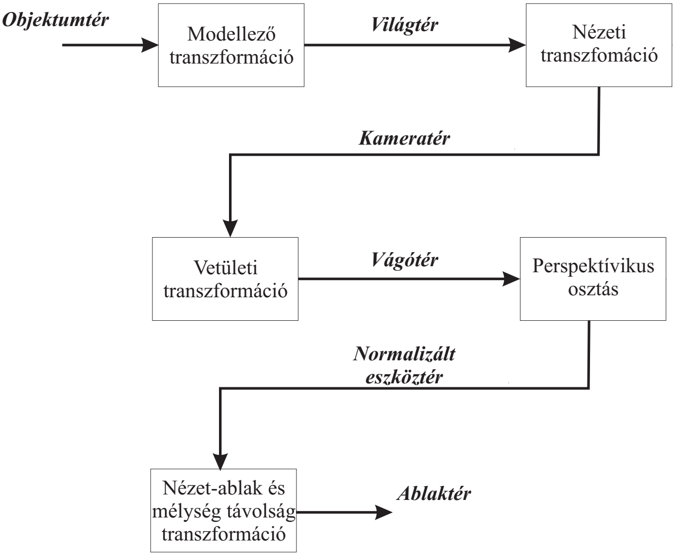
Geometriai transzformációk

- Objektumok transzformálása és animálása a 3D-s színtéren
- Mátrix és oszlop vektor szorzása
  - $\mathbf{x}' = \mathbf{A}\mathbf{x}$
    - $\mathbf{x}'$ transzformált oszlopvektor (vertex pozíció)
    - $\mathbf{A}$ transzformációs mátrix (transzformációt leíró mátrix)
    - $\mathbf{x}$ transzformálandó oszlopvektor (vertex pozíció)
  - $\mathbf{x}'' = \mathbf{B}\mathbf{x}' = \mathbf{B}\mathbf{A}\mathbf{x}$

Transzformációs csővezeték

Transzformációs csővezeték



- Vertex pozíció meghatározása egy koordináta rendszerben
- Minden objektum egy saját objektumtérrel rendelkezik
- Pozíció ábrázolása vektorként (pl. koordináta-hármasok)
- Transzformációk segítségével az egyik térben lévő pozíciót egy másik térben lévő pozícióra képezünk le
- Normálvektor
    - Adott felületre merőleges egység hosszú vektor

- Négy-komponensű (x, y, z, w) alak
  - Egy Descartes koordinátával megadott (x, y, z) helyvektor speciális esete
- $(x, y, z, w) = (x', y', z', w')$,
  - ha $\exists$ egy olyan $h \neq 0$,
  - hogy $x' = hx$, $y' = hy$, $z' = hz$, $w' = hw$
- $w = 0$ homogén pozíció esetén végtelenben lévő pont
- $(0, 0, 0, 0)$ homogén koordináta nem megengedett

- $w \neq 0$ esetén (x, y, z, w) szokásos jelölése
  - $\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}, 1\right)$
- (x, y, z) Descartes koordinátához egy 1-est negyedik komponensként hozzávéve adhatjuk meg a homogén alakot
  - $(x, y, z, 1)$
- Egy (x, y, z, w) homogén pozíció homogén osztás után
  - $(x', y', z', 1)$ pozícióként fog megjelenni
  - Az utolsó 1-es komponens elhagyásával kapjuk meg a homogén pozícióhoz tartozó Descartes koordinátát

- Objektumok között nincsenek térbeli viszonyok definiálva az objektumtéren
- Az objektumok egymáshoz való viszonyuk meghatározása/megadása
- Egy objektumtérben lévő modell homogén vertex pozíciójának ($w = 1$) modellező transzformációval való szorzása
    - A modell világtérbe transzformált pozíciója

- 3D-s modellek elhelyezése a világtérben
  - Forgatás
  - Eltolás
  - Skálázás
  - ...
- Transzformáció megadása
  - 4×4-es homogén transzformációs mátrixok
  - Több transzformáció kombinálása mátrix szorzással egyetlen egy 4×4-es mátrixba

- Egy bizonyos nézőpontból tekintünk a létrehozott színtérre
- A szem a koordináta-rendszer origójában van
- Azt a transzformációt, ami a világtéren lévő pozíciókat a kameratérre viszi át nézeti transzformációnak nevezzük

- 4×4-es mátrix
- Tipikus nézeti transzformáció a világtéren lévő kamera pozícióját a kameratér origójába viszi
  - Eltolás és elforgatás
- Meghatározza a kamera helyét és irányítottságát
- Modell-nézeti mátrix
  - Árnyalási számítások kamera- vagy objektumtérben
  - A modellező és nézeti transzformációs mátrixok összeszorzása

- $\mathbf{T}$ eltolás mátrix

$$\mathbf{T}(\mathbf{t}) = \mathbf{T}(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- $\mathbf{T}(\mathbf{t})$ alkalmazásával egy $\mathbf{p}$ pontot a $\mathbf{p}'$ pontba tolhatunk

$$\mathbf{p}' = (p_x + t_x, p_y + t_y, p_z + t_z, 1).$$

- Inverz transzformációs mátrix
  - $\mathbf{T}(\mathbf{t})^{-1} = \mathbf{T}(-\mathbf{t})$

- $\mathbf{R}_x(\phi)$, $\mathbf{R}_y(\phi)$ és $\mathbf{R}_z(\phi)$ forgatási mátrixok
  - x, y és z tengelyek körüli forgatás ϕ szöggel

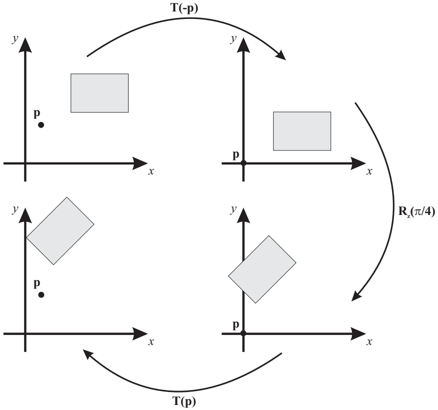
$$\mathbf{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

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

- Bal-felső 3×3-as részmátrix diagonális elemeinek összege (mátrix nyoma) állandó
  - $\mathrm{tr}(\mathbf{R}) = 1 + 2\cos\phi$
- Forgatási mátrix ortogonális mátrix
  - Inverze megegyezik a mátrix transzponáltjával
  - $\mathbf{R}^{-1} = \mathbf{R}^T$
- Inverz forgatási mátrix
  - $\mathbf{R}_i^{-1}(\phi) = \mathbf{R}_i(-\phi)$
- Forgatási mátrix determinánsa mindig eggyel egyenlő

