Foundations of Computer Graphics (Spring 2010)

CS 184, Lecture 24: Animation

http://inst.eecs.berkeley.edu/~cs184

Many slides courtesy Adam Finkelstein, James O'Brien, others

To Do

- Submit HW 4 (today)
- Start working on HW 5 (can be simple add-on)

These Lectures

- 3 classical prongs in graphics pipeline: Modeling, Rendering, Animation
- We talk a little about animation or motion
- Limited time, hence fun lectures, not covered in detail on final
- Possibility for HW 5, but only if very motivated
- Will also show historical videos

History of Computer Animation

- Video (also shown first class)

Computer Animation

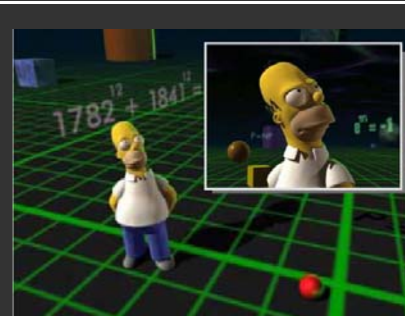
- What is animation?
  - Motion of objects (change behavior with time)
  - Often scripted with spline curve
  - Trivial example animations for HW 3

Geri's game: Pixar

- What is simulation?
  - Predict how objects move according to laws of physics
  - Graphics animation often involves "directable" simulation
  - Fracture video (O'Brien)

2D and 3D Animation

Homer 2D



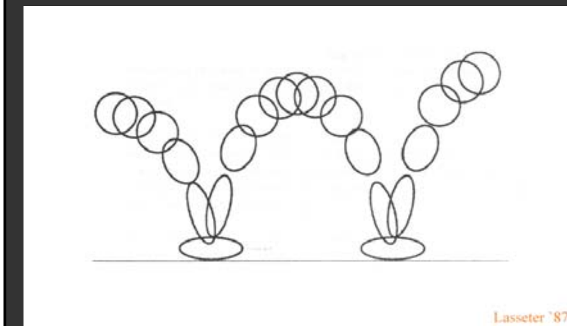
Homer 3D

Principles of Traditional Animation

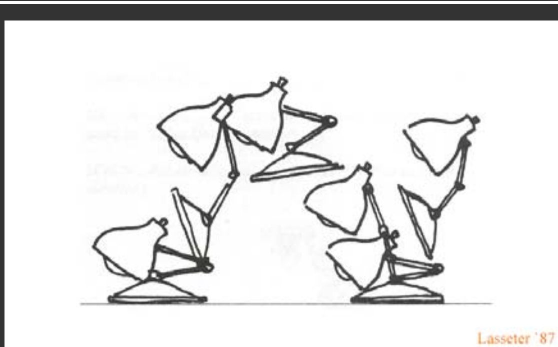
- Squash and stretch
- Slow In and out
- Anticipation
- Exaggeration
- Follow through and overlapping action
- Timing
- Staging
- Straight ahead action and pose-to-pose action
- Arcs
- Secondary action
- Appeal

Disney

Squash and Stretch



Anticipation

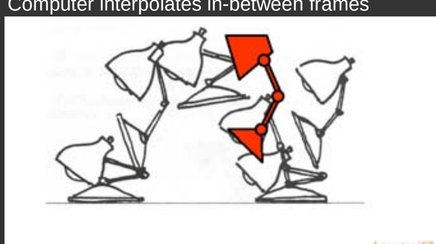


Outline

- *Keyframes*
- Articulated Figures
- Kinematics
- Dynamics

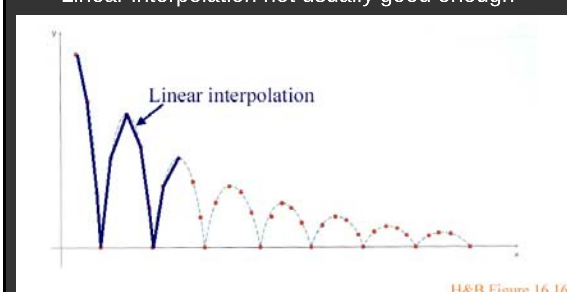
Computer Animation

- Simplest idea: Keyframing or in-betweening
- Character poses at specific keyframes
  - Computer interpolates in-between frames



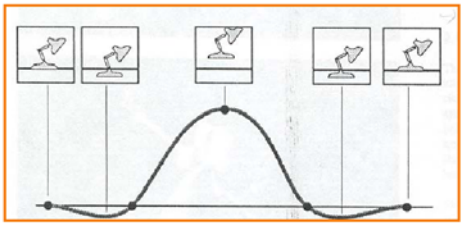
How to Interpolate?

