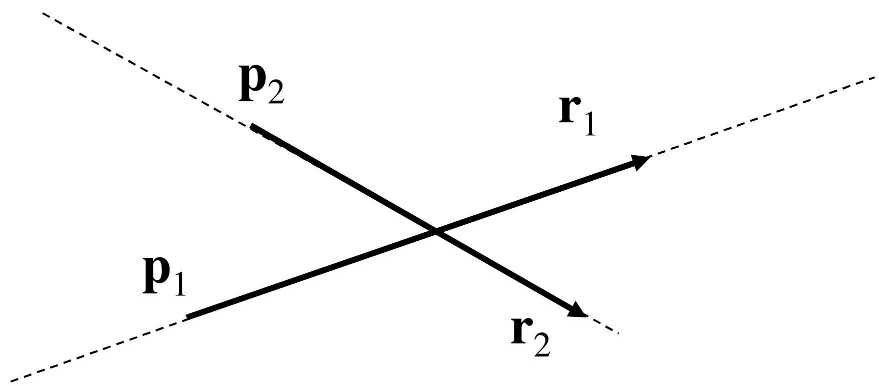
Homogene transformacije

Dr. sci. Emir Skejić, vanr. prof.
Fakultet elektrotehnike Tuzla

Primjer: Rastojanje između linija

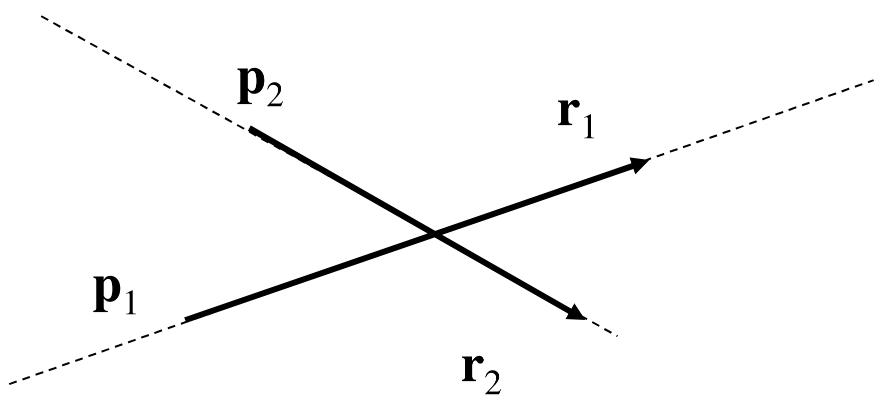
- Imamo dvije 3D linije koje se ne sijeku u prostoru
- Prva linija je definirana pomoću tačke $\mathbf{p}_1$ na liniji i linijskog vektora $\mathbf{r}_1$, a druga linija pomoću tačke $\mathbf{p}_2$ i linijskog vektora $\mathbf{r}_2$
- Odrediti najkraće rastojanje između ove dvije linije



Primjer: Rastojanje između linija

- Želimo odrediti dužinu vektora između dvije tačke projektovanog na osu ortogonalnu na dvije linije

$$rastojanje = \frac{(\mathbf{p}_2 - \mathbf{p}_1)\cdot(\mathbf{r}_1 \times \mathbf{r}_2)}{|\mathbf{r}_1 \times \mathbf{r}_2|}$$



Primjer: Rastojanje između linija

$$rastojanje = \frac{(\mathbf{p}_2 - \mathbf{p}_1)\cdot(\mathbf{r}_1 \times \mathbf{r}_2)}{\left|\mathbf{r}_1 \times \mathbf{r}_2\right|}$$

- Primijetimo da znak rezultata može biti pozitivan ili negativan zavisno od geometrijske konfiguracije, tako da moramo uzeti apsolutnu vrijednost ukoliko želimo pozitivnu udaljenost
- Također, recimo i to da jednačina može biti nedefinirana ako je $|\mathbf{r}_1 x \mathbf{r}_2|=0$, što znači da su linije paralelne. Potrebno je uraditi provjeru za ovu situaciju i upotrijebiti drugačiju jednačinu ako su linije paralelne

Linearne transformacije

$$v_x' = a_1 v_x + b_1 v_y + c_1 v_z + d_1$$
$$v_y' = a_2 v_x + b_2 v_y + c_2 v_z + d_2$$
$$v_z' = a_3 v_x + b_3 v_y + c_3 v_z + d_3$$

$$\begin{bmatrix} v_x' \\ v_y' \\ v_z' \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v} + \mathbf{d}$$

Višestruke transformacije

- Moguće je primijeniti bilo koju sekvencu transformacija:

$$\mathbf{v}' = \mathbf{M}_4 \cdot \left(\mathbf{M}_3 \cdot \left(\mathbf{M}_2 \cdot \left(\mathbf{M}_1 \cdot \mathbf{v}\right)\right)\right)$$

- Zbog toga što za matričnu algebru vrijedi zakon *asocijativnosti*, prethodni izraz se može napisati u obliku:

$$\mathbf{v}' = \left(\mathbf{M}_4 \cdot \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1\right) \cdot \mathbf{v}$$

- Ovo nam dozvoljava da matrice *spojimo* u jednu matricu:

$$\mathbf{M}_{total} = \mathbf{M}_4 \cdot \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1$$

$$\mathbf{v}' = \mathbf{M}_{total} \cdot \mathbf{v}$$

- Napomena: Za množenje matrica NE vrijedi zakon *komutativnosti*, tako da je potrebno voditi računa o redoslijedu množenja

Proizvod dvije matrice

$$\mathbf{L} = \mathbf{M} \cdot \mathbf{N}$$

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \cdot \begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix}$$

$$l_{12} = m_{11}n_{12} + m_{12}n_{22} + m_{13}n_{32}$$

Višestruke rotacije i skaliranja

- Sekvenca rotacija i skaliranja može se kombinirati u jednu matricu
- Naprimjer, može se kombinirati y-rotacija, nakon koje slijedi z-rotacija, potom neuniformno skaliranje, te konačno x-rotacija:

$$\mathbf{M} = \mathbf{R}_x(\gamma) \cdot \mathbf{S}(\mathbf{s}) \cdot \mathbf{R}_z(\beta) \cdot \mathbf{R}_y(\alpha)$$

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v}$$

Višestruke translacije

- Također, možemo iskoristiti svojstvo asocijativnosti koje vrijedi za sabiranje vektora da bismo kombinirali sekvencu translacija
- Naprimjer, translacija duž vektora t1 nakon koje slijedi translacija duž t2 te konačno duž vektora t3 mogu biti kombinirane:

$$\mathbf{d} = \mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3$$
$$\mathbf{v}' = \mathbf{v} + \mathbf{d}$$

Kombiniranje transformacija

- Vidimo da možemo kombinirati sekvencu rotacija i/ili skaliranja
- Također, možemo kombinirati i sekvencu translacija
- Međutim, šta ako želimo kombinirati translacije *s* rotacijama/skaliranjima?

