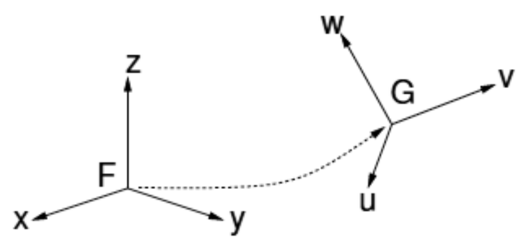
Rotações em 3D

Foley 21.1.3
Notas do Dave (lecture 29)

Descrição dos problemas

1- Como parametrizar rotações 3D?
em animações, para criar um movimento suave.
translações e rotacões 2D são simples,
mas rotação 3D nem tanto.
2- Como parametrizar orientações de frames?
-não confunda orientação de frames com o conceito de orientação no espaço (horário e antihorário).
-orientação de um frame indicam direções no espaço.
-Dado dois frames F e G:
- como representar G em função de F?
- como interpolar, de forma suave, os dois frames?

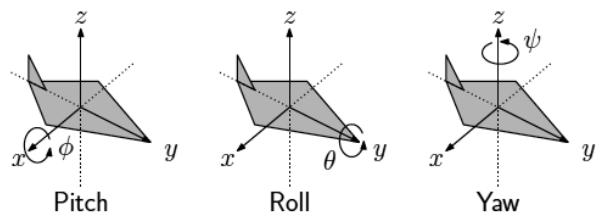
Como mover entre 2 frames



Ângulos de Euler

Leonard Euler, matemático do século XVIII, mostrou que:
combinação de um número qualquer de rotações em 3D pode ser representada por uma única rotação ao redor de um vetor apropriado.
qualquer rotação 3D pode ser dividida por 3 rotações ao redor dos eixos do sistema de coordenadas.

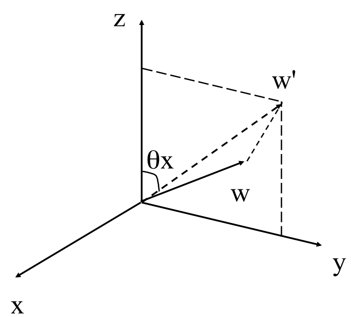
Ângulos de Euler



Processo

- Dado o frame G composto por (u,v,w)
- Frame padrão F = (x,y,z)
- Seja w' a projeção de w sobre o plano yz.
- a) Rode w' ao redor de x até alinhar w' com z (θx)
  note que w estará contido então no plano xz.

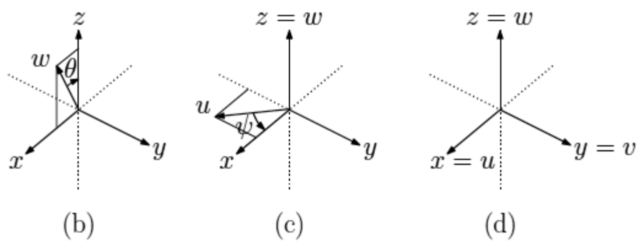
Processo



Processo

b) A seguir rode w ao redor de y (θy), para que ele se alinhe com z.
 note que u e v devem estar agora no plano xy
c) Finalmente rode ao redor de z (θz) até que u se alinhe com x.
d) note que v se alinha com y automaticamente.

Processo



Representação

- isso mostra que 3 rotações (ângulos de Euler) são suficientes para alinhar dois frames.
-É possível realizar qualquer mudança de orientação através de uma tripla (θx,θy,θz), que definem uma matriz de rotação: R(θx,θy,θz) = Rz(θz)Ry(θy)Rx(θx)

Interpolação

➡ Θ θ θ θ Ψ=(ψ ψ ψ orientações

➡Seja 0 <= α <= 1, pode-se definir:
  ➡R(α) = R((1- α)Θ + αΨ)

- Essa interpolação funciona bem quando as orientações são próximas
- Porém, não são intuitivas

Problemas

-Depende do frame de coordenadas escolhido.
  -em 2D, uma rotação de 30 graus é sempre a mesma (não depende dos eixos do sistema de coordenadas).
  -mas usando ângulos de Euler, uma rotação depende da posição do frame, e da ordem com que os vetores de base não nomeados.
-O uso de quaternions permite uma representação independente da escolha do sistema de coordenadas.

