Developing Simulation Techniques for an Interactive Clothing System

Pascal VOLINO and Nadia Magnenat THALMANN
MIRALab, C.U.I., University of Geneva
CH-1211, Switzerland
Email : [thalmann|pascal]@cui.unige.ch

Abstract

In this contribution towards creating interactive environments for garment design and simulation, we present a powerful mechanically based cloth simulation system. It is based on an optimized way to compute elastic forces between vertices of an irregular triangle mesh, which combines the precision of elasticity modelisation with the speed of a simple spring-mass particle system. Efficient numerical integration error management keeps computation speed efficient by allowing high computation timesteps and also maintains very good stability, suitable for interactive applications. Constraints, such as collisions or "elastics", are integrated in a unified way that preserves robustness and computation speed. We illustrate the potentialities of our new system through examples showing its efficiency and interactivity.

Keywords : *cloth animation, mechanical simulation, particle systems, collision response, constraints, stability, interaction, garment design tools*

1. Introduction

In a step towards unifying cloth simulation to the wonderful universe of Virtual Reality and dreaming about a world where virtual humans could manipulate cloth in real time and in a way that seems so natural for us, real humans, we present a contribution for a fast and robust cloth model suited for interactive virtual cloth design and simulation system.

Literature now brings us several techniques for cloth simulation. Many of them present physically based models for simulating in a realistic way fabric pieces based on elastic deformation and collision response. The first of them used simple mechanically-based models, such as relaxation schemes, for simulating objects such as flags or curtains ([WEI 86], [HAU 88]). More general elastic models were developed for simulating a wide range of deformable objects, including cloth ([TER 87], [TER 88]). Recently, several particle system based models attempted to simulate simple cloth object realistically using experimental fabric deformation data ([BRE 94], [EBE 96]). These models claim to be fast and flexible, as opposed to finite element models ([COL 91], [KAN 95], [EIS 96]), which are very accurate, but slow and complex to use in situations where behavior models are complicated and where collisions create non-linearities and complex boundary conditions, thus not suited for interactive applications.

Dressing a virtual body is a complex application for these models. It involves the ability to design complex garment shapes, as well as a complex simulation system able to detect and to handle multiple collisions generated between the cloth and the body. Our work contributed to the development and evolution of this topic through several contributions ([LAF 91], [THA 91], [YAN 91], [CAR 92], [YAN 93]). More recently, we studied how to consider cloth as being an object that could be considered independently from the body which wears it, involving the issues of folding, wrinkling and crumpling, with all the associated problems related to collision detection and response ([VOL 95]). Our work was materialized by several garment design and dressing systems for animated virtual actors ([WER 93], [THA 96]).

On the other hand, new V.R. technologies and efficient hardware open a very attractive perspective for developing interactive systems where virtual actors would interact autonomously with mechanically animated objects, such as the garment they themselves wear. In a nearer goal, we could take advantage of these new tools for interactively designing garments and dressing virtual actors in ways that are much more natural and close to the "real" way of manipulating fabric.

With this article, we provide simulation tools to take a step towards the requirements defined above. Of course, the main problems for interactive or real time mechanical simulation are related to computation speed issues. We should not however trade away design flexibility and mechanical modelisation accuracy that would lead to unrealistic cloth simulation. Thus, we describe here a mechanical model that allows to modelise elastic behavior of cloth surfaces discretized into irregular triangle meshes, and which is not much more complicated to a simple spring-mass modelisation. This approach combines the flexibility obtained in [VOL 95] with simulation speeds aimed in [EBE 96] and [HUT 96] which are restricted to regular meshes. Furthermore, a suited integration method has been associated to this model to maximize simulation timesteps and computation speeds without trading away mechanical stability, which is ensured in a very robust way, compatible with all the inaccuracies resulting from most tracking devices used in 3D positioning and V.R. devices.

Beside this, a new approach for handling geometrical and kinematical constraints (such as collision effects or "elastics"), generalization of the collision response process described in [VOL 95], ensures collision response as well as integration of different manipulation tools in a robust way that does not alter simulation efficiency and thus makes this system efficient for interactive purposes.

We illustrate the achievements brought by these techniques with the help of examples concerning cloth manipulation, dressing and realtime interactive manipulation.

2. A mechanical model suited for interaction

The mechanical simulation system is the core of a physically based cloth animation system, and also the most time consuming part. Our main contribution has been to improve the speed and robustness of the associated computation, first on the mechanical model itself (2.1), then on the integration algorithm and the associated numerical error managements (2.2), and finally on the constraint management, including collision response (2.3).

2.1. A fast and robust, yet realistic elastic model

The first problem of particle-system based models is to compute accurately the forces deriving from internal elasticity applied on each vertex.

2.1.1. What has been done until now

The simplest, and fastest method is to consider the surface as being a mesh of vertices, each one linked to its neighbors by a damped spring, forming a structure usually called "mass-spring" structure. Such models have already extensively been used for simple and fast cloth simulation ([PRO 95], [HUT 96]).

