Lecture 28 of 41

Collision Handling Part 2 of 2: Dynamic Collision Response, Particle Systems

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course pages: http://bit.ly/hGvXlH / http://bit.ly/eVizrE
Public mirror web site: http://www.kddresearch.org/Courses/CIS636
Instructor home page: http://www.cis.ksu.edu/~bhsu

Readings:

Last class: §8.3 – 8.4, 4.2, 5.0, 5.6, 9.1 , Eberly 2e – see http://bit.ly/ieUq45
Today: §9.1, Eberly 2e; Particle System Handout
Next class: Particle System Handout
Wikipedia, *Particle System*: http://bit.ly/hzZofl

Lecture Outline

- Reading for Last Class: §8.3 – 8.4, 4.2, 5.0, 5.6, 9.1 , Eberly 2e
- Reading for Today: §9.1, Eberly 2e; Particle System Handout
- Reading for Next Class: Particle System Handout
- Last Time: Interaction Handling
  - Human-Computer Interaction (HCI)
  - Perceptual Principles: Legibility, Consistency, Redundancy
  - Mental Models: Realism, User Expectations
  - Attention: Access Cost/Benefit, Multiple Sources, Sensory Modes
  - Memory: Self-Explanatory GUIs, Predictive Aids, Reusable Skills
- Today: Collision Response
- Today & Next Class: Particle Systems
  - Simulation of Processes, Simple Physical Bodies
  - Events: birth (emission), collision, death
  - Properties: mass, initial velocity, lifetime
- Next: Lab on Particle Systems; Dissection of Working Program

Where We Are

21	Lab 4a: Animation Basics	Flash animation handout
22	Animation 2: Rotations; Dynamics, Kinematics	Chapter 17, esp. §17.1 – 17.2
23	Demos 4: Modeling & Simulation; Rotations	Chapter 10[1], 13[2], §17.3 – 17.5
24	Collisions 1: axes, OBBs, Lab 4b	§2.4.3, 8.1, GL handout
25	Spatial Sorting: Binary Space Partitioning	Chapter 6, esp. §6.1
26	Demos 5: More CGA; Picking; HW/Exam	Chapter 7[2]; § 8.4
27	Lab 5a: Interaction Handling	§ 8.3 – 8.4; 4.2, 5.0, 5.6, 9.1
28	Collisions 2: Dynamic, Particle Systems	§ 9.1, particle system handout
	Exam 2 review; Hour Exam 2 (evening)	Chapters 5 – 6, 7[2] – 8, 12, 17
29	Lab 5b: Particle Systems	Particle system handout
30	Animation 3: Control & IK	§ 5.3, CGA handout
31	Ray Tracing 1: intersections, ray trees	Chapter 14
32	Lab 6a: Ray Tracing Basics with POV-Ray	RT handout
33	Ray Tracing 2: advanced topic survey	Chapter 15, RT handout
34	Visualization 1: Data (Quantities & Evidence)	Tufte handout (1)
35	Lab 6b: More Ray Tracing	RT handout
36	Visualization 2: Objects	Tufte handout (2 & 4)
37	Color Basics; Term Project Prep	Color handout
38	Lab 7: Fractals & Terrain Generation	Fractals/Terrain handout
39	Visualization 3: Processes; Final Review 1	Tufte handout (3)
40	Project presentations 1; Final Review 2	–
41	Project presentations 2	–
	Final Exam	Ch. 1 – 8, 10 – 15, 17, 20

Lightly-shaded entries denote the due date of a written problem set; heavily-shaded entries, that of a machine problem (programming assignment); blue-shaded entries, that of a paper review; and the green-shaded entry, that of the term project.

Green, blue and red letters denote exam review, exam, and exam solution review dates.