- A z tengellyel párhuzamos, p ponton átmenő tengely körüli forgatás
  - $\mathbf{X} = \mathbf{T}(\mathbf{p})\mathbf{R}_z(\phi)\mathbf{T}(-\mathbf{p})$



- $\mathbf{S}(\mathbf{s}) = \mathbf{S}(s_x, s_y, s_z)$ (ahol $s_x \neq 0, s_y \neq 0$ és $s_z \neq 0$) skálázó mátrix
  - az x, y és z irányokban az s_x, s_y és s_z értékekkel skáláz (kicsinyít/nagyít)

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

- Uniform skálázás
  - $s_x = s_y = s_z$
- Inverz skálázás
  - $\mathbf{S}^{-1}(\mathbf{s}) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$

- Uniform skálázási mátrix létrehozása
  - Egy homogén koordináta-vektor w komponensének a manipulációjával

$$\mathbf{S}(s) = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/5 \end{pmatrix}$$

- Tükröző mátrix,
  - ha **s** három értékéből egy negatív

- Adott irányba történő skálázás
  - $\mathbf{f}^x$, $\mathbf{f}^y$ és $\mathbf{f}^z$ ortonormált vektorok mentén
- $\mathbf{F}$ mátrix létrehozása

$$\mathbf{F} = \begin{pmatrix} \mathbf{f}^x & \mathbf{f}^y & \mathbf{f}^z & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

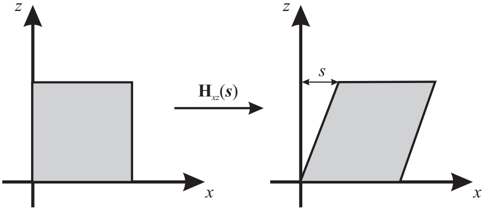
- $\mathbf{F}$ inverzével szorzunk
  - vagyis az $\mathbf{F}$ mátrix transzponáltjával
- Skálázás majd visszatranszformálás

$$\mathbf{X} = \mathbf{F}\mathbf{S}(s)\mathbf{F}^T$$

- Hat alap nyírást $\mathbf{H}_{xy}(s)$, $\mathbf{H}_{xz}(s)$, $\mathbf{H}_{yx}(s)$, $\mathbf{H}_{yz}(s)$, $\mathbf{H}_{zx}(s)$, $\mathbf{H}_{zy}(s)$ mátrixokkal jelöljük
  - Első index azt a koordinátát jelöli, amelyet a nyíró mátrix megváltoztat
  - Második index azt koordinátát jelöli, amely értékétől a változás mértéke függ
- $\mathbf{H}_{xz}(s)$ nyíráshoz tartozó mátrix

$$\mathbf{H}_{xz}(s) = \begin{pmatrix} 1 & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- $\mathbf{P} = (p_x, p_y, p_z)^T$ pontot balról megszorozva
  - $\mathbf{P}' = \mathbf{H}_{xz}(s)\mathbf{P} = (p_x + sp_z, p_y, p_z)^T$



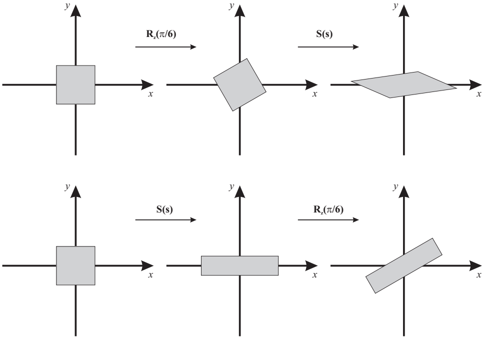
Az y és z értékek nem változnak a transzformáció során

- $\mathbf{H}_{ij}(s)$ (ahol $i \neq j$) mátrix inverze
  - Ellentétes irányba való nyírás
  - $\mathbf{H}_{ij}^{-1}(s) = \mathbf{H}_{ij}(-s)$
- Nyírás más formában való megadása

$$\mathbf{H}'_{xy}(s,t) = \begin{pmatrix} 1 & 0 & s & 0 \\ 0 & 1 & t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  - A két indexszel azt jelöljük, hogy két koordinátát nyírunk a harmadikkal
  - $\mathbf{H}'_{ij}(s,t) = \mathbf{H}_{ik}(s)\mathbf{H}_{jk}(t)$, ahol k a harmadik koordinátát jelöli
- A nyírás térfogat megőrző transzformáció
  - $|\mathbf{H}| = 1$

- Mátrix szorzás nem kommutatív
- Transzformációk végrehajtásának a sorrendje hatással van az eredményre
- Tetszőleges $\mathbf{N}$ és $\mathbf{M}$ mátrixokra $\mathbf{NM} \neq \mathbf{MN}$
- $\mathbf{TRSp} = \mathbf{T}(\mathbf{R}(\mathbf{Sp}))$



- Csak eltolás és forgatás transzformációk összefűzésével áll elő
  - A hosszakat és szögeket megőrzi
  - Felírható $\mathbf{T}(\mathbf{t})$ eltolás- és $\mathbf{R}$ elforgatásmátrix szorzataként

$$\mathbf{X} = \mathbf{T}(\mathbf{t})\mathbf{R} = \begin{pmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Inverze kiszámítható

$$\mathbf{X}^{-1} = (\mathbf{T}(\mathbf{t})\mathbf{R})^{-1} = \mathbf{R}^{-1}\mathbf{T}(\mathbf{t})^{-1} = \mathbf{R}^T\mathbf{T}(-\mathbf{t})$$

- Az inverz másképpen is kiszámítható

$$\overline{\mathsf{R}} = (\mathbf{r}_{,0} \quad \mathbf{r}_{,1} \quad \mathbf{r}_{,2}) = \begin{pmatrix} \mathbf{r}_{0,}^T \\ \mathbf{r}_{1,}^T \\ \mathbf{r}_{2,}^T \end{pmatrix}$$

$$\Rightarrow$$

$$\mathbf{X} = \begin{pmatrix} \overline{\mathsf{R}} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

$$\mathbf{X}^{-1} = \begin{pmatrix} \mathbf{r}_{0,} & \mathbf{r}_{1,} & \mathbf{r}_{2,} & -\overline{\mathsf{R}}^T\mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Lineáris transzformációk esetén pontok különbségének a képe a képek különbsége
  - $\mathbf{A}(\mathbf{x_1} - \mathbf{x_2}) = \mathbf{Ax_1} - \mathbf{Ax_2}$
- Nem szögtartó transzformációkat tartalmazó mátrixok
  - skálázás, nyírás
  - Nem mindig használhatóak fel normálvektorok transzformációjára
- A normálvektorokat a geometriai transzformációs mátrix inverzének a transzponáltjával kell megszorozni
  - $\mathbf{M}$ geometriai transzformáció
  - $\mathbf{N} = (\mathbf{M}^{-1})^T$

