Fluid-cloth coupling algorithm based on PBD and CCD

Sijie Huang*, Kun Zhou
Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China

*Corresponding author: sijiehuang@nbu.edu.cn

Abstract. In order to simulate the interaction between cloth and fluid, we use Smooth Particle Hydrodynamics (SPH) as a method in fluid simulation, we propose a Continuous collision detection (CCD) and position-based dynamics (PBD) collision constrained iterative flow coupling method. By analyzing the parallelism of the algorithm, we implement a low storage and load balanced fluid and cloth coupling algorithm parallel computing is used to accelerate the simulation. The experimental results show that our method realizes the bidirectional coupling of fluid and cloth, and improves the simulation performance.

Keywords: PBD, CCD, Fluid, cloth.

1. Introduction

In the simulation of cloth and fluid based on physics, cloth has the characteristics of deformation, and fluid particles are highly free motion. It is challenging to generate a real and fast coupling phenomenon between cloth and fluid. When the distribution accuracy is high and the fluid scale is large, we need a lot of numerical calculation.

The quality and real-time of coupled animation are directly related to collision method. Gendelman [1] used ray casting method for collision between fluid, rigid body and deformation. This method can effectively detect penetration, but it has a large amount of calculation and is very time-consuming. Harada [2] proposed a collision method based on multiple spatial evenly divided grids to realize flow distribution coupling animation. Firstly, two kinds of spatial grids with different accuracy are established for cloth and fluid, which are used for bidirectional collision detection respectively.

Du [3] proposed an expansive continuous collision detection method for flow distribution coupling. This method improves the traditional continuous collision, regards the fluid particles as spheres with a certain radius and the cloth as triangular meshes with a certain thickness. According to the continuous collision detection of the two, the high-speed flow distribution coupling animation without penetration is realized. Akinci [4] proposed a boundary sampling method, which samples the surface of the elastic body to get the sampling points, and then couples the fluid and the elastic body according to the dynamic boundary conditions. Huber [5] proposed a coupling method of normal and tangential boundary conditions to realize realistic interactive animation of flow distribution by calculating the boundary conditions. Zhang [6] proposed a streaming coupling method based on PBD adaptive time step, which can effectively accelerate the simulation. Lu [7] proposed a simplified coupling animation between fluid and deformable variant. By simplifying the deformable variant, the simulation is carried out in a simplified space, which reduces the computational complexity.
The discrete collision detection method is difficult to achieve the collision penetration between deformation and fluid, so most of the research work uses CCD method to calculate the collision, but the calculation of CCD is large and it is difficult to avoid the problem of numerical accuracy; on the other hand, the deformation boundary sampling method is used to conduct collision coupling according to the constraint relationship between points, and the large grid deformation brings re mining What kind of calculation cost will increase. Using PBD algorithm is relatively simplicity, but the accuracy cannot be guaranteed.

In order to simulate the stable animation effect of cloth and fluid, this paper combines the CCD algorithm and PBD, iterating for many times to deal with collision, effectively reducing the error of collision detection. This paper proposes an efficient processing method based on continuous structure and one-to-one computing mode, which reduces the use of memory.

2. Continuous collision detection algorithm

Compared with discrete collision detection, continuous collision detection can get the earliest collision time of two moving objects. Continuous collision detection uses the linear interpolation from the current position to the predicted position as the motion path of the object. Two moving triangles use continuous collision detection, and carry out multiple point face and edge-edge tests according to the theoretical needs.

Each test needs to test four points. Let the standardized time interval be \([0,1]\). As shown in Figure 1, the purpose of continuous collision detection is that for continuous time variable \(t \in [0,1]\), the point and face collide, that is, the point is in the triangle or on the edge; or the edge collides with the edge, and the two edges are coplanar and intersect at the time. We consider the earliest collision time \(t_0 \leq t_1\), where \(t_i\) is any one that satisfies the condition. Although the form of point surface test and edge-edge test is different, they have the same characteristics in continuous collision detection, which is also a necessary condition for collision.

