Introduction to Computer Graphics

Marie-Paule Cani & Estelle Duveau

04/02 Introduction & projective rendering

11/02 Prodedural modeling, Interactive modeling with parametric surfaces

25/02 Introduction to OpenGL + lab: first steps & modeling

04/03 Implicit surfaces 1 + lecture/lab: transformations & hierarchies

11/03 Implicit surfaces 2 + Lights & materials in OpenGL

18/03 Textures, aliasing + Lab: Lights & materials in OpenGL

25/03 Textures in OpenGL: lecture + lab

01/04 **Procedural & kinematic animation** + lab: procedural anim

08/04 Physics: particle systems + lab: physics 1

22/04 Physics: collisions, control + lab: physics 2

29/04 Animating complex objects + Realistic rendering

06/05 Talks: results of cases studies

Computer Animation

- First animation films (Disney)
  - 30 drawings / second
  - animator in chief : key drawings
  - others : secondary drawings

- Use the computer to interpolate ?
  - positions
  - orientations
  - shapes

« Descriptive animation »

The animator fully controls the motion

Towards methods that ***generate motion*** *?*

- The user defines the laws of motion

  Examples :

  - A procedure to compute it (equation of the trajectory)
  - Physical laws (gravity, collisions…)
  - Behavioral laws (artificial intelligence)

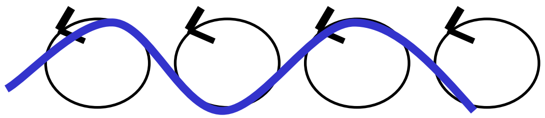
- The system generates motion from
  - The procedure
  - The initial conditions
  - Some interactive control

« Procedural animation »

- Describes a familly of motion
- Indirect control

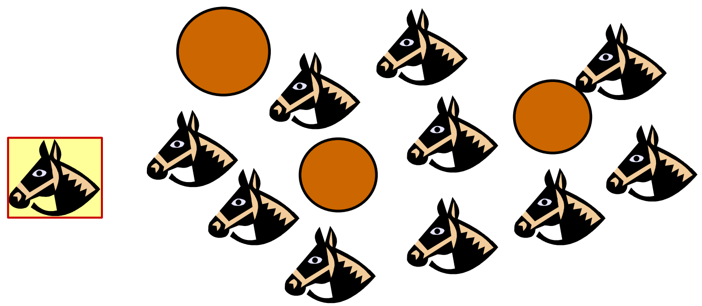
Procedural animation : Examples

- Procedural virtual ocean



- Particle systems

(fire, smoke, rain, bees, fishes…)
  - Points : X (x,y,z), V (v_x, v_y, v_z)
  - V given by a “law”
  - Birth and death of particles



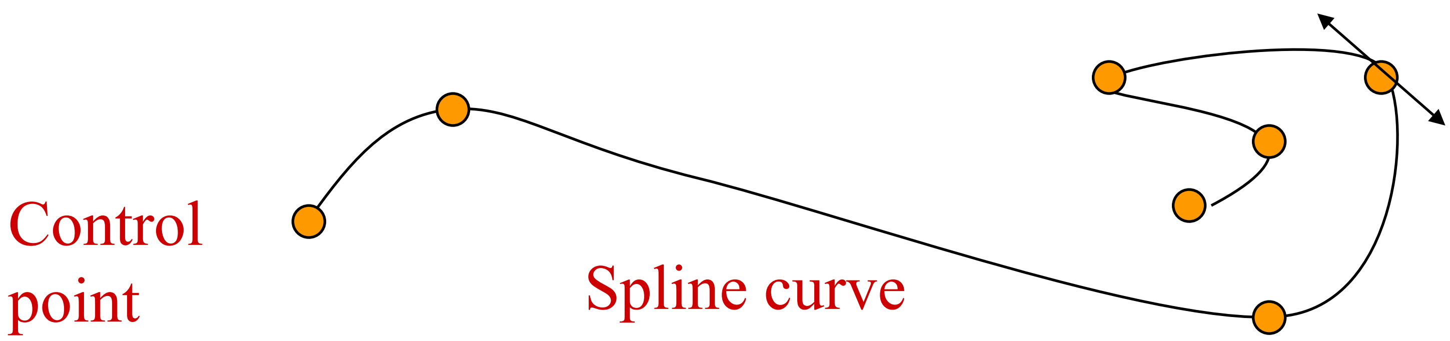
Physically-based models
later in this course

