Computing and Software Department, McMaster University

Introduction to Collision Detection

John McCutchan

November 9, 2006

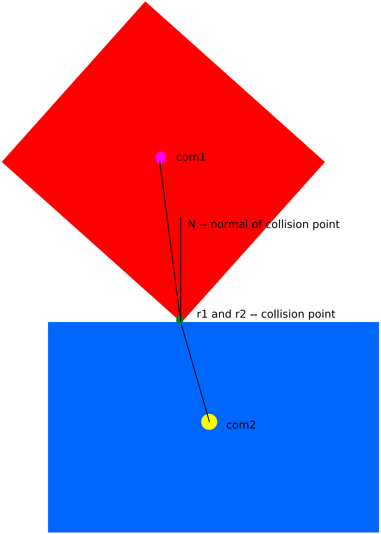
Introduction

- Collision Detection
- Terminology
- Narrow Phase
- Time Stepping
- Broad Phase

Introduction

- Invariant of a rigid body Simulator: no penetration of rigid bodies
- Collision detection system must compute the contact points and normals between each pair of rigid bodies
- As long as the velocity relative to the contact normal directions at the contact points are nonnegative the invariant will hold.

Introduction



Introduction

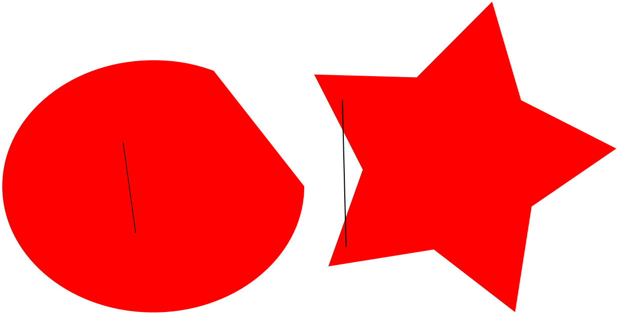
- $Vcollision_1 = v_1 + w_1 \times r_1$
- $Vcollision_2 = v_2 + w_2 \times r_2$
- $Vrelative = N \cdot (Vcollision_1 - Vcollision_2)$

Three cases need to be considered for *Vrelative*

- $Vrelative \geq 0$ – Objects are separating
- $Vrelative = 0$ – Objects are resting on each other
- $Vrelative \leq 0$ – Objects will penetrate

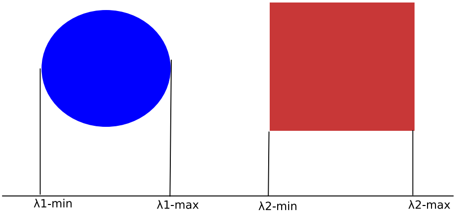
Terminology – Convex Polyhedra

A set C is convex if given two points the line segment connecting them is also in C.



Terminology – Separating Axis

A line is a separating axis of two convex sets C_1 and C_2 if the projection intervals $I_1 = [\lambda 1_{min}, \lambda 1_{max}]$ and $I_2 = [\lambda 2_{min}, \lambda 2_{max}]$ do not intersect. Specifically, $\lambda 1_{min} > \lambda 2_{max}$ or $\lambda 1_{max} < \lambda 2_{min}$



Narrow phase

Given two convex shapes we need to be able to determine two things.

- Test for intersection – Are the shapes intersecting
- Find intersection – Where are the shapes intersecting

Narrow phase – Test for intersection

Test for intersection: If a separating axis exists they are not in intersection.

```
foreach edge normal (EN) in Union(C1,C2) do:
    I1 = Projection(C1, EN);
    I2 = Projection(C2, EN);
    if Intersection(I1,I2) != EmptySet then:
        return true;
return false;
```

Narrow phase – Test for intersection

Projection:
$[\lambda_{min}, \lambda_{max}] = [min(D \cdot (X) : X \in C), max(D \cdot (X) : X \in C)]$

Narrow phase - Find Intersection

GJK – Gilbert, Johnson, and Keerthi distance algorithm.

