Cloth Simulation

Burak Ertekin
i7637445

MSc Computer Animation and Visual Effects

November 17, 2014

1 Abstract

In this paper, we explain how to make a Cloth Simulation using Mass Spring Model. We explained the internal dynamics of the spring model together with the integration methods. Explicit Euler integration method is used to obtain new positions and velocities at each time step. In the end, we talked about collision detection methods that are efficient and powerful enough to use with this model. Our aim was to implement a simple yet realistic cloth simulation.

2 Introduction

Computer Animation is started to being used more frequently in this past decade. The motion picture industry is using animation technologies to create more influential and fascinating work. Computer games gets more realistic with faster and efficient animation solutions. Cloth Simulation is an essential element in computer animation. We are using cloth-like material everyday. It is crucial to implement this feature in our animations in highly detailed and speed-efficient manner. The progress of cloth simulation started with the idea of creating elastically deformable objects. And today we have highly interactive, real-time solutions.

Mass spring system is the most common approach in soft body dynamics. Other approaches such as, finite element simulation, energy minimization methods exists. Even though some of these approaches give physically more accurate results, it is time inefficient in terms of computation. Therefore mass spring systems are widely used and developed.

In the implementation, we are using mass spring system as models. With Euler Integration and a simple collision detection algorithm, we aim to animate a fine cloth simulation. In this paper, firstly we will explain what is a mass spring system. We will get in details with the forces used in the system. Then we will pass on integration methods. Mainly we will talk about Euler Integration methods however we will also mention about other integration methods. Later on, we will see different collision detection and response methods. Finally we will mention on some interesting future work on cloth simulation.

3 Related Work

Traditionally, models in computer graphics were geometrically or algebraically defined primitives. Terzopoulos et al. suggested simulating physical properties such as tension and rigidity to model deformable objects[1]. This would give the static shapes of the models. And by including other physical properties such as mass and damping, they suggested simulating dynamics of the objects. Terzopoulos and Carignan et al. discretizes cloth as rectangular mesh[2] while Baraff et al. models the cloth as a triangular mesh of particles[3]. Wang et al. stated the advantage of triangular mesh over rectangular mesh[4].

Terzopoulos derived energy functions using a continuum formulation in their implementation. However they did not use any damping force in the internal spring force equation, only Hooke's Law: $-k\hat{x}$ was used. Carignan added a force which damps cloth stretch and shear but not bend. Cloth is a material with strong resistance to stretching motions while it has very little resistance to bending and shearing motions. As Baraff mentioned, "this results in a stiff underlying differential equation of motion". Since explicit methods are bad choice to solve stiff equations because of the necessity of small time steps, Baraff demonstrated implicit methods overcoming the computation limits of explicit methods[3]. Desburn et al. presented another method: a rapid approximation of the implicit integration[5]. Kang et al. came up with another approximation method for integration[6]. Volino et al. proposed an adaptation of the implicit Midpoint method.[7] "Collisions are a major bottleneck in cloth simulation"as said by Bridson et al.[8]. Several different approaches has been made on this topic however these approaches differ considering other methods implemented on the model. Volino et al. proposed the correction of the particle acceleration after a collision[9]. This acceleration is used as a collision force by the model and integrate with the other existing force of the particle. Baraff suggested altering positions of the cloth particles[3]. Bridson et al. combined geometric collisions with repulsion force. Selle et al. used parallel Gauss-Seidel collision/repulsion response[10]. Neal presents "CollisionGrid"system which uses height representation of terrain to approximate the collisions[11].