- Gyakorlatban
  - Csak forgatások alkalmazása esetén
    - Mátrix inverze maga a transzponált mátrix
    - Nem kell kiszámolni az inverzet
  - Csak eltolások alkalmazása esetén
    - Nincs hatással a normálvektor irányára
  - Eltolás és forgatások után nem kell normalizálni
    - Megőrzik a hosszokat
  - Több uniform skálázás alkalmazása esetén
    - Az irány nem módosul
    - Csak a hosszát kell normalizálni
    - Ha ismerjük a skálázás mértékét, akkor osztással elvégezhető
- Ha mégis ki kell számolni az inverzet
  - Elegendő a mátrix bal felső 3×3-as mátrixának az adjungáltjának a transzponáltját meghatározni
  - Osztásra nincs szükség, mivel végül normalizálni kell

- Ha a mátrix egy transzformációt vagy egyszerű transzformációk sorozatát tartalmazza adott paraméterekkel
  - A mátrixot egyszerűen lehet invertálni a paraméterek invertálásával és a mátrixok sorrendjének a megfordításával
  - $\mathbf{M} = \mathbf{T}(\mathbf{t})\mathbf{R}(\phi)$, akkor $\mathbf{M}^{-1} = \mathbf{R}(-\phi)\mathbf{T}(-\mathbf{t})$
- Ha a transzformációs mátrix ortogonális
  - A mátrix transzponáltja az inverze $\mathbf{M} = \mathbf{M}^T$
- Ha semmilyen információnk nincs
  - ***Adjungált módszer***
  - ***Cramer szabály***
  - LU felbontás
  - Gauss elimináció

- A kameratérben lévő primitíveket a képsíkra kell leképezni
  - A homogén koordináták miatt w súllyal vannak módosítva
- Kanonikus térfogat
  - Normalizált eszköz koordináták
- Vágókockán kívül eső objektumok levágása
- Kameratér koordinátáit a vágótér koordinátáiba a vetületi transzformáció segítségével transzformáljuk át

- A nézeti térfogat pontos alakja a vetületi transzformáció típusától függ
  - A perspektív transzformáció egy csonka gúlát határoz meg
  - Az ortogonális vetítés egy téglatestet visz át vágókockába
- Adott térfogaton belül lévő alakzatok láthatóak

- Párhuzamos vonalak párhuzamosak maradnak
- Legegyszerűbb transzformációs mátrix

$$\mathbf{P}_O = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- $\mathbf{P}_O$ nem invertálható, mivel a determinánsa $|\mathbf{P}_O| = 0$

- A néző számára probléma, hogy a pozitív és a negatív z értékeket levetíti a vetítési síkra
  - z irányba nem lehet vágást végrehajtani
  - A takarásban lévő felületek kezelése problémás
  - x, y és z értékek bizonyos intervallumra való lekorlátozása
    - Például a közeli síktól a távoli síkig

- Mátrix megadása egy hatossal (l, r, b, t, n, f)
  - bal, jobb, alsó, felső, közeli és távoli síkok megjelölése
- A mátrix skálázza és eltolja az ezekkel a síkokkal kialakított Tengelyhez Igazított Befoglaló Dobozt
  - Origó középpontú, tengelyhez-igazított vágókockába
  - Bal-alsó sarka (l, b, n)
  - Jobb-felső sarka (r, t, f)

- OpenGL-ben a z-tengely negatív része felé nézünk
  - $-z$ féltérben lévő modelleket szeretnénk megjeleníteni
- $n > f$, mivel a z-tengely negatív irányába nézünk
- Könnyebb dolgunk van akkor, ha a közeli értékek kisebbek, mint a távoliak
  - Felfoghatóak pozitív távolságoknak a nézeti irány mentén
  - Nem pedig z szem koordináta értékekként

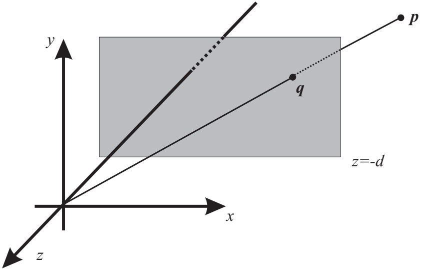
- Ortogonális transzformációs mátrix megadása:

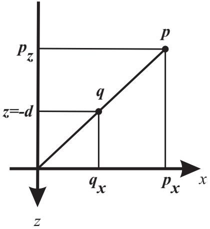
$$\mathbf{P}_o = \mathbf{S}(\mathbf{s})\mathbf{T}(\mathbf{t}) = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & -\frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- A mátrix invertálható
  - $\mathbf{P}_o^{-1} = \mathbf{T}(-\mathbf{t})\mathbf{S}((r-l)/2, (t-b)/2, (f-n)/2))$

- Perspektív vetítési mátrix
  - $z = -d, d > 0$ síkra képez le
  - A nézőpont az origóban van
  - Egy **p** pontot vetítünk
  - Képe $\mathbf{q} = (q_x, q_y, -d)$
- Hasonló háromszögek
  - $\frac{q_x}{p_x} = \frac{-d}{p_z} \implies q_x = -d\frac{p_x}{p_z}$
  - $q_y = -d\frac{p_y}{p_z}$
  - $q_z = -d$





$$\mathbf{P}_p = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{pmatrix}$$

$$\mathbf{q} = \mathbf{P}_p\mathbf{p} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} =$$

$$= \begin{pmatrix} p_x \\ p_y \\ p_z \\ -p_z/d \end{pmatrix} \Rightarrow \begin{pmatrix} -dp_x/p_z \\ -dp_y/p_z \\ -d \\ 1 \end{pmatrix}$$

- Az utolsó lépésben a w komponenssel elosztjuk a vektort

- Van egy perspektív transzformáció, amely a nézeti csonka gúlát a kanonikus nézeti térfogatba transzformálja
- Csonka gúla nézet esetén feltesszük
  - $z = n$-ben kezdődik és a $z = f$-ben végződik $0 > n > f$ értékek esetén
- Téglalap $z = n$ esetén
  - Bal alsó sarok (l, b, n)
  - Jobb felső sarok (r, t, n)

