

## 6. Koordinaten, Transformationen, Perspektive

Bei Grafik-Applikationen sind neben Modellierung der darzustellenden Objekte und Algorithmen zum Zeichnen auch *Manipulationsmöglichkeiten* von großer Bedeutung:

- Animation von Objekten (kontinuierliche Veränderung der Lage und Größe)
- interaktives Editieren (Drehen, Ändern der Größe, Spiegeln, Verzerren...)

Behandlung von Transformationen mittels vektorieller Geometrie (lineare Algebra)

$$\vec{v} \in \mathbb{R}^n \Leftrightarrow \vec{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} \quad \text{mit } v_i \in \mathbb{R} \quad (i = 0; 1; \dots; n-1)$$

Die meisten wichtigen Transformationen sind mathematisch *lineare Abbildungen*:

Eigenschaft:  $f(cx+y) = c f(x) + f(y)$

Eine lineare Abbildung ist vollständig bekannt, wenn die Bilder der Vektoren einer *Basis* bekannt sind (z.B. die Bilder der

Standard-Basisvektoren  $\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \dots, \vec{e}_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$ ).

Darstellung der linearen Abb.  $f$  durch ihre zugehörige *Matrix*  $M_f$ :  
in den Spalten von  $M_f$  stehen die Bilder der Standard-Basisvektoren unter  $f$ .

Allgemein besteht eine Matrix aus  $p \times q$  Skalaren  $m_{ij}$ .

$$M = \begin{pmatrix} m_{00} & m_{01} & \cdots & m_{0,q-1} \\ m_{10} & m_{11} & \cdots & m_{1,q-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{p-1,0} & m_{p-1,1} & \cdots & m_{p-1,q-1} \end{pmatrix} = (m_{ij})$$

Anwendung einer linearen Abbildung  $f$  auf einen Vektor  $\vec{v}$  durch *Multiplikation* der Matrix  $M_f$  mit dem Vektor (dabei Schreibweise des Vektors als Spaltenvektor und Multiplikation der Matrix *von links* an den Vektor wichtig!):

$$\vec{w} = f(\vec{v}) = M \cdot \vec{v} = \begin{pmatrix} m_{00} & \cdots & m_{0,q-1} \\ \vdots & \ddots & \vdots \\ m_{p-1,0} & \cdots & m_{p-1,q-1} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ \vdots \\ v_{q-1} \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{q-1} m_{0,k} \cdot v_k \\ \vdots \\ \sum_{k=0}^{q-1} m_{p-1,k} \cdot v_k \end{pmatrix} = \begin{pmatrix} w_0 \\ \vdots \\ w_{p-1} \end{pmatrix}$$

Nacheinander-Anwendung (*Komposition*) zweier linearer Abbildungen  $f \circ g$ : wende erst  $g$  an, dann  $f$ .

$$f \circ g(x) = f(g(x))$$

Die Komposition wird beschrieben durch das *Produkt* der zugehörigen Matrizen:

$$M_{f \circ g} = M_f \cdot M_g$$

mit

$$M \cdot N = \begin{pmatrix} m_{00} & \cdots & m_{0,q-1} \\ \vdots & \ddots & \vdots \\ m_{p-1,0} & \cdots & m_{p-1,q-1} \end{pmatrix} \cdot \begin{pmatrix} n_{00} & \cdots & n_{0,r-1} \\ \vdots & \ddots & \vdots \\ n_{q-1,0} & \cdots & n_{q-1,r-1} \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{q-1} m_{0,k} n_{k,0} & \cdots & \sum_{k=0}^{q-1} m_{0,k} n_{k,r-1} \\ \vdots & \ddots & \vdots \\ \sum_{k=0}^{q-1} m_{p-1,k} n_{k,0} & \cdots & \sum_{k=0}^{q-1} m_{p-1,k} n_{k,r-1} \end{pmatrix}$$

Beachte: Das Matrizenprodukt ist nichtkommutativ (d.h. i. allg. ist  $N \cdot M \neq M \cdot N$ ).

Lineare Abbildungen eines Vektorraums (z.B. des  $\mathbb{R}^2$ ) in sich selbst heißen auch *Endomorphismen*. Darstellung durch  $n \times n$ -Matrizen ( $n$  = Dimension des Vektorraumes).

Eindeutig umkehrbare (bijektive) Endomorphismen heißen *Automorphismen*. Ihnen entsprechen die *regulären* Matrizen:

$$f \text{ bijektiv} \Leftrightarrow M_f \text{ regulär} \Leftrightarrow \det(M_f) \neq 0$$

Die regulären  $n \times n$ -Matrizen (bzw. die Automorphismen des  $\mathbb{R}^n$ ) bilden eine Gruppe, die *allgemeine lineare Gruppe*  $GL(n)$ .

(math. Begriff der *Gruppe*:

- Abgeschlossenheit gegenüber der Verknüpfungsoperation
- Assoziativgesetz  $(MN)P = M(NP)$
- Existenz eines neutralen Elements  $E$ :  $EM = ME = M$

(hier die Einheitsmatrix  $E = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$ )

- zu jedem Element  $M$  Existenz eines Inversen  $M^{-1}$ .)

Beispiele für bijektive lin. Abbildungen: Drehungen um den Nullpunkt, zentrische Streckungen mit einem Faktor  $\neq 0$  vom Nullpunkt aus.

Beispiel für eine nicht-bijektive lin. Abb.: Projektion entlang der  $x$ -Achse auf die  $y$ -Achse.

Wichtiger Spezialfall der bijektiven lin. Abb.: *orthogonale Abbildungen*.

Entsprechend: orthogonale Matrizen.

$$\begin{aligned} M \text{ orthogonal} &\Leftrightarrow M^T \cdot M = E \text{ (d.h. die Transponierte ist die Inverse)} \\ &\Leftrightarrow \text{die Spaltenvektoren von } M \text{ bilden eine} \\ &\quad \text{Orthonormalbasis (paarw. senkrecht u. Länge 1)} \\ &\Leftrightarrow \forall \vec{x}, \vec{y} \in \mathbb{R}^n : \langle M\vec{x}, M\vec{y} \rangle = \langle \vec{x}, \vec{y} \rangle \text{ (Skalarprodukt-Invarianz)} \\ &\Leftrightarrow \forall \vec{x} \in \mathbb{R}^n : \|M\vec{x}\| = \|\vec{x}\| \text{ (Längen-Invarianz)} \end{aligned}$$

Für alle orthogonalen Matrizen hat ferner die Determinante den Betrag 1.

