Grafica Computazionale

GUI: trackball

Fabio Ganovelli

fabio.ganovelli@isti.cnr.it

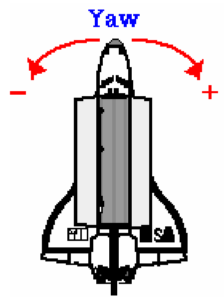
a.a. 2005-2006

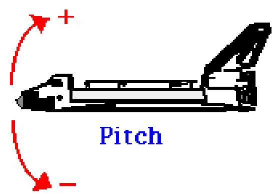
Interfacce di rotazione

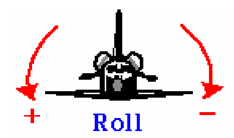
- Come può un utente specificare una rotazione tramite un interfaccia?
- Due modalità:
  - Diretta: specifica valori numerici esatti
  - Interattiva: tramite movimenti del mouse
- Come rappresento una rotazione?
  - Euler Angle
  - Axis/angle
  - Quaternions

Euler Angle

- Una rotazione viene espressa come una serie di tre rotazioni sui tre assi.
- Deriva dal modo con cui si descrive l'orientamento di un aereo
  - Yaw
  - Pitch
  - Roll
- Intuitivo per piccoli valori di pitch e roll

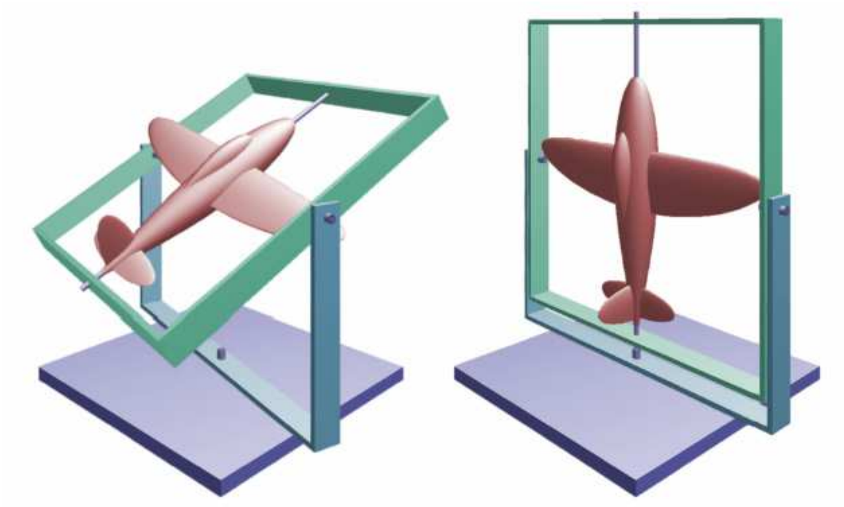






Euler Angle

- Problema ordine rotazione
  - Il risultato dipende dall'ordine in cui faccio le tre rotazioni
- Problema Gimbal Lock
  - In alcune situazioni le rotazioni fatte su un asse possono coprire quelle su un altro asse
  - Se il pitch è a 90 gradi yaw e roll si possono annullare a vicenda.

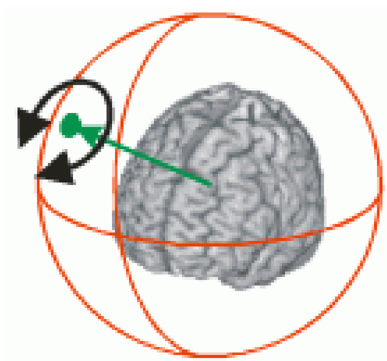


Gimbal lock nelle interfacce

- Capita ad esempio quando cerco di far specificare gli euler angle interattivamente all'utente:
  - Up/down: rot asse x
  - Left/right: rot asse y
  - Pgup/pgdn: rot asse z
- Si incarta.