1. Descriptive models
Based on interpolation methods!

To interpolate positions: interpolation splines

Hermite curves or Cardinal splines

– Local control
  - Made of polynomial curve segments
  - degree 3, class C^1



Descriptive models
Direct kinematics

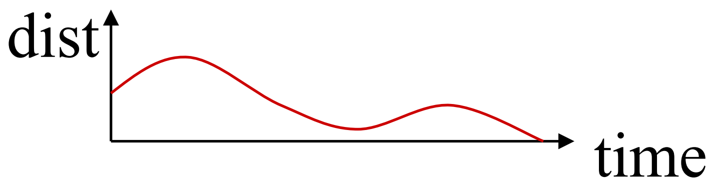
- Interpolating key positions
  - Interpolation curves

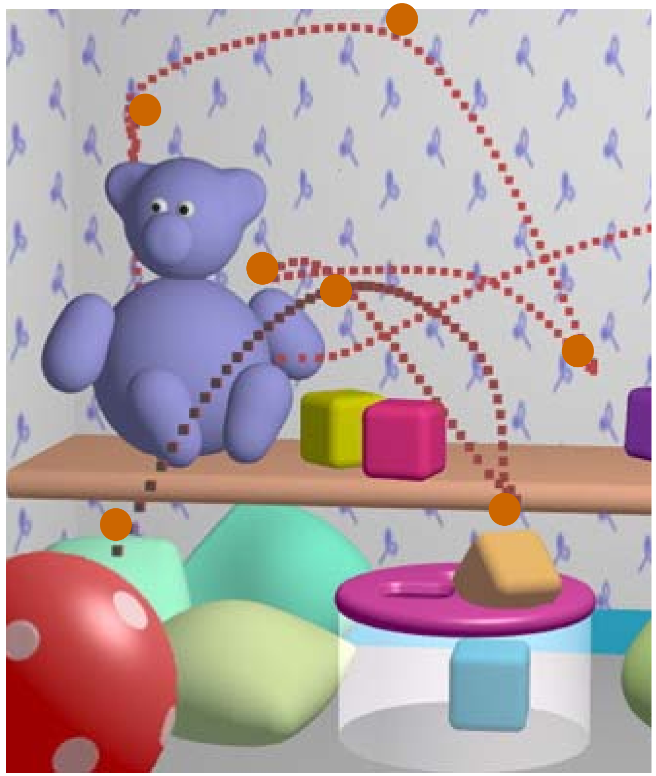
    Enable inflection points!

    (where C^0 only)

  - Control of the speed

    « velocity curve »



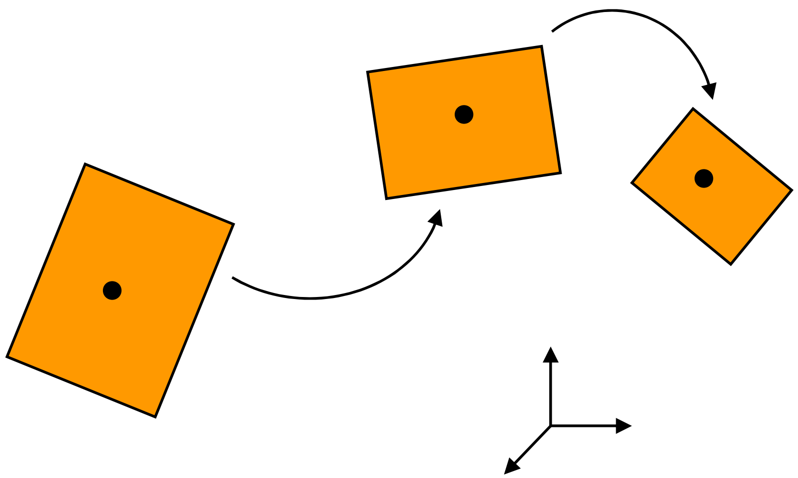


Direct kinematics

- Interpolation of orientations

  Choose the right representation !

  - Rotation matrix ?
  - Euler angle ?
  - Quaternion ?



