Effects of Different Constraint Enforcement Methods for Verlet Integration in Cloth Simulation

UCMERCED
Computer Graphics Lab

Oktar Ozgen Marcelo Kallmann
University of California Merced - http://graphics.ucmerced.edu

Introduction

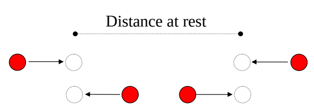
Thanks to its simplicity, performance and stability, one very popular method for real-time cloth simulation (in particular in computer games) is based on the so-called Verlet integration method[1]. This poster presents comparisons of three different constraint enforcement methods for Verlet integration. Constraint enforcement is used to maintain edge length within limits.

Verlet Integration

The Verlet method stores and uses the current and previous positions of each particle (cloth mesh vertices) as the state of the system. The velocity is thus implicitly represented by positions.

$$P_{next} = 2\,P_{cur} - P_{prev} + a\,dt^2$$

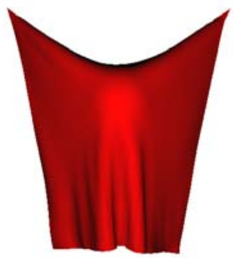
Constraint enforcement is mainly required for correcting the distance between the particles after one integration step, in order to prevent edges to deform unrealistically.

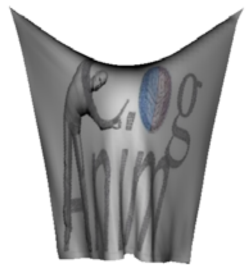


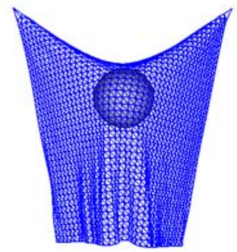
Constraint Enforcement

The usual method simply corrects the lengths over several iterations. As correcting one pair of particles may disturb a neighbor pair of particles, we seek to analyze the impact on choosing a particular order to process the constraint enforcements.

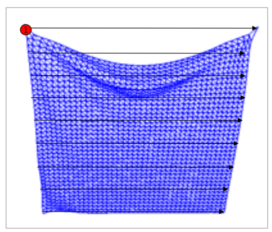
Constraint Enforcement Methods



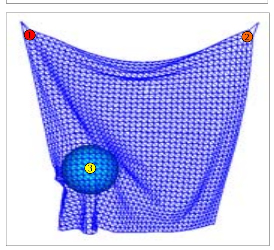




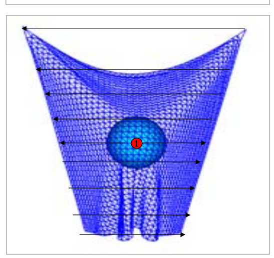
We mainly implemented and compared three different constraint enforcement methods:



Method 1: In this method, the edges of the cloth mesh are traversed in a straightforward fashion: starting from the upper left corner and ending on the lower right corner. The method is independent of the current system state. This is the traditional method that is widely used[1,2].

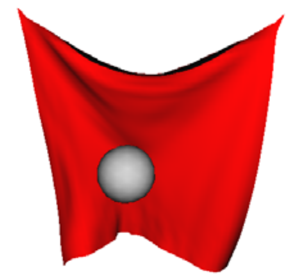


Method 2: In this method, the order in which the edges are processed is prioritized. All the pairs of particles are sorted in a priority queue in such a way that the most disturbed edge is corrected first and the least disturbed edge is corrected last.



Method 3: In this method, the contact with the object is first determined. The scan starts at the first collision point with the object. The assumption is that the disturbance is higher around this point. Scanning is then processed towards two different directions at the same time, without the need for sorting.

Results



In terms of performance, the first and the third method perform better than the second method. This is because the second method spends some time to sort the edges in each iteration. Our implementation of a cloth with resolution of 40x40 particles, using the first or the third method runs at around 39 FPS. Our implementation of the second method runs at 28 fps.

It is difficult to asses the differences on the visual quality of each obtained animations, but a demonstration of the obtained results will be presented. Numerical comparisons of each obtained simulation are left to future work.

References

[1] T. Jakobsen. Advanced Character Physics, GDC 2001, San Jose.
[2] C. Zeller. Cloth Simulation on the GPU. NVIDIA Corporation. Siggraph, 2005.