- Linear interpolation not usually good enough



Keyframe Interpolation

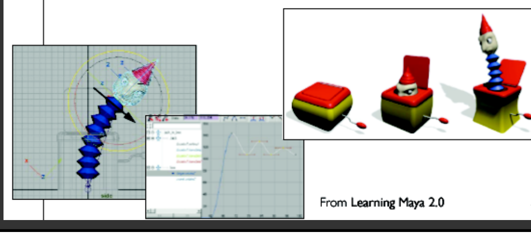
- Inbetweening:
  - Cubic spline interpolation - maybe good enough
    - » May not follow physical laws



Lasseter '87

Keyframing

- Requires a highly skilled user
- Poorly suited for interactive applications
- High quality / high expense
- Limited applicability

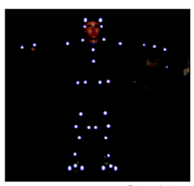


From Learning Maya 2.0

Motion Capture (recorded)

- Markers/sensors placed on subject
- Time-consuming clean-up
- Reasonable quality / reasonable price
- Manipulation algorithms an active research area

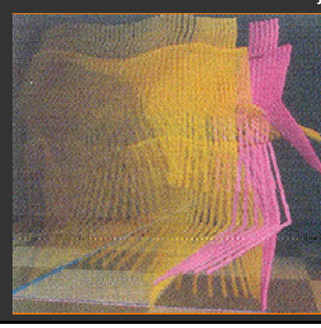
MotionAnalysis / Performance Capture Studio



Okan Arikan

Inverse Kinematics

- Consider structure of articulated object

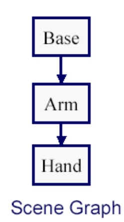


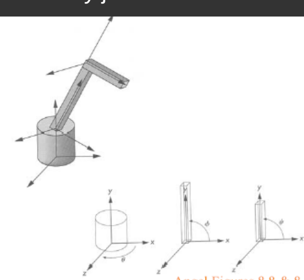
Outline

- Keyframes
- *Articulated Figures*
- Kinematics
- Dynamics

Articulated Figures

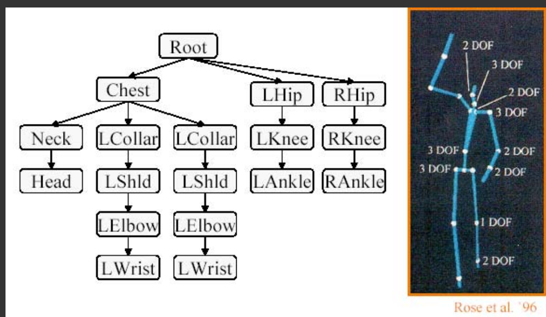
- Rigid objects connected by joints





Angel Figures 8.8 & 8.9

Humanoid Characters



Outline

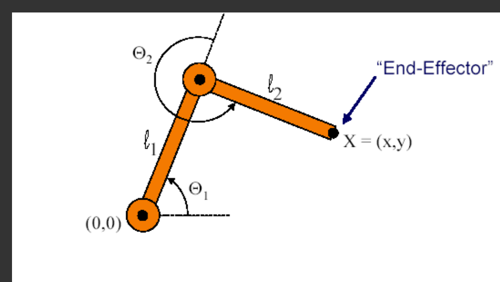
- Keyframes
- Articulated Figures
- *Kinematics*
- Dynamics

Kinematics and Dynamics

- Kinematics
  - Consider only motion. Positions, velocity, acceleration
- Dynamics
  - Considers underlying forces. Initial conditions+physics
- Articulated objects
  - Forward and inverse kinematics
  - Possibly forward and inverse dynamics
  - Many links to robotics, mechanics and other fields

