CS130: Computer Graphics

Animation

Tamar Shinar

Computer Science & Engineering

UC Riverside
Types of animation

- keyframing
- rotoscoping
- stop motion
- procedural
- simulation
- motion capture
Gertie the Dinosaur

1914
12 minutes
hand drawn
keyframe animation
registration
cycling

link
Traditional animation

Cels

Multiplane camera

Sleeping Beauty, Disney, 1959

[Wikipedia Commons]
Realistic 3D animation

- Disney’s Tron, 1981
- Pixar’s Toy Story, 1995, first 3D feature
Performance capture

Rise of the Planet of the Apes, 2011

Lord of the Rings, 2001

Avatar, 2009

Andy Serkis – Gollum, Lord of the Rings challenges – resolution, occlusion,
Paperman and the Future of 2D Animation

trailer: http://www.youtube.com/watch?v=mM6cLnscmO8

making of: http://www.youtube.com/watch?v=TZJLtujW6FY&feature=youtube_gdata_player
animation principles
- animation can bring even a flour sack to life
- animations principles common to any type of animation
12 principles of animation

1. Squash and stretch
2. Anticipation
3. Staging
4. Straight ahead action and pose to pose
5. Follow through and overlapping action
6. Slow in and slow out
7. Arcs
8. Secondary action
9. Timing
10. Exaggeration
11. Solid drawing
12. Appeal

principles are related to the underlying physics of motion timing: important information. ease in/ease out
Physics-based animation

- Many animation principles follow from underlying physics
- anticipation, follow through, secondary action, squash and stretch, ...
- Spacetime Constraints, Witkin and Kass 1988
Physics-based animation

- Many animation principles follow from underlying physics
  - anticipation, follow through, secondary action, squash and stretch, ...

keyframe animation
Keyframe animation

- draw a series of poses
- fill in the frames in between ("inbetweening")
- computer animation uses interpolation

http://anim.tmog.net
Luxo Jr.
- forward kinematics – set joint angles starting at a root and working down the tree
- inverse kinematics – set end effector (e.g., hand) and solve for state of dofs up to root
forward kinematics

[Shirley and Marschner]

inverse kinematics

IK solver connection

Effector motion

hip and knee joint angles computed automatically

[Shirley and Marschner]
forward kinematics

inverse kinematics

multiple possible states of joints

Eccentric motion

invers kinematics

[Shirley and Marschner]
Keyframe character DOFs

3 translational DOFs

48 rotational DOFs

Each joint can have up to 3 DOFs
Interpolation of keyframes

linear interpolation

spline interpolation

Straightforward to interpolate position but what about orientation?
need to consider both shape of motion and speed of motion

[Shirley and Marschner]
Character Skinning

[McAdams et al. 2011]
Character Skinning

[McAdams et al. 2011]
free form deformation

[Sederberg 1986]

[Shirley and Marschner]
facial animation
Facial animation
Facial animation
procedural animation
Artificial life

• plants - movement and growth
• evolving artificial life

virtual worlds, special effects, games
Crowd simulation

- agent-based, model behavior
- also, “global effects” – e.g., incompressibility
- emergent phenomena

[Treuille et al. 2006]
physics-based animation
Particles
Particle: basic dynamic object
Particle: basic dynamic object

mass \( m \)
Particle: basic dynamic object

mass \( m \)

3 dof

\[ \vec{x} = (x, y, z) \]
Particle: basic dynamic object

- mass: $m$
- 3 dof
- $\vec{X} = (x, y, z)$
- forces: e.g., gravity

$\vec{F} = -m\vec{g}$
Particle: basic dynamic object

Equations of motion:
Newton’s 2nd Law

\[ \vec{F} = m\vec{a} \]
Particle: basic dynamic object

Equations of motion: Newton’s 2nd Law

\[ \vec{F} = m \vec{a} \]

\[ \frac{dx}{dt} = \vec{v} \]

\[ m \frac{d\vec{v}}{dt} = \vec{F} \]

System of ODEs
Deformable bodies
Connect a bunch of particles into a 1D line segment with springs
A Mass Spring Model for Hair Simulation

Connect a bunch of particles into a 2D mesh
Connect a bunch of particles into a 3D mesh
Deformable bodies: equations of motion

Equations of motion:
Newton’s 2nd Law

\[ \vec{F} = m\vec{a} \]

\[ \frac{d\vec{x}}{dt} = \vec{v} \]

\[ m\frac{d\vec{v}}{dt} = \vec{F} \]

