Transformace (v OpenGL)

Petr Felkel

Katedra počítačové grafiky a interakce, ČVUT FEL

místnost KN:E-413 (Karlovo náměstí, budova E)

E-mail: felkel@fel.cvut.cz

Podle knihy SJ Gortlera: Foundations of Computer Graphics.
MIT Press 2012

Poslední změna: 6.3.2013

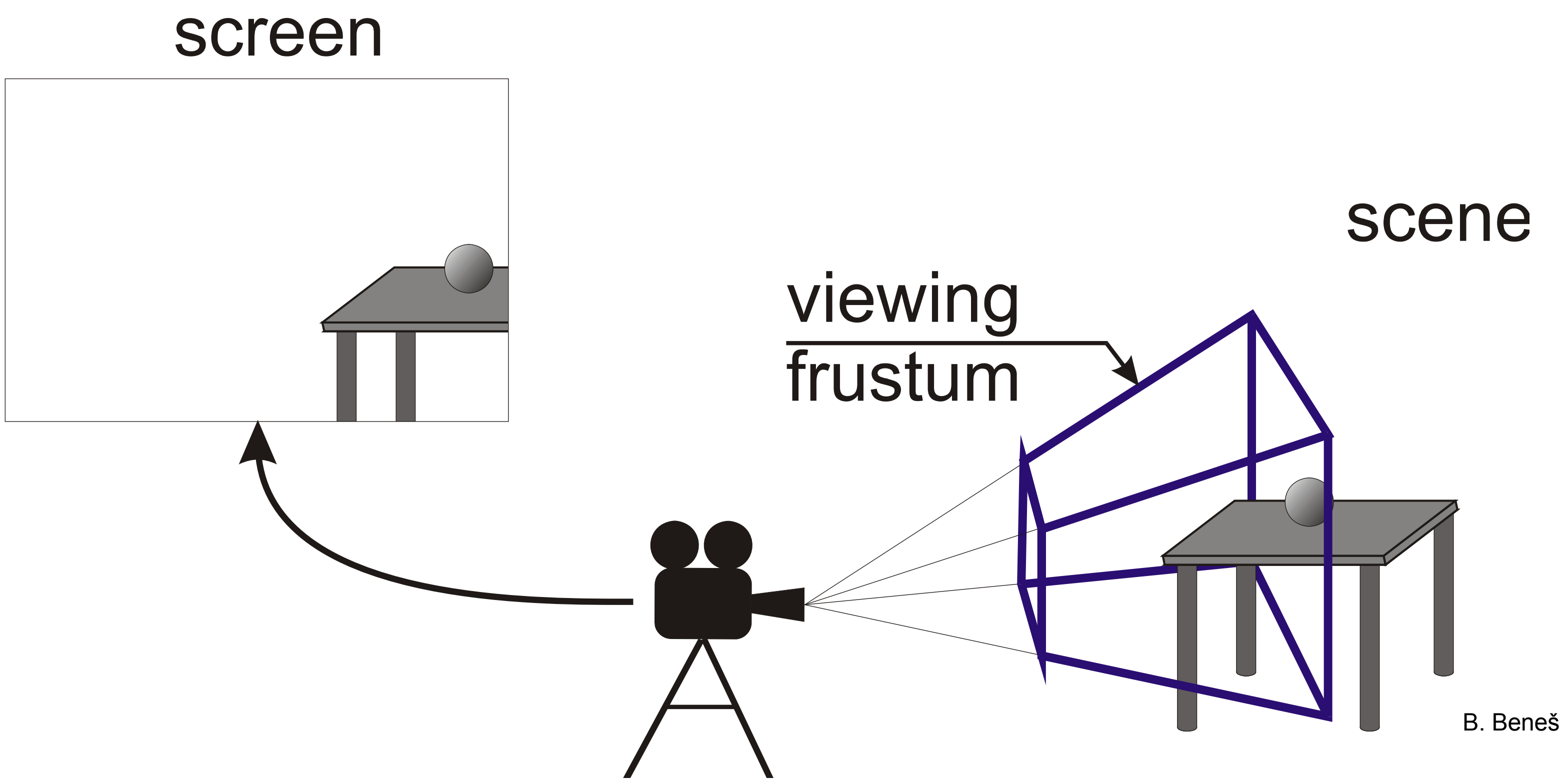
Transformace v OpenGL

- cílem grafiky je vytvářet 2D obrázky 3D scény
- musíme myslet ve 3D
- musíme umět reprezentovat body (vrcholy primitiv)
- musíme používat geometrické transformace

=> úvod do transformací v OpenGL

=> opakování lineární algebry z pohledu počítačové grafiky

Analogie s fotoaparátem



Transformace v OpenGL

Kreslení:

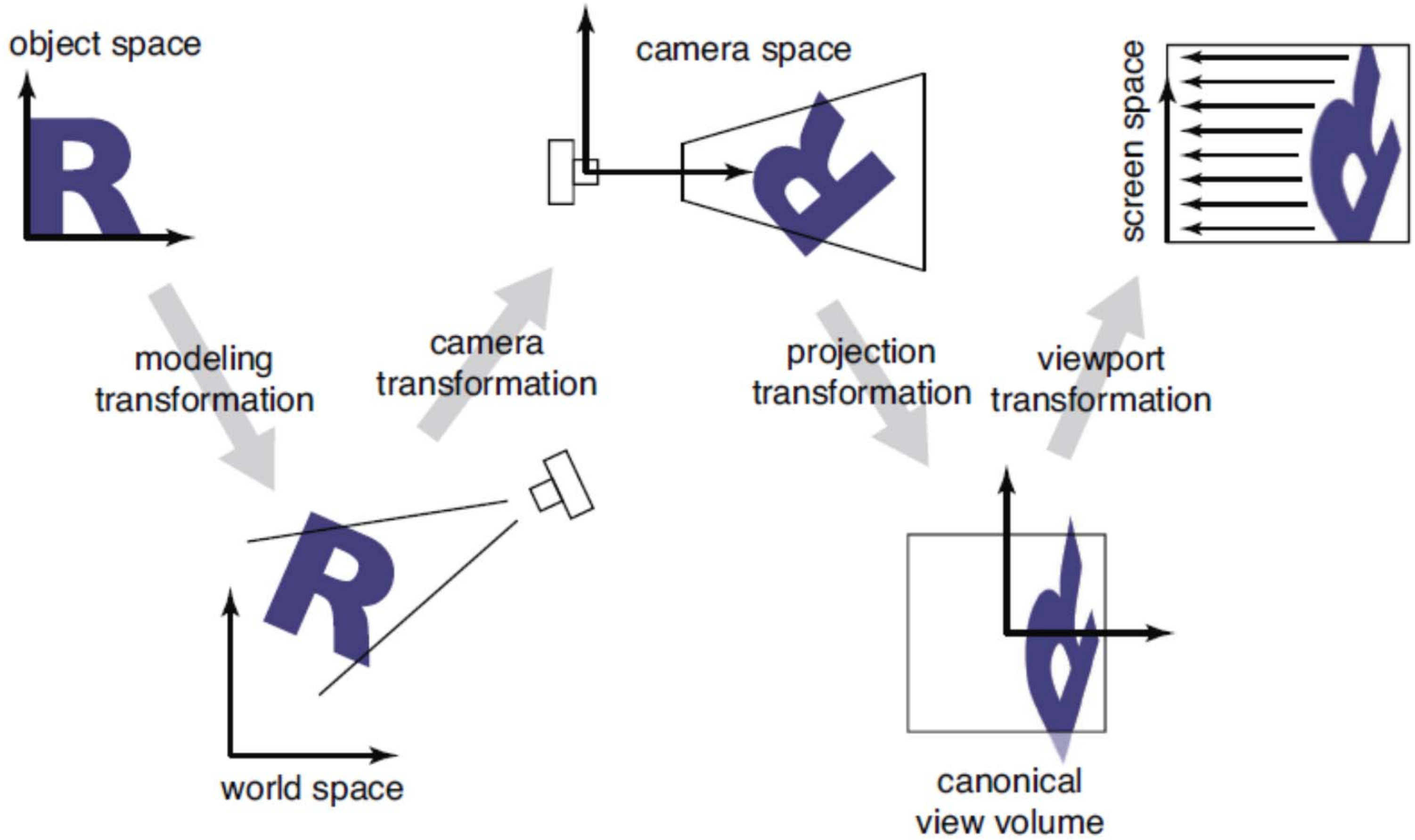
- nastavíme všechny transformace pro všechny objekty
- pak posíláme data

Posíláme jednotlivé vrcholy (OpenGL pracuje po vrcholech)

+

informaci o primitivech

Vizualizace transformací



Transformace vrcholů v OpenGL

- **modelovací transformace** (matice modelToWorld, model)
  ⇒ umístění a natočení objektů ve scéně
- **zobrazovací transformace** (matice worldToCamera, view)
  ⇒ poloha a natočení kamery