Rotation matrix

- Representation : orthogonal matrix
  - each orientation = 9 coefficients
- Interpolation :
  - Interpolate coefficients one by one
  - Re-orthogonalize and normalize

Costly and badly adapted :

- $M = k M_1 + (1\text{-}k) M_2$ can be degenerated

Impossible to approximate it by an othogonal matrix in this case

Exemple:

Axe x, angle α

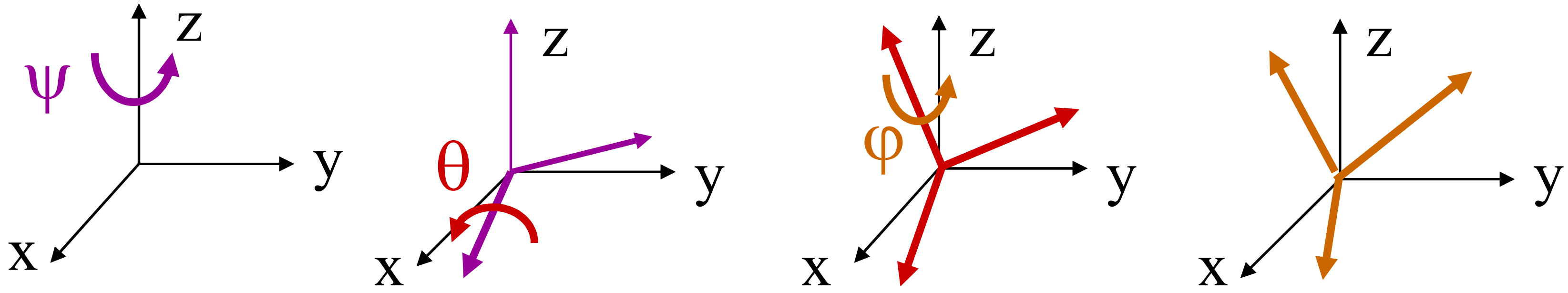
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Exemple:

M_1 = Id

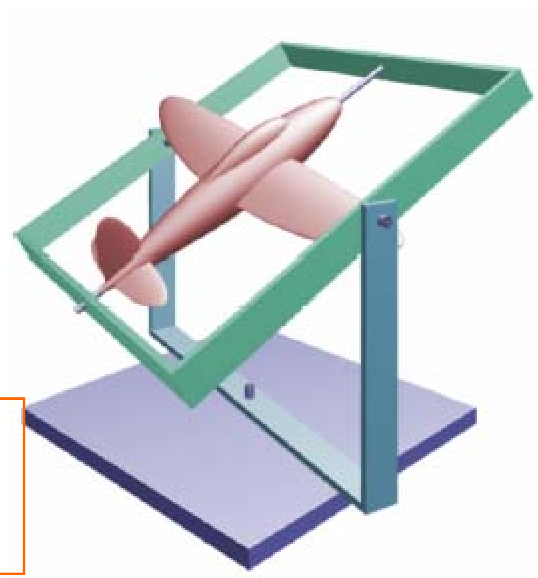
M_2 : axe x, $\alpha = \pi$

Euler Angles



Representation :

- Three angles (ψ, θ, φ)
- Intuitive : $R(V) = R_{z,\varphi}(R_{x,\theta}(R_{z,\psi}(V)))$