Most of these models rely on a regular grid. In [BRE 94] and [EBE 96], a square grid is used to compute tension, shear and bending. Internal forces are then computed using precise models resulting from experimental data. Mesh regularity is extensively used to keep geometrical properties easily computed in an accurate way. In [PRO 95] and [HUT 96], bending and shear effects are simply modeled by extra diagonal springs. However, our goals require non regular meshes as a basis for cloth structure. We need to be able to model complicated cloth shapes that may contain high curvature with as few elements as possible, and our interactive design process (cutting, seaming, local topology or size modifications) highly relies on a very general and multi-purpose triangular mesh structure.

In [VOL 95] and [THA 96], we have proposed a model derived from particle system models which computes the mechanical deformation state in each triangle elements of such an irregular mesh. By computing contributions from the edge elongations, compression and shear strains of the triangle material are found in local coordinates by solving a linear system, in a way similar to a stress rosette computation [TIM 82]. This model allowed a precise modelisation of an elastic material, taking into account the Young modulus and the Poisson coefficient, along with other parameters concerning viscosity and plasticity. However, this computation was quite expensive, as it required complex geometrical evaluations with the construction of local coordinates in each triangle element. At the opposite, the simplest spring and mass system, in which the forces applied on each vertex directly derives from the elongation of each edge connected to it, is very simple to compute, but would merely modelise more than a simple elastic material with a Poisson coefficient unrealistically high, unsuitable for any realistic cloth deformation.

Thus, our contribution is to define a new way to compute the forces applied on the vertices, comparably simple to the basic spring-mass system, but which allows precise modelisation of the Young modulus and the Poisson coefficient, which are the basic parameters of an elastic material.

2.1.2. The proposed elastic model

Let's consider a triangle $(\mathbf{P_a}, \mathbf{P_b}, \mathbf{P_c})$ in which deformations have elongated its edges from rest length $(\mathbf{L_a}, \mathbf{L_b}, \mathbf{L_c})$ to the current length $(\mathbf{l_a}, \mathbf{l_b}, \mathbf{l_c})$ (Fig. 1). In a simple elastic spring-mass model, each edge would attract its vertices to reach its rest length, and impose a displacement along its main direction, proportionally to the amount of elongation from the rest length. Please refer to the Annex for detailed formulas.

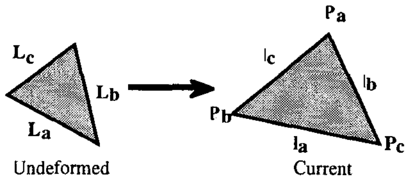


Fig. 1: Deforming a triangle element.

Quite easy to compute, this model does not however reflect the actual forces when a "full material" triangle gets deformed. Each deformed edge will produce a force component along its direction, which is usually not the deformation direction, as in the example shown in Fig. 2. The resulting effect is an extra orthogonal deformation similar to the one produced by the Poisson coefficient, but which produces unrealistic effects especially when the triangles are not equilateral (irregular meshes or high deformations).

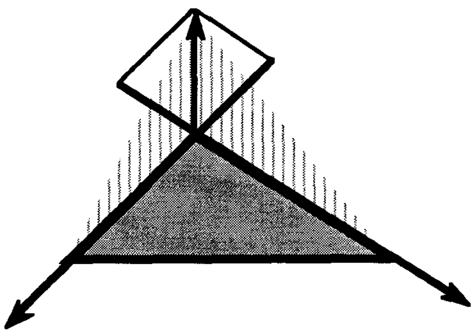


Fig. 2 : Vertical compression stretches the triangle horizontally.

The main idea of our new model is to recompute the individual elongation contribution of each edge of the triangle by taking into account the interdependence of the displacements that would be generated by each of them in their respective directions. Thus, the combined effect of the edge forces based on these corrected displacements will produce a more accurate constraint situation.

In the situation shown in Fig. 3, if we suppose that the length of the edge **J** varies an amount of $\mathbf{d_j}$, its extremity points $\mathbf{P_i}$ and $\mathbf{P_k}$ will be displaced in its direction by a amount proportional to $\mathbf{d_j}$, weighted in function of the values $\mathbf{M_i}$ and $\mathbf{M_k}$, the *inverse* mass of $\mathbf{P_i}$ and $\mathbf{P_k}$. The elongation contribution on edge **I** is then the displacement of the point $\mathbf{P_k}$ multiplied by the cosine of the angle between the two edges, $\mathbf{c_k}$. We linearize the problem by supposing that the edge angles do not vary significantly. **(i, j, k)** are all the permutations of **(a, b, c)**.

As we would like the final length variation of the edge **I** to be the value $\mathbf{l_i}$-$\mathbf{L_i}$, we equal this to the sum of the elongation contributions of the three edges **I**, **J** and **K** individually would elongate at an amount of $\mathbf{d_i}$, $\mathbf{d_j}$ and $\mathbf{d_k}$. Doing this on the three edges simultaneously yields a linear system of three equations with three unknowns ($\mathbf{d_a}$, $\mathbf{d_b}$, $\mathbf{d_c}$), shown in Annex B (4).

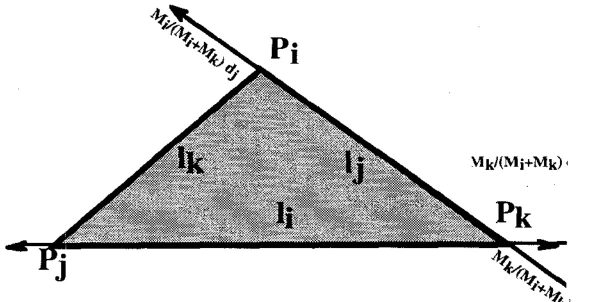


Fig. 3: Computing the elongation effect of the edge **J** on the direction of the edge **I**.

