Računalniška grafika zapiski

Tim Hajdinjak

October 24, 2024

1 Matematične osnove

Osnovni pojmi, brez katerih žal ne gre

- stolpčna matrika: $\begin{bmatrix} 2,9 \\ -4,6 \\ 0 \end{bmatrix}$
- vrstična matrika: $\begin{bmatrix} 12,5 & -9,32 & 0 \end{bmatrix}$
- transponiranje: pomeni zamenjavo osi dveh matrik: $\begin{bmatrix} 1,3 & -4,1 & 0,0 \end{bmatrix}^T = \begin{bmatrix} 1,3 \\ -4,1 \\ 0,0 \end{bmatrix}$
- enakost matrik: dve matriki sta si enaki, če imata enako število elementov po obeh oseh, in so vsi vsebovani elementi na enakih mestih
- vektor: predstavljen kot matrika, pomeni premik iz točke v točko, nimajo lokacije, smer gibanja
- seštevanje matrik: $a + b = c \iff c_i = a_i + b_i$, intuitivno:

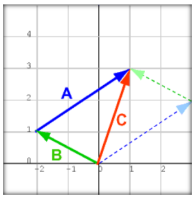


Figure 1: Geometrijsko sestevanje vektorjev. (iz repa vektorja 1 na glavo vektorja 2)

- enota za seštevanje: $a + 0 = 0 + a = a \iff \begin{bmatrix} 3 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & -1 \end{bmatrix}$
- odštevanje matrik: $a - b = c \iff c_i = a_i - b_i$
- množenje s skalarjem: $\alpha * a = b \iff b_i = \alpha * a_i$
- inverz za seštevanje: $a - a = a + -a = a + (-1)a = 0 \iff \begin{bmatrix} 2 & 5 \end{bmatrix} - \begin{bmatrix} 2 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 5 \end{bmatrix} + \begin{bmatrix} -2 & -5 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$
- NORMA oz. dolžina vektorja: $h = \begin{bmatrix} x \\ y \end{bmatrix} \implies \|h\| = \sqrt{x^2 + y^2}, oz. \|a\| = \sqrt{\sum_{i=1}^{n} a_i^2}$ (L2 norm oz. evklidska razdalja). Splošna norma: $\|a\|_p = \sqrt[p]{\sum_{i=1}^{n} |a_i|^p}$. Torej, če na izpitu reče druga splošna norma, namesto p pišeš 2. Neskončna norma → dolžina se bliža maksimalni vrednosti vektorja.
- enotski vektor, pomeni vektor dolžine 1, $\|e\| = 1$
- normalizacija: $v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \implies \hat{v} = v/\|v\| = \begin{bmatrix} v_x/\|v\| \\ v_y/\|v\| \\ v_z/\|v\| \end{bmatrix}, \hat{v}$ = enotskost vektorja (podajo samo smer), posamezne elemente vektorja delimo z njegovo dolžino.

- skalarni produkt: $u = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}, v = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} \implies u * v = u_0 * v_0 + u_1 * v_1 + u_2 * v_2$. Pravimo mu tudi "detektor pravokotnosti", saj:
  1. $u * v = 0 \rightarrow$ vektorja pravokotna
  2. $u * v < 0 \rightarrow$ iztegnjen kot
  3. $u * v > 0 \rightarrow$ ostri kot
  4. $\hat{a} * \hat{b} \rightarrow$ vrednosti med $[-1, 1], -1 = 180°, 0 = 90°, 1 = 0°$

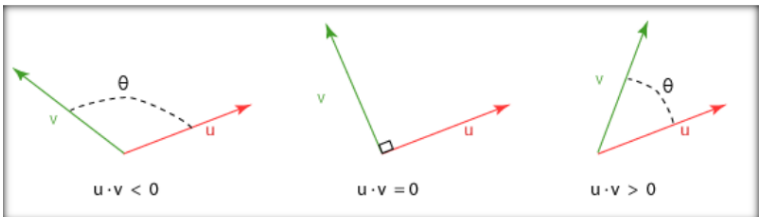


Figure 2: 2 vektorja sta ortogonalna, če je skalarni produkt enak 0 (oz. sta si pravokotna)

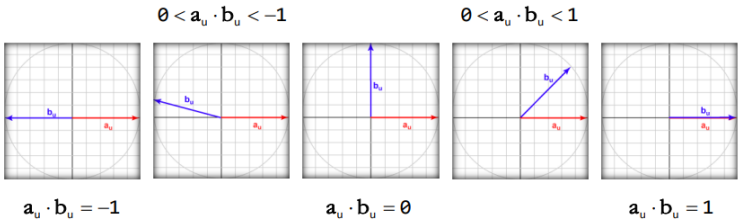


Figure 3: Skalarni produkt enotskih vektorjev

Še par pravil oz. posebnosti pri skalarnih produktih:

- $u * v = \|u\| * \|v\| * \cos \alpha$
- $v * v = \|v\|^2$, norma
- $u * 0 = 0 * u = 0$, skalarni produkt z vektorjem 0
- $0 * 0 = 0$, skalarni produkt vektorja 0
- $u * v = v * u$, komutativnost
- $u * (v + w) = u * v + u * w$, distributivnost za seštevanje
- $(\alpha u) * v = u * (\alpha v) = \alpha * (u * v)$, homogenost za množenje s skalarjem
- $u \perp v \iff u * v = 0$, skalarni produkt ortogonalnih (pravokotnih) vektorjev
- Asociativnost: nedefinirana operacija, vrstni red je pomemben

- linearna neodvisnost, projekcija vektorja na vektor: $kv = \|w\|(w_u * v_u) * v_u$

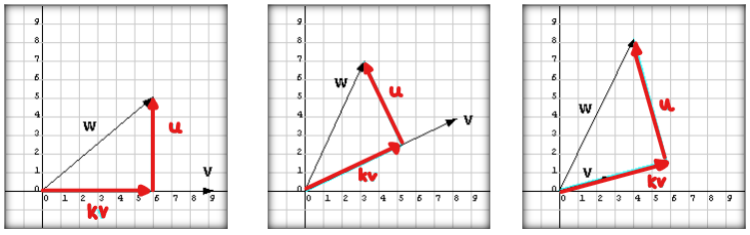


Figure 4: Projekcija vektorja na vektor

Postopek pri linearne neodvisnost:

1. Izračunaj dolžine vektorjev: $\|w\| = w * w, \|v\| = v * v$
2. Izračunaj enotske vektorje: $w_u = w/\|w\|, v_u = v/\|v\|$
3. Izračunaj kosinus kota med vektorji: $\cos\alpha = w_u * v_u$
4. Združi v projekcijo: $kv = \|w\|(w_u * v_u) * v_u$
5. Izračunaj ortogonalni vektor: $u = w - kv$

- Vektorski produkt: $u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}, v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, u \times v = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$

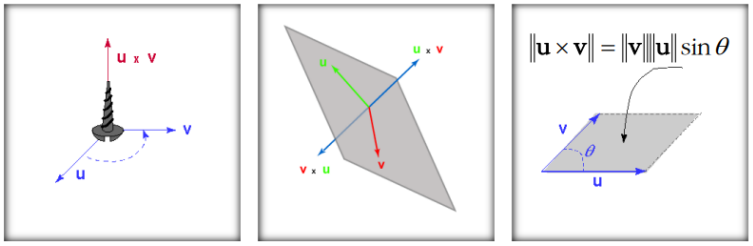


Figure 5: Vektorski produkt intuitivno

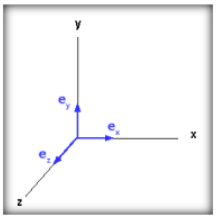


Figure 6: Enotski vektorji, ki tvorijo prostor R^3, imajo normo (dolžino) 1 in so vzajemno pravokotni. Pri tem je $e_x = (1,0,0), e_y = (0,1,0), e_z = (0,0,1)$

