Vishnu Pradheep Raveendran

Username: mrc12pvn

Assignments

Visual Interactive Simulation

Assignment 1: Particle system

Checking every pair of spheres and see if their distances are less than their radii is a general way to find the collisions. Problem arises when reach 300 spheres (for example), as there are about 50,000 pairs of spheres for potential collisions even though there are very few collisions. To solve this problem, broad phase collision method is used. In the assignment, octree broad phase collision method is used.

1.1 Octree:

Divide the cubical space (the space where all the spheres are placed) in half along each dimension, into eight smaller cubes. Then can figure out in which cube each ball is and check every pairs of ball in each smaller cube for collisions. Further divide the smaller cubes themselves into eight cubes and take every pair of spheres in each even smaller cube, so that we have even fewer pairs of spheres to check.

The problem in further dividing into smaller cubes: The more we divide up the cubes, the more frequently spheres will appear in multiple cube, which will produce duplicate pairs and false positives. To solve this problem, the octree is used. If there are a lot of spheres in it, make eight smaller cubes and let them take care of finding potential collisions. If there are not so many spheres, just use every pair of spheres as the set of potential collisions. This result in tree structures – each cube is a node in the tree and if it is divided into smaller cubes, these cubes are its eight children. It is called an octree.

1.3 Newton coulomb:

Once the ball hit on the ground, one of the impulse points upward to prevent the block to go into ground. That is, after the collision, new velocity $\vec{u}'_{relative}$ will be pointing upwards as all the tangential velocity will be cancelled by the friction.

$$u'_{relative,normal} = -\varepsilon * u_{relative,normal} \tag{1.1}$$

where $u_{relative,normal}$ is the original normal velocity, ε is the coefficient of restitution, $u'_{relative,normal}$ is the new normal velocity

$$\vec{u}'_{relative} = u'_{relative,normal} * \vec{N} \tag{1.2}$$

$\vec{N}$ stands for Normal direction.

Substituting equation (1.1) into equation (1.2) gives

$$\vec{u}'_{relative} = -\varepsilon * \vec{N}^T u'_{relative,normal} * \vec{N} \tag{1.3}$$

Bonus task (implemented):

1. Implemented intersection tests with sphere to sphere collision with Newton coulomb
2. Implemented broad phase collision detection

Assignment 2: Simulation of Cloth

The 3D particles/ point location is calculated by,

```
v=21, u=21; // where u*v gives total number of particles/ points
J=0,i=0;
  Loop j till v:
      Loop i till v:
          $\left(\left(\frac{i}{u-1}\right) * 2,\ 5,\ \left(\frac{j}{v-1}\right) * 4\right)$;
      End i ;
  End J;
```

The mass given to each particle is 0.5, damping value given is -0.0125 and gravity vector is (0.0f, -0.00981f, 0.0f) . Then the particles are connected by springs and then apply spring forces through Hooke's law,

2.1 Hooke's law:

'P0' and 'q0' are the two particles on each side (both are vectors), l_0 is the rest length of the spring (scalar)

$$l_0 = ||p0 - q0|| \tag{2.1}$$

and l is the current length of the spring

$$l = ||p - q|| \tag{2.2}$$

Hooke's law tells the energy of a string/ spring. Energy function E(p,q) is,

$$E(p, q) = \frac{1}{2}k(||p - q|| - l_0) \tag{2.3}$$

which measures the energy stored in the spring due to deformation. Where k in equation (2.3) is stiffness parameter and in assignment the stiffness parameter is 0.5. Energy of the spring is calculated by taking partial derivative with respect to 'p' ($\nabla_p E$) and 'q' ($\nabla_q E$), gives

$$\nabla_p E = (1 - \frac{l_0}{l})(p - q)^T \tag{2.4}$$

$$\nabla_q E = -\nabla_p E \tag{2.5}$$

For stretching of spring,

$$\nabla E = [\nabla_p E, -\nabla_p E] \tag{2.6}$$

and the spring force is,

$$F_{spring} = k(l_0 - \|p - q\|)\frac{p - q}{\|p - q\|} \tag{2.7}$$

For every step, derivative of Energy (∇E) is computed assuming that the derivative is constant (which is not) in the entire interval (example, from t1 to t2), leads to some error (from t1 to t2). The error can be reduced by decreasing the time step (near to zero) and this is done by numerical integration. In assignment, the Explicit Euler integrator is used and the time step value is 0.0166.

