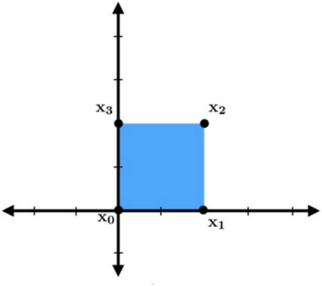
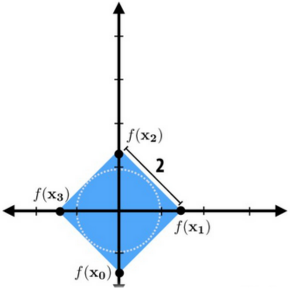
Računarska grafika
Geometrijske transformacije u 3D

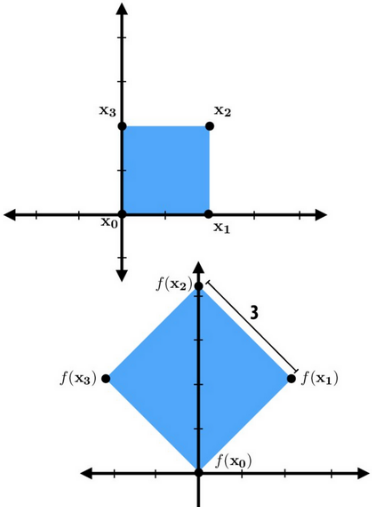
Vesna Marinković

Kojom transformacijom se prva figura slika u drugu?





Kojom transformacijom se prva figura slika u drugu?

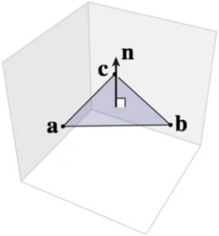


Dimenzija++ (homogene koordinate za 3D prostor)

- Tačka 3D prostora predstavlja se četvorkom vrednosti (x, y, z, W)
- Bar jedna od vrednosti x, y, z i W nije jednaka 0
- Dve četvorke predstavljaju istu tačku 3D prostora ako i samo ako je jedna četvorka umnožak druge
- 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
- 3D 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

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

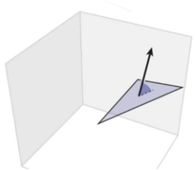
- Vektorima $a, b, c, n \in \mathbb{R}^3$ dodajemo 1 kao homogenu koordinatu i množimo vektor koordinata prethodnom matricom

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

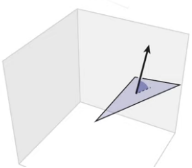


Homogene koordinate u 3D prostoru

- Na trougao želimo da primenimo transformaciju čija je matrica

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

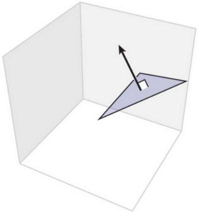
- Vektorima $a, b, c, n \in \mathbb{R}^3$ dodajemo 1 kao homogenu koordinatu i množimo vektor koordinata prethodnom matricom



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

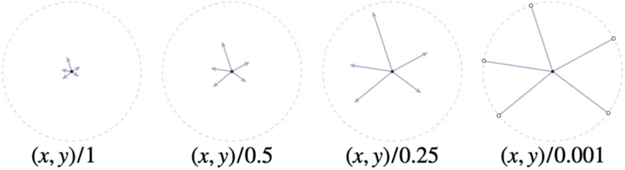
Homogene koordinate u 3D prostoru

- Kada vektor normale trougla 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



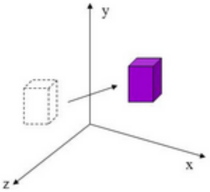
Tačke vs. vektori u homogenim koordinatama

- U opštem slučaju:
  - tačke imaju homogenu koordinatu različitu od 0 ($W \neq 0$)
  - 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

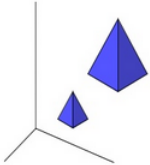


- Matrica translacije:

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

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

Skaliranje u 3D prostoru

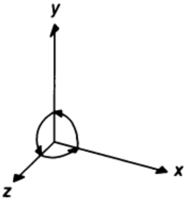


- Matrica skaliranja u odnosu na koordinatni početak:

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

- 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 $+y$

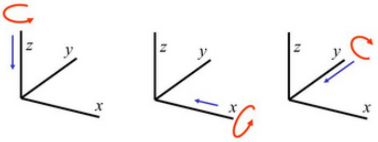
Rotacija u 3D prostoru

- 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 parametara je određena proizvoljna rotacija u 3D?
- Možemo koristiti rotacije oko koordinatnih osa, ali da li nam trebaju sve tri?