Axis/angle

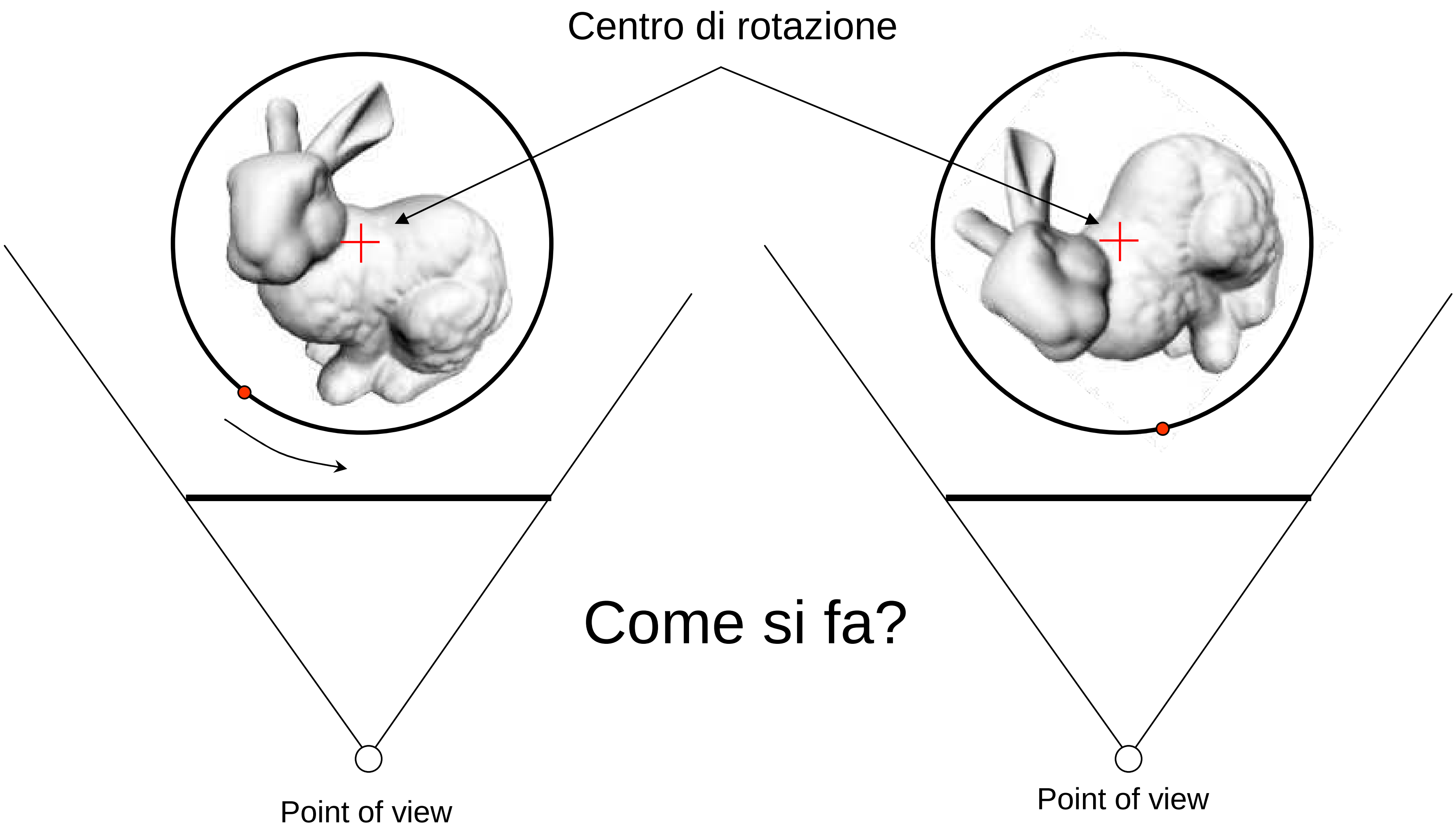
- Asse di rotazione + angolo (come il comando OpenGL/SoftOgl)
  - Si specifica un'asse di rotazione e un angolo di rotazione
  - Molto generico
  - Poco intuitivo
    - Qual'e l'asse di rotazione per girare la testa in modo da guardare in basso a destra?



Trackball

- Si immagina una sfera solidale con la scena
- Ruotando la sfera si ruota la scena
- La rotazione della sfera è effettuata prendendo un punto sulla superficie e spostandolo
- Esattamente come il dispositivo di input.

Trackball



Come si fa?