Homogene koordinate i računarska grafika

- Homogene koordinate su ključne za sve računarske grafičke sisteme
  - Sve standardne transformacije (rotacija, translacija, skaliranje) mogu biti implementirane matričnim množenjem pomoću 4x4 matrica
  - Hardverski cjevovod radi sa 4-dimenzionalnim reprezentacijama
  - Za ortografsko gledanje možemo zadržati $w = 0$ za vektore i $w = 1$ za tačke
  - Za perspektivu nam je potrebno *perspektivno dijeljenje*

Homogene koordinate

- Dijeljenje s nulom je jedan od čestih problema pri geometrijskim proračunima (npr. izračun sjecišta dva paralelna pravca)
- Da bi se ovakvi problemi izbjegli, uvodi se homogeni prostor
- Dodati dodatnu koordinatu:

  $P = (p_x, p_y, p_z, p_h)$ ili

  $\boldsymbol{x} = (x, y, z, h)$
- Kartezijeve koordinate: podijeliti sa h

  $\boldsymbol{x} = (\frac{x}{h}, \frac{y}{h}, \frac{z}{h})$

Homogene transformacije

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v}$$

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

$$v'_x = a_1 v_x + b_1 v_y + c_1 v_z + d_1$$

$$v'_y = a_2 v_x + b_2 v_y + c_2 v_z + d_2$$

$$v'_z = a_3 v_x + b_3 v_y + c_3 v_z + d_3$$

$$1 = 0 v_x + 0 v_y + 0 v_z + 1$$

Homogene transformacije

- Dakle, mi smo u osnovi uzeli 3x3 matricu rotacije/skaliranja i 3x1 vektor translacije iz originalne jednačine i iskombinirali ih u novu 4x4 matricu (za sada, uvijek ćemo imati [0 0 0 1] u zadnjem redu matrice)
- Zamijenit ćemo 3D vektor pozicije **v** s njegovom 4D verzijom $[v_x\ v_y\ v_z\ 1]$ (za sada, uvijek ćemo imati vrijednost 1 za četvrtu koordinatu)
- Upotreba 4x4 matrica transformacije omogućuje nam da u jednu matricu iskombiniramo rotacije, translacije, skaliranja, smicanja i refleksije
- Na sljedećem predavanju vidjet ćemo da se 4x4 matrice transformacije mogu također koristiti za perspektivne projekcije i druge projekcije pogleda

Homogene transformacije

- Naprimjer, translacija za vektor **r** nakon koje slijedi rotacija oko *z*-ose može biti predstavljena kao:

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & r_x \\ 0 & 1 & 0 & r_y \\ 0 & 0 & 1 & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

Matrice rotacije

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotacija oko proizvoljne ose

- Također, rotacija se može vršiti i oko proizvoljne jedinične ose **a** za neki ugao θ

$$\mathbf{R}_a\left(\mathbf{a},\theta\right)=$$

$$\begin{bmatrix} a_x^2+c_\theta(1-a_x^2) & a_xa_y(1-c_\theta)-a_zs_\theta & a_xa_z(1-c_\theta)+a_ys_\theta & 0 \\ a_xa_y(1-c_\theta)+a_zs_\theta & a_y^2+c_\theta(1-a_y^2) & a_ya_z(1-c_\theta)-a_xs_\theta & 0 \\ a_xa_z(1-c_\theta)-a_ys_\theta & a_ya_z(1-c_\theta)+a_xs_\theta & a_z^2+c_\theta(1-a_z^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Eulerova teorema

- Eulerova teorema: Bilo koja dva nezavisna ortonormalna koordinatna okvira mogu biti povezana pomoću sekvence rotacija (ne više od tri) oko koordinatnih osa, pri čemu dvije sukcesivne rotacije ne smiju biti oko iste ose.
- Ne treba ovo pomiješati s Eulerovim uglovima, Eulerovom integracijom, Newton-Eulerovom dinamikom, *inviscid* Eulerovim jednačinama, Eulerovim karakteristikama…
- Leonard Euler (1707-1783)

Eulerova teorema

- Postoje dva korisna načina posmatranja Eulerove teoreme vezano za orijentaciju:
  1. Eulerova teorema kaže da se bilo koja orijentacija može predstaviti kao rotacija oko tri ose, naprimjer x, zatim y, potom z
  2. Također, kaže i to da se bilo koja orijentacija može predstaviti kao jednostruka rotacija oko neke ose za neki ugao

Orijentacija

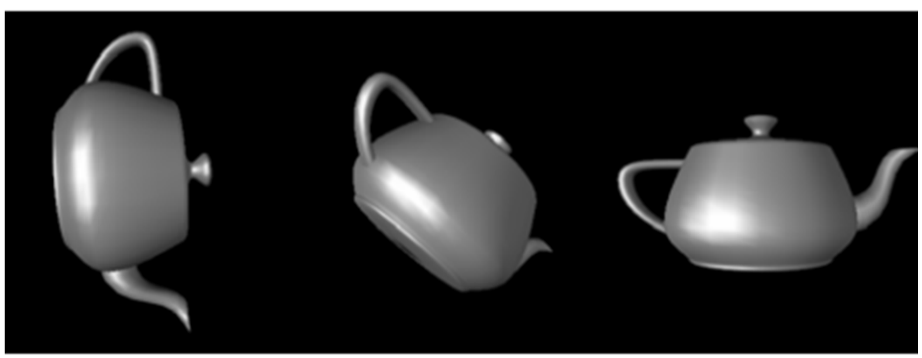
- Termin *orijentacija* se odnosi na ugaonu konfiguraciju objekta
- To je u suštini rotacijski ekvivalent "pozicije", no nažalost, ne postoji jednako jednostavna reprezentacija
- Eulerova teorema kaže da nisu potrebna više od 3 broja da bi se predstavila orijentacija
- Postoji nekoliko popularnih metoda za predstavljanje orijentacije. Neke od tih metoda koriste 3 broja, a druge pak više od 3 broja…