4 Mass Spring Model

Mass Spring System is an approach to model flexible and deformable objects. In this approach, we have a set of points connected by weightless elastic springs forming the model. The forces on these springs are represented by Hooke's Law, that is: $F = -kx$. Where k is the spring constant. In this approach, we can specify k according to our needs in the model. It is common to set a damper with each spring to create a better control of the motion[12]. A simple damping force on the i-th mass point can be represented as:

$$F_{d_i} = -k_d v_i \ (k_d > 0)$$

Where k_d is the damping factor and v_i is the current velocity of i-th mass point[6]. We use Hooke's Law and this simple damping formula to obtain our internal spring forces in our implementation. And we obtain this final equation:

$$F_i = -k_s \Delta x_i - k_d \Delta v_i \ (k_d > 0)$$

We have three different springs in a mass spring system: structural, shear and flexion springs. Interactions between different parts of the cloth are formed by structural springs. The resistance against bending is modeled by flexion springs. And shearing springs are used to model shearing resistance. Rectangular mesh has a small resistance to the diagonal bending. Wang suggested adding diagonal flexion spring for this problem[4][13]. In order to reduce the number of springs for each mass, Wang eliminated the shearing springs in one direction and proposed using diagonal flexion spring. With this they obtained a rectangular triangle mesh. Instead, they use an equilateral triangle mesh. With this new structure, they only use two types of springs: structural and flexion since triangular mesh can resist shearing itself. Even though it has the same computation, it has more resistance against bending, stretch and compression[4].

5 Integration

5.1 Explicit Euler Integration

In our implementation, we use Newton's equation of motion and integrate it with explicit Euler method to obtain the new position and velocity for our system. Explicit Euler integration scheme can be shown as:

$$v_i^{t+h} = v_i^t + F_i^t \frac{h}{m_i}$$

$$x_i^{t+h} = x_i^t + v_i^{t+h} h$$

Where v_i^t denotes the velocity of the i-th mass point at time t while F_i^t is the force acting on the mass point and x_i^t denotes the location[6]. With this method we can easily calculate x_i^{t+h}. Even though explicit Euler integration is a very simple scheme, it creates numerical instability unless the time step in simulation is small enough. As Desburn stated, the time step dt which must be inversely proportional to the square root of the stiffness[5].

5.2 Implicit Euler Integration

Implicit integration schemes offers stable animations with large time steps without problem. Therefore implicit methods are generally used for cloth simulations, it provides faster computation. Implicit Euler integration scheme can be shown as:

$$v_i^{t+h} = v_i^t + F_i^{t+h} \frac{h}{m_i}$$

$$x_i^{t+h} = x_i^t + v_i^{t+h} h$$

The only difference is F_i^t is replaced with F_t^{t+h}. In this integration method, the solution is not straightforward. To solve a differential equation like $\frac{dY}{dt} = f(Y)$ numerically, the derivative is not $f(Y^t)$ throughout the time interval, but the derivative is a weighted average of derivative $f(Y^t)$ at the beginning of the interval and $f(Y^{t+h})$ at the end of the interval. The update formula is:

$$Y^{t+h} = Y^t + h[(1-\lambda)f(Y^t) + \lambda f(Y^{t+h})]$$

λ is between 0 and 1. When λ is 0, the update formula becomes the explicit Euler method. When λ is 1, the formula is the implicit Euler method. And this substantiates the implicit formula above[6].

5.3 Other Integration Methods

Implicit methods gives you more stable results however these results are necessarily not more accurate. We can obtain more accuracy by using more complex integration methods such as Rosenbrook method. Although we need to solve four times a linear system of each integration step, our computation times becomes longer than implicit methods[7]. Volino suggested implicit Midpoint method, however in the case of nonlinear problems, they had numerical instability. Runge-Kutta integration method is a good option for numerical accuracy however it creates discontinuities in motion in case of collision. Since we are interested in cloth simulation, it is unwise to use this method in our model[5].

6 Collision Detection

A realistic collision detection mechanism is crucial for any physically plausible animation system. It is hard to implement a collision detection algorithm which works for any possible case in cloth simulation. However it is easy to detect the deformations on a cloth like object because of its thinness; the objects most likely pierce through the cloth and create an unrealistic look. Moreover, in cloth simulation all the particles has the potential to collide with each other and the environment in any given time step[8]. In order to have more realistic simulation, we have to increase particle count however this will also increase the complexity of our algorithm. On the other hand, self-collisions are very important for cloth simulation because we can form folds and wrinkles by cloth/cloth interactions.