![Figure 1. Continuous collision detection diagram.](image)

Taking point plane as an example, suppose that the test object is point \(P\) and triangle \(T(A, B, C)\), the positions of the four points at the end of the previous frame are \(x_0^0, x_0^i, x_0^j, x_0^k\), and the predicted positions of the current frame are \(x^i, x^i, x^j, x^k\). At any time point \(t \in [0,1]\) position:

\[
x^t = tx^0 + (1-t)x^i
\]

According to formula (1), the position \(x^p, x^i, x^j, x^k\) of \(t\) at any time can be obtained. Firstly, the necessary condition of coplanarity is satisfied:

\[
x_0^0 + (1-t)x^i
\]
In the above formula, $x'_{Ap}$ represents the vector from the point $P$ to the vertex $A$ of the triangle at time $t$, and $N(t)$ represents the normal vector of the triangle at time $t$. The unknown quantity in formula (2) is $t$, and the equation is a univariate cubic equation about $t$. For the satisfied solution $t \in [0,1]$, it is necessary to test the point in the plane. Let the centroid coordinate of point $P$ in triangle $T(A,B,C)$ be $a,b,c$. $a,b,c$ is solved by formula (3):

$$x'_{p} = ax'_{A} + bx'_{B} + (1 - a - b)x'_{C}$$

(3)

For any solution, it corresponds to a centroid coordinate $a,b,c$. If the centroid coordinate $a,b,c$ satisfies $a,b,c \in [0,1]$ and $a+b+c \in [0,1]$.

3. Collision constraints and solution of PBD

In PBD algorithm, the cloth motion is regarded as the motion satisfying the constraint set under the action of external force. According to Muller [8], self-collision can also be regarded as a kind of constraint. When self-collision occurs, self-collision constraint is triggered, and the processing of self-collision is regarded as the process of constraint solving based on position.

![Figure 2. Continuous collision detection diagram.](image)

As shown in Figure 2, the collision constraint is shown. For example, for the particle $P$ and triangle $T(A,B,C)$, the end position of $P$ and $A,B,C$ in the previous frame and the predicted position in the current frame are $x'_{p},x'_{A},x'_{B},x'_{C}$ and $x'^{i}_{p},x'^{i}_{A},x'^{i}_{B},x'^{i}_{C}$ respectively. We need to determine whether the predicted position triggers the collision constraint:

$$C_{s}(P,T(A,B,C)) = (x'_{p} - x'^{i}_{A}) \cdot N^{i} - h_{0}$$

(4)

$C_{s}(P,T(A,B,C))$ is the self-collision constraint between particle $P$ and triangle $T(A,B,C)$, $N^{i}$ is the normal vector of the predicted triangle, and $h_{0}$ is the thickness of the initial cloth. Calculating the gradient of $C_{s}(P,T(A,B,C))$ for each point:
\[
\begin{align*}
\frac{\partial C_r}{\partial \mathbf{x}_r} &= N^l \\
\frac{\partial C_r}{\partial \mathbf{x}_j} &= -N^l + \frac{\mathbf{x}_r - \mathbf{x}_j}{|N^l|} M_j
\end{align*}
\] (5)

\(\mathbf{x}_r^l\) is a general term for the predicted positions of the three vertices of a triangle. According to Muller [8]:

\[
M_j = \frac{\partial N^l}{\partial (\mathbf{x}_p^l - \mathbf{x}_j^l)}
\] (6)

PBD self-collision constraint is originally used in cloth self-collision, and the self-collision constraint is the constraint between dynamic points and triangles. Therefore, this paper considers applying PBD self-collision constraint to the collision between fluid particles and cloth mesh triangles.

4. Collision detection of fluid-cloth coupling based on Multigrid

4.1. Fluid simulation

The fluid is regarded as a set of incompressible free particles, and the integral method based on SPH [9] is used to simulate the fluid. In the simulation, the particle position is used as the basic data, and the particle velocity can be expressed by the position of the front and back frames. Therefore, these effects are directly realized by adjusting the position of fluid particles when solving the collision and force of fluid particles. This is applicable to the coupling of fluid and cloth, that is, when the fluid and cloth mesh collide, the coupling effect can be achieved by modifying the position of fluid particles. This method can not only effectively reduce the number of caches, but also directly modify the position, which directly reflects the effect of force.

\[
\begin{align*}
\frac{\partial \rho}{\partial t} &= 0 \\
\frac{\partial \mathbf{v}}{\partial t} &= -\frac{1}{\rho} \nabla \rho + \mu \nabla^2 \mathbf{v} + \frac{F_{\text{ext}}}{\rho}
\end{align*}
\] (7)

In formula (7), \(\rho\) denotes the particle density. Since the fluid is incompressible, \(\rho\) can also be expressed as the mass of the fluid. \(\mathbf{v}\) is the velocity of the fluid, \(\frac{\partial \mathbf{v}}{\partial t}\) is the resultant acceleration of the fluid, and density gradient \(\nabla \rho\) is the pressure inside the fluid. The density function of fluid is expressed by smooth kernel function. Fluid simulation based on SPH method is one of the most common fluid simulation methods, which will not be described here.