Orijentacija

- Jedna od metoda za pohranjivanje orijentacije je jednostavno ju pohraniti kao sekvencu rotacija oko glavnih osa, poput x, zatim y, a onda z. Ovo nazivamo *Eulerovim uglovima.* Ovo je kompaktan metod, jer koristi tačno 3 broja za pohranjivanje orijentacije
- Druga opcija je predstaviti orijentaciju kao osu jedinične dužine i skalarni ugao rotacije, tako da ukupno koristi 4 broja. Ovo je konceptualno jednostavan metod, ali ima ograničenu matematičku upotrebu te zbog toga nije previše uobičajen
- Treća opcija je upotreba 3x3 matrice za predstavljanje orijentacije (ili gornjeg 3x3 segmenta 4x4 matrice). Ovaj metod koristi 9 brojeva tako da mora sadržavati i neke dodatne informacije (tj. informacije o skaliranju i smicanju)
- Četvrta opcija je upotreba "kvaterniona", što je specijalna vrsta 4-dimenzionalnog vektora. Ovo je popularan i matematički moćan metod, ali pomalo čudnovat…
- Bez obzira koji metod se koristi za *pohranjivanje* informacija o orijentaciji, obično je informacije potrebno pretvoriti u matricu kako bi se zapravo transformirale tačke ili uradile druge korisne stvari

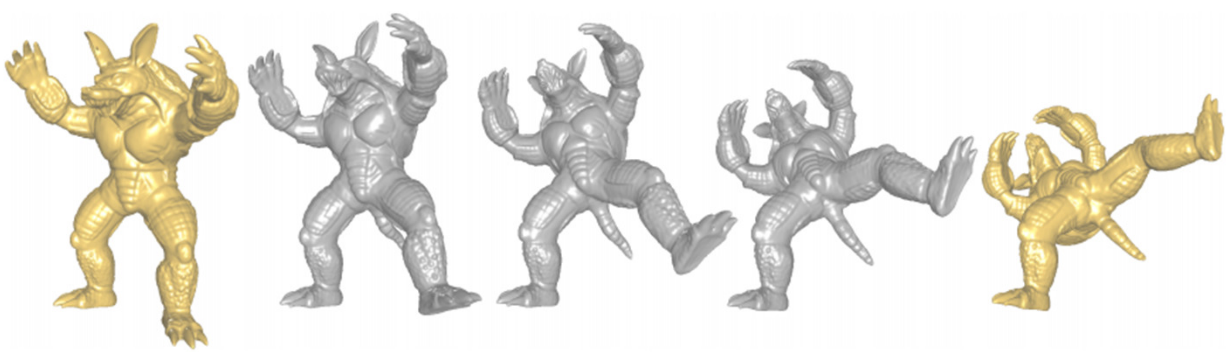
Kvaternioni

- Još jedan način za predstavljanje rotacije
- 4D brojevi: (a, b, c, d)
- Više prednosti u odnosu na 3x3 matrice rotacije
  - Manji prostor za pohranu (4 vs. 3x3=9)
  - Ulančavanje kvaterniona zahtijeva manje operacija
  - Lako se interpoliraju za glatke animacije



Kvaternioni

- Još jedan način za predstavljanje rotacije
- 4D brojevi: (a, b, c, d)
- Više prednosti u odnosu na 3x3 matrice rotacije
  - Manji prostor za pohranu (4 vs. 3x3=9)
  - Ulančavanje kvaterniona zahtijeva manje operacija
  - Lako se interpoliraju za glatke animacije



  - Artikulirana interpolacija ostvarena nakon segmentiranja objekta u krute dijelove i njihovog postavljanja

Skaliranje

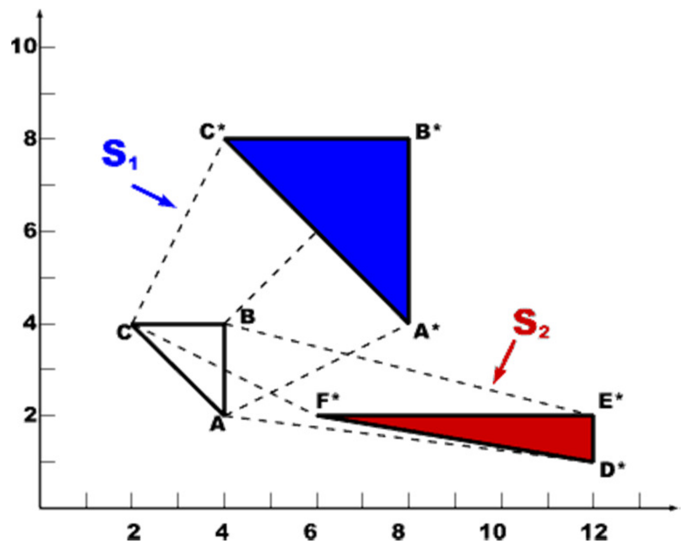
- Matrica uniformnog skaliranja skalira cijeli objekt faktorom skaliranja *s*

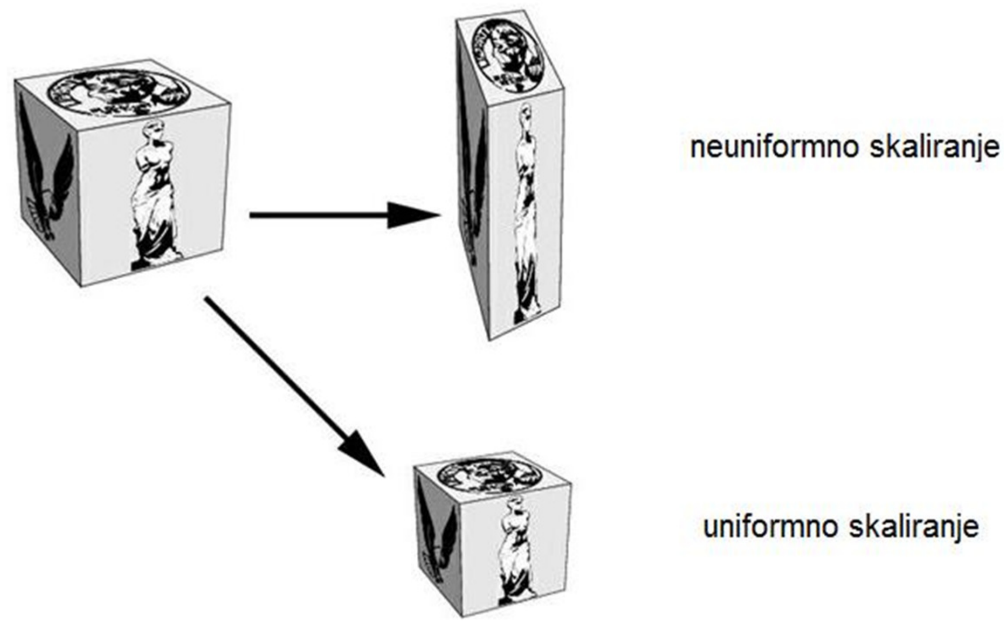
$$\mathbf{S}_u(s) = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Matrica neuniformnog skaliranja skalira nezavisno duž osa *x*, *y* i *z*

$$\mathbf{S}_n(\mathbf{s}) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Uniformno i neuniformno skaliranje