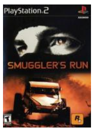
Acknowledgements: Picking, Interaction, Particles

François Guimbretière
Associate Professor
Department of Computer Science
Cornell University
http://www.cs.cornell.edu/~francois/

Cornell University Department of Computer Science

Hubert Pak Ho Shum
Postdoctoral Researcher
Advanced Science Institute
RIKEN (理研)
http://hubertshum.com/info/

RIKEN



Steve Rotenberg
Visiting Lecturer
Graphics Lab
University of California – San Diego
CEO/Chief Scientist, PixelActive
http://graphics.ucsd.edu

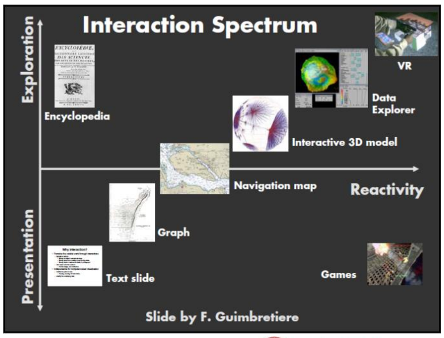
UCSD

Review [1]: Human-Computer Interaction (HCI)

- Study, Planning, & Design of Interaction between People, Computers
  - Study: intersection of computer science, behavioral science, design
  - Planning: information management tasks, media
  - Design: graphical user interfaces, displays, algorithms, systems
- Related Areas
  - Cognitive Science & Cognitive Modeling
  - Ergonomics & Human Factors
    - HCI: more emphasis on computers (*vs.* other artifacts)
    - Some overlap within information technology
  - Computational Information & Knowledge Management (CIKM)
  - Software Engineering: Operating Systems
  - Computer Graphics
  - Wearable Computers & Ubiquitous Communication/Computing

Adapted from Wikipedia, *Human-Computer Interaction*
http://bit.ly/bqrQTg

Review [2]: Interaction Spectrum for HCI & Print



Adapted from slide © 2004 F. Guimbreti\re, Cornell University
Stanford CS448B: Visualization, Fall 2004, http://bit.ly/h0hRzU

Cornell University Department of Computer Science

7

Review [3]: Perception & Mental Models in HCI

- Perceptual Principles
  - 1. Make Displays Perceivable (Legible, Audible, *etc.*)
  - 2. Avoid Absolute Judgment Limits
  - 3. Use Top-Down Processing: Be Consistent with Past Experience
  - 4. Exploit Redundancy
  - 5. Use Discriminable Elements to Minimize Confusion
- Principles Based upon Mental Models
  - 6. Maintain Pictorial Realism
  - 7. Follow User's Expectations Regarding Moving Parts

Adapted from material ♥ 2008 Wikimedia Foundation (from Wickens *et al.*, 2004)
Human-Computer Interaction, http://bit.ly/bqrQTg

8

Review [4]: Attention & Memory in HCI

- Attention-Based Principles
  - 8. Minimize Information Access Cost
  - 9. Put Multiple Information Sources Close Together, Integrate Them
  - 10. Make Use of Multiple Information Channels
- Memory-Based Principles
  - 11. Replace Memory with Visual Information
  - 12. Develop Methods for Predictive Aiding
  - 13. Ensure Consistency

Adapted from material ♥ 2008 Wikimedia Foundation (from Wickens *et al.*, 2004)
Human-Computer Interaction, http://bit.ly/bqrQTg

9

Review [5]: Particle Systems – Basic Model

A particle system is a collection of many minute particles that model some object. For each frame of an animation sequence the following steps are performed:

1. New particles are generated
2. Each new particle is assigned its own set of attributes
3. Any particles that have existed for a predetermined time are destroyed
4. The remaining particles are transformed and moved according to their dynamic attributes
5. An image of the remaining particles is rendered

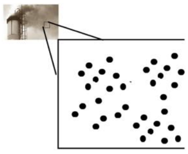
Adapted from slide ♥ 2008 H. P. H. Shum, RIKEN (理研)
Computer Animation, http://bit.ly/ig6KTK

10

Review [6]: Particle Emitters & Attributes

Each new particle has the following attributes:

- initial position
- initial velocity (speed and direction)
- initial size
- initial color
- initial transparency
- shape
- lifetime

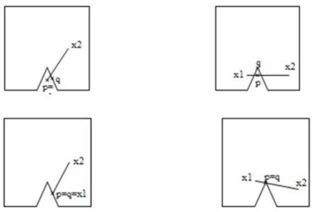


Adapted from slide ♥ 2008 H. P. H. Shum, RIKEN (理研)
Computer Animation, http://bit.ly/ig6KTK

11

Review [7]: Collision Detection Redux

- The collision detection will be done based on the points and the surfaces
- The colliding particle will be pushed out from the penetrating area
- The velocity of the colliding particle will be reduced to zero