(Die Umkehrung gilt nicht: Beispiel Scherungsmatrizen.)

Die orthogonalen  $n \times n$ -Matrizen bilden eine Untergruppe von  $GL(n)$ , die *orthogonale Gruppe*  $O(n)$ .

Alle Elemente von  $O(n)$  mit Determinante +1 bilden wiederum eine Untergruppe von  $O(n)$ , die *spezielle orthogonale Gruppe*  $SO(n)$ .

Es gilt:

$M \in SO(n) \Leftrightarrow M$  ist Produkt einer geraden Anzahl von Spiegelungen (an Hyperebenen durch den Nullpunkt).

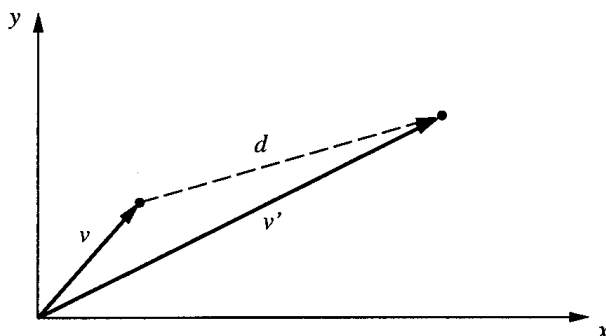
Beispiele:

- Drehungen um den Nullpunkt gehören zu  $SO(n)$ .
- Spiegelungen an Geraden durch den Nullpunkt (im  $\mathbb{R}^2$ ) bzw. an Ebenen durch den Nullpunkt (im  $\mathbb{R}^3$ ) sind orthogonal, aber gehören nicht zu  $SO(n)$ .
- Streckungen mit Streckfaktor  $c \neq 1$  sind nicht orthogonal.

Einschränkung: für jede lineare Abbildung  $f$  gilt  $f(\vec{0}) = \vec{0}$  (Nullpunkt ist Fixpunkt).

wichtige Transformationen, die nicht zu den linearen Abbildungen gehören:

Translationen (Verschiebungen).



Verschiebungen rechnerisch:

$v' = t(v) = v + d_t$  d.h. Addition eines konstanten Vektors  $d_t$ .

*Affine Abbildungen:*

alle Abbildungen  $a$ , die sich darstellen lassen als

$$a(v) = f(v) + d \quad \text{mit einer linearen Abbildung } f,$$

d.h. als Komposition einer linearen Abb. und einer Translation.

Es gilt:  $d = a(0)$ ,  $f(v) = a(v) - a(0) \Rightarrow f$  und  $d$  sind eindeutig bestimmt.

Beispiele:

- Drehungen um beliebige Punkte
- Streckungen mit beliebigen Streckzentren
- Schraubungen im  $\mathbb{R}^3$
- Spiegelungen an beliebigen Geraden im  $\mathbb{R}^2$  oder an beliebigen Ebenen im  $\mathbb{R}^3$
- Parallelprojektionen auf beliebige Ebenen im  $\mathbb{R}^3$

usw.

(lineare Abbildungen und Translationen als Spezialfälle mit enthalten!)

Die bijektiven affinen Abbildungen des  $\mathbb{R}^n$  in sich bilden eine Gruppe, die *affine Gruppe*  $\text{Aff}(n)$ .

Eine affine Abb.  $a$  heißt *Bewegung* oder *Kongruenzabbildung*, wenn ihre zugehörige lineare Abbildung  $f$  orthogonal ist, und *eigentliche Bewegung*, wenn  $f$  sogar in  $\text{SO}(n)$  liegt.

Beispiele:

Drehungen, Schraubungen, Translationen: eigentliche Bewegungen

Spiegelungen, Drehspiegelungen: uneigentliche Bewegungen

Streckungen: keine Bewegungen, da  $f(v) = c \cdot v$  nicht orthogonal (wenn  $|c| \neq 1$ ).

$\text{Bew}(n)$ : Gruppe der Bewegungen des  $\mathbb{R}^n$  (= Untergruppe von  $\text{Aff}(n)$ )

$\text{Bew}^+(n)$ : Untergruppe der eigentlichen Bewegungen.

Eine *Ähnlichkeitsabbildung* ist eine affine Abb.  $a$ , die Abstände stets um einen festen Faktor  $c > 0$  verändert:

$$d(a(u), a(v)) = c \cdot d(u, v).$$

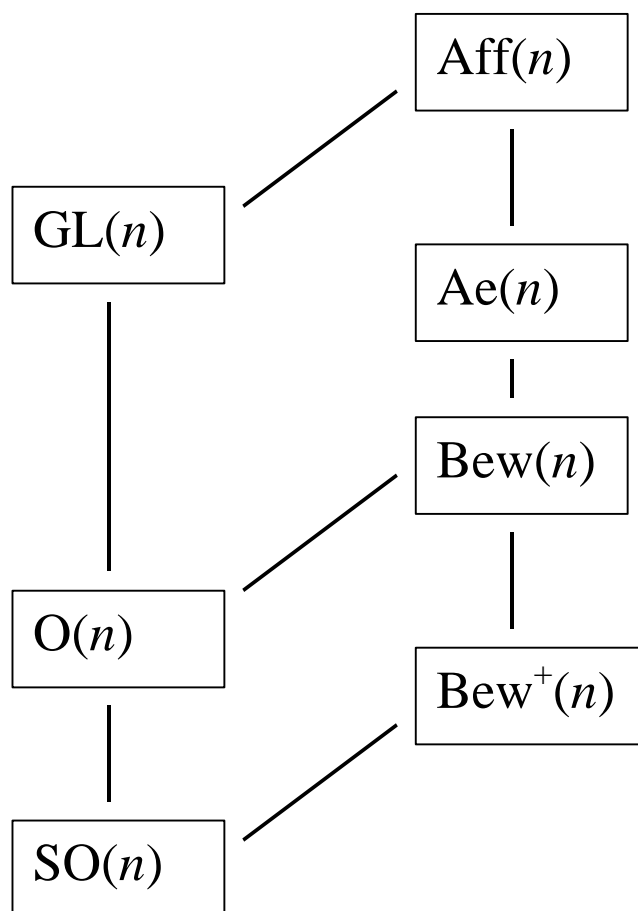
Ähnlichkeitsabbildungen liegen "zwischen" den affinen Abbildungen und den Kongruenzabbildungen (Bewegungen) (für Bewegungen ist  $c = 1$ ).