- A kamera nézeti csonka gúlájának meghatározása
  - (l, r, b, t, n, f)
- A vízszintes látómezőt a csonka gúla bal és jobb síkjai között lévő szög határozza meg
- A függőleges látómezőt az alsó és felső síkok közötti szög
- Minél nagyobb a látószög, annál többet „lát” a kamera
- Aszimmetrikus csonka gúla - sztereó látás
  - $r \neq -l$ vagy $t \neq -b$ értékekkel hozhatunk létre

- Látómező - Színtér észlelése
- Számítógép képernyője
  - Szem fizikai látó mezeje
- $\phi = 2\arctan(w/(2d))$
  - ϕ a látómező
  - w a színtér szélessége
  - d az objektumtól való távolság

- Szélesebb látómezőt beállítva az objektumok torzítva fognak megjelenni
- Perspektív transzformációs mátrix
  - A csonka gúlát az egység kockába transzformálja

$$\mathbf{P}_p = \begin{pmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

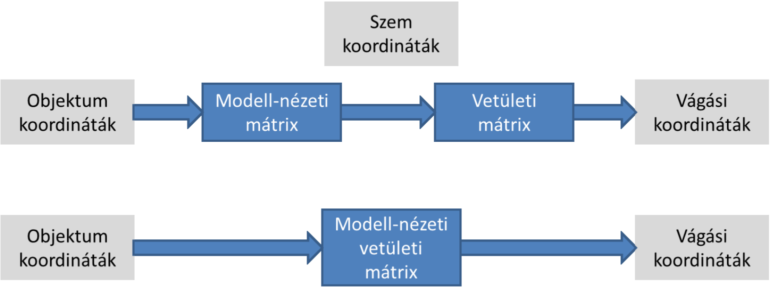
- $\mathbf{q} = (q_x, q_y, q_z, q_w)^T$ pontot transzformálva
  - w komponens értéke (a legtöbb esetben) nem nulla lesz és nem lesz egyenlő eggyel
- A vetített $\mathbf{p}$ ponthoz osztanunk kell q_w-vel
  - $\mathbf{p} = (q_x/q_w, q_y/q_w, q_z/q_w, 1)^T$

- A $\mathbf{P}_p$ mátrix
  - A $z = f$-et $+1$-re és a $z = n$-et -1-re képezi le
- A perspektív transzformáció végrehajtása után vágással és homogenizálással kapjuk meg a normalizált eszköz koordinátákat
- Először a $\mathbf{S}(1, 1, -1)$ mátrixszal szorzunk
  - A közeli és távoli értékek pozitív értékek lesznek

- A normalizált eszköz koordináták
  - A vágási koordináták homogén formában vannak tárolva
  - x és y értékekre van szükségünk mélységi értékekkel
  - Perspektív osztás
    - w-vel elosztjuk x, y és z értékeket
  - A geometriai adatok egy egységkockán belül lesznek

- Ablak koordináták
  - A normalizált vertex koordinátákat átkonvertáljuk a végső koordináta rendszerbe
  - x és y pixel pozíciókat használunk
  - Nézeti ablak
    - Adott ablakon belüli tér meghatározása
    - aktuális képernyő koordinátákban
  - A raszterizáló a vertexekből pontokat, vonalakat és poligonokat állít elő
- Mélység-távolság transzformáció
  - A vertexek z értékeit skálázza be a mélységpuffer értékeinek az intervallumába

- A vertex programok vertex pozíciókat kapnak az objektum térben
- A program mindegyik vertex-et megszorozza a modell-nézeti és vetületi mátrix-szal, mely a vágási területre viszi a vertex-eket
- Gyakorlatban két mátrix konkatenációja
    - Egy mátrix szorzás



Példa

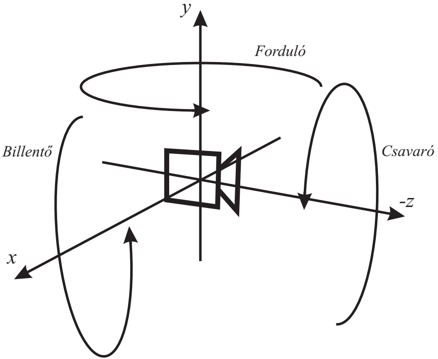
```
void C4E1v_transform(float4 position   : POSITION,
                     out float4 oPosition : POSITION,
                     out float4 color     : COLOR,
                     uniform float4x4 modelViewProj)
{
// A pozíció transzformálása
// az objektumérből a vágótérbe

  oPosition = mul(modelViewProj, position);

  color = float4(0.8, 0.8, 0.8, 1.0);
}
```

Speciális transzformációk

- Egy tetszőleges forgatás megadására
- Alkalmas egy kamera irányítottság beállítására
- A kamera az origóban, z tengely negatív irányába néz, a felfele mutató irány az y tengely pozitív irányába mutat
- Három mátrix szorzata
  - $\mathbf{E}(h, p, r) = \mathbf{R}_z(r)\mathbf{R}_x(p)\mathbf{R}_y(h)$



- Az **E** forgatások összefűzésével áll elő
  - Ortogonális
- Inverze könnyen kiszámítható
  - $\mathbf{E}^{-1} = \mathbf{E}^T = (\mathbf{R}_z\mathbf{R}_x\mathbf{R}_y)^T = \mathbf{R}_y^T\mathbf{R}_x^T\mathbf{R}_z^T$
  - Az **E** mátrix transzponáltja egyszerűbb
- Euler szögek
  - (h, p és r)

- Gimbal lock
  - A forgatások miatt egy szabadsági fokot elveszítünk
    - $h = 0$, $p = 90^\circ$
    - Ezután a z-tengely körüli forgatás lényegében az y-tengely körüli forgatásnak felel meg
  - Nem tudunk a z világtengely körül forgatni

$$\mathbf{E}(h, \pi/2, r) = \begin{pmatrix} \cos r \cos h - \sin r \sin h & 0 & \cos r \sin h + \sin r cosh \\ \sin r \cos h + \cos r \sin h & 0 & \sin r \sin h - \cos r \cos h \\ 0 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \cos(r+h) & 0 & \sin(r+h) \\ \sin(r+h) & 0 & \cos(r+h) \\ 0 & 1 & 0 \end{pmatrix}$$

- Feladat: a h, p és r Euler paraméterek kinyerése az ortogonális mátrixból

$$\mathbf{F} = \begin{pmatrix} f_{00} & f_{01} & f_{02} \\ f_{10} & f_{11} & f_{12} \\ f_{20} & f_{21} & f_{22} \end{pmatrix} = \mathbf{R}_z(r)\mathbf{R}_x(p)\mathbf{R}_y(h) = \mathbf{E}(r, p, h)$$

$$\mathbf{F} = \begin{pmatrix} \cos r \cos h - \sin r \sin p \sin h & -\sin r \cos p & \cos r \sin h + \sin r \sin p \cos h \\ \sin r \cos h + \cos r \sin p \sin h & \cos r \cos p & \sin r \sin h - \cos r \sin p \cos h \\ -\cos p \sin h & \sin p & \cos p \cos h \end{pmatrix}$$

