Lecture 17:
Animation 2
(Simulation and Kinematics)
Last Lecture

• Animation
  - History
  - Keyframe animation
  - Physical simulation
Today

• Simulation
  - Given the formulae, how to implement them

• Rigging

• Kinematics

• Motion Capture

• Conclusion
Single Particle Simulation

First study motion of a single particle

- Later, generalize to a multitude of particles

To start, assume motion of particle determined by a velocity vector field that is a function of position and time:

\[ v(x, t) \]

Slide courtesy of Prof. Ren Ng, UC Berkeley
Ordinary Differential Equation (ODE)

Computing position of particle over time requires solving a first-order ordinary differential equation:

\[ \frac{dx}{dt} = \dot{x} = v(x, t) \]

“First-order” refers to the first derivative being taken.

“Ordinary” means no “partial” derivatives, i.e. \( x \) is just a function of \( t \)
Solving for Particle Position

We can solve the ODE, subject to a given initial particle position $x_0$, by using forward numerical integration.
Euler’s Method

Euler’s Method (a.k.a. Forward Euler, Explicit Euler)

• Simple iterative method
• Commonly used
• Very inaccurate
• Most often goes unstable

\[ \mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \Delta t \dot{\mathbf{x}}^t \]

\[ \dot{\mathbf{x}}^{t+\Delta t} = \dot{\mathbf{x}}^t + \Delta t \ddot{\mathbf{x}}^t \]
Euler’s Method - Errors

With numerical integration, errors accumulate
Euler integration is particularly bad

Example:

\[ \mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \Delta t \mathbf{v}(\mathbf{x}, t) \]
Instability of Forward Euler Method

Forward Euler (explicit)

\[ \mathbf{x}^{t+\Delta t} = \mathbf{x}^{t} + \Delta t \mathbf{v}(\mathbf{x}, t) \]

Two key problems:

• Inaccuracies increase as time step \( \Delta t \) increases

• Instability is a common, serious problem that can cause simulation to diverge
Combating Instability
Some Methods to Combat Instability

Midpoint method / Modified Euler
• Average velocities at start and endpoint

Adaptive step size
• Compare one step and two half-steps, recursively, until error is acceptable

Implicit methods
• Use the velocity at the next time step (hard)

Position-based / Verlet integration
• Constrain positions and velocities of particles after time step
Midpoint Method

Midpoint method

• Compute Euler step (a)
• Compute derivative at midpoint of Euler step (b)
• Update position using midpoint derivative (c)

\[
x_{\text{mid}} = x(t) + \Delta t/2 \cdot v(x(t), t)
\]
\[
x(t + \Delta t) = x(t) + \Delta t \cdot v(x_{\text{mid}}, t)
\]

Witkin and Baraff
Modified Euler

- Average velocity at start and end of step
- Better results

\[
x^{t+\Delta t} = x^t + \frac{\Delta t}{2} \left( \dot{x}^t + \dot{x}^{t+\Delta t} \right)
\]

\[
\ddot{x}^{t+\Delta t} = \ddot{x}^t + \Delta t \dddot{x}^t
\]

\[
x^{t+\Delta t} = x^t + \Delta t \dot{x}^t + \frac{(\Delta t)^2}{2} \dddot{x}^t
\]

Slide courtesy of Prof. Ren Ng, UC Berkeley
Adaptive Step Size

Adaptive step size

• Technique for choosing step size based on error estimate
• Very practical technique
• But may need very small steps!