Jede Ähnlichkeitsabb. hat die Form  $a(v) = c \cdot M \cdot v + d$  mit einer orthogonalen Matrix  $M$  und  $c > 0$ .

$\text{Ae}(n)$ : Gruppe der Ähnlichkeitsabb. des  $\mathbb{R}^n$ .

Beispiele: Streckungen, Streckspiegelungen, Drehstreckungen.

Untergruppen-Diagramm:



Man kann affine Abbildungen aus bestimmten Grundoperationen zusammensetzen

Beispiel: Drehung um einen beliebigen, vorgegebenen Punkt  $P$  im  $\mathbb{R}^2$

- verschiebe den Drehpunkt in den Nullpunkt
- wende Drehung um den Nullpunkt an
- mache Verschiebung wieder rückgängig

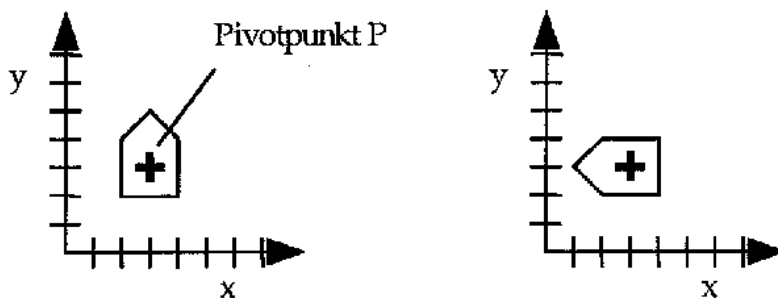
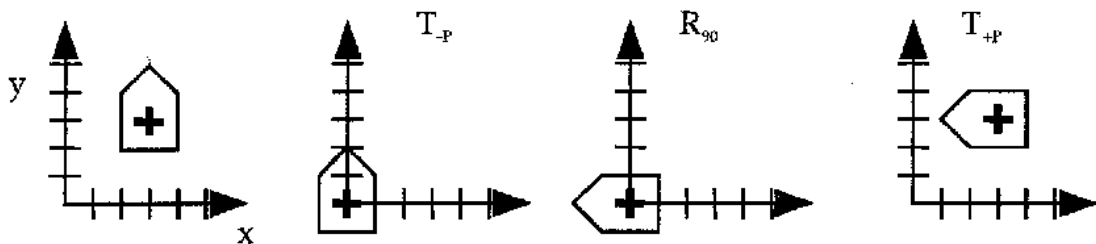


Abb. 2-20 Drehung um Fixpunkt



Man benötigt für Bewegungen im  $\mathbb{R}^2$ :

- alle Translationen
- Drehungen um den Nullpunkt um alle Winkel
- eine Spiegelung

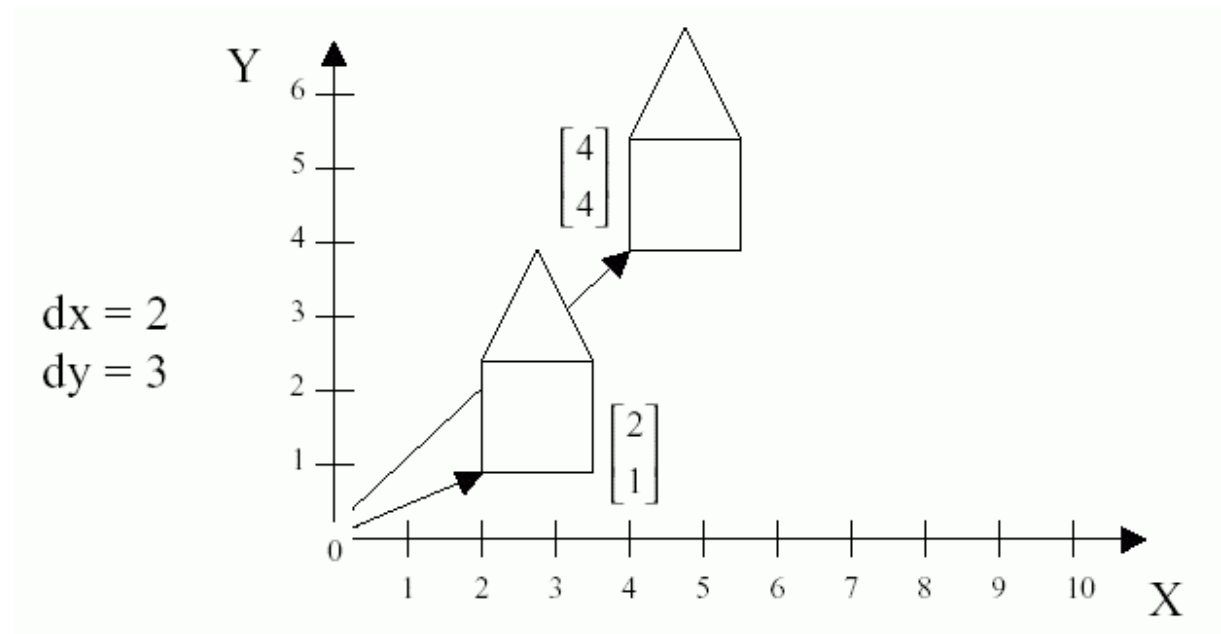
im  $\mathbb{R}^3$ :

- alle Translationen
- Drehungen um die 3 Koordinatenachsen (alle Winkel)
- eine Spiegelung an einer Ebene

