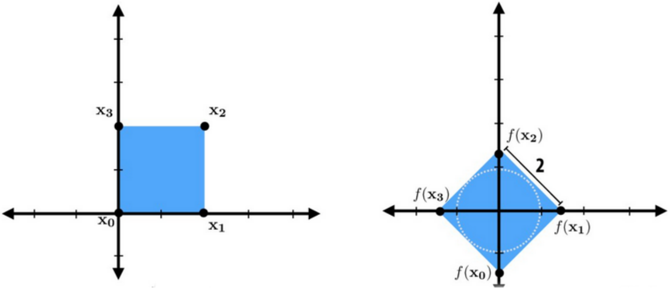
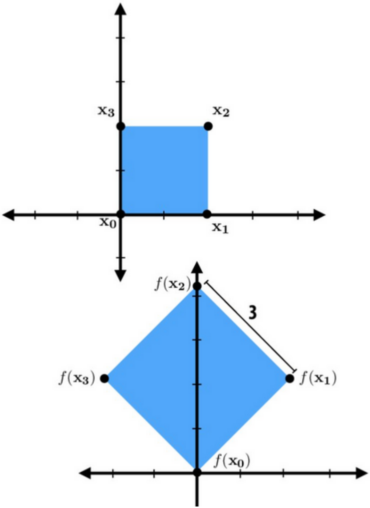
Računarska grafika
Geometrijske transformacije u 3D

Vesna Marinković

Kojom transformacijom se prva figura slika u drugu?



Kojom transformacijom se prva figura slika u drugu?

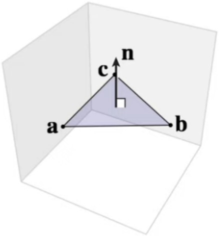


Dimenzija++ (homogene koordinate za 3D prostor)

- Tačka 3D prostora predstavlja se četvorkom koordinata (x, y, z, W)
- Dve četvorke predstavljaju istu tačku ako i samo ako je jedna četvorka umnožak druge
- Četvorka $(0, 0, 0, 0)$ nije dozvoljena
- Standardna reprezentacija tačke (x, y, z, W), $W \neq 0$ je $(x/W, y/W, z/W, 1)$
- Skupu svih četvorki kojima odgovara jedna homogena tačka odgovara prava u 4-dimenzionom prostoru: sve tačke oblika (tx, ty, tz, tW), $t \neq 0$ pripadaju jednoj pravoj 4D prostora
- Transformacije u homogenim koordinatama su predstavljene matricama dimenzije 4×4

Homogene koordinate u 3D prostoru

- Korišćenjem homogenih koordinata možemo napraviti razliku između tačaka i vektora
- Razmotrimo $\triangle abc$ sa temenima $a, b, c \in \mathbb{R}^3$ i njegov jedinični vektor normale $n \in \mathbb{R}^3$



Homogene koordinate u 3D prostoru

- Na trougao želimo da primenimo transformaciju čija je matrica

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

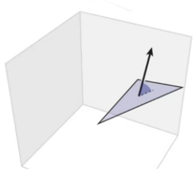
- Vektorima $a, b, c, n \in \mathbb{R}^3$ dodajemo 1 kao homogenu koordinatu i množimo vektor koordinata prethodnom matricom

Homogene koordinate u 3D prostoru

- Na trougao želimo da primenimo transformaciju čija je matrica

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

- Vektorima $a, b, c, n \in \mathbb{R}^3$ dodajemo 1 kao homogenu koordinatu i množimo vektor koordinata prethodnom matricom

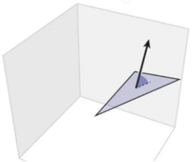


Homogene koordinate u 3D prostoru

- Na trougao želimo da primenimo transformaciju čija je matrica

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

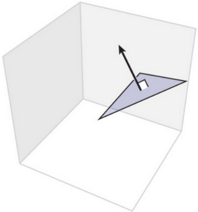
- Vektorima $a, b, c, n \in \mathbb{R}^3$ dodajemo 1 kao homogenu koordinatu i množimo vektor koordinata prethodnom matricom



- Dobijamo da normala trougla više nije upravna na trougao!
- Šta je pošlo po zlu?