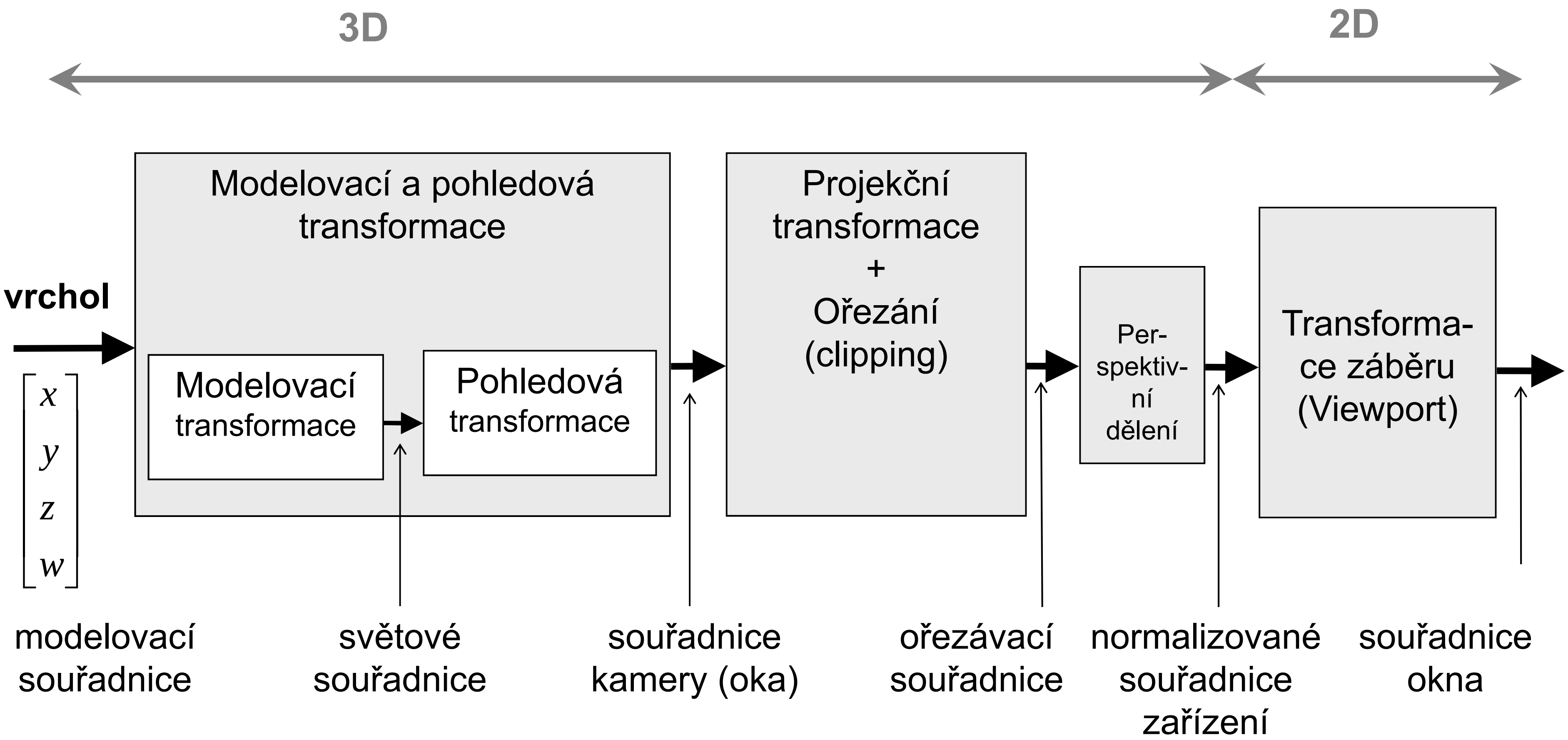
modelView

dnes

příště

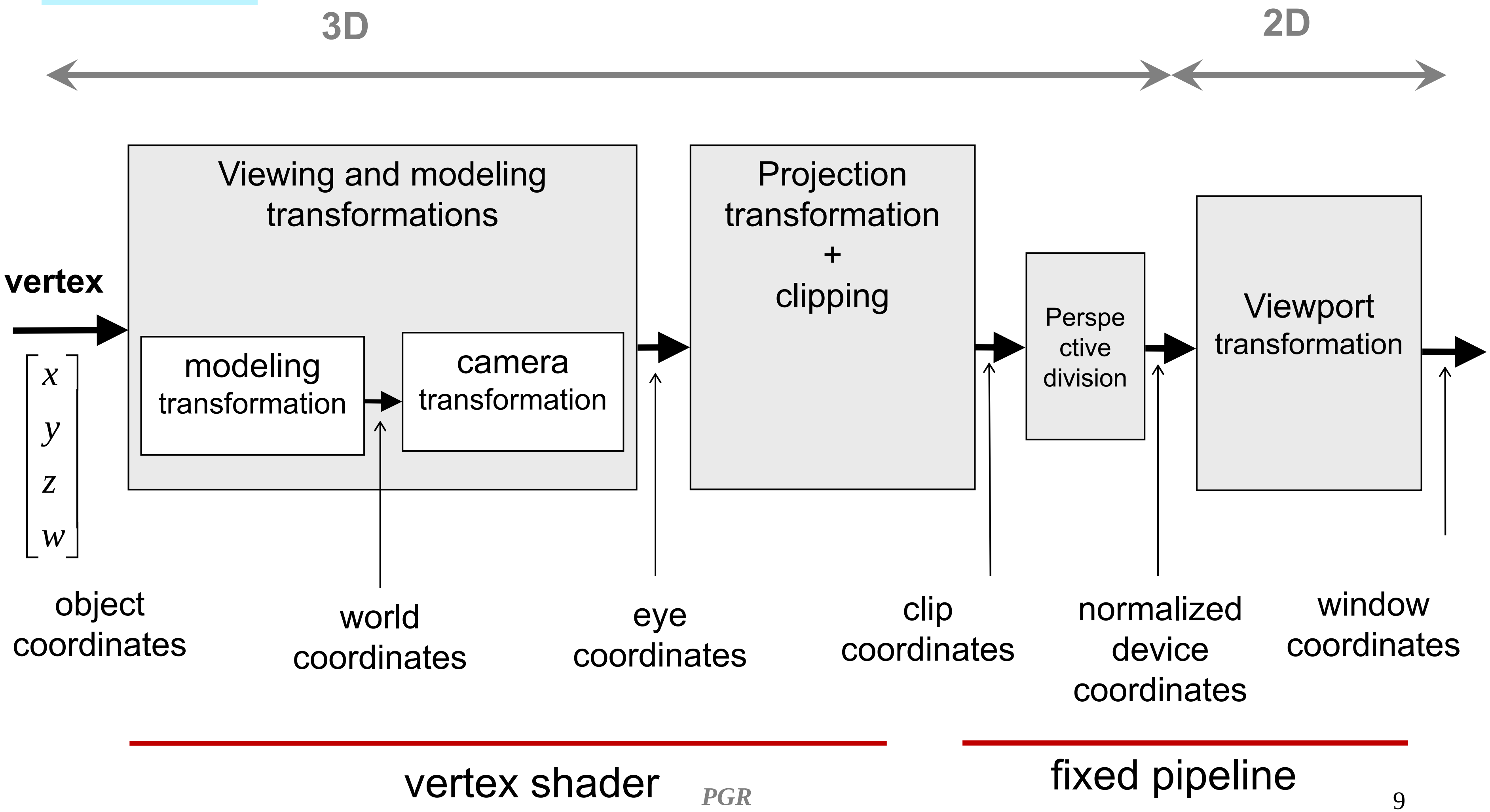
- **projekční transformace** (matice cameraToClip, projection)
  ⇒ nastavení pohledového objemu (kvádr či komolý jehlan)
  paralelní x perspektivní projekce
  ořezání
- **perspektivní dělení**
- **Transformace pracoviště (výřez, viewport)**
  ⇒ velikost a posun výřezu, který se zobrazí na obrazovce

Kroky při transformaci vrcholů

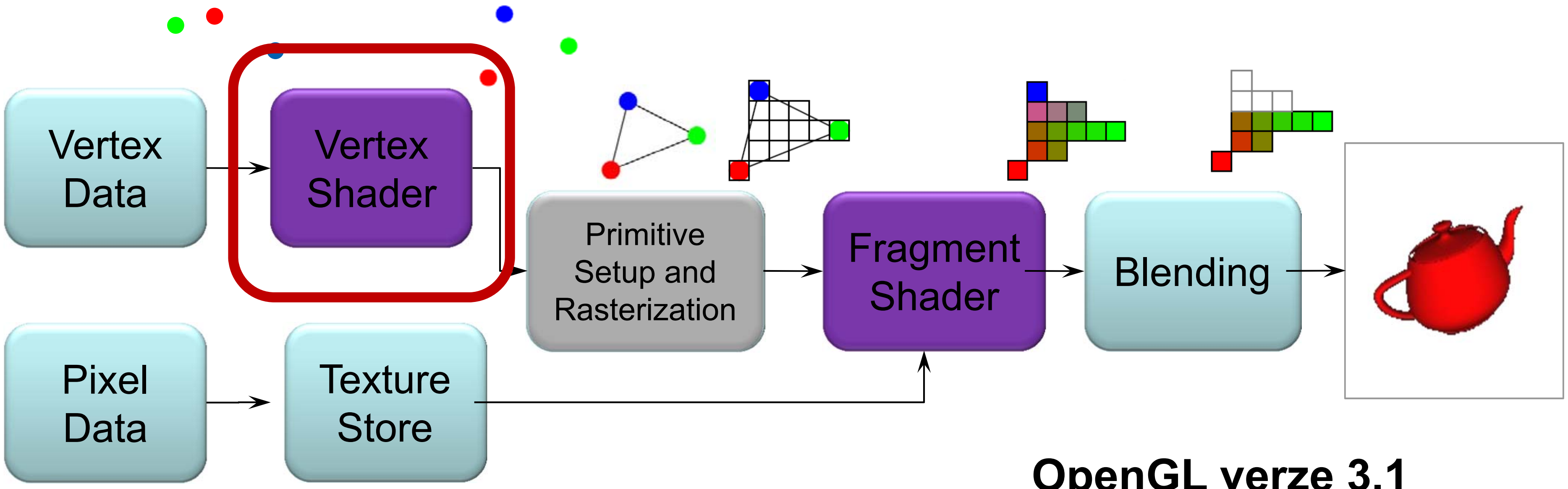


Kroky při transformaci vrcholů

In English



Transformace provádí vertex shader

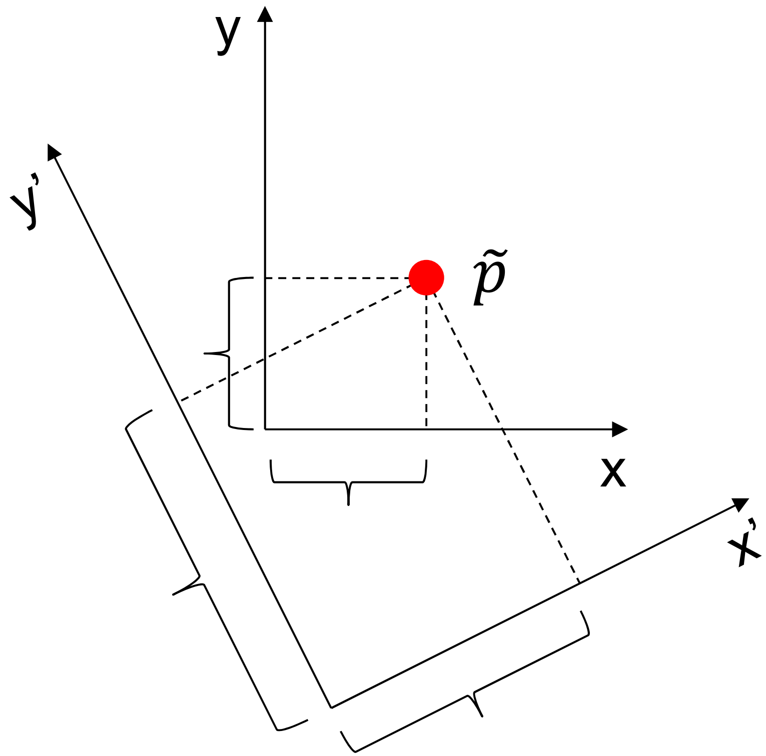


OpenGL verze 3.1

Transformace známe z lineární algebry

- V grafice používáme
  - body (vrcholy elementů) a
  - soustavy souřadnic (*Frames of reference*)
- Teď si vysvětlíme, jak s nimi pracovat
- Pořadí výkladu:
  - 1. lineární transformace
  - 2. afinní transformace

Bod a jeho souřadnice



Dvě souřadné soustavy, lišící se počátky a směry os

Geometrický bod v prostoru

- reprezentujeme obvykle jako trojici reálných čísel
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix},$$
kterým říkáme souřadnice
(vektor souřadnic, souřadný vektor, *coordinate vector*)
- Souřadnice udávají polohu v předem domluvené soustavě souřadnic

V každé soustavě souřadnic má bod jiné souřadnice – jinou n-tici reálných čísel

Definujeme čtyři základní datové typy

V každé soustavě souřadnic má bod jiné souřadnice
– jinou n-tici reálných čísel

- Proto důsledně odlišíme pojmy:
  - souřadnice,
  - soustavy souřadnic a
  - geometrické body
- Definujeme čtyři základní datové typy
  - Bod
  - Vektor
  - Souřadnice (vektor souřadnic)
  - Soustava souřadnic

Bod

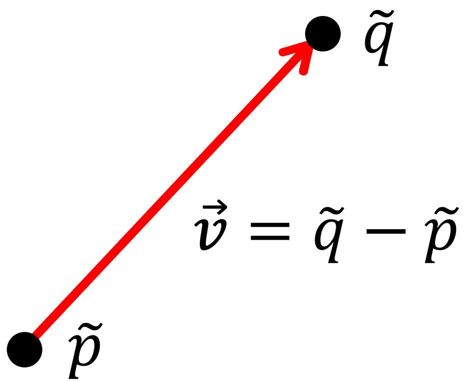
$\bullet\ \tilde{p}$

Bod $\tilde{p}$

= geometrický objekt (nečíselný)

- Reprezentuje místo (Karlovo náměstí)
- Značíme malým písmenem s vlnovkou
- Znázorňujeme kroužkem

Vektor



Vektor $\vec{v}$

= nečíselný objekt

- Reprezentuje pohyb z jednoho bodu do druhého (jdi 1 km na sever)
- Značíme $\vec{v}$,
  tj. malým písmenem se šipkou
- Znázorňujeme šipkou

Na vektor nemá žádný vliv posunutí (translace)

Souřadnice (vektor souřadnic)

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Souřadnice $\mathbf{c}$ (*coordinate vector*)

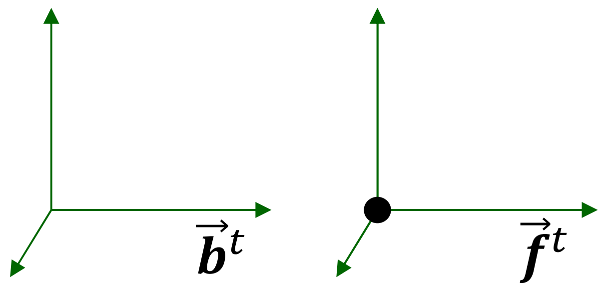
= sloupcový vektor (uspořádaná n-tice) reálných čísel, tj. čistě číselný objekt

- Popisuje bod v dané soustavě souřadnic
- Značíme ho malým tučným písmenem

$$\vec{\boldsymbol{f}}^t$$

$$\vec{\boldsymbol{b}}^t = [\ \vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_3\ ]$$

$$\vec{\boldsymbol{f}}^t = [\vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_1 \quad \tilde{o}\ ]$$



Soustava souřadnic *(coordinate system)*

= nečíselný objekt, n-tice vektorů

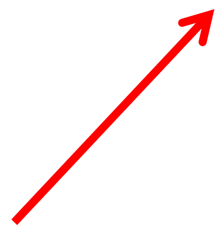
- Popisuje vztažný systém:
  - a) Báze – trojice vektorů
  - b) Frame – báze + počátek (3 vektory + bod)
- Značíme
  - tučným malým písmenem – sloupcová kolekce
  - Horní index *t* ji mění na řádkovou
  - Šipka → značí kolekci vektorů, ne čísel
- Znázorňujeme: a) bázi šipkami,
  tj. n-ticí vektorů umístěných do 1 bodu,
  b) frame bodem a bází

V dalším výkladu

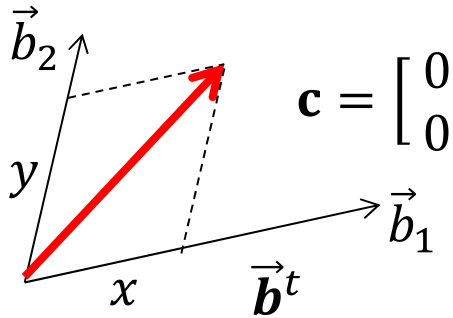
- Probereme uvedené datové typy podrobně a zjistíme, které operace s nimi lze provádět
- Zpočátku budeme hodně používat symbolické výpočty s
  - nenumerickými objekty (vektory a soustavy souřadnic) a
  - numerickými objekty (souřadnicemi)
- Teprve po zavedení všech konvencí nenumerické výpočty opustíme a přejdeme na numerické (souřadnice)

Rozdíl mezi vektorem a vektorem souřadnic

- Vektor $\vec{v}$
  - Abstraktní geometrická entita (nečíselná)
  - Reprezentuje pohyb z bodu do jiného bodu
  - „Jdi 3 km na západ“



- Souřadnice (vektor souřadnic) $\mathbf{c}$
  - Uspořádaná n-tice reálných čísel
  - $\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
  - Popisuje vektor v konkrétní zvolené soustavě souřadnic $\vec{\boldsymbol{b}}^t$
  - Souřadnice vektoru vzhledem k bázi $\vec{\boldsymbol{b}}^t$



$\mathbf{c} = \begin{bmatrix} 0.5 \\ 0.8 \end{bmatrix}$

Vektorový prostor V

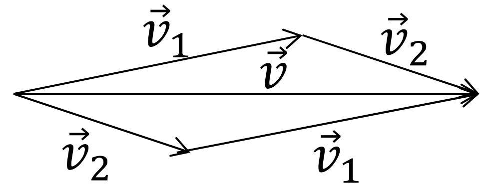
- Vektorový prostor *V*

  = Množina vektorů $\vec{v}$ (ne čísel, ale reprezentantů pohybu), pro kterou platí sada pravidel,

  - např. je definována operace sčítání

    $$\vec{v} = \vec{v}_1 + \vec{v}_2$$

    která je asociativní a komutativní

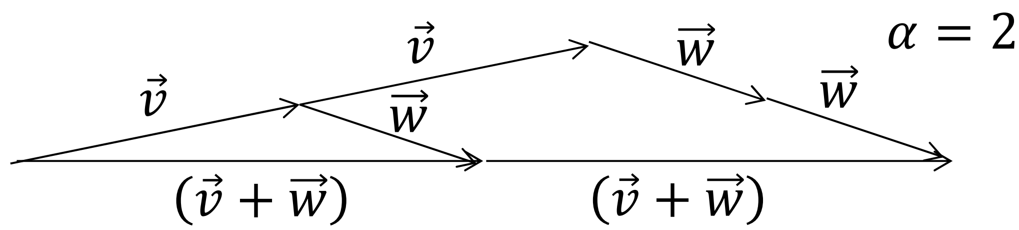
    

  - Operace násobení konstantou

    $$\vec{v} = \alpha\vec{v}_1,$$

    která je distributivní přes sčítání vektorů

    $$\alpha(\vec{v} + \vec{w}) = \alpha\vec{v} + \alpha\vec{w}$$



Báze (a lineární nezávislost)

- Báze
  - n-tice lineárně nezávislých vektorů (ve 3D trojice)
  - Jejich násobením konstantou a sčítáním lze vytvořit všechny ostatní vektory
  - Ve 3D jsou to 3 vektory
    $\vec{\boldsymbol{b}}^t = [\,\vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_3\,]$
  - Často se nazývají osy *x, y* a *z*

- Lineárně nezávislé (LN) vektory nejsou lineárně závislé
- Lineárně závislé (LZ) vektory $\vec{b}_i$, pro ně platí

$$\sum_i \alpha_i \vec{b}_i = \vec{O},\ \text{pro}\ \ \alpha_i \neq 0, i = 1, \dots, n$$

Souřadnice a vytvoření všech vektorů

- Vytvoření vektoru $\vec{v}$ o souřadnicích $\mathbf{c}_i$ a bázi $\vec{\boldsymbol{b}}^t$

$$\vec{v} = \sum_i c_i \vec{b}_i = c_1\vec{b}_1 + c_2\vec{b}_2 + c_3\vec{b}_3 + \ldots$$

- S využitím vektorové algebry lze přepsat

$$\vec{v} = \sum_i c_i \vec{b}_i = \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$\boxed{\vec{v} = \vec{\boldsymbol{b}}^t \mathbf{c},}$$

!