Wichtige Grundoperationen im  $\mathbb{R}^2$  sind somit:

(a) Translationen

$$v' = v + d$$



(b) Drehungen (Rotationen)

$$v' = R_\theta v \text{ where } v = \begin{bmatrix} x \\ y \end{bmatrix}, \quad v' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

and

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

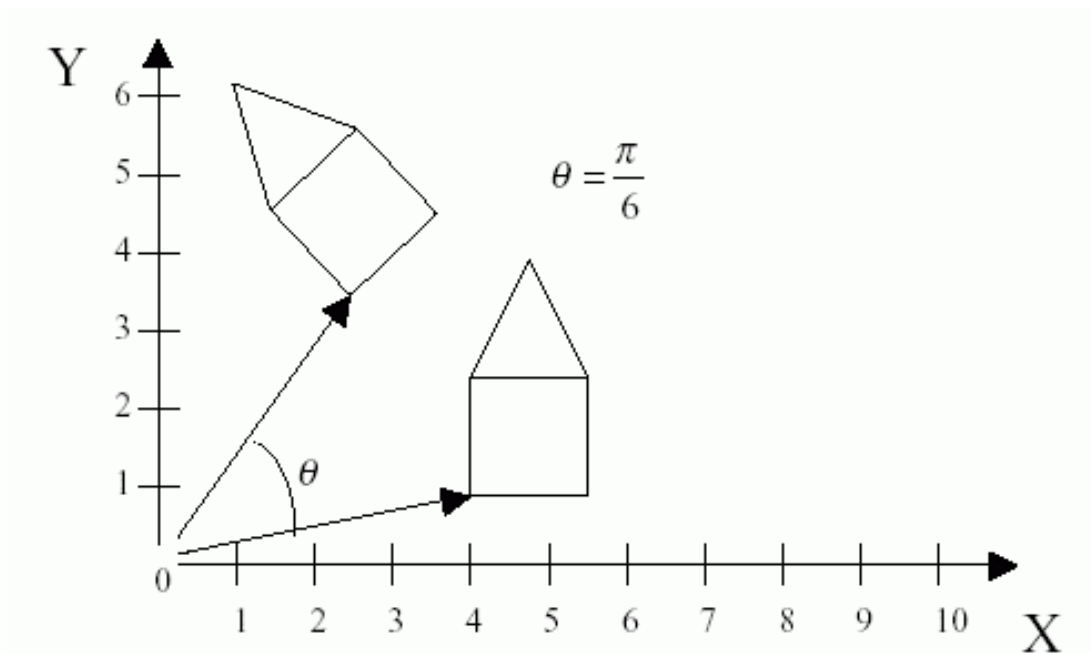
$\theta$  ist der Drehwinkel

Orientierung im math. positiven Sinne (gegen den Uhrzeigersinn)

Angabe manchmal auch in Bogenmaß:

$$\text{Winkel [Bogenmaß]} = \text{Winkel [Grad]} \cdot \pi / 180.$$





### (c) Spiegelungen

Spiegelung an der  $x$ -Achse:

Vorzeichenwechsel in der  $y$ -Koordinate

$$\text{also } v' = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot v$$

Spiegelung an der  $y$ -Achse:

$$\text{analog, } v' = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \cdot v$$

### (d) Skalierungsoperationen

oft mit unterschiedlichen Skalierungsfaktoren in  $x$ - und in  $y$ -Richtung gebraucht (keine Kongruenzabb.!):

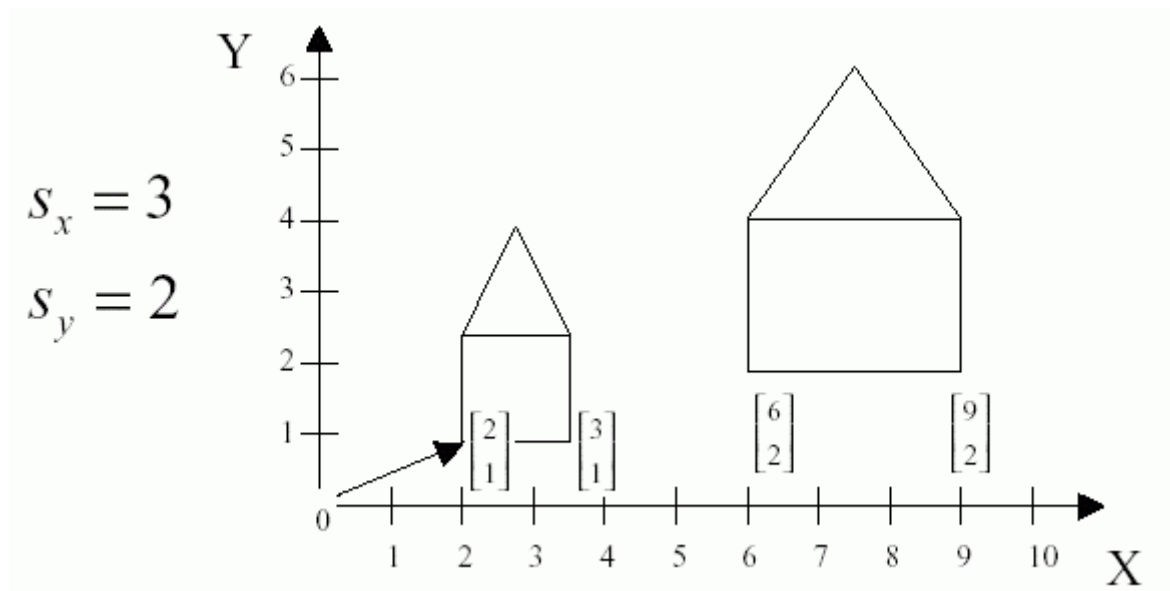
$$v' = Sv \text{ where } v = \begin{bmatrix} x \\ y \end{bmatrix}, \quad v' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\text{and } S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

