Implementation of Smoothed Particle Hydrodynamics

Popov Nicolai

1 Motivation

This work is dedicated to simulation and visualization of fluids. The importance of such technologies is pretty obvious: wind, weather, ocean waves and simply pouring of a glass of water are examples of fluid phenimena. There is still active research ongoing in the field of Computational Fluid Dynamics.

Fluid phenomena are simulated both for off-line and real-time applications. In first case the method used is made as accurate as possible, for example in aerodynamics or optimization of turbines and pipes. Less accurate real-time methods help to test whether a certain concept is promising during the design phase and also are used in medical simulators, video games and virtual environments.

The SPH concept is one the methods to simulate fluids. Its application is popular in areas from entertainment technologies to engineering. For example, the settings of free-surface fluid with complex moving boundaries are relevant for special effects productions and for the analysis of vehicles moving in water.

2 Method

SPH is an interpolation method for particle systems. With SPH, field quantities that are only defined at discrete particle locations can be evaluated anywhere in space. For this purpose, SPH distributes quantities in a local neighborhood of each particle using radial symmetrical smoothing kernels.

For arbitrary continuous function A its convolution with delta-function is equal to A, so if we use a normalized kernel function W instead of delta-function we will get an approximation to that equation, which than can be dicretized in the following way:

$$A(r) = \sum_j A_j \cdot W(r - r_j, h) \cdot V_j; \quad V_j = \frac{m_j}{\rho_j} \tag{1}$$

where j iterates over all particles, m_j is the mass of particle, r_j its position, ρ_j the density and A_j the field quantity at r_j. The equation above is used in SPH to interpolate a field A at location r by a weighted sum of contributions from all particles. The function $W(r, h)$ is called the smoothing kernel with core radius h. In the implementation of this method only kernels with finite support were used and h is the radius of support.

In most fluid equations, derivatives of field quantities appear and need to be evaluated. With the SPH approach, such derivatives only affect the smoothing kernel. The gradient of A is simply:

$$\triangledown A(r) = \sum_j A_j \cdot \triangledown W(r - r_j, h) \cdot \frac{m_j}{\rho_j} \tag{2}$$

$$\triangle A(r) = \sum_j A_j \cdot \triangle W(r - r_j, h) \cdot \frac{m_j}{\rho_j} \tag{3}$$

It is important to realize that SPH holds some inherent problems. When using SPH to derive fluid equations for particles, these equations are not guaranteed to satisfy certain physical principals such as symmetry of forces and conservation of momentum.

Navier Stokes equation describes the forces that act on a particle of the simulated incompressible fluid:

$$F_{total} = ma = m\frac{dv}{dt} = mg - \frac{m}{\rho} \bigtriangledown P + m\nu \bigtriangleup v$$

where m, v and a are the mass, velocity and acceleration of the particle, g is the gravitational acceleration and P is the pressure.

The time for simulation is discretized with step dt. The simulation includes two steps that should be done at each time moment:

I) Calculation of total force F_{total} acting on each particle
II) Numerical integration for velocity and position:

$$v^{(t+1)} = v^{(t)} + dt * a$$

$$p^{(t+1)} = p^{(t)} + dt * v$$

To calculate three forces:

$$F_{weight} = mg,$$
$$F_{pressure} = -\frac{m}{\rho} \bigtriangledown P,$$
$$F_{viscosity} = m\nu \bigtriangleup v$$

for each particle at each time moment it is necessary to calculate values of ρ_i, $\bigtriangledown P_i$, $\bigtriangleup v_i$ for particle i.

For each of this forces a special kernel function W can be used depending on the nature of the simulated force. The following ones were used in my implementation:

$$W_{poly6}(r - r_j, h) = \frac{315}{64\pi h^9}(h^2 - \|r - r_j\|^2)^3$$

$$W_{spiky}(r - r_j, h) = \frac{15}{\pi h^6}(h - \|r - r_j\|)^3$$

So W_{poly6} is used in smoothing formulas (1), (2) and (3) for all computed fields except pressure, for which W_{spiky} is used.

Fields are calculated in the following order:
1) smoothed density is updated for each particle
2) pressure is calculated from density: $P = s(\rho - \rho_0)$ and updated for each particle
3) smoothed forces of pressure and viscosity are calculated and summed.

The final formulas for this with some tricks can be found in the lecture from Computer Animation course [4].

It is worth mentioning that Navier Stokes equation does not take into account the fluid phenomena induced by interacting of moleculed at the border of two different substances, e. g., air and water. Molecules in a fluid are subject to attractive forces from neighboring molecules. Inside the fluid these intermolecular forces are equal in all directions and balance each other. In contrast, the forces acting on molecules at the free surface are unbalanced. The net forces (i.e. surface tension forces) act in the direction of the surface normal towards the fluid. They also tend to minimize the curvature of the surface. The larger the curvature, the higher the force. Surface tension also depends on a tension coefficient σ which depends on the two fluids that form the surface.

