

IMPULSE-BASED SIMULATION OF INEXTENSIBLE CLOTH

Jan Bender and Daniel Bayer

*Institut für Betriebs- und Dialogsysteme
Universität Karlsruhe
Am Fasanengarten 5
76128 Karlsruhe
Germany*

ABSTRACT

In this paper an impulse-based method for cloth simulation is presented. The simulation of cloth is required in different application areas like computer animation, virtual reality or computer games. Simulation methods often assume that cloth is an elastic material. With this assumption the simulation can be performed very efficiently using spring forces. The problem is that many textiles cannot be stretched significantly. A realistic simulation of these textiles with spring forces leads to stiff differential equations which cause a deterioration of performance. The impulse-based method described in this paper solves this problem and allows the realistic simulation of inelastic textiles.

KEYWORDS

Cloth simulation, physically-based modelling, impulse-based simulation, inelastic textiles

1. INTRODUCTION

The simulation of cloth and fabrics is an import area of research and has many applications like computer animation, games and virtual environments. Therefore this area has a long history of research. House and Breen (2000) and Magnenat-Thalmann and Volino (2005) give a good survey of the research done in cloth animations. Current research problems in clothing simulation are summarized in Choi and Ko (2005).

Cloth is often simulated as an elastic material due to performance reasons but many textiles are not noticeably stretchable. Many approaches use spring forces for the simulation of stretchable cloth. In fact today's clothing simulations are mainly based on spring-mass systems of particles (see, e.g. Choi and Ko (2002)). The realistic simulation of an inextensible piece of cloth with such an approach would require large spring constants. This leads to stiff differential equations which are hard to integrate and decrease the numerical stability (Hauth et al. (2003)). The integration can only be performed by reducing the time step size significantly or by using special methods. In both cases the consequence is a low performance.

Another approach is to use constraints to enforce the conditions instead of integrating the spring forces directly (see, e.g. Fuhrmann et al. (2003)). Especially if small tolerances are demanded this approach provides a good alternative. A system for constrained-based simulation of inextensible cloth based on the lagrangian mechanics is given in Goldenthal et al. (2007).

In this paper a new constraint-based approach is presented that uses impulses for the simulation of cloth. It is shown that the use of impulses instead of spring forces allows an accurate solution of the constraints.

2. CLOTH SIMULATION

A piece of cloth is represented by a mesh of particles that are linked by distance constraints. The following sections describe the simulation of particles and constraints and the handling of their dependencies in a mesh.

2.1 Particle simulation

A particle is a body that has a mass but no volume. Since a particle has no dimension, it has just translational degrees of freedom and no rotational ones. In the simulation the state of a particle is described by its mass m its position c and its linear velocity v . In general the mass of a particle is constant during the simulation. A time step of a particle is performed by integrating its velocity and its position over time. It is assumed that the sum of all external forces acting on a particle, like e.g. gravity, is constant during the time step. In this case the velocity and position after a step of size h can be computed directly by the following equations:

$$v(t_0 + h) = v(t_0) + \int_0^h \frac{F_{\text{ext}}}{m} dt = v(t_0) + \frac{F_{\text{ext}}}{m} h$$

$$c(t_0 + h) = c(t_0) + \int_0^h v(t_0) + \frac{F_{\text{ext}}}{m} t dt = c(t_0) + v(t_0) h + \frac{1}{2} \frac{F_{\text{ext}}}{m} h^2$$

where F_{ext} is the sum of all external forces and t_0 is the start time of the simulation step. In the simulation it is differentiated between dynamic and static particles. In contrast to dynamic particles static ones have a fix position and no velocity.

2.2 Distance constraints

A distance constraint for two particles a and b consists of two parts: a position constraint and a constraint for the velocities of the particles. The position constraint is defined as follows:

$$d(t) - d(0) = 0 \quad \text{with} \quad d(t) = |c_a(t) - c_b(t)|.$$

This means that the difference of their actual distance and their distance at the beginning of the simulation must be zero to satisfy this constraint. Hence their distance must stay constant over time. In the simulation this constraint is satisfied by impulses. At time t these impulses are computed by using a preview of the simulation step. For a distance constraint the distance of the corresponding particles $d(t+h)$ after the next simulation step of size h is determined by integrating the particle positions (see section 2.1). The difference between the value $d(t+h)$ and the distance $d(0)$ at the beginning of the simulation is exactly the error that would occur, if the simulation step would be performed without regarding the position constraint. This error $e = d(t+h) - d(0)$ can be eliminated by computing impulses and applying them at the beginning of the simulation step at time t . Since an unconstrained particle has a linear motion, the two particles must change their relative velocity by e/h in direction of the constraint in order to change their distance by e in a time step of size h . The direction of the constraint is given by the vector from particle a to particle b . In order to satisfy the conservation of momentum two impulses p and $-p$ of the same magnitude and opposite directions are applied to perform this velocity change. The impulse p can be determined by solving the equation:

