Trasformazioni geometriche

Prodotto scalare

$$a \cdot b = \sum_{i=1}^{n} a_i b_i = \|a\| \, \|b\| \cos\theta$$

- proiezione di b su a: $b \cdot \frac{a}{\|a\|}$
- $a \cdot b = 0$ quando $a \perp b$
- $\theta = \arccos\left(\frac{a \cdot b}{\|a\| \|b\|}\right)$

Prodotto vettoriale

$$a \times b = \det \begin{pmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{pmatrix}$$

- $\|a \times b\| = \|a\| \, \|b\| \sin\theta$, area del parallelogramma definito da a e b
- anticommutativo ($a \times b = -b \times a$), distribuisce su somma
- $a \times b$ è perpendicolare ad a e b

Area del triangolo:

$$\mathrm{area}(p_0, p_1, p_2) = \frac{\|(p_1 - p_0) \times (p_2 - p_0)\|}{2}$$

Coordinate polari

- $\rho = \sqrt{x^2 + y^2}$, $\theta = \mathrm{atan2}(y, x)$
- $x = \rho \cos\theta$, $y = \rho \sin\theta$

Coordinate omogenee

I vettori hanno un'unica rappresentazione:

$$v = \begin{pmatrix} v_x \\ v_y \\ 0 \end{pmatrix}$$

Forma canonica di un punto:

$$p = \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

e $\forall \lambda \neq 0 \,.\, \lambda p = p$. Utile per le proiezioni in prospettiva: i punti sulla stessa linea vengono proiettati sullo stesso.

Trasformazioni affini

Una trasformazione è affine se e solo se:

- conserva la collinearità dei punti;
- conserva il rapporto tra le lunghezze di segmenti paralleli;
- conserva il parallelismo.

Composizioni di trasformazioni lineari e traslazioni. Traslazione, scalatura e rotazione sono affini. Le trasformazioni con matrice *invertibile* ($\det \neq 0$)

$$\begin{pmatrix} a_{xx} & a_{xy} & v_x \\ a_{yx} & a_{yy} & v_y \\ 0 & 0 & 1 \end{pmatrix}$$

sono affini.

$$\begin{pmatrix} a_{xx} & a_{xy} & v_x \\ a_{yx} & a_{yy} & v_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{pmatrix} p + v \\ 1 \end{pmatrix}$$

Inversa (dimostrare):

$$\begin{pmatrix} M & t \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} M^{-1} & -M^{-1}t \\ 0 & 1 \end{pmatrix}$$

Trasformazione	**lunghezze**	**angoli**	**rapporti**	**collinearità**
traslazione	sì	sì	sì	sì
rotazione	sì	sì	sì	sì
scalatura isotropica	no	sì	sì	sì
scalatura anisotropica	no	no	sì	sì
shearing	no	no	sì	sì
proiezione ortogonale	no	no	sì	sì
trasformazione affine	no	no	sì	sì
proiezione in prospettiva	no	no	no	sì

Sistemi di riferimento

Origine (punto) e 2/3 assi (vettori)

- frame canonico C:

$$o = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad u = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad v = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

- conversione delle coordinate da $F = (o, u, v)$ a C: $p = o + xu + yv$, matrice:

$$F = \begin{pmatrix} u_x & v_x & o_x \\ u_y & v_y & o_y \\ 0 & 0 & 1 \end{pmatrix}.$$

  Trasformazione affine (con traslazione di o). La stessa operazione si può vedere come trasformare C in F.

- da F_0 a F_1:

$$p_1 = F_1^{-1} p = F_1^{-1} F_0 p_0$$

- gerarchia di frame: ogni frame espresso rispetto al padre, DAG (implementato come stack di matrici).

Traslazione

$$T_v(p) = p + v$$

Matrice (coordinate omogenee):

$$\begin{pmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{pmatrix}$$

Con questa matrice i vettori non vengono modificati, visto che rappresentano uno spostamento.