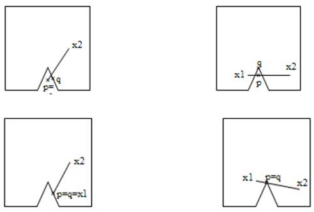


Adapted from slide ♥ 2008 H. P. H. Shum, RIKEN (理研)
Computer Animation, http://bit.ly/ig6KTK

12

Review [9]: Collision Detection Redux

- The collision detection will be done based on the points and the surfaces
- The colliding particle will be pushed out from the penetrating area
- The velocity of the colliding particle will be reduced to zero



13

Review [8]: Collisions – Detection vs. Response

- Collision Detection
  - Collision detection is a geometric problem
  - Given two moving objects defined in an initial and final configuration, determine if they intersected at some point between the two states
- Collision Response
  - The response to collisions is the actual physics problem of determining the unknown forces (or impulses) of the collision

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

14

Review [9]: Collision Detection

- Goals
  - Given: two objects with current & previous orientations specified
  - Determine: if, where, when they intersect
- Alternative (Static)
  - Given: two objects with current orientations specified
  - Determine: if they intersect
- Variants
  - Static: stationary objects (both not moving)
  - Dynamic: moving objects (one or both)
  - Queries
    - Test-intersection: determine whether objects do/will intersect
    - Find-intersection: calculate intersection set or contact set, time
  - Parametric methods: use parameters to describe objects
    - Distance-based: constrained minimization (closest points)
    - Intersection-based: solving for parameters in equation

15

Impact vs. Contact [1]: Distinction

- In physics simulation, there is usually a distinction between impacts and contacts
- Impacts are instantaneous collisions between objects where an impulse must be generated to prevent the velocities at the impact location from allowing the objects to interpenetrate
- Contacts are persistent and exist over some range of time. In a contact situation, the closing velocities at the contact location should already be 0, so forces are needed to keep the objects from accelerating into each other. With rigid bodies, contacts can include fairly complex situations like stacking, rolling, and sliding

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

16

Impact vs. Contact [2]: Handling

- Neither impact nor contact is particularly easy to handle correctly
- In the case of particles, it's not so bad, but with rigid bodies, it can be tough
- As we are mainly just concerned with the physics of particles, we will not worry about the more complex issues for now
- Also, we will just focus on handling impacts, as they are generally needed first. Continuous contact will just be handled by allowing particles to impact frame after frame

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

17

Impacts

- When two solid objects collide (such as a particle hitting a solid surface), forces are generated at the impact location that prevent the objects from interpenetrating
- These forces act over a very small time and as far as the simulation is concerned, it's easiest to treat it as an instantaneous event
- Therefore, instead of the impact applying a force, we must use an impulse

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

18

Impulse

- An impulse can be thought of as the integral of a force over some time range, which results in a finite change in momentum:

$$\mathbf{j} = \int \mathbf{f}\,dt = \Delta\mathbf{p}$$

- An impulse behaves a lot like a force, except instead of affecting an object's acceleration, it directly affects the velocity
- Impulses also obey Newton's Third Law, and so objects can exchange equal and opposite impulses
- Also, like forces, we can compute a total impulse as the sum of several individual impulses

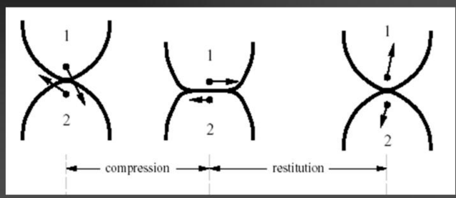
Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

19 Compression & Restitution [1]: Definition, Elasticity of Collision

- The collision can be thought of as having two phases: compression & restitution
- In the compression phase, the energy of the two objects is changed from kinetic energy of motion into deformation energy in the solids
- If the collision is perfectly *inelastic* (e=0), then all of the energy is lost and there will be no relative motion along the collision normal after the collision
- If the collision is perfectly *elastic* (e=1), then all of the deformation energy will be turned back into kinetic energy in the restitution phase and the velocity along the normal will be the opposite of what it was before the collision

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

20 Compression & Restitution [2]: Illustration



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

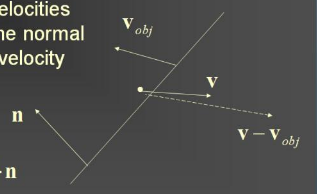
21 Calculations [1]: Impulse for Particle-Object Case