$$\Delta v_a(p, t) - \Delta v_b(-p, t) = (c_b(t) - c_a(t)) \frac{e}{h}$$

where $\Delta v_a(p, t)$ is the velocity change of particle a , when the impulse p is applied. The differentiation between dynamic and static particles is done using the following value for computing the velocity change:

$$k_a = \begin{cases} 1/m_a & , \text{if particle } a \text{ is dynamic} \\ 0 & , \text{otherwise.} \end{cases}$$

The resulting equation for the impulse is:

$$\Delta v_a(p, t) - \Delta v_b(-p, t) = k_a p - k_b (-p) = (k_a + k_b) p = (c_b(t) - c_a(t)) \frac{e}{h}$$

which can be solved, if at least one of the particles is dynamic and the particles have different positions. It is assumed that the constraint is satisfied at the beginning of a simulation step, so the particles cannot have the same positions. After the determination of the impulse p it must be applied to the particles in positive and negative direction respectively. Due to the resulting velocity change, the position constraint will be satisfied after the simulation step.

The second part of the distance constraint is a constraint for the velocities of the particles:

$$(v_b(t) - v_a(t))(c_b(t) - c_a(t)) = 0.$$

This means that the relative velocity of the particles in direction of the constraint must be zero. In contrast to the position constraint, here the required impulse can be determined directly without a preview because an impulse causes an instantaneous velocity change. The impulse that satisfies the constraint is computed by:

$$p = \frac{(v_b(t) - v_a(t))(c_b(t) - c_a(t))}{k_a + k_b}.$$

The computation of an impulse in order to correct the velocities is not absolutely necessary for the simulation of a distance constraint. A higher degree of accuracy can be achieved by regarding the velocity constraint in the simulation but at the cost of performance.

2.3 Mesh simulation

In the simulation cloth is represented by a mesh. In each vertex of this mesh one particle is created. A simple approach to simulate stretchable cloth is to introduce a damped spring on each edge of the mesh. The elasticity of the cloth is controlled by the spring constant and the damping coefficient. However the simulation of inextensible cloth using spring forces leads to stiff differential equations which cause a significant decrease of performance.

In order to simulate cloth using impulses a distance constraint as described above is defined for each pair of particles that are connected by an edge in the mesh. The constraint is satisfied, if the distance between the positions of the two corresponding particles equals the length of the edge at the beginning of the simulation. If each constraint is exactly satisfied, the simulated mesh is inextensible. In such a mesh a particle is linked by multiple distance constraints to other particles. The impulses of constraints which have a common particle influence each other. The resulting dependencies can be resolved, if the constraints are handled in an iterative process. In an iteration step an impulse is computed for each distance constraint as described in section 2.2. Since the impulses influence each other the constraints are not satisfied within one step but the errors are reduced. The iterative process stops, when all constraints are satisfied within a predefined tolerance. Schmitt et al. (2005) proofed that this iterative method converges to the physically correct solution.

The iterative method has many advantages. It is very simple to implement. Collision and contact handling with friction can be easily integrated in the iterative process (Bender et al., 2006). The maximal allowed extensibility of the simulated cloth is directly defined by the used tolerance value. The iterative process can be interrupted at any time to get a preliminary result. Even if the process is interrupted and the tolerance value is not reached, the simulation stays stable. All constraints which have no common particle and therefore no direct dependency can be solved in parallel. The disadvantage of the iterative approach is that the simulation of a complex mesh requires many iterations, if a small tolerance value is used.

The number of iterations can be reduced significantly by subdividing the mesh into strips of constraints. The distance constraints of a strip must be acyclic. In this case each strip can be simulated by the linear-time method of Jan Bender (2007). The dependencies between the single strips are still resolved by an iterative process but since only a few strips are required for a mesh, now this process converges very fast.

3. RESULTS

The presented method was implemented in C++. For the accurate handling of collisions and contacts with friction the constraint-based collision response method of Bender et al. (2006) was integrated in the iterative process described above.

