Avenues of Research in Dynamic Clothing

Nadia MAGNENAT-THALMANN, Pascal VOLINO, Frédéric CORDIER
MIRALab, C.U.I., University of Geneva - CH-1211, Switzerland
Web: http://www.miralab.unige.ch
Email: [thalmann|pascal|cordier]@miralab.unige.ch
Phone: -41 (22) 705 77 63 Fax: -41 (22) 705 77 80

Abstract

While mechanical cloth simulation systems are widely used for creating draped garments on virtual characters, the animation of virtual garments raise challenges on its own. The perception of garment beauty is first related to the dynamic accuracy of the cloth simulation model, which has to reproduce precisely the dissipative behaviors of the material, such as viscosity and plasticity, rendering the visual features of realistic cloth motion. The other important aspect is the need of adequate techniques for the real-time visualization of animated virtual garments, required for many real-time or interactive applications. This work present techniques yielding visible advances for addressing these two aspects.

Keywords: *Cloth simulation, Garment animation, Mechanical cloth modeling, numerical integration, real-time garment animation.*

1. Introduction

One of the most challenging areas in research is in the development of a robust methodology for simulating animated garments on virtual characters. In order to define a cloth simulation system that is able to simulate complex garments realistically while maintaining a reasonable computation time, advances are required both in the side of simulation of dynamic mechanical behavior of cloth as on techniques that speed up the computation sufficiently for real-time applications.

Recent applications focusing on the simulation of high-quality garments mostly focus on the static mechanical behavior of the cloth and how the garment drapes on the body. This behavior is defined by the elastic properties of the cloth material, which can be measured using standardized procedures (KES, FAST), and simulated using an adequate model. However, perception of the beauty of garment and its visual features mostly rely on how the cloth moves and follow the motion and gestures of the dressed character. Simulating accurately the garment animation requires to take into account very accurately the dissipative mechanical behavior of cloth in a model that remains efficient enough for the simulation of complex garments along a long animation sequence.

Real-time garment animation applications require to optimize the simulation some steps further, using ad-hoc simplifications that remove the computational bottlenecks of the traditional simulation schemes. Success is based on a smart simulation that adapts to the particular context of each region of the cloth, focusing the computation of relevant types of cloth motion with the final aim of preserving visual cloth motion quality with as little computation as possible.

1.1. State-of-the-Art

Along the evolution of cloth simulation techniques, focus was primarily aimed to address realism through the accurate reproduction of the mechanical features of fabric materials. The early models, developed a decade ago, had to accommodate very limited computational power and display device, and therefore were geometrical models that were only meant to reproduce the geometrical features of deforming cloth [WEI 86]. Then, real mechanical simulation took over, with accurate cloth models simulating the main mechanical properties of fabric. While some models, mostly intended for computer graphics, aimed to simulate complex garments used for dressing virtual characters [LAF 91] [CAR 92] [VOL 95] [VOL 98], other studies focused on the accurate reproduction of mechanical behavior [ANI 89] [BEH 61] [CLA 90] [KAW 75], using continuum mechanics methods [TER 88] [CHE 95], particle systems [BRE 92] [BRE 94] [VOL 97] [EBE 96], or finite elements [COL 91] [EIS 96] [GAN 95] [KAN 95]. At the meantime, specific integration techniques are developed for fast simulation [BAR 98] [EBE 00] [VOL 01]. The current techniques now fulfill the following points:

Simulation of cloth material: Accurate representation and simulation of fabric material, initially started using continuum mechanics models, and then using particle systems, and using finite element techniques, intend to simulate the basic elasticity features of the cloth material, which usually include the Kawabata parameters, with additionally some viscoelastic parameters. The anisotropic nature of fabric elasticity is usually taken into account, differentiating parameters along weft and warp directions. Simulations are carried out on simple cloth samples, usually rectangular, and also on draping experiments on simple shapes. These models use quite high discretization in order to obtain the desired accuracy, and the computation time they require is quite high.

Fig.1. Simulation of the motion of various cloth materials.

Proceedings of the Computer Animation 2002 (CA 2002)
1087-4844/02 $17.00 © 2002 IEEE

Simulation of garment drape and animation on virtual characters: In these applications, the cloth constitutes a complete garment fitting a virtual body. The early models used continuum mechanics and the more recent ones used particle systems along with powerful collision detection and response schemes for getting better performance enabling complex cloth shapes on animated characters, situations that are much more complicated to manage than simple draping problems. Recent achievements allow the duplication of complex garments encountered in fashion models.

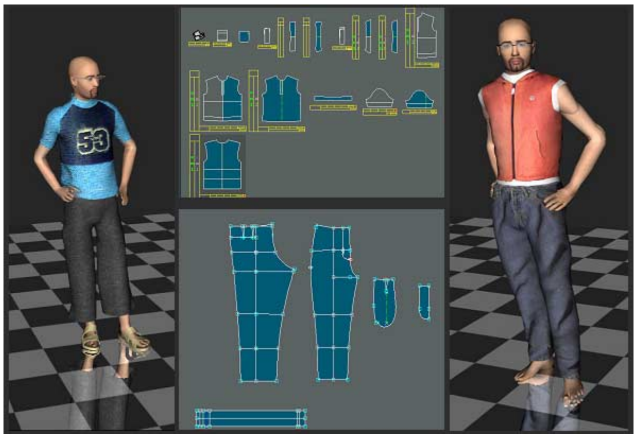


Fig.2. Patterns for virtual garment simulation.

While accurate mechanical models are nowadays able to reproduce the appearance of cloth quite realistically, they however require far too much computation for computing the animation of complex garments in real-time. This not only results from the complexity of the equations governing the mechanical model, but also from the complexity of collision detection when considering the general situation of complex garments in contact with the body, the garments surfaces being themselves in collision with each others.

Large advances have been made in the direction of improving the efficiency of the numerical techniques used for integrating the equations resulting from the mechanical model. The recent use of implicit integration schemes made possible the real-time simulation of small simple pieces of hanging cloth, but the implementation of the simplest mechanical models still do not allow real-time animation of garments. This objective may only be attained through context-dependent simplification of the simulation process through techniques related to numerical approximation, interpolation, geometrical modeling.

