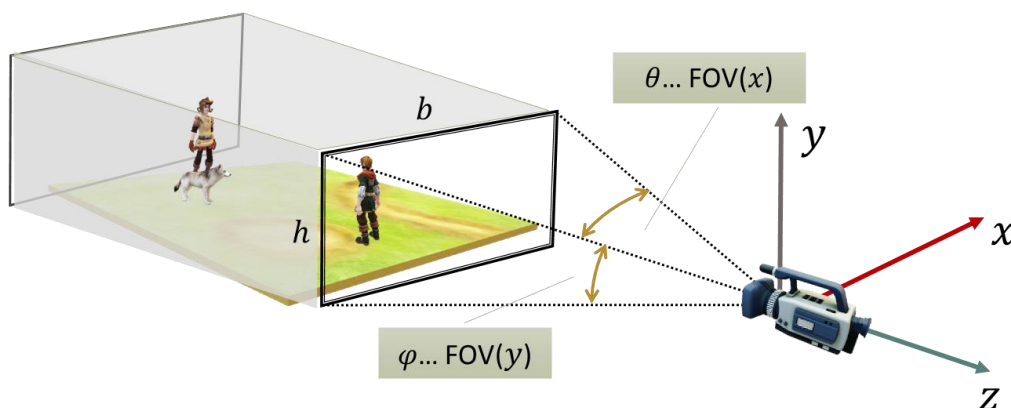


Abb. 3

Durch die Parallelprojektion der Objekte in die „Near-Clipping-Plane“ entsteht ein 2D-Abbild des **VIEW-SPACE** ohne Perspektive. Sie eignet sich für Konstruktionszeichnungen und Szenerien mit wenig Tiefe.

Zentralprojektion (perspektivische Projektion)

Die Zentralprojektion stellt entferntere Objekte kleiner dar. Das betrifft sowohl die z -Richtung (Tiefe), als auch x (links-rechts) und y (oben-unten). Ihr Clipping-Raum ist das *Sicht-Frustum*.

4

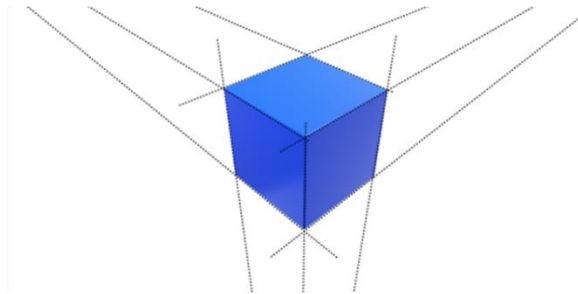


Abb. 4: Perspektivische Zentralprojektion

In der Computergrafik wird oft eine vereinfachte Version verwirklicht, die die Objekte lediglich in Abhängigkeit der z -Koordinate verkleinert. Linien parallel zu x - und y -Achse bleiben parallel. Bei Vermeidung von Weitwinkeldarstellungen und bei einigermaßen horizontaler Einstellung der virtuellen Kamera, fällt der Unterschied nicht auf.

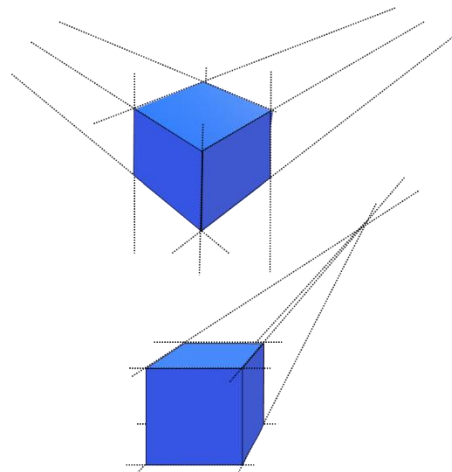


Abb. 5: Vereinfachte Zentralprojektion

3 Perspektivische Projektion

Im **WORLD-** und im **VIEW-SPACE** hat ein Würfel mit Kantenlänge 10 in der Nähe der Kamera die gleiche Größe wie in der Ferne.

$$\{W_{nah}\} = \left\{ \begin{pmatrix} 20,10,-30 \\ 20,10,-40 \end{pmatrix} \begin{pmatrix} 30,10,-30 \\ 30,10,-40 \end{pmatrix} \begin{pmatrix} 30,20,-30 \\ 30,20,-40 \end{pmatrix} \begin{pmatrix} 20,20,-30 \\ 20,20,-40 \end{pmatrix} \right\} \quad \text{Volumen: 1000}$$

$$\{W_{fern}\} = \left\{ \begin{pmatrix} 20,10,-90 \\ 20,10,-100 \end{pmatrix} \begin{pmatrix} 30,10,-90 \\ 30,10,-100 \end{pmatrix} \begin{pmatrix} 30,20,-90 \\ 30,20,-100 \end{pmatrix} \begin{pmatrix} 20,20,-90 \\ 20,20,-100 \end{pmatrix} \right\} \quad \text{Volumen: 1000}$$

Durch die Zentralprojektion der Objekte in die „Nah-Ebene“ entsteht jetzt eine perspektivische 2D-Ansicht, die ferne Objekte kleiner dargestellt als nahe.