In the assignment, I made the particles which are connected by the spring (by Hooke's law) to collide with a cuboid. This collision detection between all the particles of the cloth and cuboid is done by bounding volume hierarchy, where the cuboid is bounded by a sphere. The collision is detected when the difference between a particle vector and the centre of bounding sphere is less than a certain value.

The spring constant value used here is 0.5 and this makes the spring connected to the particles with mass 0.5 to behave in a stabilize manner. When the spring constant value is increased 30 times than the current value of 0.5, then still the springs connected to the particles with mass 0.5 behaves in a stabilize manner. But when the spring constant value is increased 50 times than the current value of 0.5, then the springs connected to the particles with mass 0.5, just explodes.

Bonus task (implemented):

1. Implemented additional spring connection : Shear springs
2. Implemented additional collision : cloth collides with cuboid

Assignment 3: Constraints - Rope

Consider two particles x1 and x2, the distance constraint between the two particles is

$$g(x) = \|x1 - x2\| - l = 0 \tag{3.1}$$

where l is a string length which is constant. Distance constraint is the one that keeps particles xi and $x(i+1)$ at given distance l_i

In the assignment, I consider four particles, which leads to applying distance constraint between four 3D particles.

3.1 Jacobian for distance constraints between particles(3D):

Jacobian for distance constraint between particles is Partial derivatives (Jacobian) of g (x) with respect to all coordinates. For two particle case,

$$\frac{\partial g_i}{\partial x^i} = \frac{1}{\|r_{ii+1}\|} r_{i,i+1} \tag{3.2}$$

where $r_{i,j} = r_{ij} = x^{(i)} - x^{(i)}$

$$\frac{\partial g_i}{\partial x^{i+1}} = -\frac{1}{\|r_{ii+1}\|} r_{i,i+1} \tag{3.3}$$

Unit vector is, $\boldsymbol{u_{ij}} = \frac{\mathbf{1}}{\|\boldsymbol{r_{ij}}\|} \boldsymbol{r_{ij}}$

Jacobian (G) is putting all these constraints together in a matrix form (for 4 particles), gives

$$G = \begin{bmatrix} u_{12} & -u_{12} & 0 & 0 \\ 0 & u_{23} & -u_{23} & 0 \\ 0 & 0 & u_{34} & -u_{34} \end{bmatrix}$$

By Newton's second law,

$$Mv = Mv^{(0)} + hf^{(0)} + G\lambda \tag{3.4}$$

where $v^{(0)}$ is initial velocity, 'h' is the time step (h>0), $f^{(0)}$ is a force acting on the system evaluated at the beginning of the time step and G is the Jacobian. The scalar λ represents the magnitude of constraint force that should be applied to bring the system to valid state. Further, the system moves away from the valid states, bigger the λ, which will push it back to a valid state. The equation (3.4) leads to shaky system. So, in assignment using SPOOK will stabilize the system.

3.2 SPOOK:

$$Mv = Mv^{(0)} + hf^{(0)} + G\lambda$$

Same (3.4) equation

$$\frac{\varepsilon}{h}\lambda + \frac{\mathbf{1}}{\mathbf{4}}(\boldsymbol{g}_{\boldsymbol{k+1}} + \mathbf{2}\boldsymbol{g}_{\boldsymbol{k}} + \boldsymbol{g}_{\boldsymbol{k-1}}) + \tau G v = 0 \tag{3.5}$$

$$\frac{\mathbf{1}}{\mathbf{4}}(\boldsymbol{g}_{\boldsymbol{k+1}} + \mathbf{2}\boldsymbol{g}_{\boldsymbol{k}} + \boldsymbol{g}_{\boldsymbol{k-1}}) = g_k + \frac{h}{4}Gv - \frac{h}{4}Gv^{(0)} \tag{3.6}$$

where $\frac{h}{4}Gv$ is unknown.