System of PDEs
contains spatial derivatives
Rigid bodies
• Limit of infinite stiffness – rigid approximations encapsulates the constitutive model
  • Body now has 6 degrees of freedom – 3 for position and 3 for orientation
• Forces acting on the body result both in a net force and net torque on the body
• Must handle elastic collisions – by defining a coefficient of restitution
  • Deformable objects undergo inelastic collisions and store energy at the collisions which then causes the bounce
• Resulting evolution equations are a system of ODEs
• Typically integrated with explicit methods
  • No time step restriction for stability since the rigid approximation means information through the body is propagated instantaneously
  • i.e., we don’t need to account for finite propagation of information through a mesh in computing the time step
Rigid bodies

6 dofs

• Limit of infinite stiffness – rigid approximations encapsulates the constitutive model
  • Body now has 6 degrees of freedom – 3 for position and 3 for orientation
• Forces acting on the body result both in a net force and net torque on the body
• Must handle elastic collisions – by defining a coefficient of restitution
  • Deformable objects undergo inelastic collisions and store energy at the collisions which then causes the bounce
• Resulting evolution equations are a system of ODEs
• Typically integrated with explicit methods
  • No time step restriction for stability since the rigid approximation means information through the body is propagated instantaneously
  • i.e., we don’t need to account for finite propagation of information through a mesh in computing the time step
Rigid bodies

6 dofs
forces and torques

\[(\mathbf{X}, \mathbf{\Omega})\]
\[(\mathbf{F}, \mathbf{\tau})\]

• Limit of infinite stiffness – rigid approximations encapsulates the constitutive model
  • Body now has 6 degrees of freedom – 3 for position and 3 for orientation
• Forces acting on the body result both in a net force and net torque on the body
• Must handle elastic collisions – by defining a coefficient of restitution
  • Deformable objects undergo inelastic collisions and store energy at the collisions which then causes the bounce
• Resulting evolution equations are a system of ODEs
• Typically integrated with explicit methods
  • No time step restriction for stability since the rigid approximation means information through the body is propagated instantaneously
  i.e., we don’t need to account for finite propagation of information through a mesh in computing the time step
Rigid bodies

6 dofs
forces and torques
elastic collisions

\( (\vec{X}, \vec{\Omega}) \)

\( (\vec{F}, \vec{\tau}) \)

• Limit of infinite stiffness – rigid approximations encapsulates the constitutive model
  • Body now has 6 degrees of freedom – 3 for position and 3 for orientation
• Forces acting on the body result both in a net force and net torque on the body
• Must handle elastic collisions – by defining a coefficient of restitution
  • Deformable objects undergo inelastic collisions and store energy at the collisions which then causes the bounce
• Resulting evolution equations are a system of ODEs
• Typically integrated with explicit methods
  • No time step restriction for stability since the rigid approximation means information through the body is propagated instantaneously
  • i.e., we don’t need to account for finite propagation of information through a mesh in computing the time step
Rigid bodies

• Limit of infinite stiffness – rigid approximations encapsulates the constitutive model
  • Body now has 6 degrees of freedom – 3 for position and 3 for orientation
• Forces acting on the body result both in a net force and net torque on the body
• Must handle elastic collisions – by defining a coefficient of restitution
  • Deformable objects undergo inelastic collisions and store energy at the collisions which then causes the bounce
• Resulting evolution equations are a system of ODEs
• Typically integrated with explicit methods
  • No time step restriction for stability since the rigid approximation means information through the body is propagated instantaneously
  • i.e., we don’t need to account for finite propagation of information through a mesh in computing the time step

6 dofs

forces and torques

elastic collisions

ODEs

\[(\vec{X}, \vec{\Omega})\]

\[(\vec{F}, \vec{\tau})\]
Rigid body phenomena

• One of the main challenges in rigid body simulation is collisions vs. contact
  • Special care to make sure that things in sliding contact don’t bounce
  • But that things get a chance to bounce before sticking
• Getting a stable stack is also hard
  • Shock propagation
• Friction between rigid bodies is harder than deformable
  • Actually takes place over some finite time step $dt$ and along nonlinear path, i.e., the body will rotate due to friction
  • Should actually be an integral – first order approximation can cause weirdness
• Articulation – very useful for creatures!
  • Velocity constraints plus drift correction
  • We do pd control similarly
• Also fracture
Rigid body phenomena

stacking

• One of the main challenges in rigid body simulation is collisions vs. contact
  • Special care to make sure that things in sliding contact don’t bounce
  • But that things get a chance to bounce before sticking
• Getting a stable stack is also hard
  • Shock propagation
• Friction between rigid bodies is harder than deformable
  • Actually takes place over some finite time step $dt$ and along nonlinear path, i.e., the body will rotate due to friction
  • Should actually be an integral – first order approximation can cause weirdness
• Articulation – very useful for creatures!
  • Velocity constraints plus drift correction
  • We do pd control similarly
• Also fracture
One of the main challenges in rigid body simulation is collisions vs. contact
  - Special care to make sure that things in sliding contact don’t bounce
  - But that things get a chance to bounce before sticking