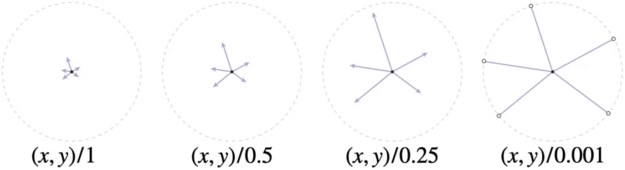
Homogene koordinate u 3D prostoru

- Kada vektor normale pomnožimo prethodnom matricom, on se rotira i translira, međutim vektor normale ne treba da se translira
- Ovo postižemo postavljanjem homogene koordinate vektora n na 0



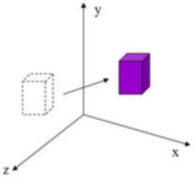
Tačke vs. vektori u homogenim koordinatama

- U opštem slučaju:
  - tačke imaju nenula homogenu koordinatu ($W \neq 0$)
  - vektori imaju homogenu koordinatu jednaku 0 ($W = 0$)
- Kako deliti homogenom koordinatom ako je jednaka 0?



- Na vektore možemo gledati kao na tačke u beskonačnosti

Translacija u 3D prostoru

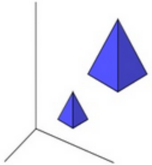


- Matrica translacije:

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

- Kompozicija dve translacije je translacija
- Transformacija inverzna translaciji je translacija

Skaliranje u 3D prostoru

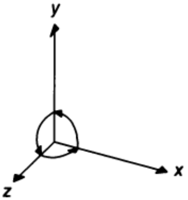


- Matrica skaliranja u odnosu na koordinatni početak:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Kompozicija dva skaliranja u odnosu na koordinatni početak je skaliranje u odnosu na koordinatni početak
- Transformacija inverzna skaliranju je skaliranje

Pozitivno orijentisani koordinatni sistem



- Razmatraćemo pozitivno orijentisani koordinatni sistem
- Uvek pišemo koordinate u istom (cikličkom) poretku x, y, z
    - Ako razmatramo rotaciju oko x ose, rotiramo $+y$ ka $+z$
    - Ako razmatramo rotaciju oko y ose, rotiramo $+z$ ka $+x$
    - Ako razmatramo rotaciju oko z ose, rotiramo $+x$ ka $+y$

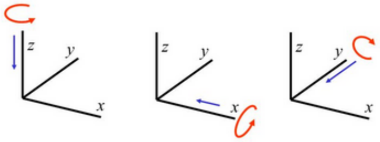
Rotacija u 3D prostoru

- Kako prepoznati rotaciju?

Rotacija u 3D prostoru

- Kako prepoznati rotaciju?
- Rotacija čuva:
  - dužine duži
  - koordinatni početak
  - orijentaciju
- U 2D prostoru postoji samo rotacija oko proizvoljne tačke, osa rotacije je uvek ista
- U 3D prostoru se objekat može rotirati oko proizvoljnog vektora – osa rotacije

Rotacije oko koordinatnih osa



- Matrica rotacije oko z ose: $\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$
- Matrica rotacije oko x ose: $\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$
- Matrica rotacije oko y ose: $\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Rotacija u 3D prostoru - stepeni slobode

- Sa koliko brojeva je određena rotacija u 3D?
- Možemo koristiti rotacije oko osa, ali da li nam trebaju sve tri?

- Da bismo rotirali jedan grad u drugi (npr. Kairo u Pariz) potrebno je zadati dva broja: geografsku dužinu i širinu
- Da li je ovo jedina rotacija kojom se ova dva grada poklapaju?

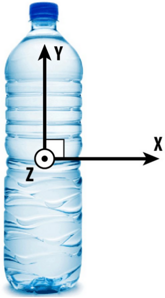
Rotacija u 3D prostoru - stepeni slobode

- Sa koliko brojeva je određena rotacija u 3D?
- Možemo koristiti rotacije oko osa, ali da li nam trebaju sve tri?