Substituting equation (3.6) into equation (3.5) gives,

$$\frac{\varepsilon}{h}\lambda + g_k + \frac{h}{4}Gv - \frac{h}{4}Gv^{(0)} + \tau G v = 0 \tag{3.7}$$

Putting all the unknown on one side,

$$\frac{\varepsilon}{h^2(1+4\frac{\tau}{h})}\lambda + Gv = \frac{-4}{h(1+4\frac{\tau}{h})}g_k + \frac{1}{1+4\frac{\tau}{h}}Gv^{(0)} \tag{3.8}$$

Where g_k is the vector constraint violation at the beginning of the time step, h is the time step, τ is relaxation time - number of time steps to stabilize the constraints and $\varepsilon = \frac{1}{k}$ (where 'k' is spring constant) which allows to do spring damping with arbitrary stiffness.

Equation 3.4 and 3.8 in system matrix is given as,

$$\begin{bmatrix} M & G^T \\ G & \Sigma \end{bmatrix} \begin{bmatrix} v \\ \lambda \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} \tag{3.9}$$

Where p= $Mv^{(0)} + hf^{(0)}$, q=$-\frac{4\gamma}{h} g_k + \gamma G v^{(0)}$, $\gamma = \frac{1}{1+\frac{4\tau}{h}}$ and ε_i=$\frac{4\varepsilon\gamma}{h^2}$. Here in the assignment Σ=$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$, M= square nxn real, symmetric, positive definite matrix – giving 0.5 as the mass value for all the four particles in the assignment, h=0.02, τ=1.3, γ=0.8

The system matrix (3.9) is not symmetric, though it is positive definite and so it cannot be processed using Gauss-Seidel iteration. So, here Schur complement is used to reduce into smaller system of linear equation,

$$D\lambda = q - GM^{-1}p \tag{3.10}$$

Where,

$$D = GM^{-1}G^T + \Sigma \tag{3.11}$$

which is the Schur complement matrix. Once λ is computed from equation (3.10), the constraint velocity v is calculated as,

$$v = M^{-1}(G^T\lambda + p) \tag{3.12}$$

To solve linear equation (3.10), the Gauss-Seidel iteration is used.

3.3 Gauss-Seidel iteration:

Gauss-Seidel iteration is a part of iterative methods, which solve linear system equations. Algorithm to solve equation (3.10) by Gauss-Seidel iterations is given below (taken from the lecture notes[5])

Gauss-Seidel iterations to solve $(GM^{-1}G^T + \Sigma)\lambda = q - Gv^{(1)}$, without residual update.

```
Given $q, M, G, \Sigma, hf^{(0)}$.
initialize $v = v^{(0)} + hM^{-1}f^{(0)}, \quad \lambda = \lambda^{(0)}$.
Compute initial residual and error, $r = Gv + \Sigma\lambda - q, \quad \xi = \|r\|^2$        ▷ Not strictly needed.
Compute blocks $D_{kk} = \sum_b G_{kb} M_{bb}^{-1} G_{kb}^T + \Sigma_{kk}$, for $k = 1, 2, \dots, n_c$
repeat
    for $k = 1, 2, \dots, n_c$ do
        $r = -q_k + \Sigma_{kk}\lambda_k$        ▷ Compute local residual from scratch
        for $b = b_{k_1}, b_{k_2}, \dots, b_{n_{b_k}}$ do
            $r = r + G_{kb} v_b$
        end for
        $\xi = \xi - \|r\|^2$        ▷ Not needed if error is not monitored
        $z = -D_{kk}^{-1} r_k$
        $\lambda_k = \lambda_k + z$
        for $b = b_{k_1}, b_{k_2}, \dots, b_{n_{b_k}}$ do        ▷ Loop over bodies connected via constraint $k$
            $v_b = v_b + M_{bb}^{-1} G_{kb}^T z$        ▷ Update velocities
        end for
        $r = -q_k + \Sigma_{kk}\lambda_k$        ▷ Recompute local residual from scratch
        for $b = b_{k_1}, b_{k_2}, \dots, b_{n_{b_k}}$ do
            $r = r + G_{kb} v_b$
        end for
        $\xi = \xi + \|r\|^2$        ▷ Not needed if error is not monitored
    end for
until Error is small, or iteration time is exceeded
```