Skaliranje koje konzervira volumen

- Možemo primijetiti da će uniformno skaliranje faktorom *s* promijeniti volumen nekog objekta za s^3, a da će neuniformno skaliranje promijeniti volumen za $s_x s_y s_z$
- Ponekad želimo uraditi i *skaliranje koje konzervira volumen* (engl. *volume preserving scale*), koje efektivno rasteže jednu osu a sabija preostale dvije
- Naprimjer, skaliranje koje konzervira volumen, duž x-ose, faktorom s_x će ose y i z skalirati za $1/\mathrm{sqrt}(s_x)$
- Ovaj put, ukupna promjena volumena je

$$s_x \cdot \frac{1}{\sqrt{s_x}} \cdot \frac{1}{\sqrt{s_x}} = 1$$

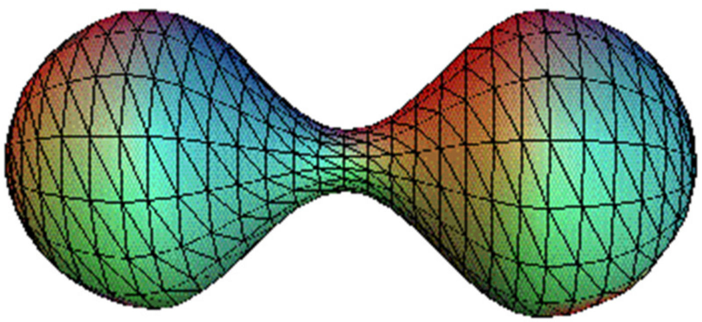
Skaliranje koje konzervira volumen

- Skaliranje koje konzervira volumen, duž *x*-ose:

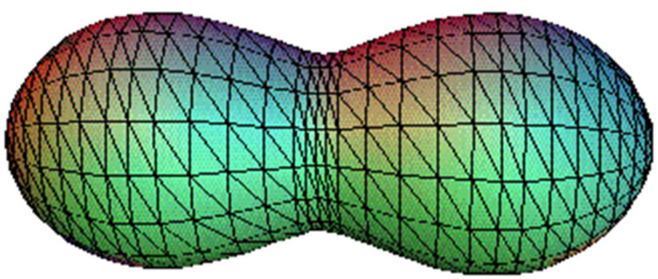
$$\mathbf{S}_{vx}(s_x) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & \dfrac{1}{\sqrt{s_x}} & 0 & 0 \\ 0 & 0 & \dfrac{1}{\sqrt{s_x}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Skaliranje koje konzervira volumen

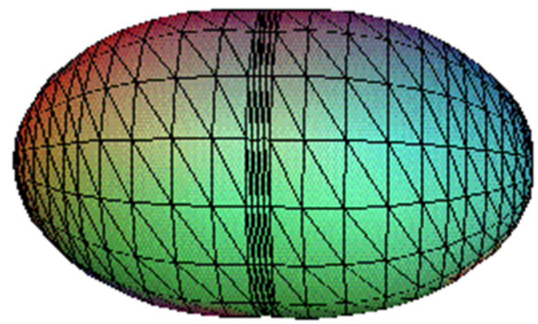
Početna konfiguracija



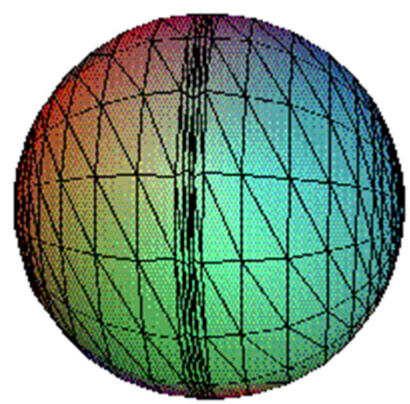
Nakon 500 vremenskih koraka



Nakon 1000 vremenskih koraka



Nakon 3000 vremenskih koraka



Smicanje

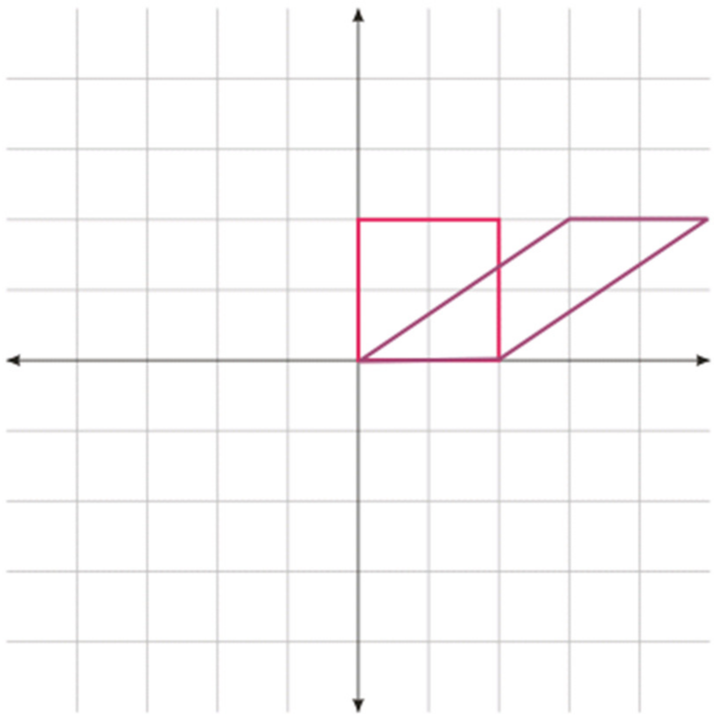
- Matrica *smicanja* (engl. *shear transformation*) izgleda ovako:

$$\mathbf{Z}(z_1 \ldots z_6) = \begin{bmatrix} 1 & z_1 & z_2 & 0 \\ z_3 & 1 & z_4 & 0 \\ z_5 & z_6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

