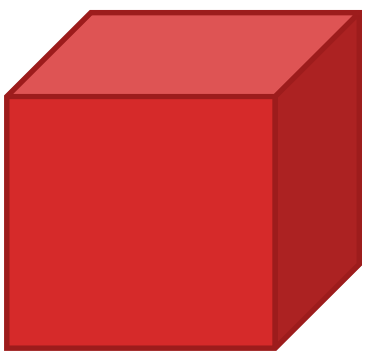
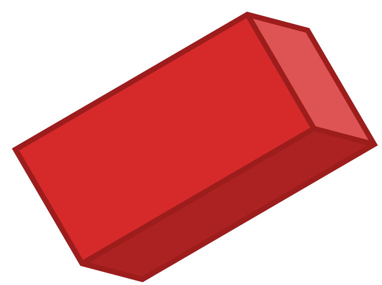
Transformationen

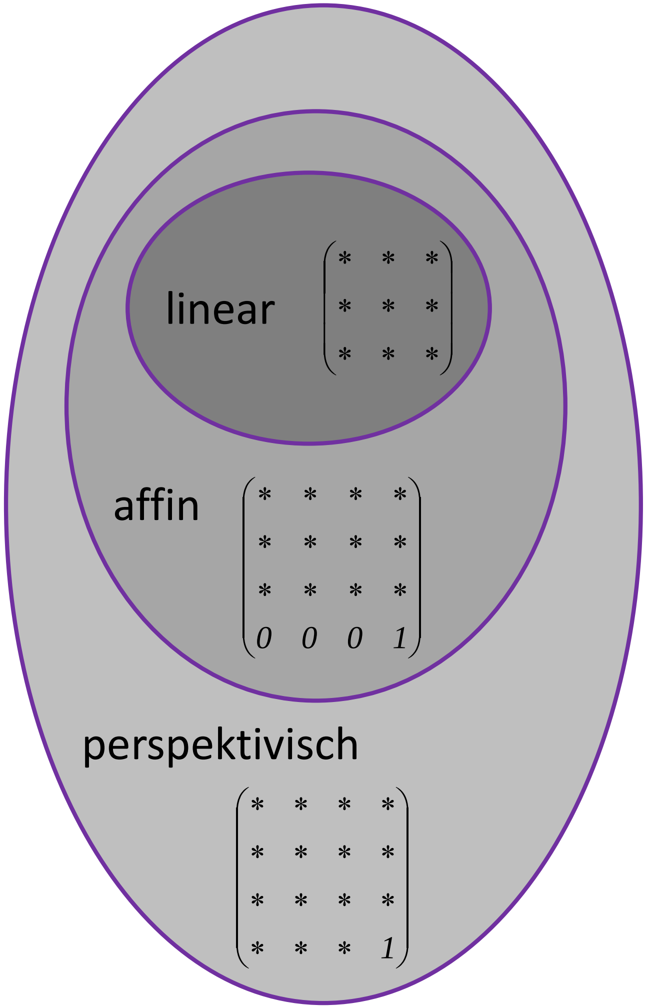


$$\boldsymbol{R}^3 \xrightarrow{f} \boldsymbol{R}^3$$

$$\underline{\boldsymbol{q}} = f(\underline{\boldsymbol{p}})$$



Inhalt


linear
$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$
affin
$\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & 0 & 0 & 1 \end{pmatrix}$
perspektivisch
$\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & 1 \end{pmatrix}$

matrices

- operations on matrices include multiplication with scalar

$$2\cdot\begin{pmatrix}1 & -4\\ 3 & 2\end{pmatrix}=\begin{pmatrix}2 & -8\\ 6 & 4\end{pmatrix}$$

- sum auf two matrices of same dimension

$$\begin{pmatrix}1 & -4\\ 3 & 2\end{pmatrix}+\begin{pmatrix}2 & 2\\ 2 & 2\end{pmatrix}=\begin{pmatrix}3 & -2\\ 5 & 4\end{pmatrix}$$

- product of two matrices with matching dimensions

$$\begin{pmatrix}a_{11} & \cdots & a_{1m}\\ \vdots & & \vdots\\ \boxed{a_{i1} \quad \cdots \quad a_{im}} & & \\ \vdots & & \vdots\\ a_{r1} & \cdots & a_{rm}\end{pmatrix}\begin{pmatrix}b_{11} & \cdots & b_{1j} & \cdots & b_{1c}\\ \vdots & & \vdots & & \vdots\\ b_{m1} & \cdots & b_{mj} & \cdots & b_{mc}\end{pmatrix}=\begin{pmatrix}p_{11} & \cdots & p_{1j} & \cdots & p_{1c}\\ \vdots & & \vdots & & \vdots\\ p_{i1} & \cdots & \boxed{p_{ij}} & \cdots & p_{ic}\\ \vdots & & \vdots & & \vdots\\ p_{r1} & \cdots & p_{rj} & \cdots & p_{rc}\end{pmatrix}$$

- Matrix multiplication is composed of scalar products:

$$\boxed{p_{ij}}=a_{i1}b_{1j}+a_{i2}b_{2j}+\ldots+a_{im}b_{mj}$$

$$\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix}=\begin{pmatrix}\vec{\boldsymbol{a}}_{1.}\\ \vec{\boldsymbol{a}}_{2.}\end{pmatrix}\begin{pmatrix}\vec{\boldsymbol{b}}_{.1} & \vec{\boldsymbol{b}}_{.2}\end{pmatrix}=\begin{pmatrix}\vec{\boldsymbol{a}}_{1.}\cdot\vec{\boldsymbol{b}}_{.1} & \vec{\boldsymbol{a}}_{1.}\cdot\vec{\boldsymbol{b}}_{.2}\\ \vec{\boldsymbol{a}}_{2.}\cdot\vec{\boldsymbol{b}}_{.1} & \vec{\boldsymbol{a}}_{2.}\cdot\vec{\boldsymbol{b}}_{.2}\end{pmatrix}$$

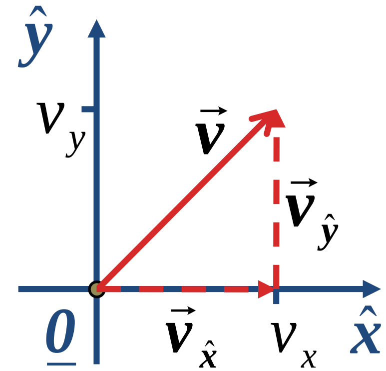
Matrixinvertierung

- Einheitsmatrix $\boldsymbol{I}$: $\boldsymbol{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$
- Inverse: $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$
- Berechnung: über Gauß-Jordan Algorithmus oder Adjunkte
- Matrixinversion in 3D mit Vektorprodukten
  - geg.: $\boldsymbol{A} = \begin{pmatrix} \vec{\boldsymbol{a}}_1^T \\ \vec{\boldsymbol{a}}_2^T \\ \vec{\boldsymbol{a}}_3^T \end{pmatrix} \in \boldsymbol{R}^{3\times 3}$
  - ges.: $\boldsymbol{B} = \begin{pmatrix} \vec{\boldsymbol{b}}_1 & \vec{\boldsymbol{b}}_2 & \vec{\boldsymbol{b}}_3 \end{pmatrix} \in \boldsymbol{R}^{3\times 3} : \boldsymbol{AB} = \boldsymbol{1}$, d.h. $\langle \vec{\boldsymbol{a}}_i, \vec{\boldsymbol{b}}_j \rangle = \delta_{ij} = \begin{cases} 1 \ldots i=j \\ 0 \ldots \text{sonst} \end{cases}$
  - somit $\vec{\boldsymbol{b}}_i \perp \vec{\boldsymbol{a}}_{j\neq i} \Rightarrow \vec{\boldsymbol{b}}_i = \dfrac{\vec{\boldsymbol{a}}_j \times \vec{\boldsymbol{a}}_k}{[\vec{\boldsymbol{a}}_i, \vec{\boldsymbol{a}}_j, \vec{\boldsymbol{a}}_k]}$ mit $\{i, j, k\} \in \{\{1,2,3\}, \{2,3,1\}, \{3,1,2\}\}$

LINEARE TRANSFORMATIONEN

Lineare Transformationen Koordinatendarstellung

- Ein Koordinatensystem ist durch einen Ursprung $\underline{0}$ und Vektoren $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$ gegeben, die die Koordinatenachsen aufspannen.
- Ein beliebiger Vektor $\vec{\boldsymbol{v}}$ kann in Komponenten $\vec{\boldsymbol{v}}_{\hat{x}}, \vec{\boldsymbol{v}}_{\hat{y}}$ entlang der Koordinatenachsen zerlegt werden, woraus sich seine komponentenweise Darstellung ergibt.
- Die Vektoren der Koordinaten-achsen bilden eine Basis, die man orthonormal nennt, wenn die Vektoren Länge eins haben und senkrecht aufeinander stehen.



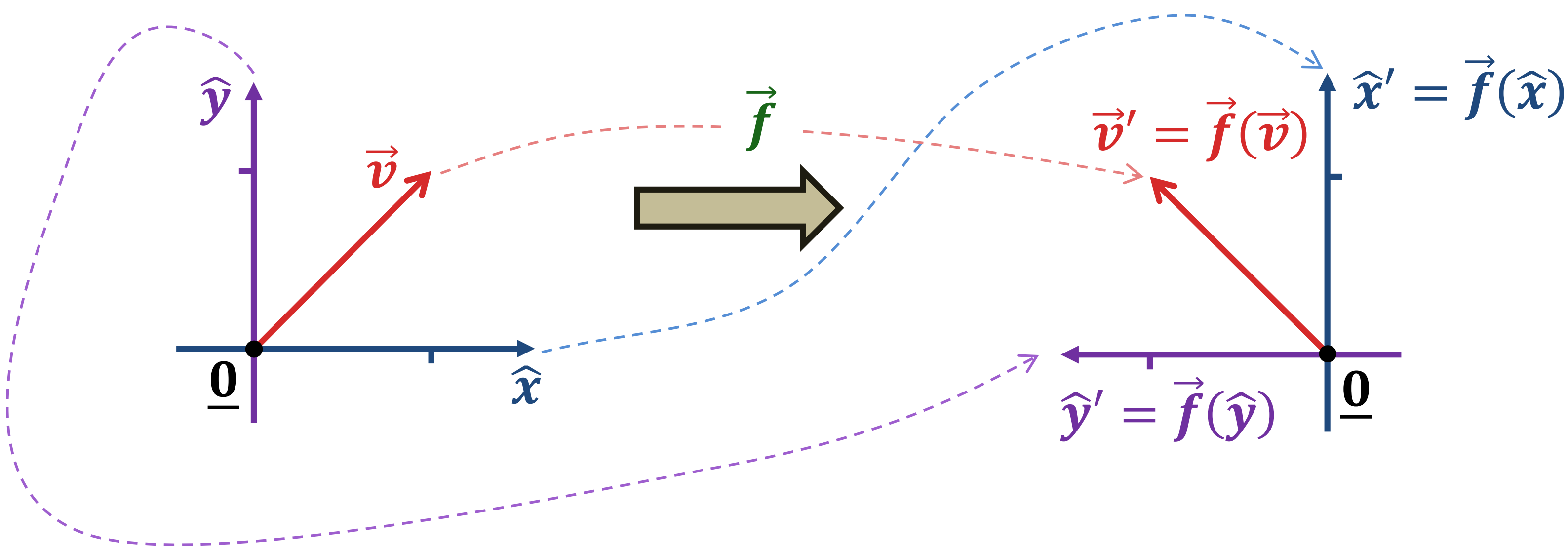
$$
\begin{aligned}
\vec{\boldsymbol{v}} &= \begin{pmatrix} v_x \\ v_y \end{pmatrix} \\
&= v_x \hat{\boldsymbol{x}} + v_y \hat{\boldsymbol{y}} \\
&= \vec{\boldsymbol{v}}_{\hat{x}} + \vec{\boldsymbol{v}}_{\hat{y}} \\
&= \langle \vec{\boldsymbol{v}}, \hat{\boldsymbol{x}} \rangle \hat{\boldsymbol{x}} + \langle \vec{\boldsymbol{v}}, \hat{\boldsymbol{y}} \rangle \hat{\boldsymbol{y}}
\end{aligned}
$$