With equal mass value of 0.5 for all the particles, it took 10 iterations to stabilize. When the mass of a bottom particle (4th particle) changed to 30 times higher than the other particles: with the iteration of 10, the 2nd particle becomes little shaky (little unstable), the 3rd and the 4th particle becomes more shaky (more unstable). When the iteration increased to 65, all the particles come back to the stable state. The direction of iterations does not affect the stability of the rope consisting of particles.

Assignment 4: Rigid Body Simulation

After implementing broad phase collision detection (octree) which was explained in section 1.1, contact constraint and its jacobian is calculated (explained below),

Consider two particles x1 and x2, the contact constraint between two particle is

$$g = n(x1 + x2) \tag{4.2}$$

where,

$$-n(normal) = \frac{x1 - x2}{\|x1 - x2\|} g_k \tag{4.2}$$

In case of assignment, for rigid bodies (spheres), the Jacobian for contact constraint between two particles is calculated as,

$$r = x1 + p1 - x2 - p2 \tag{4.3}$$

where x1 and x2 are centre of the rigid body 1 and centre of rigid body 2 respectively, p1 and p2 are the radius of the rigid body 1 and rigid body 2.

$$\frac{\partial g_i}{\partial x^{(i)}} = \frac{1}{\|r_{ii}\|} (r_{i,i+1})^T \dot{r}_{ii+1} \tag{4.4}$$

$$\frac{\partial g_i}{\partial x^{(i+1)}} = -\frac{1}{\|r_{ii+1}\|} (r_{i,i+1})^T \dot{r}_{ii+1} \tag{4.5}$$

Unit vector is, $\boldsymbol{u_{ij}} = \frac{\mathbf{1}}{\|r_{ij}\|} (r_{ij})^T \dot{r}_{ij}$

Jacobian (G) is contact constraints together in a matrix form (for 2 rigid bodies), gives

$$G = \begin{bmatrix} u_{12} & -u_{12} & 0 & 0 \end{bmatrix}$$

After calculating Jacobian, SPOOK (same parameter values as in section 3.1 are used for SPOOK except Jacobian, for this assignment) and Gauss-Seidel iteration are implemented for stabilization and to solve linear equation respectively (explanation given in section 3.1 and 3.2).

With equal mass value of 0.5 for all the spheres (with random radius) it took 100 iterations to stabilize. When the mass of spheres changed to 30 times higher than the other particles: with the iteration of 100, the spheres just explode when it is sacked up on each other. When the iteration increased to 260, all the particles come back to the stable state.

Bonus task (implemented):

1. Implemented broad phase collision detection for spheres of arbitrary size (spheres with random radius)

References

[1] *Computing constraint Jacobians, Author: Claude Lacoursiere, HPC2N/UMIT and Department of Computing Science, Umea University*

[2] *'Notes on the SPOOK method for simulating constrained mechanical systems' by Kenneth Bodin,Department of Computing Science, Umeå University*

[3] *'Notes on Discrete Mechanics' by M. Servin, Department of Physics, Umeå University*

[4] *'Simple Numerical Methods for Mechanical Systems Subject to Strong forces and Constraints' by Claude Lacoursiere, Department of Computing Science, Umea University*

[5] *'Using Gauss-Seidel for Multibody Problems' by Claude Lacoursiere, HPC2N/UMIT and Department of Computing Science, Umea University*

[6] *All Lecture notes from Visual Interactive Simulation course*

Appendix:

Videos of all the assignments are attached in E-mail, along with this report.