- Consider the case of a particle colliding with a heavy object. The object is moving with velocity $\mathbf{v}_{obj}$
- The particle has a velocity of **v** before the collision and collides with the surface with a unit normal **n**
- We want to find the collision impulse **j** applied to the particle during the collision

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

22 Calculations [2]: Final Velocity

- We take the difference between the two velocities and dot that with the normal to find the closing velocity



$$v_{close} = \left(\mathbf{v} - \mathbf{v}_{obj}\right)\cdot\mathbf{n}$$

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

23 Calculations [3]: Impulse given Velocity (Frictionless)

- Let's first consider a collision with no friction
- The collision impulse will be perpendicular to the collision plane (i.e., along the normal) and will be large enough to stop the particle (at least)

$$\mathbf{j} = -(1+e)mv_{close}\mathbf{n}$$

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

24 Friction [1]: Coulomb Friction Model

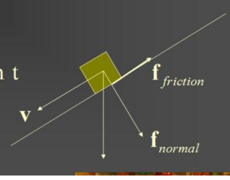
- The Coulomb friction model says:

$$\mathbf{f}_{dynamic} = \mu_d\left|\mathbf{f}_{normal}\right|\mathbf{e}$$

$$\mathbf{f}_{static} \leq \mu_s\left|\mathbf{f}_{normal}\right|\mathbf{e}$$

μ_d : dynamic friction coefficient

μ_s : static friction coefficient



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

25

Friction [2]: Dynamic Friction Equation

- As we are not considering static contact, we will just use a single dynamic friction equation
- For an impact, we can just compute the impulse in the exact same way as we would for dynamic friction
- We can use the magnitude of the elasticity impulse as the normal impulse

$$\mathbf{j}_{dynamic} = \mu_d |\mathbf{j}_{normal}| \mathbf{e}$$

26

Collision Handling

- For particles and cloth, the following approach works effectively:

1. Compute forces (springs, aero…)
2. Integrate motion (Euler step)
3. Test if particles hit anything
   - 3.1 Compute & apply impulse (adjust velocity)
   - 3.2 Adjust position

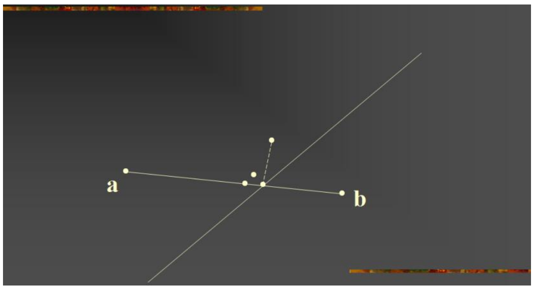
27

Position Adjustment [1]: Options Defined

- Moving the particle to a legal position isn't always easy
- There are different possibilities:
  - Move it to a position just before the collision
  - Put it at the collision point
  - Put it at the collision point plus some offset along the normal
  - Compute where it would have gone if it bounced
- Computing the bounced position is really the best, but may involve more computation and in order to do it right, it may require further collision testing…

28

Position Adjustment [2]: Options Illustrated



29

Bouncing

- Computing the bounced position is the best approach, as it is consistent with the rest of the physics model
- We need to determine when exactly the collision happened (we can just assume that the particle traveled at a constant velocity within the frame)
- We then compute the impulse and adjust the velocity
- Then, we move the particle forward by the amount of time remaining within the frame
- Ideally, we should then check collisions on this new path
- A particle getting stuck in a narrow crack might bounce several times, so we should put a cap on the maximum number of bounces allowed, then just stop the particle at some point if it exceeds the limit

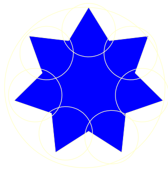
30

Collision Optimization [1]: Overview of Data Structures

- BV, BVH (bounding volume hierarchies)
  - Octree
  - KD tree
  - BSP (binary separating planes)
  - OBB tree (oriented bounding boxes- a popular form of BVH)
  - K-dop tree
- Uniform grid
- Hashing
- Dimension reduction