- Kako zarotirati jedan grad u drugi (npr. Kairo u Pariz)?

Rotacija u 3D prostoru - stepeni slobode

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

- Kako zarotirati 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?

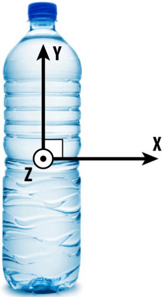
Rotacija u 3D prostoru - stepeni slobode

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

- Kako zarotirati 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

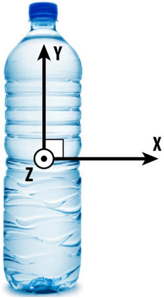
Komutativnost rotacija u 3D prostoru

- 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

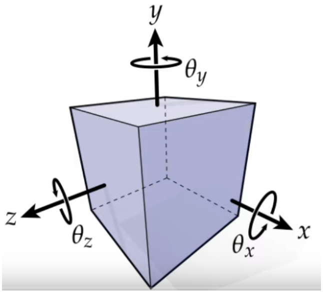
- U 2D rotacije komutiraju, a u 3D?
- Uporedimo rezultat narednih nizova rotacija
  - rotacija za 90° oko y ose, za 90° oko z ose i za 90° oko x ose
  - rotacija za 90° oko z ose, za 90° oko y ose i za 90° oko x ose



- Veoma je važan redosled kojim primenjujemo rotacije!

Predstavljanje rotacije u 3D prostoru - Ojlerovi uglovi

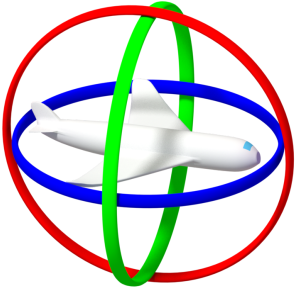
- Kako predstaviti proizvoljnu rotaciju u 3D?
- Rotacija se može predstaviti kao kompozicija tri rotacije:
  - oko x ose u Oyz ravni za ugao θ_x – pitch
  - oko y ose u Oxz ravni za ugao θ_y – yaw
  - oko z ose u Oxy ravni za ugao θ_z – roll



- 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:

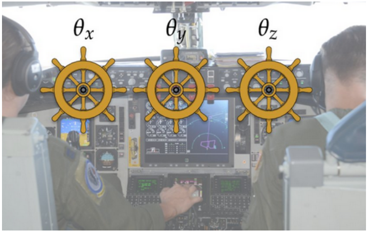
$$R = \begin{bmatrix} 0 & 0 & 1 \\ \cos\theta_z \sin\theta_x + \cos\theta_x \sin\theta_z & \cos\theta_x \cos\theta_z - \sin\theta_x \sin\theta_z & 0 \\ -\cos\theta_x \cos\theta_z + \sin\theta_x \sin\theta_z & \cos\theta_z \sin\theta_x + \cos\theta_x \sin\theta_z & 0 \end{bmatrix}$$

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

- Koja transformacije je zadata ovom matricom? Gde se javlja problem?

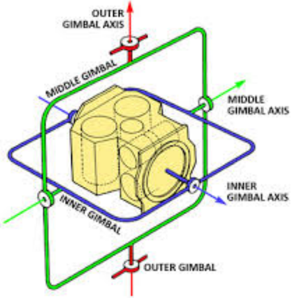
Gimbal lock (2)

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



Gimbal lock (3)

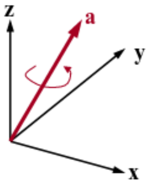
- Apolo 13 je sadržao uređaj koji meri orijentaciju kosmičkog broda
- Računar je dizajniran da izda upozorenje kada se ose gotovo izravnaju
- Mike Collins: "How about sending me a fourth gimbal for Christmas?"