Quaternioni

- Cos'è un quaternione?
- Un estensione dei numeri complessi,

$$q = w + \mathbf{i}\,x + \mathbf{j}\,y + \mathbf{k}\,z$$

$$dove \quad \mathbf{i}\cdot\mathbf{i} = \mathbf{j}\cdot\mathbf{j} = \mathbf{k}\cdot\mathbf{k} = \mathbf{i}\cdot\mathbf{j}\cdot\mathbf{k} = -1$$

$$segue: \qquad i \cdot j = k,\ \ j \cdot i = -k$$

$$i \cdot k = j,\ \ j \cdot k = -j$$

$$j \cdot k = i,\ k \cdot j = -i$$

- Spesso rappresentato come una coppia scalare-vettore:

$$q = [w, \mathbf{v}] \quad dove \quad \mathbf{v} = (x, y, z)$$

Quaternioni

- Magnitudo

$$\|q\| = \sqrt{w^2 + x^2 + y^2 + z^2}$$

- Normalizzazione a quaternione unitario

$$q = {}^{q}\!/_{\|q\|}$$

Somma e prodotto

- Dati due quaternioni

$$q_1 = w_1 + x_1 i + y_1 j + z_1 k \quad e \quad q_2 = w_2 + x_2 i + y_2 j + z_2 k$$

$$q_1 = [w_1, \mathbf{v}_1] \; e \; q_2 = [w_2, \mathbf{v}_2] \quad dove \quad \mathbf{v}_1 = (x_1, y_1, z_1) \; e \; \mathbf{v}_2 = (x_2, y_2, z_2)$$

- Vale:

$$q_1 + q_2 = [w_1 + w_2, \quad \mathbf{v}_1 + \mathbf{v}_2]$$

$$q_1 * q_2 = [w_1 w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, \quad \mathbf{v}_1 w_2 + w_1 \mathbf{v}_2 + \mathbf{v}_1 \times \mathbf{v}_2]$$

- Identità
  - somma $\qquad q_{I+} = [0, \quad (0,0,0)]$
  - prodotto $\qquad q_{I*} = [1, \quad (0,0,0)]$

Qualche proprietà

- Quaternione coniugato:

$$q = [w, v]$$
$$\bar{q} = [w, -v]$$

- Inverso di un quaternione (tranne che l'identità additiva):

$$q^{-1} = \frac{\bar{q}}{q^2}$$

- Se la parte reale è 0 il quaternione è detto puro. Se il quaternione è puro vale:

$$q_1 * q_2 = \left[-\mathbf{v}_1 \cdot \mathbf{v}_2, \quad \mathbf{v}_1 \times \mathbf{v}_2\right]$$
$$\bar{q} = -q$$

Quaternioni e rotazioni

- Dato un quaternione unitario

$$q = [w, (v_0, v_1, v_2)]$$

- e un vettore in 3 dimensioni:

  $p = (p_0, p_1, p_2)$ che si estende a quaternione *puro* come

  $p_q = [0, p_0, p_1, p_2]$

- Il vettore:

$$p' = q\ [0, p_0, p_1, p_2]\ \overline{q}$$

- È uguale al vettore p ruotato intorno all'asse v di $2\cos^{-1}(w)$

Dimostrazione

$$\cancel{2}\theta = \cancel{2}\cos^{-1}(w)$$

Poiché p e t sono unitari

$$q = [\cos\theta, p \times t] = [p \cdot t, p \times t]$$

$$\boxed{t_q * \overline{p}_q = [t \cdot p, -t \times p]} = [t \cdot p, p \times t] = q$$

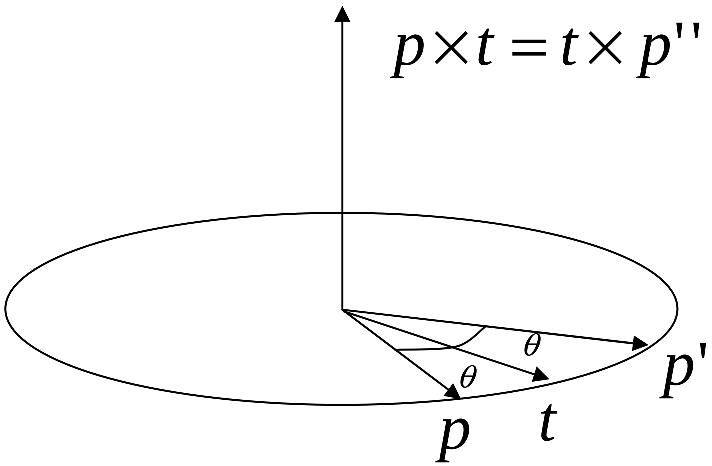
$$q * p_q * \overline{q} = t_q * \boxed{\overline{p}_q * p_q} * \overline{q} = t_q * \boxed{1} * \overline{q}$$