Solving the system leads to values of ($\mathbf{d_a}$, $\mathbf{d_b}$, $\mathbf{d_c}$), that, if applied altogether, would contribute to deform the triangle to its equilibrium state. After some variable substitutions, we obtain the force values shown in Annex B (7).

When working with almost regular meshes, we can make the approximation that all the vertices of the mesh have approximately the same mass. This yields the force values shown in Annex B (10). This assumption has proved experimentally not to alter much the behavior of the simulated elastic material in most usual situations dealing with cloth simulation.

The main positive aspect of this model is its simplicity, as it is yet able to compute realistic elastic forces in irregular triangle meshes. It involves very few vector operations, by directly computing force contributions along the edge direction without the need of any local coordinate system. Experimental tests have shown that, included in a resolution system such as the one discussed later, iterations are about twice as slow as the most basic spring-mass system (which was implemented for comparison), but it yet provides a realism similar to what was obtained in our previous work ([VOL 95]) which involved much more complicated geometrical evaluations.

2.1.3. Dealing with robustness and simulating the Poisson coefficient

A realistic elastic model should compute forces depending on the material deformations as similar as possible to the actual constraints that would be produced in a real material. However, it is impossible to guarantee that the imposed deformations are themselves realistic, depending on how they were previously computed (inaccurate modeling, inaccurate simulation, imprecise input devices,...). A robust model should deal with all situations whatever unrealistic they are, without leading to situations where the computed forces become infinite or still worsen the deformations.

In order to prevent any "infinite force" configuration, we introduce the α coefficient, which indeed represents a linear interpolation factor between a simple spring-mass system ($\alpha = 0$) and the model described above ($\alpha = 1$) (see Annex B (1)). Setting $|\alpha| < 1$ prevents any position configuration to lead to infinite forces. Furthermore, the forces remain continuous functions of the vertex positions, which increases the simulation quality.

With $\alpha = 1$, our model simulates a fabric material with a null Poisson coefficient. The simple spring-mass model simulates a material with an unrealistically high Poisson coefficient, which highly depends on the shape of the triangles. In addition to stability, the α coefficient allows us to keep control on the Poisson coefficient of the simulated fabric, as shown in Fig. 4, which would require more calculations if it had to be integrated explicitely in the force computation formulas.

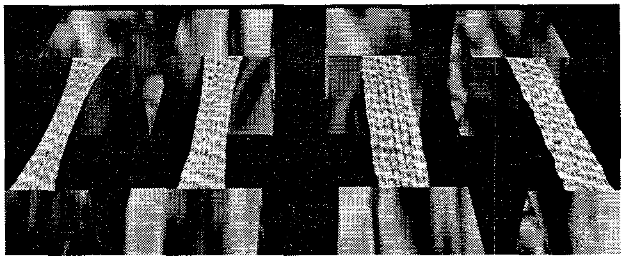


Fig. 4: A 400% stretched fabric square, with α values 0 (simple spring-mass system), 0.6, 1, 1.2 (transversal buckling).

2.2. Integrating motion equations

The dynamic system can be thought of as a huge ordinary second order differential equation system, where the variables are the positions for all vertices upon time. We may consider it as a first order system by taking as variables positions and speeds for all vertices. A numerical integration method is required for computing the evolution of the system with time.

In our previous work ([VOL 95], [THA 96]), we used an adapted midpoint method derived from the Euler formula that provided simple and quick calculation, while being adaptable enough to cope with highly nonlinear behaviors and direct handling of geometrical constraints, such as collisions. However, this method was not accurate enough, and could lead to numerical instabilities when dealing with important constraints and deformations. In order to increase numerical precision, we implemented a 4th order Runge-Kutta integration algorithm, adapted to the one described in [PRE 92]. The choice of this algorithm was motivated by several facts:

A) It only requires evaluation of the forces on the vertices depending on their positions and speeds at a given time, without any gradient evaluation, and the computation remains simple by implementing directly the formulas of part 2.1.

B) Current positions and speeds of all vertices are enough to describe the kinematical state of the system, as the integration procedure does not require any data from the previous system evolution. Position and speed may then be altered without disturbing the motion integration procedure.

C) With some adaptations, it offers an evaluation of the numerical error for the computed step, enabling efficient motion corrections and time step control.

A comparison of several integration methods has been presented in [EBE 96], showing the superiority of the Runge-Kutta method compared to more complex algorithms.

An adaptation of the Runge-Kutta method described in [PRE 92] yields the kinematical state of all the vertices of the simulated object along with an estimation of the error on this state.

This error may be used for adaptative timestep control, as done in [EBE 96]. However, for interactive applications where computation time is sensitive and where the timestep should be kept as high as possible, we would rather trade away precision for computation speed.

Whatever approximations we allow, we should ensure that the model keeps being stable and does not "explode" when becoming too imprecise. Our idea is to use the error estimations as a damping factor that would put the model in the lowest deformation energy state within the error range. Thus, we perform a kinematical correction on the position and velocity of the vertices accordingly, out of and after the dynamical computation process. This correction is performed on each edge by comparing and correcting the kinematical state of its extremity vertices.

