Chapter 15

Smoothed Particle Hydrodynamics

15.1 Simple Particle Systems

As mentioned above, for splashes, spray or water jets, particles are probably the best choice. In those simple cases it is often not even necessary to simulate the interaction of particles with themselves. We call a particle system without particle-particle interaction a *simple* particle system. Such a system can be implemented very efficiently which means that a large number of particles can be simulated in real-time. All we need is a set of N particles $0 \leq i < N$ with masses m_i, positions $\mathbf{x}_i$, velocities $\mathbf{v}_i$ and accumulated external forces $\mathbf{f}_i$. These particles are either created and initialized with meaningful positions and velocites before the simulation starts or they are generated during the simulation by *emitters*. An emitter typically generates the particles with a certain rate [particle/s] and a certain velocity distribution around a principal main direction. To make sure that particles are not only generated but also die after a certain while, they are given lifetimes. When their lifetime gets close to their maximum lifetime, they are faded out smoothly. The lifetimes can also be used to color the particles appropriately. Since there are no particle-particle interactions, only per-particle forces exists and the governing equation is a set of decoupled ordinary differential equations

$$\dot{\mathbf{x}}_i = \mathbf{v}_i \tag{15.1}$$

$$\dot{\mathbf{v}}_i = \mathbf{f}_i / m_i \tag{15.2}$$

which can be efficiently integrated using an explicit Euler integration scheme for instance.

15.2 Particle-Particle Interactions

If we want to simulate small bodies of water with particles, it is quite essential that the particles feel each other. Otherwise, they all aggregate at a single spot in a corner or along an edge. In general, an interaction force for particles has the form

$$\mathbf{f}(\mathbf{x}_i, \mathbf{x}_j) = F(|\mathbf{x}_i - \mathbf{x}_j|) \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}, \tag{15.3}$$

where $F()$ is the magnitude of the force depending only on the distance between the particles. The force acts along the connection between the particles, otherwise it would introduce a torque. A popular choice is the *Lennard-Jones*

force, often used in molecular dynamics simulations

$$\mathbf{f}(\mathbf{x}_i, \mathbf{x}_j) = \left(\frac{k_1}{|\mathbf{x}_i - \mathbf{x}_j|^m} - \frac{k_2}{|\mathbf{x}_i - \mathbf{x}_j|^n} \right) \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}, \tag{15.4}$$

with k_1, k_2, m and n control parameters. Popular choices are $k_1 = k_2 = k$, $m = 4$ and $n = 2$.

With N particles, there are $O(N^2)$ possible interactions which we have to evaluate. This quadratic complexity can be the show stopper. Image we were able to simulate ten thousand particles without interactions in real time. This means we would now have to evaluate hundred million interactions also in real-time. To avoid quadratic complexity one introduces a cutoff distance d beyond which the particles do not feel each other anymore. In order to keep the simulation stable it is important that the function $F()$ is not just cut off to zero at distance d but that it is C^0 and C^1 continuous at d as well. Now at each time step, the particles are filled into a regular grid with cell spacing d. After this, potential interaction partners for a given particle only need to be searched for in the same or in adjacent cells. If the particles are evenly distributed and the number of particles per cell is bound by a constant, the interactions can be evaluated in linear time.

15.3 SPH

The next question we ask is, can we do better than using Lennard-Jones interaction forces? Is it possible to use the Navier-Stokes equations and solve them on the particles? The answer is yes [MCG03]. The first problem we have to solve is to generate smooth, continuous fields from quantities that are only given on the particles, i.e. at discrete locations in space. The Smoothed Particle Hydrodynamics method, originally devised for the simulation of stars [Mon92], solves this problem. At its core, it defines a way to smooth discretely sampled attribute fields. This is done via so-called *smoothing kernels* $W(r)$. The kernel defines a scalar weighting function in the vicinity of the position $\mathbf{x}_i$ of particle i via $W(|\mathbf{x}-\mathbf{x}_i|)$. Defined this way, the kernel is symmetric around the particle because it only depends on the distance to the particle. The kernel function also needs to be normalized meaning that $\int W(|\mathbf{x} - \mathbf{x}_i|)d\mathbf{x} = 1$. A popular choice is the *poly6* kernel

$$W_{\text{poly6}}(r) = \frac{315}{64\pi d^9} \begin{cases} (d^2 - r^2)^3 & 0 \leq r \leq d \\ 0 & \text{otherwise,} \end{cases} \tag{15.5}$$

because r only appears squared and no square root has to be evaluated. We now have all the ingredients to compute a smooth density field from the individual positions and masses of the particles:

$$\rho(\mathbf{x}) = \sum_j m_j W(|\mathbf{x} - \mathbf{x}_j|). \tag{15.6}$$

The density of particle i is then simply $\rho_i = \rho(\mathbf{x}_i)$. Now it becomes apparent why the kernels need to be normalized. With this restriction, the totaly mass of the system computed as the integral of the density field yields

$$\int \rho(\mathbf{x})d\mathbf{x} = \sum_j \left(m_j \int W(|\mathbf{x} - \mathbf{x}_j|)d\mathbf{x} \right) = \sum_j m_j \tag{15.7}$$

Having the density values of the individual particles, we can now compute smoothed fields A_s of arbitrary attributes A_i of the particles as

$$A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(|\mathbf{x} - \mathbf{x}_j|). \tag{15.8}$$