- $\cos p \neq 0$

$$h = \arctan(-\frac{f_{20}}{f_{22}}),$$
$$p = \arcsin(f_{21}),$$
$$r = \arctan(-\frac{f_{01}}{f_{11}}).$$

- $\cos p = 0$
  - $f_{01} = f_{11} = 0$
  - $\sin p = \pm 1$

$$\mathbf{F} = \begin{pmatrix} \cos(r \pm h)) & 0 & \sin(r \pm h) \\ \sin(r \pm h) & 0 & \cos(r \pm h) \\ 0 & \pm 1 & 0 \end{pmatrix}$$

- A többi paramétert $h = 0$-ra
  - $\sin r / \cos r = \tan r = f_{10}/f_{00}$
  - $r = \arctan(f_{10}/f_{00})$

- Különböző transzformációk paramétereinek meghatározása
  - Transzformációk összefűzésével előállított mátrixból
- Különböző esetek
  - Csak a skálázási paramétereket nyerjük ki
  - Meghatározni azt, hogy a modellen csak egy merevtest-transzformációt alkalmaztak-e vagy sem
  - Egy animáció kulcspozíciói közötti interpolálás végrehajtása
  - A nyírások eltávolítása a forgatási mátrixból

- Felbontások
  - Eltolás és forgatási mátrix származtatása
  - Triviális eltolási értékek kinyerése merevtest-transzformáció esetén
    - Ortogonális mátrix utolsó oszlopa
  - A mátrix determinánsa meghatározza, hogy végrehajtottak-e tükrözést
- A forgatás, skálázás és nyírás szétválasztása komolyabb erőfeszítést igényel

- **r** forgatási tengely normalizált
- A transzformáció α szöggel forgat **r** körül
- Találni kell két másik egység hosszú tetszőleges tengelyt
  - Egymásra és **r**-rel ortogonálisak (merőlegesek), azaz ortonormáltak
  - Ezek bázist alkotnak
- Ötlet
  - A bázist változtassuk meg az alapról az új bázisra
  - Forgassuk el az adott objektumot α szöggel mondjuk az x-tengely körül
  - Transzformáljunk vissza az alap bázisba

- Számítsuk ki a bázis ortonormált tengelyeit
  - Az első tengely az **r**
- A második **s**-tengely megkeresése
  - $\mathbf{t} = \mathbf{r} \times \mathbf{s}$
  - Numerikusan stabil módszer
    - **r** abszolút értékben legkisebb komponensének 0-ra állítása
    - A két másik komponens megcserélése után negáljuk az elsőt ezek közül

- Az r, s és t vektort helyezzük el egy mátrix soraiban

$$\mathbf{M} = \begin{pmatrix} \mathbf{r}^T \\ \mathbf{s}^T \\ \mathbf{t}^T \end{pmatrix}$$

- A mátrix az **r** vektort az x-tengelybe, az **s** vektort az y-tengelybe és a **t** vektort a z-tengelybe transzformálja

$$\mathbf{X} = \mathbf{M}^T \mathbf{R}_x(\alpha)\mathbf{M}. \tag{1}$$

- Goldman módszere
  - Tetszőleges normalizált **r**-tengely körüli forgatásra ϕ szöggel

$$\mathbf{R} = \begin{pmatrix} \cos\phi + (1-\cos\phi)r_x^2 & (1-\cos\phi)r_x r_y - r_z \sin\phi & (1-\cos\phi)r_x r_z + r_y \sin\phi \\ (1-\cos\phi)r_x r_y + r_z \sin\phi & \cos\phi + (1-\cos\phi)r_y^2 & (1-\cos\phi)r_y r_z - r_x \sin\phi \\ (1-\cos\phi)r_x r_z - r_y \sin\phi & (1-\cos\phi)r_y r_z + r_x \sin\phi & \cos\phi + (1-\cos\phi)r_z^2 \end{pmatrix}$$

Kvaterniók

Kvaterniók

- A kvaterniókat vektorként ábrázoljuk és $\hat{\mathbf{q}}$-val jelöljük
- A $\hat{\mathbf{q}}$ kvaterniót a következőképpen definiáljuk

$$\hat{\mathbf{q}} = (\mathbf{q}_v, q_w) = iq_x + jq_y + kq_z + q_w = \mathbf{q}_v + q_w$$
$$\mathbf{q}_v = iq_x + jq_y + kq_z = (q_x, q_y, q_z)$$
$$i^2 = j^2 = k^2 = -1, jk = -kj = i, ki = -ik = j, ij = -ji = k$$

  - $\hat{\mathbf{q}}$ kvaternió valós része q_w
  - A képzetes része $\mathbf{q}_v$, ahol i, j és k a képzetes egységek
- q_v képzetes részen értelmezve van az összes vektor művelet
  - Például a skálázás, skaláris szorzat és kereszt szorzat

- Két $\hat{\mathbf{q}}$ és $\hat{\mathbf{r}}$ szorzatát a következőképpen tudjuk levezetni
  - A képzetes egységek szorzása nem kommutatív

$$
\begin{aligned}
\hat{\mathbf{q}}\hat{\mathbf{r}} &= (iq_x + jq_y + kq_z + q_w)(ir_x + jr_y + kr_z + r_w) \\
&= i(q_y r_z - q_z r_y + r_w q_x + q_w r_x) \\
&+ j(q_z r_x - q_x r_z + r_w q_y + q_w r_y) \\
&+ k(q_x r_y - q_y r_x + r_w q_z + q_w r_z) \\
&+ q_w r_w - q_x r_x - q_y r_y - q_z r_z = \\
&= (\mathbf{q}_v \times \mathbf{r}_v + r_w \mathbf{q}_v + q_w \mathbf{r}_v), q_w r_w - \mathbf{q}_v \cdot \mathbf{r}_v
\end{aligned}
$$

Összeadás: $\hat{\mathbf{q}} + \hat{\mathbf{r}} = (\mathbf{q}_v + \mathbf{r}_v, q_w + r_w)$