A) The error on position values are compensated by modifying the vertex positions according to mechanical conservation laws to lead to the minimum deformation potential energy of the considered edge within the position error range.

B) The error on velocity values are compensated damping the vertex speed difference to lead with the minimum deformation kinetic energy of the considered edge within the velocity error range.

This correction technique ensures stability of the model even if mechanical damping parameters are very low, as mechanical deformation energy decrease is guaranteed by the corrections.

Obviously, if the error becomes too high and when it cannot be compensated by corrections, more standard timestep control schemes have to be implemented: The timestep is reduced and the dynamical process is recomputed if the error exceeds a given threshold, and the timestep is increased when the error becomes small.

As a result, we get a very robust simulation system that prevents deformation energy to increase because of numerical imprecision. Implemented using our mechanical model described in 2.1., which guarantees no singular situations where the forces become infinite, our model is almost impossible to break because of instability. This is particularly important when using inaccurate tracking devices for object manipulation in interactive or V.R. applications.

2.3. Handling dissipative forces, collisions and other constraints

In an approach introduced in [VOL 95], we handled geometrical constraints, such as those generated by collisions, using kinematical correction on the constrained elements: Positions and velocities were corrected according to the mechanical conservation laws to fit the constraints precisely. This approach permitted to skip the use of "potential walls" to enforce the constraints, which produced intense and discontinuous forces that could perturb the dynamical simulation. More recently, [EBE 96] used a similar way to handle friction effects.

Our new system generalizes this approach by giving a procedure for dealing with several types of constraints, such as collisions, but also seaming elastics described in 3., or some types of dissipative forces such as friction or high damping.

As mentioned earlier, the first motivation is to limit the value of the forces that are enforcing these constraints. These forces may be intense, thus participating to important energy transfer, and their discontinuity may lead to important simulation errors concerning their effects.

Constraint effects are split into three components:

A) An immediate correction on the position and velocities of the relevant vertices, taken into account before the dynamical simulation process, aimed to reflect the immediate effects of the constraint. Though less accurately simulated, damping effects integrated here are ensured to be perfectly dissipative, whatever their intensity, without altering the simulation.

B) A force correction in the dynamic simulation, that will attenuate or cancel the acceleration difference between the constrained vertices, in order to maintain the imposed kinematical constraints.

C) A force contribution in the dynamic simulation, when long-range constraints participate actively in the dynamics with continuous forces and durable effects. Along with force correction, it may be used for modeling an imposed acceleration constraint.

Our system implements several types of constraints:

Collision response is mainly a geometrical constraint imposing a minimum distance between two surface elements in contact. Position and velocity immediate correction (A) is performed in order to put the elements in an acceptable position and prevent their speeds to push them further together. Force correction (B) then enforces the maintaining of collision distances between iterations. Friction effects, which are usually intense, are simulated by velocity correction (A) and force correction (B) to simulate solid Coulombian friction. The collision detection aspect has been developed in our previous work [VOL 94] and [VOL 95].

Elastics are interaction tools introduced in [VOL 95] to permit seaming of cloth panels in a garment simulation system. An elastic attracts two vertices together and produces an attach point holding two elements together. In our previous work, they were simulated by adaptive forces that pulled the vertices together. However, their behavior were thus difficult to control and could lead to unpredictable results when interacting with collisions. Simulating them as kinematical constraints is the main improvement of our work. We use a combination of force correction and contribution (B) (C) to handle them as speed constraints producing a smooth acceleration toward the goal, parametrized by only one user defined time constant. They produce predictable results whatever the kind of objects they are attached to, and handling them along with collisions in an unique system makes them interact smoothly.

Damping is usually modeled by a force contribution (C) that opposes a speed difference. However, if damping is high (for example when dealing with non-linear models or solid friction), its effect will have time constants that may become much smaller than the simulation timestep, perturbing the simulation efficiency and accuracy.

Purely dissipative effects may then be simulated directly by a velocity damping (A) that is guaranteed to be dissipative whatever its intensity.

3. Results: Interactive tools for cloth design

Through the following examples, we illustrate the advances brought by our new system and the new potentialities concerning interactive cloth applications.

3.1. Interactive fabric manipulation

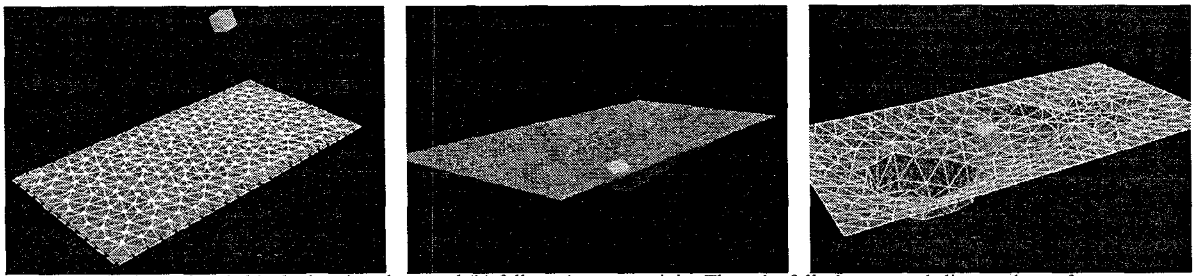
With our new model, interactive cloth manipulation is now possible, as shown by the following example. Here, we illustrate basic manipulations that are performed on a cloth element discretized to about 400 triangles. The display frame rate varies from 8 to 20 frames per second on a 250MHz R4400 SGI Indigo II, which is quite a comfortable speed for motion perception. Most of the time, the numerical approximation correction scheme allows us to perform the simulation with only one iteration per frame. The following figures are a sample of the system's interactive possibilities, the user performing actions that affect immediately and continuously the objects.