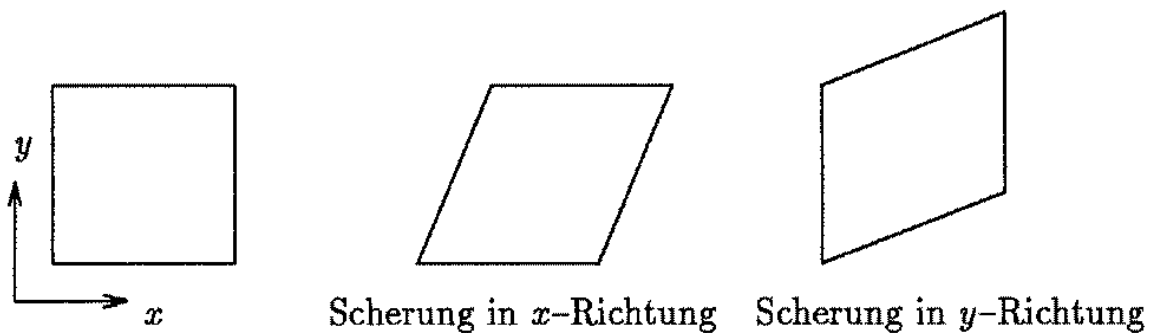
$$x' = s_x x$$

$$y' = s_y y$$

Proportionen und Winkel bleiben nur erhalten, wenn  $s_x = s_y$   
 (dann liegt eine Ähnlichkeitsabb. vor)  
 andernfalls Verzerrung des abgebildeten Objekts:



Manchmal benötigt man auch  
 (e) Scherungen (engl.: *shear*)



Scherungen sind flächenerhaltend, aber nicht orthogonal  
 (Figuren werden verzerrt).

Matrix einer Scherung in x-Richtung:

$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$$

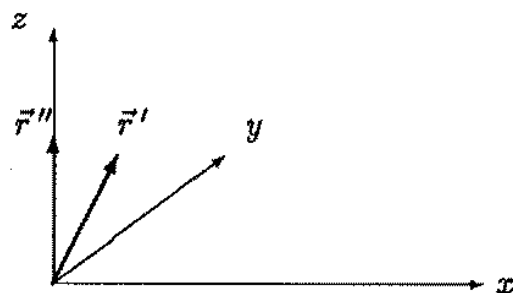
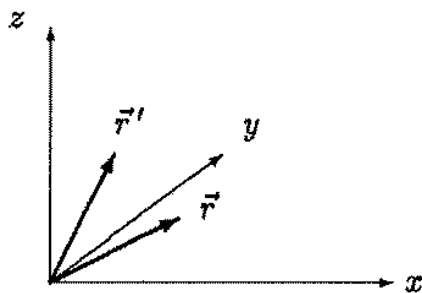
für die y-Richtung analog.

Im  $\mathbb{R}^3$  analoge Grundoperationen;

- anstelle der Spiegelung an Koordinatenachsen: Sp. an Koordinatenebenen ( $xy$ ,  $xz$ ,  $yz$ )
- anstelle der Drehung um den Nullpunkt: Drehungen um die Koordinatenachsen  $R_{x,\alpha}$ ,  $R_{y,\alpha}$ ,  $R_{z,\alpha}$

Rückführung einer Drehung um eine beliebige Achse  $AB$  mit Winkel  $q$  auf diese Grund-Drehungen:

1.	Translation der Achse um $-A$ zum Nullpunkt	$T_{-A}$
2.	Rotation der Drehachse $r$ in die $yz$ -Ebene (drehe um die $y$ -Achse mit Winkel $-\alpha$ )	$R_{y,-\alpha}$
3.	Rotiere die Drehachse auf die $z$ -Achse (drehe um die $x$ -Achse mit Winkel $\beta$ )	$R_{x,\beta}$
4.	Rotiere um die $z$ -Achse mit Winkel $q$	$R_{z,q}$
5.	Invertiere Transformation von Schritt 3	$R_{x,-\beta}$
6.	Invertiere Transformation von Schritt 2	$R_{y,\alpha}$
7.	Invertiere Transformation von Schritt 1	$T_A$



Veranschaulichung der Schritte 2 und 3.  $r$  ist der Richtungsvektor der Rotationsachse, um die die Rotation erfolgen soll. Links: Drehung um die  $y$ -Achse, die  $r$  in die  $yz$ -Ebene dreht (Ergebnis  $r'$ ). Rechts: Drehung von  $r'$  um die  $x$ -Achse, so dass er in Richtung der positiven  $z$ -Achse zeigt.

Die Winkel  $\alpha$  und  $\beta$  lassen sich durch trigonometrische Beziehungen aus den Koordinaten von  $A$  und  $B$  berechnen.

## Koordinatentransformationen

Oft muss man von einem Koordinatensystem in ein anderes umrechnen

Beispiel:

Weltkoordinaten → normierte Koordinaten → Gerätekoord.

Koordinatentransformation: komplementär zur Transformation von Objekten

Objekttransformation:	Objekt ändert seine Lage (und evtl. seine Form) Koordinatensystem bleibt fest
Koordinatentransformation:	Objekt bleibt fest Koordinatensystem ändert seine Lage (u. evtl. seine Form)

rechnerisch selber Formalismus:

Multiplikation des Koordinatenvektors mit Transformationsmatrix, ggf. + Translationsanteil (wenn Ursprünge der Koordinatensysteme unterschiedlich)

wenn Translationsanteil entfällt:

Spalten der Matrix = Basisvektoren des alten Koordinatensystems, ausgedrückt durch Koordinaten im neuen System

PostScript: Befehle **translate**, **rotate**, **scale** ändern das Koordinatensystem

Unterschiede zu Objekttransformationen:

- Reihenfolge von Objektkonstruktion und Transformation vertauscht
- Änderung des Koordinatensystems beeinflusst *alle* nachfolgenden Konstruktionen

## Homogene Koordinaten

Darstellung affiner Abbildungen und Koordinatentransformationen in der Form  $M \cdot v + d$  macht Berechnungen schwerfällig.

Idee: alles in der Form  $M \cdot v$  darstellen

⇒ bei Komposition von Transformationen nur noch Matrixmultiplikationen,  
einheitliche Form aller affinen Abb.

dies wird erreicht durch *Hinzunahme einer weiteren Koordinate*

- aus  $(x, y)$  wird  $(x, y, 1)$
- aus  $(x, y, z)$  wird  $(x, y, z, 1)$

dafür werden Vektoren, die sich nur um einen festen Faktor  $w \neq 0$  unterscheiden, miteinander identifiziert (Klassenbildung).