Lineare Transformationen
Definition

Computergraphik und Visualisierung

- Die einfachste Art der Transformationen wird durch eine lineare Abbildung repräsentiert, die Linearkombinationen erhält, d.h.

$$\vec{f}\left(a\vec{\mathbf{u}}+b\vec{\mathbf{v}}\right)=a\vec{f}\left(\vec{\mathbf{u}}\right)+b\vec{f}\left(\vec{\mathbf{v}}\right)$$



- Bei einer linearen Transformation bleibt der Ursprung erhalten.

Lineare Transformationen Matrixdarstellung

- Jede lineare Abbildung kann als Matrix dargestellt werden und ein Vektor wird durch Matrix-Vektor-Multiplikation transformiert

$$\forall \text{ lineare } \vec{f} : \boldsymbol{R}^n \to \boldsymbol{R}^n : \exists \boldsymbol{M} \in \boldsymbol{R}^{n\times n} : \forall \vec{\mathbf{v}} \in \boldsymbol{R}^n : \vec{f}\left(\vec{\mathbf{v}}\right) = \boldsymbol{M}\vec{\mathbf{v}}$$

- in den Spalten von $\boldsymbol{M}$ stehen die Bilder der Basisvektoren

$$\boldsymbol{R}^{2\times 2} : \boldsymbol{M} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}' & \vec{\mathbf{y}}' \end{pmatrix} = \begin{pmatrix} \vec{f}\left(\hat{\mathbf{x}}\right) & \vec{f}\left(\vec{\mathbf{y}}\right) \end{pmatrix}$$

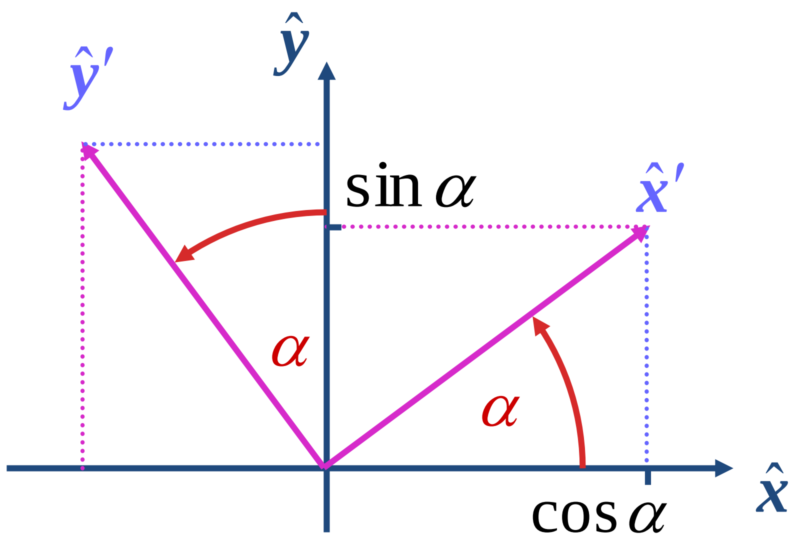
- dies folgt zum Bsp. für x durch Multiplikation von $\boldsymbol{M}$ mit $\hat{\boldsymbol{x}}$

$$\hat{\mathbf{x}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \hat{\mathbf{x}}' = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} M_{11} \\ M_{21} \end{pmatrix} = M_{11}\hat{\mathbf{x}} + M_{21}\hat{\mathbf{y}}$$

Lineare Transformationen
Aufstellen einer speziellen Matrix

Computergraphik und Visualisierung

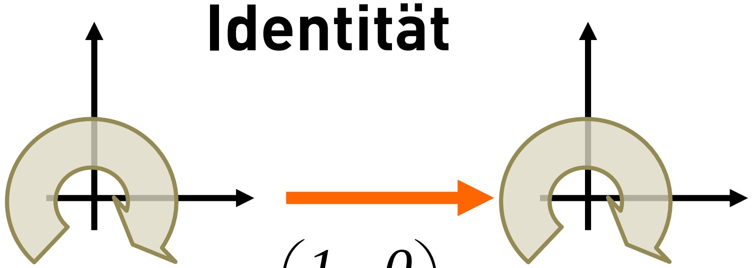
- Beim Entwurf einer speziellen Transformation muss man sich somit nur überlegen, wie die Basisvektoren abgebildet werden
- Bsp.: Rotation um den Winkel α gegen den Urzeigersinn



$$M = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

Lineare Transformationen
Spiegelungen

Identität

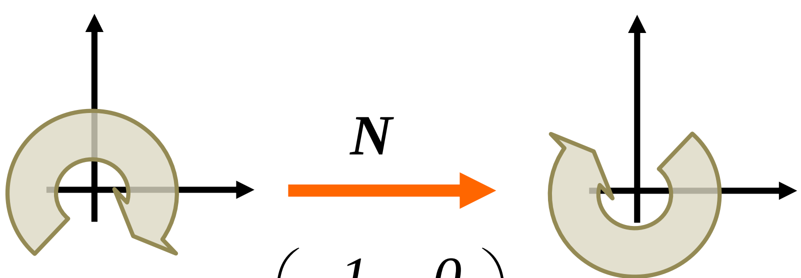


$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{I}^{-1} = \boldsymbol{I}$$

- Im 2D entspricht die Punktspiegelung einer Rotation um 180 Grad
- Achsenspiegelungen wie Punktspiegelung im 3D kehren nicht symmetrische Objekte in sich um. Flächeninhalt / Volumen wird dabei negiert

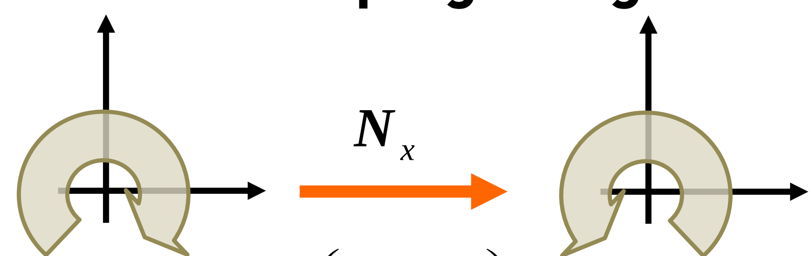
Punktspiegelung



$$\boldsymbol{N} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\boldsymbol{N}^{-1} = \boldsymbol{N}$$

Achsenspiegelung



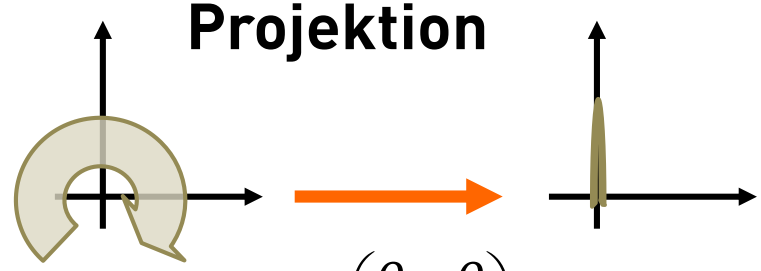
$$\boldsymbol{N}_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{N}_x^{-1} = \boldsymbol{N}_x$$

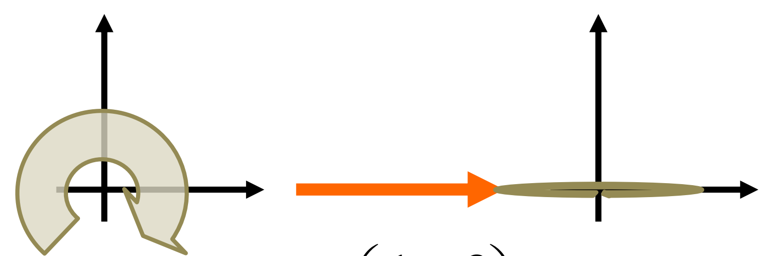
Lineare Transformationen
Projektionen

- Projektionen werden verwendet, um auf die Bildebene zu projizieren. Dabei wird die Komponente in Projektionsrichtung einfach weggelassen (auf 0 gesetzt)
- Projektionen sind nicht umkehrbar

Projektion



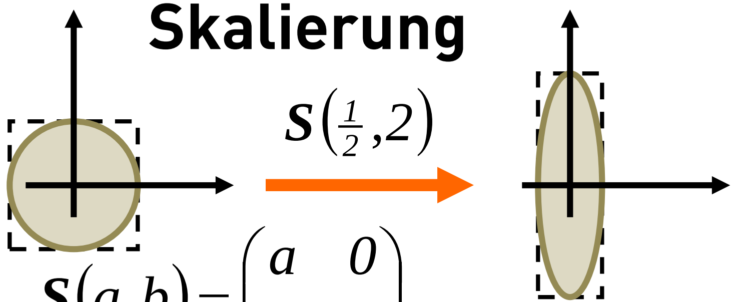
$$\Pi_x = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$



$$\Pi_y = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Computergraphik und Visualisierung