3.2. An improved clothing system

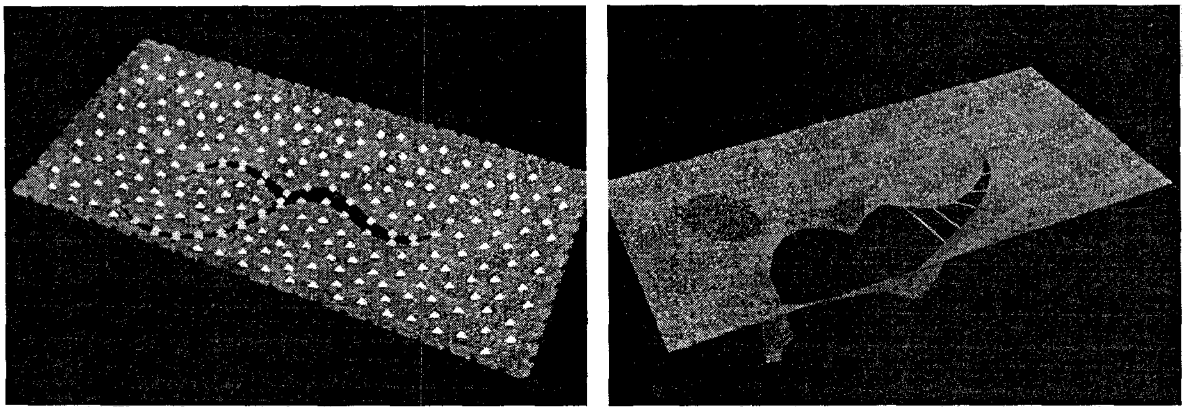
We take advantage of the new advances brought by algorithms in our software described in [VOL 95] and [THA 96], by highly improving the garment generation and simulation speed.

The following example illustrates how garments are assembled around the body and seamed together.

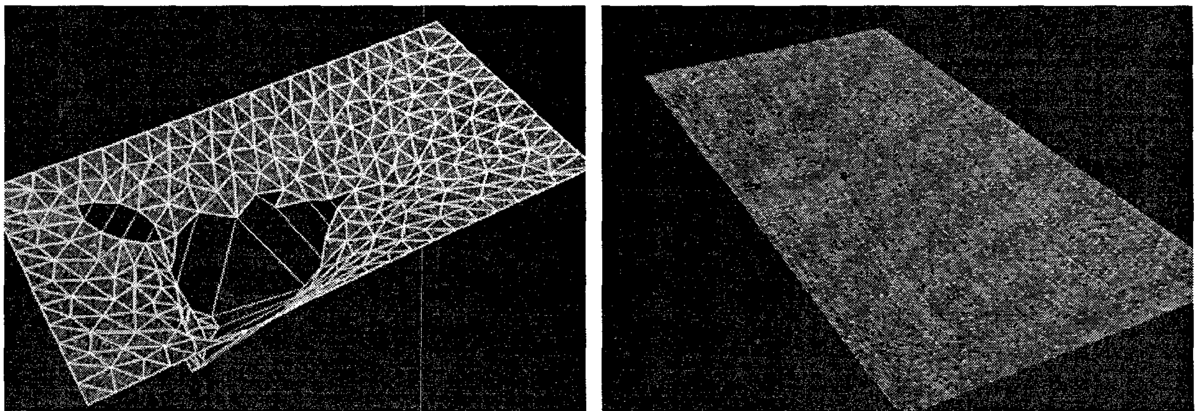
Fig. 5: Interactive cloth manipulation:



(a) The surface is blocked on its edges, and (b) falls on its own weight. The cube falls, bumps and slips on the surface.

