Implementing Provot's Deformation Constraints in Mass-Spring Models of Cloth Behaviour

Jack Pilley
V00805926

Abstract

Mass-spring models are a common and intuitive method of simulating cloth, however this model allows for super elastic deformation which is unrealistic. Areas under high tension will stretch beyond the point real fabric would tear, this is called "super-elongation." This paper describes the implementation process, and an evaluation of, X. Provot's solution to super elongation in mass-spring cloth models [1]. While the technique described by Provot did reduce super elongation and improve the stability of the simulation, I was unable to fully replicate the results of their paper.

1 Introduction

Provot proposes a mass-spring model for simulating cloth where the cloth is represented by a grid of particles having mass, connected by springs. The spring arrangement Provot uses (figure 1) is commonly used and appears in other

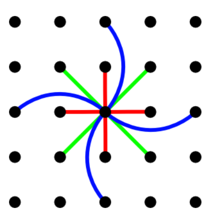


Figure 1: Spring configuration

works, such as [2] and [3]. Using Provot's terminology, the black dots represent the particle masses, the red lines indicate structural springs, the green lines represent shear springs, and the blue lines represent flexion springs. The structural springs resist stretching and compression, the shear springs resist shearing, and the flexion springs resist bending. An issue with this model, identified by Provot [1], is the "super-elastic" effect. Springs tied to fixed particles, or otherwise experiencing extreme forces, will stretch beyond the point real cloth would tear. Provot's solution is to move neighbouring particles to within a maximum distance after each simulation step. This secondary movement is done once per step for each structural and shear spring without any consideration of order. As this movement might cause other springs to become super-elongated, Provot's method relies on the assumption that only a small number of springs will ever be in this state at once, specifically those connected to fixed particles.

While increasing the stiffness of the springs would produce similar results to the solution Provot proposes, increasing the stiffness also requires shorter time steps during integration. This is because the simulation becomes unstable if the time step is longer than the period of the spring system, and a higher stiffness decreases that period.

Occasionally in this paper, the simulation will be described as "exploding." In these cases, the particles began to take on large random velocities causing the cloth to lose any coherence (figure 2). A simulation described as "unstable" either exploded inconsistently, or would temporarily lose coherence.

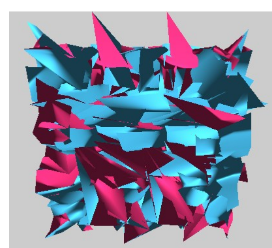


Figure 2: An exploding simulation

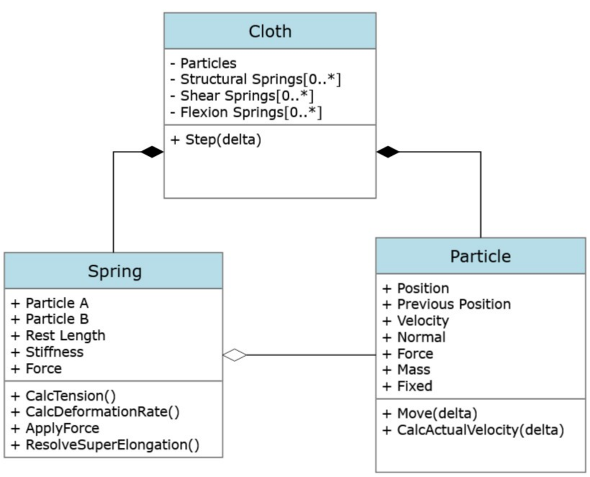
2 Related Work

The earliest use of particles to simulate cloth I could find was from 1988 by Haumann and Parent [4]. While their method wasn't specifically meant to simulate cloth, rather just elastic surfaces, it could produce visually pleasing results. Breen, House, and Getto [5] proposed a solution using an energy function for each particle and searching for the minimum energy state at each simulation step. The energy function is defined as the sum of four functions: repel (prevent particles intersecting), stretch (keep particles together), bend (the angle between each set of three adjacent particles), and trellis (shearing parallel to the particle grid). The search for the minimum energy state was done by stochastic gradient descent. This method shares some similarities with Provot's method for preventing super-elasticity, although the use case is different. Breen, House, and Getto [5] were specifically investigating cloth draped over some object, which didn't involve fixed particles in the cloth around which the super-elastic effect occurs.

3 Overview

Performance is especially important for real time applications. While a later section will discuss this in depth, we will first briefly describe the data structures used in this implementation and how they relate to performance.

3.1 Data Structures



Firgure 3: Class UML Diagram

Memory layout is important for performance on modern systems. Keeping data contiguous and iterating in a predictable pattern will reduce cache misses and allow the CPU to prefetch data [6]. Figure 3 provides an overview of the classes used in the implementation. The cloth is represented by a class holding an array of particles and three arrays of springs, one for each spring type (structural, shear, or flexion). The purpose of dividing the springs into three arrays is to minimize the number of branches required while iterating over the springs. This lets the springs be treated differently without potentially disrupting the CPU's instruction pipelining [7]. Arrays were selected over other storage classes because they provide excellent cache locality [6].