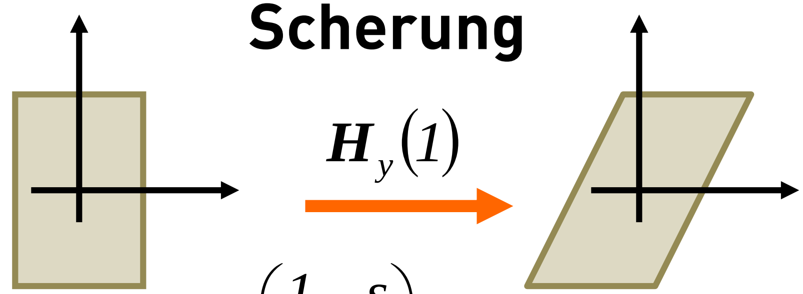
Skalierung



$$S(a,b)=\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$$

$$S(a,b)^{-1}=S\left(\tfrac{1}{a},\tfrac{1}{b}\right)$$

Scherung



$$H_y(s)=\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$$

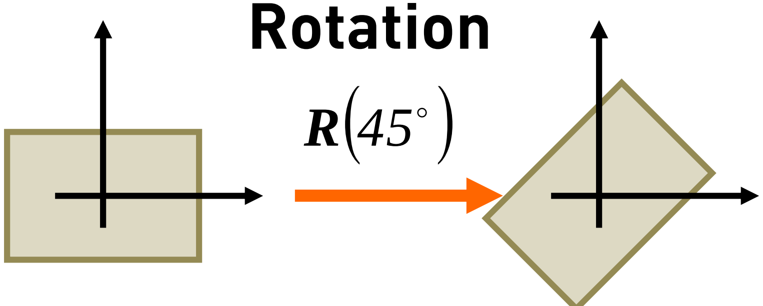
$$H_y(s)^{-1}=H_y(-s)$$

$$H_x(s_y,s_z)=\begin{pmatrix} 1 & 0 & 0 \\ s_y & 1 & 0 \\ s_z & 0 & 1 \end{pmatrix}$$

$$H_y(s_x,s_z)=\begin{pmatrix} 1 & s_x & 0 \\ 0 & 1 & 0 \\ 0 & s_z & 1 \end{pmatrix}$$

$$H_z(s_x,s_y)=\begin{pmatrix} 1 & 0 & s_x \\ 0 & 1 & s_y \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation



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

$$R(\alpha)^{-1}=R(\alpha)^T=R(-\alpha)$$

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

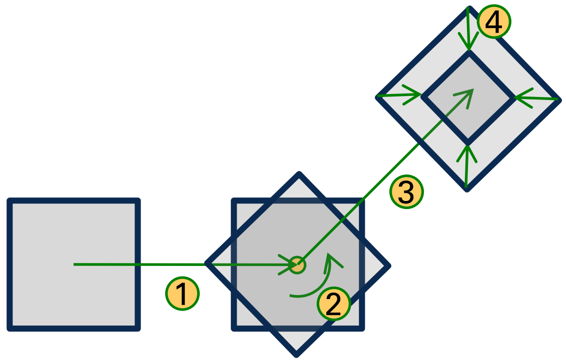
$$R_y(\alpha)=\begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}$$

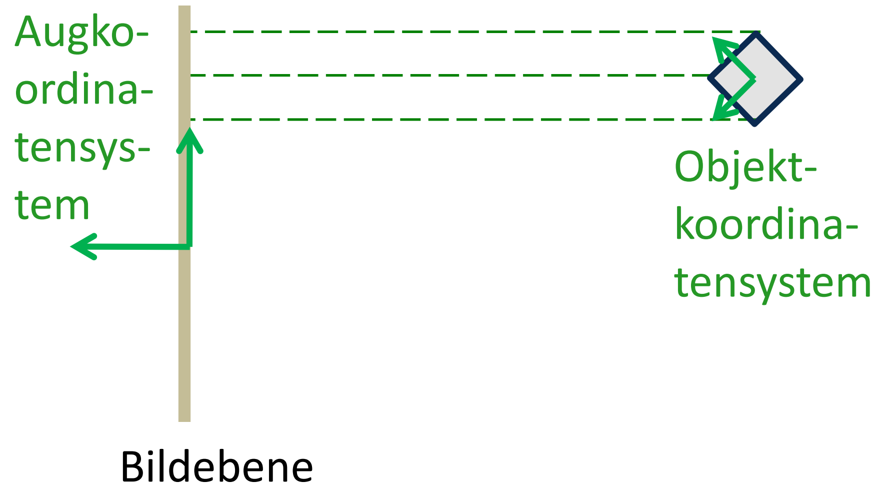
$$R_z(\alpha)=\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

SYSTEMTRANSFORMATIONEN

Systemtransformationen
Einsatz von Transformationen

Computergraphik und Visualisierung





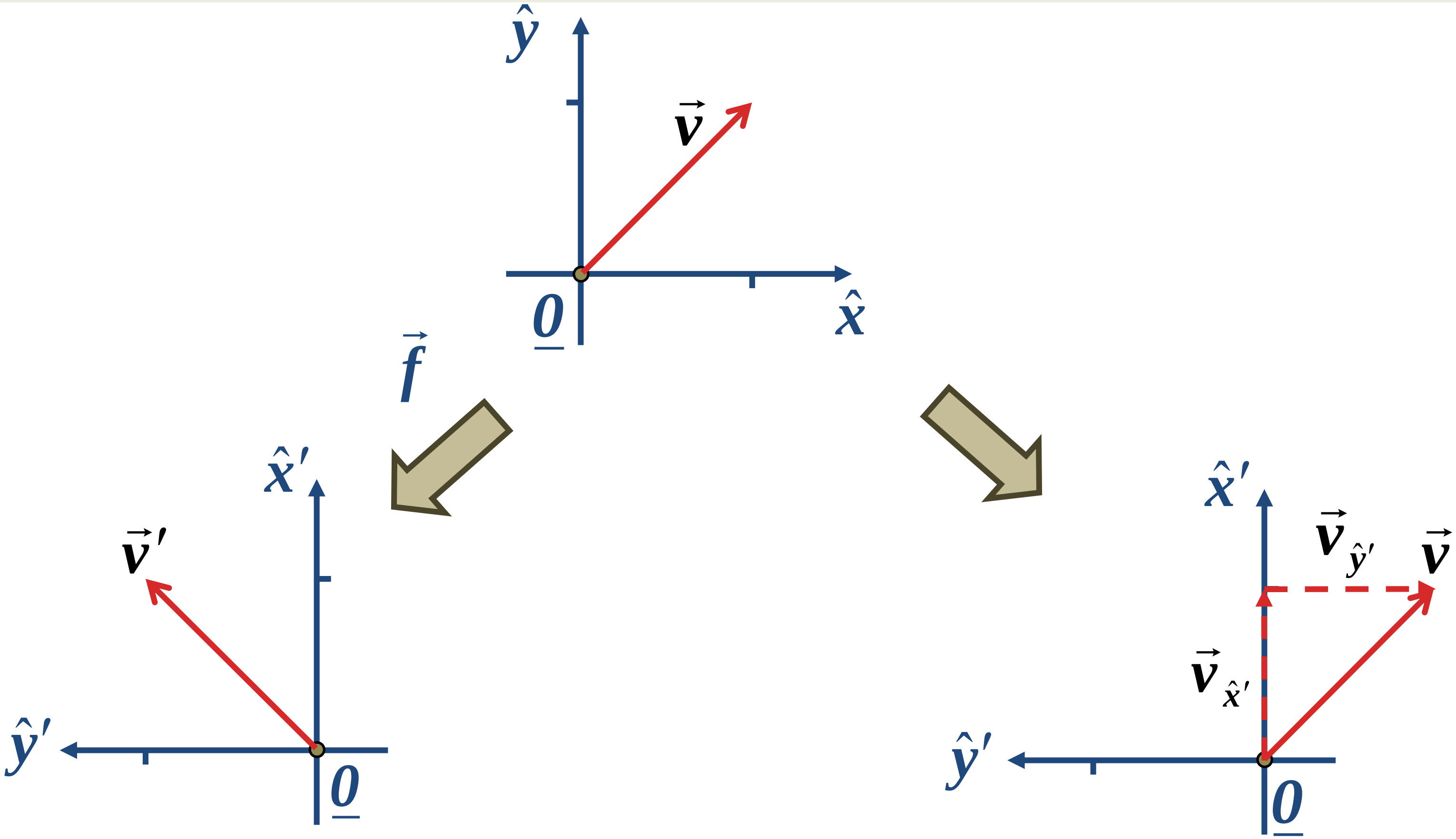
Modelltransformation (bisher)

- Positionierung von Objekten in einer Szene
- Beispiel
  - Translation
  - Rotation
  - Skalierung

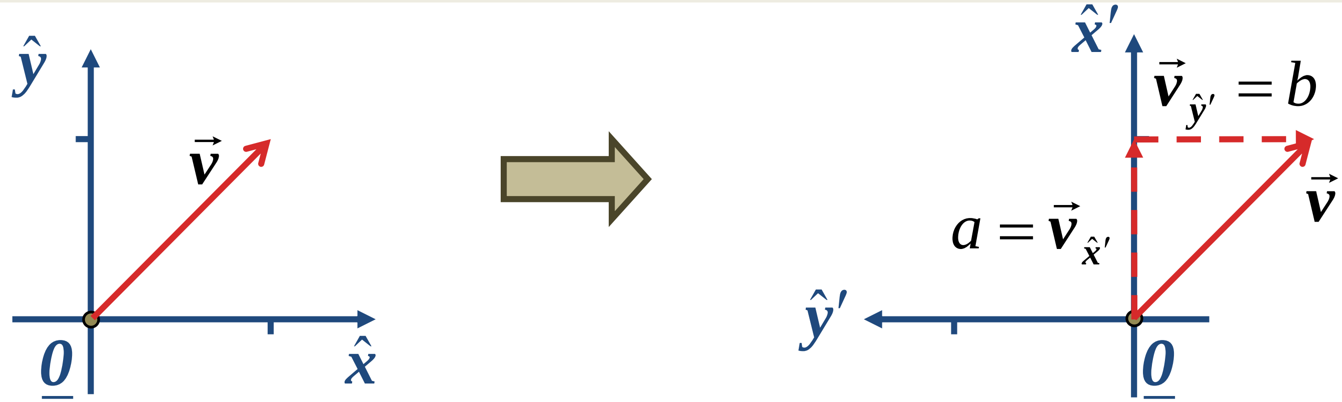
Systemtransformation

- Umrechnung der Koordinaten in ein anderes Koordinatensystem z.B. zur Projektion von Geometrie in die Bildebene

Systemtransformationen
Definition



- Modelltransformation transformiert $\vec{v}$ so zu $\vec{v}'$, dass $\hat{x}, \hat{y}$ auf $\hat{x}', \hat{y}'$ abgebildet werden
- Systemtransformation berechnet die Koordinatendarstellung von $\vec{v}$ in $\hat{x}', \hat{y}'$



$$\vec{v} = a\hat{\mathbf{x}}' + b\hat{\mathbf{y}}' \quad = \begin{pmatrix}\hat{\mathbf{x}}' & \hat{\mathbf{y}}'\end{pmatrix}\begin{pmatrix}a \\ b\end{pmatrix} \Rightarrow \begin{pmatrix}a \\ b\end{pmatrix} = \begin{pmatrix}\hat{\mathbf{x}}' & \hat{\mathbf{y}}'\end{pmatrix}^{-1}\vec{v}$$