Thus, to make the simulation more realistic in 3D, the surface tension force was added. The approach from article [1] was used for this purpose. According to this paper a special «color field» is introduced and is 1 at particle locations and 0 everywhere else. The smoothed color field:

$$C(r) = \sum_j \frac{m_j}{\rho_j} \cdot W(r - r_j, h)$$

The gradient field of the smoothed color field is equal to the surface normal field pointing into the fluid and the divergence of n (i.e. laplacian of color field) measures the curvature of the surface

$$\triangledown C = n, \quad \kappa = -\frac{\triangle C}{\|n\|}$$

The minus is necessary to get positive curvature for convex fluid volumes.

To distribute the surface traction among particles near the surface authors of [1] use a normalized scalar field

$$\delta S = \begin{cases} \|n\|, & \text{particle near the surface} \\ 0, & \text{other particles} \end{cases}$$

For the force density acting near the surface we get Finally, for the surface tension:

$$F_{surface} = \sigma\kappa \frac{n}{\|n\|} \cdot \delta S \cdot \frac{m}{\rho} = \begin{cases} \sigma\kappa\frac{m}{\rho}n, & \text{particle near the surface} \\ 0, & \text{other particles} \end{cases}$$

But it is not described how to classify particles into ones that are near the surface and the rest ones. Finding a threshold for normal lenght to distinguish between surface particles and other ones is not feasible because the normal lenght is not zero for particles that are remote from the fluid surface. So in my implementation this force was calculated for all particles with normal lenght big enough to divide on it (it is necessary to evaluate curvature). So to calculate $F_{surface}$ it is necessary to calculate smoothed gradient and laplacian fields of color field for each particle. This decision has two downsides: the first one being the extra coputation overload and the second one is the unnecessary particle movement in the volume of fluid which are small but add unwanted perturbations though.

Another method called uniform grid acceleration was used to speed up all the calculations of smoothed fields. It uses two data structures for mapping between particle id and three numbers representing its coordinates in 3D grid Both of them are updated before doing calculations for next time moment. Then grid coordinates are used to loop over particles only in the 27 grid cells - a 3x3 cube such that its central cell includes the particle for which the forces are calculated but not all other particles of fluid. The obtained acceleration result is obvious for a relatively big number of particles of fluid.

3 Implementation details

The code of the SPH lab assignment from the Computer Animation course [5] was used as the base for this project which initially was running simulation in 2D. Although all the computations of fields were done in 3D with zero z-component, I added a cube box to the scene and enable its rotations to switch to full 3D. To do so I used the functionality of CGP library [**?**].

To increase interaction of the user with the simulation I added the possibility to change the gravity direction. If it is not fixed by the user then its value is updated to the opposite of camera UP vector for each frame.

Another possibility is to turn off the gravity at all. This function was used to test if the fluid forms a sphere in the weightlessness in order to tune the surface tension coefficient σ.

Visualization in 3D is simple: the particles of fluid are depicted as blue spheres. This solution is far from ideal one, because the surface of water (i. e. border between water and air as well as water and box wall) is not triangulated and thus glares can not be depicted.

The following parameters were added to sph_parameters_structure in order to utilize them in the above described calculations: sigma, norm_threshold, scene_up_vector, particle_id2grid_cell and
grid_cell2particle_ids.

I used W_{poly6} kernel for the computation of surface tension and calculated its gradient (W_gradient_color_field)and laplacian (W_laplacian_color_field).

The update functions are replaced to the new ones for density and force but pressure. And the relection is added for all walls of the cubic box where liquid particles are placed.

4 Future Work

Point splatting approach mentioned in [2] can be used to triangulate the surface of liquid to make it look more realistic.

There is a pretty simple solution to classify particles into ones near the surface and the rest ones that requires creating a dataset to use it for training a linear classificator as described in work classify.

References

[1] Müller, M., Charypar, D. and Gross, M., 2003, July. Particle-based fluid simulation for interactive applications. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (pp. 154-159).

[2] Koschier, D., Bender, J., Solenthaler, B. and Teschner, M., 2020. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. arXiv preprint arXiv:2009.06944.

[3] Zorilla, F., Ritter, M., Sappl, J., Rauch, W. and Harders, M., 2020. Accelerating surface tension calculation in SPH via particle classification and Monte Carlo integration. Computers, 9(2), p.23.

[4] Lecture with notes on SPH fluid simulation

[5] Computer Animation course

[6] CGP library repo