"*homogene Koordinaten*"

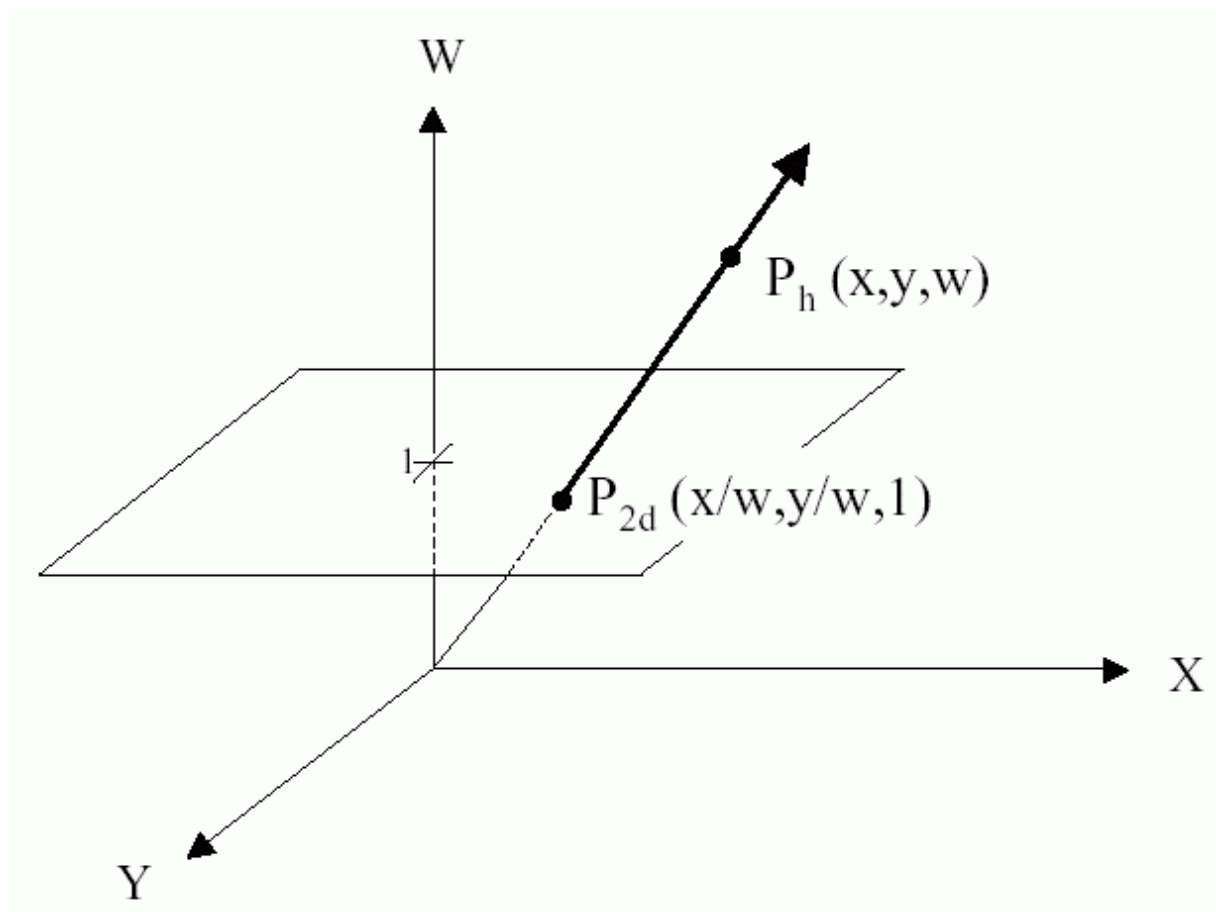
dem homogenen Vektor  $(x, y, w)$  entspricht der kartesische Vektor  $(x/w, y/w)$ .

$$P_{2d}(x, y) \rightarrow P_h(wx, wy, w), \quad w \neq 0$$

$$P_h(x', y', w), \quad w \neq 0$$

$$P_{2d}(x, y) = P_{2d}\left(\frac{x'}{w}, \frac{y'}{w}\right)$$

Jeder Punkt auf der Geraden durch den Nullpunkt und durch  $(x/w, y/w, 1)$  repräsentiert im homogenen Koordinatensystem denselben kartesischen Punkt  $(x, y)$  aus dem gewöhnlichen 2D-Koordinatensystem:



In der Tat lassen sich nun die Translationen gleichwertig mit allen anderen affinen Abb. als Produkte "Matrix mal Vektor" berechnen:

$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}; \quad v' = T(dx, dy)v$$

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad v' = S(s_x, s_y)v$$

$$R(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad v' = R(\phi)v$$

*Beispiel:*

Verschiebe mittels homogener Koordinaten den Punkt (2; 3) um den Vektor (7; 5) und drehe anschließend das Ergebnis um  $90^\circ$  um den Ursprung.

$$T(7; 5): \begin{pmatrix} 1 & 0 & 7 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 9 \\ 8 \\ 1 \end{pmatrix}, \text{ entspr. } (9; 8)$$

$R(90^\circ)$ :  $\sin 90^\circ = 1$ ,  $\cos 90^\circ = 0$ , somit

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 9 \\ 8 \\ 1 \end{pmatrix} = \begin{pmatrix} -8 \\ 9 \\ 1 \end{pmatrix} = v', \text{ entspr. } (-8, 9)$$

zusammen:  $v' = R(90^\circ) \cdot T(7; 9) \cdot v$ .

Analog im dreidimensionalen Fall:

vierte, homogenisierende Koordinate,

Darstellung der affinen Transformationen durch  $4 \times 4$ -Matrizen.