31

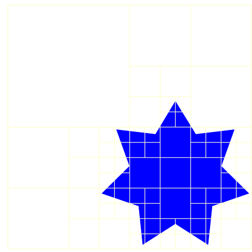
Bounding Volume Hierarchies Redux



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

32

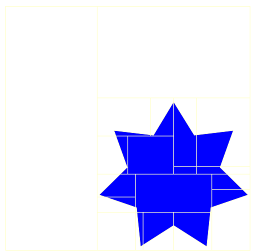
Adaptive Spatial Partitioning [1]: (Quadtrees & Octrees) Redux



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

33

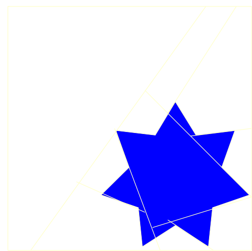
Adaptive Spatial Partitioning [2]: k-D Trees



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

34

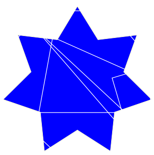
Adaptive Spatial Partitioning [3]: BSP Trees Redux



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

35

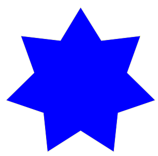
Adaptive Spatial Partitioning [4]: Oriented Bounding Box Trees



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

36

Adaptive Spatial Partitioning [5]: K-Discrete Orientation Polytopes



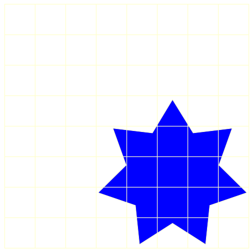
Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

Testing BVHs

```
TestBVH(A,B) {
    if(not overlap(A_BV, B_BV) return FALSE;
    else if(isLeaf(A)) {
        if(isLeaf(B)) {
            for each triangle pair (T_a,T_b)
                if(overlap(T_a,T_b)) AddIntersectionToList();
        }
        else {
            for each child C_b of B
                TestBVH(A,C_b);
        }
    }
    else {
        for each child C_a of A
            TestBVH(C_a,B)
    }
}
```

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

Uniform Grids



Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

Collision Optimization [2]: 2-D vs. 3-D Optimization Structures

- All of these optimization structures can be used in either 2D or 3D
- Packing in memory may affect caching and performance

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

Collision Optimization [3]: Pair Reduction

- At a minimum, any moving object should have some sort of bounding sphere (or other simple primitive)
- Before a pair of objects is tested in any detail, we can quickly test if their bounding spheres intersect
- When there are lots of moving objects, even this quick bounding sphere test can take too long, as it must be applied N^2 times if there are N objects
- Reducing this N^2 problem is called *pair reduction*
- Pair testing isn't a big issue until N>50 or so…

Adapted from slides ♥ 2004 – 2005 S. Rotenberg, UCSD
CSE169: Computer Animation, Winter 2005 , http://bit.ly/f0ViAN

Summary

- **Reading for Last Class: §8.3 – 8.4, 4.2, 5.0, 5.6, 9.1 , Eberly 2e**
- **Reading for Today: §9.1, Eberly 2e; Particle System Handout**
- **Reading for Next Class: Particle System Handout**
- **Last Time: Interaction Handling & Human Computer Interaction (HCI)**
  - **Spectrum of interaction**
  - **Kinds of interaction**
    - **User input: selection, control**
    - **Stimuli: much more when we cover visualization, color**
- **Today: Collision Response**
- **Today & Next Class: Particle Systems**
  - **Collision response**
  - **Simulation of Processes, Simple Physical Bodies**
  - **Events: birth (emission), collision, death**
  - **Properties: mass, initial velocity, lifetime**
- **Next: Lab on Particle Systems; Dissection of Working Program**

Terminology

- **Human-Computer Interaction (HCI)**
  - ***aka* Man-Machine Interaction (archaic), Computer-Human Interaction**
  - **Study, planning, design of interaction between humans & computers**
  - **User interface: operation & control of machine, feedback to human**
- **Particle Systems – Simulation of Processes, Simple Physical Bodies**
  - **Events**
    - **Birth – particle generated based on shape, position of emitter**
    - **Collision – particle with object (including other particles)**
    - **Death – end of particle life, due to collision or expiration**
  - **Properties**
    - **Mass**
    - **Initial velocity**
    - **Lifetime**
    - ***etc.***
- **Data Structures: *k*-D Trees, K-Discrete Oriented Polytopes**