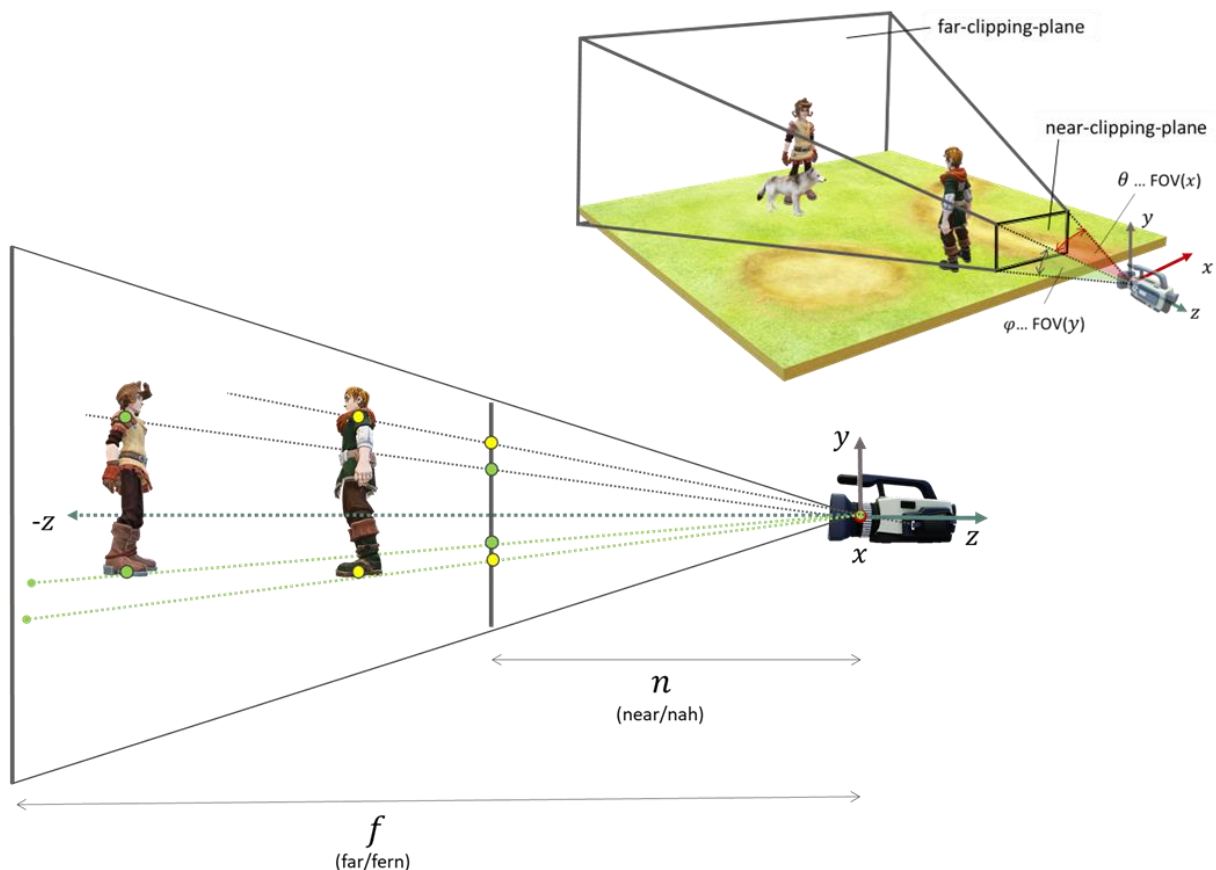


Abb. 6: Projektion in die Nah-Ebene. Seitliche Darstellung aus $(-x)$ -Richtung.

Es sei nochmals darauf hingewiesen, dass die Kamera in Richtung negativer z -Achse ausgerichtet ist. Die z -Koordinate aller Vertices im Sicht-Frustum sind somit negativ ($z < 0$, daher $(-z) > 0$). Die Werte n und f hingegen sind lediglich Abstände und hier positiv ($n, f > 0$).


$$\Rightarrow y_{proj} = \frac{n \cdot y}{(-z)} \quad \text{und analog} \quad x_{proj} = \frac{n \cdot x}{(-z)}$$