- kde $\vec{v}$ je vektor,
- $\vec{\boldsymbol{b}}^t$ je řádka vektorů báze a
- $\mathbf{c}$ je sloupcový vektor souřadnic

Lineární transformace ve 3D a matice 3x3

Lineární transformace $\mathcal{L}$ (V=vektorový prostor)

= transformace vektoru z V do V, pro kterou platí:

$$\mathcal{L}(\vec{u} + \vec{v}) = \mathcal{L}(\vec{u}) + \mathcal{L}(\vec{v}) \qquad (1)$$

$$\mathcal{L}(\alpha\vec{v}) = \alpha\mathcal{L}(\vec{v}), \text{ kde } \alpha \in \mathbb{R} \qquad (2)$$

- Značíme

  $\vec{v} \Rightarrow \mathcal{L}(\vec{v})$ … vektor $\vec{v}$ je transformací $\mathcal{L}$ transformován do vektoru $\mathcal{L}(\vec{v})$

- Lze popsat maticí 3x3
- To proto, že lineární transformace lze jednoznačně definovat popsáním jejího účinku na bázové vektory

Odvození matice 3x3 – transformaci vektoru přepíšeme na transformaci bází

Lineární transformaci $\mathcal{L}(\vec{v})$ vektoru $\vec{v}$ rozepíšeme

$$\vec{v} \Rightarrow \mathcal{L}(\vec{v}) = \mathcal{L}\left(\sum_i c_i \vec{b}_i\right) = \sum_i \mathcal{L}(c_i \vec{b}_i) = \sum_i c_i \mathcal{L}(\vec{b}_i)$$

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathcal{L}(\vec{b}_1) & \mathcal{L}(\vec{b}_2) & \mathcal{L}(\vec{b}_3) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Každá transformovaná báze $\mathcal{L}(\vec{b}_1)$ je také vektorem z *V*,

- lze ji tedy lze zapsat jako lineární kombinaci vektorů původní báze se správnými souřadnicemi $M_{i,j}$

$$\mathcal{L}(\vec{b}_1) = \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} M_{1,1} \\ M_{2,1} \\ M_{3,1} \end{bmatrix}$$

Transformace vektorů báze a matice 3x3

- Pro jednu novou bázi

$$\mathcal{L}(\vec{b}_1) = \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} M_{1,1} \\ M_{2,1} \\ M_{3,1} \end{bmatrix}$$

- Pro všechny tři nové báze

$$\begin{bmatrix} \mathcal{L}(\vec{b}_1) & \mathcal{L}(\vec{b}_2) & \mathcal{L}(\vec{b}_3) \end{bmatrix} = \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}$$

- Matice 3x3 reálných čísel
- Obsahuje po sloupcích souřadnice vektorů nové báze v soustavě původní báze

Lineární transformace vektoru maticově

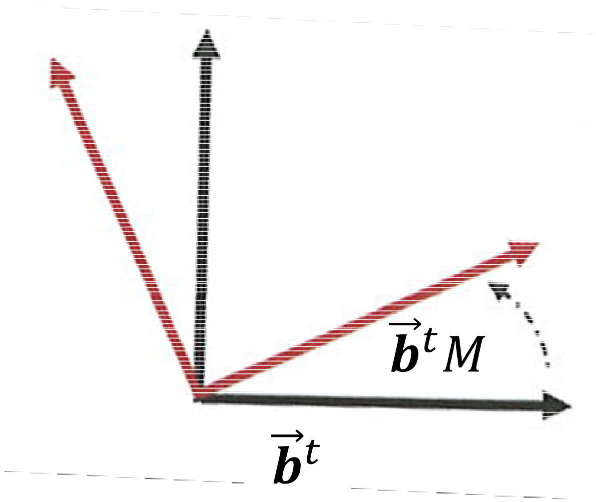
Lineární transformace vektoru pomocí matice 3x3

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$\boxed{\vec{\boldsymbol{b}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{b}}^t M \mathbf{c}}$$

Lineární transformace vektoru maticově

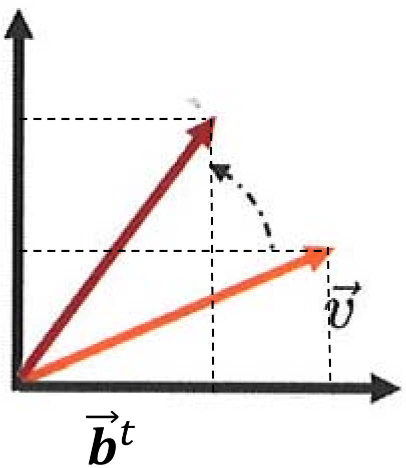
$$\vec{\boldsymbol{b}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{b}}^t M \mathbf{c}$$



Transformace báze

$$\vec{\boldsymbol{b}}^t \Rightarrow \vec{\boldsymbol{b}}^t M$$

Původní souřadnice v nové bázi



Transformace souřadnic

$$\mathbf{c} \Rightarrow M\mathbf{c}$$

Nové souřadnice ve staré bázi

[Gortler]

Identita (jednotková matice)

Identita

- Nemění nijak vektory $\vec{\boldsymbol{b}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{b}}^t I \mathbf{c} = \vec{\boldsymbol{b}}^t \mathbf{c}$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \text{ jednotková diagonální matice}$$

Inverzní matice

Inverzní matice k matici M

- Reprezentuje inverzní transformaci vektorů
- Značíme ji M^{-1}
- Platí: $MM^{-1} = M^{-1}M = I$
- Existuje, jen pokud je matice M invertibilní
  - transformace nesmí mapovat různé vektory do jednoho
  - to u lineárních transformací v grafice platí
- Počítá se
  - Algebraicky – mechanickým dosazením do vzorce
  - Přímo – pro grafické transformace známe inverzní matici

Využití matice M pro změnu bází

- Pomocí transformací lze vyjádřit rovnost (=) dvou bází

$$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{b}}^t M$$

$$\vec{\boldsymbol{a}}^t M^{-1} = \vec{\boldsymbol{b}}^t$$

Jednoduše tak můžeme vektor popsat v různých bázích

$\vec{v} = \vec{\boldsymbol{b}}^t \mathbf{c}$ souřadnice $\mathbf{c}$ v bázi $\vec{\boldsymbol{b}}^t$

$\vec{v} = \vec{\boldsymbol{a}}^t M^{-1} \mathbf{c}$ souřadnice $M^{-1}\mathbf{c}$ v bázi $\vec{\boldsymbol{a}}^t$

Vektor netransformujeme, je to stejný vektor, vyjádřený v různých bázích

Další struktura

Skalární součin vektorů $\vec{u}.\vec{v}$ (DOT product)

- vstup: 2 vektory, výstup reálné číslo
- Umožňuje definovat čtverec délky vektoru (kvadratickou normu)
  $\|\vec{v}\|^2 \coloneqq \vec{v}.\vec{v}$
- Závisí na úhlu vektorů $[0 \ldots \pi]$
  $\cos\theta \coloneqq \dfrac{\vec{u}.\vec{v}}{\|\vec{u}\|\|\vec{v}\|}$
- Ortogonální (kolmé) vektory: $\vec{u}.\vec{v} = 0$
- Ortonormání báze – všechny vektory báze jsou jednotkové a vzájemně ortogonální

Skalární součin vektorů v ortonormální bázi

- Skalární součin vektorů v souřadnicích

$$\vec{\boldsymbol{b}}^t \mathbf{c} \,.\, \vec{\boldsymbol{b}}^t \boldsymbol{d} = \left(\sum_i c_i \vec{b}_i\right) . \left(\sum_j d_j \vec{b}_j\right)$$

$$= \sum_i \sum_j c_i d_j \left(\vec{b}_i \,.\, \vec{b}_j\right)$$

$$= \sum_i c_i d_i$$

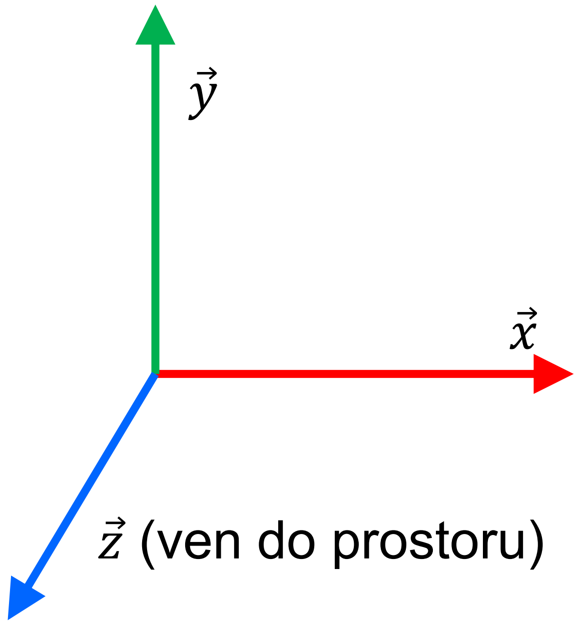
v ortonormální bázi

$$\left(\vec{b}_i \,.\, \vec{b}_j\right) \quad = 0 \quad \text{pro i} \neq j$$

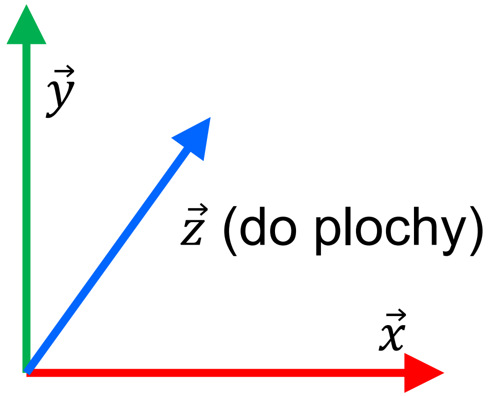
$$= 1 \quad \text{pro i} = j$$

Orientace ortonormální báze

Pravotočivá (OpenGL)



Levotočivá



Vektorový součin (*cross product*)

Vektorový součin (*cross product*)

$\vec{u} \times \vec{v} = \|\vec{u}\|\|\vec{v}\| \sin\theta \,.\, \vec{n},$ kde $\vec{n}$ je kolmé k rovině $\rho(\vec{u}, \vec{v})$

a báze $[\,\vec{u} \quad \vec{v} \quad \vec{n}\,]$ je pravotočivá

V pravotočivé ortonormální bázi $\vec{\boldsymbol{b}}^t$ se spočítá

$$\left(\vec{\boldsymbol{b}}^t \mathbf{c}\right) \times \left(\vec{\boldsymbol{b}}^t \boldsymbol{d}\right) = \vec{\boldsymbol{b}}^t \begin{bmatrix} c_2 d_3 - c_3 d_2 \\ c_3 d_1 - c_1 d_3 \\ c_1 d_2 - c_2 d_1 \end{bmatrix}$$

Rotace

- Nejtypičtější lineární transformace
- Ve 2D

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

- Maticově

$$\begin{bmatrix}\vec{b}_1 & \vec{b}_2\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} \Rightarrow \begin{bmatrix}\vec{b}_1 & \vec{b}_2\end{bmatrix}\begin{bmatrix}cos\,\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$$

- Z toho rotace báze je

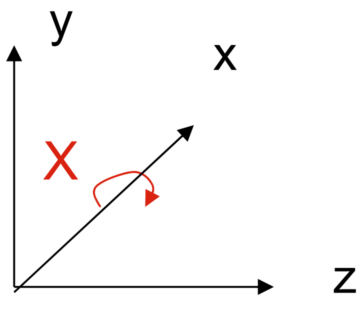
$$\begin{bmatrix}\vec{b}_1 & \vec{b}_2\end{bmatrix} \Rightarrow \begin{bmatrix}\vec{b}_1 & \vec{b}_2\end{bmatrix}\begin{bmatrix}cos\,\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$

Rotace ve 3D

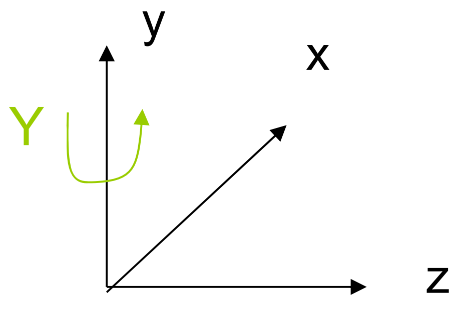
- Ve 3D rotace kolem osy z o úhel θ, $c = \cos\theta, s = \sin\theta$

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

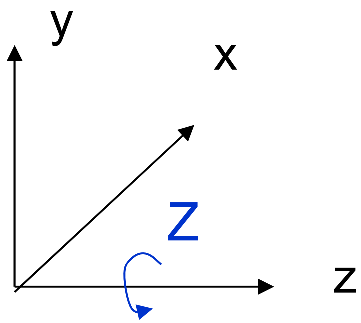
- Matice rotací podle souřadných os



$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}$$



$$\begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix}$$



$$\begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Souřadnice na ose otáčení se nemění

Rotace ve 3D

- Na pořadí rotací záleží (nejsou komutativní)!
- Navíc možnost splynutí dvou os - gimbal lock
  probereme později – viz trackball a kvaterniony
- Obecná rotace kolem jednotkového vektoru $\vec{k} = \left[\mathrm{k}_x, k_y, k_z\right]^t$
  o úhel θ, $c = \cos\theta, s = \sin\theta\,, v = 1 - c$

$$R = \begin{bmatrix} k_x^2 v + c & k_x k_y v - k_z s & k_x k_z v + k_y s \\ k_y k_x v + k_z s & k_y^2 v + c & k_y k_z v - k_x s \\ k_z k_x v - k_y s & k_z k_y v + k_x s & k_z^2 v + c \end{bmatrix}$$