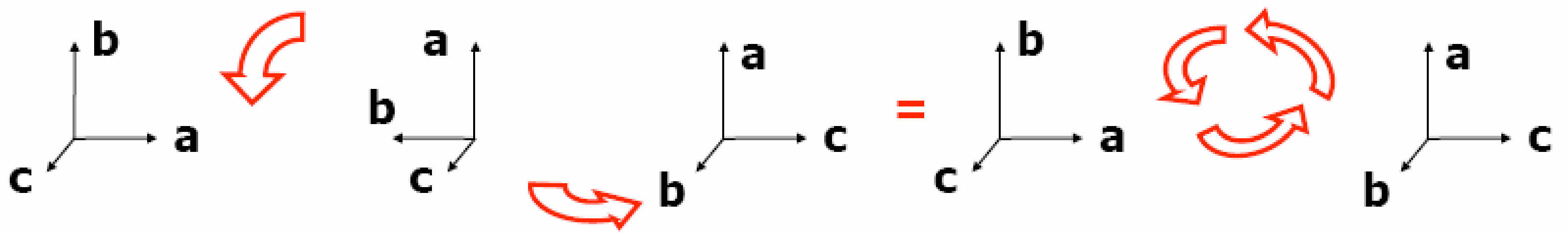


« Roll, pitch, yaw »
in flight simulators

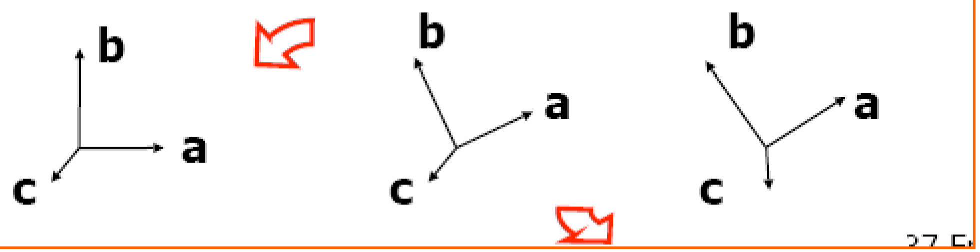
Interpolating Euler Angles

- \+ less costly : 3 values for 3 degrees of Freedom (DoF)
- – non-invariant by rotation, and un-natural result

rotation of 90° around Z, then 90° around Y = 120° around (1, 1, 1)

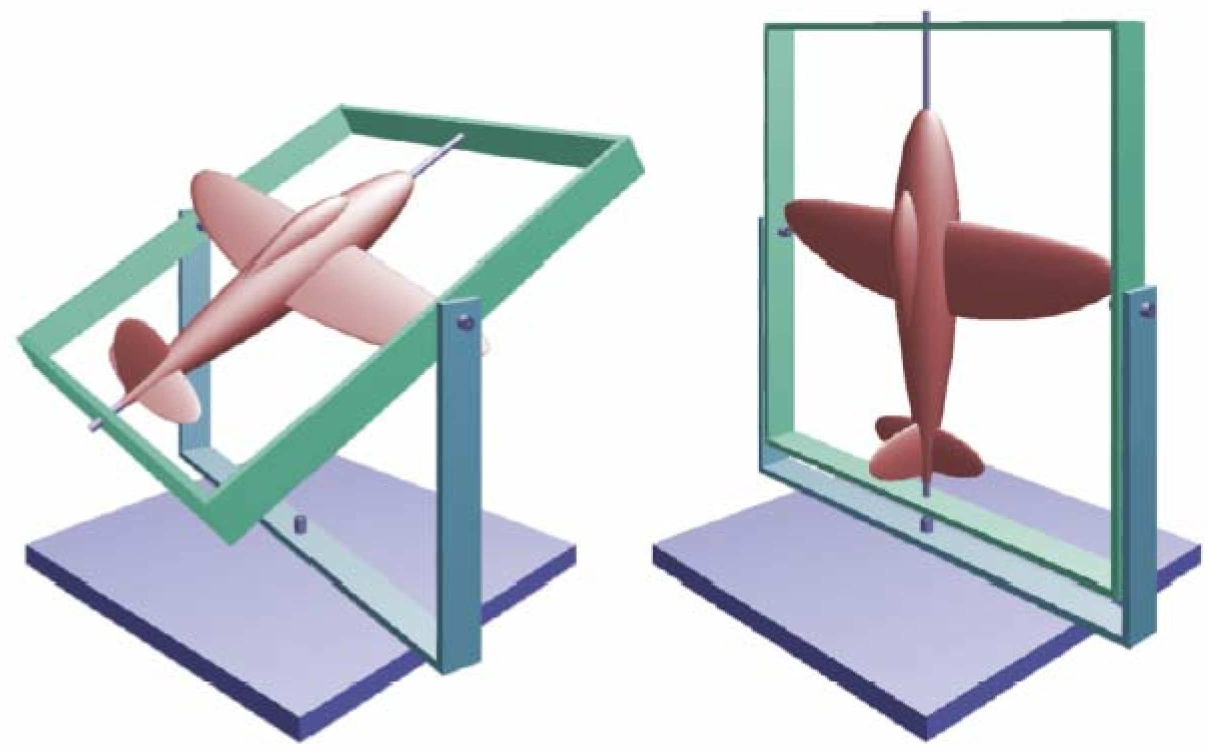


But rotation of 30° around Z then 30° around Y ≠ 40° around (1, 1, 1)



