Dartmouth COSC 89.18/189.02 Computational Methods for Physical Systems, Winter 2019

Instructor: Bo Zhu (bo.zhu@dartmouth.edu)

Course Notes 2: Collision Detection and Response

1. Collision Detection for Implicit Geometries

An implicit geometry is defined as a signed distance function $\phi(\vec{x})$, taking a position as input and returning a signed scalar specifying the distance from $\vec{x}$ to the nearest point on the surface of the geometry. If the distance is negative, then $\vec{x}$ is inside the geometry; if the distance is zero, then $\vec{x}$ is on the boundary; if the distance is positive, then $\vec{x}$ is outside the geometry.

The gradient of the signed distance, $\nabla\phi$, specifies the normal direction of the surface.

The simplest implicit geometry is a sphere in 3D or a circle in 2D. For simplicity, we use the name "sphere" to describe a dimensionally independent geometry in the following context. The signed distance function of a sphere can be written as:

$$\phi(\vec{x}) = |\vec{x} - \vec{c}| - r,$$

with $\vec{c}$ as its center and $\vec{r}$ as its radius.

The collision between any two spheres $(\vec{c}_i, r_i)$ and $(\vec{c}_j, r_j)$ can be detected by simply checking if:

$$f(\vec{c}_i, \vec{c}_j, r_i, r_j) = |\vec{c}_i - \vec{c}_j| - (r_i + r_j) < 0$$

To detect the collisions between a sphere and a general implicit geometry (assuming the size of the sphere is much smaller than the geometry), we can check whether the signed distance of the sphere's center to the boundary of the geometry is smaller than its radius:

$$f(\vec{c}_i, r_i, \phi) = \phi(\vec{c}_i) - r_i < 0$$

To detect the collisions among two general implicit geometries, we need to sample a set of points on the boundary of one geometry and check the signed distance of each sampled point to the other geometry. To detect collisions among n geometries, O(n^2) queries need to performed. A grid or tree structure is usually needed to accelerate the process.

2. Penalty-based Collision Response

The penalty-based collision response force shares a very similar formula with the spring force. The general idea is to add a virtual spring in the region where collision happens to push the two objects apart.

Let's first consider the simplest case, a pair of spheres i and j, with mass $m_{i(or\ j)}$, position $\vec{c}_{i(or\ j)}$, and velocity $\vec{v}_{i(or\ j)}$. If the two spheres do not collide, which means $f(\vec{c}_i, \vec{c}_j, r_i, r_j) > 0$, the collision force is simply zero. If they collide, we add a virtual spring $s(k_s, k_d, l_0)$ with $l_0 = r_i + r_j$. Similar to the mass-spring model, the colliding force is modeled as the sum of the spring force $\vec{f}_s$ and the damping force $\vec{f}_d$, with:

$$\vec{f}_i^s = k_s(|\vec{x}_j - \vec{x}_i| - l_0)\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|}$$

$$\vec{f}_i^d = k_d\left((\vec{v}_j - \vec{v}_i)\cdot\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|}\right)\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|}$$

To calculate the collision force between a sphere $\vec{c}_i$ and a general object ϕ (assuming the object is much larger than the sphere), we need to first check if $\phi(\vec{c}_i) - r_i < 0$. If so, a virtual spring with the zero rest length along the direction of the surface normal $\nabla\phi$ is added to penalize the intersection between the sphere and the object. Similarly, the colliding force is modeled as the sum of the spring force $\vec{f}_s$ and the damping force $\vec{f}_d$, with:

$$\vec{f}_i^s = k_s(\phi(\vec{c}_i) - r_i)(-\nabla\phi)$$

$$\vec{f}_i^d = k_d\left((\vec{0} - \vec{v}_i)\cdot(-\nabla\phi)\right)(-\nabla\phi)$$

If you compare these equations with the spring and damping forces in a mass-spring model, you will find that we use the direction $-\nabla\phi$ pointing from the exterior to the interior of the object to replace the direction pointing from $\vec{x}_i$ to $\vec{x}_j$. Here we assume that the object does not move (as the spherical terrain you see in the starter code), so its velocity is $\vec{0}$.