$$\overline{q} = \overline{(p' * \overline{t}_q)} = t_q * \overline{p}'$$

Poiché p è unitario

$$t_q * \overline{q} = t_q * t_q * \overline{p}' = \boxed{-1 * \overline{p}' = p'}$$

$$p \times t = t \times p''$$



Conversioni

- Da quaternione a matrice

$$\begin{bmatrix} 1-2x^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2xy-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{bmatrix}$$

Si ricava espandendo il prodotto: $q * p_q * \overline{q}$

- Da quaternione ad axis/angle

$$q = [w, \mathbf{v}] \quad axis = \mathbf{v} / |\mathbf{v}| \;\; e \;\; angle = 2\arccos(w)$$

Conversioni

- Da axis angle a quaternioni

$$axis = (a_x, a_y, a_z)\ e\ \ angle = \theta$$

$$q = [w, \mathbf{v}]$$

$$\mathbf{v} = \sin(\tfrac{\theta}{2}) \cdot (a_x, a_y, a_z)$$

$$w = \cos(\tfrac{\theta}{2})$$

- Da euler angle a quaternion

$$q_x = [\cos(\tfrac{a}{2}), \quad (\sin(\tfrac{a}{2}), 0, 0)]$$

$$q_y = [\cos(\tfrac{b}{2}), \quad (0, \sin(\tfrac{b}{2}), 0)]$$

$$q_z = [\cos(\tfrac{c}{2}), \quad (0, 0, \sin(\tfrac{c}{2}))]$$

$$q = q_x * q_y * q_z$$

Evitare il gimbal lock

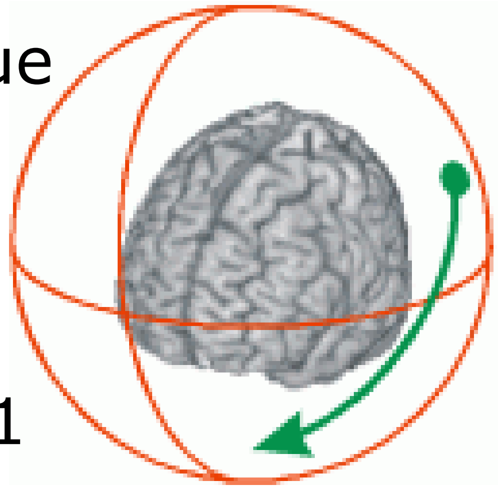
- Euler angle è molto intuitivo per piccole rotazioni: se ruoto di angoli piccoli quello che ottengo è esattamente quello che mi aspetto
- Soluzione:
  - Tenere la rotazione come un quaternione
  - Ad ogni pressione di tasto generare un quaternione corrispondente al piccolo euler angle
  - Ad es. se premo *left* genero un quaternione

$$q_y = [\cos(\tfrac{\delta}{2}), \quad (0, \sin(\tfrac{\delta}{2}), 0)]$$

  - Comporre il risultato con moltiplicazione tra quaternioni e tenere il risultato come base;

Trackball

- Come si mappa il movimento del mouse in una rotazione?
  - Si immagina una sfera circoscritta all’oggetto con cui si vuole interagire
  - Ogni drag del mouse definisce due punti p1 e p2(inizio e fine del drag) sulla sfera
  - Si considera la rotazione che descrive l’arco di cerchio sulla superficie sferica delimitato da p1 e p2



Trackball

- La rotazione così calcolata viene trasformata in un quaternione e composta con la trasf corrente
- Se una volta rilasciato il mouse, si continua comporre con l'ultimo quaternione calcolato, si ottiene l'effetto di spinning.