Repeat until error is below threshold:

• Compute $x_T$ an Euler step, size $T$
• Compute $x_{T/2}$ two Euler steps, size $T/2$
• Compute error $\| x_T - x_{T/2} \|$
• If (error > threshold) reduce step size and try again
Implicit Euler Method

Implicit methods

• Informally called backward methods
• Use derivatives in the future, for the current step

\[ x^{t+\Delta t} = x^t + \Delta t \dot{x}^{t+\Delta t} \]

\[ \dot{x}^{t+\Delta t} = \dot{x}^t + \Delta t \ddot{x}^{t+\Delta t} \]
Implicit Euler Method

Implicit methods

- Informally called backward methods
- Use derivatives in the future, for the current step

\[ x^{t+\Delta t} = x^t + \Delta t \, \dot{x}^{t+\Delta t} \]
\[ \dot{x}^{t+\Delta t} = \dot{x}^t + \Delta t \, \ddot{x}^{t+\Delta t} \]

- Solve nonlinear problem for \( x^{t+\Delta t} \) and \( \dot{x}^{t+\Delta t} \)
- Use root-finding algorithm, e.g. Newton’s method
- Offers much better stability
Position-Based / Verlet Integration

Idea:

• After modified Euler forward-step, constrain positions of particles to prevent divergent, unstable behavior
• Use constrained positions to calculate velocity
• Both of these ideas will dissipate energy, stabilize

Pros / cons

• Fast and simple
• Not physically based, dissipates energy (error)

Details in Assignment 8
Particle Systems
Particle Systems

Model dynamical systems as collections of large numbers of particles

Each particle’s motion is defined by a set of physical (or non-physical) forces

Popular technique in graphics and games
  • Easy to understand, implement
  • Scalable: fewer particles for speed, more for higher complexity

Challenges
  • May need many particles (e.g. fluids)
  • May need acceleration structures (e.g. to find nearest particles for interactions)
Particle System Animations

For each frame in animation
- [If needed] Create new particles
- Calculate forces on each particle
- Update each particle’s position and velocity
- [If needed] Remove dead particles
- Render particles
Particle System Forces

Attraction and repulsion forces
- Gravity, electromagnetism, ...
- Springs, propulsion, ...

Damping forces
- Friction, air drag, viscosity, ...

Collisions
- Walls, containers, fixed objects, ...
- Dynamic objects, character body parts, ...
Gravitational Attraction

Newton’s universal law of gravitation

• Gravitational pull between particles

\[ F_g = G \frac{m_1 m_2}{d^2} \]
\[ G = 6.67428 \times 10^{-11} \text{ Nm}^2\text{kg}^{-2} \]
Example: Galaxy Simulation

Disk galaxy simulation, NASA Goddard

Slide courtesy of Prof. Ren Ng, UC Berkeley
Example: Particle-Based Fluids

Macklin and Müller, Position Based Fluids

Slide courtesy of Prof. Ren Ng, UC Berkeley
Example: Material Point Method

Chenfanfu Jiang et al., The Affine Particle In Cell Method
Simulated Flocking as an ODE

Model each bird as a particle
Subject to very simple forces:

- attraction to center of neighbors
- repulsion from individual neighbors
- alignment toward average trajectory of neighbors

Simulate evolution of large particle system numerically
Emergent complex behavior (also seen in fish, bees, ...)

Credit: Craig Reynolds (see http://www.red3d.com/cwr/boids/)

Slide credit: Keenan Crane
Example: Crowds + “Rock” Dynamics
Animation Production Pipeline
Rigging
Rigging

• Rigging is a set of higher level controls on a character that allow more rapid & intuitive modification of pose, deformations, expression, etc.

• Important
  - Like strings on a puppet
  - Captures all meaningful character changes
  - Varies from character to character

• Expensive to create
  - Manual effort
  - Requires both artistic and technical training
Rigging Example

Courtesy Matthew Lailler via Keenan Crane
Blend Shapes

• Instead of skeleton, interpolate directly between surfaces

• E.g., model a collection of facial expressions:

• Simplest scheme: take linear combination of vertex positions

• Spline used to control choice of weights over time
Blend Shapes

Courtesy Félix Ferrand
Forward Kinematics
(Slides by Prof. James O’Brien)
Forward Kinematics

Articulated skeleton

- Topology (what’s connected to what)
- Geometric relations from joints
- Tree structure (in absence of loops)

Joint types

- Pin (1D rotation)
- Ball (2D rotation)
- Prismatic joint (translation)
Forward Kinematics