- Da bismo rotirali jedan grad u drugi (npr. Kairo u Pariz) potrebno je zadati dva broja: geografsku dužinu i širinu
- Da li je ovo jedina rotacija kojom se ova dva grada poklapaju?
- Nije – možemo ostaviti Pariz fiksnim dok rotiramo globus
- Potrebna su sva tri stepena slobode

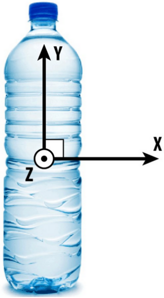
Komutativnost rotacija u 3D prostoru

- U 2D rotacije komutiraju, a u 3D?
- Uporedimo rezultat narednih nizova rotacija
  - rotacija za 90° oko y ose, za 90° oko z ose i za 90° oko x ose
  - rotacija za 90° oko z ose, za 90° oko y ose i za 90° oko x ose



Komutativnost rotacija u 3D prostoru

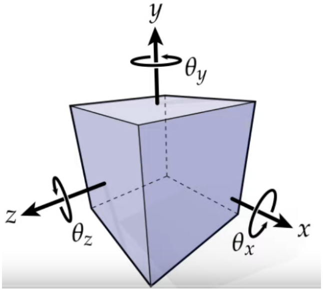
- U 2D rotacije komutiraju, a u 3D?
- Uporedimo rezultat narednih nizova rotacija
  - rotacija za 90° oko y ose, za 90° oko z ose i za 90° oko x ose
  - rotacija za 90° oko z ose, za 90° oko y ose i za 90° oko x ose



- Veoma je važan redosled kojim primenjujemo rotacije!

Predstavljanje rotacije u 3D prostoru - Ojlerovi uglovi

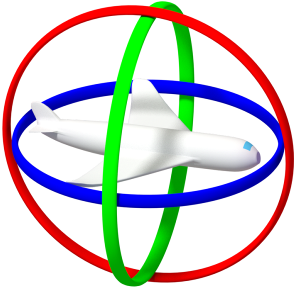
- Kako predstaviti proizvoljnu rotaciju u 3D?
- Rotacija se može predstaviti kao kompozicija tri rotacije:
  - oko x ose u Oyz ravni za ugao θ_x – pitch
  - oko y ose u Oxz ravni za ugao θ_y – yaw
  - oko z ose u Oxy ravni za ugao θ_z – roll



- Ova shema naziva se Ojlerovim uglovima

Prednosti i nedostaci korišćenja Ojlerovih uglova

- Prednost: koncept je veoma lak za razumevanje
- Nedostatak: Gimbal lock



- Kada koristimo Ojlerove uglove θ_x, θ_y i θ_z možemo stići u konfiguraciju kada ne možemo vršiti rotaciju oko jedne od tri koordinatnih osa

Računanje Ojlerovih uglova

- Podsetimo se matrica rotacija oko koordinatnih osa

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \quad R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \quad R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Proizvod ovih matrica predstavlja rotaciju preko Ojlerovih uglova

$$R = R_x R_y R_z = \begin{bmatrix} \cos\theta_y \cos\theta_z & -\cos\theta_y \sin\theta_z & \sin\theta_y \\ \cos\theta_z \sin\theta_x \sin\theta_y + \cos\theta_x \sin\theta_z & \cos\theta_x \cos\theta_z - \sin\theta_x \sin\theta_y \sin\theta_z & -\cos\theta_y \sin\theta_x \\ -\cos\theta_x \cos\theta_z \sin\theta_y + \sin\theta_x \sin\theta_z & \cos\theta_z \sin\theta_x + \cos\theta_x \sin\theta_y \sin\theta_z & \cos\theta_x \cos\theta_y \end{bmatrix}$$

Računanje Ojlerovih uglova

- Podsetimo se matrica rotacija oko koordinatnih osa

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \quad R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \quad R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Proizvod ovih matrica predstavlja rotaciju preko Ojlerovih uglova

$$R = R_x R_y R_z = \begin{bmatrix} \cos\theta_y \cos\theta_z & -\cos\theta_y \sin\theta_z & \sin\theta_y \\ \cos\theta_z \sin\theta_x \sin\theta_y + \cos\theta_x \sin\theta_z & \cos\theta_x \cos\theta_z - \sin\theta_x \sin\theta_y \sin\theta_z & -\cos\theta_y \sin\theta_x \\ -\cos\theta_x \cos\theta_z \sin\theta_y + \sin\theta_x \sin\theta_z & \cos\theta_z \sin\theta_x + \cos\theta_x \sin\theta_y \sin\theta_z & \cos\theta_x \cos\theta_y \end{bmatrix}$$

- Iz matrice možemo izračunati vrednosti Ojlerovih uglova:

$$\begin{aligned} \theta_y &= \arcsin(R_{02}) \\ \theta_z &= -\arctan\frac{R_{01}}{R_{00}} \\ \theta_x &= -\arctan\frac{R_{12}}{R_{22}} \end{aligned}$$