In our simulation we use a simple method of position alteration as our collision detection/response solution. Even though this solution has some problems in some cases, it generally works fine with low-resolution work. As Baraff mentioned, cloth/cloth collision are detected by checking the particles and the edges of cloth triangles. They overcame the cloth/cloth collision problem by inserting a strong damping force to push the cloth apart. As for the cloth/object intersections, they test each individual cloth particle with the faces of the object. The faces of the object are grouped to a hierarchical bounding box tree. The problem with position alteration is that particle's neighbours affected by this alteration[3]. To overcome this problem, Baraff suggested to add an arbitrary correction value y_i which is used only to move to desired location during the backward Euler step[3].

$$\Delta x_i = h(v_{0_i} + \Delta v_i) + y_i$$

Bridson proposed to use an impulse needs to be applied to two points in the cloth: $I\hat{n}$ and $-I\hat{n}$ where I is the magnitude and $\hat{n}$ is the direction. For point/triangle case, where $\vec{x_1}\vec{x_2}\vec{x_3}$ are interior points of triangle with barycentric coordinates w_1, w_2, w_3 interacting with point $\vec{x_4}$, the adjusted impulses are

$$\tilde{I} = \frac{2I}{1 + w_1^2 + w_2^2 + w_3^2}$$

$$\vec{v_i}^{new} = \vec{v_i} + w_i(\tilde{I}/m)\hat{n} \quad i = 1,2,3 \qquad \vec{v_4}^{new} = \vec{v_4} - (\tilde{I}/m)\hat{n}$$

For edge/edge case; when a point with position a along the edge $\vec{x_1}\vec{x_2}$ interacts with another point with position b along the edge $\vec{x_3}\vec{x_4}$, the adjusted impulses are

$$\tilde{I} = \frac{2I}{a^2 + (1-a)^2 + b^2 + (1-b)^2}$$

$$\vec{v_1}^{new} = \vec{v_1} + (1-a)(\tilde{I}/m)\hat{n} \qquad \vec{v_2}^{new} = \vec{v_2} + a(\tilde{I}/m)\hat{n}$$

$$\vec{v_3}^{new} = \vec{v_3} - (1-b)(\tilde{I}/m)\hat{n} \qquad \vec{v_4}^{new} = \vec{v_4} - b(\tilde{I}/m)\hat{n}$$

"Weighting the impulses in this way introduces appropriate torques for off-center interactions as well as giving continuity across triangle boundaries, and converges to the expected formulas when the interior points approach mesh nodes"[8].

Geometric self-collisions which occur in folding and contact situations can be really expensive. To resolve this problem, we use self-repulsions. It is similar to collisions, we have both point/triangle and edge/edge interactions. We have inelastic collision impulse to stop approaching interactions pairs and elastic repulsion impulse to push interaction pairs further apart[10]. Inelastic repulsion impulse is $I_c = mv_n/2$ where v_n is the normal velocity and m is the mass of particles. Elastic repulsion of a spring with a stiffness k is

$$I_r = -min\left(\Delta tkd, m\left(\frac{0.1d}{\Delta t} - v_n\right)\right)$$

where

$$d = h - (x_4 + w_1x_1 - w_2x_2 - w_3x_3)\hat{n}$$

and where w_i are the barycentric weights of the free point x_4 projected to the triangle, $x_{\{1,2,3\}}$ are the triangle's point locations, h is the repulsion thickness and $\hat{n}$ is the triangle normal[8][10].