Zakonitosti pri vektorskem produktu:

1. $u \times v = -(v \times u)$, antikomutativnost
2. $u \times (v + w) = u \times v + u \times w$, distributivnost za seštevanje
3. $(\alpha u) \times v = u \times (\alpha v) = \alpha(u \times v)$, homogenost za množenje s skalarjem
4. Asociativnost ne obstaja: $u \times (v \times w) \neq (u \times v) \times w$
5. $u \parallel v \iff u \times v = 0$, vektorski produkt kolinearnih vektorjev
6. $u \times 0 = 0 \times v = 0$, vektorski produkt vektorja 0
7. $0 \times 0 = 0$, vektorski produkt z vektorjem 0
8. $e_x \times e_y = e_z, e_y \times e_z = e_x, e_z \times e_x = e_y$, vektorski produkt koordinatnih osi, zanimivost: vidimo lahko desno pravilo

- Splošna matrika, notacija: $A_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$
- Seštevanje matrik: $A + B = C \iff c_{ij} = a_{ij} + b_{ij}$, primer: $\begin{bmatrix} 2 & 0 \\ -1 & 2 \\ 3 & 5 \end{bmatrix} + \begin{bmatrix} 1 & 3 \\ -1 & 2 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ -2 & 4 \\ 5 & 4 \end{bmatrix}$

Zakonitosti pri seštevanju splošnih matrik:

1. Možno le, če sta matriki enakih dimenzij!
2. $A + B = B + A$, komutativnost
3. $(A + B) + C = A + (B + C)$, asociativnost
4. $A + 0 = 0 + A = A$, enota za seštevanje
5. $A - A = A + (-1)A = 0$, inverz za seštevanje

- Množenje matrik s skalarjem: $\alpha A = B \iff b_{ij} = \alpha a_{ij}$, primer: $3 * \begin{bmatrix} 2 & 0 \\ -1 & 2 \\ 3 & 5 \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ -3 & 6 \\ 9 & 15 \end{bmatrix}$

  Zakonitosti pri množenju matrik s skalarjem:

  1. $\alpha(A + B) = \alpha A + \alpha B$, distributivnost seštevanja matrik
  2. $(\alpha + \beta)A = \alpha A + \beta A$, distributivnost seštevanja skalarjev
  3. $(\alpha\beta)A = \alpha(\beta A)$, asociativnost
  4. $(-1)A = -A$, množenje s skalarjem -1

- Množenje matrik: $A_{n\times m}B_{m\times p} = C_{n\times p} \iff c_{ij} = \sum_{k=1}^{m} a_{ik}b_{kj}$

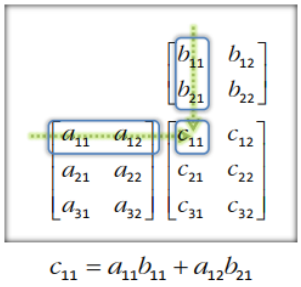


Figure 7: Miselni vzorec za množenje splošnih matrik

Zakonitosti pri množenju matrik:

1. $AB \neq BA$, komutativnost ne velja
2. $(AB)C = A(BC)$, asociativnost
3. $A(B + C) = AB + AC$, distributivnost za seštevanje
4. $(A + B)C = AC + BC$, distributivnost za seštevanje
5. $(\alpha A)B = A(\alpha B) = \alpha(AB)$, homogenost za množenje s skalarjem
6. $0A = 0$, množenje s skalarjem 0
7. $A0 = 0A = 0$, množenje z matriko 0

- Enota za množenje oz. identiteta: $I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{n\times n}$, 1 po diagonali

  1. $AB = BA = I \iff B = A^{-1}$
  2. $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$, inverz za množenje

- Transponiranje: $A^T = B \iff b_{ij} = a_{ji}$ Lastnosti transponiranja:

  1. $(A^T)^T = A$
  2. $(\alpha A)^T = \alpha A^T$
  3. $(A + B)^T = A^T + B^T$
  4. $(ABC)^T = C^T B^T A^T$
  5. $(A^{-1})^T = (A^T)^{-1}$

2 Transformacije in homogene koordinate

- Teselacija je matematični koncept, ki se nanaša na prekrivanje ravnine z enakimi (homogenimi, urejeno) ali različnimi (nehomogenimi, neurejeno) geometrijskimi oblikami brez prekrivanja. Te oblike so lahko trikotniki, kvadrati, šestkotniki,..., ki se ponavljajo na urejen način, da zapolnijo celotno površino. Več oblik dodajamo, bolj natančen bo naš objekt.
- Mozaičenje: razbitje površine na manjše koščke.
- LOD (Level of Detail), tehnika, kjer se uporabljajo različne različice 3D modela z različnimi stopnjami podrobnosti glede na razdaljo modela od kamere. Torej, ko je objekt blizu kamere, se uporablja višji nivo (več teselacije), ko pa je daleč, pa nižji (manj teselacije). Namen LOD je izboljšati učinkovitost izrisa ter optimizacijo delovanja sistema.
- Ko pa aplikacija oz. program določi natančnost objektov, jih game engine uredi tako, da zmanjša število podatkov (manj teselacije).
- Vedno delamo z ogljišči, vse ostalo je potem posledica. S trikotniki opišemo ploskovne površine.
- Linearna transformacija: $p' = f(p)$, preslikave, ki preslikajo eno točko v drugo.
- Razteg/skaliranje: enakomeren (uniformen) oz. neenakomeren (neuniformen)

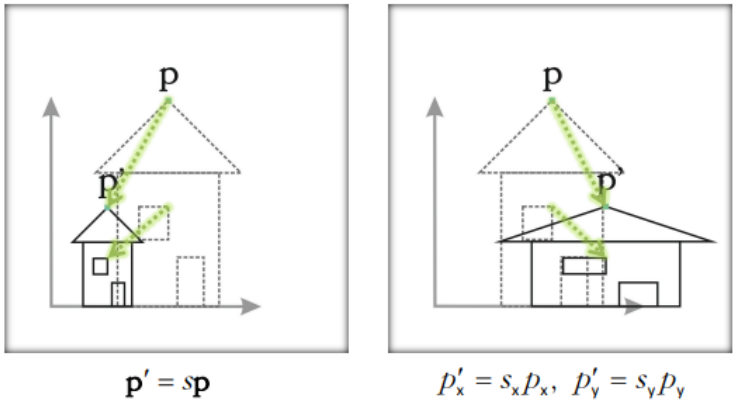


Figure 8: Primer enakomernega ter neenakomernega skaliranja

- Striženje: spreminjamo eno izmed osi na podlagi vrednosti druge osi. Če želimo striženje za kot ϕ, uporabimo kotangens kota.

