Computer Animation

Courtesy of Adam Finkelstein
Computer Animation

• What is animation?
  ◦ Make objects change over time according to scripted actions

• What is simulation?
  ◦ Predict how objects change over time according to physical laws

Pixar

University of Illinois
3-D and 2-D animation

Homer 3-D

Homer 2-D
Outline

- Principles of animation
- Keyframe animation
- Articulated figures
- Kinematics
- Dynamics
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
Principles of Traditional Animation

- Squash and stretch
Principles of Traditional Animation

• Slow In and Out
Principles of Traditional Animation

• Anticipation (and squash & stretch)

Lasseter ´87
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
Computer Animation

Animation pipeline

- 3D modeling
- Articulation
- Motion specification
- Motion simulation
- Shading
- Lighting
- Rendering
- Postprocessing
  » Compositing

Pixar
Keyframe Animation

• Define character poses at specific time steps called “keyframes”
Keyframe Animation

• Interpolate variables describing keyframes to determine poses for character “in-between”

Lasseter `87
Keyframe Animation

- Inbetweening:
  - Linear interpolation - usually not enough continuity
Keyframe Animation

• Inbetweening:
  ◦ Spline interpolation - maybe good enough
Keyframe Animation

• Inbetweening:
  ◦ Cubic spline interpolation - maybe good enough
    » May not follow physical laws
Keyframe Animation

• Inbetweening:
  ◦ Cubic spline interpolation - maybe good enough
    » May not follow physical laws
Keyframe Animation

• Inbetweening:
  ◦ Inverse kinematics or dynamics

Rose et al. `96
Outline

• Principles of animation
• Keyframe animation
• Articulated figures
• Kinematics
• Dynamics
Articulated Figures

- Character poses described by set of rigid bodies connected by “joints”

Scene Graph
Articulated Figures

• Well-suited for humanoid characters

Rose et al. ’96
Articulated Figures

Joints provide handles for moving articulated figure
Articulated Figures

• Inbetweening
  ◦ Compute joint angles between keyframes

Good arm  Bad arm

Watt & Watt
Example: Walk Cycle

• Articulated figure:
Example: Walk Cycle

• Hip joint orientation:
Example: Walk Cycle

- Knee joint orientation:
Example: Walk Cycle

- Ankle joint orientation:
Example: Run Cycle

Mike Marr, COS 426, Princeton University, 1995
Example: Ice Skating

(Mao Chen, Zaijin Guan, Zhiyan Liu, Xiaohu Qie, CS426, Fall98, Princeton University)
Outline

• Principles of animation
• Keyframe animation
• Articulated figures
• Kinematics
• Dynamics
Kinematics and Dynamics

• Kinematics
  ◦ Considers only motion
  ◦ Determined by positions, velocities, accelerations

• Dynamics
  ◦ Considers underlying forces
  ◦ Compute motion from initial conditions and physics
Example: 2-Link Structure

- Two links connected by rotational joints

\[ \Theta_1, \Theta_2 \]

\[ X = (x,y) \]

"End-Effector"
Forward Kinematics

- Animator specifies joint angles: $\Theta_1$ and $\Theta_2$
- Computer finds positions of end-effector: $X$

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

- Joint motions can be specified by spline curves

\[ X = (x, y) \]
Forward Kinematics

- Joint motions can be specified by initial conditions and velocities

\[ \Theta_1(0) = 60^\circ \quad \Theta_2(0) = 250^\circ \]

\[ \frac{d\Theta_1}{dt} = 1.2 \quad \frac{d\Theta_2}{dt} = -0.1 \]
Example: 2-Link Structure

• What if animator knows position of “end-effector”
Inverse Kinematics

• Animator specifies end-effector positions: \( X \)
• Computer finds joint angles: \( \Theta_1 \) and \( \Theta_2 \):

\[
\begin{align*}
\Theta_2 &= \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \\
\Theta_1 &= \frac{-(l_2 \sin(\Theta_2))x + (l_1 + l_2 \cos(\Theta_2))y}{(l_2 \sin(\Theta_2))y + (l_1 + l_2 \cos(\Theta_2))x}
\end{align*}
\]
Inverse Kinematics

- End-effector positions can be specified by splines
Inverse Kinematics

• Problem for more complex structures
  ○ System of equations is usually under-defined
  ○ Multiple solutions

Three unknowns: $\Theta_1, \Theta_2, \Theta_3$
Two equations: $x, y$
Inverse Kinematics

- Solution for more complex structures:
  - Find best solution (e.g., minimize energy in motion)
  - Non-linear optimization
Summary of Kinematics

• Forward kinematics
  ◦ Specify conditions (joint angles)
  ◦ Compute positions of end-effectors

• Inverse kinematics
  ◦ “Goal-directed” motion
  ◦ Specify goal positions of end effectors
  ◦ Compute conditions required to achieve goals

Inverse kinematics provides easier specification for many animation tasks, but it is computationally more difficult
Overview

• Kinematics
  ○ Considers only motion
  ○ Determined by positions, velocities, accelerations

• Dynamics
  ○ Considers underlying forces
  ○ Compute motion from initial conditions and physics
Dynamics

- Simulation of physics insures realism of motion

Lasseter ‘87
Spacetime Constraints

• Animator specifies constraints:
  ◦ What the character’s physical structure is
    » e.g., articulated figure
  ◦ What the character has to do
    » e.g., jump from here to there within time \( t \)
  ◦ What other physical structures are present
    » e.g., floor to push off and land
  ◦ How the motion should be performed
    » e.g., minimize energy
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:
    $$m x'' - f - mg = 0$$
  ○ Suppose we want to move from $a$ to $b$ within $t_0$ to $t_1$ with minimum jet fuel:
    $$\text{Minimize } \int_{t_0}^{t_1} |f(t)|^2 dt \text{ subject to } x(t_0)=a \text{ and } x(t_1)=b$$

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

- Solve with iterative optimization methods

Witkin & Kass ‘88
Spacetime Constraints

• Advantages:
  ○ Free animator from having to specify details of physically realistic motion with spline curves
  ○ Easy to vary motions due to new parameters and/or new constraints

• Challenges:
  ○ Specifying constraints and objective functions
  ○ Avoiding local minima during optimization
Spacetime Constraints

• Adapting motion:

Original Jump

Heavier Base

Witkin & Kass `88
Spacetime Constraints

- Adapting motion:

Witkin & Kass `88

Hurdle
Spacetime Constraints

• Adapting motion:

Ski Jump

Witkin & Kass `88
Spacetime Constraints

- Editing motion:
Spacetime Constraints

• Morphing motion:
Dynamics

• Other physical simulations:
  ○ Rigid bodies
  ○ Soft bodies
  ○ Cloth
  ○ Liquids
  ○ Gases
  ○ etc.

Hot Gases
(Foster & Metaxas `97)

Cloth
(Baraff & Witkin `98)
Summary

• Principles of animation
• Keyframe animation
• Articulated figures
• Kinematics
  ◦ Forward kinematics
  ◦ Inverse kinematics
• Dynamics
  ◦ Space-time constraints
  ◦ Also other physical simulations