- Die Systemtransformation ergibt sich durch Invertieren der Matrix mit den transformierten Basisvektoren in den Spalten
- D.h. Modell- und Systemtransformationen sind zueinander inverse Transformationen

Systemtransformationen
Erweiterung der Notation

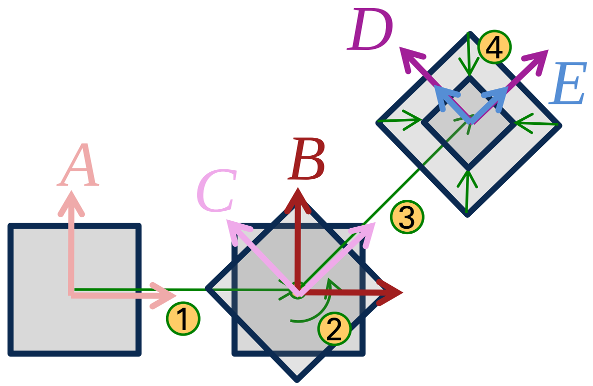
Computergraphik und Visualisierung

- Vektor als Element des Vektorraumes $\vec{\boldsymbol{v}}$
- Vektor in Komponentendarstellung des Koordinatensystems A $\vec{\boldsymbol{v}}_A$
- Matrixnotation mit 2 Interpretationen: $\boldsymbol{M}_A^B$
  1. Systemtransformation von Koordinatensystem B ins Koordinatensystem A
  2. Modelltransformation im System A mit Abbildung der Basis A auf B
- Einsatz
  - Systemtransformationsgleichung $\vec{\boldsymbol{v}}_A = \boldsymbol{M}_A^B \vec{\boldsymbol{v}}_B$
  - Modelltransformationsgleichung $\vec{\boldsymbol{v}}'_A = \boldsymbol{M}_A^B \vec{\boldsymbol{v}}_A$
  - Umkehrtransformation $\boldsymbol{M}_B^A = \left(\boldsymbol{M}_A^B\right)^{-1}$
  - Verkettung $\boldsymbol{M}_A^C = \boldsymbol{M}_A^B \boldsymbol{M}_B^C$

Systemtransformationen Verkettung

Computergraphik und Visualisierung

- Bei der Verkettung von Modell-transformationen über Matrix-multiplikation ist die zweite Transformation relativ zu den Bildbasisvektoren der ersten Transformation zu spezifizieren
- Man multipliziert deshalb die zweite Transformationsmatrix von rechts an die erste.
- Die so erhaltene Gesamttransformation ist gleichzeitig die Systemtransformation, die vom letzten Koordinatensystem ins Anfangskoordinatensystem transformiert



$$A \xrightarrow[1]{\tilde{\boldsymbol{T}}_A^B} B \xrightarrow[2]{\tilde{\boldsymbol{R}}_B^C} C \xrightarrow[3]{\tilde{\boldsymbol{T}}_C^D} D \xrightarrow[4]{\tilde{\boldsymbol{S}}_D^E} E$$

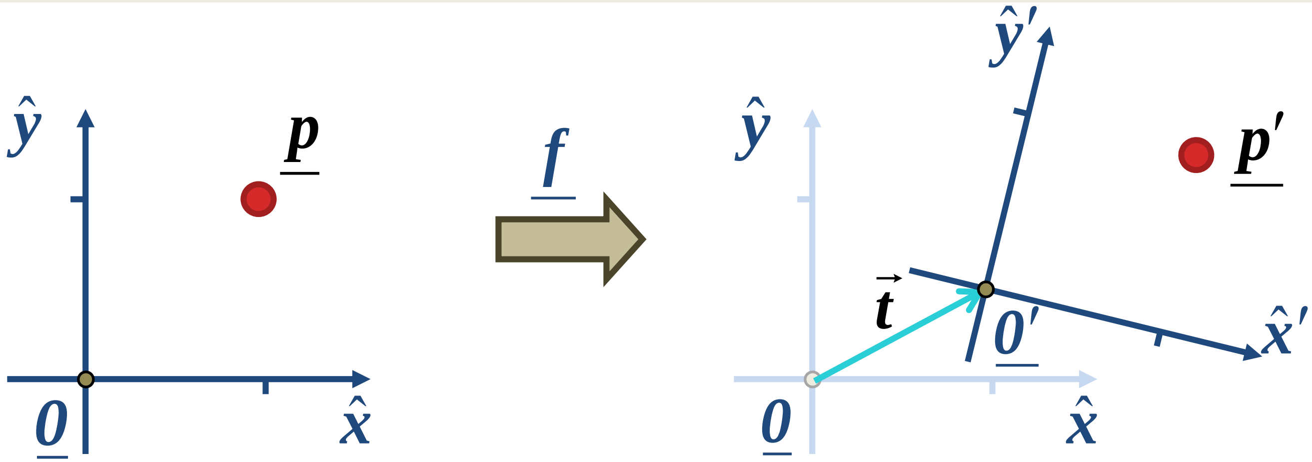
Abfolge der Transformationen zwischen Koordinatensystemen A bis E

$$\tilde{\boldsymbol{M}}_A^E = \tilde{\boldsymbol{T}}_A^B \tilde{\boldsymbol{R}}_B^C \tilde{\boldsymbol{T}}_C^D \tilde{\boldsymbol{S}}_D^E$$

Gesamttransformation

AFFINE TRANSFORMATIONEN

Affine Transformationen
Integration von Translationen



- Bei Punkten sind zusätzlich zu linearen Transformationen auch Verschiebungen (Translationen) sinnvoll.
- Affine Abbildungen erweitern lineare Abbildungen um Translationen, indem zusätzlich der Ursprung des Koordinatensystems verschoben wird.
- Mit der Menge aller n-dimensionalen Punkte $\underline{\boldsymbol{A}}^n$ gilt:

$$\forall \text{ affine } \underline{\boldsymbol{f}} : \boldsymbol{A}^n \to \boldsymbol{A}^n : \exists \boldsymbol{M} \in \boldsymbol{R}^{n\times n}, \vec{\boldsymbol{t}} \in \boldsymbol{R}^n : \forall \underline{\boldsymbol{p}} \in \boldsymbol{A}^n : \underline{\boldsymbol{f}}\left(\underline{\boldsymbol{p}}\right) = \boldsymbol{M}\,\underline{\boldsymbol{p}} + \vec{\boldsymbol{t}}$$

Affine Transformationen
Homogene Darstellung

Computergraphik und Visualisierung

- Die Transformationsvorschrift einer affinen Transformation kann durch Erweiterung der Komponentendarstellung des Punktes wieder mit einer Matrix-Vektor-Multiplikation geschrieben werden
- Die neue Komponente wird *w*-Komponente genannt
- Differenzvektoren werden nicht transliert und erhalten eine Null als *w*-Komponente:

$$\vec{v} = \underline{p} - \underline{q}$$

$$\Rightarrow \vec{v}' = \underline{p}' - \underline{q}' = M\underline{p} + \vec{t} - \left(M\underline{q} + \vec{t}\right) = M\left(\underline{p} - \underline{q}\right) = M\vec{v}$$

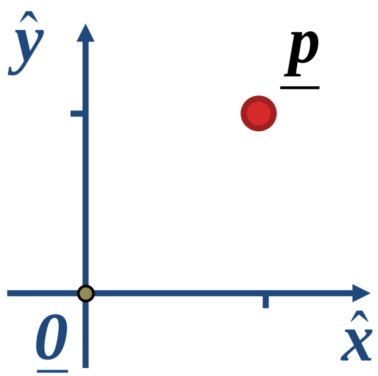
$$\underline{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \Rightarrow \tilde{p} = \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

$$\{M, \vec{t}\} \Rightarrow \tilde{M} = \begin{pmatrix} M_{11} & M_{12} & t_x \\ M_{21} & M_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$\underline{p}' = M\underline{p} + \vec{t} \quad \Rightarrow \quad \tilde{p}' = \tilde{M}\tilde{p}$$

$$\vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \Rightarrow \tilde{v} = \begin{pmatrix} v_x \\ v_y \\ 0 \end{pmatrix}$$

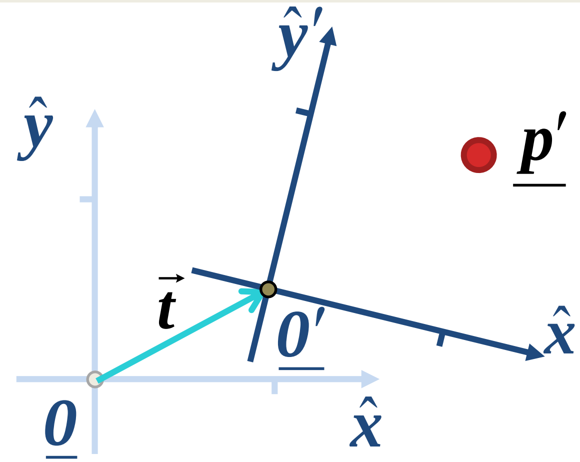
Affine Transformationen
Homogene Darstellung interpretiert



$\vec{f}$

$$\tilde{M} = \begin{pmatrix} M_{11} & M_{12} & t_x \\ M_{21} & M_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

$\tilde{x}'$ $\quad \tilde{y}'$ $\quad \tilde{0}'$



- Auch in der homogenen Darstellung können die Spalten der Transformationsmatrix als Bilder der Basis interpretiert werden – jeweils in homogener Darstellung als Vektor bei w=0 oder Punkt bei w=1
- Die w-Spalte ist das Bild des Ursprungs (Punkt) der zur Basis des linearen Falls hinzukommt.