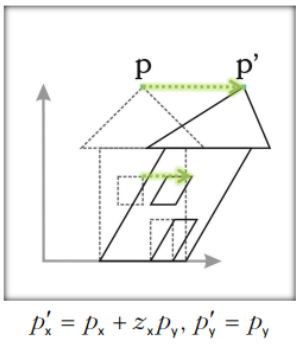
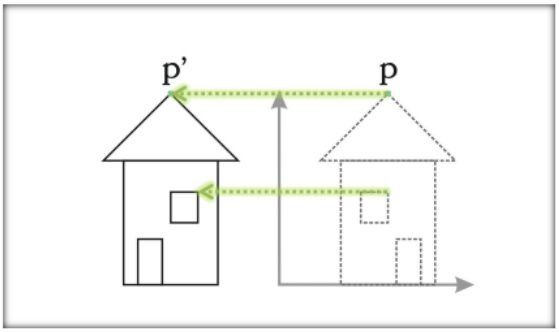


Figure 9: Primer striženja

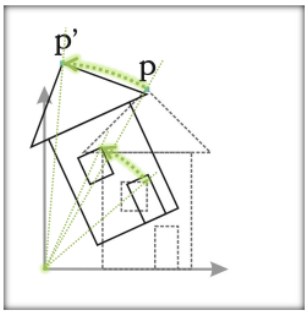
- Zrcaljenje: zrcaljenje čez koordinatne osi. Če zrcalimo dvakrat, dobimo isto. Če želimo zrcalit čez os $x = y$, uporabimo: $p'_x = p_y$ in $p'_y = p_x$ (x in y os se zamenjata). "Razteg z negativnim številom"



$p'_x = -p_x,\ p'_y = p_y$

Figure 10: Primer zrcaljenja preko Y osi

- Vrtenje: vedno vrtimo v nasprotno smer urinega kazalca okrog izhodišča.



$p'_x = p_x \cos\theta - p_y \sin\theta,\ p'_x = p_x \sin\theta + p_y \cos\theta$

Figure 11: Primer vrtenja okrog izhodišča

Izpeljava:

1. Zapis s polarnimi koordinatami:
   $p_x = \|p\| \cos\phi$
   $p_y = \|p\| \sin\phi$
2. Vrtenje v polarnih koordinatah: (povemo kot in radij (oddaljenost od izhodišča))
   $p'_x = \|p\| \cos(\phi + \theta)$
   $p'_y = \|p\| \sin(\phi + \theta)$
3. Adicijski izreki kotnih funkcij: ($\cos(\phi + \theta) = \cos\phi\cos\theta$, enako za sin)
   $p'_x = \|p\| \cos\phi\cos\theta - \|p\| \sin\phi\sin\theta$
   $p'_y = \|p\| \cos\phi\sin\theta + \|p\| \sin\phi\cos\theta$
4. Pretvorba v kartezične koordinate: (iz 1. izrazimo $\|p\|$, in se 3. pokrajša:)
   $p'_x = p_x \cos\theta - p_y \sin\theta$
   $p'_y = p_x \sin\theta + p_y \cos\theta$

- Linearne transformacije, omenjene zgoraj, lahko zapišemo kot matrike:
  - Nova točka = matrika $*$ točka
  - Matrika 2x2 za 2D
  - Matrika 3x3 za 3D

1. $p' = Mp$, nova točka = matrika $*$ točka
2. $S(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$, za skaliranje gremo vzdolž padajoče diagonale

3. $Z(z_x, z_y) = \begin{bmatrix} 1 & z_x \\ z_y & 1 \end{bmatrix}$, striženje za nek faktor
4. $Z(\theta, \phi) = \begin{bmatrix} 1 & \cot\theta \\ \cot\phi & 1 \end{bmatrix}$, striženje za kot ϕ
5. $M_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$, zrcaljenje čez Y os
6. $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, zrcaljenje čez X os
7. $M_{x=y} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, zrcaljenje čez os x=y
8. $R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, rotiranje za kot θ v 2D prostoru, za obe koordinati X in Y
9. Vrtenje v 3D prostoru: $p' = R_{x|y|z}(\theta)p$. Tukaj v bistvu vrtimo objekt preko neke koordinatne osi A, da se giblje v ravnini koordinatnih osi B in C. Primer: če vrtimo okoli koordinatne osi Z, se bo objekt vrtel preko X in Y koordinate, Y = X in Z ter X = Y in Z. To pomeni, da če vrtimo preko, ne vem, koordinatne osi Z, bomo v 3x3 matriki (ker smo v 3D), dali zadnjo vrstico in stolpec (oz. vrstici in stolpcu, ki pripada tej koordinatni osi), vse 0, razen en element na 1, kjer se stikata.
   Primer: $R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (matrika, ki zavrti objekt preko koordinatne osi Z), in preko zgornje enačbe dobimo: $p' = R_z(\theta)p = \begin{bmatrix} p_x\cos\theta - p_y\sin\theta \\ p_x\sin\theta + p_y\cos\theta \\ p_z \end{bmatrix}$. Matriki za preostali koordinatni osi X in Y: $R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$, $R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$

2.1 Afine transformacije

- $p' = Mp + t$, linearne transformacije s premiki
- $v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = v_x e_x + v_y e_y + v_z e_z$, vektor zapisan tudi s pomočjo enotskih vektorjev
- $p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = p_x e_x + p_y e_y + p_z e_z + o$, dodatek homogene koordinate (1 = točka, 0 = vektor), z njimi lahko izvajam premike (točka je 1, ker je odvisna od izhodišča (zato to prištejemo), vektor pa samo navaja premike (ni vezan na izhodišče sistema))
- $p_h = \begin{bmatrix} wp_x \\ wp_y \\ wp_z \\ w \end{bmatrix} = wp_x e_x + wp_y e_y + wp_z e_z + wo$, če pa želimo prehod iz homogene v nehomogene elemente: vsako koordinato delimo z w
- $p_h = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$, $v_h = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix}$, $p'_h = p_h + v_h = \begin{bmatrix} p_x + v_x \\ p_y + v_y \\ p_z + v_z \\ 1 \end{bmatrix}$, dobim točko, če pa odštejem, dobim vektor ("točki lahko prištejemo vektor in dobimo točko")

- Premik v homogenih koordinatah: $p' = T(t)p$, T = translate, premaknemo točko za točko t.
  Primer: $p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, t = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix} \Longrightarrow p' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}$
  Za premike obstaja inverzna operacija, s tem vrnemo neko točko nazaj kjer je bila:
  $T(t) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$, inverz: $T(t)^{-1} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- Razteg v homogenih koordinatah: $p' = S(s_x, s_y, s_z)p$, S = scale:
  $p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \Longrightarrow p' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x p_x \\ s_y p_y \\ s_z p_z \\ 1 \end{bmatrix}$
  Raztegi poznajo tudi inverzno operacijo: $S(s_x, s_y, s_z)^{-1} = S(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z})$
- Vrtenje v homogenih koordinatah (3D): $p' = R(\theta)p$, R = rotate, rotiramo za kot θ, na določeni koord. osi:
  $R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
  Vrtenje ima tudi inverzno operacijo, vendar samo, če poznamo kot: $R(\theta)^{-1} = R(-\theta) = R(\theta)^T$
- Striženje v homogenih koordinatah: $p' = Z(z_1, \ldots, z_6)p$, Z = shear, $Z(z_1, \ldots, z_6) = \begin{bmatrix} 1 & z_1 & z_2 & 0 \\ z_3 & 1 & z_4 & 0 \\ z_5 & z_6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
  V 2D prostoru gre striženje po določeni koordinatni osi, v 3D pa po vzdolž ravnine.
- Ortonorminirana baza: vektorji so med seboj pravokotni: če so matrike ortonorminirane, lahko za izračun inverza samo transponiramo
- Transformacijska matrika: matrika, v kateri lahko hkrati delamo več operacij