Konjugált: $\hat{\mathbf{q}}^* = (\mathbf{q}_v, q_w)^* = (-\mathbf{q}_v, q_w)$

Norma: $n(\hat{\mathbf{q}}) = \sqrt{\hat{\mathbf{q}}\hat{\mathbf{q}}^*} = \sqrt{\hat{\mathbf{q}}^*\hat{\mathbf{q}}} = \sqrt{\mathbf{q}_v \cdot \mathbf{q}_v + q_w^2} = \sqrt{q_x^2 + q_y^2 + q_z^2 + q_w^2}$

Egység: $\hat{\mathbf{i}} = (\mathbf{0}, 1)$

- Multiplikatív inverz esetén
  - $\hat{\mathbf{q}}^{-1}\hat{\mathbf{q}} = \hat{\mathbf{q}}\hat{\mathbf{q}}^{-1} = 1$

$$n(\hat{\mathbf{q}})^2 = \hat{\mathbf{q}}\hat{\mathbf{q}}^*$$

$$\Longleftrightarrow$$

$$\frac{\hat{\mathbf{q}}\hat{\mathbf{q}}^*}{n(\hat{\mathbf{q}})^2} = 1$$

  - $\hat{\mathbf{q}}^{-1} = \frac{1}{n(\hat{\mathbf{q}})^2}\hat{\mathbf{q}}^*$

Konjugálási szabályok

$$(\hat{\mathbf{q}}^*)^* = \hat{\mathbf{q}}$$
$$(\hat{\mathbf{q}} + \hat{\mathbf{r}})^* = \hat{\mathbf{q}}^* + \hat{\mathbf{r}}^*$$
$$(\hat{\mathbf{q}}\hat{\mathbf{r}})^* = \hat{\mathbf{r}}^*\hat{\mathbf{q}}^*$$

Normálási szabályok

$$n(\hat{\mathbf{q}}^*) = n(\hat{\mathbf{q}})$$
$$n(\hat{\mathbf{q}}\hat{\mathbf{r}}) = n(\hat{\mathbf{q}})n(\hat{\mathbf{r}})$$

Linearitás

$$\hat{\mathbf{p}}(s\hat{\mathbf{q}} + t\hat{\mathbf{r}}) = s\hat{\mathbf{p}}\hat{\mathbf{q}} + t\hat{\mathbf{p}}\hat{\mathbf{r}}$$
$$(s\hat{\mathbf{p}} + t\hat{\mathbf{q}})\hat{\mathbf{r}} = s\hat{\mathbf{p}}\hat{\mathbf{r}} + t\hat{\mathbf{q}}\hat{\mathbf{r}}$$

Asszocivitás

$$\hat{\mathbf{p}}(\hat{\mathbf{q}}\hat{\mathbf{r}}) = (\hat{\mathbf{p}}\hat{\mathbf{q}})\hat{\mathbf{r}}$$

- Egységkvaternió $\hat{\mathbf{q}} = (\mathbf{q}_v, q_w)$ normája 1-gyel egyenlő $(n(\hat{\mathbf{q}}) = 1)$
- $\hat{\mathbf{q}}$ megadható
  - $\hat{\mathbf{q}} = (\sin\phi \mathbf{u}_q, \cos\phi) = \sin\phi \; \mathbf{u}_q + \cos\phi$
    - $\mathbf{u}_q$ egy három-dimenziós vektor
    - $\|\mathbf{u}_q\| = 1$

$$
\begin{aligned}
n(\hat{\mathbf{q}}) = n(\sin\phi \mathbf{u}_q, \cos\phi) &= \sqrt{\sin^2\phi(\mathbf{u}_q \cdot \mathbf{u}_q) + \cos^2\phi} \\
&= \sqrt{\sin^2\phi + \cos^2\phi} = 1
\end{aligned}
$$

- akkor és csak akkor, ha $\mathbf{u}_q \cdot \mathbf{u}_q = 1 = \|\mathbf{u}_q\|^2$

- Komplex számok esetén
  - Egy két dimenziós egység vektor megadható $\cos\phi + i\sin\phi = e^{i\phi}$ formában
  - Alkalmazható kvaterniók esetén is
    - $\hat{\mathbf{q}} = \sin\phi\ \mathbf{u}_q + \cos\phi = e^{\phi\mathbf{u}_q}$
- A logaritmus és hatvány függvények ezek alapján

Logaritmus:

$$\log\hat{\mathbf{q}} = \log(e^{\phi\mathbf{u}_q}) = \phi\mathbf{u}_q$$

Hatvány:

$$\hat{\mathbf{q}}^t = \left(e^{\phi\mathbf{u}_q}\right)^t = e^{\phi t\mathbf{u}_q} = \sin(\phi t)\ \mathbf{u}_q + \cos(\phi t)$$

- Egység hosszú kvaterniók (egységkvaterniók) tanulmányozása
  - Az egységkvaterniók egy három dimenziós forgatást fejezhetnek ki
- Egy pont vagy egy vektor négy koordinátáját $\mathbf{p} = (p_x, p_y, p_z, p_w)^T$ helyezzük el egy $\hat{\mathbf{p}}$ kvaternió komponenseibe
  - Tfh. van egy egységkvaterniónk $\hat{\mathbf{q}} = (\sin\phi\ \mathbf{u}_q, \cos\phi)$
  - Ekkor $\hat{\mathbf{q}}\hat{\mathbf{p}}\hat{\mathbf{q}}^{-1}$
    - $\hat{\mathbf{p}}$-t forgatja el (és így $\mathbf{p}$-t is) az $\mathbf{u}_q$-tengely körül 2ϕ szöggel
    - Ezt a forgatást bármely tengely körüli forgatás esetén használhatjuk
  - $\hat{\mathbf{q}}$ és $\hat{\mathbf{q}}^{-1}$ ugyanazt a forgatást hajtja végre
    - Egy mátrixból való kvaternió kinyerése akár $\hat{\mathbf{q}}$-val vagy $-\hat{\mathbf{q}}$-val is visszatérhet

- Adott $\hat{\mathbf{q}}$ és $\hat{\mathbf{r}}$ két egységkvaternió
- Két kvaternióval $\hat{\mathbf{p}}$-n végrehajtott összetett transzformáció
  - $\hat{\mathbf{r}}(\hat{\mathbf{q}}\hat{\mathbf{p}}\hat{\mathbf{q}}^*)\hat{\mathbf{r}}^* = (\hat{\mathbf{r}}\hat{\mathbf{q}})\hat{\mathbf{p}}(\hat{\mathbf{r}}\hat{\mathbf{q}})^* = \hat{\mathbf{c}}\hat{\mathbf{p}}\hat{\mathbf{c}}^*$
    - ahol $\hat{\mathbf{c}} = \hat{\mathbf{r}}\hat{\mathbf{q}}$ szintén egy egységkvaternió, amelyet $\hat{\mathbf{q}}$ és $\hat{\mathbf{r}}$ kvaterniók összefűzésével kapunk meg