Grund-Transformationen:

- Translation 
$$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Scaling 
$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation about X-axis

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation about Y-axis

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation about Z-axis

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Anwendung dieser Transformationen in Szenengraphen

3D-Szenen werden oft in gerichteten, azyklischen Graphen abgelegt, den sog. *Szenengraphen*, die durch Aufrufe einer Spezialbibliothek erzeugt werden

z.B. in OpenInventor, VRML, Java3D

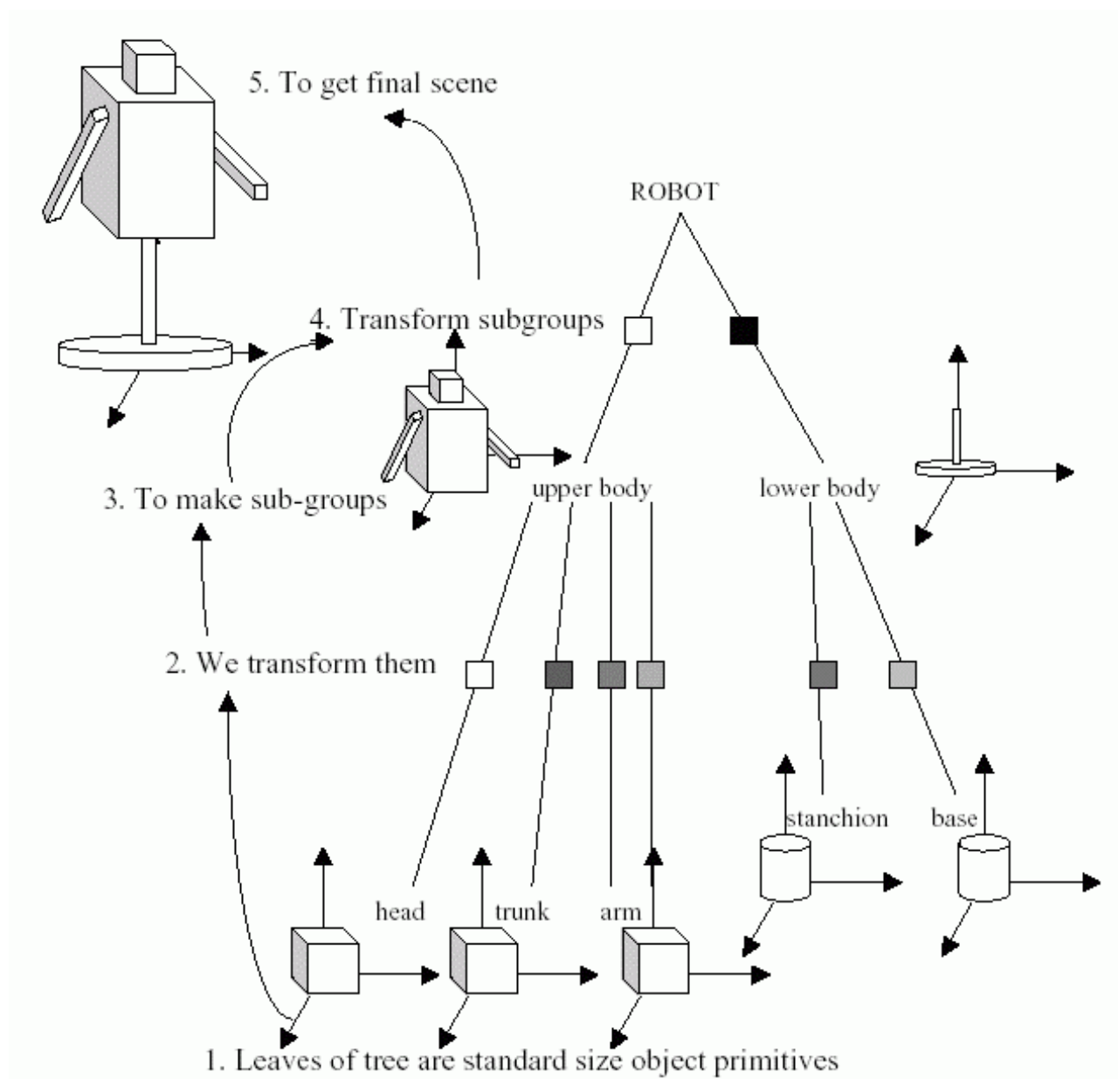
(→ spätere Übung)



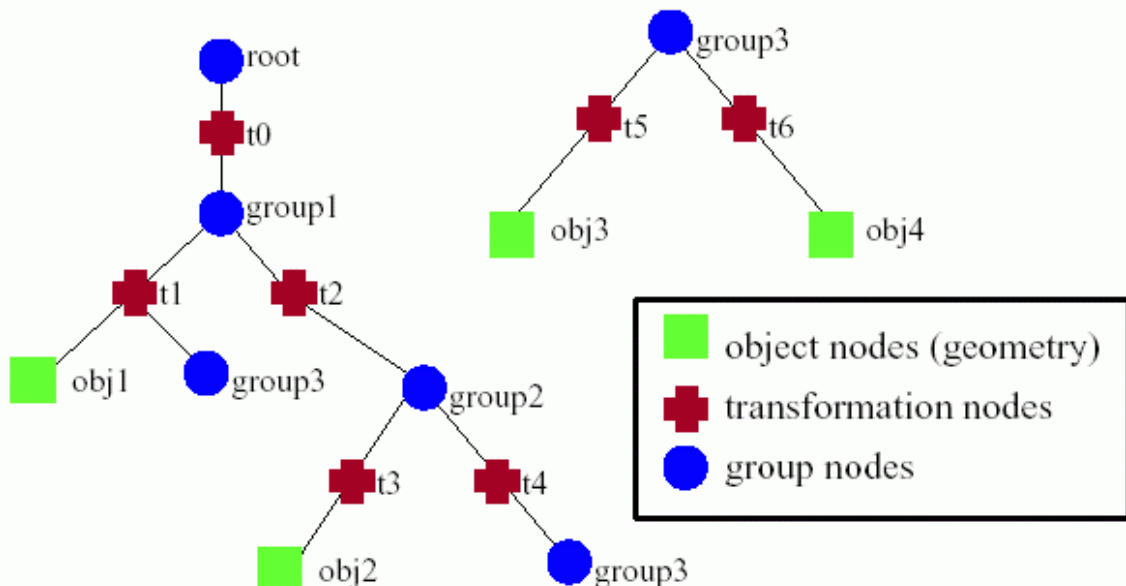
Typisches Szenengraphen-Format (es gibt zahlreiche Varianten!):

- Graphische Primitive (grundlegende Objekte: Würfel, Kugel, Kegel etc.) mit *default*-Eigenschaften (Position am Koordinatenursprung, Einheitsvolumen...) bilden Blattknoten des Graphen
- Attribute (Farbe, Textur etc.) und Transformationen sind ebenfalls Knoten des Graphen und werden auf Objektknoten je nach deren Lage im Graphen angewandt.