Scalatura

$$S_{s_x,s_y}(p) = \begin{pmatrix} s_x p_x \\ s_y p_y \end{pmatrix}$$

Matrice:

$$\begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$$

- se $s_x = s_y$ allora scalatura uniforme/*isotropica*, altrimenti non uniforme/*anisotropica*
- inversa: $S_{s_x^{-1}, s_y^{-1}}$

Shear

Shear su x:

$$Sh_{k,y}(p) = \begin{pmatrix} p_x + kp_y \\ p_y \end{pmatrix}$$

Matrice:

$$\begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$$

su y:

$$\begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix}$$

- traslazione lungo un asse proporzionalmente al valore dell'altro
- $k = \cot \theta$
- inversa: Sh_{-k}

Rotazione

In senso antiorario di angolo α:

$$R_\alpha(p) = \begin{pmatrix} \rho \cos(\theta + \alpha) \\ \rho \sin(\theta + \alpha) \end{pmatrix} = \begin{pmatrix} p_x \cos \alpha - p_y \sin \alpha \\ p_x \sin \alpha + p_y \cos \alpha \end{pmatrix}$$

Matrice:

$$\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

Intorno ad un punto generico:

$$R_{\alpha,c} = T_c R_\alpha T_{-c} = \begin{pmatrix} R'_\alpha & (I - R'_\alpha)c \\ 0 & 1 \end{pmatrix},$$

dove $R_{\alpha,c}, T_{\pm c}, R_\alpha$ sono in coordinate omogenee, R'_α no.

R_α è ortonormale (righe/colonne di norma 1 e ortogonali), quindi $R_\alpha^{-1} = R_\alpha^{\mathrm{t}}$.

In 3D

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_y(\alpha) = \begin{pmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Intorno all'asse $o_r + dz_r$: dato un frame F_r con asse z uguale a z_r,

$$R_{o_r,z_r}(\alpha) = F_r R_z(\alpha) F_r^{-1}$$

Per trovare F_r:

- si sceglie $a \neq z_r$, non troppo vicino per garantire stabilità numerica, di solito $a = e_i$ con $i = \arg\min |z_{r,i}|$;
- si calcolano gli altri assi:
$$x_r = \frac{z_r \times a}{\|z_r \times a\|} \qquad y_r = \frac{z_r \times x_r}{\|z_r \times x_r\|}$$
- si usa o_r, o un qualsiasi punto di $o_r + dz_r$, come origine.

Angoli di eulero / gimbal

Alternativa ad asse + angolo:

yaw/imbardata rotazione intorno all'asse y

pitch/beccheggio asse x

roll/rollio asse z

Ciascun anello definisce un frame rispetto a quello precedente. Funzione surgettiva ma non iniettiva (perdita di un grado di libertà – gimbal lock).

Formula di Rodriguez

Per un asse r che attraversa l'origine:

$$p' = p\cos\alpha + (1 - \cos\alpha)(p \cdot r)r + (r \times p)\sin\alpha$$

Quaternioni

I numeri complessi rappresentano rotazioni in 2D:

$$(p_x + ip_y)(\cos\alpha + i\sin\alpha) = R_\alpha p.$$

In 3D, si usano i quaternioni:

$$q = (\underbrace{w}_{s}, \underbrace{x, y, z}_{v}) = w + ix + jy + kz,$$

dove i, j, i sono unità immaginarie tali che:

- $i^2 = j^2 = k^2 = ijk = -1$
- $ij = k$, $ji = -k$
- $jk = i$, $kj = -i$
- $ki = j$, $ik = -j$

regola del cerchio, senso orario + antiorario -.

Operazioni:

- $q_1 + q_2 = (s_1 + s_2, v_1 + v_2)$
- $q_1 q_2 = (s_1 s_2 - v_1 \cdot v_2,\ s_1 v_2 + s_2 v_1 + v_1 \times v_2)$

- $\overline{q} = (s, -v)$
- $|q| = \sqrt{\overline{q}q} = \sqrt{s^2 + v \cdot v}$
- $q^{-1} = \frac{\overline{q}}{|q|^2}$

La rotazione di angolo θ intorno all'asse v si rappresenta con il quaternione unitario

$$q = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}v\right).$$

Per applicarla ad un punto rappresentato come quaternione puro $p = (0, v_p)$,

$$p' = qp\overline{q} = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}v\right)(0, v_p)\left(\cos\frac{\theta}{2}, -\sin\frac{\theta}{2}v\right).$$

La rotazione rappresentata dal quaternione $q = (s, v)$ è:

$$\theta = 2\,\mathrm{atan2}(s, \|v\|) \qquad v_r = \frac{v}{\|v\|}$$

Si possono comporre rotazioni moltiplicando i quaternioni.
I quaternioni sono più compatti, efficienti e stabili delle matrici di rotazione, e sono facili da comporre e interpolare.

