Computer Animation

Tom Funkhouser

Princeton University

COS 426, Spring 2006

Computer Animation

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

Pixar

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

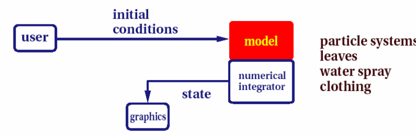
University of Illinois

Simulation

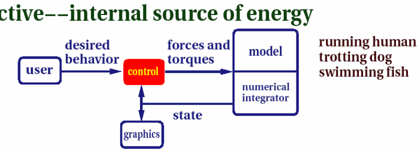
- Dynamics
  - Considers underlying forces
  - Compute motion from initial conditions and physics
- Kinematics
  - Considers only motion
  - Determined by positions, velocities, accelerations

Dynamics

Passive––no muscles or motors



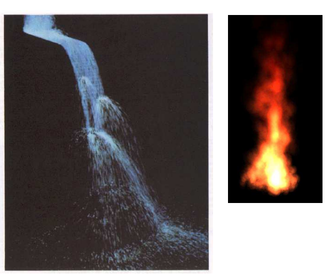
Active––internal source of energy



Hodgins

Passive Dynamics

- No muscles or motors
  - Smoke
  - Water
  - Cloth
  - Fire
  - Fireworks
  - Dice

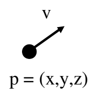


Passive Dynamics

- Physical laws
  - Newton's laws
  - Hook's law
  - Etc.
- Physical phenomena
  - Gravity
  - Momentum
  - Friction
  - Collisions
  - Elasticity
  - Fracture

Particle Systems

- A particle is a point mass
  - Mass
  - Position
  - Velocity
  - Forces
  - Color
  - Lifetime
- Use lots of particles to model complex phenomena
  - Keep array of particles
  - Newton's laws



Particle Systems

- For each frame:
  - Create new particles and assign attributes
  - Delete any expired particles
  - Update particles based on attributes and physics
  - Render particles

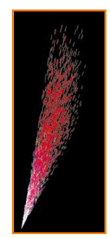
Particle Systems

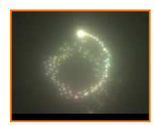
- For each frame:
  - Ø Create new particles and assign attributes
  - Ø Delete any expired particles
  - Update particles based on attributes and physics
  - Render particles

Creating/Deleting Particles

- Where to create particles?
  - Around some center
  - Along some path
  - Surface of shape
  - Where particle density is low
- When to delete particles?
  - Where particle density is high
  - Life span
  - Random

This is where user controls animation





Example: Wrath of Khan

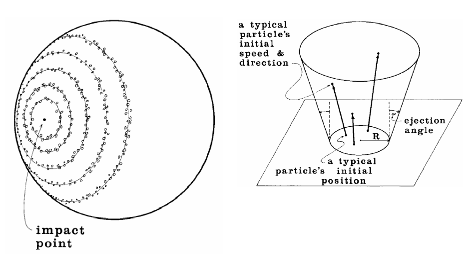
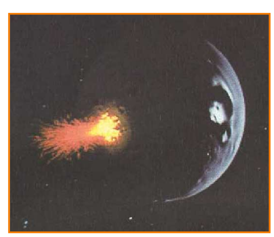


Fig. 2. Distribution of particle systems on the planet's surface.

Reeves

Example: Wrath of Khan



Reeves

Example: Wrath of Khan

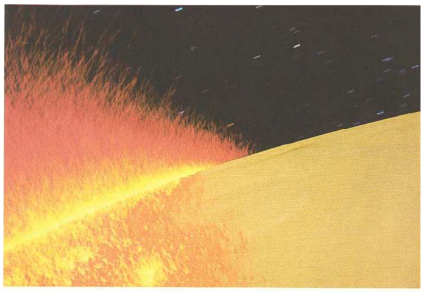


Fig. 7. Wall of fire about to engulf camera.

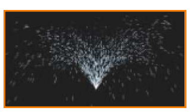
Reeves

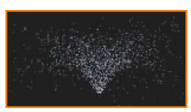
Particle Systems

- For each frame:
  - Create new particles and assign attributes
  - Delete any expired particles
  - Update particles based on attributes and physics
  - Ø Render particles

Rendering Particles

- Volumes
  - Ray casting, etc.
- Points
  - Render as individual points
- Line segments
  - Motion blur over time





Example: Fire

Allan

Particle Systems

- For each frame:
  - Create new particles and assign attributes
  - Delete any expired particles
  - Ø Update particles based on attributes and physics
  - Render particles

Equations of Motion

- Newton's Law for a point mass
  - f = ma
- Computing particle motion requires solving second-order differential equation