Was bei der Projektion von 3D in 2D allerdings abhandenkommt, ist die Tiefe. Damit geht aber auch die Information verloren, welche Objekte andere ganz oder teilweise verdecken, weil sie sich vor diesen befinden. Aus diesem Grund lassen wir die z-Werte vorerst nicht weg. (Abb. 8)

$$z_{proj} = z$$

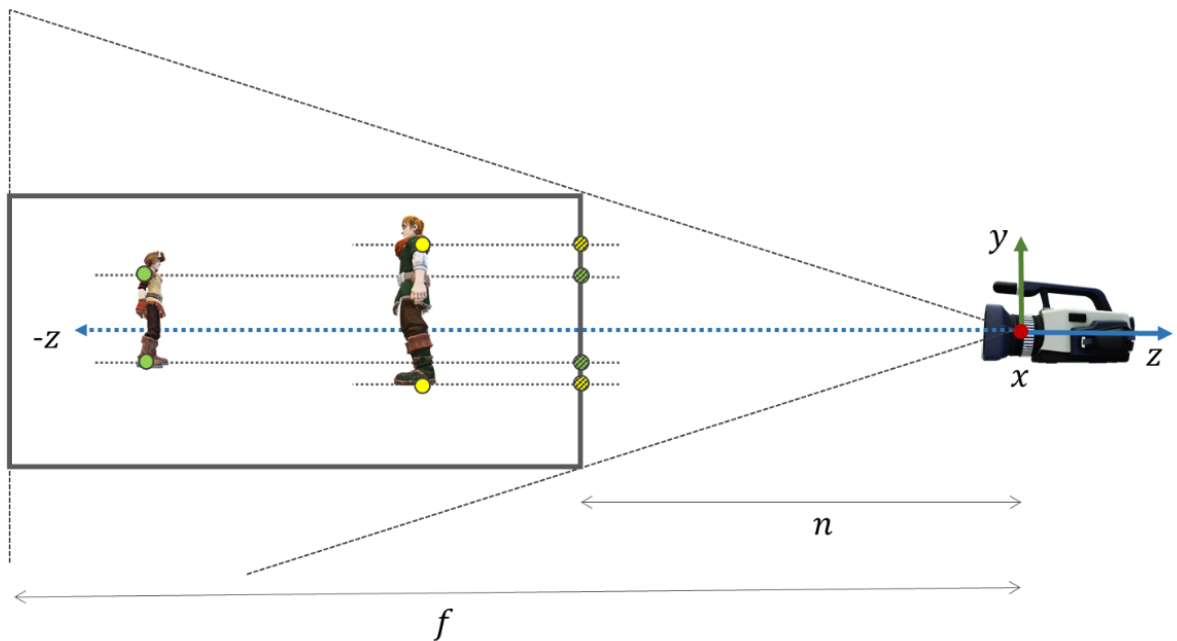


Abb. 8: Aus dem **Sicht-Frustum** (Pyramidenstumpf) ist ein **Sicht-Quader** geworden.

Die Normierung

In einem letzten Schritt wird dieser Quader aus Gründen der Standardisierung zu einem Kubus mit den Maßen $2 \times 2 \times 2$ (Abb. 9) komprimiert und befindet sich (gedacht „in der Kamera“) im Zentrum des Koordinatensystems. (PROJECTION-SPACE)

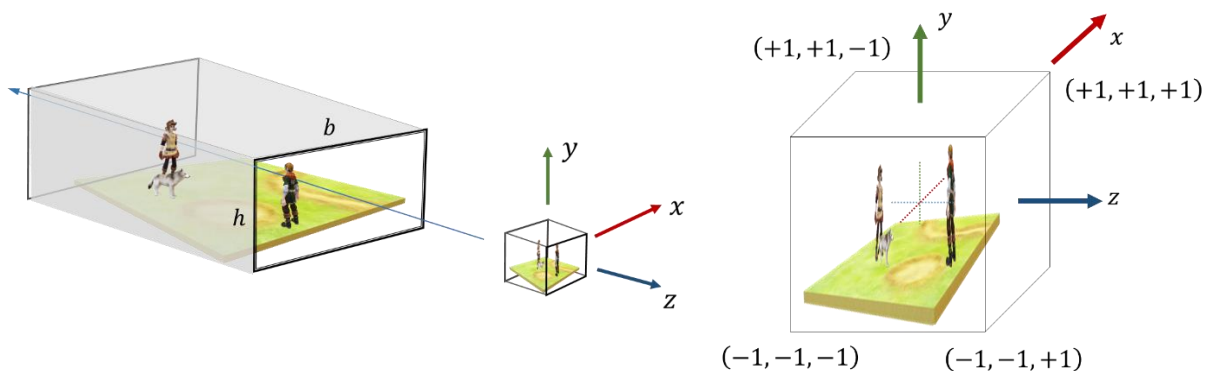


Abb. 9

DIE TRANSFORMATIONSPARAMETER (NORMIERTER PROJECTION-SPACE)

Die Koordinaten im PROJECTION-SPACE heißen normalisierte Gerätekoordinaten („normalized digital coordinates“), kurz **NDC**. Der Koordinatenursprung der **NDC**-Koordinaten liegt im Zentrum des Würfels und die Koordinaten liegen alle im Bereich $[-1, +1]$.