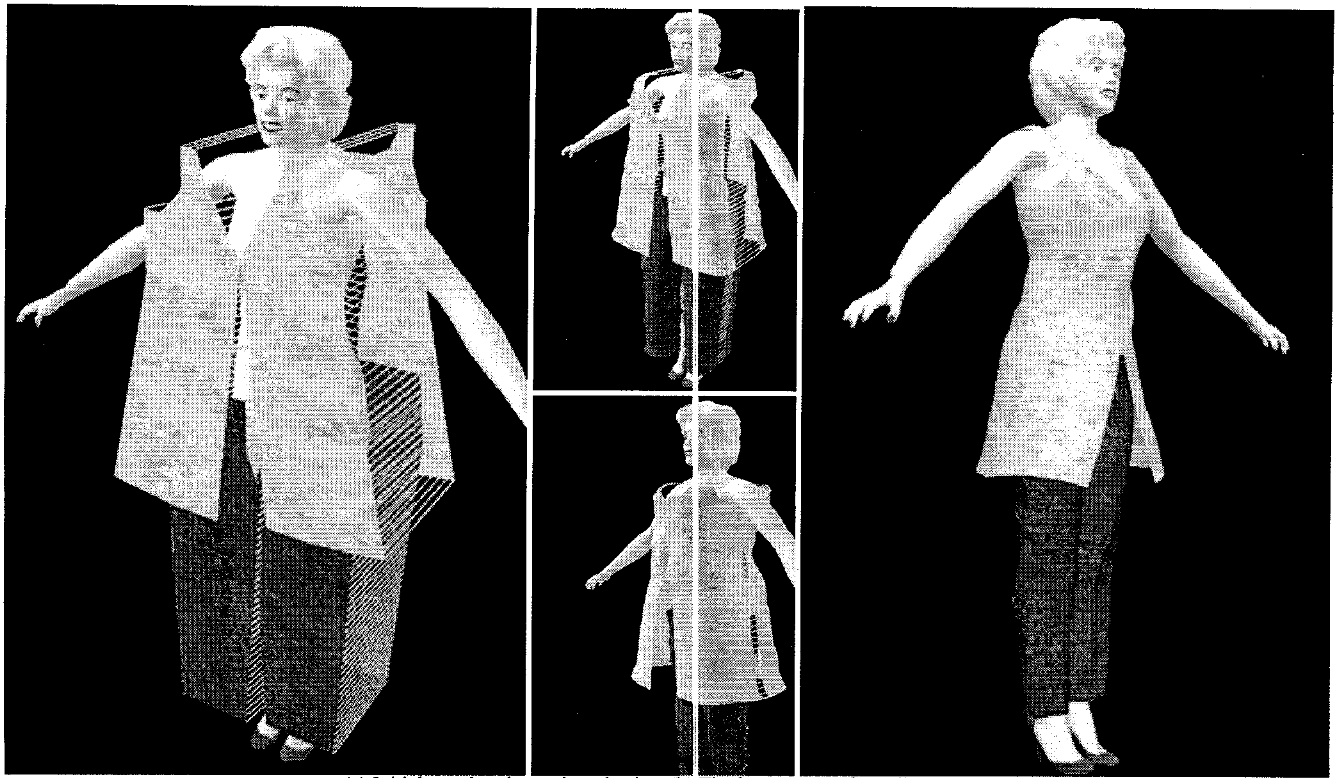


(c) The surface may be locally expanded or shrunk. (d) Cuts can then be performed. (e) Extra material can be removed.



Elastics pull the surface borders together (f) and finally (g) seaming fills the holes.

Fig. 6: Dressing a virtual actor:



(a) Initial panel and seaming elastics. (b) Final garments, after adjustments.
Intermediate steps: after 5 seconds computation and after 20 seconds, ready for panel seaming.

The main improvements from the results of our previous work is the computation time. The results above were obtained on a 150MHz R5000 SGI Indy, and take into account mechanical simulation computation as well as full collision detection and self-collision detection. As soon as the cloth begins to fit the body, collision detection becomes the major computation weight, which reached more than 70% in this example. Implementing incremental collision detection algorithms could reduce this.

3.3. Using tracking devices for cloth manipulation

In a first attempt to use our clothing system with Virtual Reality devices, we adapted a Flock-of-Bird type tracking system to move cloth object held by some of its vertices. Direct manipulation of the object is then possible in real time, with the six degrees of freedom. Several trackers may be used simultaneously to manipulate several objects or several parts of a same object, enabling for example two-hand manipulation.

On a 250MHz R4400 SGI Indigo II, the "feeling" of interactive manipulation remains very good with objects of less than 300 triangles for which the display rate is about 10 frames per second, and a good quality realtime simulation is reached with objects of less than 100 triangles, where the display rate exceeds 40 frames per second. These computation times include full collision handling.

Yet, the system accomodates very well to noisy tracking signals and remains stable even when the noise becomes too high for interaction use. The system is robust enough to cope with a few erroneous values which would send the object far away during a few frames.

We are now developing more natural manipulation tools using for example datagloves. A grasping system is being implemented, enabling two-hand cloth manipulation, and therefore many new and exiting new possibilities for an interactive clothing system.

4. Conclusion

Interactive cloth applications represent a wide subject on the crossroad of image synthesis, human animation, cloth design and Virtual Reality. We brought a contribution by providing through this work a fast and robust mechanical simulation system. Not really more complicated than a simple spring-mass particle system, it however simulates quite accurately an elastic material discretized into any irregular triangle mesh. Associated with Runge-Kutta integration and using numerical error evaluations as damping position and speed corrections, our model is robust, and yet performant by keeping computation timesteps high. A powerful constraint integration scheme also provides a powerful way of handling collisions, as well as a general support for extra interaction and manipulation tools.

Suited for complicated garment simulations as well as for interactive clothing tools, applications may be extended in any direction of the crossroad. We demonstrated the model's efficiency by some simple examples showing the speed, robustness, flexibility, and adaptability for tracking devices and Virtual Reality applications.

We would now like to take advantage of these potentialities to push further in the direction of interactive clothing applications. First, a powerful set of virtual tools would allow us to design, assemble and manipulate garments in a very natural way, enabling us to visually "feel" the fabric material in the 3D space. Then, using our VLNET system [CAP 97], we are preparing tools and techniques for a collaborative interactive system where distant partners together design and fit a common dress on a virtual being.