Among simple attempts to optimize computation speed using numerical simplification and algorithmic improvements, Meyer et al. [MEY 00] and Desbrun et al. [DES 99] have used a hybrid explicit/implicit integration algorithm to animate real-time clothes, integrated with this is a voxel-based collision detection algorithm. Their method seems to be limited by the maximum number of polygons they can animate in real-time.

Other research has focused on the collision detection, stating that it is one of the bottlenecks to real-time animation. Vassilev et al. [VAS 01] proposed to use the z-buffer for collision detection to generate depth and normal maps. Computation time of their collision detection does not depend on the complexity of the body. However, the maps need to be pre-computed before simulation, restricting the real-time application.

Another approach presented by Grzeszczuk et al. [GRZ 98] uses a neural network to animate dynamic objects. They replaced physics-based models by a large neural network that automatically learns to simulate similar motions by observing the models in action. Their method works in real-time. However, it has not been proven that this method can be used for complex simulation such as cloth.

James et al. [JAM 99] have also worked on real time simulation; their paper describes the boundary integral equation formulation of static linear elasticity as well as the related Boundary Element Method discretization technique. Their model is not dynamic, but rather a collection of static postures, limiting its potential applications.

Some other researchers have used geometrical approaches [WEI 86] [HIN 90] [HNG 95]. Geometrical models do not consider the physical properties of the cloth, therefore providing techniques that produce fast results. However, these techniques are not able to reproduce the dynamics of clothes. Moreover, geometrical techniques require a considerable degree of user intervention. They can be regarded as a form of advanced drawing tools.

Hybrid approaches try to combine geometrical and physical deformations. Kang et al. [KAN 01] improved the visual quality of the garments of small number of polygons by tessellating the triangles. With a wrinkled cubic spline curve, their tessellation algorithm is able to simulate the wrinkles. Oshita et al. [OSH 01] use a similar approach. Both these methods are mainly applicable to flat surfaces where physical simulation can be done with a very small number of polygons. However, highly curved surfaces, such as sleeves, need to be simulated with a higher number of polygons.

To our knowledge, no previously established technique exists for simulating fully dressed virtual humans in real-time using a physical simulation. The performance leap necessary to obtain the real-time simulation of complete garments cannot be obtained by further optimization of classic simulation techniques, despite the recent developments of simple models using particle systems, implicit integration and optimized collision detection. They require more drastic simplifications of the simulation process to be carried out, possibly at the expense of mechanical and geometrical accuracy. In this paper, we propose a method that is based on a hybrid approach where the cloth is segmented into various sections where different algorithms are applied.

2. Accurate Cloth Simulation

Accuracy is however the key to realistic garment animation, not only for the representation of mechanical elasticity which defines the shape of a draped cloth, but also for the dynamic properties that govern dissipation of mechanical energy, such as viscosity and plasticity, and which actually have an important visual effect on the cloth animation, such as oscillation damping and wrinkle propagation. This is particularly important when, beyond draping, the motion of the cloth has to be computed accurately.

The challenge is the design of a system that is able to combine accurate simulation of all mechanical properties of the fabric, both static elasticity and dynamic viscosity and damping, to a system which should be able to carry out the simulation of complex virtual garments worn by animated characters in a manageable amount of time.

2.1. Mechanical Properties of Cloth

The garment industry measures the property of actual fabric material through normalized procedures that identify characteristic measures. Among them, the FAST procedure identifies the major linear elastic components, such as weft, warp and shear Young modulus, west and warp bending modulus. The more detailed Kawabata Evaluation System (KES) [KAW 75] also includes additional parameters characterizing nonlinearity and hysteresis of the behavior curves, along with additional measurements such as fabric thickness.

Yet, all these parameters only characterize elastic behavior, where the internal forces on the fabric only depend on the deformation of the surface. While such a model is sufficient do compute accurately the static drape of a garment through any mechanical relaxation process that would find the equilibrium state of the mechanical system, no dissipative terms are actually present in the equations for reducing garment motion through viscosity effects. While it is actually possible to use such a model to produce visually realistic garment animations through the use of additional damping terms in the equations, there is no quantitative certitude on the validity of such terms and on their role for damping away the nasty residual oscillations that perturb the realism of garment animations.

The reproduction of real motion of cloth requires the implementation of an accurate mechanical model which is able to simulate accurately all the main parameters of fabric elasticity, as well as viscosity and plasticity, that can be efficiently integrated using the implicit methods traditionally used for particle systems. This model would be suited to accurate reproduction of the dynamic effects that brings realism to the simulation, with particular emphasis to the energy dissipation properties of viscosity and plasticity.

2.2. An Accurate Simulation Scheme

While most particle-system based systems restrict themselves to the simple, but inaccurate spring-mass schemes, the full evaluation of the strains and stresses of the plain surface is required for accurate reproduction of cloth viscoelasticity. This evaluation is ideally carried out on a discrete polygonal mesh describing the surface, generally formed by triangles. The presented model is suitable for the mechanical simulation of cloth viscoelasticity, using any triangle mesh for surface representation.

For a cloth mechanical model, we consider the weft and warp orientations as the main referential for the computation of strains and stresses. For this, we build a local "fabric" two-dimensional surface coordinate system that is normalized to the equilibrium deformation of the surface: The two unit vectors **U** and **V** describing this coordinate system are aligned to the weft and warp directions, which initially are orthogonal, and their length on the fabric at rest deformation is unity (fig.3).

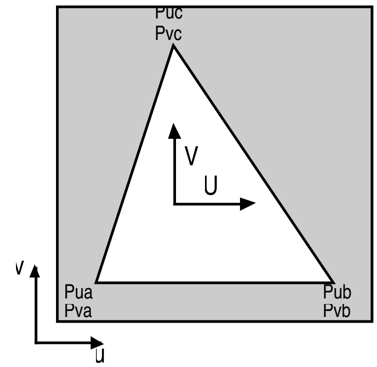


Fig.3. Definition of fabric coordinates.