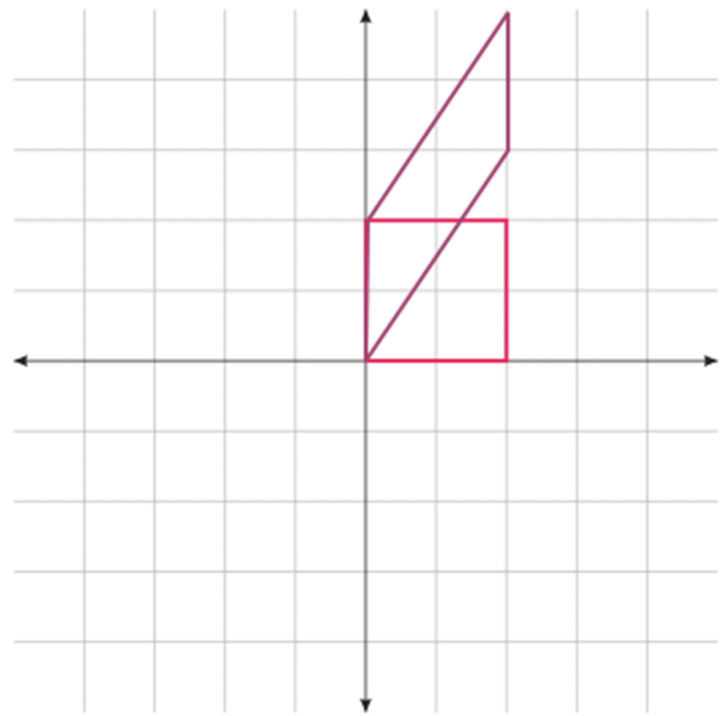
- Kod čistog smicanja samo jedna od konstanti je nenulta vrijednost
- Smicanje može biti interpretirano i kao neuniformno skaliranje duž rotiranog skupa osa
- Smicanja se ponekad koriste u računarskoj grafici za jednostavne deformacije ili efekte poput crtanih filmova

Smicanje

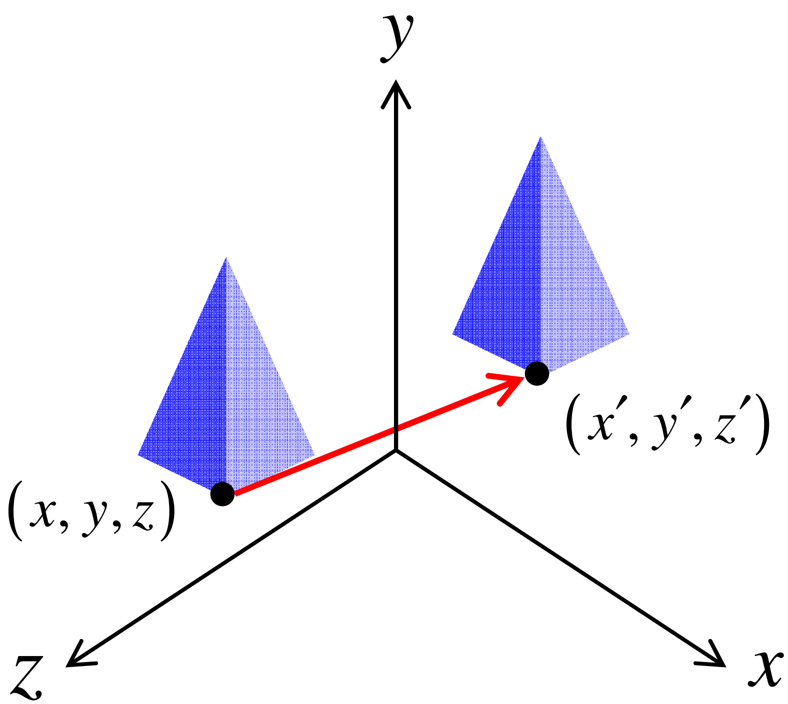
Smicanje duž x-ose



Smicanje duž y-ose



Translacija



$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Translacija

- 4x4 matrica translacije koja translira objekt za vektor **r** je:

$$\mathbf{T}(\mathbf{r}) = \begin{bmatrix} 1 & 0 & 0 & r_x \\ 0 & 1 & 0 & r_y \\ 0 & 0 & 1 & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pivot tačke

- Standardne matrice rotacije obrću objekt oko neke ose iz koordinatnog ishodišta
- Šta ako želimo da pivot tačka bude negdje drugdje?
- Sljedeća transformacija vrši rotaciju oko z-ose obrtanjem oko tačke **r**

$$\mathbf{M} = \mathbf{T}(\mathbf{r}) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(-\mathbf{r})$$

Opšta 4x4 matrica

- Sve matrice koje smo do sada vidjeli imaju [0 0 0 1] u zadnjem redu
- Proizvod koji se dobija množenjem bilo koje dvije matrice ovog oblika također će imati [0 0 0 1] u zadnjem redu
- Može se reći da ovaj skup matrica formira multiplikativnu grupu 3D linearnih transformacija
- Bilo koja matrica u ovoj multiplikativnoj grupi može se konstruirati množenjem sekvence osnovnih rotacija, translacija, skaliranja i smicanja

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Opšta 4x4 matrica

- Kasnije ćemo vidjeti da možemo izmijeniti donji red da bismo obavili projekcije pogleda, ali za sada ćemo razmatrati samo standardne 3D transformacije
- Vidimo da postoji 12 različitih brojeva u gornjem 3x4 segmentu 4x4 matrice
- Postoji također 12 stepeni slobode za jedan objekt podvrgnut linearnoj transformaciji u 3D prostoru
  - 3 stepena slobode su predstavljena pomoću tri translatorne ose
  - 3 stepena slobode su za rotacije u 3 ravni (xy, yz, xz)
  - 3 stepena slobode su skaliranja duž 3 glavne ose
  - a posljednja 3 su smicanja u 3 glavne ravni (xy, yz, xz)

- 3 broja za translaciju se jednostavno dekodiraju (d_x, d_y, d_z)
- Međutim, preostalih 9 brojeva je kodirano u 9 brojeva u gornjem 3x3 segmentu matrice

Afine transformacije

- Sve transformacije koje smo do sada vidjeli su primjeri *afinih* transformacija
- Ako imamo jedan par paralelnih linija i transformiramo ih s afinom transformacijom, one će ostati paralelne
- Afine transformacije su brze za proračun i vrlo korisne u računarskoj grafici