Problem with Euler Angles: gimbal lock

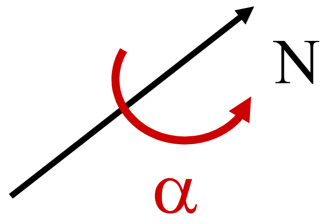
– Two or more axes aligned = loss of rotation DOF



http://www.fho-emden.de/~hoffmann/gimbal09082002.pdf

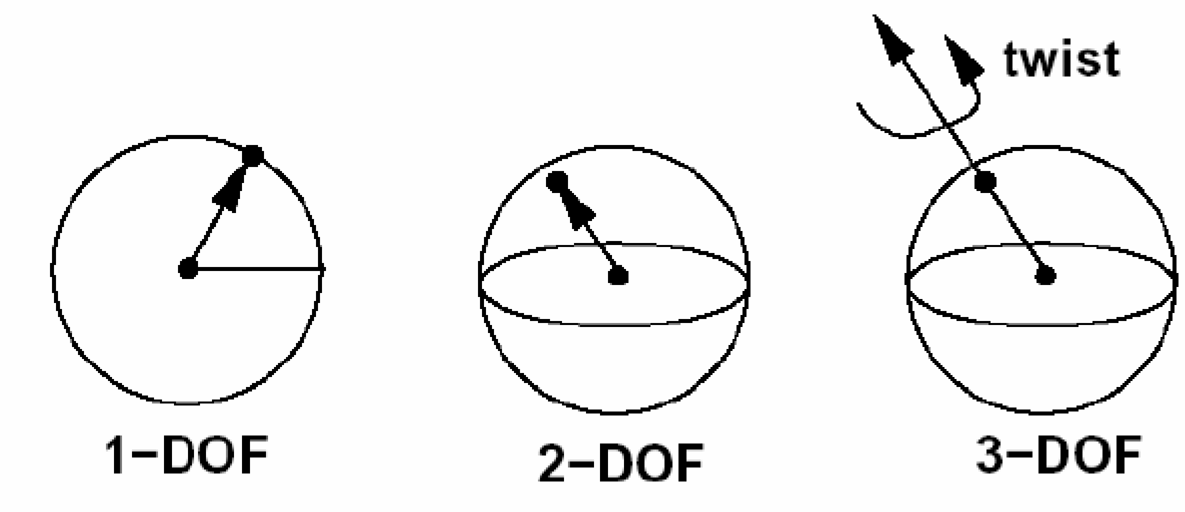
Quaternions

Representation : $q = (\cos(\alpha/2), \sin(\alpha/2)N) \in S^4$



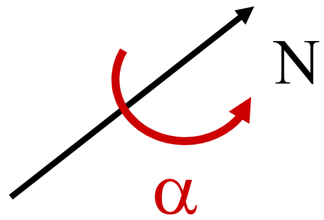
By analogy:

1, 2, 3-DoF rotations as points on 2D, 3D, 4D spheres



Quaternions

Representation : $q = (\cos(\alpha/2), \sin(\alpha/2)N) \in S^4$



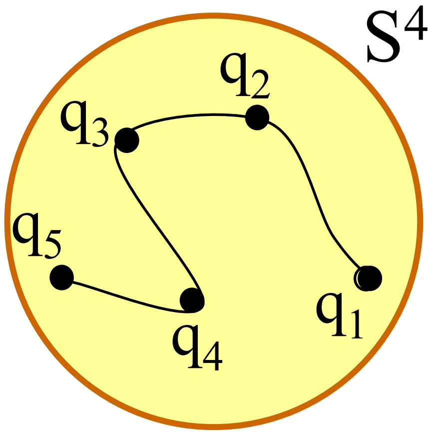
- Algebra of quaternions

  $p \,.\, q = (p_r\, q_r - p_p\, q_p \,,\, p_r\, q_p + q_r\, p_p + p_p \wedge q_p)$