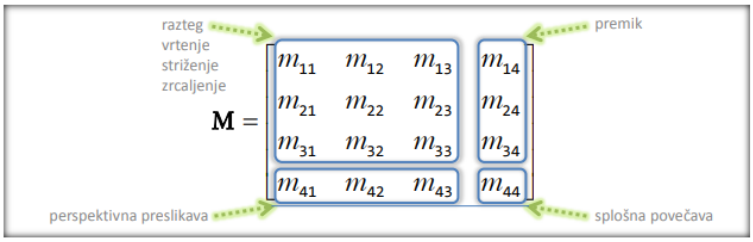


Figure 12: Definicija transformacijske matrike

2.2 Veriženje transformacij

- $p' = M_3 M_2 M_1 p$, najprej M_1, nato M_2 ter šele nato M_3 (najprej tista, ki je najbližja točki)
- Eulerjevi koti: $R_E(\theta_h, \theta_p, \theta_r) = R_z(\theta_r) R_x(\theta_p) R_y(\theta_h)$

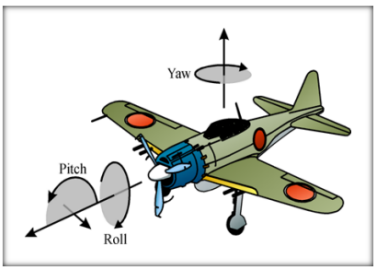


Figure 13: Eulerjevi koti, vizualno

Pitch, roll, in yaw so trije osnovni koti rotacije, ki se uporabljajo za opis orientacije telesa v tridimenzionalnem prostoru. V kontekstu Eulerjevih kotov se te rotacije nanašajo na vrtenje okrog treh pravokotnih osi, pogosto imenovanih x, y, in z osi. Eulerjevi koti so kotni parametri, ki definirajo vrtenje objekta z zaporedjem treh rotacij okoli določenih osi. Te rotacije so razdeljene na yaw, pitch, in roll, kar opisuje vrtenje glede na osnovne osi. Pomen osi:

- Yaw: Rotacija okoli navpične osi (Z), sprememba smeri levo-desno.
- Pitch: Rotacija okoli prečne osi (Y), dviganje ali spuščanje nosu.
- Roll: Rotacija okoli dolžinske osi (X), nagibanje v levo ali desno.

Eulerjevi koti se definirajo kot sekvenca treh rotacij, običajno v določenem vrstnem redu. Ena pogosta sekvenca je yaw → pitch → roll (rotacije okoli z → y → x osi), kjer vsaka naslednja rotacija vpliva na orientacijo po prejšnji. Vendar obstaja več različnih vrst Eulerjevih kotov, odvisno od izbire vrstnega reda osi.

- Kardanska zapora (angl. gimbal lock): izgubimo eno prostorsko stopnjo, če zavrtimo skozi eno os toliko, da bo poravnano z drugo osjo, ker če je 0, bo z vrtenjem tudi ostal pri 0 (zmanjša se število neodvisnih osi, saj dve osi postaneta sočasni). Je pojav, ki se pojavi pri uporabi Eulerjevih kotov za opis rotacij v tridimenzionalnem prostoru. Kardanska zapora nastane, ko se dve rotacijski osi poravnata, kar povzroči izgubo ene stopnje prostosti (degree of freedom) in vodi do nezmožnosti natančnega opisovanja rotacij okoli vseh osi.
- Zakonitosti veriženja transformacij:
  1. $p' = Mp = M_3M_2M_1p$
  2. $M_3(M_2(M_1p)) = (M_3M_2)M_1p = M_3(M_2M_1)p = (M_3M_2M_1)p$, komutativnost
  3. $p'^T = p^TM^T$, transponiranje
  4. $p'^T = p^TM_1^TM_2^TM_3^T$, transponiranje, obratni vrstni red

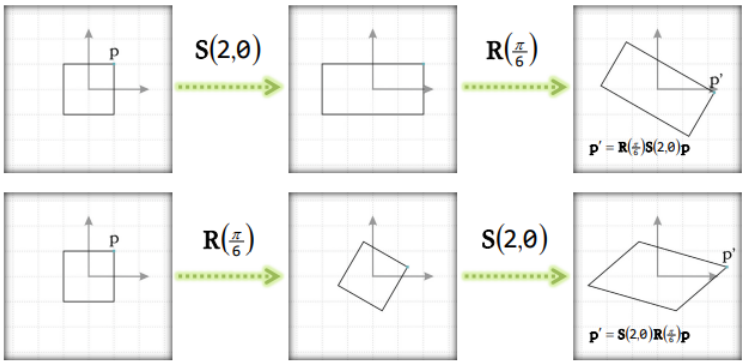


Figure 14: Primer, kako pomemben je vrstni red operacij

5. Vrstni red operacij: SRT → skaliranje → vrtenje → premiki (dogovor). Vse operacije izvajamo preko izhodišča

- Transformacije vrtenja okrog poljubne točke: $p' = M_3 M_2 M_1 p$, $p' = T(r)R(\theta)T(-r)p$

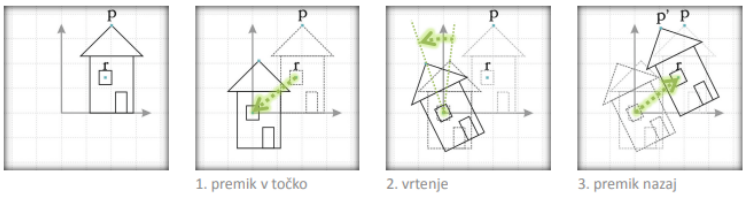


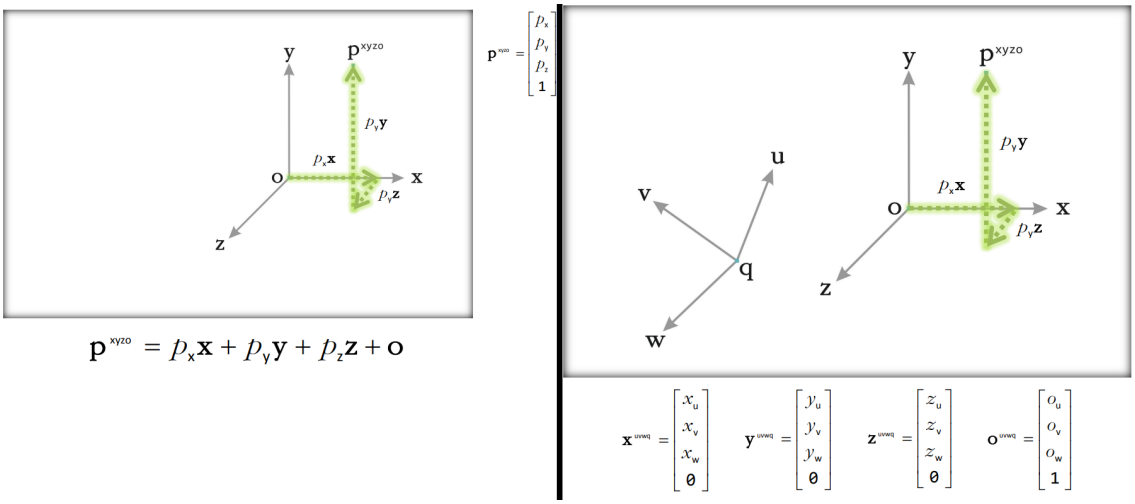
Figure 15: Postopek vrtenja okoli poljubne točke

2.3 Toge transformacije

- Vse transformacije, ki spreminjajo lokacijo in orientacijo, vendar ohranjajo obliko objekta (kot, volumen). To sta samo premik in vrtenje.