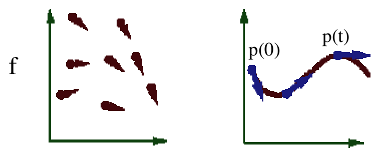
$$\ddot{x} = \frac{f(x, \dot{x}, t)}{m}$$

- Add variable v to form coupled first-order differential equations

$$\begin{cases} \dot{x} = v \\ \dot{v} = \dfrac{f}{m} \end{cases}$$

Solving the Equations of Motion

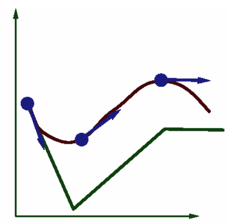
- Initial value problem
  - Know p(0), v(0), a(0)
  - Can compute force at any time and position
  - Compute p(t) by forward integration



Hodgins

Solving the Equations of Motion

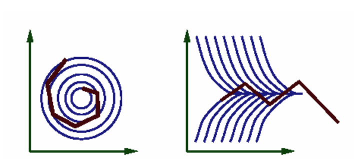
- Euler integration
  - $p(t+\Delta t)=p(t) + \Delta t\ f(x,t)$



Hodgins

Solving the Equations of Motion

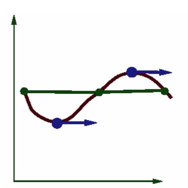
- Euler integration
  - $p(t+\Delta t)=p(t) + \Delta t\ f(x,t)$
- Problem:
  - Accuracy decreases as Δt gets bigger



Hodgins

Solving the Equations of Motion

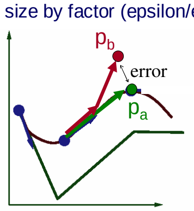
- Midpoint method (2nd order Runge-Kutta)
  - Compute an Euler step
  - Evalute f at the midpoint
  - Take an Euler step using midpoint force
    - » $p(t+\Delta t)=p(t) + \Delta t\ f(\ p(t) + 0.5*\Delta t\ f(t),t)$



Hodgins

Solving the Equations of Motion

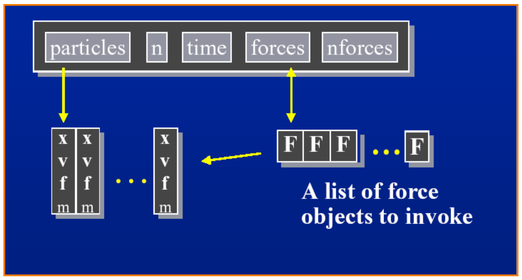
- Adapting step size
  - Compute p_a by taking one step of size h
  - Compute p_b by taking 2 steps of size h/2
  - Error = $|\ p_a\ -\ p_b\,|$
  - Adjust step size by factor (epsilon/error)$^{1/f}$



Particle System Forces

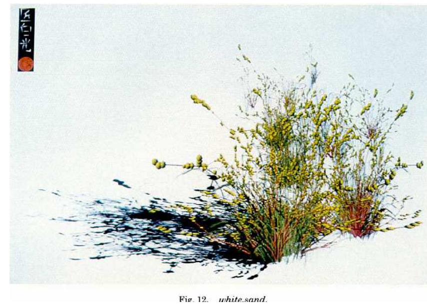
- Force fields
  - Gravity, wind, pressure
- Viscosity/damping
  - Liquids, drag
- Collisions
  - Environment
  - Other particles
- Other particles
  - Springs between neighboring particles (mesh)
  - Useful for cloth

Particle System Forces



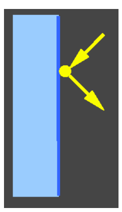
Witkin

Example: Grass in Wind



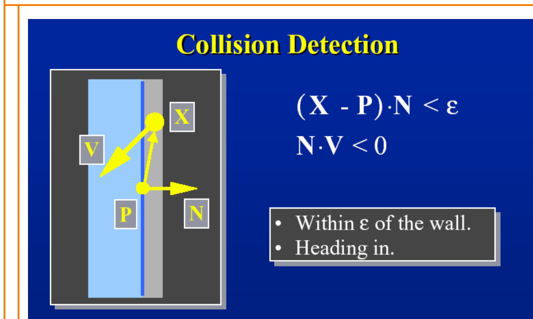
Example: Bouncing Off Wall

- Requires
  - Collision detection
  - Collision response (dynamic forces)