Each triangle is described by its three vertices which have their own and constant 2D (**u**,**v**) fabric coordinates, describing the position of the triangle on the fabric. The separate 3D (**x**,**y**,**z**) geometrical coordinates describe the current position of the triangle in space. Using the values (**Pua**,**Pva**), (**Pub**,**Pvb**), (**Puc**,**Pvc**) being the (**u**,**v**) coordinates of the triangle vertices **A**, **B**, **C** respectively (fig.2), we compute as preprocessing the following values to be used in subsequent computations:

$$\begin{aligned} \mathbf{Rua} &= \mathbf{D}^{-1}(\mathbf{Pvb}-\mathbf{Pvc}) & \mathbf{Rva} &= -\mathbf{D}^{-1}(\mathbf{Pub}-\mathbf{Puc}) \\ \mathbf{Rub} &= -\mathbf{D}^{-1}(\mathbf{Pva}-\mathbf{Pvc}) & \mathbf{Rvb} &= \mathbf{D}^{-1}(\mathbf{Pua}-\mathbf{Puc}) \\ \mathbf{Ruc} &= \mathbf{D}^{-1}(\mathbf{Pva}-\mathbf{Pvb}) & \mathbf{Rvc} &= -\mathbf{D}^{-1}(\mathbf{Pua}-\mathbf{Pub}) \end{aligned} \tag{1}$$

where:

$$\mathbf{D} = \mathbf{Pua}(\mathbf{Pvb}-\mathbf{Pvc}) - \mathbf{Pub}(\mathbf{Pva}-\mathbf{Pvc}) + \mathbf{Puc}(\mathbf{Pva}-\mathbf{Pvb}) \tag{2}$$

The deformation state of the triangle is computed for given geometrical position of the triangle vertices using simple geometry. The starting point for that is the current value of the **U** and **V** referential vectors in the deformed triangle, computed from the current position of the triangle vertices **A**, **B**, **C** as follows:

$$\begin{aligned} \mathbf{U} &= \mathbf{Rua}\,\mathbf{A} + \mathbf{Rub}\,\mathbf{B} + \mathbf{Ruc}\,\mathbf{C} \\ \mathbf{V} &= \mathbf{Rva}\,\mathbf{A} + \mathbf{Rvb}\,\mathbf{B} + \mathbf{Rvc}\,\mathbf{C} \end{aligned} \tag{3a}$$

Using the hypothesis that the **U** and **V** vectors are orthonormal in the triangle in its undeformed state, we consider for measuring in-plane deformation the following values (fig.4):

* Elongation stress along the **u** (weft) direction: **Euu = |U| - 1**.
* Elongation stress along the **v** (warp) direction: **Evv = |V| - 1**.
* Shearing stress between the **u** and **v** directions: **Euv = (U.V) / (|U|.|V|)**.

In a similar manner, we can compute the deformation speed, to be used for instance in viscosity force evaluations. Starting from the speed of the triangle vertices **A'**, **B'**, **C'**, we compute the evolution of the **U** and **V** referential vectors as follows:

$$\begin{aligned} \mathbf{U}' &= \mathbf{Rua}\,\mathbf{A}' + \mathbf{Rub}\,\mathbf{B}' + \mathbf{Ruc}\,\mathbf{C}' \\ \mathbf{V}' &= \mathbf{Rva}\,\mathbf{A}' + \mathbf{Rvb}\,\mathbf{B}' + \mathbf{Rvc}\,\mathbf{C}' \end{aligned} \tag{3b}$$

Proceedings of the Computer Animation 2002 (CA 2002)
1087-4844/02 $17.00 © 2002 IEEE

The in-plane deformation speeds are then computed:

* Elongation evolution along the **u** (weft) direction: **Euu' = (U.U') / |U|**.
* Elongation evolution along the **v** (warp) direction: **Evv' = (V.V') / |V|**.
* Shearing evolution between the **u** and **v** directions: **Euv' = (U.V' + V.U') / (|U|.|V|)**.

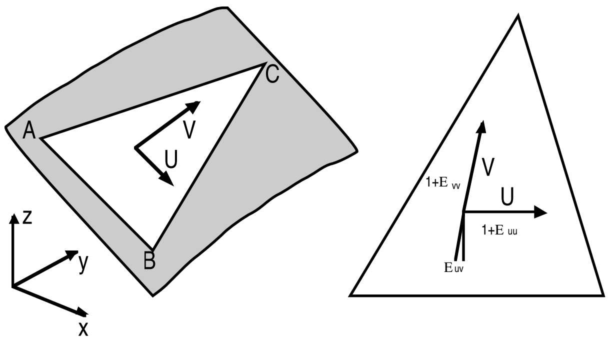


Fig.4. Deformation in world coordinates, and measurement of deformations.

A mechanical model of the cloth computes the strains **Suu**, **Svv**, **Suv** out of the corresponding stresses **Euu**, **Evv**, **Euv** (elasticity) and evolutions **Euu'**, **Evv'**, **Euv'** (viscosity). Each of these strains contribute to equivalent forces applied on the triangle vertices, each vertex force being equal to the strain integral applied on the corresponding opposite edge. Naming **U*** and **V*** the normalized vectors **U** and **V**, the are expressed as follows (here written for vertex **A**) (fig.5):

$$\mathbf{Fa} = -0.5\,\mathbf{D}\left(\mathbf{U}^{*}\left(\mathbf{Suu\,Rua} + \mathbf{Suv\,Rva}\right) + \mathbf{V}^{*}\left(\mathbf{Suv\,Rua} + \mathbf{Svv\,Rva}\right)\right) \quad (4)$$

This can also be established by expressing the deformation energy of the triangle and differentiating against displacement of the vertices along weft and warp directions.

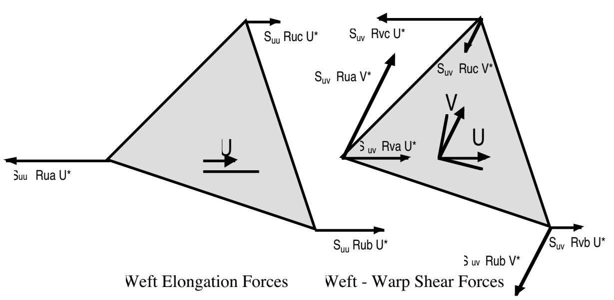
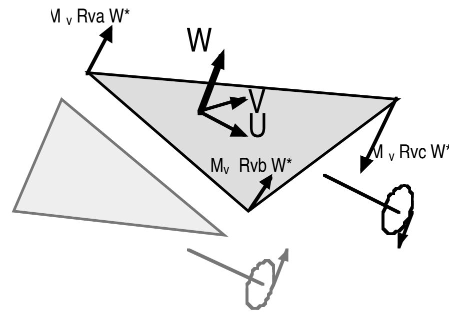


Fig.5. Computing vertex forces for elongation and shear.