Interpolazione di rotazioni

Lineare

L'interpolazione lineare di matrici ($R_t = R_0(1-t) + R_1 t$) non funziona, il risultato non è una matrice di rotazione. Si può normalizzare ma funziona solo per angoli piccoli(?)

Con angolo e asse:

$$\theta_t = \theta_0(1-t) + \theta_1 t \qquad r(t) = \frac{r_0(1-t) + r_1 t}{\|r_0(1-t) + r_1 t\|},$$

ma la velocità di variazione dell'asse non è costante (più veloce al centro).

Slerp: spherical linear interpolation

Per variare l'asse con velocità angolare uniforme:

$$\alpha = \arccos(r_0 \cdot r_1) \qquad r(t) = \frac{\sin(t\alpha)}{\sin\alpha}r_0 + \frac{\sin((1-t)\alpha)}{\sin\alpha}r_1$$

con r_0, r_1 vettori unitari. Funziona anche se sono quaternioni di rotazione.
Si può usare direttamente con due quaternioni unitari, sostituendoli a r_0, r_1.

Rototraslazione

Composizione di traslazioni e rotazioni. Una rotazione *seguita* da traslazione ha la forma:

$$T_v R_\alpha = \begin{pmatrix} R_\alpha & v \\ 0 & 1 \end{pmatrix}$$

Proiezione

Prospettiva

Su un piano parallelo a xy passante per $(0, 0, -d)$, in coordinate omogenee:

$$p' = \begin{pmatrix} -d\frac{p_x}{p_z} \\ -d\frac{p_y}{p_z} \\ -d \\ 1 \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \\ -p_z/d \end{pmatrix} = \begin{pmatrix} dp_x \\ dp_y \\ dp_z \\ -p_z \end{pmatrix}$$

Matrici:

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -\frac{1}{d} & 0\end{pmatrix}\begin{pmatrix}p_x\\ p_y\\ p_z\\ 1\end{pmatrix}=\begin{pmatrix}p_x\\ p_y\\ p_z\\ -p_z/d\end{pmatrix}\qquad\begin{pmatrix}d & 0 & 0 & 0\\ 0 & d & 0 & 0\\ 0 & 0 & d & 0\\ 0 & 0 & -1 & 0\end{pmatrix}\begin{pmatrix}p_x\\ p_y\\ p_z\\ 1\end{pmatrix}=\begin{pmatrix}dp_x\\ dp_y\\ dp_z\\ -p_z\end{pmatrix}$$

La proiezione in prospettiva non è né affine né invertibile. Non preserva angoli e rapporti delle distanze, ma conserva il parallelismo.

- oggetti più vicini diventano più grandi;
- gli oggetti più lontani del piano vengono rimpiccioliti, quelli più vicini ingranditi (ma nella pratica sono rimossi perché il piano di proiezione corrisponde con il near);
- valori piccoli per d producono un effetto grandangolare, maggiore è d più il risultato si avvicina ad una proiezione ortogonale.

Matrice di OpenGL:

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

Trasforma il volume di vista (tronco di piramide) nel clip space. La proiezione sul near plane diventa una semplice proiezione ortogonale, visto che i proiettori ora sono paralleli, ed è effettuata automaticamente.

Ortogonale

Si sostituisce la coordinata z con $-d$:

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & -d\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}p_x\\ p_y\\ p_z\\ 1\end{pmatrix}=\begin{pmatrix}p_x\\ p_y\\ -d\\ 1\end{pmatrix}$$

Proiettori paralleli.

Matrice di OpenGL:

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

Trasforma il volume di vista (parallelepipedo) nel clip space. L'eliminazione della z per proiettare sul near plane è automatica.