- A mátrixszorzás egyes rendszerekben hardveresen támogatott
    - Hatékonyabb ilyen módon elvégezni a szorzásokat, mint az előzőekben megadott formában
- Szükség van a kvaterniók mátrix formába való átalakítása mindkét irányba

- Egy $\hat{\mathbf{q}}$ kvaternió az $\mathbf{M}^q$ mátrixba a következőképpen alakítható át

$$\mathbf{M}^q = \begin{pmatrix} 1 - s(q_y^2 + q_z^2) & s(q_x q_y - q_w q_z) & s(q_x q_z + q_w q_y) & 0 \\ s(q_x q_y + q_w q_z) & 1 - s(q_x^2 + q_z^2) & s(q_y q_z - q_w q_x) & 0 \\ s(q_x q_z - q_w q_y) & s(q_y q_z + q_w q_x) & 1 - s(q_x^2 + q_y^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

  - ahol $s = 2/n(\hat{\mathbf{q}})$
- Egységkvaterniók esetén ez a következőképpen egyszerűsödik:

$$\mathbf{M}^q = \begin{pmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) & 0 \\ 2(q_x q_y + q_w q_z) & 1 - s(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) & 0 \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Nincs szükség trigonometrikus függvények használatára

- Fordított irányú átalakítás egy kicsit bonyolultabb
- Mátrix elemekből előállított különbségek

$$\begin{aligned} m^q_{21} - m^q_{12} &= 4q_w q_x, \\ m^q_{02} - m^q_{20} &= 4q_w q_y, \\ m^q_{10} - m^q_{01} &= 4q_w q_z. \end{aligned}$$

- Ha q_w-t ismerjük, akkor a $\mathbf{v}_q$ és így $\hat{\mathbf{q}}$ kiszámítható

- A mátrix nyoma

$$\mathrm{tr}(\mathbf{M}^{\mathbf{q}}) = 4 - 2s(q_x^2 + q_y^2 + q_z^2) = 4\left(1 - \frac{q_x^2 + q_y^2 + q_z^2}{q_x^2 + q_y^2 + q_z^2 + q_w^2}\right)$$
$$= \frac{4q_w^2}{q_x^2 + q_y^2 + q_z^2 + q_w^2} = \frac{4q_w^2}{n(\hat{\mathbf{q}})}$$

- Ebből következik

$$q_w = \tfrac{1}{2}\sqrt{\mathrm{tr}(\mathbf{M}^q)} \qquad q_x = \tfrac{m_{21}^q - m_{12}^q}{4q_w}$$
$$q_y = \tfrac{m_{02}^q - m_{20}^q}{4q_w} \qquad q_z = \tfrac{m_{10}^q - m_{01}^q}{4q_w}$$

- Numerikusan stabil eljárás
  - Kis számokkal való osztások elkerülése
- Legyen $t = q - w^2 - q_x^2 - q_y^2 - q_z^2$

$$
\begin{aligned}
m_{00} &= t + 2q_x^2, \\
m_{11} &= t + 2q_y^2, \\
m_{22} &= t + 2q_z^2, \\
u &= m_{00} + m_{11} + m_{22} = t + 2q_w^2,
\end{aligned}
$$

  - Melyik a legnagyobb a q_x, q_y, q_z és q_w közül
    - m_{00}, m_{11}, m_{22} és u alapján meghatározhatjuk
  - q_w esetén előző fólia

- Ellenkező esetben

$$
\begin{aligned}
4q_x^2 &= +m_{00} - m_{11} - m_{22} + m_{33} \\
4q_y^2 &= -m_{00} + m_{11} - m_{22} + m_{33} \\
4q_z^2 &= -m_{00} - m_{11} + m_{22} + m_{33} \\
4q_w^2 &= \mathrm{tr}(\mathbf{M}^q).
\end{aligned}
$$

  - Mátrix elemekből előállított különbségek segítségével a maradék $\hat{\mathbf{q}}$ komponenseket is meg lehet határozni

- Olyan művelet, amelyet $\hat{\mathbf{q}}$ és $\hat{\mathbf{r}}$ egységkvaternió és egy $t \in [0,1]$ paraméter esetén egy interpolált kvaterniót számít ki
  - Hasznos objektumok animálásánál
  - Kevésbé hasznos kamera irányítottságok interpolálásakor
    - A kamera felfele mutató vektor megdőlhet az interpolálás alatt
- Algebrai formája egy $\hat{\mathbf{s}}$ összetett kvaternió
  - $\hat{\mathbf{s}}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t) = (\hat{\mathbf{r}}\hat{\mathbf{q}}^{-1})^t \hat{\mathbf{q}}$
- Szoftveres megvalósításkor
  - $\hat{\mathbf{s}}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t) = \mathtt{slerp}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t) = \frac{\sin(\phi(q-t))}{\sin\phi}\hat{\mathbf{q}} + \frac{\sin\phi t}{\sin\phi}\hat{\mathbf{r}}$

- ϕ kiszámításához $\cos\phi = q_x r_x + q_y r_y + q_z r_z + q_w r_w$ összefüggést használjuk fel
    - $t \in [0, 1]$ -re a `slerp` függvény egyedi interpolált kvaterniókat számít ki amelyek együtt a legrövidebb ívet alkotják egy négydimenziós egységgömbön $\hat{\mathbf{q}}(t = 0)$-tól $\hat{\mathbf{r}}(t = 1)$-ig
    - Az ív a körön helyezkedik el
        - A $\hat{\mathbf{q}}$, $\hat{\mathbf{r}}$ és az origó által meghatározott sík és a négy dimenziós egységgömb metszeteként alakul ki

- A `slerp` függvény tökéletesen alkalmas két orientáció/irányítottság közötti interpolációra
- Euler transzformációval elvégezni nem olyan egyszerű
  - Több Euler szög interpolálásakor gimbal lock állhat elő
- Kettőnél több orientáció esetén
  - $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_{n-1}$
  - Amennyiben a `slerp` függvényt alkalmazzuk minden szakaszon, akkor hirtelen irányváltás látható az orientáció interpolálásakor