In this first-order model, elements are flat, hence curvature forces can only be represented as interactions between several elements, which in practice will be implemented as rotation momentum interactions between element couples. For each element, the normal cross-product vector **W = U^V** is computed, and normalized to a vector **W***. Assuming small surface metric deformations (**U** and **V** near orthonormal), rotation momentums of amplitude **Mu** and **Mv** around the **U** and **V** vectors respectively can be obtained by exerting a force along **W** on each vertex (here written for vertex **A**) (Fig.6):

$$\mathbf{Fa} = \left(\mathbf{Mv\,Rua} - \mathbf{Mu\,Rva}\right)\mathbf{W}^{*} \quad (5)$$



Warp Bending Momentum Forces

Fig.6. Computing vertex forces for curvature.

Curvature effects can be simulated by creating opposite momentum effects between edge-adjacent elements in the mesh, related to a curvature amplitude and speed measurements derived from the angle value and evolution around this edge, and then distributed along weft and warp curvatures **Cu** and **Cv** according to the orientation of the edge in the (u,v) coordinate system.

The described model can be implemented for nonlinear behavior laws using nonlinear formulations of the relations between strain and stress. Plasticity may also be implemented by adding an hysteretic deformation "memory" altering equilibrium state for each deformation mode considered (Fig.7).

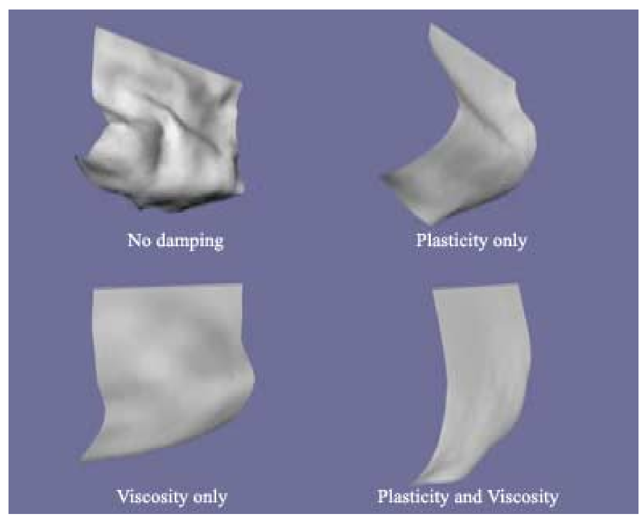


Fig.7. Testing viscosity and plasticity effects on cloth animation.

It is also easy to differentiate the force expressions on each vertex detailed above against the position and speed of the vertices of the triangle, in order to take advantage of the implicit numerical resolution schemes as discussed in the next section. Such formulation is indeed equivalent to the formulation of a first-order finite-element system that represents accurately all the features of metric elasticity, curvature elasticity being represented by additional inter-element interactions.

Fig.8. Accurate garment simulation with plasticity effects.

2.3. Efficient Numerical Integration

Implicit integration models are algorithms for solving differential equation systems that are known to remain stable for stiff problems, which are problems for which the time constants related to local state artifacts are much smaller than the time constants of the global system evolution to be simulated. Using standard explicit integration schemes such as Euler, Midpoint, Runge-Kutta and their derivatives, the numerical errors accumulate locally and create numerical instability, unless the simulation time step is small enough for the local time constants. Implicit integration methods such as implicit Euler and Rosenbrook methods do not exhibit this behavior and allow large simulation time steps related to the global evolution to be simulated.

The numerical differential equation systems resulting form particle system cloth simulation systems are actually stiff problems, as the time constants associated to the local particle vibrations are much smaller than those of the global evolution of fabric deformation. This actually increases with the stiffness of the material to be simulated, as well the as smoothness of the discretization. [BAR 98] made a breakthrough in cloth simulation by actually implementing an implicit integration method (the implicit Euler step) in a particle system cloth simulation system. While it proposed a complex implementation to render most of the mechanical parameters resulting from continuum mechanics considerations, we propose a much simpler implementation, based on any form of mechanical interaction of a particle system mechanical representation, adapted for the fast simulation of interactive and real-time systems.

While the explicit Euler step extrapolates the system state (particle position and speed) of the system at the next timestep through direct extrapolation from the current step using the state derivative (particle acceleration), the implicit Euler step performs the extrapolation using an evaluation of the derivative at the final step. This evaluation is obtained through the knowledge of the partial derivatives of this derivative against the system state, and first-order extrapolation. By naming **P**, **P'**, **P"** respectively the position, speed and acceleration vectors of all the particles of the system, the implicit Euler step is formulated, for a timestep **dt**, as follows:

$$\begin{bmatrix}\mathbf{P(t+dt)}\\ \mathbf{P'(t+dt)}\end{bmatrix}-\begin{bmatrix}\mathbf{P(t)}\\ \mathbf{P'(t)}\end{bmatrix}=\begin{bmatrix}\mathbf{P'(t+dt)}\\ \mathbf{P''(t+dt)}\end{bmatrix}\mathbf{dt} \tag{6}$$

$$=\left(\begin{bmatrix}\mathbf{P'(t)}\\ \mathbf{P''(t)}\end{bmatrix}+\begin{bmatrix}\partial\mathbf{P'}/\partial\mathbf{P} & \partial\mathbf{P'}/\partial\mathbf{P'}\\ \partial\mathbf{P''}/\partial\mathbf{P} & \partial\mathbf{P''}/\partial\mathbf{P'}\end{bmatrix}\left(\begin{bmatrix}\mathbf{P(t+dt)}\\ \mathbf{P'(t+dt)}\end{bmatrix}-\begin{bmatrix}\mathbf{P(t)}\\ \mathbf{P'(t)}\end{bmatrix}\right)\right)\mathbf{dt}$$