2.4 Prehodi med koordinatnimi sistemi

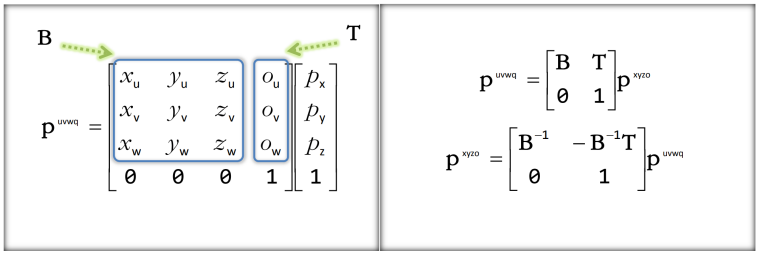
- "Točki p, podani v koord. sistemu (x,y,z,o), poišči koordinate glede na koord. sistem (u,v,w,q)"



$$\mathbf{p}^{xyzo} = p_x\mathbf{x} + p_y\mathbf{y} + p_z\mathbf{z} + \mathbf{o}$$

Tako lahko zapišemo: $p^{uvwq} = p_x x^{uvwq} + p_y y^{uvwq} + p_z z^{uvwq} + o^{uvwq}$, oz. $p^{uvwq} = \begin{bmatrix} x & y & z & o \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$

Kar lahko zapišemo v skupno transformacijsko matriko, hkrati pa dobimo tudi inverz, torej prehod iz (x,y,z,o) v (u,v,w,q) ter nazaj. Pri tem oznaka B pomeni operacije raztega, vrtenja, skaliranja ter striženja. B je tudi ortonorminirana baza, torej je inverz samo transponiranje.



$$\mathbf{p}^{uvwq} = \begin{bmatrix} \mathbf{B} & \mathbf{T} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{p}^{xyzo}$$

$$\mathbf{p}^{xyzo} = \begin{bmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{T} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{p}^{uvwq}$$

- Ortogonalna/ortonormalna matrika = njen inverz je transponirana matrika $A^{-1} = A^T$.

- Za prehod iz levosučnega v desnosučni koordinatni sistem - če imamo dva vektorja, ki določata ravnino, zrcalimo en vektor.
- Kako predstavimo vektor v homogenih koordinatah - dodamo en stolpec, ki predstavlja izhodišče.
- Zrcaljenje ni toga transformacija.
- Prehod iz homogenih v nehomogene koordinate - vzamemo zadnji stolpec in z njim delimo vse ostale stolpce (pri vektorjih tega nimamo, ker je zadnji stolpec vse 0, pri točkah pa imamo).
- Kako pridobimo nasprotno operacijo za vrtenje - če poznamo kot, vrtimo v negativni kot; če poznamo matriko, jo transponiramo, ker je ortogonalna.
- Vrtenje okoli točke = sosledje operacij - premik točke v izhodišče, premik okoli izhodišča, premik nazaj iz točke - zmnožimo matrike med sabo.

3 Projekcije

3.1 Vrste

- Projekcija je 2D ravnina, ki izgleda 3D, daje optično iluzijo "globine" → Paralaksa: vzporedne premice se stikajo nekje v neskončnosti (prostor ima globino). Gibalna paralaksa: predmeti ki so bližje, se bodo premikali hitreje.
- Vrste:
  1. Pravokotna projekcija: gledamo pravokotno na 2D ravnino (vzdolž ene osi), uporablja se predvsem v tehnični dokumentaciji, ko poznamo merila objektov

Figure 17: Primeri pravokotne projekcije (od zgoraj, čelna ter stranska)

  2. Aksonometrične oblike v pravokotni projekciji: objekt gledamo pod določenim kotom (α, β, γ), kar prikaže objekt v 3D. Poznamo trimetrične (vsak kot drugačen, torej 3 različni koti, X Y in Z so v različnem razmerju), dimetrične (2 različna kota, višina in širina sta isti, samo globina je polovična) ter izometrična (1 kot za vse, torej X Y in Z so v istem razmerju).

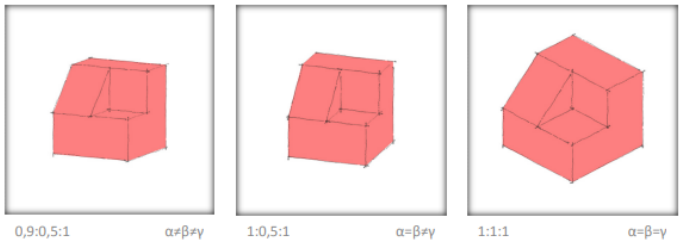


Figure 18: Primeri aksonometričnih oblik (trimetrična, dimetrične ter izometrična)

  3. Poševna projekcija: objekt vlečemo diagonalno na ravnino (namesto da gledamo pod kotom, ga poravnamo ter vlečemo diagonalno), vidimo tudi globino ter realne dimenzije. Poznamo:
     - kavalirska: globina je v enakem razmerju
     - kabinetna: globina polovično manjša

Obe podvrsti se razlikujeta v kotu vpadnic, rezultat pa je drugačna percepcija globine.

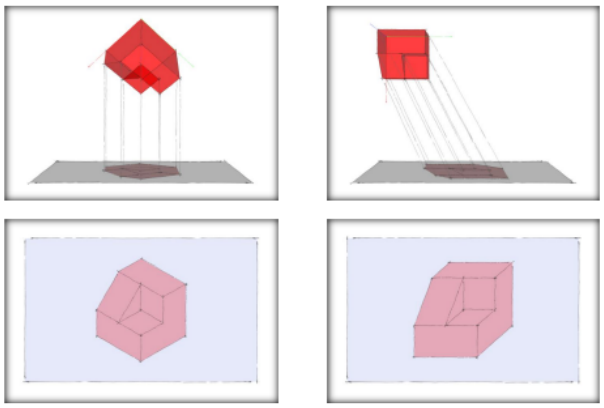


Figure 19: Primera poševnih projekcij (kavalirska ter kabinetna)

4. Perspektivna projekcija: predmet v 3D, poznamo:
   - enobežna: eno lice poravnano z ravnino
   - dvobežno: dve ravnini poravnani
   - tribežna: poljuben kot

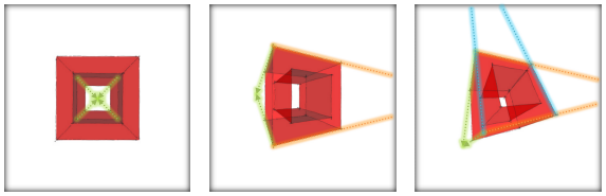


Figure 20: Primeri perspektivnih projekcij (enobežna, dvobežna ter tribežna)

3.2 Grafični cevovod

- Komunikacija s strojno opremo, vanj vstopa ena točka naenkrat.
- Procedura:

1. Model: izračuna/določi se geometrija objekta (točke (ki določajo ogljišča), povezave med točkami → ploskve), lastnosti ogljišč (barve, koordinate teksture,...), teksture, luči, kamera. Koordinatni sistem, v katerem je objekt trenutno, je poravnan z eno osjo oz. ogljiščem.
2. Transformacija modela in pogleda: objekt pozicioniramo v večji svet (koordinatni sistem sveta), katerega gledamo preko projekcije (koordinatni sistem kamere). Točke iz koord. sistema predmeta pretvorimo v koord. sistem pogleda.
3. Izračun osvetlitve: Sprehodimo se čez vsak deklariran vir svetlobe, ter izračunamo, kako osvetljena je točka glede na položaje luči.
4. Projekcija: prehod v normaliziran koordinatni sistem, predstavlja pogled kamere (prehod iz 3D koordinat v 2D) → homogene koordinate transformira v tretji koordinatni sistem.
5. Rasterizacija in prekrivanje: odgovarjamo na dve vprašanji:
   - Kako napolnit vsebino izračunanih primitivov? (izris primitivov)
   - Čim hitrejši izris ter katere primitive izrišemo? (ugotavljanje vidnosti). En znan algoritem je PAINTERS ALGORITHM (zelo potraten), primitive sortiramo glede na razdaljo kamere, vsakega pobarvamo, ter tisti, ki bo na koncu najbližji kamere, bo viden (pri tem algoritmu pomaga predpomnilnik (Z-buffer), ki hrani globino).