Změna měřítka (scale)

- Symetrická – ve všech směrech stejná

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Asymetrická – různá pro každou osu

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Rekapitulace pojmů vektor a bod

Vektor $\vec{v}$ = nečíselný objekt
reprezentuje pohyb z jednoho bodu do druhého
(jdi 1 km na sever)

- Operace sčítání, násobení konstantou, nulový vektor
- Pro vektory nemá smysl posunutí

Bod $\tilde{p}$ = geometrický objekt (nečíselný)
pevně umístěný v geometrickém světě

- Lze lineárně transformovat jako vektory
- Má pro něj smysl posunutí (to pro vektory nemá smysl)
- Body nelze sčítat (Václavské + Karlovo náměstí??),
  nelze násobit konstantou (6x Újezd?), neexistuje nulový bod)

Afinní prostor

= množina bodů + přidružený n-rozměrný vektorový prostor,

plus existují operace, které je propojují

$\tilde{p} - \tilde{q} = \vec{v}$ odečtení bodů – dá vektor pohybu z $\tilde{q}$ do $\tilde{p}$

$\tilde{q} + \vec{v} = \tilde{p}$ přičtení vektoru k bodu – nová pozice do které jsme se přemístili po vektoru $\vec{v}$

Body lze

- lineárně transformovat (rotace)
- i posunovat (translace)
  –> zavedeme afinní transformace

K uložení translace (+ později promítání) slouží matice 4x4

Afinní soustava souřadnic

- Afinní soustava souřadnic (*frame*)
  $\vec{\boldsymbol{f}}^t = [\vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_1 \quad \tilde{o}\,]$
  je tvořena 3 vektory báze (3 vektory os) + počátkem $\tilde{o}$
- Libovolný bod popsán jako posunutí z počátku

$$\tilde{p} = \tilde{o} + \sum_i c_i \vec{b}_i = [\vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_3 \quad \tilde{o}\,] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} = \vec{\boldsymbol{f}}^t \mathbf{c}$$

- souřadnice c_4 pro body je vždy nastavena na 1
  (aby se nehýbalo počátkem), definováno $1\tilde{o} = \tilde{o}$
- Pro vektory ji nastavujeme 0, definováno $0\tilde{o} = \vec{0}$

Afinní transformace a matice 4x4

Afinní matice $\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$, 4. řádka $[0 \quad 0 \quad 0 \quad 1]$!!!

Afinní transformace bodu

$$[\vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_3 \quad \tilde{o}\,] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} \Rightarrow [\vec{b}_1 \quad \vec{b}_2 \quad \vec{b}_3 \quad \tilde{o}\,] \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix}$$

$$\boxed{\vec{\boldsymbol{f}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{f}}^t A \mathbf{c}}$$

definováno $0\tilde{o} = \vec{0}$

$1\tilde{o} = \tilde{o}$

Afinní transformace soustavy souřadnic

Afinní transformace soustavy souřadnic *(frame)*

$$\begin{bmatrix}\vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o}\end{bmatrix} \Rightarrow \begin{bmatrix}\vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o}\end{bmatrix}\begin{bmatrix}a & b & c & d\\ e & f & g & h\\ i & j & k & l\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\boxed{\vec{\boldsymbol{f}}^t \Rightarrow \vec{\boldsymbol{f}}^t A}$$

definováno $0\tilde{o} = \vec{0}$

$1\tilde{o} = \tilde{o}$

Lineární transformace bodů

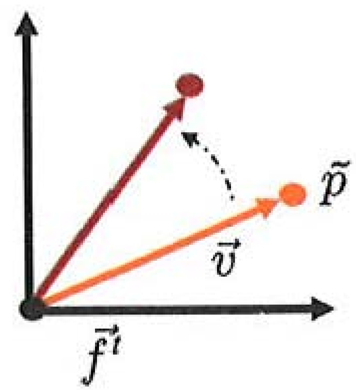
- Lineární transformace se vloží do levého horního rohu

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} a & b & c & 0 \\ e & f & g & 0 \\ i & j & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix}$$

- Lineární matici 4x4 lze zkráceně zapsat takto

$$L = \begin{bmatrix} l & 0 \\ 0 & 1 \end{bmatrix},$$

kde l je matice 3x3,
0 vpravo nahoře je matice nul 3x1
0 vlevo dole je matice nul 1x3
1 vpravo dole je skalár



[Gortler]

Pozn.: lineární transformace bodu odpovídá lineární transformaci jeho rádiusvektoru (vektoru ofsetu umístěnému do počátku $\tilde{o}$)

Posunutí (translace)

- Posunutí se ukládá do 4. sloupce

$$\begin{bmatrix}\vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o}\end{bmatrix}\begin{bmatrix}c_1\\c_2\\c_3\\1\end{bmatrix} \Rightarrow \begin{bmatrix}\vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o}\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & t_x\\0 & 1 & 0 & t_y\\0 & 0 & 1 & t_z\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}c_1\\c_2\\c_3\\1\end{bmatrix}$$

- Translační matici 4x4 lze zkráceně zapsat takto

$$T = \begin{bmatrix} i & t \\ 0 & 1 \end{bmatrix},$$

kde
- i je jednotková matice 3x3,
- 0 vpravo nahoře je posun v matici 3x1
- 0 vlevo dole je matice nul 1x3
- 1 vpravo dole je skalár

Pozn.: Kdybychom zadali 4. souřadnici **c** rovnu nule, transformovali bychom vektor a translace by se neuplatnila (což je správně)

Rozložení matice na translační a lineární

- Afinní matice se dá rozložit na translační a lineární část

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & b & c & 0 \\ e & f & g & 0 \\ i & j & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} l & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} i & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boxed{A = TL}$$

POZOR
– násobení matic není komutativní
– záleží na pořadí *TL*
(Lze i rozklad na $A = LT'$ - neuvádíme)

Pozn: Pokud by byla lineární částí matice rotace, psali bychom

$$A = TR$$

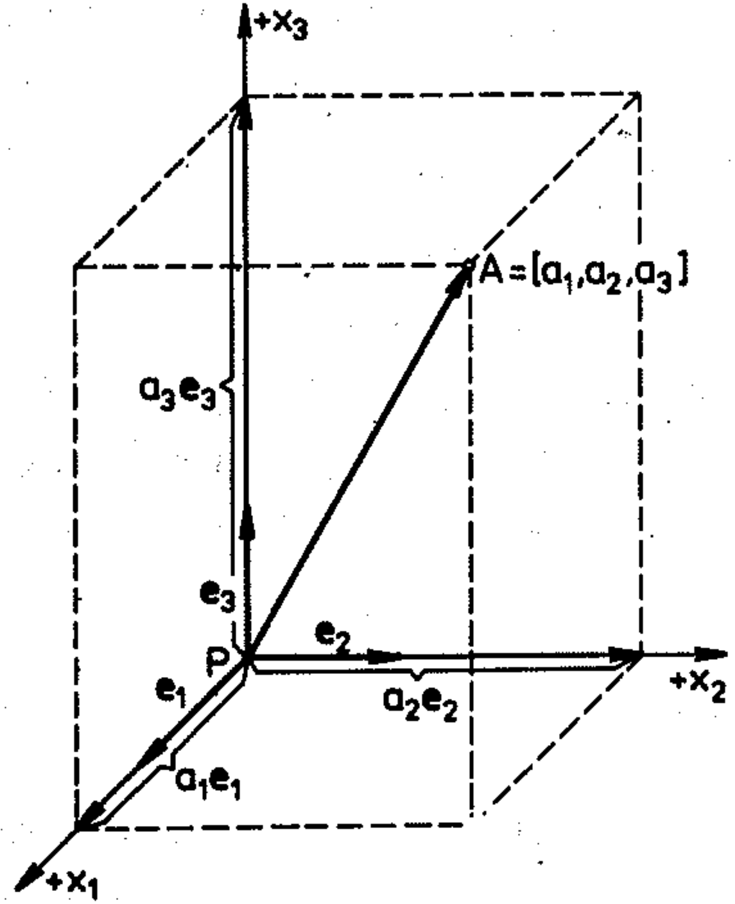
Rotace s translací jsou tzv. *Rigid body transformations (RBT)*, zachovávají skalární součiny vektorů, pravotočivost báze a vzdálenosti mezi body

Transformace normál

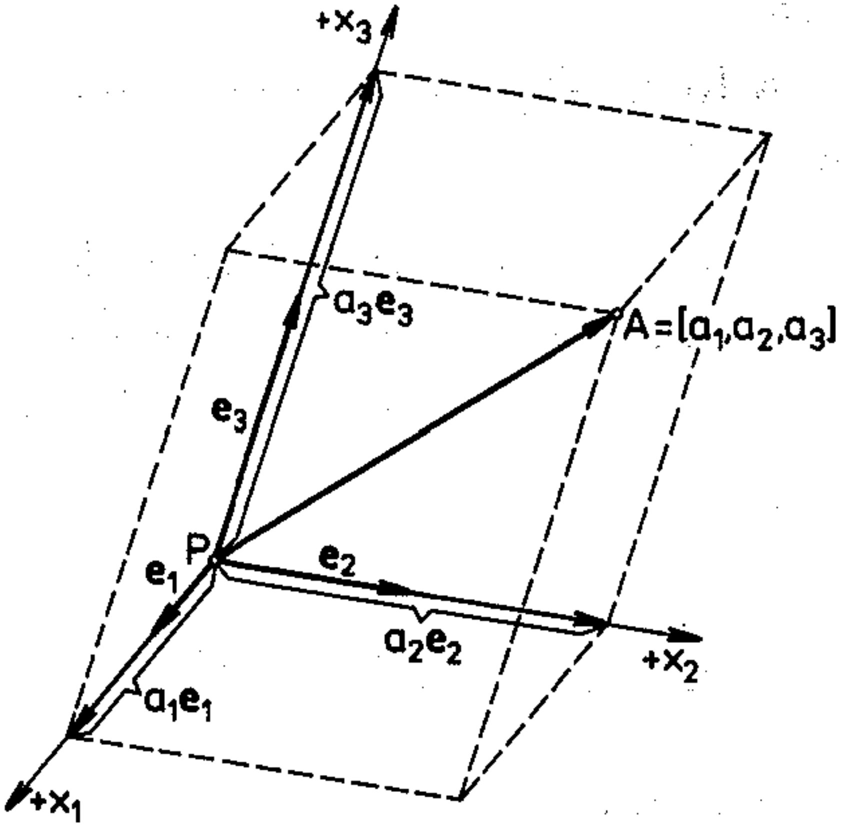
- Normály nutno transformovat inverzní transponovanou lineární části matice
- $\begin{bmatrix} nx' \\ ny' \\ nz' \end{bmatrix} = l^{-t} \begin{bmatrix} nx \\ ny \\ nz \end{bmatrix}$
- Pro rotaci je $l^{-t} = l$, protože je l ortonormální maticí (rotace je RBT transformace)
- Pro nesymetrickou změnu měřítka je nutno použít l^{-t}

Kartézská a afinní soustava souřadnic

[Budínský]



Bod A v kartézské soustavě souřadnic $[\vec{e}_1 \quad \vec{e}_2 \quad \vec{e}_3 \quad \tilde{p}\,]$



Bod A v afinní soustavě souřadnic $[\vec{e}_1 \quad \vec{e}_2 \quad \vec{e}_3 \quad \tilde{p}\,]$

Vztažná soustava souřadnic

- V grafice se používá současně celá řada soustav souřadnic (modelová, pohledová, kamery,…)
- Vždy musíme provést transformaci v té správné
- Příklad: dán bod $\tilde{p}$ a transformace $S = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- S 2x zvětší vzdálenosti od počátku ve směru osy x
- Dokud nezvolíme soustavu souřadnic, není jasné, jak transformaci provést
  - nevíme, která osa je osa *x*
  - neznáme počátek soustavy

Zvětšení ve dvou různých soustavách

$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{f}}^t A$

$\vec{\boldsymbol{a}}^t A^{-1} = \vec{\boldsymbol{f}}^t$

- Dán bod $\tilde{p} = \vec{\boldsymbol{f}}^t \mathbf{c}$ v soustavě souřadnic $\vec{\boldsymbol{f}}^t$ a druhá soustava souřadnic $\vec{\boldsymbol{a}}^t$, kde $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{f}}^t A$
- V $\vec{\boldsymbol{a}}^t$ má bod $\tilde{p} = \vec{\boldsymbol{a}}^t \mathbf{d}$ souřadnice $\mathbf{d} = A^{-1}\mathbf{c}$

Transformace S

Čteme: Bod $\tilde{p}$ je transformován S vzhledem k soustavě souřadnic $\vec{\boldsymbol{f}}^t$

- V soustavě $\vec{\boldsymbol{f}}^t$: $\vec{\boldsymbol{f}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{f}}^t S \mathbf{c}$
- V soustavě $\vec{\boldsymbol{a}}^t$: $\vec{\boldsymbol{f}}^t \mathbf{c} = \vec{\boldsymbol{a}}^t \mathbf{d} \Rightarrow \vec{\boldsymbol{a}}^t S \mathbf{d} = \vec{\boldsymbol{f}}^t A S A^{-1} \mathbf{c}$

- Bod se posune pokaždé úplně jinam!!!