The partial derivative sub-matrices **dP'/dP** and **dP'/dP'** have null and identity values respectively. The sub-matrices **dP"/dP** and **dP"/dP'** relate respectively to the elasticity and viscosity force variations, with respect to the positions and speeds of the particles. Turning this into an equation system and rearranging leads to:

$$\begin{aligned}\mathbf{P'(t+dt)} &= \mathbf{P'(t)} + \mathbf{H}^{-1}\,\mathbf{Y}\\ \mathbf{P(t+dt)} &= \mathbf{P(t)} + \mathbf{P'(t+dt)}\,\mathbf{dt}\end{aligned} \tag{7}$$

Where:

$$\begin{aligned}\mathbf{H} &= \mathbf{I} - \frac{\partial\mathbf{P''}}{\partial\mathbf{P'}}\,\mathbf{dt} - \frac{\partial\mathbf{P''}}{\partial\mathbf{P}}\,\mathbf{dt}^2\\ \mathbf{Y} &= \mathbf{P''(t)}\,\mathbf{dt} + \frac{\partial\mathbf{P''}}{\partial\mathbf{P}}\mathbf{P'(t)}\,\mathbf{dt}^2\end{aligned} \tag{8}$$

Naming **F** the force exerted on the particles and **L** the inertia matrix being a diagonal (thus symmetric) matrix containing the particle inverse mass $\mathbf{M_i^{-1}}$ at each diagonal location corresponding to the particle **i**, Newton's law is expressed as **P"(t) = LF(t)**. Hence, we have:

$$\mathbf{P'(t+dt)} = \mathbf{P'(t)} + \mathbf{L}^{\mathbf{T}}\,\mathbf{H}^{-1}\,\mathbf{Y} \tag{9}$$

With:

$$\begin{aligned}\mathbf{H} &= \mathbf{L}^{\mathbf{T}} - \mathbf{L}\,\frac{\partial\mathbf{F}}{\partial\mathbf{P'}}\,\mathbf{L}^{\mathbf{T}}\,\mathbf{dt} - \mathbf{L}\,\frac{\partial\mathbf{F}}{\partial\mathbf{P}}\,\mathbf{L}^{\mathbf{T}}\,\mathbf{dt}^2\\ \mathbf{Y} &= \mathbf{L}\,\mathbf{F(t)}\,\mathbf{dt} + \mathbf{L}\,\frac{\partial\mathbf{F}}{\partial\mathbf{P}}\,\mathbf{P'(t)}\,\mathbf{dt}^2\end{aligned} \tag{10}$$

The biggest difficulty in implementing implicit methods is the resolution of a sparse linear system, which usually implies the appropriate data structure for managing huge sparse matrices and operations on them. The conjugate gradient algorithm can however remove this difficulty in the case of particle systems involving sets of simple interactions between reduced number of particles.

In [VOL 00] is described an efficient implementation of the Conjugate Gradient method that suppress the necessity of storing the sparse matrix **H**, by computing "on the fly" the contribution of each interaction in the matrix. Using the mechanical model described in the previous section, the partial derivatives of the forces of the three vertices of a triangle are integrated in this formulation for elongation, shear, and Poisson coefficient successively. Partial derivatives for curvature momentum between two triangle elements are integrated in a similar manner as a interaction between six vertices.

This method may be used in any implicit integration scheme that takes advantage of the Conjugate Gradient method [BAR 98] [EBE 00]. In [VOL 01] is discussed the relative efficiency of these methods against other explicit integration methods, comparing them in terms of accuracy, computation efficiency and robustness both for draping problems and for dynamic simulation reproducing accurately the motion of cloth.

While efficiently managed using geometric constraints [VOL 97], collision response can be efficiently integrated in this model as an equivalent particular adaptive form of mechanical interaction that fits into the formulation of this implicit integration scheme [VOL 00]. This allows the efficient simulation of complex collision features required for the animation of multilayer cloth (Fig.9), folds and crumples currently found in complex garments.

Fig.9. Animated multilayer garments.

Implemented with simple spring-mass interactions, this method also allows the inaccurate, but very efficient and robust simulation of simple cloth object suitable for real-time garment simulation, as discussed in the next section.

3. Real-Time Garment Animation

When observing a garment worn on a moving character, we notice that the movement of the garment can be classified into several categories depending on how the garment is laid on and whether it sticks to, or flows on, the body surface. For instance, a tight pair of trousers will mainly follow the movement of the legs, whilst a skirt will float around the legs. The first part of the study is to identify all the possible categories:

* Garment regions that stick to the body with a constant offset. In this case, the cloth follows exactly the movement of the underlying skin surface.
* Garment regions that flow around the body. The movement of the cloth does not follow exactly the movement of the body. In case of a long skirt, the left side of the skirt can collide with the right legs.
* Garment regions that move within a certain distance to the body surface are placed in another category. The best examples are shirtsleeves. The assumption in this case is that the cloth surface always collides with the same skin surface and its movement is mainly perpendicular to the body surface.

These three categories are animated with three different cloth layers. The idea behind the proposed method is to avoid the heavy calculation of physical deformation and of collision detection wherever possible, i.e. where collision detection is not necessary. The main interest of our approach is to pre-process the target cloth and body model so that they are efficiently computable during runtime. The skin and the garment are divided into a set of segments and the associated simulation method is defined for each. For each layer, we propose solutions and explain why they have been chosen.

3.1 "Tight cloth"

Tight clothes keep constant distances with the underlying skin surface. The deformation of this layer follows the deformation of the underlying skin. We choose to use the skin deformation method to animate this layer. This method does not involve any collision detection or physical deformation. It has no impact on the calculation time. Therefore, it is necessary to construct the skeletal information of this cloth layer. This information is defined by mapping the attachment information of the underlying skin to these cloth vertices.