Affine Transformationen
Definition

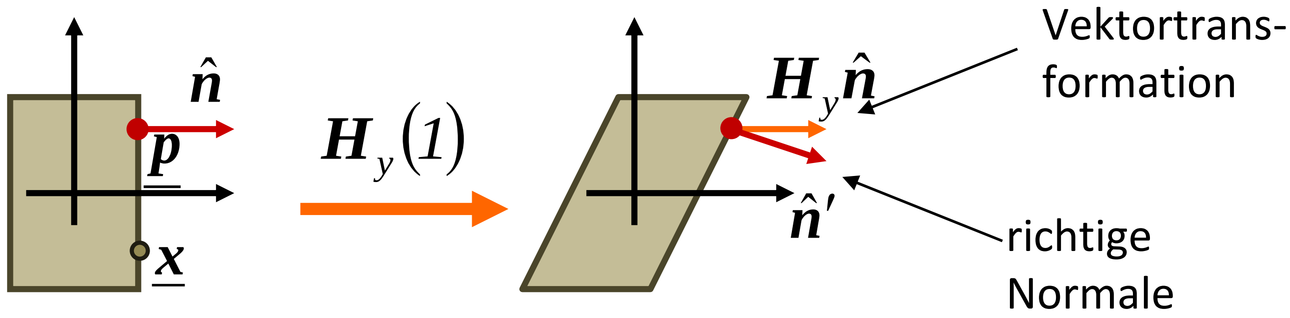
Computergraphik und Visualisierung

- Nimmt man lineare Transformationen und Translationen zusammen, so ergeben sich die affinen Transformationen
- In der homogenen Darstellung kann man sie auf Punkte ($w=1$) wie Vektoren ($w=0$) anwenden. Dabei werden stets Punkte auf Punkte und Vektoren auf Vektoren abgebildet.
- Affine Transformationen erhalten Linearkombinationen, bei denen sich die Gewichte zu eins summieren; diese werden Affinkombinationen genannt:

$$\underline{\boldsymbol{f}}\Big(\underbrace{(1-\lambda)\underline{\boldsymbol{p}}+\lambda\underline{\boldsymbol{q}}}_{\text{Affinkom-bination}}\Big)\overset{\text{affine Invarianz}}{=}(1-\lambda)\underline{\boldsymbol{f}}\big(\underline{\boldsymbol{p}}\big)+\lambda\,\underline{\boldsymbol{f}}\big(\underline{\boldsymbol{q}}\big)$$

→ Bei Kurven mit affin invarianter Basis kann die Kurve über die Kontrollpunkte transformiert werden

Affine Transformationen
Transformation von Normalen



- Affine Transformation erhalten nicht die Winkel zwischen Vektoren. Damit transformierte Normalenvektoren nach der Transformation immer noch senkrecht auf der Fläche stehen müssen sie mit der inverstransponierten Transformationsmatrix multipliziert werden:

$$\vec{\boldsymbol{n}}' = \left(\boldsymbol{M}^{-1}\right)^T \hat{\boldsymbol{n}}$$

- Die so transformierte Normale steht wieder senkrecht zur Fläche, was aus der Betrachtung eines Differenzvektors in der Fläche folgt:

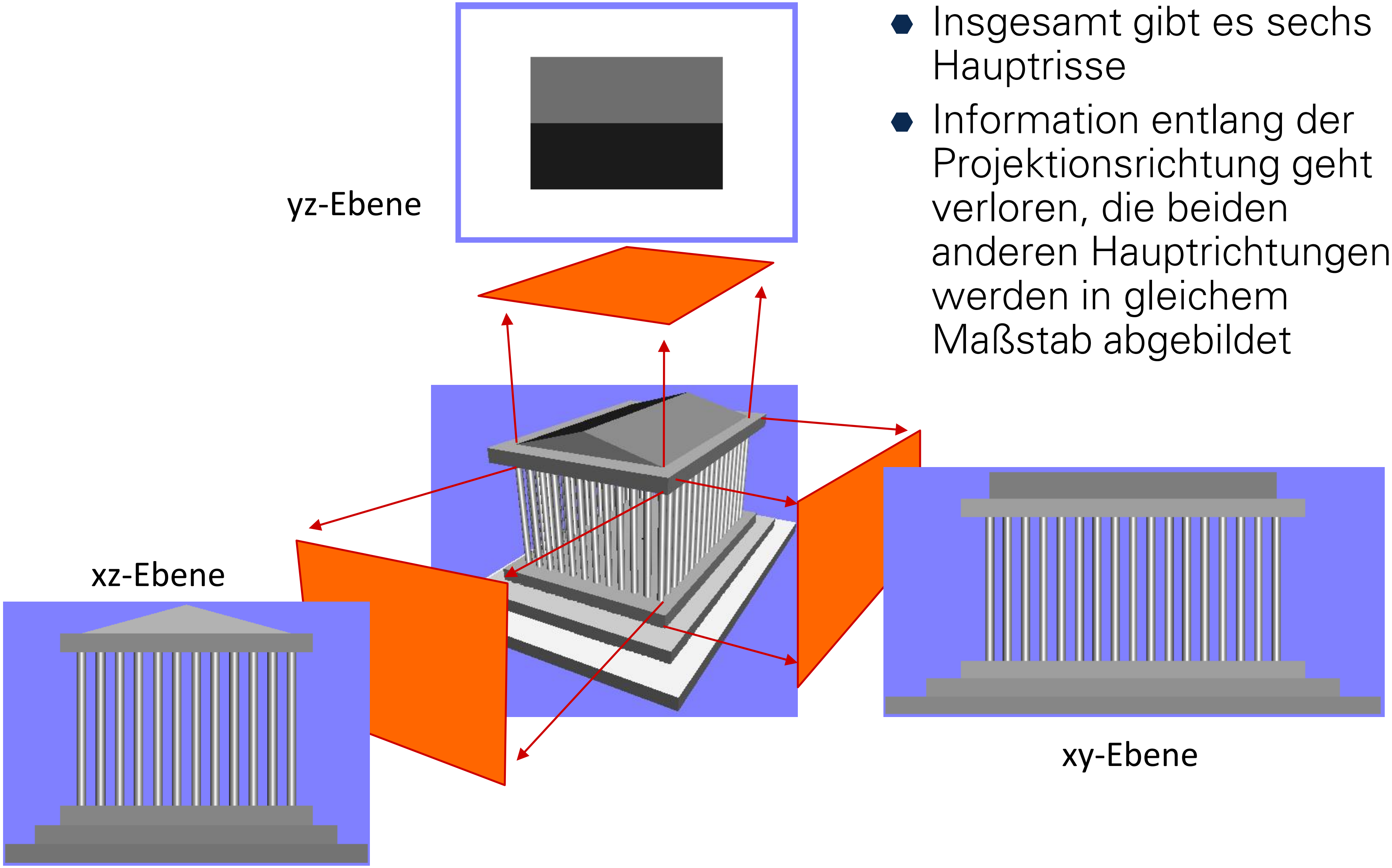
$$\vec{\boldsymbol{v}} = \underline{\boldsymbol{x}} - \underline{\boldsymbol{p}} \text{ mit } \vec{\boldsymbol{v}} \perp \hat{\boldsymbol{n}}, \text{ d.h. } \langle \vec{\boldsymbol{v}}, \hat{\boldsymbol{n}} \rangle = 0$$

$$\vec{\boldsymbol{v}}' = \boldsymbol{M}\vec{\boldsymbol{v}} \Rightarrow \langle \vec{\boldsymbol{v}}', \hat{\boldsymbol{n}}' \rangle = \left(\boldsymbol{M}\vec{\boldsymbol{v}}\right)^T \left(\boldsymbol{M}^{-1}\right)^T \hat{\boldsymbol{n}} = \vec{\boldsymbol{v}}^T \boldsymbol{M}^T \left(\boldsymbol{M}^{-1}\right)^T \hat{\boldsymbol{n}} = \vec{\boldsymbol{v}}^T \hat{\boldsymbol{n}} = \langle \vec{\boldsymbol{v}}, \hat{\boldsymbol{n}} \rangle = 0$$

- Die transformierte Normale muss abschließend normiert werden

ANSICHTSTRANSFORMATIONEN

Ansichtstransformationen Parallelprojektion

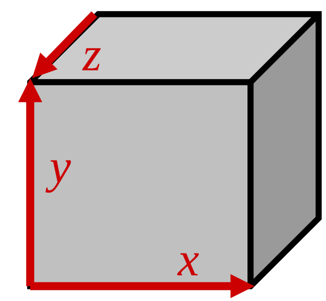


- Insgesamt gibt es sechs Hauptrisse
- Information entlang der Projektionsrichtung geht verloren, die beiden anderen Hauptrichtungen werden in gleichem Maßstab abgebildet

Ansichtstransformationen
Übersicht Klassischer Projektionen

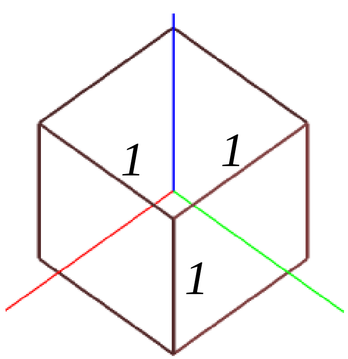
- ebene Projektionen
  - parallel
    - rechtwinklig
      - Hauptrisse
      - Axiometrie
        - isometrisch
        - dimetrisch
        - trimetrisch
    - schiefwinklig
      - Kavalier
      - Kabinett
  - perspektivisch
    - 1-Punkt
    - 2-Punkt
    - 3-Punkt

- Objekt hat drei orthogonale Hauptrichtungen (z.B. Quader, Haus, Auto, …)
- Koordinatensystem wird an Hauptrichtungen ausgerichtet

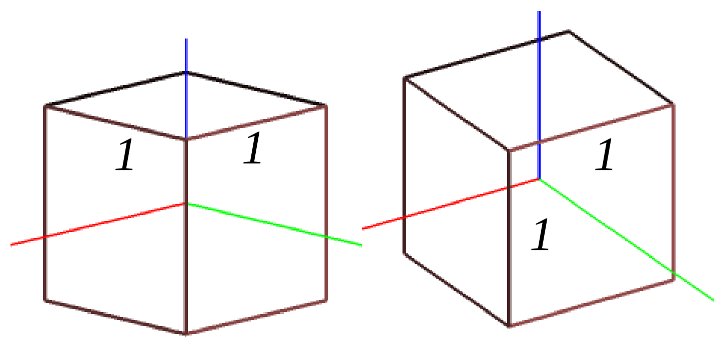


Ansichtstransformationen
Rechtwinklige Projektion

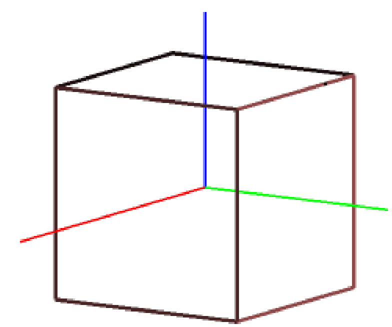
- Projiziert man nicht entlang einer der Hauptrichtungen, so unterscheidet man
  - isometrisch Projektion … wenn alle Hauptachsen im selben Verhältnis abgebildet werden
  - dimetrische Projektion … wenn zwei Hauptachsen im selben Verhältnis abgebildet werden
  - trimetrische Projektion … wenn alle Hauptachsen in unterschiedlichem Verhältnis abgebildet werden