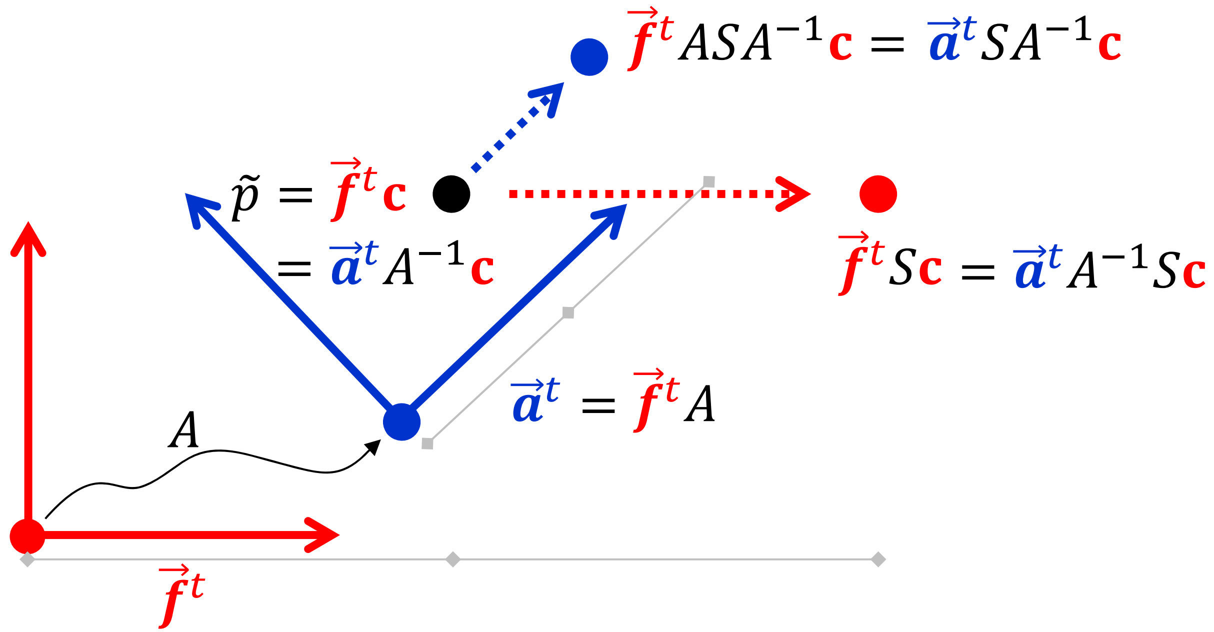
Pravidlo levé ruky: transformuje se vůči soustavě souřadnic nejblíže vlevo od transformační matice

Zvětšení ve dvou různých soustavách

$$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{f}}^t A$$
$$\vec{\boldsymbol{a}}^t A^{-1} = \vec{\boldsymbol{f}}^t$$

Změna měřítka S

- V soustavě $\vec{\boldsymbol{f}}^t$: $\quad \vec{\boldsymbol{f}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{f}}^t S \mathbf{c}$
- V soustavě $\vec{\boldsymbol{a}}^t$: $\quad \vec{\boldsymbol{f}}^t \mathbf{c} = \vec{\boldsymbol{a}}^t \mathbf{d} \Rightarrow \vec{\boldsymbol{a}}^t S \mathbf{d} = \vec{\boldsymbol{f}}^t A S A^{-1} \mathbf{c}, \quad \mathbf{d} = A^{-1}\mathbf{c}$



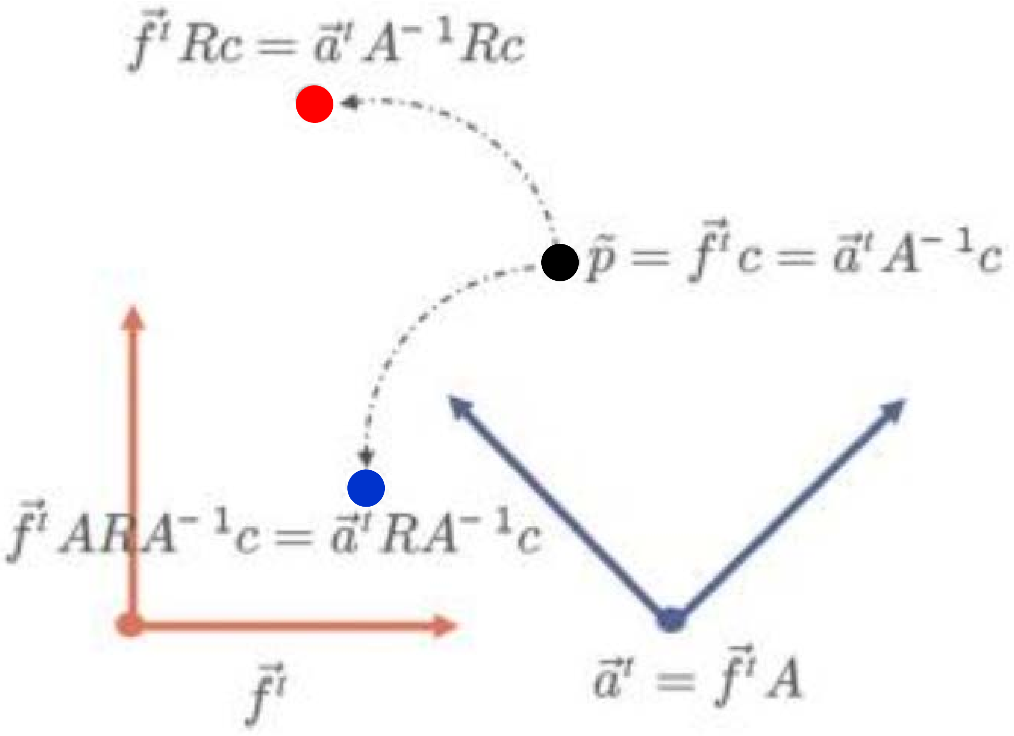
[Gortler]

Rotace ve dvou různých soustavách

$$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{f}}^t A$$
$$\vec{\boldsymbol{a}}^t A^{-1} = \vec{\boldsymbol{f}}^t$$

Rotace R

- V soustavě $\vec{\boldsymbol{f}}^t$: $\quad \vec{\boldsymbol{f}}^t\mathbf{c} \Rightarrow \vec{\boldsymbol{f}}^t R\mathbf{c}$
- V soustavě $\vec{\boldsymbol{a}}^t$: $\quad \vec{\boldsymbol{f}}^t\mathbf{c} = \vec{\boldsymbol{a}}^t\mathbf{d} \Rightarrow \vec{\boldsymbol{a}}^t R\mathbf{d} = \vec{\boldsymbol{f}}^t ARA^{-1}\mathbf{c}$



[Gortler]

Pravidlo levé ruky

- Transformuje se vůči soustavě souřadnic nejblíže vlevo od transformační matice
- Čteme tedy:
  - $\vec{\boldsymbol{f}}^t S \mathbf{c}$ Bod $\tilde{p}$ je transformován S vzhledem k soustavě souřadnic $\vec{\boldsymbol{f}}^t$
  - $\vec{\boldsymbol{a}}^t S A^{-1} \mathbf{c}$ Bod $\tilde{p}$ je transformován S vzhledem k soustavě souřadnic $\vec{\boldsymbol{a}}^t$
  - $\vec{\boldsymbol{f}}^t \Rightarrow \vec{\boldsymbol{f}}^t S$ Soustava souřadnic $\vec{\boldsymbol{f}}^t$ je transformována S vzhledem k $\vec{\boldsymbol{f}}^t$
  - $\vec{\boldsymbol{f}}^t = \vec{\boldsymbol{a}}^t A^{-1} \Rightarrow \vec{\boldsymbol{a}}^t S A^{-1}$ $\vec{\boldsymbol{f}}^t$ je transformována S vzhledem k $\vec{\boldsymbol{a}}^t$

Transformace vzhledem k pomocné soustavě souřadnic

$$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{f}}^t A$$
$$\vec{\boldsymbol{a}}^t A^{-1} = \vec{\boldsymbol{f}}^t$$

Transformace M vůči pomocné soustavě

Př. Otáčení zeměkoule kolem slunce

Soustava souřadnic zeměkoule $\vec{\boldsymbol{f}}^t$ pozice země $\vec{\boldsymbol{f}}^t \mathbf{c}$

Soustava souřadnic slunce $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{f}}^t A$

Transformace vzhledem k soustavě $\vec{\boldsymbol{a}}^t$:

$\vec{\boldsymbol{f}}^t$

$= \vec{\boldsymbol{a}}^t A^{-1}$ $\vec{\boldsymbol{f}}^t$ vyjádřeno pomocí $\vec{\boldsymbol{a}}^t$

$\Rightarrow \vec{\boldsymbol{a}}^t M A^{-1}$ transformace v soustavě souřadnic $\vec{\boldsymbol{a}}^t$ (podle pravidla levé ruky vložíme M za $\vec{\boldsymbol{a}}^t$)

$= \vec{\boldsymbol{f}}^t A M A^{-1}$ návrat a pomocné soustavy souřadnic

Dva způsoby vyhodnocení složené transformace

- Je dána složená transformace $\vec{\boldsymbol{f}}^t \Rightarrow \vec{\boldsymbol{f}}^t TR$, T posun o +1 v x, R okolo z
- Lze vyhodnocovat zleva doprava nebo zprava doleva
- Vztažná soustava je vždy ta nalevo od matice

1) Vyhodnocení *zleva doprava* – lokálně v nové soustavě

1a) $\vec{\boldsymbol{f}}^t \Rightarrow \vec{\boldsymbol{f}}^t T = \vec{\boldsymbol{f}}'^t$ $\quad$ $\vec{\boldsymbol{f}}^t$ se posune o 1 vůči $\vec{\boldsymbol{f}}^t$ $\quad$ T vůči $\vec{\boldsymbol{f}}^t$

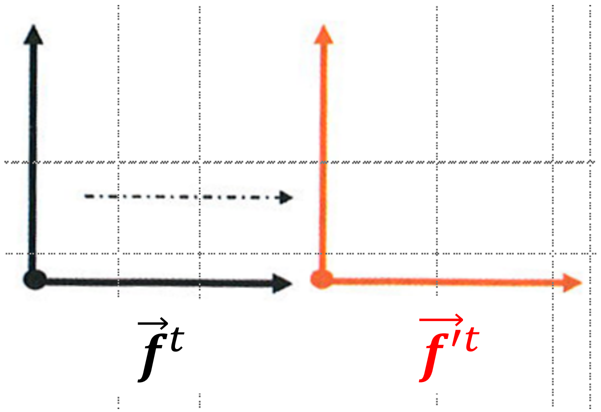
1b) $\vec{\boldsymbol{f}}'^t \Rightarrow \vec{\boldsymbol{f}}'^t R$ $\quad$ $\vec{\boldsymbol{f}}'^t$ se otočí vzhledem k $\vec{\boldsymbol{f}}'^t$ $\quad$ R vůči $\vec{\boldsymbol{f}}'^t$

2) Vyhodnocení *zprava doleva* – globálně v původní soustavě

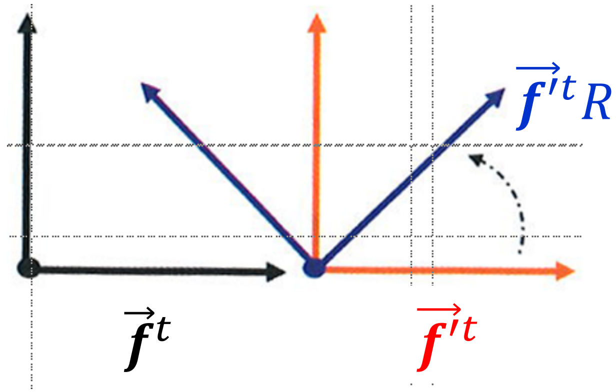
2a) $\vec{\boldsymbol{f}}^t \Rightarrow \vec{\boldsymbol{f}}^t R = \vec{\boldsymbol{f}}^{\circ t}$ $\quad$ $\vec{\boldsymbol{f}}^t$ se otočí vzhledem k $\vec{\boldsymbol{f}}^t$ $\quad$ R vůči $\vec{\boldsymbol{f}}^t$

2b) $\vec{\boldsymbol{f}}^{\circ t} = \vec{\boldsymbol{f}}^t R \Rightarrow \vec{\boldsymbol{f}}^t TR$ $\quad$ $\vec{\boldsymbol{f}}^{\circ t}$ se posune vzhledem k $\vec{\boldsymbol{f}}^t$ $\quad$ T vůči $\vec{\boldsymbol{f}}^t$

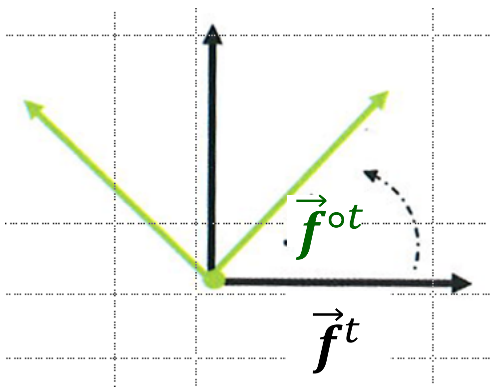
Dva způsoby interpretace složené transformace



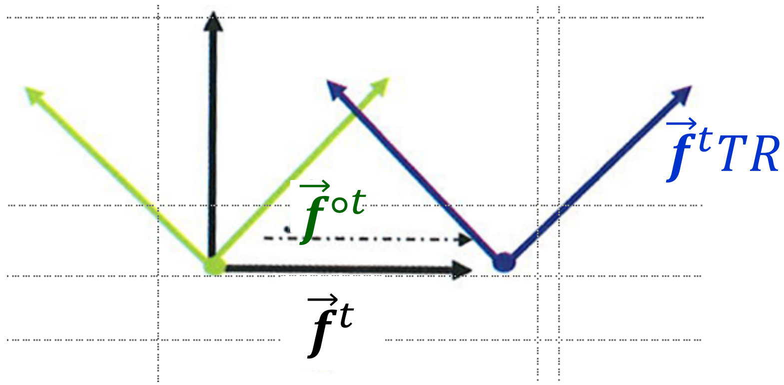
1a) Lokální translace



1b) Lokální rotace



2a) Globální rotace



2b) Globální translace

Dva způsoby interpretace složené transformace - shrnutí

Složená transformace $\vec{\boldsymbol{f}}^t \Rightarrow \vec{\boldsymbol{f}}^t TR$, T posun o +1 v x, R okolo z

se dá číst dvěma způsoby:

1) *zleva doprava* – každá další transformace se provede vůči nově vzniklé „lokální“ soustavě

- Napřed posun T vzhledem k $\vec{\boldsymbol{f}}^t$
- Pak rotace R vzhledem k mezivýsledku $\vec{\boldsymbol{f}}^t T$ po translaci