Acknowledgements

We are grateful to the Swiss National Research Foundation (FNRS) for funding this project, as well as Stéphane Carion who is developing the V.R. tools and interface.

Fig. 7: Cloth manipulation with Virtual Reality tools.

Bibliography

[BRE 92] : **D.E. Breen, D.H. House, P.H. Getto**, *"A Physical-Based Particle Model of Woven Cloth"*, Visual Computer, 8(5-6), pp 264-277, 1992.

[BRE 94] : **D.E. Breen, D.H. House, M.J. Wozny**, *"Predicting the Drap of Woven Cloth Using Interacting Particles"*, Computer Graphics (SIGGRAPH proceedings 1994), Orlando, USA, pp 365-272, July 1994.

[CAP 97] : **T.K. Capin, I.S. Pandzic, H. Noser, N. Magnenat-Thalmann , D. Thalmann**, *"Virtual Human Representation and Communication in VLNET Networked Virtual Environments"*, IEEE Computer Graphics and Applications, Special Issue on Multimedia Highways, 1997 (to appear).

[CAR 92] : **M. Carignan, Y. Yang, N. Magnenat-Thalmann, D. Thalmann**, *"Dressing Animated Synthetic Actors with Complex Deformable Clothes"*, Computer Graphics (SIGGRAPH proceedings 1992), 26(2), pp 99-104, 1992.

[COL 91] : **J.R. Collier et al**, *"Drape Prediction by Means, of Finite-Element Analysis"*, Journal of the Textile Institute, 82 (1), pp 96-107, 1991.

[EBE 96] : **B. Eberhardt, A. Weber, W. Strasser**, *"A Fast, Flexible, Particle-System Model for Cloth Draping"*, Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications), pp 52-59, Sept. 1996.

[EIS 96] : **J.W. Eischen, S. Deng, T.G. Clapp**, *"Finite-Element Modeling and Control of Flexible Fabric Parts"*, Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications), pp 71-80, Sept. 1996.

[HAU 88] : **D.R. Haumann, R.E. Parent**, *"The Behavioral Test-Bed: Obtaining Complex Bevavior with Simple Rules"*, The Visual Computer, Springer-Verlag, 4, pp 332-347, 1988.

[HUT 96] : **D. Hutchinson, M. Preston, T. Hewitt**, *"Adaptative Refinement for Mass-Spring Simulations"*, Eurographics Workshop on Animation and Simulation, Poitiers, France, pp 31-45, Aug. 1996.

[KAN 95] : **T.J. Kang, W.R. Yu,** *"Drape Simulation of Wowen Fabric Using the Finite-Element Method"*, Journal of the Textile Institute, 86 (4), pp 635-648, 1995.

[LAF 91] : **B. Lafleur, N. Magnenat-Thalmann, D. Thalmann**, *"Cloth Animation with Self-Collision Detection"*, IFIP conference on Modeling in Computer Graphics proceedings, Springer, pp 179-197, 1991.

[PRE 92] : **W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery**, "Numerical Recipes in C", Second edition, Cambridge University Press, 1992.

[PRO 95] : **X. Provot**, *"Deformation Constraints in a Mass-Spring Model to Describe Rigide Cloth Behaviour"*, Graphics Interface proceedings 1995, Quebec City, Canada, pp 147-154, May 1995.

[PRO 96] : **E. Promayon, P. Baconnier, C. Puech**, *"Physically-Based Deformations Constrained in Displacements and Volume"*, Computer Graphics Forum (Eurographics proceedings 1996), Poitiers, France, 15 (3), pp 155-164, Aug. 1996.

[TER 87] : **D. Terzopoulos, J.C. Platt, H. Barr**, *"Elastically Deformable Models"*, Computer Graphics (SIGGRAPH proceedings 1997), 21, pp 205-214, 1987.

[TER 88] : **D. Terzopoulos, K. Fleischer**, *"Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture"*, Computer Graphics (SIGGRAPH proceedings 1988), 22, pp 269-278, 1988.

[TIM 82] : **S.P. Timoshenko, J.N. Goodier**, "Theory of Elasticity", Third edition, McGraw-Hill Int. Editions.

[THA 91] : **N. Magnenat-Thalmann, D. Thalmann**, *"Complex Models for Visualizing Synthetic Actors"*, IEEE Computer Graphics and Applications, 11 (5), pp 32-44, 1991.

[THA 96] : **P. Volino, N. Magnenat-Thalmann, J. Shen, D. Thalmann**, *"An Evolving System for Simulating Clothes on Virtual Actors"*, Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications), pp 42-51, Sept. 1996.

[VOL 94] : **P. Volino, N. Magnenat-Thalmann**, *"Efficient Self-Collision Detection on Smoothly Discretised Surface Animation Using Geometrical Shape Regularity"*, Computer Graphics Forum (Eurographics proceedings 1994) Oslo, Norway, 13 (3), pp 155-166, Sept. 1994.

[VOL 95] : **P. Volino, M. Courchesne, N. Magnenat-Thalmann,** *"Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects"*, Computer Graphics (SIGGRAPH proceedings 1995), Los Angeles, USA, pp 137-144, Aug. 1995.