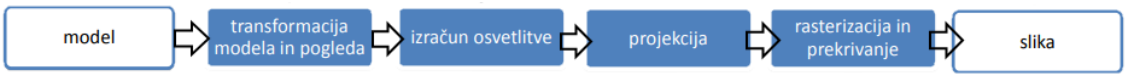


Figure 21: Grafični cevovod

3.3 Koordinatni sistemi

- Koord. sistem predmeta: en predmet v svojem koord. sistemu: p^{object}
- Koord. sistem sveta: način, kako različne predmete vežemo v prostor (veriženje transformacij): $p^{world} = Mp^{object}$. V svet vmeščamo predmete z verigo transformacij → vse te lahko zmnožimo vnaprej in dobimo matriko M, ki vse to opiše; nato vsako točko pomnožimo z matriko M.
- Koord. sistem pogleda: umeščanje kamere v prostor ("navidezen element"): $p^{camera} = C^{-1}Mp^{object}$. pogled → svet = C, medtem ko svet → pogled = C^{-1}, $C = \begin{bmatrix} u & v & w & e \end{bmatrix}$ (podamo točko iz katere gledamo (e (eye)) ter točko kaj gledamo (f = focus) → iz teh dveh vektorjev lahko izračunamo, kaj gledamo. UP → vektor, ki kaže kaj je zgoraj (kako je rotirana kamera). -Z (negativna os), Y pozitiven (v zrak), X je desno poravnan. Platno se nahaja nekje med kamero in objektom. Da pogledamo na predmet, moramo narediti inverz vmeščanja kamere v prostor (C^{-1})

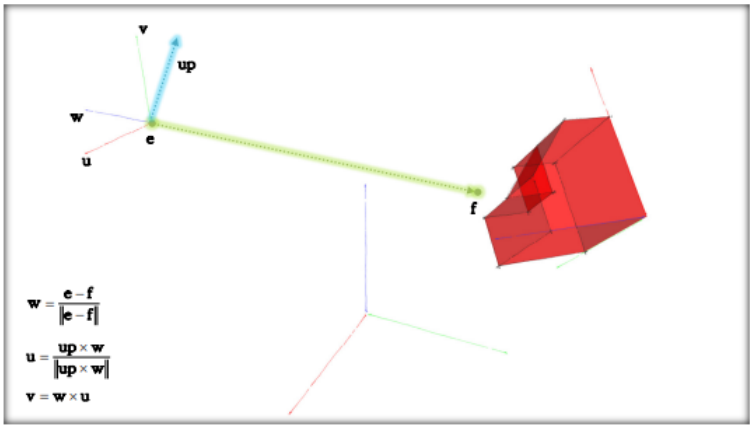


Figure 22: Koordinatni sistem pogleda

- Perspektivna projekcija: kako izračunaš p', če veš origin in fokus.

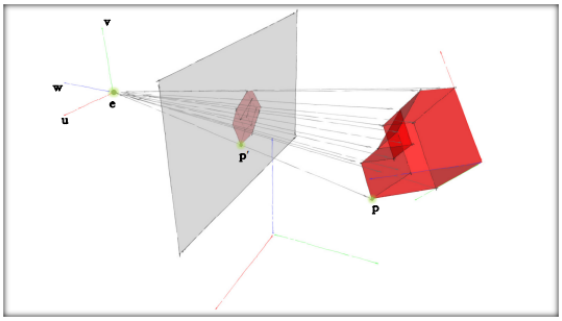


Figure 23: Perspektivna projekcija

- Vzporedna projekcija: vpadnice iz točk so vzporedne

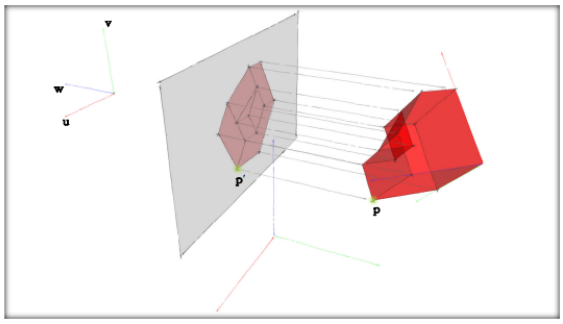


Figure 24: Vzporedna projekcija

3.4 Projekcijska transformacija

- Vzporedna pravokotna projekcija: da izračunaš p', daš Z = 0

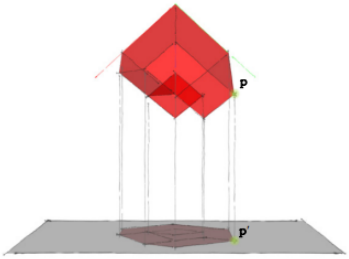


Figure 25: Vzporedna pravokotna projekcija

- Perspektivna projekcija: p' izračunamo s pomočjo trikotniškega pravila → razmerje dveh oddaljenosti = dveh višin.

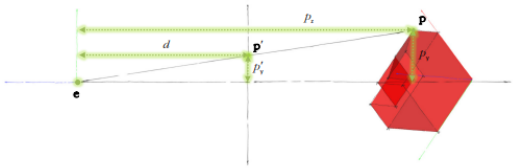


Figure 26: Perspektivna projekcija, primer

Preko trikotniškega pravila dobimo: $\frac{p'_y}{d} = \frac{p_y}{p_z}, \frac{p'_x}{d} = \frac{p_x}{p_z}, p'_z = d$, d je globina, kjer se nahaja projekcijska ravnina.
TRIK: uporaba homogenih komponent, realiziramo celotno projekcijo naenkrat (pomaga pri parallel computing, ker je vsaka točka zase). Tako dobimo transformacijsko matriko: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ p_z/d \end{bmatrix} \Longrightarrow \begin{bmatrix} p_x d/p_z \\ p_y d/p_z \\ d \\ 1 \end{bmatrix}$. Če pa tretiramo d kot oddaljenost, pa damo povsod kjer je d v matriki nasproten predznak (torej minus).

3.5 Vidno polje

- Terminologija:
  - Near clipping in far clipping edge → kar je preblizu/predaleč ne bomo videli.

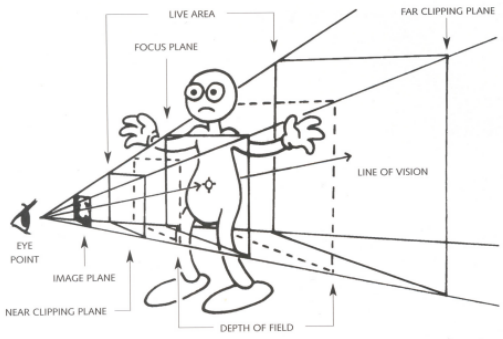


Figure 27: Prikaz vseh terminologij vidnega polja

- Perspektivna projekcija:
  - 4 stran piramida, pove kaj gledamo/kaj bomo videli (prirezana piramida gledana), ostalo ne pošiljaj v cevovod.