Getting a stable stack is also hard
  - Shock propagation

Friction between rigid bodies is harder than deformable
  - Actually takes place over some finite time step \( dt \) and along nonlinear path, i.e., the body will rotate due to friction
  - Should actually be an integral – first order approximation can cause weirdness

Articulation – very useful for creatures!
  - Velocity constraints plus drift correction
  - We do pd control similarly

Also fracture
Rigid body phenomena

• One of the main challenges in rigid body simulation is collisions vs. contact
  • Special care to make sure that things in sliding contact don't bounce
  • But that things get a chance to bounce before sticking

• Getting a stable stack is also hard
  • Shock propagation

• Friction between rigid bodies is harder than deformable
  • Actually takes place over some finite time step \( dt \) and along nonlinear path, i.e., the body will rotate due to friction
  • Should actually be an integral – first order approximation can cause weirdness

• Articulation – very useful for creatures!
  • velocity constraints plus drift correction
  • We do pd control similarly

• Also fracture
Rigid body phenomena

stacking

collisions, contact

friction

articulation, control

• One of the main challenges in rigid body simulation is collisions vs. contact
  • Special care to make sure that things in sliding contact don’t bounce
  • But that things get a chance to bounce before sticking
• Getting a stable stack is also hard
  • Shock propagation
• Friction between rigid bodies is harder than deformable
  • Actually takes place over some finite time step dt and along nonlinear path, i.e., the body will rotate due to friction
  • Should actually be an integral – first order approximation can cause weirdness
• Articulation – very useful for creatures!
  • velocity constraints plus drift correction
  • We do pd control similarly
• Also fracture
Articulated rigid bodies

Rachel Weinstein, Joey Teran and Ron Fedkiw
Articulated rigid bodies

Rachel Weinstein, Joey Teran and Ron Fedkiw
Rigid body simulation

[Weinstein et al 2006]
Rigid and deformable solids coupled together...

* Rigid body examples, etc.
Rigid and deformable solids coupled together...

* Rigid body examples, etc.
Fracture

[Molino et al. 2004]
Contact and collision
Simultaneous resolution of contact, elastic deformation, articulation constraints
Simultaneous resolution of contact, elastic deformation, articulation constraints
our rigid/deformable simulator in Pixar’s WALL-E
our rigid/deformable simulator in Pixar’s WALL-E
Fluid simulation
In fluid simulation, we often use a grid-based representation. The level set function may be e.g., Heaviside, signed distance.
Fluid equations of motion: Navier-Stokes equations

\[ \vec{F} = m\vec{a} \]

\[ \rho(u_t + u \cdot \nabla u) = \mu \Delta u - \nabla p + \vec{f} \]
A Vortex Particle Method for Smoke, Water and Explosions
Selle, A., Rasmussen, N. and Fedkiw, R.
A Vortex Particle Method for Smoke, Water and Explosions
Selle, A., Rasmussen, N. and Fedkiw, R.
- Navier–Stokes equations for viscous, incompressible flow
- we can use multiple pls and previous work for boundary condition capturing at interfaces to simulate many different liquids interacting
- Navier–Stokes equations for viscous, incompressible flow
- we can use multiple pls and previous work for boundary condition capturing at interfaces to simulate many different liquids interacting
- Couple to Navier-Stokes based fluid simulator
- Wrinkling and cellular patterns in flame front
• Couple to Navier-Stokes based fluid simulator
• Wrinkling and cellular patterns in flame front
– other surface phenomena, such as fire (one material converting into another)
- other surface phenomena, such as fire (one material converting into another)
Two-way Coupled SPH and Particle Level Set Fluid Simulation

Control of virtual character

[Shinar et al. 2008]
Control of virtual character

[Shinar et al. 2008]

issues: control algorithms, interaction with environment
rigid/deformable simulator in Pixar’s WALL-E
rigid/deformable simulator in Pixar’s WALL-E