A nice property of this formulation is that the gradient of such a field can easily be computed by replacing the kernel by the gradient of the kernel

$$\nabla A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|\mathbf{x} - \mathbf{x}_j|). \tag{15.9}$$

In the Eulerian (grid based) formulation, fluids are described by a velocity field $\mathbf{v}$, a density field ρ and a pressure field p. The evolution of these quantities over time is given by two equations. The first equation assures conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{15.10}$$

while the Navier-Stokes equation formulates conservation of momentum

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}, \tag{15.11}$$

where $\mathbf{g}$ is an external body force and μ the viscosity of the fluid. The use of particles instead of a stationary grid simplifies these two equations substantially. First, because the number of particles is constant and each particle has a constant mass, mass conservation is guaranteed and the first equation can be omitted completely. Second, the expression $\partial \mathbf{v}/\partial t + \mathbf{v} \cdot \nabla \mathbf{v}$ on the left hand side of the Navier-Stokes equation can be replaced by the substantial derivative $D\mathbf{v}/Dt$. Since the particles move with the fluid, the substantial derivative of the velocity field is simply the time derivative of the velocity of the particles meaning that the convective term $\mathbf{v} \cdot \nabla \mathbf{v}$ is not needed for particle systems.

There are three body forces (unit $[N/m^3]$) left on the right hand side of the Navier-Stokes equation modeling pressure ($-\nabla p$), external forces ($\rho \mathbf{g}$) and viscosity ($\mu \nabla^2 \mathbf{v}$). The sum of these body forces $\mathbf{f} = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}$ determines the change of momentum $\rho \frac{\partial \mathbf{v}}{\partial t}$ of the particles on the left hand side. For the acceleration of particle i we, thus, get:

$$\mathbf{a}_i = \frac{\partial \mathbf{v}_i}{\partial t} = \frac{\mathbf{f}_i}{\rho_i}, \tag{15.12}$$

where $\mathbf{v}_i$ is the velocity of particle i and $\mathbf{f}_i$ and ρ_i are the body force and the density field evaluated at the location of particle i, repectively. We will now see how the body forces can be evaluated using SPH.

15.3.1 Pressure

Application of the SPH rule described in Eqn. 15.9 to the pressure term $-\nabla p$ yields

$$\mathbf{f}_i^{\text{pressure}} = -\nabla p(\mathbf{x}_i) = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|). \tag{15.13}$$

Unfortunately, this force is not symmetric as can be seen when only two particles interact. Since the gradient of the kernel is zero at its center, particle i only uses the pressure of particle j to compute its pressure force and vice versa. Because the pressures at the locations of the two particles are not equal in general, the pressure forces will not be symmetric. Different ways of symmetrization of Eqn. 15.13 have been proposed in the literature. Here is a very simple solution which is stable and fast to compute:

$$\mathbf{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|). \tag{15.14}$$

Since particles only carry the three quantities mass, position and velocity, the pressure at particle locations has to be evaluated first. This is done in two steps. Eqn.15.6 yields the density at the location of the particle. Then, the pressure can be computed via the ideal gas state equation

$$p = k(\rho - \rho_0), \tag{15.15}$$

where k is a gas constant that depends on the temperature and ρ_0 is the environmental pressure. Since pressure forces depend on the gradient of the pressure field, the offset mathematically has not effect on pressure forces. However, the offset does influence the gradient of a field smoothed by SPH and makes the simulation numerically more stable.

However, incompressibility is not enforced strictly as in the Eulerian case. Pressure forces are only generated after the fact, when density variations have already formed. This is clearly a penalty method and yields a bouncy behavior of the fluid. To avoid this effect, one can predict the densities and compute pressure forces on predicted densities or estimate the divergence of the velocity field using SPH and then solving the Poisson equation on the particles [PTB+03].

15.3.2 Viscosity

Application of the SPH rule to the viscosity term $\mu\nabla^2\mathbf{v}$ again yields asymmetric forces

$$\mathbf{f}_i^{\text{viscosity}} = \mu\nabla^2\mathbf{v}(\mathbf{x}_i) = \mu\sum_j m_j\frac{\mathbf{v}_j}{\rho_j}\nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|). \tag{15.16}$$

because the velocity field varies from particle to particle. Since viscosity forces are only dependent on velocity differences and not on absolute velocities, there is a natural way to symmetrize the viscosity forces by using velocity differences:

$$\mathbf{f}_i^{\text{viscosity}} = \mu\sum_j m_j\frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j}\nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|). \tag{15.17}$$

A possible interpretation of Eqn. 15.17 is to look at the neighbors of particle i from i's own moving frame of reference. Then particle i is accelerated in the direction of the relative speed of its environment.

15.3.3 External Forces

External forces such as gravity, collision forces and forces caused by user interaction are applied directly to the particles as in the case of simple particle systems without the use of SPH.

15.4 Rendering

There are many ways to render fluids which are defined via particles. One of the simplest and fastest methods is to use sprites. If a smoothing filter is applied to the depth buffer before the sprites are drawn, the granularity of the particles can be hidden to a certain degree. Another way of rendering particle fluids is to draw an iso surface of the density field and triangulate it using the marching cubes algorithm.