GJK computes the distance between two convex shapes. It can also find the closest points on each object and the normal to the collision. GJK can also be used for ray casting.

$$[r1, r2] = GJKClosestPoints(C1, C2)$$

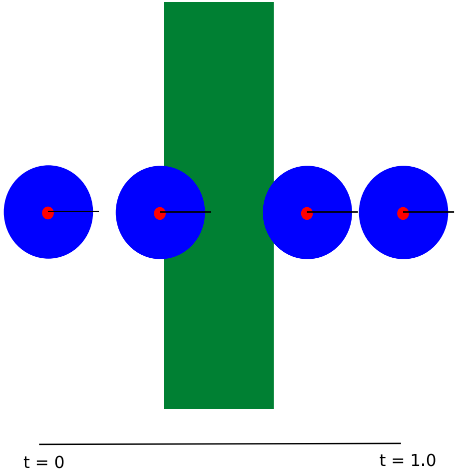
$$N = r1 - r2$$

$$Distance = length(N)$$

if $Distance < \epsilon$ then the objects have collided

Much More detail on Friday

Time stepping

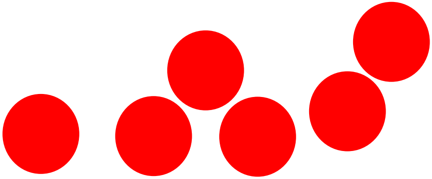


Time stepping

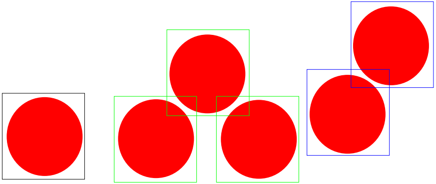
Problem: Need a method to predict the time that two convex shapes with constant linear velocity start to collide.

- Work around 1: keep rigid bodies velocities small
- Work around 2: keep time step size small
- Work around 3: allow for small amount of penetration between bodies
- Solution: GJK ray casting can determine when two convex objects with constant linear velocity and fixed rotation start to collide.

Broad Phase



Broad Phase



Broad Phase

Problem 1: Don't want to be naive and perform $O(n^2)$ collision checks
Solution 1: Keep a set of potentially overlapping bodies (Encounters)
Problem 2: How do we build this list of encounters quickly?
Solution 2: The Axis Aligned Bounding Box Sweep and Prune algorithm

Broad Phase - Sweep and Prune

- Sort end points – careful to handle the case where endpoints are equal but one is type begin and the other is type end
- Sweep over sorted end points – emit encounter when two intervals overlap

Broad Phase - Sweep

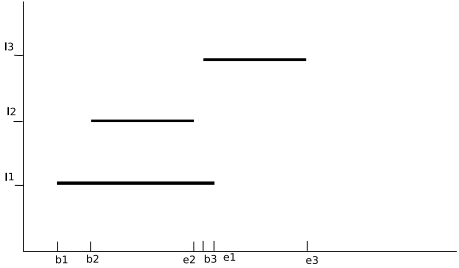
Sweep phase:

- Keep active set of intervals 'A'. Add interval to A when sweep encounters a 'begin' and remove interval from A when sweep encounters corresponding 'end'.
- When new interval is added to A emit an encounter between each member of A and the new interval.

Complexity: $O(nlogn)$ (sort) + $O(n+m)$ (interval overlap check) $= O(nlogn+m)$

Broad Phase - Sweep

Simplify problem by working with each axis (X,Y,Z) independently



Broad Phase - Sweep

$IL = [b1, b2, e2, b3, e1, e3]$

- $A = \emptyset$ and $S = \emptyset$
- b1 encountered: $A = [I1]$ and $S = \emptyset$
- b2 encountered: $A = [I1, I2]$ and $S = [(1, 2)]$
- e2 encountered: $A = [I1]$ and $S = [(1, 2)]$
- b3 encountered: $A = [I1, I3]$ and $S = [(1, 2), (1, 3)]$
- e1 encountered: $A = [I3]$ and $S = [(1, 2), (1, 3)]$
- e3 encountered: $A = \emptyset$ and $S = [(1, 2), (1, 3)]$

Broad Phase - Sweep

Problem 1: What happens when intervals change size and/or move?
Solution 1: Since interval list is already sorted insertion sort can be done in $O(n+e)$ where e is number of exchanges
Problem 2: Still need to perform sweep phase of algorithm after each update phase
Solution 2: Combine encounter set update with insertion sort!

Broad Phase – modified insertion sort

Four exchange cases need to be considered:

- b exchanged with b: Encounter set remains unchanged
- e exchanged with e: Encounter set remains unchanged
- b exchanged with e: Intervals end overlapping
- e exchanged with b: Intervals begin to overlap

See page 358-359 of text book for example of this.