3.2 "Loose cloth"

For loose clothes, the relative movements of clothes to the skin remain relatively small, keeping a certain distance from the skin surface. To get an intuitive understanding of such cases, consider the movement of sleeve in relation with the arm: for a certain region of the garment, the collision area falls within a fixed region of the skin surface during simulation. With this in mind, the scope of the collision detection can be severely limited. A basic assumption made is that the movement of the garment largely depends on that of the underlying skin and yet it should not follow the skin surface rigidly. It is necessary to simulate the local displacement of the garment from the skin surface.

In this approach, we use the assumption that cylinders can approximate the limbs. Each vertex is included in a sphere that is attached to the skin.

Initially, the position of the sphere centres are calculated. There movements follow the underlying skin. During simulation, the movement of the vertices inside their sphere follows the equation of the rigid body motion. In case a vertex leaves its sphere, a kinematic correction is applied to the velocity and the position to put the vertex back into the spherical area. The Fig.10 shows a cross-section of a limb with a garment.

The size of the sphere is a function of the angle between the cloth acceleration vector and the normal to the skin surface. The cloth acceleration vector is the subtraction of the gravity to the acceleration of the associated skin surface. This function is defined in a way that the size of the sphere follows roughly the catenary shape, i.e. the shape of a hanging wire. It can be proven that if a heavy flexible cable is suspended between two points, then it takes the shape of a curve with the equation:

$$\mathbf{y} = \mathbf{c} + \mathbf{a}\cosh\left(\frac{\mathbf{x}}{\mathbf{a}}\right) \quad (11)$$

Fig. gives an example of the variation of the size of the sphere along the body surface.

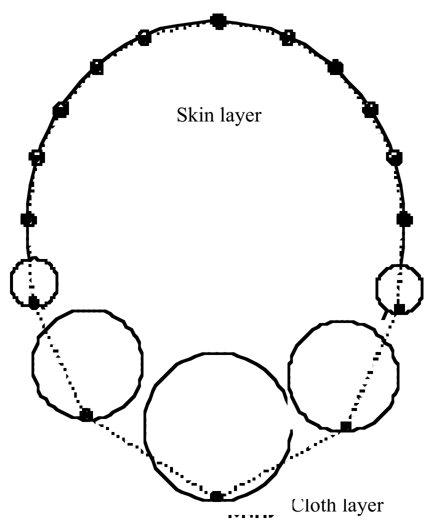


Fig.10. Cross section of a limb with a garment.

We approximate the limbs (arms, legs…) to cylinders and then in order to determine the size of the spheres for the cloth deformation we consider the equation of the circle (the cylinder that approximates the limb) and the equation of the catenary (the garment holt by the limb) as shown on the graph in Fig.11.

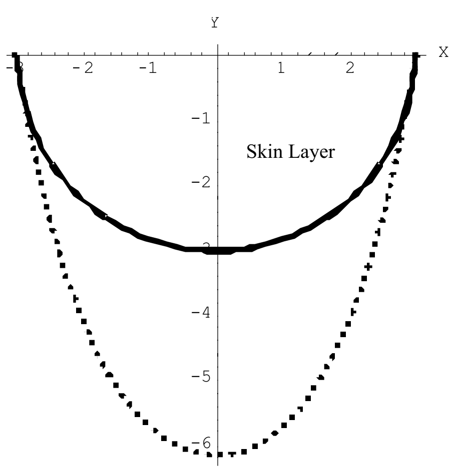


Fig.11: Equation of the catenary and the sphere.

The equation of the half sphere of diameter **2r** in Cartesian coordinate system (x, y) is:

$$\mathbf{y} = \pm\sqrt{\mathbf{r}^2 - \mathbf{x}^2} \quad (12)$$

The equation of the catenary in the same coordinate system Fig.:

$$\mathbf{y} = \mathbf{a}\left(\cosh\frac{\mathbf{x}}{\mathbf{a}} - \cosh\frac{\mathbf{r}}{\mathbf{a}}\right) \quad (13)$$

where *2r* is the diameter of the limb and *a* is the scaling factor of the catenary curve. It is used to control the size (largeness) of the garment. We define the ideal size of the sphere as the difference of the two equations above:

$$\mathbf{a}\left(\cosh\frac{\mathbf{x}}{\mathbf{a}} - \cosh\frac{\mathbf{r}}{\mathbf{a}}\right) - \sqrt{\mathbf{r}^2 - \mathbf{x}^2} \quad (14)$$

We define a quadratic equation to approximate the size of the sphere. This equation should be fast to compute, as it will be used intensively to calculate the movement of every vertex at every frame. This approximation is done by fitting the polynomial function $\mathbf{P(x)}$ to the three points $\mathbf{P_0(x=-r)}$, $\mathbf{P_1(x=0)}$ and $\mathbf{P_2(x=r)}$.

$$\mathbf{P}(\mathbf{x}) = \mathbf{C}\left(\left(\frac{\mathbf{x}}{\mathbf{r}}\right)^2 - \mathbf{1}\right) \quad (15)$$

Where **C** is a constant related to the size (largeness) of the cloth:

$$\mathbf{C} = \mathbf{a}\left(\mathbf{1} - \cosh\frac{\mathbf{r}}{\mathbf{a}}\right) + \mathbf{r} \quad (16)$$

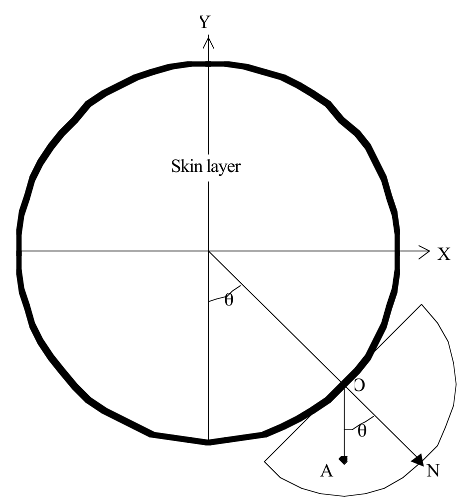


Fig.12. Computation of the sphere diameter.

The value $\mathbf{x/r = \sin\ \theta}$ is evaluated by projecting the acceleration component *OA* to the normal *ON* of the skin surface as shown in Fig.12. *OA* is the acceleration applied to the cloth. It is the subtraction of the acceleration of the skin vertex *O* to the gravity component. This method ensures that the rest shape of the garment takes the shape of real garment hanged on a limb.