Gimbal lock

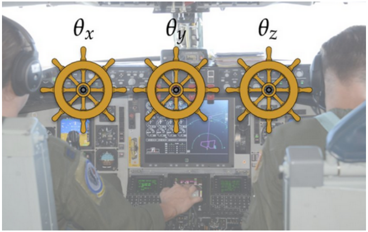
- Specijalan slučaj: $\theta_y = \frac{\pi}{2}$
  tada važi: $\cos\theta_y = 0, \sin\theta_y = 1$
- Matrica rotacije R se pojednostavljuje:

$$R = \begin{bmatrix} 0 & 0 & 1 \\ \cos\theta_z \sin\theta_x + \cos\theta_x \sin\theta_z & \cos\theta_x \cos\theta_z - \sin\theta_x \sin\theta_z & 0 \\ -\cos\theta_x \cos\theta_z + \sin\theta_x \sin\theta_z & \cos\theta_z \sin\theta_x + \cos\theta_x \sin\theta_z & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ \sin(\theta_x + \theta_z) & \cos(\theta_x + \theta_z) & 0 \\ -\cos(\theta_x + \theta_z) & \sin(\theta_x + \theta_z) & 0 \end{bmatrix}$$

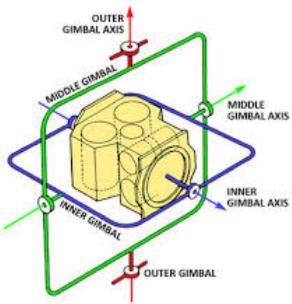
Gimbal lock (2)

- Kako god podesili uglove θ_x i θ_z, možemo vršiti rotaciju samo u jednoj ravni
- Problem je u matematičkoj parametrizaciji rotacije
- Gubi se jedan stepen slobode u mehanizmu sa tri osovine
- Loš dizajn za kontrole aviona



Gimbal lock (3)

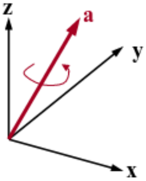
- Apolo 13 je sadržao inercionu mernu jedinicu (IMU) – uređaj koji meri orijentaciju kosmičkog broda
- Računar je dizajniran da izda upozorenje kada se ose gotovo izravnaju
- Mike Collins: "How about sending me a fourth gimbal for Christmas?"



https://www.youtube.com/watch?v=OmCzZ-D8Wdk

Rotacija za datu osu i ugao rotacije

- Neka je $a = [a_x \ a_y \ a_z]^T$ osa rotacije, a φ ugao za koji vršimo rotaciju

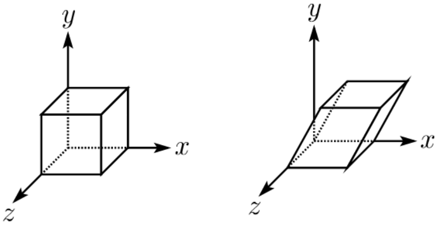


- Rodrigezova formula:

$$\begin{bmatrix} \cos\varphi + a_x^2(1-\cos\varphi) & a_x a_y(1-\cos\varphi) - a_z\sin\varphi & a_x a_z(1-\cos\varphi) + a_y\sin\varphi \\ a_x a_y(1-\cos\varphi) + a_z\sin\varphi & \cos\varphi + a_y^2(1-\cos\varphi) & a_y a_z(1-\cos\varphi) - a_x\sin\varphi \\ a_x a_z(1-\cos\varphi) - a_y\sin\varphi & a_y a_z(1-\cos\varphi) + a_x\sin\varphi & \cos\varphi + a_z^2(1-\cos\varphi) \end{bmatrix}$$

- Alternativa: kvaternioni

Smicanje u 3D prostoru



- U 3D prostoru smicanje najčešće podrazumeva iskošenje objekta u smeru paralelnom nekoj koordinatnoj ravni (duž neke koordinatne ose)
- Matrica smicanja: $\begin{bmatrix} 1 & sh_{xy} & sh_{xz} & 0 \\ sh_{yx} & 1 & sh_{yz} & 0 \\ sh_{zx} & sh_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Smicanje u 3D prostoru