Die NDC-Koordinaten eines Punktes drücken somit die Position in Prozenten aus.

BEISPIEL

$P = (-0,7 \ 0,3 \ 0,5)$ bedeutet 70% links, 30% oben und 50% vorne.

Die Transformation der Koordinaten $x \rightarrow x_{proj} \rightarrow x_{NDC}$ und $y \rightarrow y_{proj} \rightarrow y_{NDC}$ ist eine reine Skalierung (λ, μ), während für $z = z_{proj} \rightarrow z_{NDC}$ skaliert und zum Koordinatenursprung verschoben wird (α, β).

Die Gleichungen für x_{NDC} und y_{NDC} folgen aus der Abbildung.

$$x_{proj} : x_{NDC} = b : 2 \text{ und } y_{proj} : y_{NDC} = h : 2 .$$

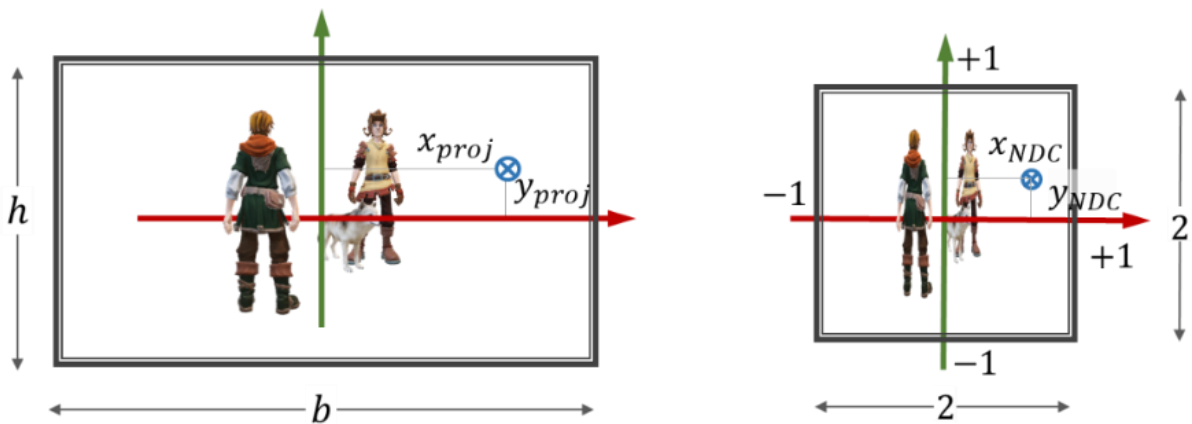


Abb. 10

Daraus ergibt sich

$$x_{NDC} = \frac{2}{b} \cdot x_{proj} = \frac{2}{b} \cdot \left(\frac{n}{-z} \cdot x \right) = \frac{\lambda x}{-z} \quad \text{mit } \lambda = \frac{2n}{b}$$

$$y_{NDC} = \frac{2}{h} \cdot y_{proj} = \frac{2}{h} \cdot \left(\frac{n}{-z} \cdot y \right) = \frac{\mu y}{-z} \quad \text{mit } \mu = \frac{2n}{h}$$

$$z_{NDC}^* = \alpha \cdot z_{proj} + \beta = \alpha \cdot z + \beta$$

Die Transformationsmatrix (Perspective-Projection -Matrix)

Gesucht ist nun die passende Transformationsmatrix, die die Koordinaten eines Vertex $(x \ y \ z \ 1)$ aus dem **VIEW-SPACE** direkt in normierte Koordinaten $(x_{NDC} \ y_{NDC} \ z_{NDC} \ 1)$ des **PROJECTION-SPACE** überführt.

Die für die Perspektive nötige Division durch $(-z)$ kann leider nicht mit Hilfe einer Matrizenmultiplikation realisiert werden und muss extra gestellt werden.

$$X_{NDC} = \begin{bmatrix} x_{NDC} \\ y_{NDC} \\ z_{NDC} \\ 1 \end{bmatrix} = \underbrace{\frac{1}{(-z)}}_{\substack{\text{erzeugt} \\ \text{Perspektive}}} \cdot \underbrace{\begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \rho & \sigma \\ 0 & 0 & -1 & 0 \end{bmatrix}}_{\substack{P_{NDC} \\ \text{Projection-Matrix}}} \cdot \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{X_{View}} = \frac{1}{(-z)} \cdot \begin{bmatrix} \lambda \cdot x \\ \mu \cdot y \\ \rho z + \sigma \\ (-z) \end{bmatrix} = \begin{bmatrix} \frac{\lambda \cdot x}{(-z)} \\ \frac{\mu \cdot y}{(-z)} \\ \frac{\rho z + \sigma}{(-z)} \\ 1 \end{bmatrix}$$

Die unterste Zeile der Matrix wurde dabei so gestaltet, dass die vierte homogene Koordinate den Wert $(-z)$ erhält und nach der Division den für Punkte charakteristischen Wert 1 annimmt.