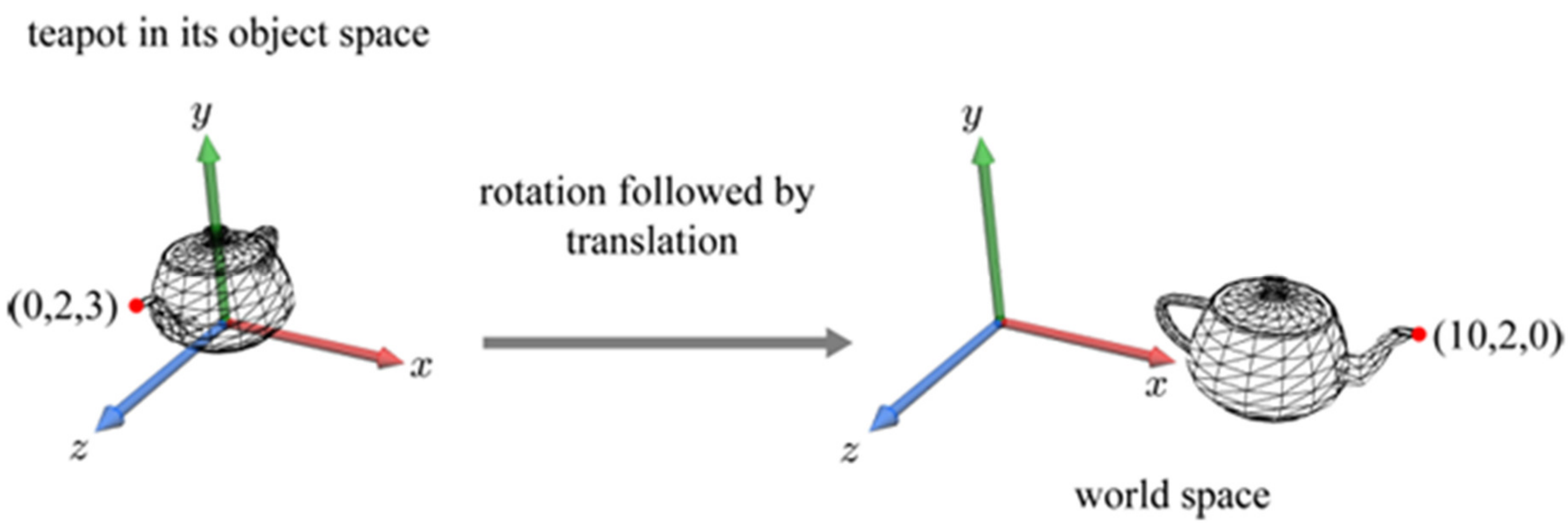
Prostor objekta

- Prostor u kojem je objekt definiran naziva se *prostor objekta* ili *lokalni prostor*
- Obično je objekt lociran u ili u blizini ishodišta koordinatnog sistema i poravnat s xyz osama na nekakav razuman način
- Mjerne jedinice u ovom prostoru mogu biti bilo šta što odaberemo (tj. metri, itd.)
- 3D objekt će biti pohranjen na disk i u memoriju u ovom koordinatnom sistemu
- Kada idemo nacrtati objekt, mi ćemo ga htjeti transformirati u drugačiji prostor

Svjetski prostor

- Definirat ćemo novi prostor pod imenom *svjetski prostor* (engl. *world space*) ili *globalni prostor* (engl. *global space*)
- Ovaj prostor reprezentira 3D svijet ili scenu i može sadržavati nekoliko objekata razmještenih na raznim lokacijama
- Svakom objektu u svijetu je potrebna matrica koja transformira njegove vrhove iz njegovog vlastitog prostora objekta u ovaj svjetski prostor
- Ova matrica se naziva *svjetska matrica* objekta ili često samo matrica objekta
- Naprimjer, ukoliko u nekoj prostoriji imamo 100 stolica, trebamo samo jedanput pohraniti podatke o prostoru objekta za stolicu, a zatim možemo upotrijebiti 100 različitih matrica da bismo transformirali model stolice na 100 lokacija u svjetskom prostoru

Primjer



Transformacija čajnika iz lokalnog prostora u svjetski prostor.

ABCD vektori

- Spomenuli smo da se informacije o translaciji lako ekstrahuju direktno iz matrice, dok su informacije o rotaciji kodirane u gornji 3x3 segment matrice
- Postoji li geometrijski način da se interpretira ovih 9 brojeva?
- Naravno da postoji! 9 konstanti obrazuje 3 vektora pod imenom **a**, **b** i **c**. Ako o matrici razmišljamo kao o transformaciji iz prostora objekta u svjetski prostor, tada je vektor **a** ustvari x-osa objekta zarotirana u svjetski prostor, **b** je njegova y-osa u svjetskom prostoru, a **c** je njegova z-osa u svjetskom prostoru. **d** je, naravno, pozicija u svjetskom prostoru.

Ortonormalnost

- Ako su vektori a, b i c jedinične dužine te ortogonalni jedan na drugog, kažemo da je matrica *ortonormalna*
- Tehnički uzevši, samo je gornji 3x3 segment matrice ortonormalan, tako da ovaj termin nije striktno ispravan ukoliko u matrici postoji translacija
- Bolji termin za 4x4 matrice bi bio nazvati ih *rigidnim* što implicira da one samo rotiraju i transliraju objekt, bez ikakvog nerigidnog skaliranja ili distorzija usljed smicanja

Ortonormalnost

- Ako 4x4 matrica reprezentira rigidnu transformaciju, tada će gornji 3x3 segment matrice biti ortonormalan

$$|\mathbf{a}| = |\mathbf{b}| = |\mathbf{c}| = 1$$

$$\mathbf{a} = \mathbf{b} \times \mathbf{c}$$

$$\mathbf{b} = \mathbf{c} \times \mathbf{a}$$

$$\mathbf{c} = \mathbf{a} \times \mathbf{b}$$

Primjer: Pomak nadesno

- Imamo kruto tijelo (engl. *rigid object*) s matricom **M**. Želimo ga pomaknuti 3 jedinice nadesno

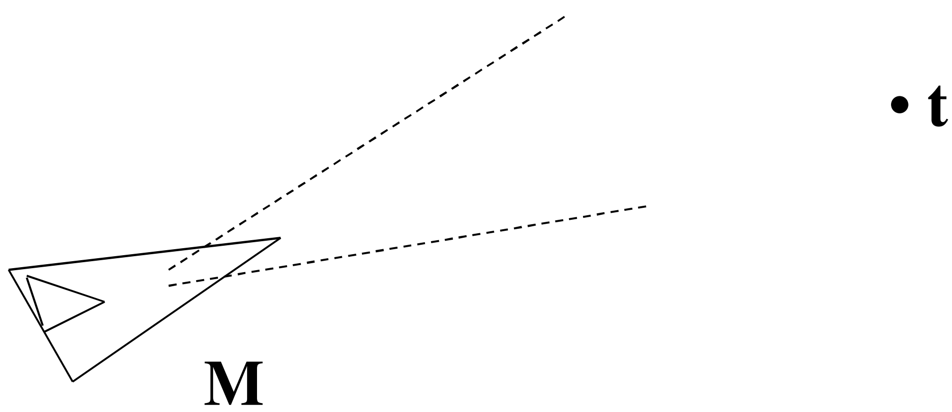
Primjer: Pomak nadesno

- Imamo kruto tijelo (engl. *rigid object*) s matricom **M**. Želimo ga pomaknuti 3 jedinice nadesno