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

Intuitivno definisanje imaginarne jedinice

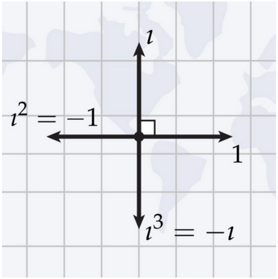
- Uobičajeno se uvodi naredna definicija: $i := \sqrt{-1}$

Intuitivno definisanje imaginarne jedinice

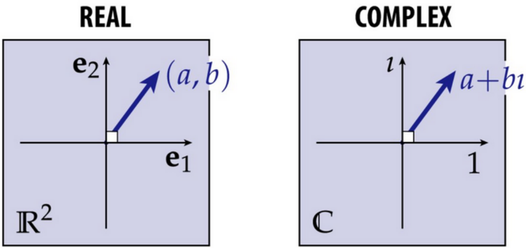
- Uobičajeno se uvodi naredna definicija: $i := \sqrt{-1}$
- Ne sviđa nam se jer se iz nje ne stiče geometrijsko značenje

Intuitivno definisanje imaginarne jedinice

- Uobičajeno se uvodi naredna definicija: $i := \sqrt{-1}$
- Ne sviđa nam se jer se iz nje ne stiče geometrijsko značenje
- Imaginarna jedinica je zaokret za $\pi/2$ u smeru suprotnom od kretanja kazaljke na časovniku
- Množenje vektora sa i ga rotira za $\pi/2$

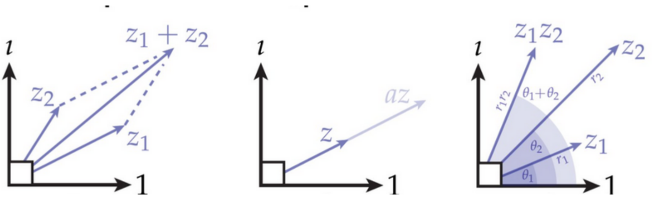


Kompleksni brojevi



- Umesto e_1 i e_2 koristimo 1 i i kao bazne vektore
- Umesto uređenih parova zapisujemo ih kao sumu
- Ponaša se kao 2D prostor
- Definišemo novu vrstu proizvoda dva vektora

Množenje kompleksnih brojeva – geometrijska interpretacija



- Množenjem kompleksnih brojeva se:
  - uglovi sabiraju
  - dužine množe
- Ako su vektori zadati polarnim koordinatama, za $z_1 = (r_1, \theta_1)$ i $z_2 = (r_2, \theta_2)$ važi $z_1 z_2 = (r_1 r_2, \theta_1 + \theta_2)$

Množenje kompleksnih brojeva – algebarska interpretacija

- Razmotrimo slučaj kada su vektori zadati u terminima dekartovskih koordinata $(1, i)$:

$$z_1 = a + bi$$

$$z_2 = c + di$$

$$z_1 z_2 = ac + adi + bci + bdi^2 = (ac - bd) + (ad + bc)i$$

- $(ac - bd)$ se naziva realnim delom, a $(ad + bc)$ imaginarnim delom kompleksnog broja
- Važno je proveriti da li se ovo poklapa sa geometrijskim tumačenjem proizvoda

Množenje kompleksnih brojeva – polarna forma

- Ojlerova formula: $e^{i\phi} = \cos\phi + i\sin\phi$
- Može se iskoristiti za definisanje množenja kompleksnih brojeva

$$z_1 = ae^{i\phi}$$

$$z_2 = be^{i\theta}$$

$$z_1 z_2 = abe^{i(\phi+\theta)}$$

- Rotacija vektora u za ugao θ se može formulisati u polarnim koordinatama preko jediničnog kompleksnog broja a:

$$u = re^{i\alpha}$$

$$a = e^{i\theta}$$

$$au = re^{i(\alpha+\theta)}$$

- Veoma jednostavna interpretacija

Rotacije u 3D

- Kako najjednostavnije formulisati rotacije u 3D?