- Razmotrimo smicanje jedinične kocke u pravcu koordinatne ravni Oxz: strana kocke koja leži u Oxz ravni se ne menja, dok se strana koja leži u ravni $y = 1$ translira za vektor (a, b)
- Koordinatni početak i jedinični vektori u smeru x i z koordinatne ose se ne menjaju
- Matrica smicanja u pravcu Oxz ravni (duž y ose):

$$\begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Korisne su za generisanje projekcija (na primer, kako nacrtati 3D kocku na 2D ekranu)

Kompozicije transformacija u 3D prostoru

- Svakoj kompoziciji rotacija, skaliranja, translacija i smicanja u 3D prostoru odgovara matrica oblika:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R^* & t^* \\ \mathbf{0} & 1 \end{bmatrix}$$

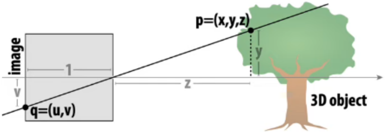
  gde matrica R^* opisuje kombinovani efekat rotacija, skaliranja i smicanja, a t^* vektor koji opisuje kombinovani efekat svih translacija

- Umesto množenja matricom 4×4, može se koristiti efikasnije izračunavanje:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R^* \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t^*$$

Perspektivna projekcija u terminima homogenih koordinata

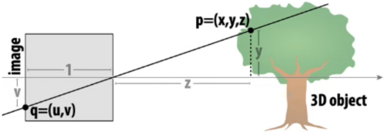
- Setimo se modela tačkaste kamere i ideje da je jedino potrebno podeliti x i y koordinatu sa z: $(x, y, z) \to (x/z, y/z)$



- Kako izvesti matricu perspektivne projekcije?

Perspektivna projekcija u terminima homogenih koordinata

- Setimo se modela tačkaste kamere i ideje da je jedino potrebno podeliti x i y koordinatu sa z: $(x, y, z) \to (x/z, y/z)$



- Kako izvesti matricu perspektivne projekcije?
- Razmotrimo matricu koja kopira z vrednost u homogenu koordinatu

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

- Deljenjem x, y i z koordinate homogenom koordinatom dobijamo perspektivnu projekciju na ravan $z = 1$

Afina preslikavanja tačaka, pravih i ravni

- Matrice transformacija određuju slike pojedinačnih tačaka
- Kako dobiti sliku prave pri afinoj transformaciji?

Afina preslikavanja tačaka, pravih i ravni

- Matrice transformacija određuju slike pojedinačnih tačaka
- Kako dobiti sliku prave pri afinoj transformaciji?
  - afine transformacije slikaju prave u prave
  - sliku prave možemo dobiti kao pravu određenu slikama dve tačke polazne prave

Afina preslikavanja tačaka, pravih i ravni

- Matrice transformacija određuju slike pojedinačnih tačaka
- Kako dobiti sliku prave pri afinoj transformaciji?
  - afine transformacije slikaju prave u prave
  - sliku prave možemo dobiti kao pravu određenu slikama dve tačke polazne prave
- Kako dobiti sliku ravni pri afinoj transformaciji?

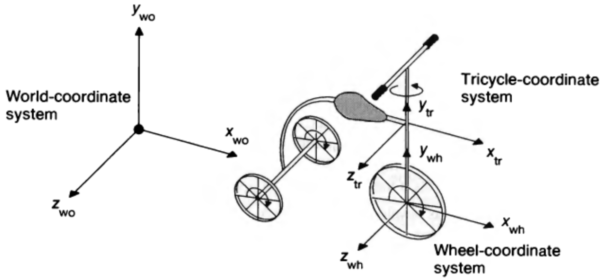
Afina preslikavanja tačaka, pravih i ravni

- Matrice transformacija određuju slike pojedinačnih tačaka
- Kako dobiti sliku prave pri afinoj transformaciji?
  - afine transformacije slikaju prave u prave
  - sliku prave možemo dobiti kao pravu određenu slikama dve tačke polazne prave
- Kako dobiti sliku ravni pri afinoj transformaciji?
  - afine transformacije slikaju ravan u ravan
  - sliku ravni možemo dobiti:
    - kao ravan određenu slikama tri nekolinearne tačke polazne ravni
    - kao ravan određenu slikom jedne tačke i slikom vektora normale polazne ravni