M.**d** = **M**.**d** + 3***M**.**a**

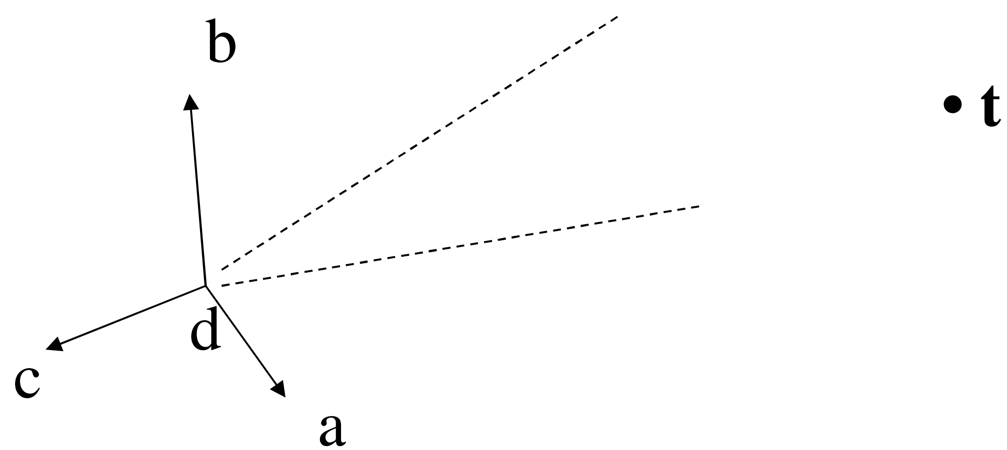
Primjer: Meta "naciljana"

- Da bi avion naciljao projektil, meta mora biti unutar 10 stepeni konusno ispred aviona. Ako je **M** matrica aviona a **t** pozicija mete, napisati izraz koji određuje da li avion može naciljati metu.



Primjer: Meta "naciljana"

- Da bi avion naciljao projektil, meta mora biti unutar 10 stepeni konusno ispred aviona. Ako je **M** matrica aviona a **t** pozicija mete, napisati izraz koji određuje da li avion može naciljati metu.



Primjer: Meta "naciljana"

- Potrebno je ispitati vrijednost ugla između *heading* vektora (-**c**) i vektora od **d** do **t:**

$$da\ li\ je\ \cos^{-1}\left(\frac{(\mathbf{d}-\mathbf{t})\cdot\mathbf{c}}{|\mathbf{d}-\mathbf{t}|}\right)<10°$$

- Postupak se može ubrzati uvrštavajući u nejednakost vrijednost kosinusa (cos(10°) = 0.985)

$$da\ li\ je\ \frac{(\mathbf{d}-\mathbf{t})\cdot\mathbf{c}}{|\mathbf{d}-\mathbf{t}|}>0.985$$

Primjer: Meta "naciljana"

- Postupak se može dalje ubrzati eliminacijom operacija dijeljenja i kvadratnog korijena u proračunu magnitude:

$$da\ li\ je\ \left(\left(\left(\mathbf{d}-\mathbf{t}\right)\cdot\mathbf{c}\right)^{2} > 0.970 * \left|\mathbf{d}-\mathbf{t}\right|^{2}\right)$$

- Sve u svemu, ovaj postupak zahtijeva 8 operacija množenja i 8 operacija sabiranja

Proizvod vektora pozicije i matrice

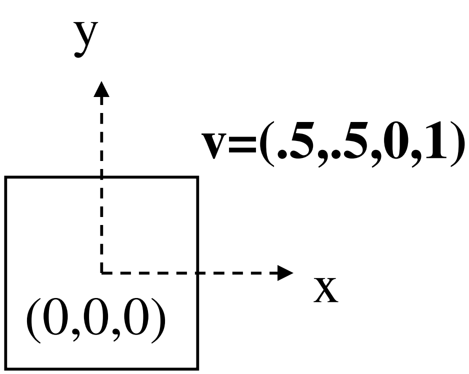
$$\mathbf{v}' = \mathbf{M}\cdot\mathbf{v} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

$$v'_x = v_x a_x + v_y b_x + v_z c_x + d_x$$
$$v'_y = v_x a_y + v_y b_y + v_z c_y + d_y$$
$$v'_z = v_x a_z + v_y b_z + v_z c_z + d_z$$

$$\boxed{\mathbf{v}' = v_x\mathbf{a} + v_y\mathbf{b} + v_z\mathbf{c} + \mathbf{d}}$$

Proizvod vektora pozicije i matrice

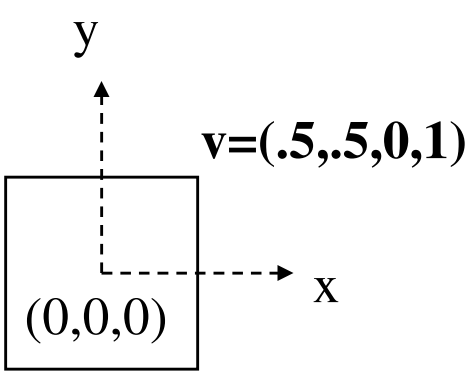
$$\mathbf{v}' = v_x\mathbf{a} + v_y\mathbf{b} + v_z\mathbf{c} + \mathbf{d}$$