Figure 1 shows the results of two different simulations with the impulse-based method described in this paper. The tolerance value used in the iterative process defines the maximal extensibility of the simulated cloth. In both simulations the chosen value prevented the cloth from stretching more than 0.01 percent. The introduced method also works with much smaller tolerance values but at the cost of performance. By integrating the linear-time method of Jan Bender (2007) as mentioned above a mesh of 900 particles and 1740 distance constraints can be simulated about three times faster than real-time on a PC with a 2.4 GHz Intel Core 2 Quad processor.

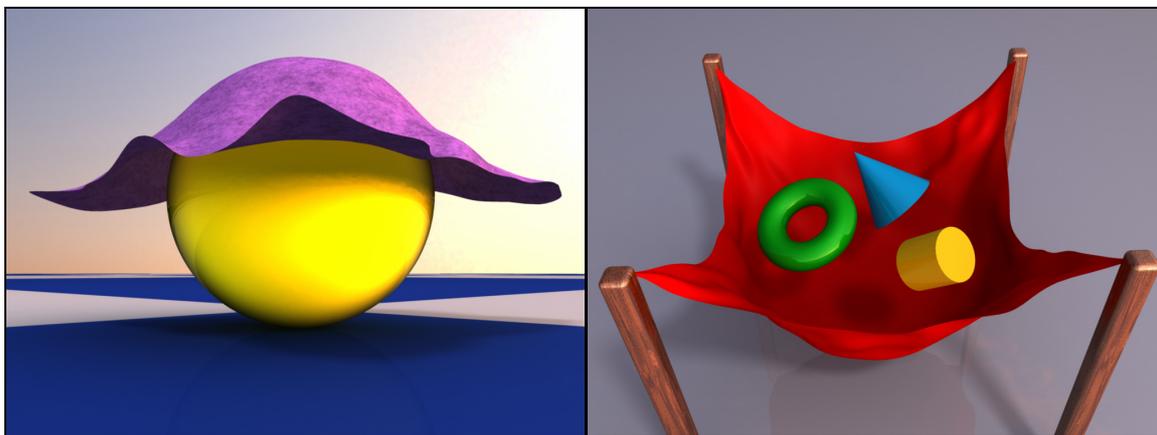


Figure 1. Results of the impulse-based cloth simulation

4. CONCLUSION

The presented method for cloth simulation allows the accurate handling of inextensible textiles. These textiles are simulated by using a mesh of particles linked by distance constraints which are satisfied by the computation of impulses. An impulse is determined by using a preview of the corresponding constraint state. The constraint dependencies in a mesh are resolved either iteratively or in combination with a linear-time method to increase the performance. The extensibility of the cloth can be controlled by a tolerance value. One of the main features of the new method is its stability. Since a constraint is directly resolved by using a preview, even totally destroyed models can be repaired. Hence in each step an early result can be obtained at any time by interrupting the iterative process without the loss of stability. This can be used to increase the performance. Other features are that the method is simple to implement and that collision and contact handling with friction can be easily integrated in the presented simulation process.

REFERENCES

- Bender, J. and Schmitt, A., 2006, Constraint-based collision and contact handling using impulses. *Proceedings of the 19th international conference on computer animation & social agents, Geneva, Switzerland*, pp. 3-12.
- Bender, J., 2007, Impulse-based dynamic simulation in linear time. *In Journal of Computer Animation and Virtual Worlds*, Vol. 18, No. 4-5, pp 225-233.
- Choi, K.-J. and Ko, H.-S, 2002. Stable but responsive cloth. *ACM Trans. on Graphics*, Vol. 21, No. 3, pp. 604–611.
- Choi, K.-J. and Ko, H.-S, 2005. Research problems in clothing simulation. *Computer-Aided Design*, Vol. 37, No. 6, pp. 585–592.
- Fuhrmann, A. et al., 2003. Interactive animation of cloth including self collision detection. *WSCG'03*, pp. 141–148.
- Goldenthal, Rony et al., 2007. Efficient Simulation of Inextensible Cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, Vol. 26, No. 3.
- Hauth, M. et al., 2003. Analysis of numerical methods for the simulation of deformable models. *The Visual Computer*, Vol. 19, No. 7-8, pp. 581–600.

- House, D.H. and Breen, D.E., 2000. *Cloth modeling and animation*. A.K. Peters, Ltd., Natick, MA, USA.
- Magenat-Thalmann, N. and Volino, P., 2005. From early draping to haute couture models: 20 years of research. The *Visual Computer*, Vol. 21, No. 8-10, pp. 506–519.
- Schmitt, A. et al., 2005, On the Convergence and Correctness of Impulse-Based Dynamic Simulation. *Technical Report 2005-17*. University of Karlsruhe, Germany.