Dva različita pogleda na transformacije

- Do sada su transformacije korišćene za preslikavanje skupa tačaka jednog koordinatnog sistema u isti taj koordinatni sistem
  – objekat je transformisan, a koordinatni sistem je ostajao isti
- Transformaciju možemo posmatrati i kao promenu koordinatnog sistema
  – objekat se ne transformiše, već se računaju njegove koordinate u novom koordinatnom sistemu
- Prvi pristup je pogodniji kada se objekat kreće i mi pratimo njegove koordinate; drugi kada više objekata u svojim pojedinačnim koordinatnim sistemima treba objediniti u jedinstven koordinatni sistem

Primer promene koordinatnog sistema



- Tricikl je opisan u svom koordinatnom sistemu
- Želimo da u svetskom koordinatnom sistemu odredimo koordinate proizvoljne tačke na prednjem točku dok se točak okreće
- Biramo pogodniji koordinatni sistem – onaj u kojem je centar točka koordinatni početak, a osa točka jednaka z osi i u njemu računamo nove koordinate tačke sa točka

Transformacije kao promena koordinatnog sistema

- $M_{i \leftarrow j}$ – matrica transformacije koordinatnog sistema j u koordinatni sistem i
- $P^{(i)}$ – koordinate tačke P u koordinatnom sistemu i, $P^{(j)}$ u koordinatnom sistemu j i $P^{(k)}$ u koordinatnom sistemu k
- Važi:

$$P^{(i)} = M_{i \leftarrow j} P^{(j)} = M_{i \leftarrow j} M_{j \leftarrow k} P^{(k)} = M_{i \leftarrow k} P^{(k)}$$

  i

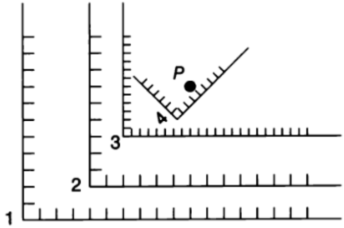
$$M_{i \leftarrow j} M_{j \leftarrow k} = M_{i \leftarrow k}$$

  kao i

$$M_{j \leftarrow i} = M_{i \leftarrow j}^{-1}$$

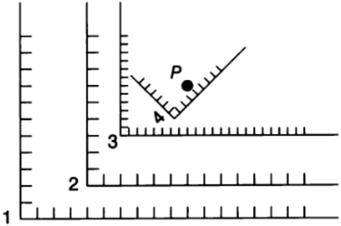
- Iz poslednje dve jednakosti sledi važno svojstvo: za proizvoljnu promenu koordinatnog sistema dovoljno je imati opis svođenja iz različitih koordinatnih sistema na jedan koordinatni sistem

Transformacije kao promena koordinatnog sistema



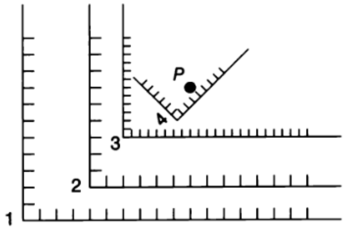
- Data su četiri različita koordinatna sistema označena brojevima $1, 2, 3$ i 4
- Čemu je jednaka matrica $M_{1 \leftarrow 2}$, čemu matrica $M_{2 \leftarrow 3}$, čemu matrica $M_{3 \leftarrow 4}$?

Transformacije kao promena koordinatnog sistema



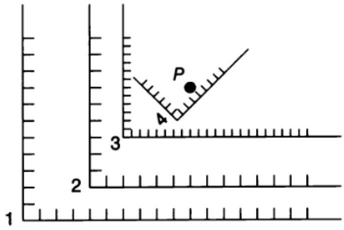
- $M_{1 \leftarrow 2} = ?$ $M_{2 \leftarrow 3} = ?$ $M_{3 \leftarrow 4} = ?$

Transformacije kao promena koordinatnog sistema



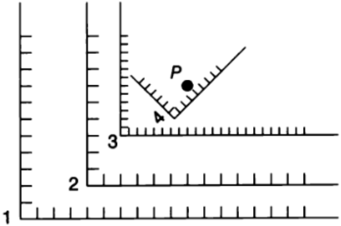
- $M_{1\leftarrow 2} =?$ $M_{2\leftarrow 3} =?$ $M_{3\leftarrow 4} =?$
- $M_{1\leftarrow 2} = T_{4,2}$

Transformacije kao promena koordinatnog sistema



- $M_{1\leftarrow 2} =?$ $M_{2\leftarrow 3} =?$ $M_{3\leftarrow 4} =?$
- $M_{1\leftarrow 2} = T_{4,2}$
- $M_{2\leftarrow 3} = T_{2,3} \cdot S_{0.5,0.5}$