Neuformulierung der z-Koordinate

9

Wir haben allerdings ein Problem mit der Koordinatentransformation bei z_{NDC} bekommen. Die z -Koordinate wurde ebenfalls durch $(-z)$ dividiert, was nicht vorgesehen war.

Andererseits ist die z_{NDC} -Koordinate nur insofern wichtig, als sie für die Ordnung „was liegt vorne, was hinten“ benötigt wird. Da sie letztlich in der 2dimensionalen Darstellung ohnehin wegfällt, könnte sie beliebig transformiert werden, solange diese Ordnung erhalten bleibt. Bei geeigneten Parametern ρ und σ ist das auch möglich.

Abb. 11 zeigt die Situation entlang der z -Achse. Bei Einhaltung der Randbedingungen für

$$z_{NDC} = \frac{\rho z + \sigma}{(-z)}$$

ist diese Transformation brauchbar.

Um die Parameter ρ und σ zu bestimmen, betrachten wir die in Abb. 11 markierten Punkte mit bekannten Koordinaten N und F als Randbedingungen.

$$N = (0, 0, -n) \rightarrow N_{NDC} = (0, 0, +1)$$

$$F = (0, 0, -f) \rightarrow F_{NDC} = (0, 0, -1)$$

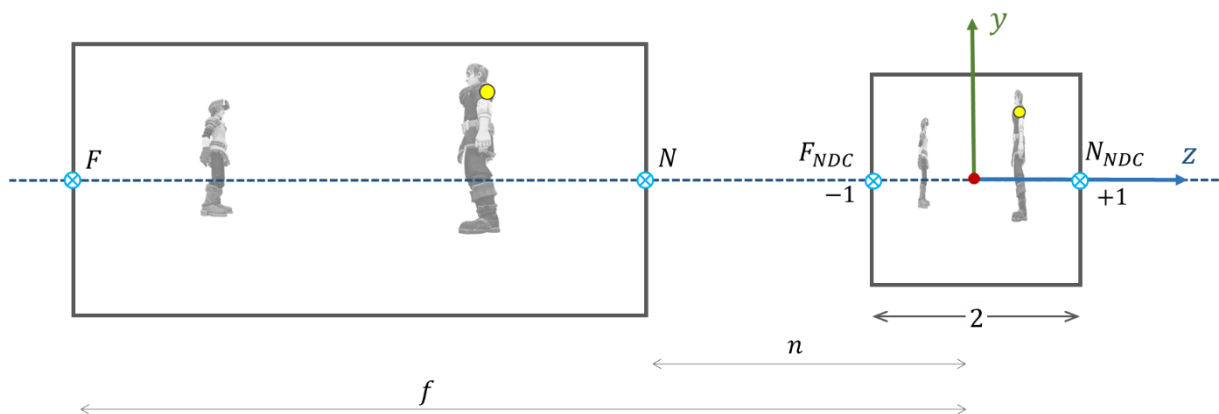


Abb. 11 Sicht-Quader (links) und Projection-Space (rechts)

Daraus folgen die Gleichungen

$$\left. \begin{array}{l} N: \frac{\rho(-n) + \sigma}{(-n)} = +1 \\ F: \frac{\rho(-f) + \sigma}{(-f)} = -1 \end{array} \right\}$$

Die Lösung dieses Gleichungssystems ergibt

$$z_{NDC} = \frac{\rho z + \sigma}{(-z)} \quad , \quad \rho = -\frac{f+n}{f-n} \quad \text{und} \quad \sigma = \frac{-2nf}{f-n}$$

Die *Projection-Matrix* für perspektivische Projektion lautet somit

$$P_{NDC} = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \rho & \sigma \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{2n}{b} & 0 & 0 & 0 \\ 0 & \frac{2n}{h} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{-2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

Bei festgelegtem Seitenverhältnis (*aspect ratio*) $a_{spect} = b : h =^{z.B.} 16 : 9$ sowie gegebenem Sehwinkel $\varphi =: FOV_y$ lautet die *Projection-Matrix*

$$P_{NDC} = \begin{bmatrix} \frac{1}{a_{spect} \cdot \tan(\frac{FOV_y}{2})} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\frac{FOV_y}{2})} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{-2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Begründung: $\tan\left(\frac{\varphi}{2}\right) = \frac{\frac{h}{2}}{n} = \frac{h}{2n} \Rightarrow \mu = \frac{2n}{h} = \frac{1}{\tan(\frac{\varphi}{2})}$
bzw. $\lambda = \frac{2n}{b} = \frac{2n}{a_{spect} \cdot h} = \frac{1}{a_{spect}} \cdot \mu = \frac{1}{a_{spect} \cdot \tan(\frac{\varphi}{2})}$