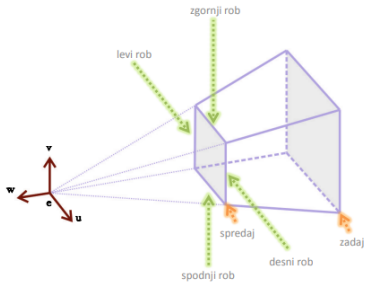


Figure 28: Perspektivna projekcija vidnega polja

  - simetrična piramida → enako razmerje višine in širine stranic (največkrat gledamo na sredino okna, ki je enako oddaljena od zgornjega, spodnjega, levega, desnega roba → vem kakšen je vertikalni zorni kot, rabim samo še spredaj in zadaj)

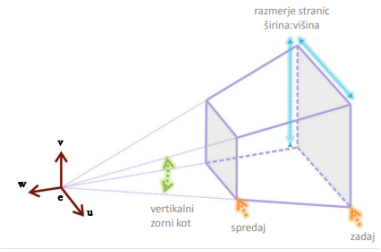


Figure 29: Perspektivna projekcija vidnega polja, simetrično

- Vzporedna projekcija: kvader/paralelepiped gledanja

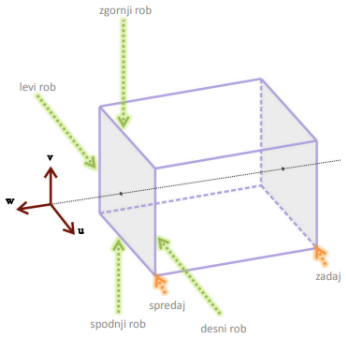


Figure 30: Vzporedna projekcija vidnega polja

3.6 Izločanje in obrezovanje

- Optimizacija izrisa, da ne render-amo celotnega vidnega polja. To naredimo tako, da piramido in kvader združimo v normaliziran sistem koordinat od $\begin{bmatrix}-1\\-1\\-1\end{bmatrix}$ do $\begin{bmatrix}1\\1\\1\end{bmatrix}$. Pri tem se vzporedna skrči (glede na razdaljo spredaj in zadaj), pri perspektivni pa se zadnja skrči, sprednja pa raztegne (to pomeni da bodo objekti bližje kamere večji, objekti dlje pa manjši.

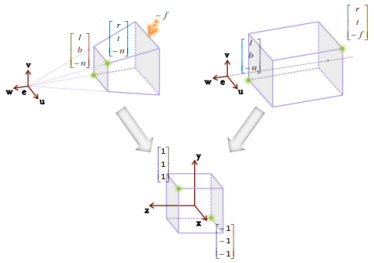


Figure 31: Obrezovanje, normalizirane koordinate

Vse to lahko združimo sedaj v projekcijsko matriko. Poznamo več vrst:

1. Perspektivna projekcija, prirezana piramida gledanja: $P_p(l,r,t,b,n,f) = \begin{bmatrix} \frac{2*n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2*n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2*n*f}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$,

   l = left, r = right, t = top, b = bottom, n = near, f = far

2. Perspektivna projekcija, simetrična piramida gledanja: $P_p(\alpha,a,n,f) = \begin{bmatrix} \frac{1}{a-\tan\frac{\alpha}{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\frac{\alpha}{2}} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2*n*f}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$,

   α = kot kamere, a = razmerje (stiskanje/povečevanje), n = near, f = far

3. Pravokotna projekcija: $P_o(l,r,t,b,n,f) = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$

4. Pravokotna projekcija, simetrična: $P_o(w,h,n,f) = \begin{bmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$, w = width,

h = height, n = near, f = far. Ni več perspektive, imamo pa translacijo, da nam zamakne kvader na sredino normaliziranega koordinatnega sistema.

5. Poševna projekcija, zamik sveta s pomočjo striženja (zamaknemo koord. objektov), nato pa izvedemo pravokotno projekcijo. $P = P_o H(\theta, \phi) = P_o \begin{bmatrix} 1 & 0 & \cot\theta & 0 \\ 0 & 1 & \cot\phi & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
6. Matrika naprave oz. transformacija v koordinate naprave, pove kako za ekrane (monitorje), razpon (0,0) do npr. (1920, 1080) izrišemo posamezne piksle na zaslon: $D(x_o, x_1, y_0, y_1) = \begin{bmatrix} \frac{x_1 - x_0}{2} & 0 & 0 & \frac{x_0 + x_1}{2} \\ 0 & -\frac{y_1 - y_0}{2} & 0 & \frac{y_0 + y_1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Pri tem pazi, ker sta X in Y obrnjena, torej z večanjem X greš bolj levo, pri Y pa bolj dol.
7. Transformacijska veriga: vse kar smo obdelali, združimo: $p' = DPC^{-1}Mp$, D = koord. sistem naprave, P = normaliziran koord. sistem, C = koord. sistem pogleda, M = koord. sistem sveta, p = koord. sistem predmeta. Točka je tako predstavljna z $p' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ w' \end{bmatrix}$, posamezen piksel/slikovni element pa dobimo z p'_x/w' in p'_y/w'

4 Krivulje in ploskve

- V računalniški grafiki ni zaoblenih robov.
- Krivulje nastanejo, ko iz dovolj velike razdalje gledamo ravno površino.
- Uporablja se v risbah, skicah, grafih, tipografijah (TTF - true type font), modeliranje,...
- podajanje krivulj: krivulje podamo kot množico točk, te so lahko:
  1. kontrolne: tiste točke, ki vplivajo na krivuljo (vpliv na krivuljo je lokaliziran)
  2. interpolirane: tiste točke, skozi katere krivulja gre. Če bi krivulja imela le interpolirane točke, bi bilo neskončno možnih predstavitev krivulje.
- Hermitska krivulja: q_0 in q_1 aproksimirata krivuljo, krivulja postane interpolirana, tangenta pa pove, v katero smer krivulja potuje.

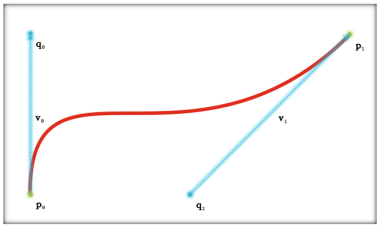


Figure 32: Primer hermitske krivulje

- Enačba krivulje: krivuljo lahko matematično (v obliki grafov) zapišemo na 3 načine:
  1. eksplicitno: povemo 1 odvisno koordinato, ki temelji na neodvisni, npr. $p_y = kp_x + m$. Problem: pri isti vrednosti x dobiš enako vrednost, ne moreš zapisat krivulje, ki se križa v eni točki.

2. implicitno: uporabimo obe koordinati in enačimo z 0, sedaj lahko zapišemo krožnice ($p_x^2 + p_y^2 - r^2 = 0$), parabole, vendar še zmeraj ne moreš zapisat krivulj, ki se križajo. Prednost tega pristopa je, da če veš X, lahko izračunaš Y ter lahko izveš, ali se določena točka nahaja v/izven krivulje (< 0 = znotraj, > 0 = izven).
3. parametrično: zapis krivulje ni odvisen od koord. sistema. Vsako koordinato se zapiše kot funkcijo, imamo opravka tudi s polinomi. Točka kot funkcija ene spremenljivke pomeni sprehod po krivulji oz. sprehod po zalogi vrednosti funkcije. V vsaki točki lahko tudi izračunamo odvod, s tem dobimo tangento, ki pa predstavlja smer gibanja točk. $p(t) = \begin{bmatrix} p_x(t) \\ p_y(t) \end{bmatrix}, p_x(t) = f(t), p_y(t) = g(t)$

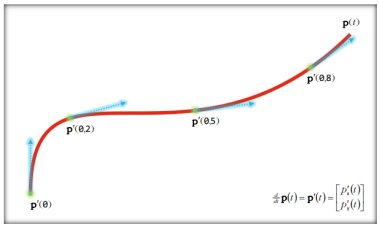


Figure 33: Primer parametriče krivulje, vidimo lahko sprehod po zalogi vrednosti $[0, 1]$ ter tangente.