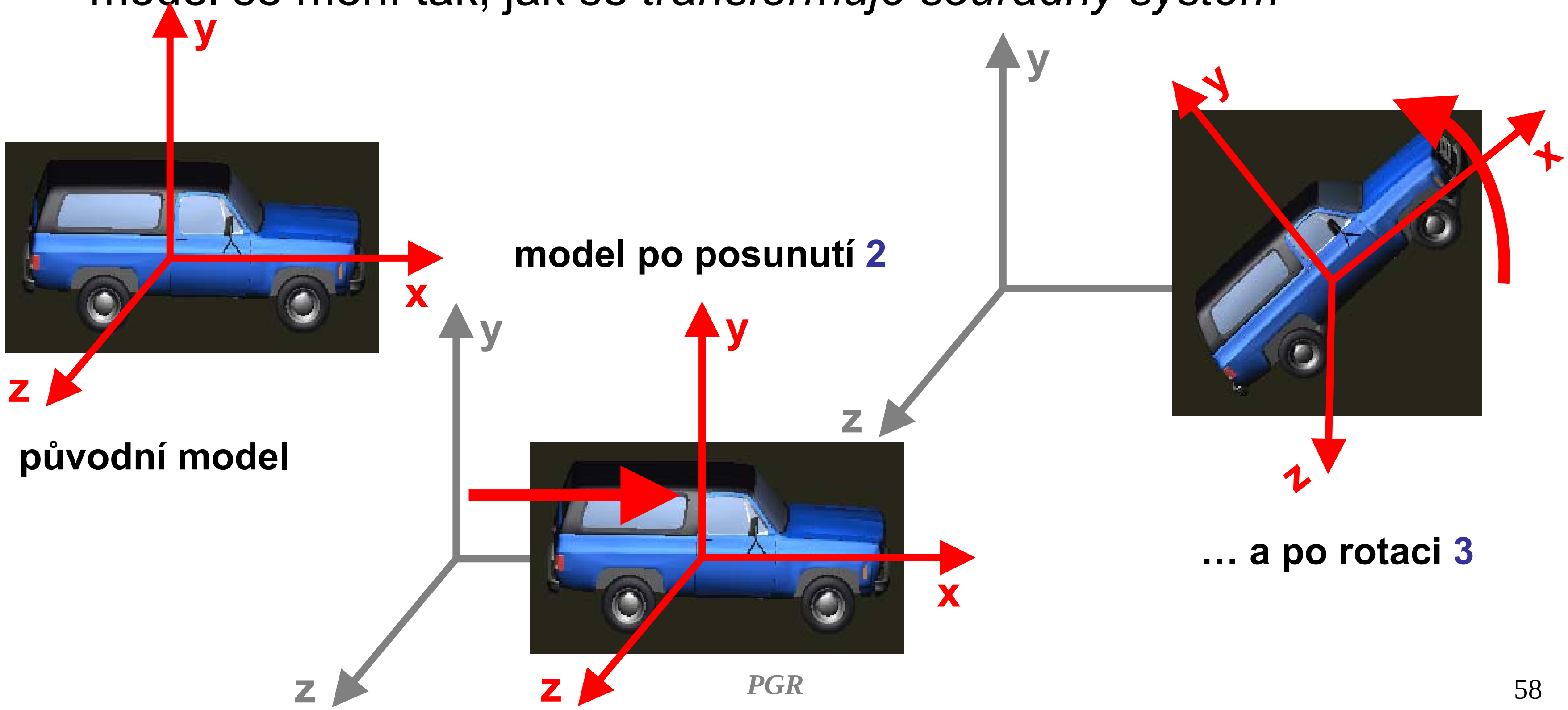
2) *zprava doleva* – každá další transformace se provede vůči původní „ globální“ soustavě

- Napřed rotace R vzhledem k $\vec{\boldsymbol{f}}^t$
- Pak translace T vzhledem k $\vec{\boldsymbol{f}}^t$

1) Interpretace $\vec{f}^t \Rightarrow \vec{f}^t TR$ zleva doprava

Pohybujeme lokálním souřadným systémem

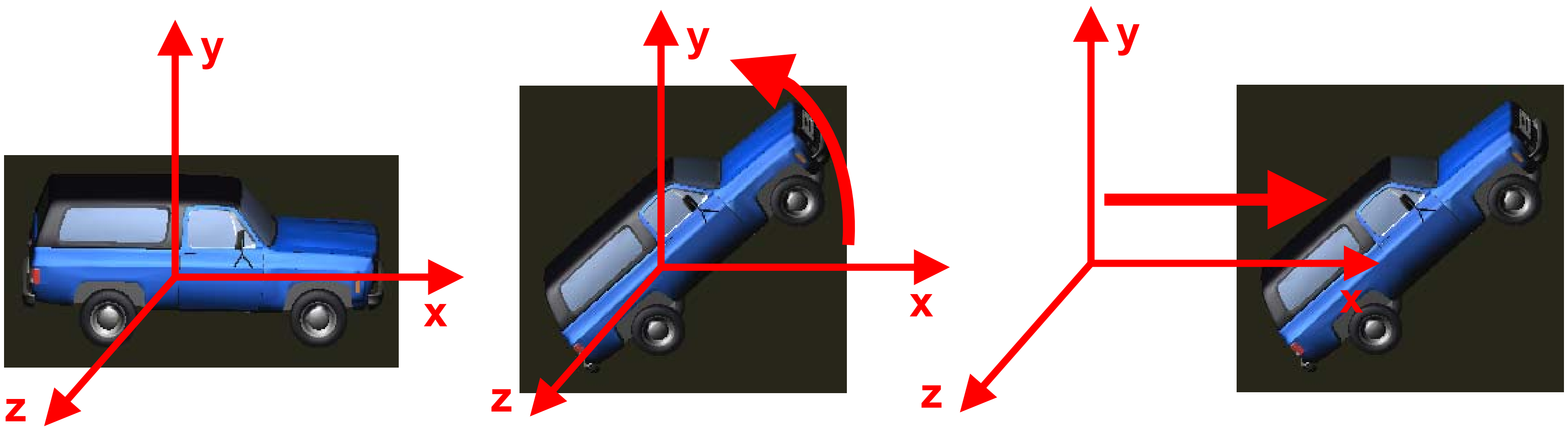
- další transformace vůči nově vzniklé „lokální“ soustavě
- lokální souřadný systém je pevně spojen s modelem
  model se mění tak, jak se *transformuje souřadný systém*



2) Interpretace $\vec{f}^t \Rightarrow \vec{f}^t TR$ zprava doleva

Pevný souřadný systém

- další transformace vůči původní „ globální“ soustavě
- V programu čteme odzadu dopředu

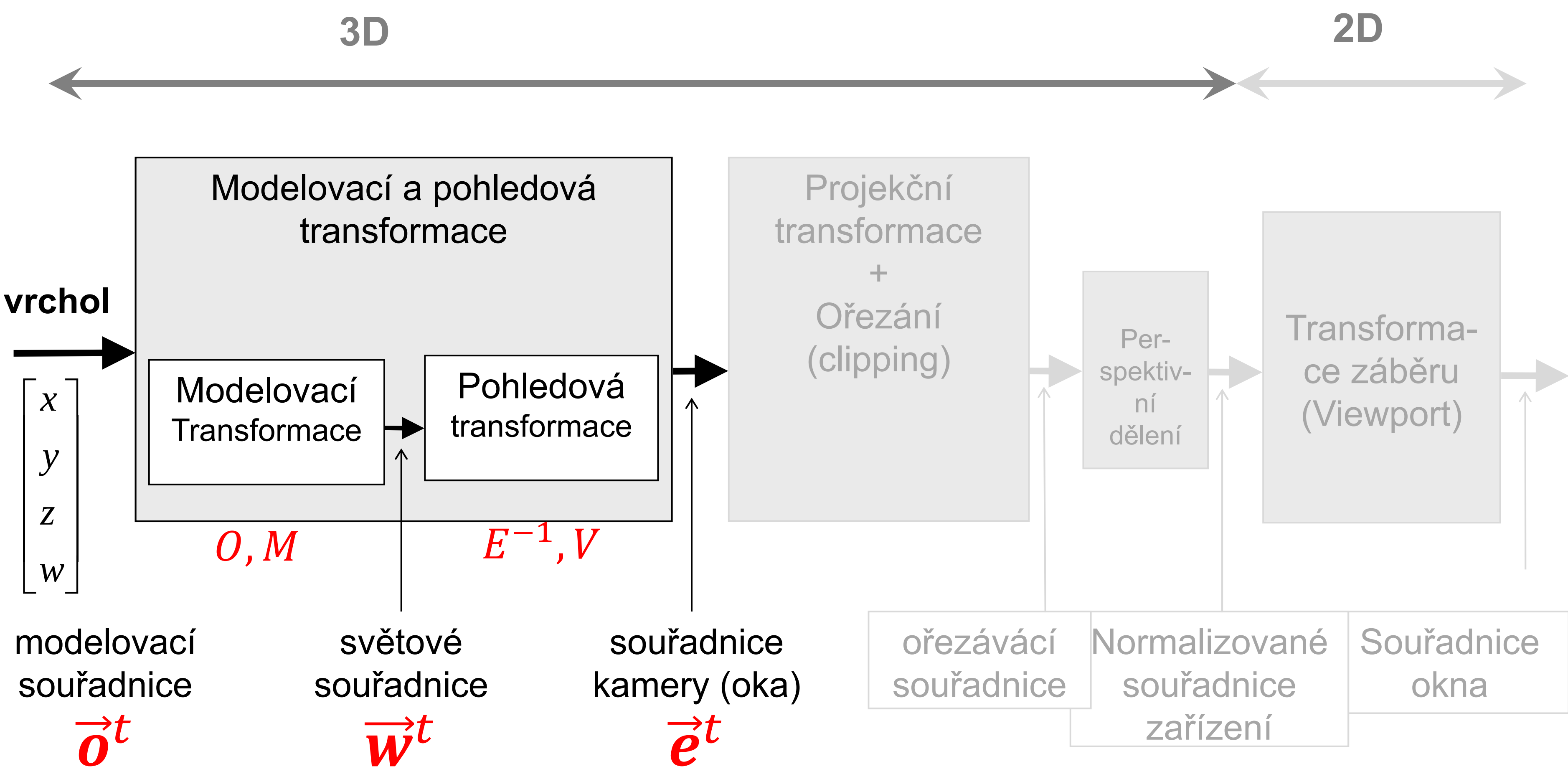


původní model 4 **model po rotaci 3** **… a po posunutí 2**

Souřadné soustavy v počítačové grafice

- Souřadné soustavy: světová, objektová a kamerová
- Pohyb objektů
- Pohyb kamerou
- Hierarchie transformací

Kroky při transformaci vrcholů



$\vec{o}^t = \vec{w}^t O$ $\quad$ $\vec{e}^t = \vec{w}^t E$

Světová soustava souřadnic (*world frame*)

Světová soustava souřadnic $\vec{\boldsymbol{w}}^t$

= základní pravotočivá soustava souřadnic

- je pevná (nikdy ji neměníme),
- v ní definujeme ostatní souřadné systémy,
- souřadnice v ní nazýváme světové souřadnice.

- Na modelování objektů vhodná moc není:
  - každá instance objektu by musela mít jiné světové souřadnice
  - při pohybu bychom museli neustále měnit souřadnice všech bodů hýbajícího se objektu

Objektová soustava souřadnic (*object frame*)

Objektová (modelová) soustava souřadnic $\vec{\boldsymbol{o}}^t$

= pravotočivá souřadná soustava pro modelování objektů,

- Umožnuje vymodelovat objekty
  (obvykle vycentrované okolo počátku a s jednotkovou velikostí)
  v objektových (modelových) souřadnicích, které se již při pohybu objektu nemění
- Při pohybu objektem se mění souřadná soustava $\vec{\boldsymbol{o}}^t$
- Pozici objektové souřadné soustavy vůči světové definuje matice O $(nazývaná\text{ též } M)$

$$\vec{\boldsymbol{o}}^t = \vec{\boldsymbol{w}}^t O, kde$$

- O je afinní rigidní (*rigid body*) matice 4x4
- Každý modelovaný objekt má jinou souřadnou soustavu $\vec{\boldsymbol{o}}^t$,
  proto má každý objekt svou jedinečnou matici O

Souřadná soustava kamery (*eye frame*)

Souřadná soustava kamery $\vec{\boldsymbol{e}}^t$

= pravotočivá ortonormální souřadná soustava popisující umístění a natočení kamery ve scéně

- Kamera v grafice je v počátku souřadné soustavy $\vec{\boldsymbol{e}}^t$ a hledí směrem k záporné ose *z*
- Matice kamery E je rigidní maticí 4x4

$$\vec{\boldsymbol{e}}^t = \vec{\boldsymbol{w}}^t E$$

- Souřadnice v soustavě kamery (*eye coordinates*) určují, kde se který vrchol objeví ve výsledném 2D obrázku
- Proto se počítají pro každý vrchol

Výpočet souřadnic bodu v prostoru kamery

- Souřadnice v soustavě kamery (*eye coordinates*) určují, kde se který vertex objeví ve výsledném 2D obrázku
- Proto se počítají pro každý vrchol

$$\tilde{p} = \vec{\boldsymbol{o}}^t \mathbf{c}$$
$$= \vec{\boldsymbol{w}}^t O\mathbf{c}$$
$$= \vec{\boldsymbol{e}}^t E^{-1} O\mathbf{c}$$

$\tilde{p}$	= geometrický bod
$\mathbf{c}$	= objektové souřadnice bodu $\tilde{p}$
$O\mathbf{c}$	= světové souřadnice bodu $\tilde{p}$
$E^{-1}O\mathbf{c}$	= kamerové souřadnice bodu $\tilde{p}$

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = E^{-1} O \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

o	= index objektové souřadnice
w	= index světové souřadnice
e	= index kamerové souřadnice bodu $\tilde{p}$

$E^{-1}O\mathbf{c}$ $\qquad\qquad$ $\mathbf{c}$

Transformace v grafickém programu

V grafickém programu se obvykle uchovávají matice

- O, M – pro převod z objektových do světových souřadnic
  - Ta se nazývá modelová matice, *model matrix*
  - Existuje jedna pro každý objekt
- $V = E^{-1}$ – pro převod ze světových souřadnic do prostoru kamery
  - Pohledová matice, *view matrix*
  - Existuje jedna pro každou kameru
- Dále se ukládá projekční matice, a matice záběru
  - o nich budeme mluvit příště

$$\tilde{p} = \vec{\boldsymbol{o}}^t \mathbf{c} = \vec{\boldsymbol{w}}^t O \mathbf{c} = \vec{\boldsymbol{e}}^t E^{-1} O \mathbf{c}$$

Změna matice O při transformaci objektu vzhledem k ke zvolené soustavě souřadnic

- Transformace objektové soustavy souřadnic maticí M vzhledem k pomocné soustavě souřadnic $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{w}}^t A$

$\vec{\boldsymbol{o}}^t$ ... soustava souřadnic objektu (modelová)