  $q^{-1} = (q_r \,, -q_p) \,/\, q_r^{\,2} + q_p\, q_p$ $\qquad I = (1,0,0,0)$

- Apply a rotation

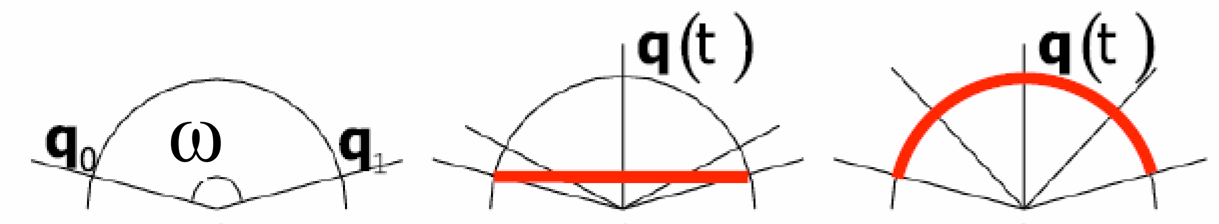
  $R(V) = (0,V)\, q^{-1}$

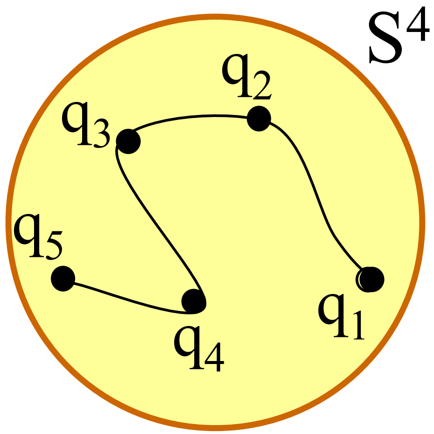


- Compose two rotations : p.q

Quaternions

- Interpolate quaternions? : splines on S^4
- Interpolation method?





  - Linear
    non-uniform speed!

$$\text{lerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \mathbf{q}(t) = \mathbf{q}_0(1-t) + \mathbf{q}_1 t$$

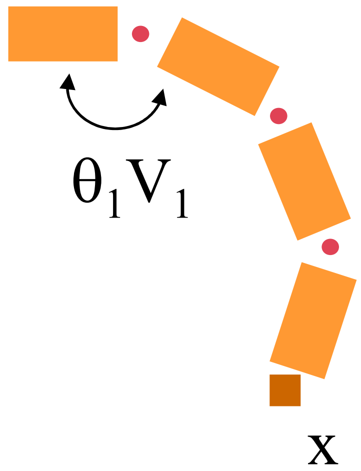
  - Use spherical!

$$\text{slerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \mathbf{q}(t) = \frac{\mathbf{q}_0 \sin((1-t)\omega) + \mathbf{q}_1 \sin(t\omega)}{\sin(\omega)}$$

Descriptive models
Hierarchical strucutres

They are essential for animation!

– Eyes move with head
– Hands move with arms
– Feet move wirth legs…

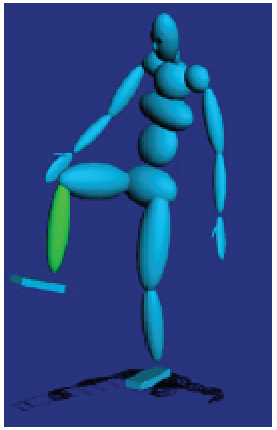


- Frame hierarchy
  – Generalized coordinates: Dof at each joint
  – Root expressed in the world frame

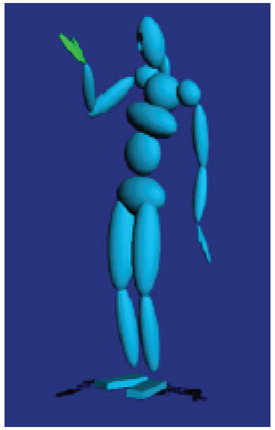
Descriptive models
Hierarchical strucutres

