Collision detection algorithms

Andrew Kickertz

16 April 2012

Contents

1 Introduction

- Bowling collision simulation can be broken down into 3 components:
  1. Translation and rotation of bodies
  2. Determining when bodies collide
  3. Determining where they go afterward
- This presentation is on collision detection. Various algorithms will be presented – from simple/fast/crude to complex/slow/accurate. The collision of 2 bowling pins will serve as the test case for most methods.
- Martin Baker's website [1] outlines many of the techniques discussed.

2 Bounding circles or spheres

This is the simplest test possible:

- Construct a sphere around the 2 bodies of interest
- Test if the spheres are touching:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2) + (z_1 - z_2)^2)} \quad =? \quad r_1 + r_2$$

, where x, y, and z locate the centers of spheres 1 and 2.

- This can be used in the `events` component of Matlab's `ode` suite:

```
function [value isTerminal direction] = ...
    events_spherecollision(t,x)

% Parse the state vector x into mass center coordinates
x1 = ...;
y1 = ...;
```

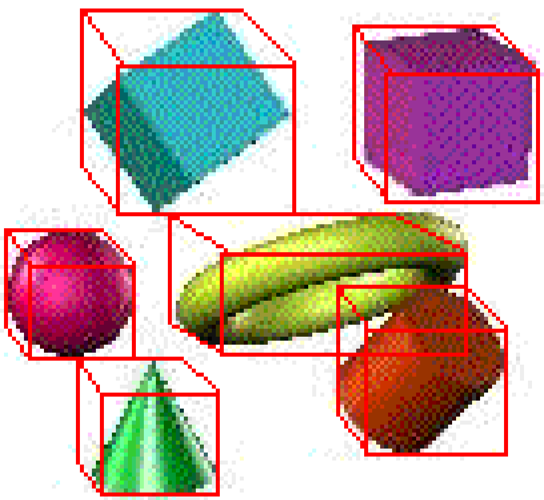
```
z1 = ...;
x2 = ...;
y2 = ...;
z2 = ...;
r1 = ###;
r2 = ###;

dx = x1 - x2;
dy = y1 - y2;
dz = z1 - z2;
value = sqrt(dx^2 + dy^2 + dz^2) - r1 - r2;
isTerminal = 1;
direction  = -1;
```

- See bounding circles video.

3 Bounding boxes

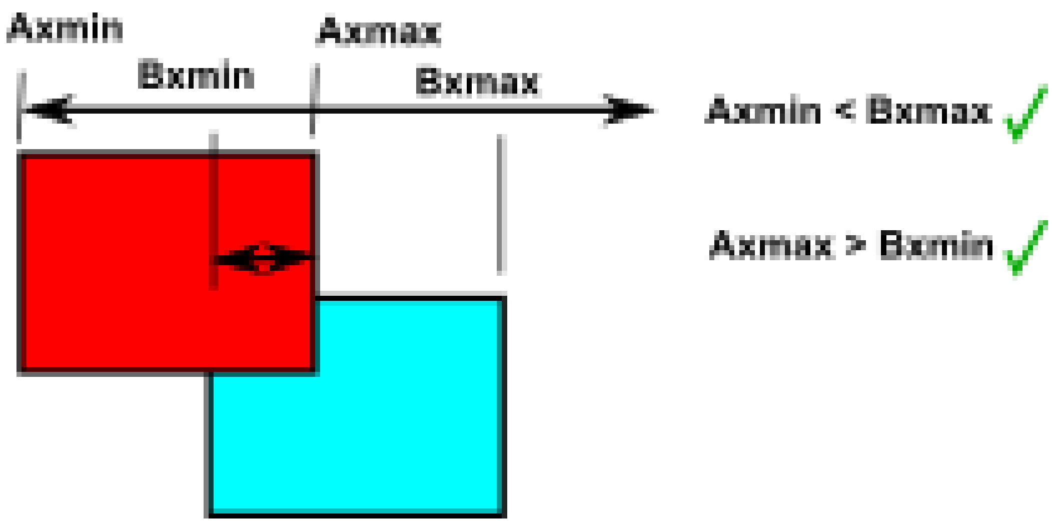
- Circles and spheres work nicely for hockey pucks, but don't resemble bowling pins very well.
- Bounding boxes are better.
- Bounding boxes are drawn for each body in a common inertial reference frame.



(Image from [1])

- Overlap can be tested with the following inequalities:

$$
\begin{array}{ccc}
A_{xMin} < B_{xMax} & \text{and} & A_{xMax} > B_{xMin} \\
A_{yMin} < B_{yMax} & \text{and} & A_{yMax} > B_{yMin} \\
A_{zMin} < B_{zMax} & \text{and} & A_{zMax} > B_{zMin}
\end{array}
$$



(Image from [1])

- See bounding box video.

4 Oriented bounding boxes

- Bounding boxes in a common inertial frame are ok when the pins are upright or laying down, but do not represent the bodies very well when they are tipped 45° in any direction.
- Bounding the body with a box in its principle axes better captures the body, but requires more computation to detect collision.
- I haven't tried this one yet.

5 Multiple bounding boxes

- One way to refine collision detection is to imagine that each body is composed of several smaller bodies.
- You could use several boxes per body and detect if any box of body 1 is touching any box of body 2.
- Haven't tried this either.

6 Meshes

- To get a near exact representation of any general body, meshes can be used. In other words, each body can be defined by n connecting triangles, quadrilaterals, or polygons.
- Collision can be detected by computing the minimum distance of any vertex on body 1 to any vertex on body 2. If this distance is less than a certain threshold, flag collision.
- Alternatively, you could use fancier geometry to determine if any of the polygons are overlapping.
- See mesh generation and collision detection videos.

7 Sphere-swept surfaces

- Recall how painless it was to detect collisions between spheres.
- The shape of a bowling pin conveniently lends itself to be constructed of many spheres.
- See Lee et al., 2007 [2].
- See sphere-swept pin generation video.

- The Matlab multivariable constrained minimization function `fmincon` is used to minimize

$$D(u, v) = ||\mathbf{C}_1(u) - \mathbf{C}_2(v)|| - r_1(u) - r_2(v)$$

such that

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$

where

$$\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix}$$

The constraints impose that u and v do not extend beyond the range of pin:

$$t_{min} \leq u \leq t_{max}$$
$$t_{min} \leq v \leq t_{max}$$

- Collision detection results will look the same as mesh results.

8 Synthesis

- Meshes are accurate if you use enough vertices, but then they take forever to compute.
- Sphere-swept surfaces provide an fast, accurate alternative for pins. Using constrained optimization rather than checking each point makes computation much faster. But still slow if you want to detect collisions between 10 pins and a ball.
- Using bounding spheres or boxes as a preliminary test cuts down compute time dramatically.
- See video of psuedo-bounding spheres and sphere-swept collision detection combined.

References

[1] Martin Baker. 3d theory - collision detection. *http://www.euclideanspace.com/threed/animation/collisiondetect/index.htm*, 2011. 2, 5, 6

[2] Kwanhee Lee, Joon-Kyung Seong, Ku-Jin Kim, and Sung Je Hong. Minimum distance between two sphere-swept surfaces. *Computer-Aided Design*, 39(2007):452–459, 2007. 10