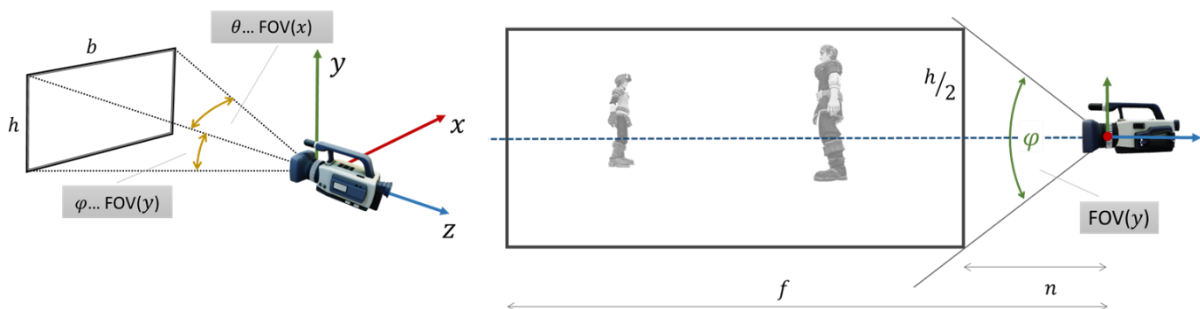


Abb. 12

BEISPIEL: Nah-Ebene bei $n = 2$, Fern-Ebene bei $f = 6 \Rightarrow \rho = -2, \sigma = -6$

Abb. 13 zeigt die Abbildung der z -Koordinate aus dem **VIEW-SPACE** in den **PROJECTION-SPACE** (NDC). Sowohl die lineare Abbildung $z_{NDC} = \rho z + \sigma$ als auch die nichtlineare Abbildung $z_{NDC} = \frac{\rho z + \sigma}{-z}$ sind monoton steigend, lassen also die Ordnung unverändert.

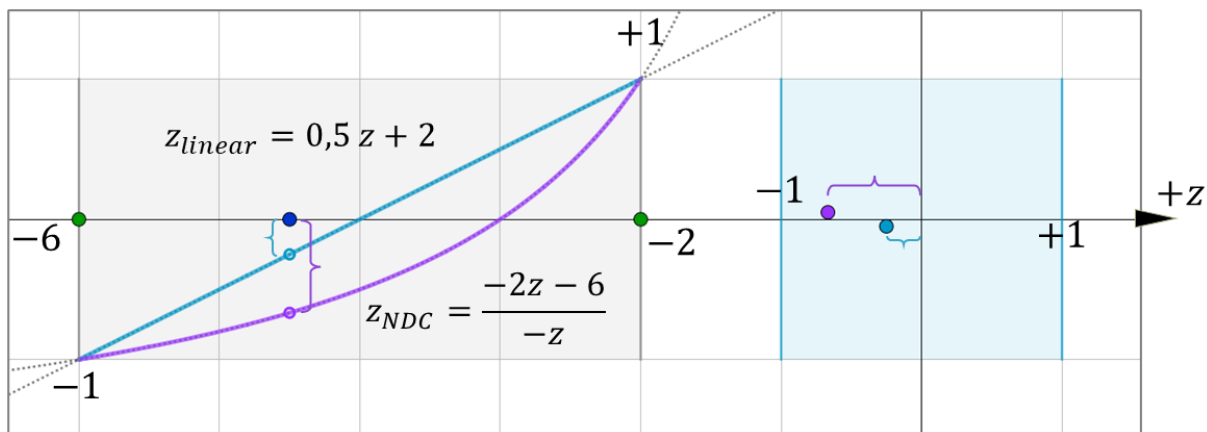


Abb. 13: Lineare versus nichtlineare Transformation der z-Koordinaten in den NDC-Space



– Fighting

Z-fighting hat nichts mit der 1919 erstmals veröffentlichten Romanfigur „Zorro, Rächer der Armen“, zu tun. Es geht vielmehr um das Problem, dass bei der nichtlinearen Abbildung der z -Werte die fernen z_{NDC} -Werte im Projection-Space dichter beieinander liegen als im Nahbereich. (Abb. 14)