isometrisch



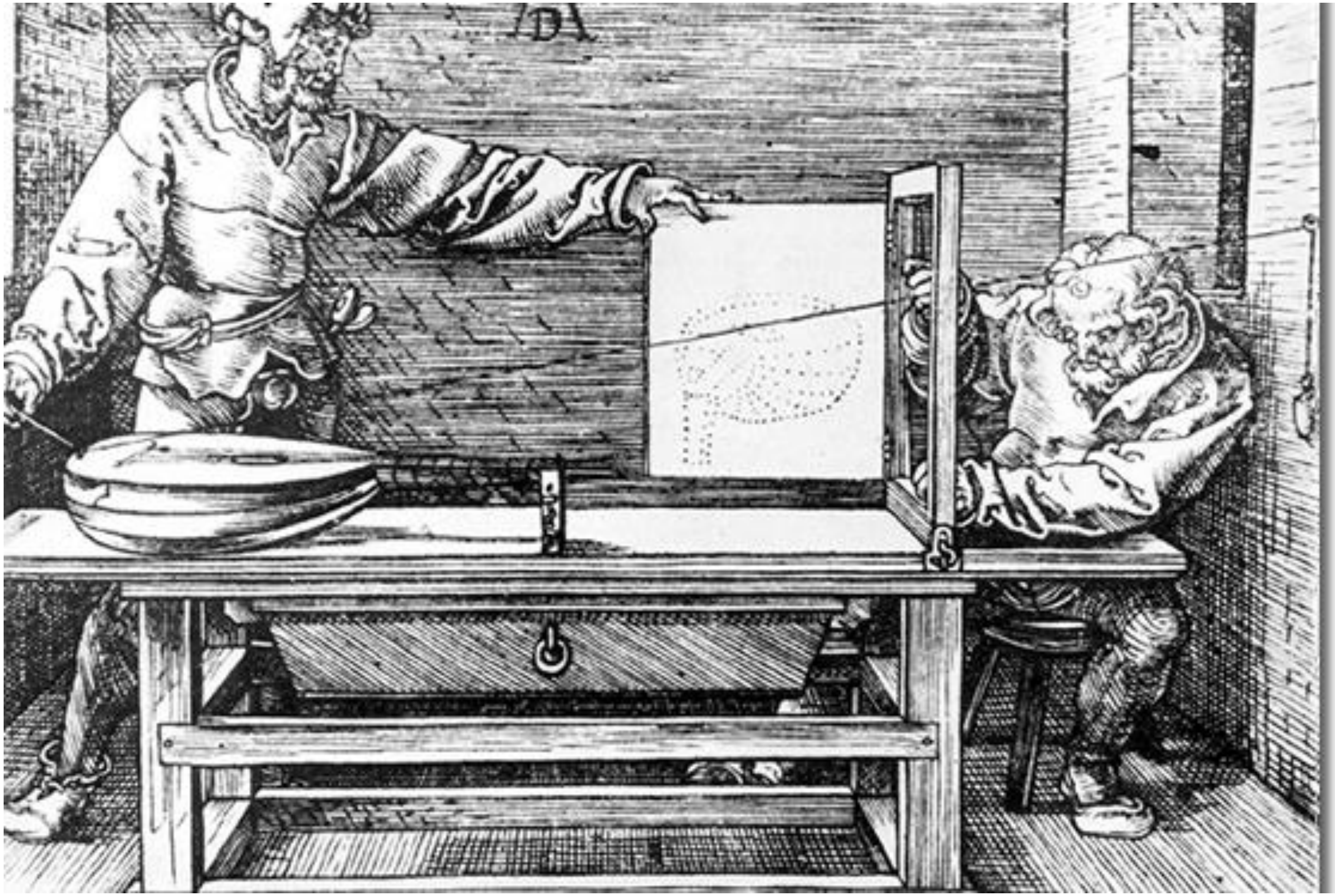
dimetrisch



trimetrisch

Ansichtstransformationen
Perspektivische Projektion

Computergraphik und Visualisierung



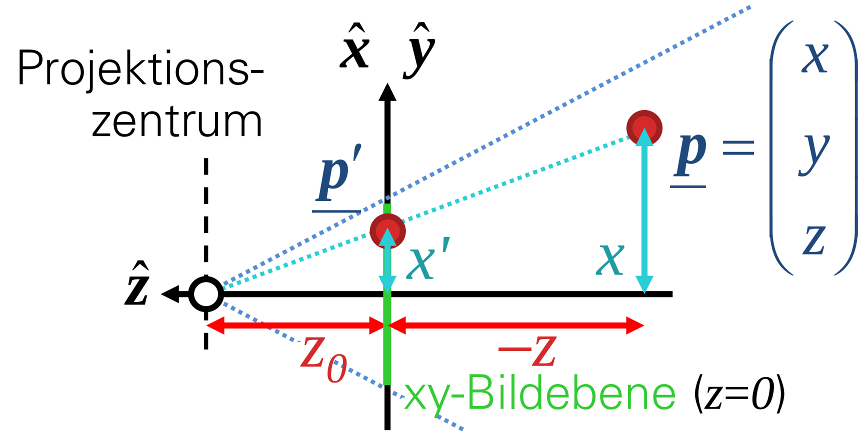
Albrecht Dürer, Underweysung der Messung, mit dem Zirckel und Richtscheyt, in Linien, Ebenen unnd gantzen corporen/Viertes Buch, 1525

Ansichtstransformationen
Perspektivische Projektion

- Um Punkte entlang von Strahlen durch den Augpunkt auf die Bildebene zu projizieren, benötigt man rationale Transformationen der Form (im folgenden 1D):

$$x' = f(x) = \frac{ax+b}{cx+d}$$

- Auch diese können mit Hilfe der homogenen Matrizenschreibweise repräsentiert werden
- Dazu wird die w-Komponente als Nenner interpretiert, der den x,y und z-Komponenten gemein ist



$$x' = \frac{z_0 x}{z_0 - z} \qquad y' = \frac{z_0 y}{z_0 - z} \qquad z' = 0$$

$$\begin{bmatrix} z_0 x \\ z_0 y \\ 0 \\ z_0 - z \end{bmatrix} = \begin{bmatrix} z_0 & 0 & 0 & 0 \\ 0 & z_0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & z_0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

HOMOGENE DARSTELLUNG

Homogene Darstellung Rationale Zahlen im 1D

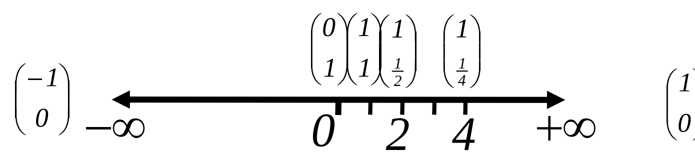
- Die Homogene Darstellung kann auch verwendet werden, um Quotienten zu repräsentieren. In der x-Komponente wird der Zähler und in der w-Komponente der Nenner gespeichert.
- Der Wert eines Bruchs ändert sich nicht, wenn man Zähler & Nenner mit einer Zahl ≠0 multipliziert.
- Deshalb werden alle homogene Vektoren, die sich nur durch einen skalaren Faktor unterscheiden identifizieren (≈)
- Im 1D repräsentieren alle homogenen Vektoren mit einer 0 in der w-Komponente unendlich – es ist jedoch –∞ und +∞ zu identifizieren
- (0,0), d.h. 0 durch 0, ist kein gültiger homogener Vektor

$$q = \frac{x}{w} \Rightarrow \tilde{q} = \begin{pmatrix} x \\ w \end{pmatrix}$$

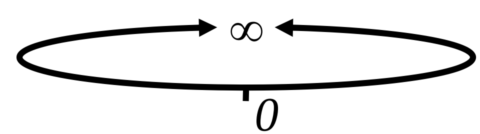
Repräsentation eines Bruchs

$$\forall \lambda \neq 0 : q = \frac{x}{w} = \frac{\lambda \cdot x}{\lambda \cdot w} \Rightarrow \begin{pmatrix} x \\ w \end{pmatrix} \approx \lambda \cdot \begin{pmatrix} x \\ w \end{pmatrix}$$

homogene Vektoren kann man mit Skalar multiplizieren



reale Zahlen



homogene Zahlen

Homogene Darstellung
Rationale Funktionen im 1D

Computergraphik und Visualisierung

- Rationale Funktionen können mit homogenen Matrizen dargestellt werden. Wieder werden skalare Vielfache identifiziert.
- Die Anwendung einer rationalen Funktion auf einen homogenen Vektor ergibt sich wieder durch Matrix-Vektor-Multiplikation
- Die Verkettung von zwei rationalen Abbildungen ergibt sich durch Matrix-Matrix-Multiplikation

$$f(q) = \frac{aq+b}{cq+d} \Rightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix} \approx \lambda \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Repräsentation rationaler Funktionen

$$\frac{aq+b}{cq+d} \Rightarrow \begin{pmatrix} ax/w+b \\ cx/w+d \end{pmatrix} \approx$$

$$\begin{pmatrix} ax+bw \\ cx+dw \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix}$$

Anwendung auf einen homogenen Vektor

$$f_1(q) = \frac{a_1 q + b_1}{c_1 q + d_1} \Rightarrow \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}, \; f_2(q) = \frac{a_2 q + b_2}{c_2 q + d_2} \Rightarrow \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix}$$

$$f_1\left(f_2(q)\right) = \frac{a_1 \frac{a_2 q + b_2}{c_2 q + d_2} + b_1}{c_1 \frac{a_2 q + b_2}{c_2 q + d_2} + d_1} \Rightarrow \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix} \cdot \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix}$$

Verkettung von rationalen Funktionen

PERSPEKTIVISCHE TRANSFORMATIONEN

Perspektivische Transformationen Zweidimensional

Computergraphik und Visualisierung

- Erweitert man rationale Funktionen auf 2 oder 3 Dimensionen, wird nur eine w-Komponente für einen gemeinsamen Nenner hinzugefügt
- Mit der homogenen Matrix-Darstellung können nun lineare, affine und perspektivische Transformation repräsentiert werden
- Homogene Vektoren mit 0 in der w-Komponente sind keine Differenzvektoren mehr, sondern Punkte, die in der Richtung des Vektors ins Unendliche geschoben sind.
- Perspektivische Transformation von Punkten:

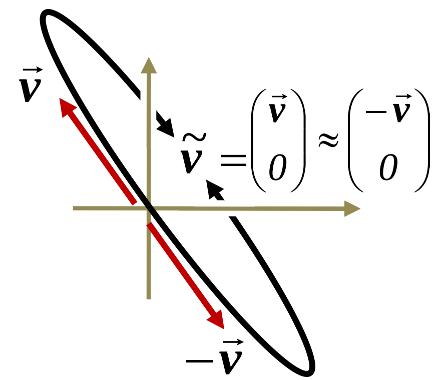
Lineare Transformationen

$$\tilde{\boldsymbol{M}} = \begin{pmatrix} M_{11} & M_{12} & t_x \\ M_{21} & M_{22} & t_y \\ p_x & p_y & 1 \end{pmatrix}$$

