# Supplement on collision detection and handling for TSBK07/11 2024

*First version 2024-02-22*

A few things were unclear or incomplete in the course book on collision detection and handling. This supplement is an attempt to make these things more clear, complete or correct.

## Collision detection

The following notes are some brief notes on the methods presented in the lectures and the book, to clarify a few things.

### On containment/intersection

Containment/intersection is a straight forward *exact* narrow phase algorithm and is included in the course as such. This algorithm was published by Moore&Williams in 1988. There are newer, faster algorithms, typically based on making more local operations, but the containment/intersection stands, for the purpose of the course, as a reference algorithm, showing how you need to take differet cases into account when testing for collisions.

### SAT is a broad phase algorithm

SAT is, as pointed out at the lecture, not practical for the narrow phase, and that is not what it should be considered. I should not have presented that as a narrow phase method, I take that back! Its main point is to handle OBB collisions. That brings the complexity down to something that fits the broad phase.

### The name of the method is "sphere trees"

The hierarcical approach using spheres is called "sphere trees" and was first published by Hubbard 1996 in a paper called "Time-critical collision". It is well known and many variations and developments have followed.

A notable detail is how it can be computed. By computing a distance transform of a shape, you can find the spheres in the form of local maxima. A subset of these will be smaller than the actual shape but a good approximation, while we get closer and closer to the actual shape by including smaller and smaller spheres.

While it is disturbing how algorithms like containment/intersection and even more advanced exact narrow phase algorithms are limited to convex shapes, sphere trees stand out as a method that combines free rotations and simple tests with a flexible choice of precision.

## Collision handling

The following sections are largely copied from the supplement "Beachball physics" for TSBK03. Here it complements a similar section of the course book that is too brief on these subjects, the older 2017 version, in particular. The latest version is better, but let me tell the whole story.

### Plastic collisions

If you have two objects that are both moving, with intial speeds $v_{1b}$ and $v_{2b}$, (where b is "before" and a is "after") and mass $m_1$ and $m_2$, respectively, you get

$$v_{1b}m_1 + v_{2b}m_2 = v_am_1 + v_am_2 = v_a(m_1 + m_2)$$

and we have a solution as

$$v_a = (v_{1b}m_1 + v_{2b}m_2)/(m_1 + m_2)$$

Thus, plastic collisions are simple to resolve. For elastic collisions, things are more hairy.

## Elastic collisions

For elastic collisions, we add conservation of kinetic energy, the kinetic energy is constant

$$v_1^2m_1 + v_2^2m_2 = \text{constant}$$

Let us take two billiard balls as example, idealized to giving totally elastic collisions. We only consider one-dimensional movement. As above we let them have the masses $m_1$ and $m_2$ and the velocities before collision $v_{1b}$ and $v_{2b}$. After collision, they will get the velocities $v_{1a}$ and $v_{2a}$.

$$v_{1b}m_1 + v_{2b}m_2 = v_{1a}m_1 + v_{2a}m_2$$

$$v_{1b}^2m_1 + v_{2b}^2m_2 = v_{1a}^2m_1 + v_{2a}^2m_2$$

These formulas has one trivial solution, no collision, so $v_{1b} = v_{1a}$ and $v_{2b} = v_{2a}$. The other solution is found by realizing that

$$v_{1b}^2m_1 + v_{2b}^2m_2 = v_{1a}^2m_1 + v_{2a}^2m_2$$

$$\Rightarrow$$

$$(v_{1b}^2 - v_{1a}^2)m_1 = (v_{2a}^2 - v_{2b}^2)m_2 = (v_{1b} - v_{1a})(v_{1b} + v_{1a})m_1 = (v_{2a} - v_{2b})(v_{2a} + v_{2b})m_2$$

This rewrite, to differences of pairs of squares, is the key to the whole problem! We now rephrase the first equation to

$$m_1(v_{1b} - v_{1a}) = (v_{2a} - v_{2b})m_2$$

and combining these two we get

$$v_{1b} + v_{1a} = v_{2a} + v_{2b} \Rightarrow v_{1a} = v_{2a} + v_{2b} - v_{1b}$$

or

$$v_{1b} + v_{1a} = v_{2a} + v_{2b} \Rightarrow v_{2a} = v_{1a} + v_{1b} - v_{2b}$$

which makes the whole thing almost look too simple! From there, it is straight-forward to eliminate $v_{1a}$ and get $v_{2a}$, and then get $v_{1a}$ in a similar way, and it will solve as:

using

$$v_{1b}m_1 + v_{2b}m_2 = v_{1a}m_1 + v_{2a}m_2$$

eliminate either from above and get

$$v_{1b}m_1 + v_{2b}m_2 = v_{1a}m_1 + (v_{1a} + v_{1b} - v_{2b})m_2$$

$$v_{1b}m_1 + v_{2b}m_2 = (v_{2a} + v_{2b} - v_{1b})m_1 + v_{2a}m_2$$

$$\Rightarrow$$

$$v_{1a} = (v_{1b}(m_1 - m_2) + 2v_{2b}m_2)/(m_1 + m_2)$$

$$v_{2a} = (2v_{1b}m_1 + v_{2b}(m_2 - m_1))/(m_1 + m_2)$$

This derivation was briefly hinted in Polygons Feel No Pain but too brief to understand. My fault, sorry.

To make the solution complete, we add a *restitution* parameter ε, where 1 is elastic and 0 is plastic. We use this to blend between the plastic and elastic collision formulas. To do this, we just multiply the formulas above with ε for the elastic part and (1-ε) for the plastic part. This will give us the final formula (adapted from Palmer [1], page 163):

$$v_{1a} = (v_{1b} (m_1 - \varepsilon m_2) + (1+\varepsilon)v_{2b}m_2)/(m_1 + m_2)$$

$$v_{2a} = ((1+\varepsilon)v_{1b}\, m_1 + v_{2b}(m_2 - \varepsilon m_1))/(m_1 + m_2)$$

Try to set ε to 0 or 1 and you will see that the expressions end up as the elastic and plastic cases above!

# References

[1] Grant Palmer, "Physics for Game Programmers", APress 2005

(More to add here when I have time:)

Moore&Williams 1988

Hubbard 1996

Watt