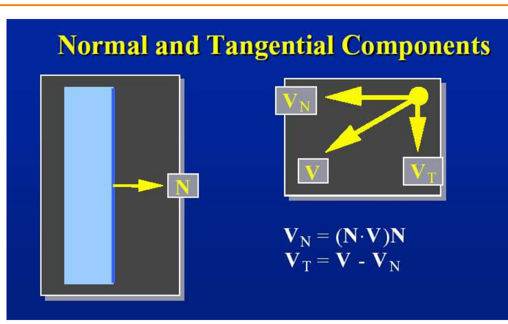
Witkin

Example: Bouncing Off Wall



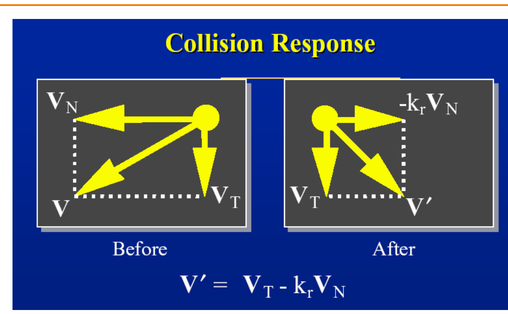
Witkin

Example: Bouncing Off Wall



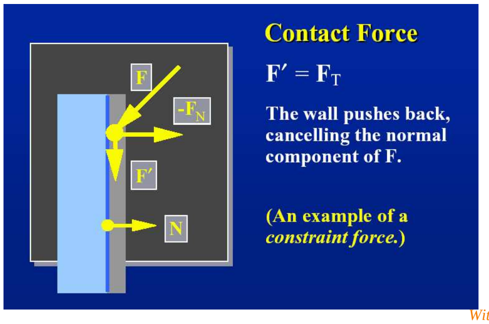
Witkin

Example: Bouncing Off Wall



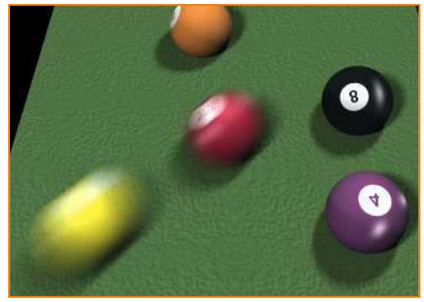
Witkin

Example: Bouncing Off Wall



Witkin

Example: Pool Balls



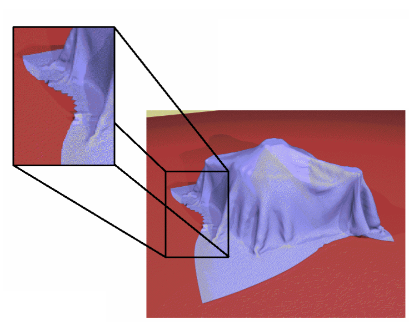
Example: Complex Objects

Allan

Example: Complex Objects

Allan

Example: Cloth



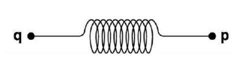
Breen

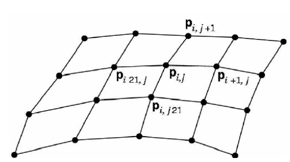
Example: Cloth

- Spring-mass mesh
- Hooke's law

$$\mathbf{f} = -k_s(|\mathbf{d}| - s)\frac{\mathbf{d}}{|\mathbf{d}|}$$

f = force
k_s = spring constant
d = p - q
s = resting length

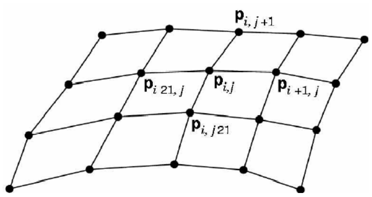


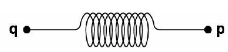


Hodgins

Example: Cloth

- Spring-mass mesh





Hodgins

Example: Cloth

- Hooke's law

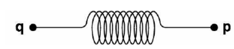
$$\mathbf{f} = -k_s(|\mathbf{d}| - s)\frac{\mathbf{d}}{|\mathbf{d}|}$$

f = force
k_s = spring constant
d = p - q
s = resting length

- Damping term

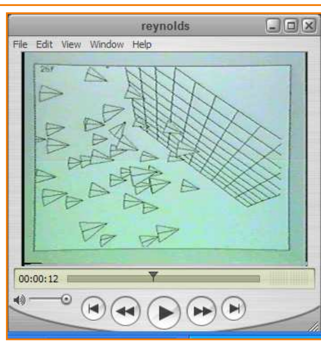
$$\mathbf{f} = -\left(k_s(|\mathbf{d}| - s) + k_d\frac{\dot{\mathbf{d}} \cdot \mathbf{d}}{|\mathbf{d}|}\right)\frac{\mathbf{d}}{|\mathbf{d}|}$$

$$\dot{\mathbf{d}} = \dot{\mathbf{p}} - \dot{\mathbf{q}}$$



Hodgins

Example: Flocks & Herds



Reynolds

Example: Flocks & Herds

Hodgins

Summary

- Particle systems
  - Lots of particles
  - Simple physics
- Interesting behaviors
  - Waterfalls
  - Smoke
  - Cloth
  - Flocks
- Solving motion equations
  - Simplest method is Euler integration
  - Better to use adaptive step sizes