Kinematics

Tom Funkhouser
Princeton University
COS 426, Fall 2006

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

Outline

- Articulated figures
- Keyframe animation
- Kinematics
- Dynamics
- Guidelines
**Articulated Figures**

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

![Scene Graph](Image)

**Articulated Figures**

- Well-suited for humanoid characters

![Angel Plate 1](Image)

**Articulated Figures**

- Joints provide handles for moving articulated figure

![Mike Marr, COS 426, Princeton University, 1995](Image)

**Outline**

- Articulated figures
- Keyframe animation
- Kinematics
- Dynamics
- Guidelines

![Angel Plate 1](Image)

**Keyframe Animation**

- Define character poses at specific time steps called “keyframes”

![Lasseter ’87](Image)

**Keyframe Animation**

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

![Lasseter ’87](Image)
### Keyframe Animation

- **Inbetweening:**
  - Linear interpolation - usually not enough continuity
  
  ![Linear interpolation graph](H&B Figure 16.16)

- **Inbetweening:**
  - Spline interpolation - maybe good enough
  
  ![Spline interpolation graph](H&B Figure 16.11)

### Cubic Spline Interpolation

- May not follow physical laws

  ![Cubic spline interpolation example](Watt & Watt 1987)

### Example: Walk Cycle

- **Articulated figure:**
  - Hip
  - Upper leg (hip rot)
  - Hip rotate
  - Lower leg (knee rot)
  - Flip rotate + knee rot
  - Foot (ankle rot)

  ![Walk cycle example](Watt & Watt)

- **Hip joint orientation:**
  - 45°
  - -35°

  ![Hip joint orientation example](Watt & Watt)
Example: Walk Cycle

- Knee joint orientation:

Example: Walk Cycle

- Ankle joint orientation:

Example: Robot

Example: Ice Skating

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

Outline

- Articulated figures
- Keyframe animation
  - Kinematics
- Dynamics
- Guidelines

Animating Motion

- Kinematics
  - Considers only motion
- Dynamics
  - Considers underlying forces
  - Compute motion from initial conditions and physics
**Example: 2-Link Structure**
- Two links connected by rotational joints

![Diagram](image1)

**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 initial conditions and velocities

![Diagram](image2)

**Inverse Kinematics**
- Animator specifies end-effector positions: \( X \)
- Computer finds joint angles: \( \theta_1 \) and \( \theta_2 \):

\[
\begin{align*}
\theta_2 &= \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \\
\theta_1 &= -\frac{(l_2 \sin \theta_2)x + (l_1 + l_2 \cos \theta_2)y}{(l_2 \sin \theta_2)y + (l_1 + l_2 \cos \theta_2)x}
\end{align*}
\]

**Example: 2-Link Structure**
- What if animator knows position of "end-effector"

![Diagram](image3)

**Forward Kinematics**
- Joint motions can be specified by spline curves

![Diagram](image4)

**Forward Kinematics**
- Joint motions can be specified by initial conditions and velocities

\[
\begin{align*}
\theta_1(0) &= 60^\circ \\
\theta_2(0) &= 250^\circ \\
\frac{d\theta_1}{dt} &= 1.2 \\
\frac{d\theta_2}{dt} &= -0.1
\end{align*}
\]

**Inverse Kinematics**
- Animator specifies end-effector positions: \( X \)
- Computer finds joint angles: \( \theta_1 \) and \( \theta_2 \):

\[
\begin{align*}
\theta_2 &= \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \\
\theta_1 &= -\frac{(l_2 \sin \theta_2)x + (l_1 + l_2 \cos \theta_2)y}{(l_2 \sin \theta_2)y + (l_1 + l_2 \cos \theta_2)x}
\end{align*}
\]
Inverse Kinematics

- End-effector positions can be specified by spline curves

\[ X = (x, y) \]

\[ \theta_1 \]

\[ \theta_2 \]

\[ \theta_3 \]

\[ l_1 \]

\[ l_2 \]

\[ l_3 \]

(0,0)

x

y

Example: Ball Boy

“Ballboy”

Example: Toy Story II

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
  - Suitable for in-betweening

Inverse kinematics provides easier specification for many animation tasks, but it is computationally more difficult.
Outline
- Articulated figures
- Keyframe animation
- Kinematics

Dynamics
- Simulation of physics insures realism of motion

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:
    \[ m\ddot{x} - f - mg = 0 \]
  - Suppose we want to move from \( a \) to \( b \) within \( t_a \) to \( t_f \) with minimum jet fuel:
    Minimize \( \int_{t_a}^{t_f} \| f(t) \| \, dt \) subject to \( x(t_a) = a \) and \( x(t_f) = b \)

Spacetime Constraints
- Discretize time steps:
  \[
  \begin{align*}
  x' &= \frac{x - x_{i-1}}{h} \\
  x'' &= \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} \\
  m(x'' - \ddot{x} - \frac{2x_i + x_{i-1}}{h}) - f_i - mg &= 0 \\
  \text{Minimize } h \sum_i |f_i| \text{ subject to } x_0 = a \text{ and } x_n = b
  \end{align*}
  \]

Spacetime Constraints
- Solve with iterative optimization methods
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

Example: Monsters, Inc.

- Specifying constraints and objective functions
- Avoiding local minima during optimization

- Adapting motion:
  - Physically realistic motion with spline curves

- Adapting motion:
  - Original Jump

- Adapting motion:
  - Heavier Base

- Adapting motion:
  - Ski Jump

- Adapting motion:
  - Hurdle

- Adapting motion:
  - Original Jump

- Adapting motion:
  - Heavier Base

- Adapting motion:
  - Ski Jump

- Adapting motion:
  - Hurdle
Summary

- Articulated figures
  - Hierarchies parts connected by joints
- Keyframe animation
  - Poses specified at key times
  - In-betweening to fill in the rest
- Kinematics
  - Forward kinematics
  - Inverse kinematics
- Dynamics
  - Space-time constraints
  - Also other physical simulations in previous lecture