In this method, no collision detection is necessary. Our particle system is very simple, each vertex moves independently, therefore no integration method is required. The resulting motion of the garments provides the appearance of dynamic movements as well as the shape of real clothes.

3.3 "Floating cloth"

Layer 3 is composed of vertices that freely float around the body. This will take care of cases, such as a large skirt floating around the legs. Any part on this skirt can collide with any part of the leg. The simulation of this layer uses a classical approach with particle system and collision avoidance.

* **Particle system**: We use a simple mass-spring system. The simulation is performed using the Implicit Euler Integration described in section 2.3. The garment is modelled by a simple mass-spring system. We consider two interactions: gravity and forces applied by the springs joining the particles.
* **Collision response**: Calculating collision detection between the cloth and the skin mesh would not be feasible in real-time, therefore we use a simplified model of the body. Given the assumption that the floating clothes are mainly skirts, the collision detection is calculated for the legs only and two cylinders model each leg. By simply calculating the distances between the skirt and the four leg segments, the collision detection can be performed. It is possible to further optimize by restricting the number of collision distances that we compute for each segment. During the pre-processing stage, a list of possible colliding vertices is defined for each segment. This list is defined by calculating the distance between the vertices and the legs and the normal orientation for each vertex on the skirt. The collision response consists of applying a kinematical correction to the position and velocity of the colliding vertices.

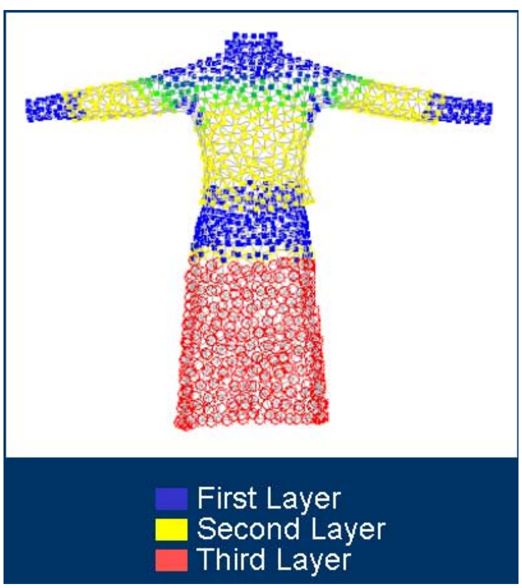


Fig.13. Multilayer realtime garments.

3.4. Performance

Our method has been tested on several dressed virtual humans. Fig.14 illustrates how computational time changes with an increasing number of cloth triangles. The code was executed on a 1GHz PC with 512 MB RAM and a GeForce2 graphics card. The computational times do not include the deformation of the avatar skin and the real-time rendering. A single iteration corresponds to 0.04 seconds worth of real-time animation.

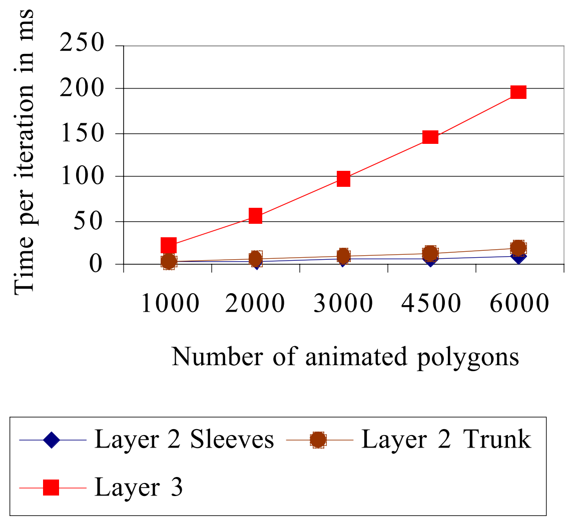


Fig.14. Computing time per iteration versus number of triangles.

As can be seen, the dependence is linear which indicates the low $O(n)$ complexity of the algorithm. The deformation of Layer 3 is in average thirteen times slower than the deformation of the Layer 2. This is because Layer 2 uses an optimized model with no collision detection and no integration method. The deformation of Layer 1 is even faster because it uses skeletal deformation. The Layer 3 algorithm is able to simulate a maximum of 1,000 polygons in real-time. In most cases, this is sufficient to produce aesthetically pleasing results. Another advantage is that most of the clothes can be animated using Layer 2. Layer 3 is used in only a limited number of cases, such as large skirts or large trousers. Fig.15 and Fig.16 demonstrate three examples of real-time animations. These simulations (rendering, cloth and skin deformations) ran at a speed of 25 to 30 frames per second.

Fig.15. A virtual human with a skirt and jacket.

Fig.16. Virtual human with a trouser and stretch pullover

Fig.17. summarizes the performances of these three animations.

Nb. of Polygons \ Model Type	Skirt and Jacket	Dress	Trouser and Pullover
Original model Cloth & Body	8413	5887	7445
After optimization Cloth & Body	4874	4526	5187
On Layer 1	542	847	2697
On Layer 2	1505	0	0
On Layer 3	618	885	593
Performance (fps)	29	26	25

Fig.17. Performance table.

4. Advances and Perspectives

Both accurate and real-time simulation contribute to important advances for animation of garments.

The presented accurate simulation model is powerful enough for simulating accurately dissipative parameters such as viscosity and plasticity. This important feature allows the model to reach animation realism that reflects the actual motion of cloth which is essential to the visual perception of the material and beauty feature of garments. Unlike most particle-system models claiming accurate simulation of mechanical properties, this system does not rely on regular grid-based meshes, but can deal with any triangular mesh resulting from complex garment design.

Real-time garment simulation mostly relies on geometrical techniques suppressing the need of computation-consuming collision detection and response techniques, and on the fact that not all the cloth vertices need to be animated with a complete physical method. The method is based on a smart partition the garments into layers reflecting different interaction contexts between the cloth and the body, each of them animated with the most appropriate method that fulfils the requirement of visual realism. Since the proposed method uses the deformation of the underlying skin, this method is applicable only on top of deformable objects such as virtual human.