Translationen

Perspektivische Transformationen

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \approx \begin{pmatrix} -x \\ -y \\ 0 \end{pmatrix}$$



homogene Vektoren mit w=0 sind entlang ±Richtung in xy-Komponente unendlich weit entfernte Punkte

$$\underline{\boldsymbol{p}} = \begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \tilde{\boldsymbol{p}} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \tilde{\boldsymbol{p}}' = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \tilde{\boldsymbol{M}}\tilde{\boldsymbol{p}} \Rightarrow \underline{\boldsymbol{p}}' = \begin{pmatrix} x'/w' \\ y'/w' \end{pmatrix}$$

1. Homogenisierung
2. Transformation
3. w-Clip

Perspektivische Transformationen Dreidimensional

Computergraphik und Visualisierung

- Die perspektivische Projektion auf die Bildebene kann als Verkettung einer (umkehrbaren) perspektivischen Abbildung gefolgt von einer Projektion entlang der z-Richtung interpretiert werden
- Zur Definition der perspektivischen Abbildung nutzt man direkt den Matrixeintrag p_z in Spalte z und Zeile w. Ohne Perspektive ist dieser Eintrag 0, wohingegen z_0 unendlich wird.
- Entsprechend werden $\tilde{\boldsymbol{P}}_x$ und $\tilde{\boldsymbol{P}}_y$ definiert

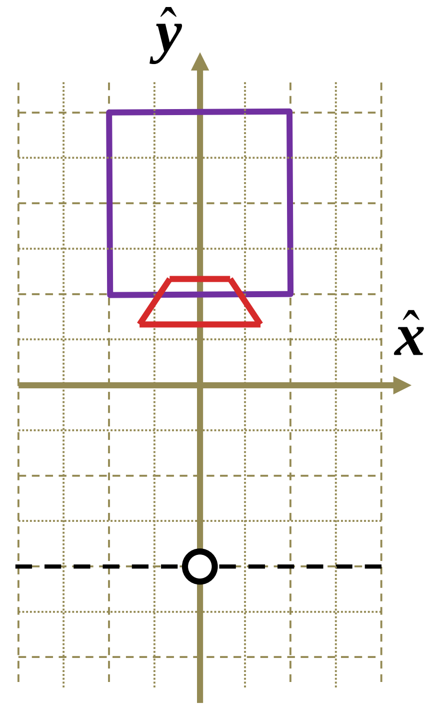
$$\begin{bmatrix} z_0 & 0 & 0 & 0 \\ 0 & z_0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & z_0 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{z_0} & 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\tilde{\boldsymbol{\Pi}}_z} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{z_0} & 1 \end{bmatrix}}_{\tilde{\boldsymbol{P}}_z(p_z)}$$

$$\text{mit } p_z = \frac{-1}{z_0}$$

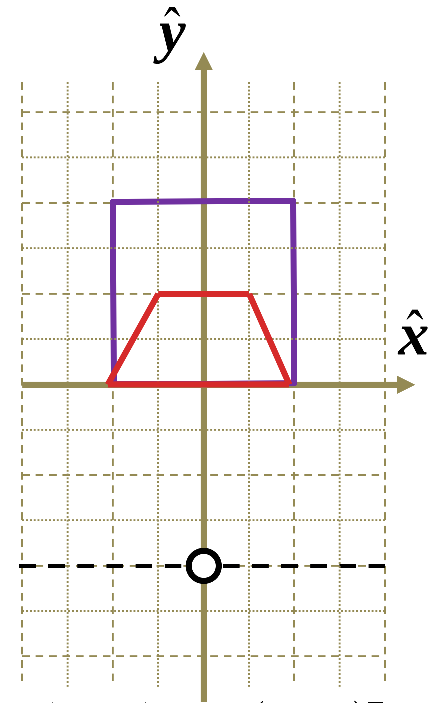
Perspektivische Transformationen Graphische Illustration im 2D – 1

- Die Wirkung einer perspektivischen Transformation $\tilde{\boldsymbol{P}}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix}$ kann man untersuchen, wenn man eine Referenzgeometrie abbildet – in den Beispielen wird das lila Quadrat auf das rote Viereck abgebildet
- Bei anschließender Projektion auf die Bildebene entsteht das Bild einer perspektivischen Projektion
- Punkte auf der x-Achse bleiben an der selben Stelle und heißen Fixpunkte



$$(1 \quad 3)^T \rightarrow \left(\tfrac{2}{5} \quad \tfrac{6}{5}\right)^T$$

$$(1 \quad 1)^T \rightarrow \left(\tfrac{2}{3} \quad \tfrac{2}{3}\right)^T$$



$$(1 \quad 2)^T \rightarrow \left(\tfrac{1}{2} \quad 1\right)^T$$

$$(1 \quad 0)^T \rightarrow (1 \quad 0)^T$$

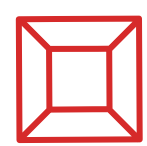


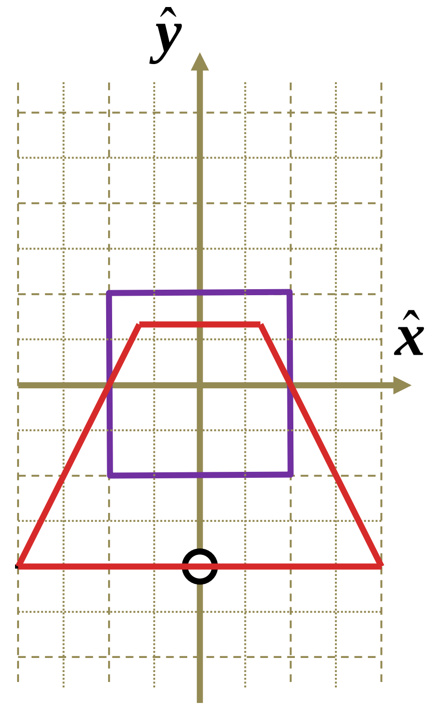
Bild nach Projektion eines Würfels in 3D:

Perspektivische Transformationen
Graphische Illustration im 2D – 2

Computergraphik und Visualisierung

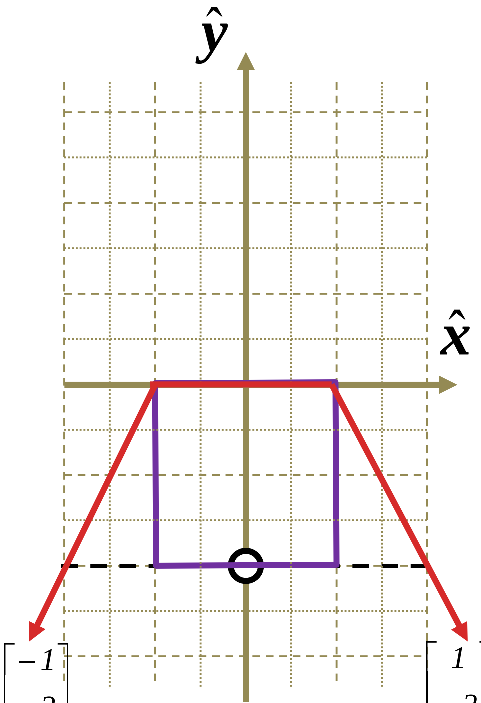
$$\tilde{\boldsymbol{P}}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix}$$

- Punkte, die zwischen Auge und x-Achse liegen werden nach außen und unten geschoben
- Punkte auf der zur x-Achse parallelen Gerade durch den Augpunkt werden auf Punkte im Unendlichen abgebildet



$$(1 \quad 1)^T \rightarrow (\tfrac{2}{3} \quad \tfrac{2}{3})^T$$

$$(1 \quad -1)^T \rightarrow (2 \quad -1)^T$$



$$\begin{bmatrix} -1 \\ -2 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

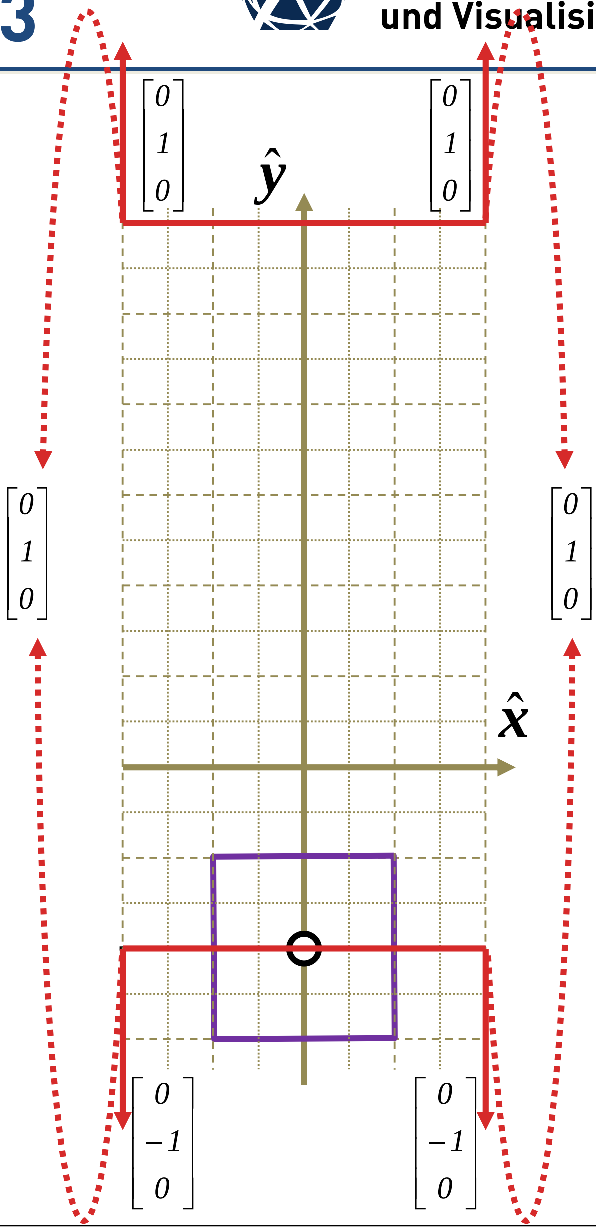
$$(1 \quad 0)^T \rightarrow (1 \quad 0)^T$$

$$[1 \quad -2 \quad 1]^T \rightarrow [1 \quad -2 \quad 0]$$

Perspektivische Transformationen
Graphische Illustration im 2D – 3

$$\tilde{\boldsymbol{P}}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix}$$

- Geraden werden zwar stets auf Geraden abgebildet; Strecken können jedoch auseinander gerissen werden, wenn ein Endpunkt vor und der andere hinter dem Auge liegt
- Um zu vermeiden, dass nach der perspektivischen Abbildung Objekte von hinter dem Auge vor das Auge gebracht werden, muss vorher an der near-clipping Ebene geclippt werden.