Another approach to collision detection is to use a height field representation of terrain to approximate collision detection and response[11]. The "CollisionGrid"system defines the geometry of objects in single axial direction. By projecting the position of each particle on the CollisionGrid plane and bilinearly interpolate the nearest four data points to determine the height of the surface. The value is compared to height of the particle along the collision axis to determine if there is a collision. As for collision response, its resolved manually by changing position and velocity of particles[11].

7 Conclusion and Future Work

Cloth Simulation is a very fascinating topic in computer animation for a long time. We tried to explain the basic way to create a cloth. However the area of research for this topic is extensive. Rendering, collision detection and response, integration methods, fabric, shearing and tearing can be told as the main topics[14][15]. Due to its particle based structure, the computation costs increases drastically when we increase the particle count. In order to obtain more realistic look on the simulation, we need to use more particles in the simulation. In addition to the existing structure; when there is a tearing happens on the cloth, new particles are formed. Therefore, there is always a new demand for new methods in cloth simulation to create more realistic and interactive work.

References

[1] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 205–214, Aug. 1987. [Online]. Available: http://doi.acm.org/10.1145/37402.37427

[2] M. Carignan, Y. Yang, N. Magnenat-Thalmann, and D. Thalmann, "Dressing animated synthetic actors with complex deformable clothes," in *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1992*, J. J. Thomas, Ed. ACM, 1992, pp. 99–104. [Online]. Available: http://doi.acm.org/10.1145/133994.134017

[3] D. Baraff and A. Witkin, "Large steps in cloth simulation," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 43–54. [Online]. Available: http://doi.acm.org/10.1145/280814.280821

[4] X. Wang and V. Devarajan, "2d structured mass-spring system parameter optimization based on axisymmetric bending for rigid cloth simulation," in *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry*, ser. VRCAI '04. New York, NY, USA: ACM, 2004, pp. 317–323. [Online]. Available: http://doi.acm.org/10.1145/1044588.1044657

[5] M. Desbrun, P. Schröder, and A. Barr, "Interactive animation of structured deformable objects," in *Proceedings of the 1999 Conference on Graphics Interface '99*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 1–8. [Online]. Available: http://dl.acm.org/citation.cfm?id=351631.351638

[6] Y.-M. Kang, J.-H. Choi, H.-G. Cho, and C.-J. Park, "Fast and stable animation of cloth with an approximated implicit method," in *Proceedings of the International Conference on Computer Graphics*, ser. CGI '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 247–. [Online]. Available: http://dl.acm.org/citation.cfm?id=792759.793087

[7] P. Volino and N. M. Thalmann, "Implementing fast cloth simulation with collision response," in *Proceedings of the International Conference on Computer Graphics*, ser. CGI '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 257–. [Online]. Available: http://dl.acm.org/citation.cfm?id=792759.793086

[8] R. Bridson, R. Fedkiw, and J. Anderson, "Robust treatment of collisions, contact and friction for cloth animation," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 594–603, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566623

[9] P. Volino, M. Courchesne, and N. Magnenat Thalmann, "Versatile and efficient techniques for simulating cloth and other deformable objects," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 137–144. [Online]. Available: http://doi.acm.org/10.1145/218380.218432

[10] A. Selle, J. Su, G. Irving, and R. Fedkiw, "Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 2, pp. 339–350, 2009. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TVCG.2008.79

[11] K. Neal, "Collision approximation for real-time cloth simulation," in *ACM SIGGRAPH 2004 Posters*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 89–. [Online]. Available: http://doi.acm.org/10.1145/1186415.1186520

[12] R. Parent, *Computer Animation, Second Edition: Algorithms and Techniques*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[13] X. P. Institut and X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behavior," in *In Graphics Interface*, 1996, pp. 147–154.

[14] M. S. Souza, A. von Wangenheim, and E. Comunello, "Fast simulation of cloth tearing," *SBC*, vol. 5, no. 1, 2014.

[15] N. Metaaphanon, Y. Bando, B. Chen, and T. Nishita, "Simulation of tearing cloth with frayed edges," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1837–1844, 2009. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2009.01561.x