$= \vec{\boldsymbol{w}}^t O$ ... světová soustava souřadnic

$= \vec{\boldsymbol{a}}^t A^{-1} O$ ... pomocná soustava souřadnic

$\Rightarrow \vec{\boldsymbol{a}}^t M A^{-1} O$ ... transformovaná pomocná soustava souřadnic

$= \vec{\boldsymbol{w}}^t A M A^{-1} O$ ... světová soustava souřadnic transformovaná vzhledem k pomocné

Nová matice O bude mít hodnotu $O \leftarrow AMA^{-1}O$

Volba vztažné soustavy $\vec{a}^t$ k transformaci 1/2

$$0 \leftarrow AMA^{-1}O$$
$$0 \leftarrow OMO^{-1}O$$
$$0 \leftarrow OM$$

Transformujeme vůči $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{w}}^t A$:

a) Objektové soustavě souřadnic ($\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{o}}^t$)
   - Transformujeme přímo matici $O \leftarrow OM$
   - Pro bod: $\tilde{p} = \vec{\boldsymbol{o}}^t \mathbf{c} \Rightarrow \vec{\boldsymbol{o}}^t M \mathbf{c}$
   - Nevýhodou je, že se směr pohybu může při změně pohledu obrátit
   - Např. směr doprava v $\vec{\boldsymbol{o}}^t$ vůbec nemusí souviset se směrem doprava na obrázku (v $\vec{\boldsymbol{e}}^t$)

b) Kameře (vzhledem k $\vec{\boldsymbol{e}}^t$, $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{e}}^t$)
   - Vyřeší se problém s osami
   - Objekt ale krouží kolem počátku kamerové soustavy souřadnic

c) Středu objektu, ale podle os kamery

…

Volba vztažné soustavy $\vec{a}^t$ k transformaci 2/2

c) Rotace vůči středu objektu, ale podle os kamery

- Matice rozložíme na translační a rotační část
  - $O = (O)_T(O)_R$
  - $E = (E)_T(E)_R$
- $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{w}}^t(O)_T(E)_R$
- Stejného efektu (otáčení objektu okolo středu)
  - Dosáhneme přímou úpravou matice O
  - Jejím otočením okolo vektoru se souřadnicemi $\mathbf{k}_e$, transformovaného ze souřadnic kamery do souřadnic objektu $\mathbf{k}_o$
  - $O \leftarrow OM'$, kde M'je rotace okolo obecné osy $\mathbf{k}_o$ v objektové soustavě souřadnic.
  - $\mathbf{k}_o = O^{-1}E\mathbf{k}_e$ … souřadnice obecného vektoru – osy otáčení

$$\vec{\boldsymbol{e}}^t = \vec{\boldsymbol{w}}^t E = \vec{\mathbf{o}}^t O^{-1} E$$

$$\mathbf{k}_o = O^{-1}E\mathbf{k}_e$$

Změna matice kamery E

Přemístění kamery do jiného místa

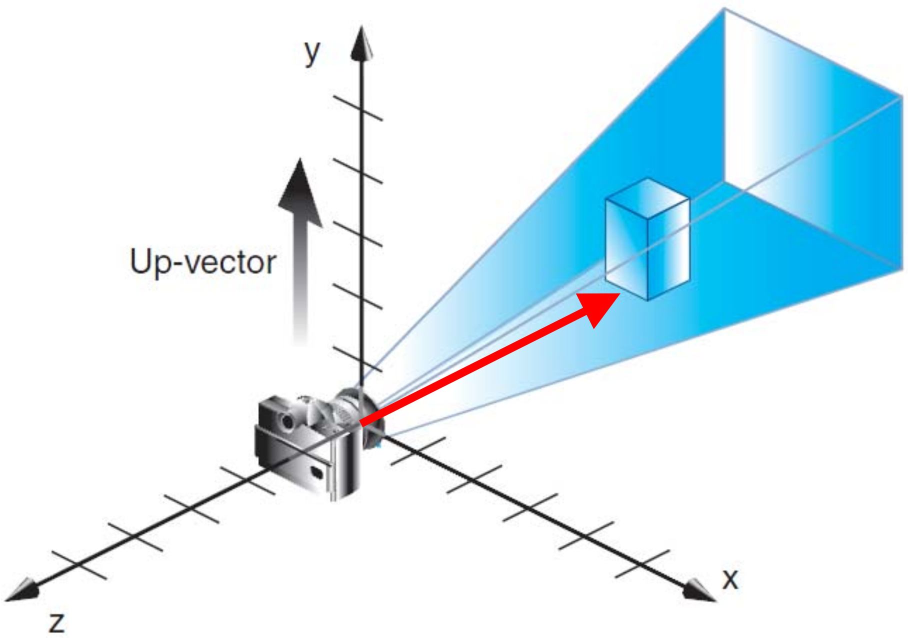
a) Pomocí pomocné soustavy souřadnic – jako u bodů
- Kamera bude kroužit okolo středu objektu
- (V LookAt měníme center)

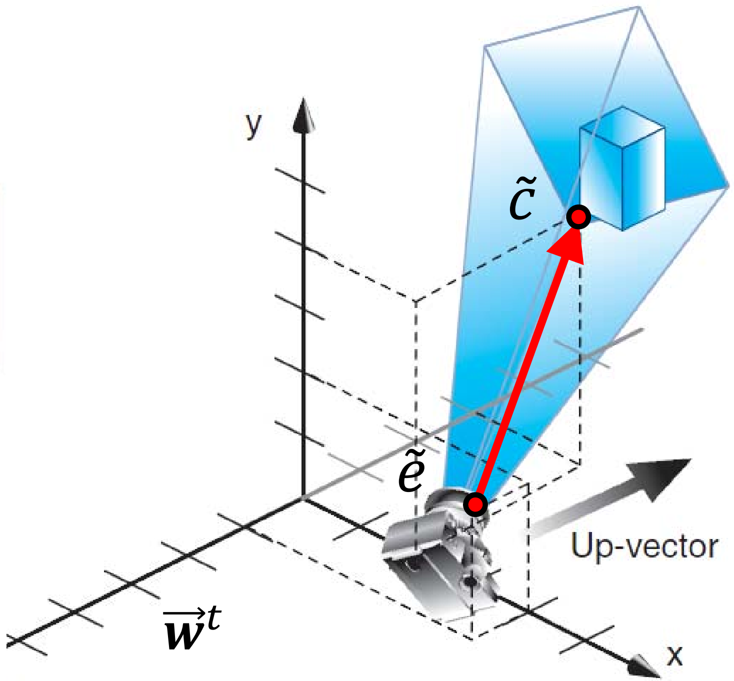
b) Přímo v soustavě souřadnic kamery
- Simuluje pohyb osoby kameramana – např. otáčení hlavou
- Používá se při simulaci pohledu z pozice hráče (FPS)
- (V LookAt měníme eye)

Kamera v základní poloze a metoda LookAt

- V základní poloze je kamera v počátku
- Hledí směrem k ose $-z$
- *Up-vector* směřuje k $+\mathrm{y}$

Metoda LookAt umístí kameru do scény, nasměruje objektiv a nakloní ji





[Redbook]

LookAt

$\vec{y}$ vznikne jako normalizovaný průmět vektoru $\overrightarrow{up}$ do roviny kamery (xy)

Dáno: $\tilde{c} = center, \tilde{e} = eye, \overrightarrow{up}$

Resp: jejich světové souřadnice $\mathbf{c}, \boldsymbol{e}, \boldsymbol{up}$

$$\boldsymbol{z} = normalize(\mathbf{c} - \boldsymbol{e})$$

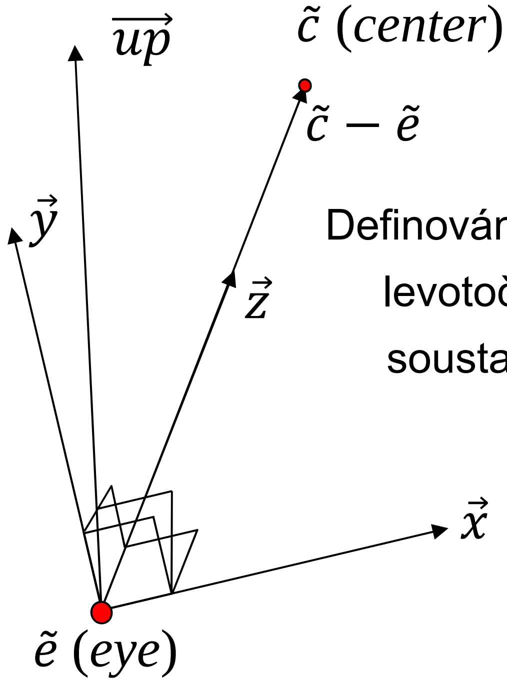
$$\boldsymbol{x} = normalize(\boldsymbol{z} \times \boldsymbol{up})$$

$$\mathbf{y} = (\boldsymbol{x} \times \boldsymbol{z})$$

kde $normalize(\boldsymbol{v}) = \boldsymbol{v} / \sqrt{v_1^2 + v_2^2 + v_3^2}$

$$E = \begin{bmatrix} x_1 & y_1 & -z_1 & e_1 \\ x_2 & y_2 & -z_2 & e_2 \\ x_3 & y_3 & -z_3 & e_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pravotočivá Soustava (-z místo z)



Definována jako levotočivá soustava

LookAt

- Inverzní matici E^{-1}, která se používá jako pohledová matice V, lze získat jako $((E)_R)^t((E)_T)^{-1}$
- Tedy

$$V = E^{-1} = \underbrace{\begin{bmatrix} x_1 & x_2 & x_3 & 0 \\ y_1 & y_2 & y_3 & 0 \\ -z_1 & -z_2 & -z_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\left((E)_R\right)^t} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -e_1 \\ 0 & 1 & 0 & -e_2 \\ 0 & 0 & 1 & -e_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\left((E)_T\right)^{-1} = E_{-T}}$$

- Využijeme rozkladu E na translační a rotační část

$$E = (E)_T(E)_R$$

Hierarchie transformací

- Při popisu objektu, složeného z více pohyblivých částí
  - každá část mívá svoji ortonormální soustavu souřadnic
  - chceme zachovat možnost hýbat celým objektem, i částmi
  - musíme tedy transformace částí definovat ne ve světové soustavě souřadnic, ale vždy v soustavě nadřazené části

    $\vec{\boldsymbol{o}}^t = \vec{\boldsymbol{w}}^t O$ tělo robota

    $\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{o}}^t A$ rameno

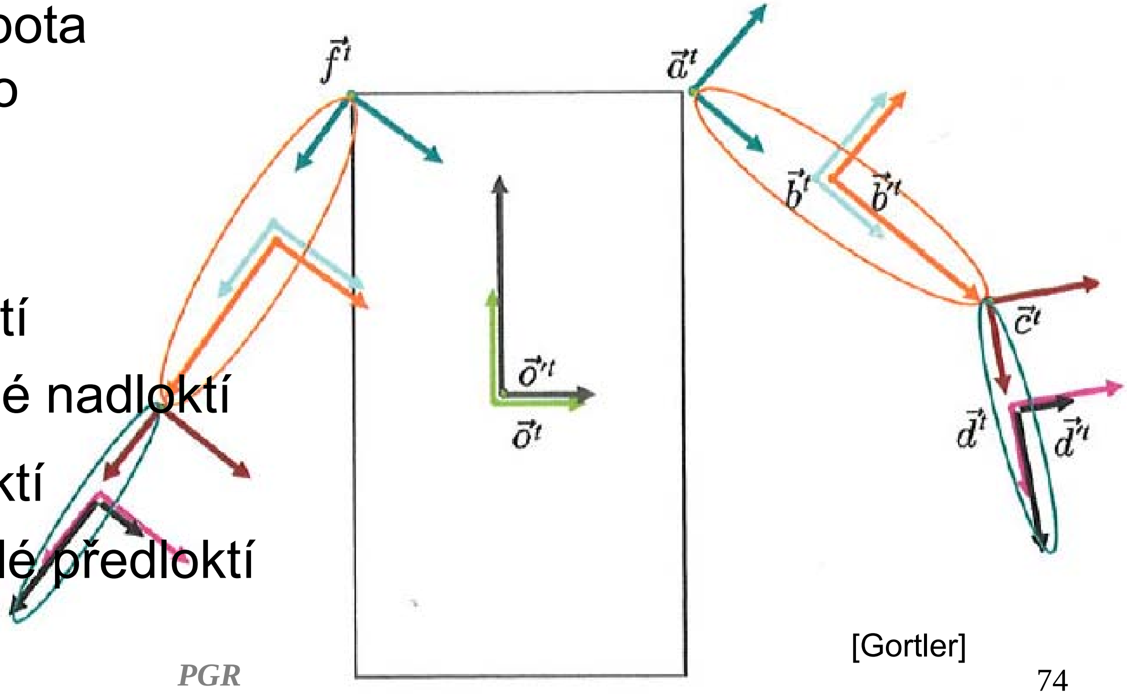
    $\vec{\boldsymbol{c}}^t = \vec{\boldsymbol{a}}^t C$ loket

  - $\vec{\boldsymbol{b}}^t = \vec{\boldsymbol{a}}^t B$ nadloktí

    $\vec{\boldsymbol{b}}'^t = \vec{\boldsymbol{b}}^t B'$ protáhlé nadloktí

  - $\vec{\boldsymbol{d}}^t = \vec{\boldsymbol{c}}^t D$ předloktí

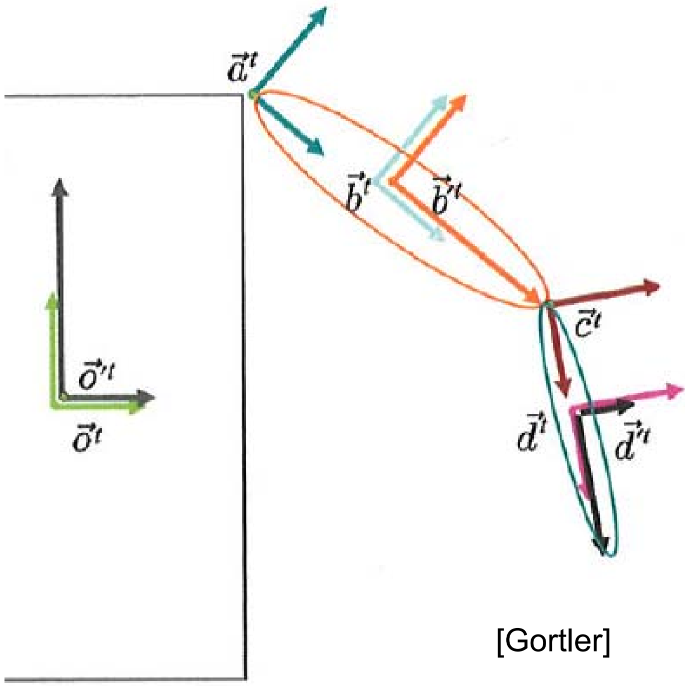
    $\vec{\boldsymbol{d}}'^t = \vec{\boldsymbol{d}}^t D'$ protáhlé předloktí