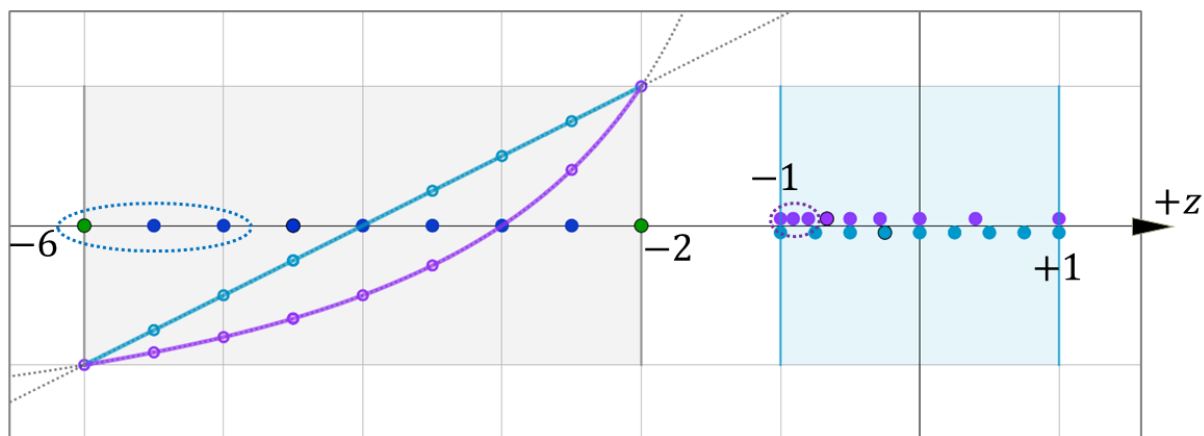


Abb. 14

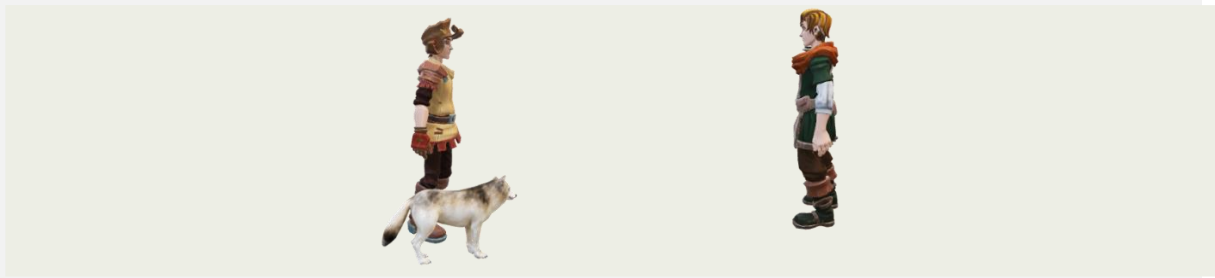
Im Nahbereich ist die Auflösung der Entfernungswerte somit größer als in der Ferne. Das kann dort zu Entscheidungsproblemen führen, wenn die z -Werte nahezu identisch sind (z -Fighting, Abb. 15).



Abb. 15: An manchen Stellen der Scheibe wird die Wand vor der Scheibe angeordnet. (Künstliche Imitation von z-Fighting)

4 Beispiel (Fortsetzung)

13



ANGABE: Wolf, linke Ohrspitze; Maße in cm

$$Q_0 = (5 \ 2 \ -1.5)$$

$$Q_{View} = V \cdot M \cdot Q_0 = (48,140 \ -545,867 \ -552,446 \ 1)$$



Abb. 16

PROJECTION-SPACE

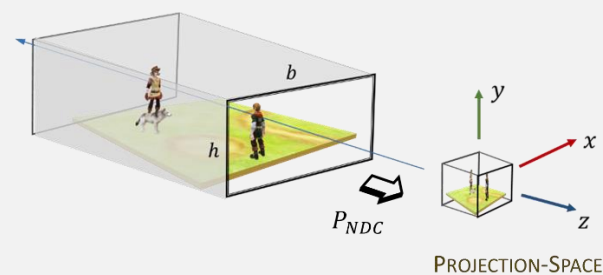


Abb. 17

ANGABEN:

$FOV_y = 100^\circ$; *aspect ratio*: 16/9 ; *near*: $n = 1 \text{ m} \hat{=} 100$; *far*: $f = 5 \text{ km} \hat{=} 500000$

Ergebnis: ➤ $P_{NDC} = \begin{bmatrix} 0,47199 & 0 & 0 & 0 \\ 0 & 0,8390996312 & 0 & 0 \\ 0 & 0 & -1,00040 & -200,04001 \\ 0 & 0 & -1 & 0 \end{bmatrix}$

Für die Ohrspitze des Wolfes bedeutet das

$Q_{View} = \begin{bmatrix} 48,140 \\ -545,867 \\ -552,446 \\ 1 \end{bmatrix}$ und $z_{Q_{View}} = -552,446$ in den z-Buffer!

$$Q_{NDC} = \frac{1}{(-z)} \cdot \begin{bmatrix} 0,47199 & 0 & 0 & 0 \\ 0 & 0,83910 & 0 & 0 \\ 0 & 0 & -1,00040 & -200,04001 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 48,140 \\ -545,867 \\ -552,446 \\ 1 \end{bmatrix} =$$

$$= \frac{1}{552,445846} \cdot \begin{bmatrix} 22,72164 \\ -458,03717 \\ 352,62686 \\ 552,44585 \end{bmatrix} =^* \begin{bmatrix} 0,04113 \\ -0,82908 \\ 0,63828 \\ 1 \end{bmatrix}$$