- Valamilyen spline-t kell használunk interpoláláskor
    - Vezessük be $\hat{\mathbf{a}}_i$ és $\hat{\mathbf{b}}_{i+1}$ kvaterniókat $\hat{\mathbf{q}}_i$ és $\hat{\mathbf{q}}_{i+1}$ között
    - Interpoláció során áthaladunk a kezdeti $\hat{\mathbf{q}}_i, i \in [0, \ldots, n-1]$ orientációkon, de az $\hat{\mathbf{a}}_i$-ken nem
    - Arra használjuk, hogy jelezzük az érintőleges irányítottságokat az eredeti orientációknál

$$\hat{\mathbf{a}}_i = \hat{\mathbf{b}}_i = \hat{\mathbf{q}}_i \exp\left[-\frac{\log(\hat{\mathbf{q}}_i^{-1}\hat{\mathbf{q}}_{i-1}) + \log(\hat{\mathbf{q}}_i^{-1}\hat{\mathbf{q}}_{i+1})}{4}\right]$$

- A $\hat{\mathbf{q}}_i$-t, $\hat{\mathbf{a}}_i$-t és $\hat{\mathbf{b}}_i$-t a kvaterniók gömbi interpolációjára használjuk sima köbös spline használatával a következő egyenlet szerint

$$\begin{aligned}&\texttt{squad}(\hat{\mathbf{q}}_i, \hat{\mathbf{q}}_{i+1}, \hat{\mathbf{a}}_i, \hat{\mathbf{a}}_{i+1}, t) = \\ &\texttt{slerp}(\texttt{slerp}(\hat{\mathbf{q}}_i, \hat{\mathbf{q}}_{i+1}, t), \texttt{slerp}(\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_{i+1}, t), 2t(1-t))\end{aligned}$$

- Egy **s** irányvektort egy **t** irányvektorba forgatunk
  - Normalizáljuk **s**-t és **t**-t
  - Kiszámítjuk az **u** egység forgatási tengelyt
    - $\mathbf{u} = (\mathbf{s} \times \mathbf{t}) / \|\mathbf{s} \times \mathbf{t}\|$ egyenlőség alapján
  - Legyen $e = \mathbf{a} \cdot \mathbf{t} = \cos(2\phi)$ és $\|\mathbf{s} \times \mathbf{t}\| = \sin(2\phi)$
    - ahol 2ϕ az **s** és **t** közötti szög
  - $\hat{\mathbf{q}} = (\mathbf{u} \sin \phi, \cos \phi)$
    - Az adott forgatást hajtja végre **s**-ből **t**-be
  - $\hat{\mathbf{q}} = \left(\frac{\sin \phi}{\sin 2\phi} (\mathbf{s} \times \mathbf{t}), \cos \phi \right)$-t egyszerűsítve

$$\hat{\mathbf{q}} = (\mathbf{q}_v, q_w) = \left(\frac{1}{\sqrt{2(1+e)}} (\mathbf{s} \times \mathbf{t}), \frac{\sqrt{2(1+e)}}{2} \right)$$

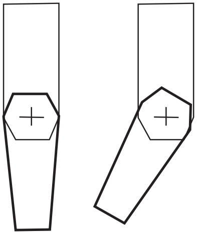
    - A numerikus instabilitást elkerülhetjük
  - Nullával való osztás
    - Ellentétes irányú **t** és **s**
    - Bármely **s**-re merőleges forgatási tengely használható **t** forgatására

- **s**-ből **r**-be való forgatásmátrix ábrázolása

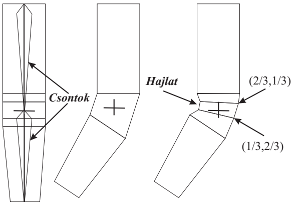
$$\mathbf{R}(\mathbf{s},\mathbf{t}) = \begin{pmatrix} e + hv_x^2 & hv_xv_y - v_z & hv_xv_z + v_y & 0 \\ hv_xv_y + v_z & e + hv_y^2 & hv_yv_z - v_x & 0 \\ hv_xv_z - v_y & hv_yv_z + v_x & e + hv_z^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned} \mathbf{v} &= \mathbf{s} \times \mathbf{t}, \\ e &= \cos(2\phi) = \mathbf{s} \cdot \mathbf{t}, \\ h &= \frac{1 - \cos 2\phi}{\sin^2(2\phi)} = \frac{1 - e}{\mathbf{v} \cdot \mathbf{v}} = \frac{1}{1 + e}. \end{aligned}$$

- Egy digitális karakter animálásakor az alkarját és felkarját mozgatjuk
- Merevtest-transzformáció
  - Nem hasonlít a két rész kapcsolódása a könyökre
- Csontokat definiálnak
  - Bőr vesz körül és a bőr a csontok mozgásának megfelelően változik
- Az al- és a felkart elkülönülve animáljuk
  - A két rész kapcsolódásánál egy rugalmas bőrrel kapcsoljuk össze a részeket



Merevtest-transzformáció



Vertex keveredés

Vertex keveredés

- A rugalmas rész egy vertex halmazát a felkar
- A másik vertex halmazt az alkar mátrixszal transzformáljuk
- Azok a háromszögek, amelyek vertexeit különböző mátrixokkal transzformáltuk, a transzformáció következtében megnyúlnak illetve összemennek

Vertex keveredés

- Megengedhetjük egy egyszerű vertex esetén
  - Több különböző mátrixszal transzformáljuk
  - Az eredményeket súlyozzuk és keverjük
  - $\mathbf{u}(t) = \sum_{i=0}^{n-1} w_i \mathbf{B}_i(t) \mathbf{M}_i^{-1} \mathbf{p}$,
    - ahol $\sum_{i=0}^{n-1} w_i = 1$, $w_i \geq 0$
    - $\mathbf{p}$ az eredeti vertex
    - $\mathbf{u}(t)$ transzformált vertex
    - $\mathbf{M}_i$ mátrix az iniciális koordinátarendszerből a világ koordináta-rendszerbe transzformál
    - $\mathbf{B}_i(t)$ az i-ik csont világ transzformációja
    - w_i az i-ik csont súlya a $\mathbf{p}$ vertex esetén

Vertex keveredés

- Az $\mathbf{M}_i$ mátrix inverze modelltérből a csont saját terébe transzformál
- A $\mathbf{B}_i(t)$ csont aktuális transzformációjával vissza a modelltérbe transzformál
- A gyakorlatban a $\mathbf{B}_i(t)$ és az $\mathbf{M}_i^{-1}$ mátrixokat összefűzzük
  - Mindegyik csont és az animáció mindegyik képkockája esetén
  - Az eredménymátrixot használjuk a vertex transzformáció végrehajtására

Összefoglalás

Miről volt szó?

- Transzformációs csővezeték
- Speciális transzformációk
  - Euler transzformáció
  - Paraméterek kinyerése az Euler transzformációból
  - Mátrix felbontás
  - Forgatás tetszőleges tengely mentén
- Kvaterniók
- Vertex keveredés