Creating a single system offering both accurate simulation and real-time animation opens the field of new interactive virtual garment design and simulation applications, that both offer the accuracy needed by the garment industry and the interactive features required for virtual prototyping and visualization, possibly using virtual characters animated in real-time using motion tracking techniques or by their own awareness of their environment, related to a wide range of new applications such as virtual try-on, web applications, games, autonomous virtual worlds, etc.

Acknowledgements

We would like to thank Marlène Arevalo and Christiane Luible for the design of the clothes.
This project is financed by The Swiss National Research Foundation (FNRS).

Bibliography

[AMI 89] J. Amirbayat, J.W.S. Hearle, "The Anatomy of Buckling of Textile Fabrics", Journal of the Textile Institute, 80, pp 51-69, 1989.

[BAR 98] D. Baraff, A. Witkin, "Large Steps in Cloth Simulation", Proc. SIGGRAPH'98, Addison-Wesley, 32, pp 106-117, 1998.

[BEH 61] B. Behre, "Mechanical Properties of Textile Fabrics", Textile Research Journal, 31(2), pp 87-122, 1961.

[BRE 92] D.E. Breen, D.H. House, P.H. Getto, "A Physically-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 Drape of Woven Cloth Using Interacting Particles", Proc. SIGGRAPH'94, Addison-Wesley, pp 365-372, July 1994.

[CAR 92] M. Carignan, Y. Yang, N. Magnenat-Thalmann, D. Thalmann, "Dressing Animated Synthetic Actors with Complex Deformable Clothes", Proc. SIGGRAPH'92, Addison-Wesley, 26(2), pp.99-104, 1992.

[CLA 90] T.G. Clapp, H. Peng, "Buckling of Woven Fabrics", Textile Research Journal, 60, pp 228-234 & 285-292, 1990.

[COL 91] J.R. Collier, B.J. Collier, G. O'Toole, S.M. Sargand, "Drape Prediction by means of Finite-Element Analysis", Journal of the Textile Institute, 82 (1), pp 96-107, 1991.

[DES 99] M. Desbrun, P. Schröder, and A.H. Barr, "Interactive Animation of Structured Deformable Objects", Graphics Interface'99 proceedings, pages 1–8, June 1999.

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

[EBE 00] B. Eberhardt, O. Etzmuss, M. Hauth, "Implicit-Explicit Schemes for Fast Animation with Particles Systems", Proceedings of the Eurographics workshop on Computer Animation and Simulation, pp 137-151, 2000.

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

[GAN 95] L. Gan et al, "A Study of Fabric Deformation using Non-Linear Finite Elements", Textile Research Journal, 65(11), pp 660-668, 1995.

[GOV 95] B. Chen, M. Govindaraj, "A Physically-Based Model for Fabric Drape using Flexible Shell Theory", Textile Research Journal, 65(6), pp 324-330, 1995.

[GRZ 98] R. Grzeszczuk, D. Terzopoulos, G. Hinton, "Neuroanimator: Fast neural network emulation and control of physics-based models", SIGGRAPH 98 Conference Proceedings, Annual Conference Series, pages 9–20. ACM SIGGRAPH, Addison Wesley, July 1998.

[HIN 90] B.K. Hinds and J. McCartney, Interactive Garment Design, Visual Computer, Springler-Verlag, Vol. 6, pp 53-61, 1990.

[HNG 95] H. Ng, R.L. Grimsdale, "GEOFF-A Geometrical Editor for Fold Formation", Lecture Notes in Computer Science Vol. 1024: Image Analysis Applications and Computer Graphic, R. Chin, et al., eds., Springer-Verlag, Berlin, pp 124-131, 1995.

[JAM 99] D. James and D. Pai, "Accurate real-time deformable objects", SIGGRAPH 99 Conference Proceedings, Annual Conference Series, pages 65–72. ACM SIGGRAPH, Addison Wesley, August 1999.

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

[KAN 01] Y.M. Kang, J.H. Choi, H.G. Cho, D.H. Lee, "An efficient animation of wrinkled cloth with approximate implicit integration", The Visual Computer Journal, Springer-Verlag, 2001.

[KAW 75] S. Kawabata, "The Standardization and Analysis of Hand Evaluation", Hand Eval. and Stand. Committee of the Textile Machinery Society of Japan, Osaka, 1975.

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

[MEY 00] M. Meyer, G. Debunne, M. Desbrun, A.H. Barr, "Interactive Animation of Cloth-like Objects in Virtual Reality", Journal of Visualization and Computer Animation 2000, John Wiley & Sons, 2000.

[OSH 01] M. Oshita, A. Makinouchi, Real-time Cloth Simulation with Sparse Particles and Curved Faces, Computer Animation, IEEE Press, 2001.

[TER 88] D. Terzopoulos, K. Fleischer, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture", Proc. SIGGRAPH'88, Addison-Wesley, 22, pp 269-278, 1988.

[VAS 01] T. Vassilev, B. Spanlang, "Fast Cloth Animation on Walking Avatars", Eurographics, Blackwell Publishers, September 2001.

[VOL 95] P. Volino, M. Courchesne, N. Magnenat-Thalmann, "Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects", Computer Graphics (SIGGRAPH'95 proceedings), Addison-Wesley, pp 137-144, 1995.

[VOL 97] P. Volino, N. Magnenat-Thalmann, "Developing Simulation Techniques for an Interactive Clothing System", Virtual Systems and Multimedia (VSMM'97 proceedings), Geneva, Switzerland, pp 109-118, 1997.

[VOL 98] P. Volino, N. Magnenat-Thalmann, "The State-of-Art in Virtual Clothing", ISCIS'98 proceedings, 1998.

[VOL 00] P. Volino, N. Magnenat-Thalmann, "Implementing Fast Cloth Simulation with Collision Response", Computer Graphics International 2000, pp 257-266, 2000.

[VOL 01] P. Volino, N. Magnenat-Thalmann, "Comparing Efficiency of Integration Methods for Cloth Simulation", Computer Graphics International 2001, pp 257-266, 2001.

[WEI 86] J. Weil, "The Synthesis of Cloth Objects", Proc. SIGGRAPH '86, Addison-Wesley, 24, pp.243-252, 1986.