* Ergebnis bei Berechnung mit 10-Nachkommastellen-Genauigkeit

5 Die [MVP]-Matrix (Model-View-Projection-Matrix) ▲

In der Praxis werden im **VERTEX-SHADER** zuerst alle Transformationsmatrizen miteinander multipliziert und mit dieser [MVP]-Matrix dann alle Vertex-Koordinaten des Modells in einem Rutsch direkt in den Projection-Space umgerechnet.

15

$$[MVP] \cdot \{Model\} = P \left(V \left(M(\{Model\}) \right) \right) = P \cdot V \cdot M \cdot \{Model\}$$

BEISPIEL (FORTSETZUNG)

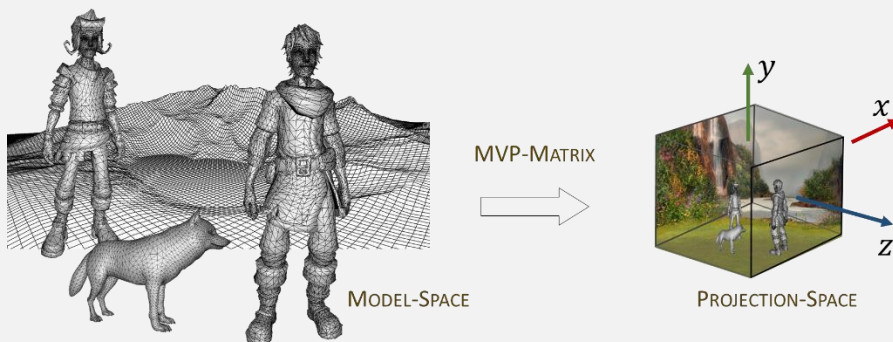


Abb. 18

Model to View to Projection

$$[MVP] = [Model\ to\ View\ to\ Projection] = [M\ to\ V\ to\ P] =: (P \cdot V \cdot M)$$

$$[MVP] = P \cdot V \cdot M =$$

$$= \underbrace{\begin{bmatrix} 0,47199 & 0 & 0 & 0 \\ 0 & 0,8390996312 & 0 & 0 \\ 0 & 0 & -1,00040 & -200,04001 \\ 0 & 0 & -1 & 0 \end{bmatrix}}_{View\ to\ ProjectionNDC} \cdot \underbrace{\begin{bmatrix} 0,866 & 0 & 0,5 & -356,699 \\ -0,171 & 0,940 & 0,296 & -621,386 \\ -0,470 & -0,342 & 0,814 & -537,722 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{World\ to\ View} \cdot \underbrace{\begin{bmatrix} 6,691 & 0 & -7,431 & 300 \\ 0 & 10 & 0 & 56 \\ 7,431 & 0 & 6,691 & 260 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{Model\ to\ World}$$

$$[MVP] = \begin{bmatrix} 0,98133 & 0 & -1,45854 & 15,62718 \\ -2,80718 & 7,88496 & 2,72943 & -455,67706 \\ 9,19158 & 3,42157 & -8,94058 & 286,39656 \\ 9,19158 & 3,42020 & -8,93701 & 486,24203 \end{bmatrix}$$

* Ergebnis bei einer Rechengenauigkeit von 10-Nachkommastellen (<https://matrixcalc.org>)

Beispiel Ohrspitze:

$$[MVP] \cdot Q_0 = [MVP] \cdot \begin{bmatrix} 5 \\ 2 \\ -1,5 \\ 1 \end{bmatrix} = (P \cdot V \cdot M) \cdot \begin{bmatrix} 5 \\ 2 \\ -1,5 \\ 1 \end{bmatrix} = \begin{bmatrix} 22,72164 \\ -458,03717 \\ 352,62686 \\ 552,44585 \end{bmatrix} = \begin{bmatrix} x_{NDC} \cdot W \\ y_{NDC} \cdot W \\ z_{NDC} \cdot W \\ W \end{bmatrix}$$

wobei $w = (-z_{Q_{view}})$ als vierte Koordinate aufscheint.

$$Q_{NDC} = \frac{1}{w} \cdot \begin{bmatrix} 22,72164 \\ -458,03717 \\ 352,62686 \\ 552,44585 \end{bmatrix} = \frac{1}{552,44585} \cdot \begin{bmatrix} 22,72164 \\ -458,03717 \\ 352,62686 \\ 552,44585 \end{bmatrix} = \begin{bmatrix} 0,04113 \\ -0,82908 \\ 0,63828 \\ 1 \end{bmatrix}$$