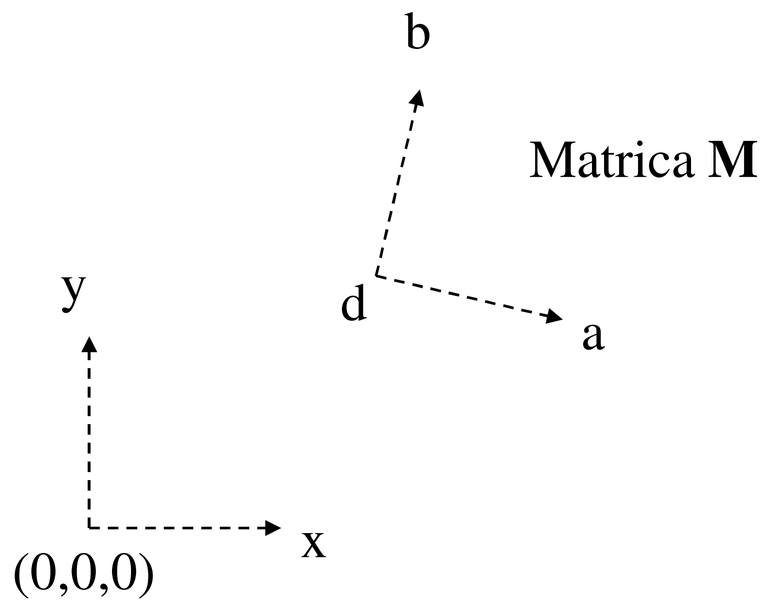
Lokalni prostor

Proizvod vektora pozicije i matrice

$$\mathbf{v}' = v_x\mathbf{a} + v_y\mathbf{b} + v_z\mathbf{c} + \mathbf{d}$$



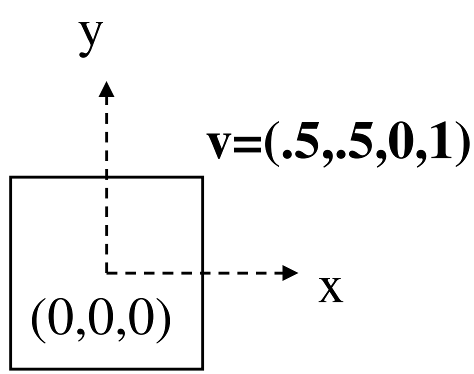
Lokalni prostor



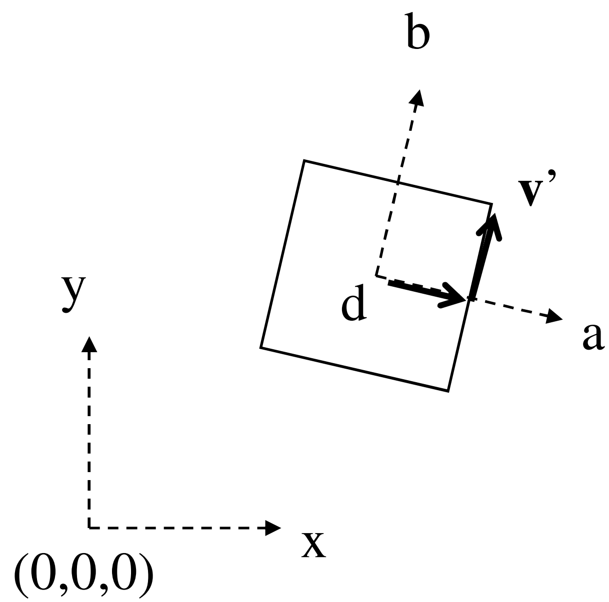
Svjetski prostor

Proizvod vektora pozicije i matrice

$$\mathbf{v}' = v_x\mathbf{a} + v_y\mathbf{b} + v_z\mathbf{c} + \mathbf{d}$$



Lokalni prostor



Svjetski prostor

Vektori pozicija vs. vektori smjera

- Vektori koji predstavljaju poziciju u 3D prostoru proširuju se u 4D kao:

$$\begin{bmatrix} v_x & v_y & v_z & 1 \end{bmatrix}$$

- Vektori koji predstavljaju smjer proširuju se kao:

$$\begin{bmatrix} v_x & v_y & v_z & 0 \end{bmatrix}$$

Proizvod vektora smjera i matrice

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix}$$

$$v'_x = v_x a_x + v_y b_x + v_z c_x$$
$$v'_y = v_x a_y + v_y b_y + v_z c_y$$
$$v'_z = v_x a_z + v_y b_z + v_z c_z$$

$$\mathbf{v}' = v_x \mathbf{a} + v_y \mathbf{b} + v_z \mathbf{c}$$

Proizvod dvije matrice

$$\mathbf{M}' = \mathbf{N} \cdot \mathbf{M} \qquad \mathbf{M} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Vektori **abcd** iz **M'** su vektori **abcd** iz **M** transformirani pomoću matrice **N**
- Primijetimo da se vektori **a**, **b** i **c** transformiraju kao vektori smjera, dok se vektor **d** transformira kao vektor pozicije

Jedinična matrica

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

- Bacimo još jedan pogled na jediničnu matricu
- To je osa **a** poravnata sa x, **b** poravnata sa y, te **c** poravnata sa z
- Pozicija **d** je u ishodištu koordinatnog sistema
- Zbog toga jedinična matrica predstavlja transformaciju bez rotacije ili translacije

Determinante

- Determinanta 4x4 matrice bez projekcije je jednaka determinanti gornjeg 3x3 segmenta ove matrice

$$\begin{vmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{vmatrix} = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$$

Determinante

- Determinanta je skalarna veličina koja predstavlja promjenu volumena koju će prouzrokovati transformacija
- Ortonormalna matrica će imati determinantu jednaku 1, ali će i neortonormalne matrice koje konzerviraju volumen također imati determinantu jednaku 1
- Zaravnjena ili degenerirana matrica ima determinantu jednaku 0
- Matrica koja je "zrcaljena" imaće negativnu determinantu

Inverzna matrica

- Ako **M** transformira **v** u svjetski prostor, onda $\mathbf{M}^{-1}$ transformira **v′** natrag u prostor objekta

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v}$$

$$\mathbf{v} = \mathbf{M}^{-1} \cdot \mathbf{v}'$$

$$\mathbf{M} \cdot \mathbf{M}^{-1} = \mathbf{M}^{-1} \cdot \mathbf{M} = \mathbf{I}$$

Matrica kamere

- Baš kao što se za bilo koji objekt (stolica, stol…) može upotrijebiti matrica da bi se objekt postavio u svjetski prostor, može se imati i virtualna kamera i upotrijebiti matrica da bi se kamera postavila u svjetski prostor
- Matrica kamere u suštini transformira vrhove iz *prostora kamere* u svjetski prostor

Prostor kamere

- Recimo da želimo renderirati sliku stolice iz određene tačke pogleda kamere
- Stolica je postavljena u svjetskom prostoru s matricom **M**
- Kamera je postavljena u svjetskom prostoru s matricom **C**
- Sljedeća transformacija transformira vrhove iz prostora objekta stolice u svjetski prostor, a zatim iz svjetskog prostora u prostor kamere:

$$\mathbf{v}' = \mathbf{C}^{-1} \cdot \mathbf{M} \cdot \mathbf{v}$$

- Da bismo stolicu doista renderirali, moramo također projektovati tačku pogleda kamere u 2D prostor. O tome ćemo govoriti na narednom predavanju...