Beispiel (aus vanDam 2001):



im folgenden Szenengraphen beeinflusst Transformation t0 alle Objekte,  
aber t2 beeinflusst nur obj2 und eine Instanz von group3  
(welche je eine Instanz von obj3 und obj4 enthält).  
t2 wirkt nicht auf obj1 und auf die andere Instanz von group3.



Transformationsknoten enthalten normalerweise mindestens eine Matrix (für homogene Koordinatendarstellung), die die Transformation beschreibt; evtl. noch weitere Parameter.

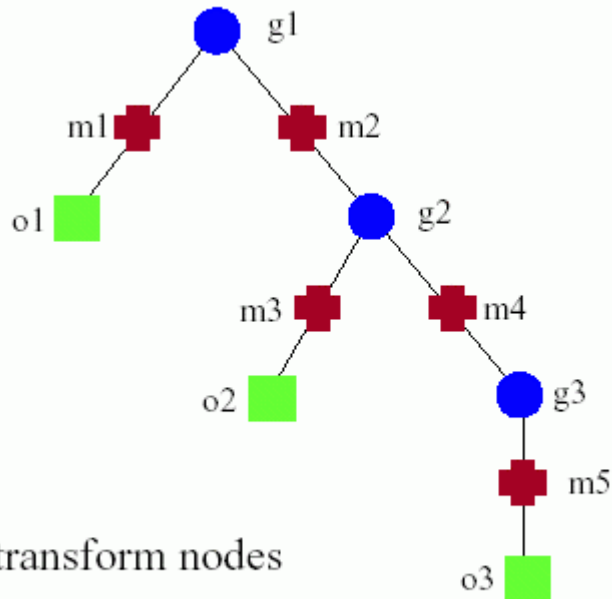
Bestimmung der Komposition der Transformationsmatrizen (CTM: *composite transformation matrix*), die auf einen Objektknoten anzuwenden ist:

Multiplikation der Matrizen in den Transformationsknoten bei top-down-Durchlauf des Baumes

(Details abhängig vom Grafikpaket).

Beispiel (aus vanDam 2001):

- An example:



**g**: group nodes

**m**: matrices of the transform nodes

**o**: object nodes

$$\text{CTM}(o1) = m1$$

$$\text{CTM}(o2) = m2 \cdot m3$$

$$\text{CTM}(o3) = m2 \cdot m4 \cdot m5$$

für einen Punkt  $v$  im Objekt  $o3$  errechnet sich seine Position im Weltkoordinatensystem (gültig an der Wurzel des Baumes) als:

$$\text{CTM} \cdot v = (m2 \cdot m4 \cdot m5) \cdot v.$$

*Transformations-Knoten in VRML 97:*

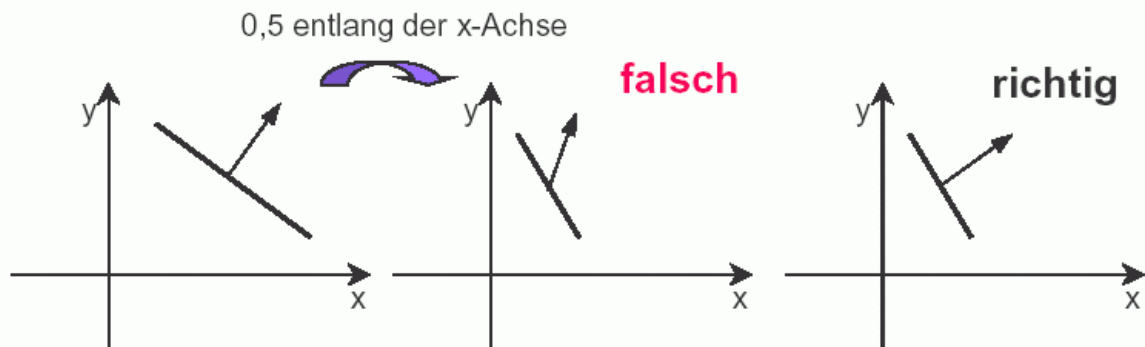
VRML 97 greift nicht auf eine allgemeine  $4 \times 4$ -Matrixdarstellung für Transformationen zurück, sondern verwendet eine Skalierung  $\mathbf{S}$ , eine Rotation  $\mathbf{R}$  und eine Translation  $\mathbf{T}(t)$  in fester Folge:

$$\mathbf{X} = \mathbf{T}(t) \mathbf{T}(\mathbf{c}) \mathbf{R} \mathbf{R}_s \mathbf{S} \mathbf{R}_s^T \mathbf{T}(-\mathbf{c})$$

1. Translation  $\mathbf{T}$  zum *center point*  $\mathbf{c}$
2. Skalierungsrotation  $\mathbf{R}_s^T$ , Skalierung  $\mathbf{S}$ , Rückrotation  $\mathbf{R}_s$
3. Rotation  $\mathbf{R}$
4. Rücktranslation vom *center point*  $\mathbf{c}$
5. Translation  $\mathbf{T}(t)$

## Transformation von Normalen

- ◆ Matrizen können Punkte, Linien, Polygone, ... transformieren
- ◆ Vorsicht bei **Normalen!**



Wird eine Matrix **M** benutzt um Geometrie zu transformieren, dann müssen zugehörige Flächennormalen mit

$$\mathbf{N} = (\mathbf{M}^{-1})^T$$

transformiert werden.

Hier ohne Beweis angegeben, siehe z.B. Turkowski's Beitrag in *Graphics Gems*, A. Glassner (Edt.)

(Krömker 2001)