Example: simple two segment arm in 2D
Forward Kinematics

Animator provides angles, and computer determines position $p$ of end-effector

$$pz = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$px = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Warning: Z-up Coordinate System
Forward Kinematics

Animation is described as angle parameter values as a function of time

\[ p_z = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \]
\[ p_x = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \]

Warning: Z-up Coordinate System
Example Walk Cycle
Kinematics Pros and Cons

Strengths

• Direct control is convenient
• Implementation is straightforward

Weaknesses

• Animation may be inconsistent with physics
• Time consuming for artists
Inverse Kinematics
Inverse Kinematics
Inverse Kinematics

Animator provides position of end-effector, and computer must determine joint angles that satisfy constraints.
Inverse Kinematics

Direct inverse kinematics: for two-segment arm, can solve for parameters analytically

\[ \theta_2 = \cos^{-1} \left( \frac{p_z^2 + p_x^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \]

\[ \theta_1 = \frac{-p_z l_2 \sin(\theta_2) + p_x (l_1 + l_2 \cos(\theta_2))}{p_x l_2 \sin(\theta_2) + p_z (l_1 + l_2 \cos(\theta_2))} \]
Inverse Kinematics

Why is the problem hard?

- Multiple solutions separated in configuration space
Inverse Kinematics

Why is the problem hard?

- Multiple solutions connected in configuration space
Inverse Kinematics

Why is the problem hard?

- Solutions may not always exist
Inverse Kinematics

Numerical solution to general N-link IK problem

• Choose an initial configuration
• Define an error metric (e.g. square of distance between goal and current position)
• Compute gradient of error as function of configuration
• Apply gradient descent (or Newton’s method, or other optimization procedure)
Style-Based IK

Grochow et al., Style Based Inverse Kinematics
Motion Capture
Motion Capture

Data-driven approach to creating animation sequences

• Record real-world performances (e.g. person executing an activity)

• Extract pose as a function of time from the data collected
Warmup: Motion Capture in Production

Behind the Scenes of Alita: Battle Angel at Weta Digital | TE Connectivity,
https://www.youtube.com/watch?v=7fsf42xnBdo
Motion Capture Pros and Cons

Strengths

• Can capture large amounts of real data quickly
• Realism can be high

Weaknesses

• Complex and costly set-ups
• Captured animation may not meet artistic needs, requiring alterations
Motion Capture Equipment

**Optical**  
(More on following slides)

**Magnetic**  
Sense magnetic fields to infer position / orientation.  
Tethered.

**Mechanical**  
Measure joint angles directly.  
Restricts motion.

Slide courtesy of Prof. Ren Ng, UC Berkeley
Optical Motion Capture

- Markers on subject
- Positions by triangulation from multiple cameras
- 8+ cameras, 240 Hz, occlusions are difficult

Retroflective markers attached to subject

IR illumination and cameras

Slide credit: Steve Marschner
Optical Motion Capture

Ronda Rousey in Electronic Arts’ motion capture studio

Source: http://fightland.vice.com/blog/ronda-rousey-20-the-queen-of-all-media
Motion Data

Subset of motion curves from captured walking motion.

From Witkin and Popovic, 1995

Slide courtesy of Prof. Ren Ng, UC Berkeley
Challenges of Facial Animation

Uncanny valley

• In robotics and graphics
• As artificial character appearance approaches human realism, our emotional response goes negative, until it achieves a sufficiently convincing level of realism in expression

Cartoon. Brave, Pixar


Slide courtesy of Prof. Ren Ng, UC Berkeley
Facial Motion Capture

Discovery, “Avatar: Motion Capture Mirrors Emotions”, https://youtu.be/1wK1Ixr-UmM

Slide courtesy of Prof. Ren Ng, UC Berkeley
Comparison

Video Credit to: Yaoyi Bai, University of Pennsylvania, CIS 562 F16 Homework8
Comparison

Video Credit to: Jiahao Liu, University of Pennsylvania, CIS 562 F16 Homework8
Conclusion

Rasterization

Geometry

Ray tracing

Animation / simulation
Thank you!