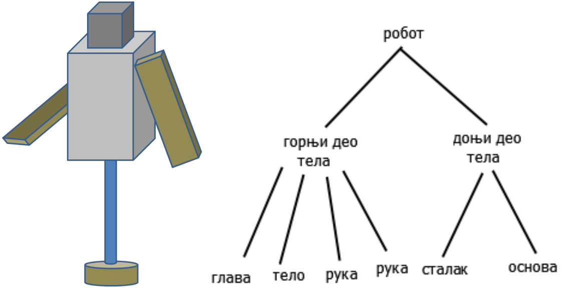
Transformacije kao promena koordinatnog sistema



- $M_{1\leftarrow 2} =?$ $M_{2\leftarrow 3} =?$ $M_{3\leftarrow 4} =?$
- $M_{1\leftarrow 2} = T_{4,2}$
- $M_{2\leftarrow 3} = T_{2,3} \cdot S_{0.5,0.5}$
- $M_{3\leftarrow 4} = T_{6.7,1.8} \cdot R_{\pi/4}$
- Iz prethodnog važi: $M_{1\leftarrow 3} = T_{4,2} \cdot T_{2,3} \cdot S_{0.5,0.5} = T_{6,5} \cdot S_{0.5,0.5}$
- Matrice $M_{i\leftarrow j}$ imaju isti oblik kao i matrice transformacija

Motivacija za uvođenje grafa scene

- Scene su najčešće jako složene i sadrže veliki broj složenih objekata
- Pojedinačni objekti najčešće imaju hijerarhijsku strukturu
- I sama scena se može sagledati hijerarhijski
- Primer: izgraditi robota
- Nije jednostavno direktno zadati transformacije; lakše je transformacije zadati relativno u odnosu na druge delove robota



Graf scene

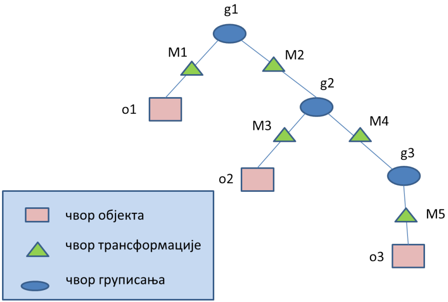
- Za predstavljanje scene potrebno je imati neku robusnu i proširivu strukturu podataka
- Grafičke aplikacije često predstavljaju složene scene u vidu grafa scene
- Graf scene je struktura podataka koja se koristi za predstavljanje hijerarhijskog odnosa među transformacijama koje se primenjuju na skup objekata u trodimenzionalnom prostoru

Format grafa scene

- Graf scene sadrži:
  - čvorove objekata (kocke, valjci, lopte, ...) koji su podrazumevano jedinične veličine i pozicionirani u koordinatnom početku
  - čvorove atributa (boja, tekstura, ...)
  - čvorove transformacija, ...
- Pritom se transformacije mogu primeniti na:
  - listove (glava, osnova, ...)
  - grupe objekata (gornji i donji deo tela)
- Pogodan je za predstavljanje hijerarhijske strukture objekta na osnovu pozicija i atributa njegovih sastavnih delova, ali i za animiranje složenog objekta koji se sastoji iz većeg broja delova

Primer grafa scene

- Najjednostavniji oblik grafa scene je usmereno stablo (grane su usmerene od roditelja ka detetu)
- Kumulativna matrica transformacije M se gradi kako se penjemo uz stablo – transformacije na višem nivou se dodaju na početak niza



- Primer: za objekat o_1 važi $M = M1$, za objekat o_2 važi $M2 \cdot M3$, za objekat o_3 važi $M2 \cdot M4 \cdot M5$

Instanciranje

- Pretpostavimo da treba nacrtati scenu sa velikim brojem kopija objekata
- Umesto da za svaki novi objekat koji je potreban pravimo novu primitivu, možemo ponovo iskoristiti već konstruisane objekte – instanciranje

Definisanje grupa u grafu scene

- Ako koristimo instanciranje, graf scene prestaje da bude stablo, ali će biti usmereni aciklički graf (eng. DAG)
- Transformacije definisane u okviru grupe $g3$ se ne menjaju, ali su različite kumulativne matrice transformacije za svako korišćenje grupe $g3$ kao celine: $T_0 \cdot T_1$ vs $T_0 \cdot T_2 \cdot T_4$