Kvaternioni

- Autor: William Hamilton
- Nalik kompleksnim brojevima ali za 3D rotacije
- Pokazuje se da nije moguće izvršiti 3D rotaciju korišćenjem 3 koordinate, već je neophodno imati četiri koordinate

$$H = \{a + bi + cj + dk \mid a, b, c, d \in R\}$$

- Proizvod kvaterniona određen je sa:

$$i^2 = j^2 = k^2 = -1, \; ij = k = -ji$$

- Množenje kvaterniona nije komutativno: $pq \neq qp$

Proizvod kvaterniona

- Data su dva kvaterniona

$$q = a_1 + b_1 i + c_1 j + d_1 k$$

$$p = a_2 + b_2 i + c_2 j + d_2 k$$

- Njihov proizvod jednak je:

$$qp = a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 + (a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2) i$$

$$(a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2) j + (a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2) k$$

- Nadamo se da postoji jednostavniji izraz...

Kvaternioni – alternativni zapis

- Realni deo je skalar
- Tri imaginarna dela broja kodiraju vektor u 3D prostoru
- Na kvaternion možemo gledati kao na uređen par skalara i 3D vektora
- Proizvod kvaterniona ima jednostavniji zapis:

$$(a, u)(b, v) = (ab - u \cdot v, av + bu + u \times v)$$

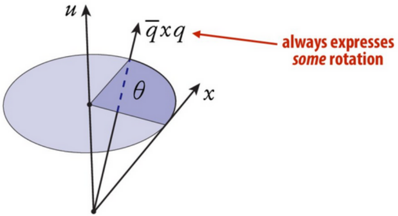
- Za čisto imaginarne brojeve dobijamo još jednostavnije:

$$uv = u \times v - u \cdot v$$

- Proizvod kvaterniona objedinjuje skalarni i vektorski proizvod

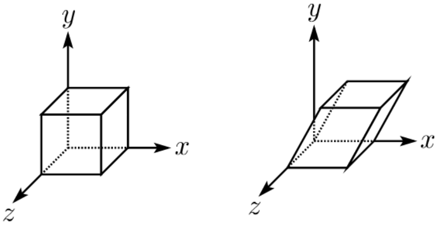
3D rotacije preko kvaterniona

- Glavna uloga kvaterniona u grafici je za definisanje rotacija
- Razmotrimo čisto imaginarni kvaternion x (vektor) i jedinični kvaternion q: $|q|^2 = 1$



- Rotaciju vektora x oko ose u za ugao θ možemo predstaviti putem množenja kvaterniona: $\bar{q}xq$, gde je $q = \cos(\theta/2) + \sin(\theta/2)u$
- Pogodni su za glatku interpolaciju između dve rotacije

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
- Jedinični vektori u smeru x i z koordinatne ose se ne menjaju, a vektor u smeru y ose se translira duž x i z ose
- Matrica smicanja u pravcu Oxz ravni (duž y ose):

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

- Imaju primenu u generisanju kosih projekcija

Kompozicije transformacija u 3D prostoru

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

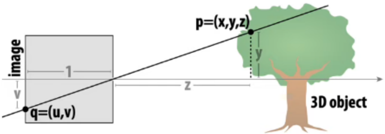
$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R^* & t^* \\ \mathbf{0} & 1 \end{bmatrix}$$

  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

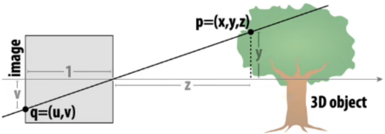
- Podsetimo se generisanja perspektivne projekcije kod tačkaste kamere
- Vrednosti x i y koordinate delimo sa z: $(x, y, z) \rightarrow (x/z, y/z)$



- Kako izvesti matricu perspektivne projekcije na ravan $z = 1$?

Perspektivna projekcija u terminima homogenih koordinata

- Podsetimo se generisanja perspektivne projekcije kod tačkaste kamere
- Vredosti x i y koordinate delimo sa z: $(x, y, z) \to (x/z, y/z)$



- Kako izvesti matricu perspektivne projekcije na ravan $z = 1$?
- 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 koordinate projekcije

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