The particles are represented by a struct holding five three dimensional vectors which store the particle's current position, previous position, velocity, normal direction, and accumulated force. The particle also stores its mass as a scalar value, and if it is fixed (immovable) as a boolean value.

The springs hold pointers to the particles they connect, their rest length and stiffness as scalars, and a three dimensional vector holding the force between the two particles calculated that step.

3.2 Forces and Integration

Following Provot's paper, the force generated by each spring is given by

$$F = -K\left(\overrightarrow{AB} - \frac{l_0 \cdot \overrightarrow{AB}}{\|\overrightarrow{AB}\|}\right)$$

where K is the stiffness, $\overrightarrow{AB}$ is the vector from particle A to particle B and l_0 is the rest length of the spring. F is added to A's accumulated force and $-F$ is added to B's accumulated force. Notably, Provot does not include damping in the spring force.

Where as the spring forces can be thought of as internal to the cloth, external forces are also applied. The force of gravity is calculated is calculated using

$$F_g = m\vec{g}$$

where m is the mass of the particle and $\vec{g}$ is the acceleration due to gravity. Damping is applied as an external force in the form of wind resistance, rather than as part of the spring forces. The damping force is given by

$$F_d = -D \cdot \vec{v}$$

where D is the damping coefficient and $\vec{v}$ is the particle's velocity. Wind is the last external force, given by

$$F_w = [\vec{n} \cdot (\vec{w} - \vec{v})] \cdot \vec{n}$$

where $\vec{n}$ is the normal of the cloth surface as the location of the particle, $\vec{w}$ is the velocity of the wind, and $\vec{v}$ is the velocity of the particle. All of these forces are accumulated by each particle along with the internal forces.

Integration is done using a forward Euler method.

$$\vec{a}_{t+\Delta t} = \vec{F}_{Total} / m$$

$$\vec{v}_{t+\Delta t} = \vec{v}_t + \vec{a}_{t+\Delta t} \cdot \Delta t$$

$$\vec{p}_{t+\Delta t} = \vec{p}_t + \vec{v}_{t+\Delta t} \cdot \Delta t$$

Note that although it isn't used in the integration, the previous position needs to be saved. The reason for this is explained is the next section.

3.3 Correcting Super Elongation

We define super elongation to be when a spring's deformation rate exceeds some selected value based on how elastic the cloth should be. Deformation rate is defined as

$$\tau = {l - l_0} / {l_0}$$

where l is the current length of the spring and l_0 is the spring's rest length. A spring that is super elongated has been stretched beyond the point real fabric would tear.

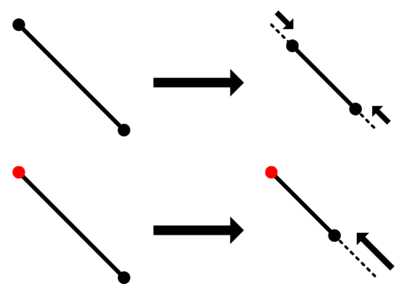


Figure 4: Super elongation correction. The top section is the process for two free particles. The bottom is the process for one free particle and one fixed particle.

To correct for super elongation, Provot moves the connected particles together so that the deformation rate of the spring is exactly the maximum rate (figure 4).

Provot's paper does not consider that after correcting for super elongation, the velocity of a particle calculated in the integration step might no longer match the final position of that particle. To correct for this, we recalculate the actual velocity of the particles using

$$\vec{v}_t = {\vec{p}_t - \vec{p}_{t-\Delta t}} / {\Delta t}$$

after the correction step. It is important to recalculate the particle's velocity because correcting super elongation will prevent the spring from applying a force large enough to counter the other forces acting on the particle. As velocity is accumulated over multiple frame, this will cause the simulation to eventually explode, as the particle will be treated as having an incredibly high velocity.

The issue of correcting a particle's velocity could be avoided by using an integration method that does not rely on velocity, such as Verlet integration. In that case, the velocity calculation can simply be deferred to after correcting for super elongation.

4 Evaluation

4.1 Performance

With 2,500 particles and 14,502 springs, and applying Provot's adjustment of super elongated springs, each simulation step took an average of 0.54 milliseconds to compute with a standard deviation of 0.12 milliseconds. Without applying the adjustment, each step took an average of 0.38 milliseconds with a standard deviation of 0.08 milliseconds. The computer used to run these simulations uses Windows 10 with 16GB of RAM, a Nvidia GTX970 GPU, and an Intel Core i7-8700 CPU. This indicates that the adjustment step slows the simulation by about 40%, which is significantly more than the 15% reported by Provot. However, simulating an equivalently stiff fabric by increasing the stiffness of the springs rather than applying the adjustment method required a time step roughly one tenth as long to maintain simulation stability (i.e. running ten times the number of steps per second) as compared to using the adjustment method. This performance difference is in line with Provot's reported results of a simulation taking 9 minutes when using the stiffer springs, and 1 minute when using the adjustment method to achieve the same results.