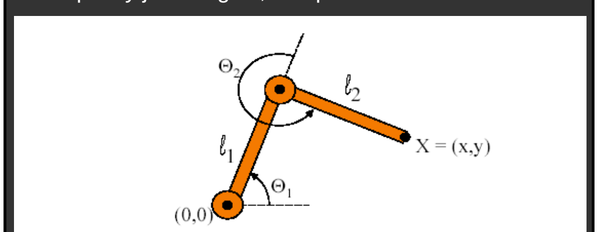
Simple 2 link arm

- 2 links connected by rotational joints



Forward Kinematics

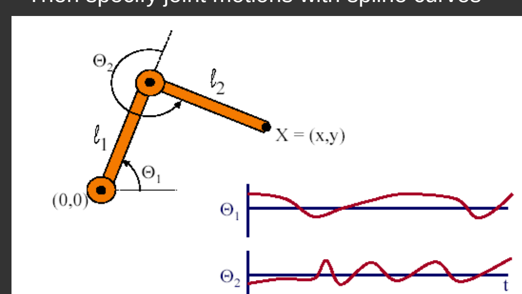
- Specify joint angles, computer finds end-effector



$$X = (l_1 \cos\Theta_1 + l_2 \cos(\Theta_1 + \Theta_2), l_1 \sin\Theta_1 + l_2 \sin(\Theta_1 + \Theta_2))$$

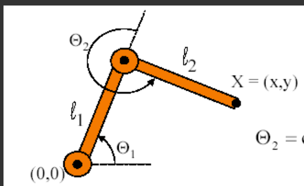
Forward Kinematics

- Then specify joint motions with spline curves



Inverse Kinematics

- Animator knows/specifies end-effector
  - System must compute joint angles
  - Harder, topic of next lecture, possible HW 5



$$\Theta_2 = \cos^{-1}\left(\frac{x^2 + x^2 - l_1^{\,2} - l_2^{\,2}}{2 l_1 l_2}\right)$$

$$\Theta_1 = \frac{-(l_2 \sin(\Theta_2))x + (l_1 + l_2 \cos(\Theta_2))y}{(l_2 \sin(\Theta_2))y + (l_1 + l_2 \cos(\Theta_2))x}$$

Summary of Kinematics

- Forward kinematics
  - Specify joint angles, system computes end-effector
- Inverse kinematics
  - Easier to specify for most animations
  - Animator specifies end-effector
  - System computes joint angles (harder)
  - “Goal-Directed” motion (animator specifies end-goals)

Outline

- Keyframes
- Articulated Figures
- Kinematics
- *Dynamics*

Dynamics

- Consider underlying forces
- Motion from initial conditions, forces
- In graphics, include goals
  - Optimization to satisfy goals and physics

Dynamics

- Simulation to ensure physical realism
- Spacetime Constraints [Witkin and Kass 88]
  - Goals (e.g. jump from here to there)
  - Optimized motion (e.g. minimize energy or torque)
  - Character’s physical structure (articulation)
  - Other constraints (foot contact, floor etc.)
  - Iterative optimization given constraint, objective

Spacetime Constraints

- Computer finds the “best” physical motion satisfying constraints
- Example: particle with jet propulsion
  - **x**(t) is position of particle at time t
  - **f**(t) is force of jet propulsion at time t
  - Particle’s equation of motion is:

    $$mx'' - f - mg = 0$$

  - Suppose we want to move from a to b within t_0 to t_1 with minimum jet fuel:

    $$\text{Minimize } \int_{t_0}^{t_1} |f(t)|^2 dt \quad \text{subject to } x(t_0) = a \text{ and } x(t_1) = b$$

Witkin & Kass '88

Spacetime Constraints

- Discretize time steps:

$$x'_i = \frac{x_i - x_{i-1}}{h}$$

$$x''_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2}$$

$$m\left(x''_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2}\right) - f_i - mg = 0$$

Minimize $h\sum_i |f_i|^2$ subject to $x_0 = a$ and $x_1 = b$

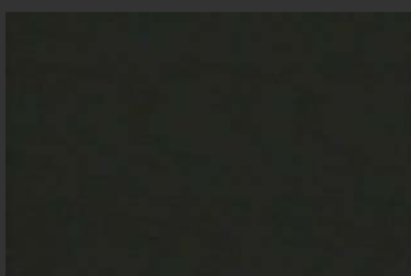
Witkin & Kass '88

Spacetime Constraints

Advantages
- Directly specify goals, not low-level joint angles etc.
- Can easily edit and vary motions

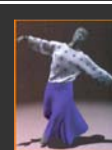
Disadvantages
- Specifying constraints, objective functions
- Optimization, and avoiding local minima

Video

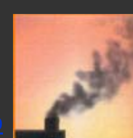


Dynamics: Physical Simulation

- Rigid Bodies
- Soft deformable objects
- Cloth
- Liquids (water)
- Gases (smoke, fluids)
- Wrinkle Synthesis Video



Cloth
(Baraff & Witkin '98)



Hot Gases
(Foster & Metaxas '97)

History of Computer Animation 2

- Part 2 of video