4.2. Collision treatment strategy of fluid cloth coupling

The collision between fluid particles and triangles is set in a time step. Any fluid particle can collide with at most one cloth mesh triangle. Therefore, in the process of solving the bottom collision, if the fluid particles \(p_i^l\) and \(T_j^c\) collide and the earliest time of collision is calculated, the asynchronous one-way collision processing can be performed on the fluid particles \(p_i^l\) at the same time. The collision processing formula is as follows:
\[
\Delta p_i^l = k_{cl} (1 - t_{col}) \text{nor} \left( \sum_{p_a \in T_j^c} n_{p_a} \right) * \Delta x_{p_i^l}
\]  

(8)

Where \( \Delta p_i^l \) is the collision correction of \( p_i^l \), \( k_{cl} \) is the collision coefficient, \( t_{col} \) is the time of collision, \( \sum_{p_a \in T_j^c} n_{p_a} \) is the sum of the three vertex normal vectors of \( T_j^c \), and the unit normal vector is obtained after unitization. \( \Delta x_{p_i^l} \) is the distance vector of \( p_i^l \) from the current frame position to the predicted position. \( \sum_{p_a \in T_j^c} n_{p_a} \) is used instead of the normal vector of the triangle itself is to smooth the collision between the fluid particles and the cloth, so that the collision effect is as smooth as possible. In order to simulate the friction effect caused by collision, formula (9) is used to calculate the tangential position correction:

\[
\begin{aligned}
\Delta x_{p_i^l}^{\text{tan}} &= \frac{(\Delta p_i^l \cdot \text{nor} \left( \sum_{p_a \in T_j^c} n_{p_a} \right) * \Delta x_{p_i^l})}{\| \Delta x_{p_i^l} \|} \\
\Delta x_{p_i^l}^{\text{tan}} &= \mu_{cl} \max(0, 1 - \frac{|\Delta p_i^l|}{\Delta x_{p_i^l}}) \Delta x_{p_i^l}^{\text{tan}}
\end{aligned}
\]  

(9)

\( \Delta x_{p_i^l}^{\text{tan}} \) is the tangential position correction, \( \mu_{cl} \) is the viscosity coefficient between streamers. Finally, the position correction of fluid particles is the sum of the normal and tangential position corrections.

4.3. Load balancing and efficient processing

The solution of CCD is very time-consuming, if the solution of multiple CCD cubic equations is processed in one stream processing unit at the same time, the computational efficiency will be greatly reduced; but on the other hand, if the solution of cubic equations is carried out in multiple stream processing units, the data reduction will be the same.

This is a problem of GPU load balancing and efficient processing. Based on this, we propose an efficient processing method based on continuous structure and one-to-one computing mode. The organizational structure of the cache is shown in the following figure3:

![Figure 3. Load balancing of stream processing.](image-url)
"stream" is a stream processing unit. Each "stream" corresponds to the calculation of a CCD or PBD. \( U_{\text{cl.pair}} \) uses \( r32i \) to store collision pairs, and each potential collision pair only stores the index number of fluid particles. At the same time, the first \( U_{\text{cl.pair}} \) head address associated with \( T_j \) and the number of collision pairs associated with \( T_j \) are stored in another cache about \( T_j \), and the collision calculation of basic objects is carried out for potential collision pairs in each computing stream unit, which ensures that a stream computing unit can calculate CCD or PBD at most once, and ensures high parallelism and load balancing.

5. Experiment and results
In order to illustrate the experiment, we first introduce the flow of the algorithm in this chapter. The simulation calculation is carried out according to the inherent simulation method of cloth and fluid. The process is shown in the corresponding calculation flow on both sides of Figure 4.