Deslocamento angular

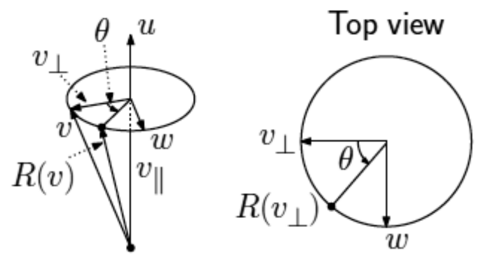
-Outra forma de representar rotações em 3D é através de um ângulo θ ao redor de um vetor u (de comprimento unitário).

-Processo:

- -considere um vetor v, que desejamos rodar.
- -considere v como a soma das projeções paralela (vl) e perpedicular (vd) de v com relação a u
  - note que a componente paralela não é afetada pela rotação, apenas a componente perpendicular.
- -construa um vetor w ortogonal a vd sobre o plano de rotação (vide equações em aula).

Deslocamento angular

$$v_{\|} = (u \cdot v)u \qquad v_{\perp} = v - v_{\|} = v - (u \cdot v)u.$$



Deslocamento Angular

Seja w um vetor perpendicular a vd

w = u x vp = u x (v – vl) = u x v – u x vl = u x v

Assim:

Rd = cos(θ) vd + sin(θ) w

R = Rd + Rl = vl + cos(θ) vd + sin(θ) w

= (u.v) u + cos(θ)(v - (u.v) u) + sin(θ) w

= cos(θ)v + (1-cos(θ)) (u.v) u + sin(θ) (u x v)

Observe: usa apenas

R = cos(θ)v + (1-cos(θ)) (u.v) u + sin(θ) (u x v)

Vantagem sobre Euler angles:

usa apenas funções geométricas intrínsecas, e portanto não depende do frame de coordenadas.

Quaternions

Primeiro, vamos lembrar algumas propriedades de números complexos

Números complexos
- 1 * 1 = 1
- i * *i* = -1 => *i* = sqrt (-1)
- 1 * *i* = *i* * 1 = *i*

Propriedades de números complexos

•
como um vetor 2D com coords (a,b)
- sqrt (a^2 + b^2)
- módulo 1 => coordenada = (cos w, sin w)
  - w
- conjugado = (a, -b)
- (cos w, sin -w) = (cos -w, sin -w)
  - ou seja, o conjugado é equivalente a negação do ângulo

Quaternions e números complexos

•
- bases complexas, *i* e *j*.
  - não conseguiu, mas após alguns anos, ele inventou um truque usando 3 bases complexas *i*, *j* e *k*.

Bases complexas

vetores de base: i, j, k

i*i = j*j = k*k = -1
i * j = k mas j * i = -k
j * k = i mas k * j = -i
k * i = j mas i * k = -j

Um QUATERNION é definido como um número complexo, generalizado, na forma
q = q0 + q1 * i + q2 * j + q3 * k

Interpretações

Quaternion como número complexo
(q0, q1, q2, q3)

Quaternion como escalar mais vetor
Seja o quartenio q = (s, u) onde
s = escalar = q0
u = vetor (q1,q2,q3)

Propriedades

- conjugado: q* = (s, -u)
- magnitude ou módulo:
  - $|q| = sqrt(q_0^2+q_1^2+q_2^2+q_3^2) = sqrt(s^2 + (u.u))$
- unitário: |q| = 1
- puro: s = 0
- inverso: $q^{-1} = q^* / |q|^2$
  - válido para quaternions não nulos
  - para quaternions unitários: $q^{-1} = q^*$

Multiplicação de quaternions

- seja q = (s,u) e p = (t,v)

q.p = (st - (u.v), sv + tu + u x v)
onde:
u.v = produto escalar
u x v = produto vetorial

Exercício: desenvolva essa expressão, e mostre que está correta

Exemplo

- Considere um quaternion unitário q = (s,u) e seu conjugado q = (s, -u)
  - Qual o valor de q.q*?
  - qual o valor de $q.q^{-1}$?