4.2 Qualitative Results

While Provot's method for correcting super elongation does improve the results of the simulation (figure 5), there is still excessive stretching in situations with few fixed particles (figure 6). This does not match Provot's results, which showed a flag in a similar configuration without excessive stretching.

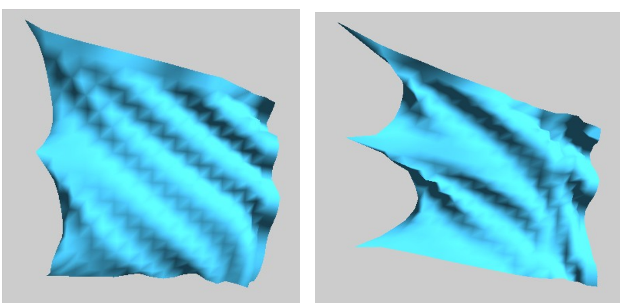


Figure 5: The left image has super elongation correction enabled, the right image has it disabled.

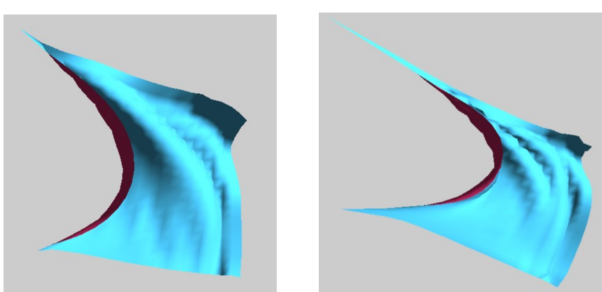


Figure 6: The left image has super elongation correction enabled, the right image has it disabled. Note that there is still excessive stretching even with the correction enabled.

Given these results, an attempt was made to find modifications that would provide results closer to Provot's. The following proved successful. First, Provot divides the deformation rate calculations and super elongation correction into two discrete steps. All deformation rates are calculated, then the adjustments are made. It was found that calculating the deformation rate, then immediately adjusting the examined spring provided more stable results. Second, running multiple correction steps within a single simulation step gave results closer to Provot's. As a drawback, this requires more time to compute, and it is unclear if it is more efficient than increasing stiffness and using a smaller time step.

The following modifications were unsuccessful. Sorting the springs by deformation rate and then adjusting from most deformed to least deformed caused vibrations to appear around high tension areas. Temporarily fixing particles which had already been adjusted in a simulation step caused significant instability and occasionally made the simulation explode.

5 Conclusion

While the exact results could not be replicated, Provot's solution to super elongation does improve both the stability of cloth simulations, and reduces the computation required to achieve desirable results.

A possible modification that should be explored in the future is adjusting springs in order of their distance from the fixed particles. By temporarily fixing particles which had already been adjusted, this would pull all particles towards the normally fixed particles. However, attempting to temporarily fix particles caused instability in other scenarios.

References

[1] X. Provot, "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour", in *Proceedings of Graphics Interface '95*, Quebec, Canada, 1995, pp. 147–154, doi: https://doi.org/10.20380/GI1995.17

[2] E. Lengyel, "Fluid and Cloth Simulation," in *Mathematics for 3D Game Programming and Computer Graphics*, 3rd Ed. MA, USA: Course Technology, 2012, ch. 15, sec. 2, pp. 457 – 460

[3] D. House and D. Breen, “Particle Representation of Woven Fabrics,” in *Cloth Modeling and Animation,* 1st Ed. NY, USA: A K Peters/CRC Press, 2000, pp. 55 – 77

[4] D. R. Haumann and R. E. Parent, “The behavioral test-bed: Obtaining complex behavior from simple rules”, *The Visual Computer*, vol. 4, no. 6, pp. 332–347, Nov. 1988.

[5] D. E. Breen, D. H. House, and P. H. Getto, “A physically-based particle model of woven cloth,” in *The Visual Computer*, vol. 8, no. 5, pp. 264–277, Sep. 1992.

[6] S. Meyers, NOKIA Technology Center Wrocław. Code::dive Conference 2014 - Scott Meyers: Cpu Caches and Why You Care. (Jan. 5, 2015). Accessed: Apr. 6, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=WDIkqP4JbkE

[7] C. Chen, G. Novick, and K. Shimano, “Pipelining.” stanford.edu. https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/pipelining/index.html (accessed Apr. 6, 2023).