Example

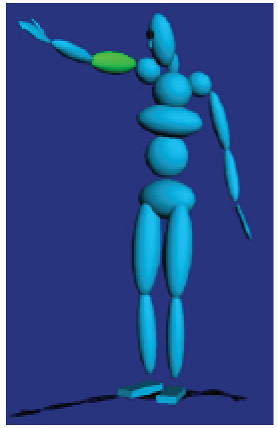
1 DOF: knee



2 DOF: wrist



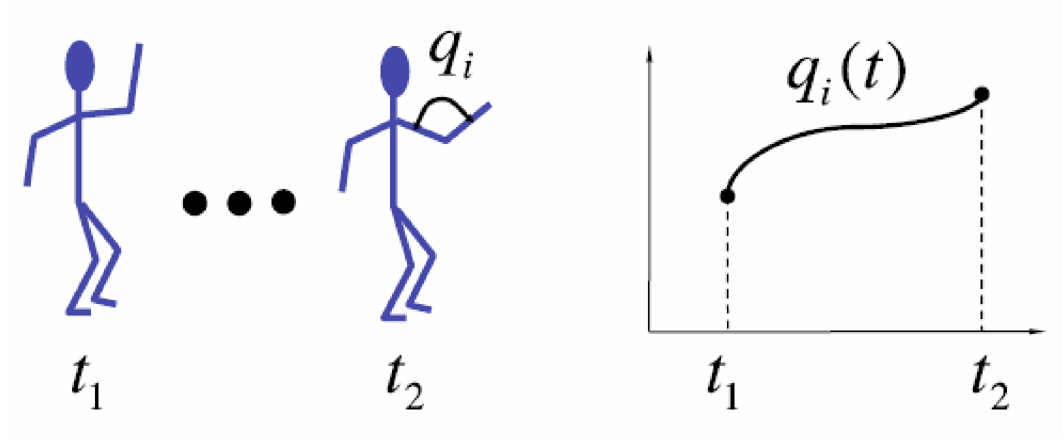
3 DOF: arm



- Frame hierarchy
  - Generalized coordinates: Dof at each joint
  - Root expressed in the world frame

Descriptive models
Forward kinematics

Interpolate key orientations



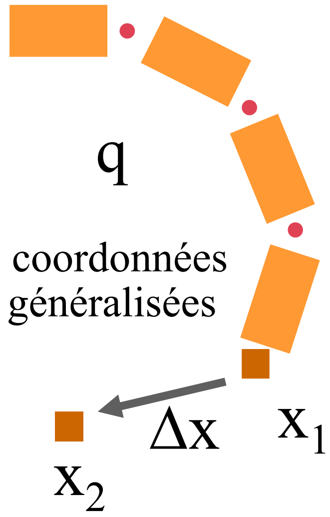
- Difficult to control extremities!
  (example : horizontal foot while cycling)
- Top-down set-up method
  - Try to compensate un-desired motion!

Descriptive models

Inverse kinematics

- Control of the end of a chain
  - Automatically compute the other orientations ?

$x_1 = f(q) \qquad x_2 = f(\ ????)$



Method from robotics

- inversion of a non-linear system

$\Delta x = J\,\Delta q$, avec $J_{ij} = \dfrac{\partial x_i}{\partial q_j}$, Jacobian matrix [J]

- Underconstrained system, pseudo-inverse : $J^+ = J^t\,(J\,J^t)^{-1}$

$\Delta q = J^+\,\Delta x$ (secondary task: $\Delta q = J^+\,\Delta x + (I - J^+ J)\,\Delta z$)

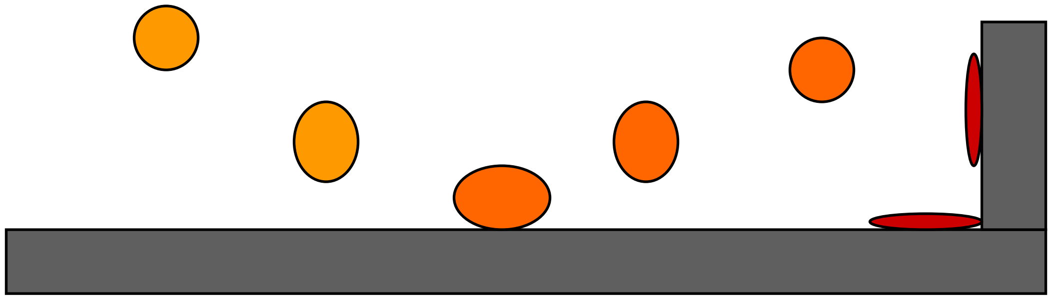
Descriptive Models
Animate Deformations

Interpolate « key shapes »

[Lasseter 1987]

- Example : « Disney effects»
  - Change scaling, color…

$$k(u) = (u^3\ u^2\ u\ 1)\ M_{spline}\ [k_{i-1}\ k_i\ k_{i+1}\ k_{i+2}]^t$$



Descriptive Models

Animate Deformations

Animate a geometric model = animate its parameters

Example: spline of subdivision surfaces

- Intermediate shapes
  – Generated by trajectories of control points

  Adapted to structured object (constant topology)

- Bounding volumes (collisions …)
  – Bspline: shape in the convex envelope of the control points

Temporal vs multi-target interpolation

Example of an animated face

- Temporal interpolation
  - Model and store all successive key- faces
- Multi-target interpolation
  - Model a few « extreme faces » from a « neutral face »
  - Animate a trajectory in this space

  (barycenters)