Rotação usando quaternions

- Um ponto ou vetor 3D pode ser representado por um quaternion puro
  - $p = (0, v)$
- Uma rotação de θ ao redor de u pode ser representado por um quaternion unitário
  - $q = (\cos(\theta/2), \sin(\theta/2)\, u)$
  - por que $\cos(\theta/2)$ e $\sin(\theta/2)$?
    - por assim funciona ;-)

Operador de rotação

- Para os quaternions representando:
  - rotação $q = (s, u)$
  - vetor $p = (0, v)$
- Operador de rotação é dado por:

$$Rq(p) = q\, p\, q^{-1} = q\, p\, q^*$$

Operador de rotação

- mostre que:

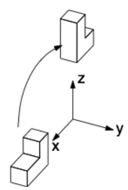
$$Rq(p) = (0, (s^2-(u.u))v + 2u(u.v) + 2s(u \times v))$$

- como $q = (\cos(w/2), \sin(w/2)\, u)$

$$Rq(p) = (0, \cos(w)\, v + (1-\cos(w))\, u\, (u.v) + \sin(w)\, (u \times v))$$

compare esse resultado com o deslocamento angular

Exemplo



- Considere:
  - uma rotação ao redor do eixo y por w = -90 graus.
  - Qual o quaternion que representa essa rotação?
  - transforme o vetor (1,0,0) usando essa rotação.

Composição de rotações

- Sejam as rotações q e q'
  - Caso q'' = q' q, mostre que
    - Rq''(p) = Rq'q(p)
  - Dica:
    - você pode usar a propriedade de quaternions: $q''^{-1} = (q'\, q)^{-1} = q^{-1}\, q'^{-1}$

Interpolação linear - lerp

- Dados dois pontos p0 e p1:

slerp(p0,p1;a)

- Interpolação linear (lerp)
  lerp (p0, p1; a) = (1-a) p0 + a p1

nlerp – int. Linear normalizada

- Considere p0 e p1 vetores unitários em uma esfera

  nlerp (p0,p1;a) = normalize (lerp(p0,p1;a))

- Resultado bom para ângulos pequenos

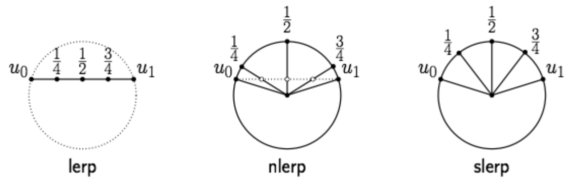


Fig. 104: Lerp, nlerp, and slerp.

Interpolação esférica - slerp

- sejam u0 e u1 dois pontos sobre a esfera
- seja w = arccos(u0 . u1)
- seja a um real no intervalo [0.0, 1.0]

$$slerp(p0, p1; a) = \frac{\sin(1-a)w}{\sin w} u0 + \frac{\sin aw}{\sin w} u1$$

Interpolação de quaternions

- Podemos utilizar nlerp ou slerp
- O uso de nlerp é imediato
- Porém, ao aplicar slerp:
  - caso o ângulo entre q0.q1 < 0 (ângulo maior que PI)
  - então substitua q1 por -q1

Representação matricial

- Para o quaternion q

  q = (cos(w/2), sin(w/2) u) = (w, x, y, z)

$1-2y^2-2z^2$	2xy - 2wz	2xz + 2wy	0
2xy + 2wz	$1-2x^2-2z^2$	2yz - 2wx	0
2xz - 2wy	2yz + 2wz	$1-2x^2-2y^2$	0
0	0	0	1

O que você deve saber

-Rotações x orientação
- -formas representação:
  - ângulos de Euler
  - deslocamento angular
  - quaternions
- -Vantagens e desvantagens de cada método

-Composição de rotações
-Interpolação entre rotações