[WEI 86] : **J. Weil**, *"The Synthesis of Cloth Objects"*, Computer Graphics (SIGGRAPH precoeedings 1986) 24, pp 243-252, 1986.

[WER 93] : **H.M. Werner, N. Magnenat-Thalmann, D. Thalmann,** *"User Interface for Fashion Design"*, IFIP trans. on Graphic Design and Visualisation, pp 197-204, 1993.

[YAN 91] : **Y. Yang, N.Magnenat-Thalmann**, *"Techniques for Cloth Animation"*, New trends in Animation and Visualisation, John Wiley & Sons Ltd, pp 243-256, 1991.

[YAN 93] : **Y. Yang, N.Magnenat-Thalmann**, *"An Improved Algorithm for Collision Detection in Cloth Animation with Human Body"*, Computer Graphics and Applications (Pacific Graphics proceedings 1993), 1, pp 237-251, 1993.

Annex: Detailed formulas for the proposed model

Considering a triangle $(\mathbf{P_a}, \mathbf{P_b}, \mathbf{P_c})$ in which in-plane deformations have elongated its edges from rest length $(\mathbf{L_a}, \mathbf{L_b}, \mathbf{L_c})$ to the current length $(\mathbf{l_a}, \mathbf{l_b}, \mathbf{l_c})$:

$$l_i = \left\| \overrightarrow{P_j P_k} \right\| \qquad \begin{array}{c} i \\ j \\ k \end{array} = \begin{pmatrix} a & b & c \\ b, & c, & a \\ c & a & b \end{pmatrix} \tag{0}$$

The current angle cosine $(\mathbf{c_a}, \mathbf{c_b}, \mathbf{c_c})$ is also computed (taking $\boldsymbol{\alpha = 1}$):

$$c_i = \alpha \frac{l_j^2 + l_k^2 - l_i^2}{2 l_j l_k} \tag{1}$$

A. Simple spring-mass model

Computing the elongation of the edge springs:

$$d_i = l_i - L_i \qquad i = (a, b, c) \tag{2}$$

Computing the spring forces from the elongation:

$$F_i = \frac{R_i}{L_i} d_i = \frac{R_i}{L_i} (l_i - L_i) \tag{3}$$

$\mathbf{R_i}$ is a strength factor linked to the surface elasticity as well as a "shape factor" of this edge in the triangle (quite often, the rest length of the associated height).

B. The proposed model

Computing the desired elongation of each edge as sum of displacement contributions generated by all edge spring forces, $\mathbf{M_i}$ being the *inverse* mass of the vertex $\mathbf{P_i}$:

$$d_i + c_k \frac{M_k}{M_k + M_i} d_j + c_j \frac{M_j}{M_j + M_i} d_k = l_i - L_i \qquad \begin{array}{c} i \\ j \\ k \end{array} = \begin{pmatrix} a & b & c \\ b, & c, & a \\ c & a & b \end{pmatrix} \tag{4}$$

Simplifying calculus by computing the following intermediate values:

$$u_i = c_i M_i \qquad N_i = M_j + M_k \tag{5}$$

The system (4) is then simplified to:

$$d_i + \frac{u_k}{N_j} d_j + \frac{u_j}{N_k} d_k = l_i - L_i \qquad \begin{array}{c} i \\ j \\ k \end{array} = \begin{pmatrix} a & b & c \\ b, & c, & a \\ c & a & b \end{pmatrix} \tag{6}$$

Computing the forces from displacement contributions by solving the system (6):

$$F_i = \frac{R_i}{L_i} d_i = \frac{R_i}{L_i} N_i \frac{(u_i^2 - N_j N_k)(l_i - L_i) - (u_i u_j - N_k u_k)(l_j - L_j) - (u_i u_k - N_j u_j)(l_k - L_k)}{u_i^2 N_i + u_j^2 N_j + u_k^2 N_k - 2 u_i u_j u_k - N_i N_j N_k} \tag{7}$$

Taking the assumption that all vertices have comparable mass:

$$M_i = M_j = M_k \tag{8}$$

The system (4) is simplified as follows:

$$d_i + \frac{1}{2} c_k d_j + \frac{1}{2} c_j d_k = l_i - L_i \qquad \begin{array}{c} i \\ j \\ k \end{array} = \begin{pmatrix} a & b & c \\ b, & c, & a \\ c & a & b \end{pmatrix} \tag{9}$$

Computing the forces by solving the system (9):

$$F_i = \frac{R_i}{L_i} d_i = \frac{R_i}{L_i} \frac{(c_i^2 - 4)(l_i - L_i) - (c_i c_j - 2 c_k)(l_j - L_j) - (c_i c_k - 2 c_j)(l_k - L_k)}{c_i^2 + c_j^2 + c_k^2 - c_i c_j c_k - 4} \tag{10}$$

The acceleration contribution from the triangle applied on the vertices are finally computed from (1), (7) or (10) by:

$$\overrightarrow{A_i} = M_i \left(\frac{F_j}{l_j} \overrightarrow{P_i P_k} + \frac{F_k}{l_k} \overrightarrow{P_i P_j} \right) \qquad \begin{array}{c} i \\ j \\ k \end{array} = \begin{pmatrix} a & b & c \\ b, & c, & a \\ c & a & b \end{pmatrix} \tag{11}$$