![Figure 4. The flow of the cloth-fluid coupling algorithm](image)

According to the fluid particles, the mesh is threaded, and the motion bounding box of the fluid particles is constructed. Then, the intersection test of the fluid particle motion bounding box and the cloth mesh triangle motion bounding box is carried out; the collision pairs that pass the bounding box test enter the PBD and CCD iterative solution process. After processing, the cache information of the current frame is cleared and updating the corresponding cache data.

In order to test the high-level collision elimination effect of the fluid particles and cloth mesh, the collision set pair is calculated. The test scene is: the fluid particles start to fall from the top of the cloth, and the cloth deforms under the action of force. Firstly, some frames of the intermediate process are obtained according to the animation, and the detection results are calculated by using the method in this chapter. As shown in Figure 5 below, the acquisition of collision pairs between fluid particles and cloth mesh is more accurate. In Figure 5, the corresponding collision pair elements in the collision set represent fluid particles and mesh triangles respectively. The fluid particles belonging to the collision pair elements are marked in red, and the cloth mesh triangles belonging to the collision pair elements are marked in yellow.
In this paper, the effect comparison between PBD and CCD iterative solution of fluid cloth collision and DCD method of cloth collision is proposed. As shown in Figure 6, it is difficult to detect all the collision pairs by using CCD (PBD iterative solution), and it also causes the fluid to "bounce" off the cloth surface instead of flowing down the cloth mesh surface. The method proposed in this paper is based on the conservation of momentum and the position correction method is used to deal with the two-way streaming coupling, which can effectively deal with the streaming collision without the above phenomenon.

Figure 7 shows how the bounding box of fluid cloth coupling changes with frames using these two methods. As can be seen from the figure, after 1300 frames, there is a stable collision logarithm of about 40000. At the same time, the fluid finally stabilizes on the concave surface of cloth, which is consistent with the stable number of collision pairs. After 1300 frames, there will be a large number of fluid particles penetrating by CCD, which leads to the decrease of collision pairs between 1300 frames and 1380 frames.
6. Conclusions
We propose a CCD + PBD iterative method of fluid cloth coupling animation algorithm. In the collision detection link, the collision detection and processing of fluid particles and cloth mesh triangles are carried out on the basis of space sharing in parallel. In order to reduce the number of atom operations in GPU to the greatest extent, the method of batch address application is adopted. Finally, this paper uses a stream computing unit to process at most one CCD to allocate computing tasks, which further balances the computing load.

The disadvantage of this method is that the weight in the original PBD self-collision constraint solution can be well understood as the mass of the particle, but in two different media, the relative relationship of the mass is not well defined, and the adjustment of the weight in this paper only starts from the final effect, which is also a problem that this paper does not consider in depth.

References
[1] Guendelman E, Selle A, Losasso F, et al. Coupling water and smoke to thin deformable and rigid shells [J]. ACM Transactions on Graphics (TOG), 2005, 24 (3): 973 - 981.
[2] Harada T, Koshizuka S, Kawaguchi Y. Real-time Fluid Simulation Coupled with Cloth [C]. TPCG. 2007: 13 - 20.
[3] Du P, Tang M, Meng C, et al. A fluid/cloth coupling method for high velocity collision simulation[C]. Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry. ACM, 2012: 309 - 314.
[4] Akinci N, Cornelis J, Akinci G, et al. Coupling elastic solids with smoothed particle hydrodynamics fluids [J]. Computer Animation and Virtual Worlds, 2013, 24 (3-4): 195 - 203.
[5] Huber M, Eberhardt B, Weiskopf D. Boundary handling at cloth–fluid contact [C]. Computer Graphics Forum. 2015, 34 (1): 14 - 25.
[6] Zhang Y, Ban X, Liu X, et al. Adapting Time Steps for SPH Cloth-Fluid Coupling[C]. Cyberworlds (CW), 2016 International Conference on. IEEE, 2016: 143 - 146.
[7] Lu W, Jin N, Fedkiw R. Two-way coupling of fluids to reduced deformable bodies[C]. ACM Siggraph/eurographics Symposium on Computer Animation. Eurographics Association, 2016:67 - 76.
[8] Matthias Müller, Heidelberger B, Hennix M , et al. Position Based Dynamics [J]. Journal of Visual Communication & Image Representation, 2007, 18 (2): 109 - 118.
[9] Matthias Müller, Charypar D, Gross M. Particle-Based Fluid Simulation for Interactive Applications [C]. ACM Siggraph/eurographics Symposium on Computer Animation. ACM, 2003.