- Polinomska funkcija: $f(t) = \sum_{i=0}^{n} c_i t^i$, vsota i do n (št. prostostih stopenj) koeficientov pomnoženih z neodvisno komponento t, na i-to potenco. Krivulja n-te stopnje ima (n+1) koeficientov/členov ter lahko (n-1)-krat zamenja smer.

4.1 Polinomske krivulje

- Razžirjena polinomska funkcija ima n dimenzij, namesto koeficientov imamo stolpčne matrike. $p(t) = \sum_{i=0}^{n} c_i t^i$

$$\mathbf{p}(t) = \mathbf{c}_0 + \mathbf{c}_1 t$$

$$\mathbf{p}(t) = \begin{bmatrix} p_x(t) \\ p_y(t) \\ p_z(t) \end{bmatrix}, \quad \mathbf{c}_0 = \begin{bmatrix} c_{0x} \\ c_{0y} \\ c_{0z} \end{bmatrix}, \quad \mathbf{c}_1 = \begin{bmatrix} c_{1x} \\ c_{1y} \\ c_{1z} \end{bmatrix}$$

$$p_x(t) = c_{0x} + c_{1x} t$$
$$p_y(t) = c_{0y} + c_{1y} t$$
$$p_z(t) = c_{0z} + c_{1z} t$$

Figure 34: Primer zapisa linearne krivulje

- Linearna interpolacija dveh točk: funkcija lerp: $p(t) = lerp(t, p_0, p_1)$. Točki p_0 in p_1 predstavljata $t = 0$ in $t = 1$. Pri tem je t sprehod od p_0 do p_1, s tem da ga vlečeš po zalogi vrednosti. Imamo 3 različne predstavitve:
  1. Utežena vsota dveh točk: $p(t) = p_0(1-t) + p_1 t = p_0 B_0(t) + p_1 B_1(t)$. Funkciji $B_0(t) = 1 - t$ in $B_1(t) = t$ sta uteži (blending function), in če v katerikoli točki sešteješ njuni vrednosti, dobiš 1 (utežena vsota dveh točk).
  2. Zapis v obliki polinoma: $p(t) = p_0 + (p_1 - p_0)t = p_0 + vt$. Pri tem $v = p_1 - p_0$ predstavlja smer gibanja točk oz. kako se premaknit do cilja.

3. Matrični zapis: $p(t) = GBT(t) = \begin{bmatrix} p_0 & p_1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ t \end{bmatrix}$, G = geometrij oz. točke ki jih podam, B = matrika geometrijske baze oz. matrika ki podaja mešalne funkcije, ter T = matrika polinoma. Če množimo najprej GB skupaj ter nato s T, dobimo zapis koeficientov s polinomi, če pa BT ter nato z G, pa zapis točk z utežmi.

$$\mathbf{p}(t) = (\mathbf{GB})\mathbf{T}(t) = \mathbf{CT}(t)\begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 - \mathbf{p}_0 \end{bmatrix}\begin{bmatrix} 1 \\ t \end{bmatrix} = \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0)t$$

$$\mathbf{p}(t) = \mathbf{G}(\mathbf{BT}(t)) = \mathbf{GB}(t) = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 \end{bmatrix}\begin{bmatrix} 1-t \\ t \end{bmatrix} = \mathbf{p}_0(1-t) + \mathbf{p}_1 t$$

- Pri izračunu tangente krivulje, je to najlažje v matričnem zapisu, saj se matrika T spremeni iz $1, t$ v $0, 1$.

4.2 Hermitske krivulje

- Kubični polinomi
- Podati moramo začetno in končno točko, ter vektorja, ki predstavljata odvod (oz. tangenti na krivuljo). Koncept se posreduje v Bezierjeve krivulje.

4.3 Bezierjeve krivulje

- "Konveksna ovojnica kontrolnih točk"
- Konveksna ovojnica: povezane točke, ki predstavljajo lik (vsaj 4 točke). Točke ne zapustijo krivulje.
- Kontrolni poligon: nastala krivulja ne bo večkrat zamenjali smeri kot ta poligon (točke povežemo po vrstnem redu).
- Prva in zadnja točka sta interpolirani, vse ostale pa na krivuljo vplivajo (kontrolne točke).
- de Casteau konstrukcija: vzameš t pri določeni vrednosti in povežeš po linearni interpolaciji z drugimi vrednostimi (q), to rekurzivno ponavljaš (za vsako vrednost t) in dobiš krivuljo.

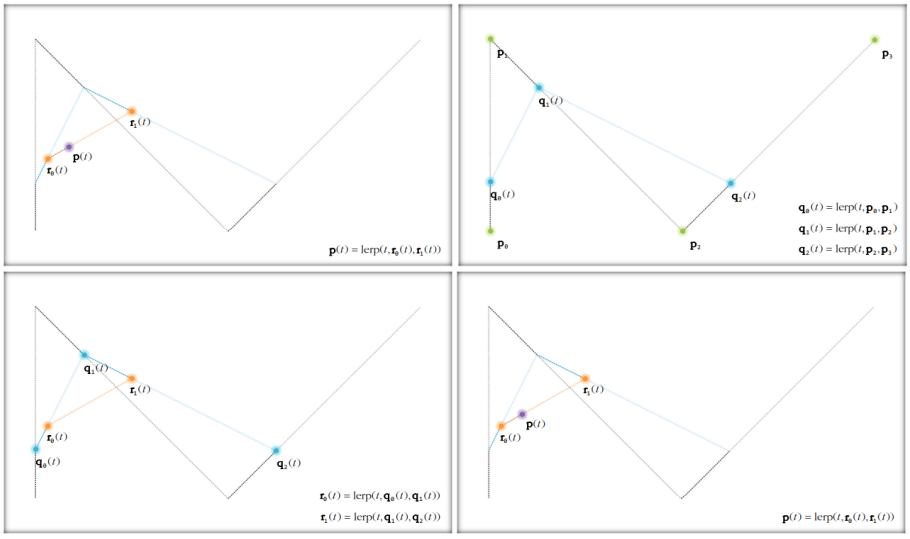
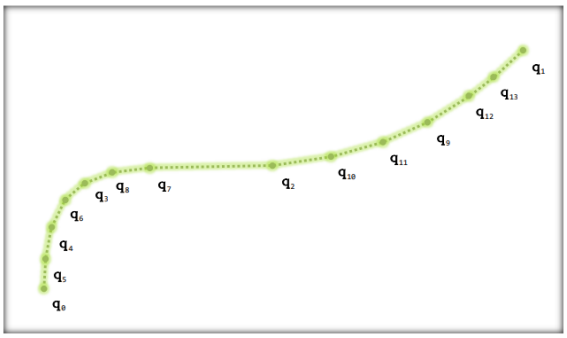


Figure 35: de Casteau kontrukcija, intuitivno

- Utežena vsota kontrolnih točk: B_1 in B_2 zalogo vrednosti 1 pri t nikoli ne bosta dosegla (gresta proti njej, ampak ne čež njo, ker ti dve povlečeta k sebi).
- Bernsteinovi polinomi: mešalna funkcija B s stopnjo.
- Pri matričnem zapisu, vse krivulje imajo isto matriko geometrijske baze, le matrika polinoma se spreminja.
- Pri risanju Bezierjevih krivulj, pogledaš kako ravna je krivulja v določeni točki in če je dovolj ravna, bo na tistem območju malo točk, če ne pa več točk.



4.4 Sestavljene krivulje