Perspektivische Transformationen
Transformation von Ebenen

Computergraphik und Visualisierung

- Eine Ebene in Hesse 'scher Normalform kann mit Hilfe von homogenen Vektoren durch ein Skalarprodukt definiert werden
- D.h. die Ebene kann mit einem homogenen Vektor repräsentiert werden, der die Komponenten des Normalenvektors und den Abstand zum Ursprung enthält
- Ähnlich wie im affinen Fall kann man zeigen, dass sich die homogene Ebenendarstellung mit der Inverstransponierten transformiert:

$$n_x x + n_y y + n_z z + d = 0$$

$$\tilde{\boldsymbol{n}} = \begin{pmatrix} n_x \\ n_y \\ n_z \\ d \end{pmatrix} \quad \tilde{\boldsymbol{x}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \Rightarrow \tilde{\boldsymbol{n}}^T \tilde{\boldsymbol{x}} = 0$$

$$\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{M}} \tilde{\boldsymbol{x}}, \quad \tilde{\boldsymbol{n}}' = \left(\tilde{\boldsymbol{M}}^{-1}\right)^T \tilde{\boldsymbol{n}}$$

$$\Rightarrow \tilde{\boldsymbol{n}}'^T \tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{n}}^T \tilde{\boldsymbol{M}}^{-1} \tilde{\boldsymbol{M}} \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{n}}^T \tilde{\boldsymbol{x}}$$

$$\left.\begin{matrix} \hat{\boldsymbol{n}} = \begin{pmatrix} n_x \\ n_y \end{pmatrix} \\ d \end{matrix}\right\} \Rightarrow \tilde{\boldsymbol{n}} = \begin{pmatrix} n_x \\ n_y \\ d \end{pmatrix} \Rightarrow \tilde{\boldsymbol{n}}' = \begin{pmatrix} n'_x \\ n'_y \\ d' \end{pmatrix} = \left(\tilde{\boldsymbol{M}}^{-1}\right)^T \tilde{\boldsymbol{n}} \Rightarrow \left(\begin{matrix} \hat{\boldsymbol{n}}' = \begin{pmatrix} n'_x / l \\ n'_y / l \end{pmatrix}, l = \sqrt{n'^2_x + n'^2_y} \\ d' = d / l \end{matrix} \right.$$

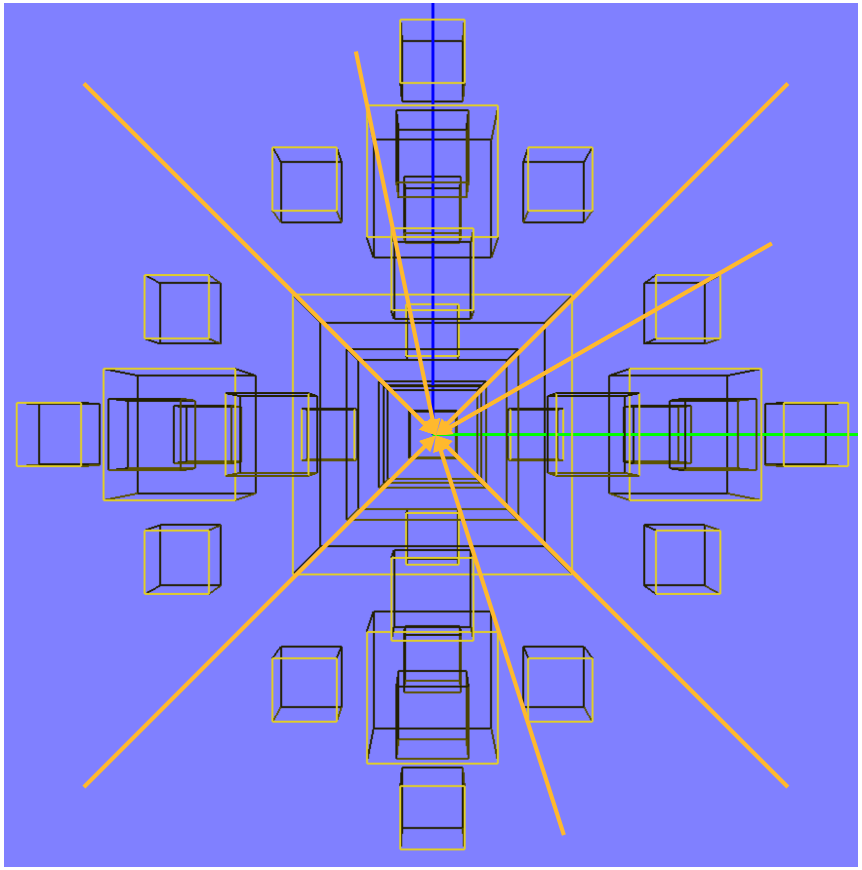
Perspektivische Transformation von Ebenen (nicht im OpenGL-Standard)

FLUCHTPUNKTE

Fluchtpunkte
Definition

Computergraphik und Visualisierung

- Nach einer perspektivischen Abbildung können sich zuvor parallele Linien in einem Punkt schneiden.
- Ist dies der Fall nennt man den Schnittpunkt Fluchtpunkt.
- Jede Raumrichtung kann einen unterschiedlichen Fluchtpunkt haben.
- Man spricht in der klassischen Projektion von 1-, 2- oder 3-Punktperspektive, wenn es für 1, 2 oder 3 Hauptrichtungen einen so genannten Hauptfluchtpunkt gibt



Zentralperspektive: alle Linien, die parallel zur Tiefenrichtung sind, schneiden sich im Bildmittelpunkt. Die Linien parallel zu den anderen Hauptrichtungen bleiben parallel.

Fluchtpunkte Berechnung

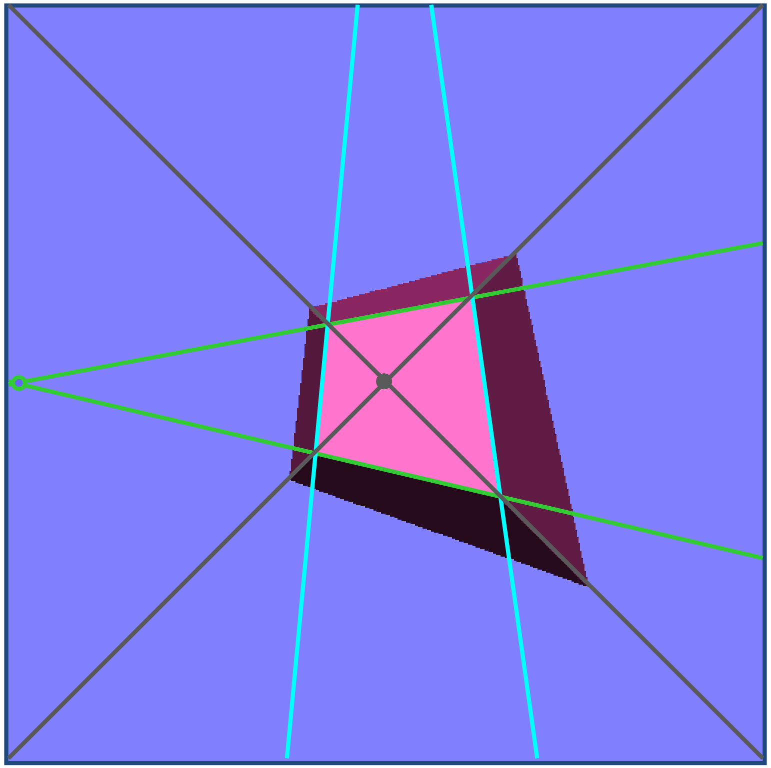
- Für eine gegebe Richtung $\vec{v}$ kann man den Fluchtpunkt $\underline{f}_{\vec{v}}$ einfach mit Hilfe der Abbildungsmatrix $\tilde{P}(\vec{p})$ berechnen:

$$\tilde{f}_{\vec{v}} = \tilde{P}(\vec{p})\tilde{v} \text{ mit } \tilde{v} = \begin{pmatrix} \vec{v} \\ 0 \end{pmatrix}$$

- Beispiel x-Hauptfluchtpunkt:

$$\tilde{f}_{\hat{x}}\left(\tilde{P}(\vec{p})\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ p_x \end{bmatrix}$$

$$\Rightarrow \underline{f}_{\hat{x}}\left(\tilde{P}(\vec{p})\right) = \begin{pmatrix} 1/p_x \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_0 \\ 0 \\ 0 \end{pmatrix}$$



3-Punktperspektive: x- und z-Hauptrichtungen haben einen Fluchtpunkt im Bild, der y-Fluchtpunkt liegt oberhalb des Bildes.

➡ In den Spalten einer homogenen Matrix stehen die homogenen Fluchtpunkte der Hauptachsen und das homogene Bild des Ursprungs

VERGLEICH

Vergleich Transformationsformeln

Computergraphik und Visualisierung

	Linear	Affin	Perspektivisch
repräsentierbare Elemente	• Vektoren • Ortsvektoren	• Punkte • Vektoren	• Punkte (evtl. im unendlichen) • Ebenen
Darstellung der Abbildung	• Matrix	• homogene Matrix	• homogene Matrix
Unterstützte Transformationen	• Skalierung, Spiegelung, Projektion, Rotation, Scherung	• Zusätzlich Translation	• zusätzlich perspektivische Transformation
Transformationsformel für Raumelemente bzw. Vektoren	$\vec{v}' = M\vec{v}$	$\begin{pmatrix} \mathbf{p}' \\ 1 \end{pmatrix} = \tilde{M} \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}$ $\begin{pmatrix} \vec{v}' \\ 0 \end{pmatrix} = \tilde{M} \begin{pmatrix} \vec{v} \\ 0 \end{pmatrix}$	$\begin{pmatrix} w' \cdot \mathbf{p}' \\ w' \end{pmatrix} = \tilde{M} \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}$
Transformationsformel Normalen / Ebenen	$\vec{n}' = \left(M^{-1}\right)^T \hat{n}$	$\begin{pmatrix} \vec{n}' \\ 0 \end{pmatrix} = \left(\tilde{M}^{-1}\right)^T \begin{pmatrix} \hat{n} \\ 0 \end{pmatrix}$	$\begin{pmatrix} \vec{n}' \\ d' \end{pmatrix} = \left(\tilde{M}^{-1}\right)^T \begin{pmatrix} \hat{n} \\ d \end{pmatrix}$

Vergleich Erhaltungsgrößen

	Rotation	Scherung	linear	Translation	affin	perspektiv
Längen	✓			✓		
Winkel	✓			✓		
Flächen	✓	✓		✓		
Längenverhältnisse und affine Koordinaten	✓	✓	✓	✓	✓	
Reihenfolge von Punkten auf Geraden	✓	✓	✓	✓	✓	
Verhältnisse von Verhältnissen	✓	✓	✓	✓	✓	✓
Geraden, Ebenen	✓	✓	✓	✓	✓	✓
Geradensegmente	✓	✓	✓	✓	✓	

$$\boldsymbol{M}^{-1} = \boldsymbol{M}^{T}$$

$$\det \boldsymbol{M} = \pm 1$$