[Gortler]

Hierarchie transformací

- Při vykreslování celého objektu skládáme transformace
  - tělo: $E^{-1}O$
  - Rameno: $E^{-1}OA$
  - Nadloktí: $E^{-1}OABB'$



[Gortler]

$\vec{\boldsymbol{o}}^t = \vec{\boldsymbol{w}}^t O$ tělo robota
$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{o}}^t A$ rameno
$\vec{\boldsymbol{c}}^t = \vec{\boldsymbol{a}}^t C$ loket

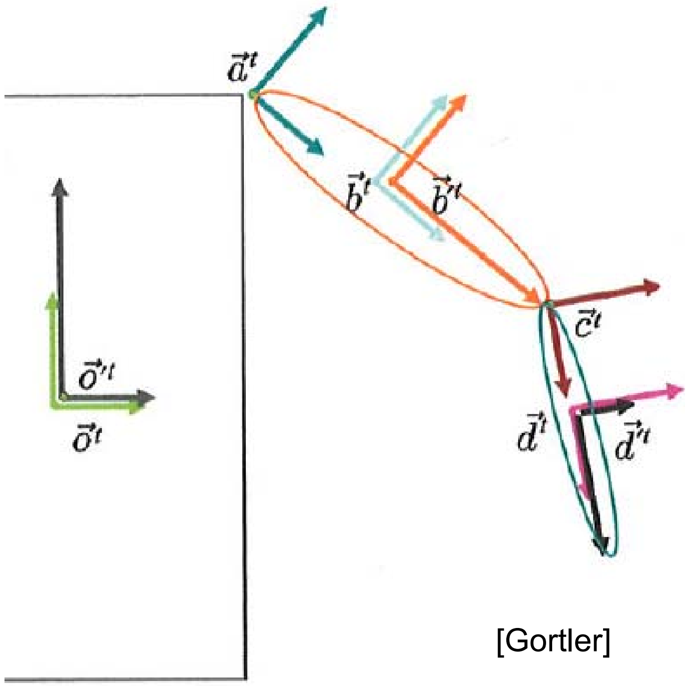
$\vec{\boldsymbol{b}}^t = \vec{\boldsymbol{a}}^t B$ nadloktí
$\overrightarrow{\boldsymbol{b}'}^t = \vec{\boldsymbol{b}}^t B'$ protáhlé nadloktí
$\vec{\boldsymbol{d}}^t = \vec{\boldsymbol{c}}^t D$ předloktí
$\overrightarrow{\boldsymbol{d}'}^t = \vec{\boldsymbol{d}}^t D'$ protáhlé předloktí

Hierarchie transformací

- Při pohybu částí
  měníme transformace částí
  - Celý robot – měníme O
  - Ruka v rameni – měníme A
  - Loket – měníme C

$\vec{\boldsymbol{o}}^t = \vec{\boldsymbol{w}}^t O$ tělo robota
$\vec{\boldsymbol{a}}^t = \vec{\boldsymbol{o}}^t A$ rameno
$\vec{\boldsymbol{c}}^t = \vec{\boldsymbol{a}}^t C$ loket

$\vec{\boldsymbol{b}}^t = \vec{\boldsymbol{a}}^t B$ nadloktí
$\vec{\boldsymbol{b}'}^t = \vec{\boldsymbol{b}}^t B'$ protáhlé nadloktí
$\vec{\boldsymbol{d}}^t = \vec{\boldsymbol{c}}^t D$ předloktí
$\vec{\boldsymbol{d}'}^t = \vec{\boldsymbol{d}}^t D'$ protáhlé předloktí



[Gortler]

Hierarchie transformací

- Hierarchii transformací zaznamenáme
  - v grafu scény jako hierarchii uzlů – p593t2
  - v programu použitím zásobníku matic

```
matrixStack.init(inv(E));   // E na TOS
matrixStack.push (O);      // zduplikuje TOS, vynásobí TOS O - EO
  matrixStack.push (O’); // zduplikuje TOS, vynásobí TOS O’- EOO’
    drawStretchedBody – vykreslí objekt s transformací TOS
  matrixStack.pop();       // ubere TOS, vrátí na TOS EO

matrixStack.push (A); // zduplikuje TOS, vynásobí TOS O’- EOA
matrixStack.push (B); // zduplikuje TOS, vynásobí TOS O’- EOAB
    drawStretchedArm – vykreslí ruku s transformací TOS
  matrixStack.pop();       // ubere TOS, vrátí na TOS EOA
```

Mikropřehled transformací a jejich inverzí

Identita

Jednotková matice, například funkce M = **Identity()**

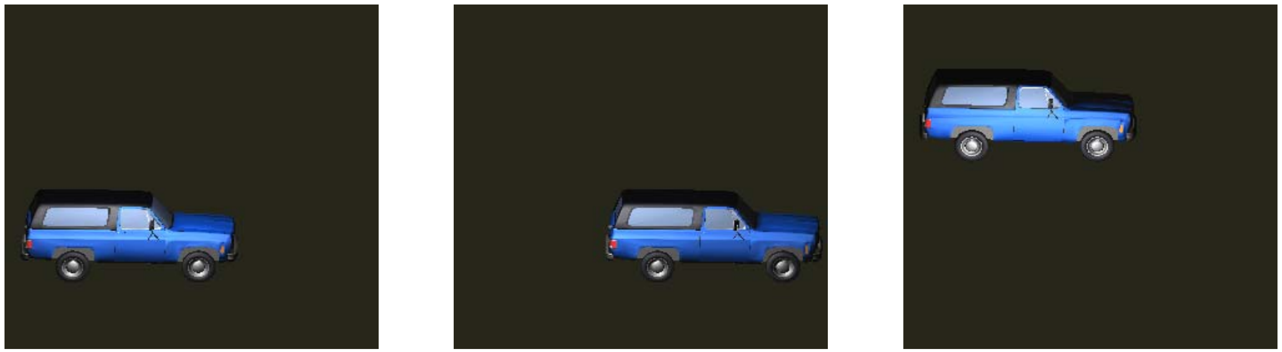
$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Modelovací transformace

Matice posunutí (translace)

- realizuje posunutí dané vektorem **(mx, my, mz)** (resp. posune lokální soustavu souřadnic o stejné hodnoty)

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & mx \\ 0 & 1 & 0 & my \\ 0 & 0 & 1 & mz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{a} \quad \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -mx \\ 0 & 1 & 0 & -my \\ 0 & 0 & 1 & -mz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Modelovací transformace

Matice pro změnu měřítka (škálování)

- resp., souřadné osy lokální soustavy souřadnic se prodlouží či zkrátí dle **sx**, **sy** a **sz** as nimi se transformuje i přidružený objekt

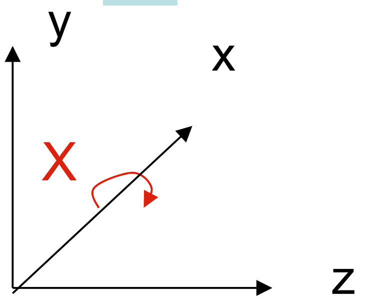
$$\mathbf{S} = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{a} \quad \mathbf{S}^{-1} = \begin{bmatrix} 1/sx & 0 & 0 & 0 \\ 0 & 1/sy & 0 & 0 \\ 0 & 0 & 1/sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



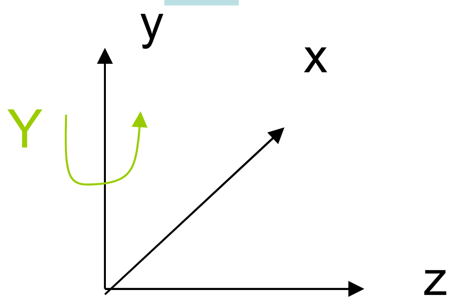
Modelovací transformace

Speciální případy rotací – rotace podle souřadnicových os

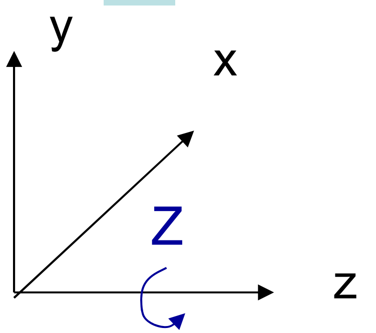
X



Y



Z



$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

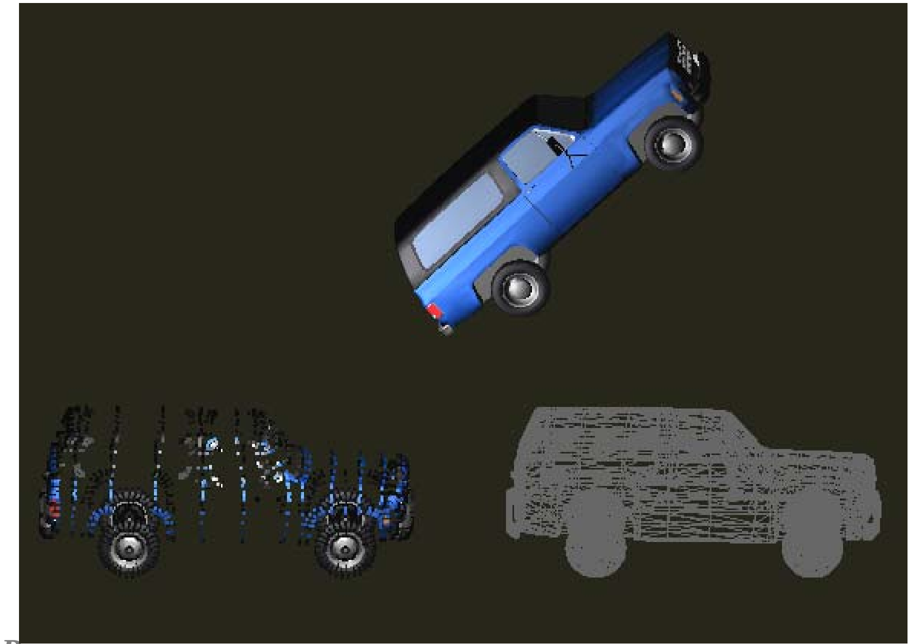
Inverzní rotace = rotace okolo stejné osy o opačný úhel $-\theta$

Rotace není komutativní

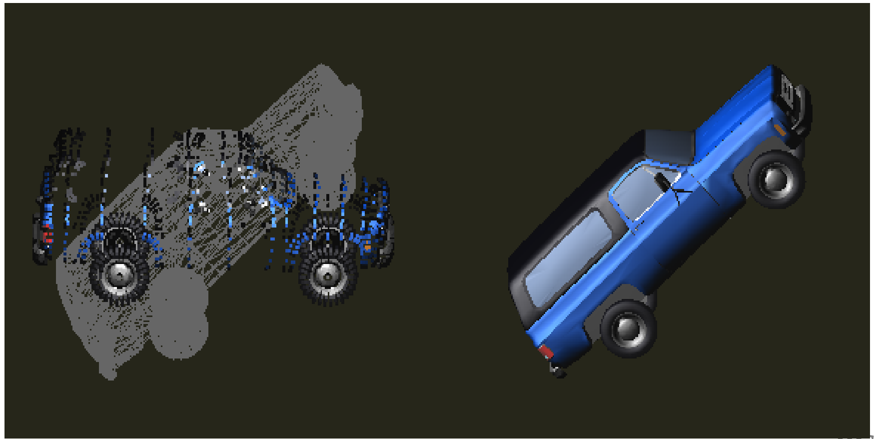
Pozor! Záleží na pořadí transformací, neboť
maticové násobení není komutativní!

⇒ tj., R.T není totéž co T.R

R.T.[ ]



T.R.[ ]



Pohledová transformace - LookAt

M = LookAt(…);

passMatrixToVS(M);

drawModel();

LookAt((1.0, 0.0, 1.0),
(0.0, 0.0, 0.0),
(0.0, 1.0, 0.0));

LookAt((-1.0, -0.5, 1.0),
(0.0, 0.0, 0.0),
(0.0, 1.0, 0.0));

LookAt((1.0, 1.0, 0.0),
(0.0, 0.0, 0.0),
(-1.0, 0.0, 0.0));

Transformace v OpenGL

Lineární složka: rotace (zkosení)
rotation (shear)

změna měřítka
scale

posunutí
translation

$$v' = [x', y', z', w']^T = M.v = \begin{bmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

transformovaný vrchol

projekce
projection

(příští týden)

vrchol před transformací

vertex to be transformed

[Sloup]

Obecná transformace pro zobrazení

- Cílem je složit ze tří matic správnou matici:

M = projection * view * model

- Matici předáme do „Vertex Shaderu“ (uložena po sloupcích)
- „Vertex shader“ provede násobení každého vrcholu

$$v' = [x', y', z', w']^T = M.v = \begin{bmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

souřadnice k zobrazení
na obrazové rovině

souřadnice v objektovém prostoru

Předání matice do „Vertex Shader“

- Ukázka pro předání všech tří matic a násobení vektoru maticemi přímo na grafické kartě:

```
#version 330

in vec4 position;

uniform mat4 projection ;   // P
uniform mat4 view ;         // E-1, V
uniform mat4 model;         // O, M

void main() {
   vec4 worldPos = model * position;
   vec4 cameraPos = view * worldPos;
   gl_Position = projection * cameraPos;
}
```

Na straně CPU je se matice nahraje na GPU pomocí následujícího kódu

```
projectionMatrixLoc = glGetUniformLocation(theProgram,
   “projection ”);
float fFrustumScale = 1.0f;
float fzNear = 0.5f; float fzFar = 3.0f;
float matrix[16];
memset(matrix, 0, sizeof(float) * 16);
   matrix[0] = fFrustumScale;
   matrix[5] = fFrustumScale;
   matrix[10] = (fzFar + fzNear) / (fzNear - fzFar);
   matrix[14] = (2 * fzFar * fzNear) / (fzNear - fzFar);
   matrix[11] = -1.0f;
glUseProgram(theProgram);
glUniformMatrix4fv(projectionMatrixLoc , 1, GL_FALSE,
   matrix);
glUseProgram(0);
```