Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects

Pascal VOLINO [(*)], **Martin COURCHESNE** [(**)] [(*)]
and **Nadia Magnenat THALMANN** [(*)] [(**)]

ABSTRACT

We are presenting techniques for simulating the motion and the deformation of cloth, fabrics or, more generally, deformable surfaces. Our main goal is to be able to simulate any kind of surface without imposing restrictions on shape or geometrical environment. In particular, we are considering difficult situations with respect to deformations and collisions, like wrinkled fabric falling on the ground. Thus, we have enhanced existing algorithms in order to cope with any possible situation. A mechanical model has been implemented to deal with any irregular triangular meshes, handle high deformations despite rough discretisation, and cope with complex interacting collisions. Thus, it should deal efficiently with situations where nonlinearities and discontinuities are really non marginal. Collision detection has also been improved to efficiently detect self-collisions, and also to correctly consider collision orientations despite the lack of surface orientation information from preset geometrical contexts, using consistency checking and correction. We illustrate these features through simulation examples.

Keywords: deformable surfaces, collision detection, collision response, mechanical simulation, animation.

1 - INTRODUCTION

Many efforts have already been made to represent the natural motion of deformable objects under relatively simple contact constraints with the environment and restricted deformation situations. But real life situations provide a wide range of complexities, such as crumpling when an object falls to the ground, high deformations, wrinkling and friction when for example a synthetic actor puts on a cloth. Previous work has emphasized the precision of the mechanical model and the realism of the deformations while restricting animation contexts. However, to create scenarios with wrinkle and crumple situations where the deformations are ruled by lots of bendings and collisions, we need an efficient model able to deal with high deformations despite rough triangulation, where friction and collision are handled in a robust way. In addition, the model should not restrict simulated objects to particular situations shapes, therefore allowing objects to be composed of both regular and irregular meshes.

Previously, collision detection was often handled in very simple ways, subject to geometrical optimizations made possible by the simplicity of the situation (for example, some parts of the garment colliding with some parts of the body). The response was directed by simple geometrical considerations (vertex-to-triangle repulsion and friction, inside-outside orientation). But if we consider very general wrinkling situations, collision detection becomes a very difficult and time consuming task, because the lack of geometrical context prevents the use of optimizations. It also becomes difficult maintaining orientation consistency between the detected collisions.

Our main goal is to develop a very versatile and robust mechanical model, that is specially designed to rapidly and easily simulate deformable surface motion in any situation. It will be associated with a powerful and general collision detection and handling system that does not require any predefined context for efficiently and accurately computing collisions and self-collisions.

Remaining as general as possible, the model should be directly applicable to any object such as complex garments formed by several panels stacked on several layers or, more generally, any other kind of surface not necessarily discretised into uniform triangulation.

2 - PREVIOUS WORK

Previous works on deformable object animation using physically based models have permitted animation of cloth-like objects in many kinds of situations. Weil [35] pioneered cloth animation using an approximated model based on relaxation of the surface. Haumann et al. produced animations with flags or leaves moving in the wind, or curtains blowing in a breeze [16]. Kunii and Gotoda used a hybrid model incorporating physical and geometrical techniques to model garment wrinkles [18]. Aono simulated wrinkle propagation on a handkerchief using an elastic

(*) **MIRAlab, C.U.I., University of Geneva**
24, Rue du Général Dufour 1211 Genève - Switzerland
Tel: (41)22 705 7763 Fax: (41)22 705 7780
E-mail: pascal@cui.unige.ch thalmann@uni2a.unige.ch

(**) **H.E.C., University of Montreal**
5255, Av. Décelles H3T-IU6 Montreal - Canada
E-mail: martin@athos.miralab.hec.ca

model [2]. Terzopoulos and Fleischer developed a general elastic model and applied it to a wide range of objects including cloth [28] [29]. Interaction of clothes with synthetic actors in motion [19] [10] [39] marked the beginning of a new era in cloth animation in more complex situations. However, there were still a number of restrictions on the simulation conditions on the geometrical structure and the mechanical situations, imposed by the simulation model or the collision detection.

Deformable objects may be represented by different geometrical models, Triangular grids are most common, but polynomial surfaces [37] [4] and particle systems [6] are also used for solutions to specific mechanical simulations. Yelling nice and accurate deformations, they constrain both the initial shape and the allowed deformations. Each model requires different techniques for modeling complex objects such as panels-and-seaming for cloth objects [39]. Furthermore, global mechanical models such as finite elements and finite difference are not suitable for situations involving constraints and nonlinearities as non-marginal situations all over the surfaces. These situations happen when modeling the highly nonlinear deformations required for wrinkles and crumples [11], and when there are numerous collisions and much friction.

For coping with these mentioned situations, we provide a model resulting directly from the integration of Newton's motion equation. It allows us to efficiently and accurately integrate the effects caused by nonlinearities and collisions, and provides a small and efficiently computed adaptive time step, the best way for handling numerous discontinuities. We also remove modeling constraints by allowing simulation of any kind of non regular triangular meshes.

Collision detection and response has been used mainly for stopping cloth from penetrating the body and, more marginally, for preventing self-collisions between different parts of the cloth. The first time-consuming problem was to extract the possible colliding elements from the whole set of elements composing the cloth and the body surfaces. Many techniques have been developed, based on different ideas and adapted for various surface representations. For example, mathematical algorithms have been developed for situations where the surfaces are represented by curved patches or parametrical surfaces, as described in [3], [4], [31], [13], [27]. In the case of representing surfaces by a huge set of flat polygons, techniques based on rasterisation [26] or the tracking of the closest distance on the convex hull [22], [20] have been developed. Unfortunately, these techniques are not well suited for efficient detection on deformable surface animations, as they require either expensive z-buffer rendering or constructing the convex hull of the objects at each frame.

Closer to meeting our requirements are algorithms based on voxelisation or hierarchical octree subdivisions, as described in [39]. They are quite simple and efficient, but require a heavy data structure update at each frame. Hierarchical groupings for bounding-box tests have also been developed [33], [34], which are very efficient for handling a huge number of surface elements, but not well suited for surface self-collision detection.

We propose an efficient algorithm for detecting collisions, especially self-collisions on animated discretised surfaces [32]. It takes advantage of adjacency between the elements of a hierachisation, built once during preprocessing, for selectively performing collision tests using a very simple surface curvature criteria.

Besides being detected, collisions have to be handled in an accurate way for collision response. With cloth animation, this problem was relatively simple as long as the only situation considered was having the cloth already and constantly worn by the body [39]. Thus, a simple vertex-to-triangle interference using some proximity criteria could be considered. Furthermore, as the geometrical situation was quite constant and collision existed mainly when the cloth penetrated the body and wrinkles penetrated each other, the considered surfaces being already "oriented". Simple geometrical considerations could determine that a vertex had crossed, or was at the correct side of a polygon.

Unfortunately, as we are now considering very general and non-restrictive situations where any surface can collide with any other, considering what and how elements are colliding becomes a nontrivial problem for obtaining correct collision response. Difficulties arise when considering the lack of preset orientation information among the surfaces. This leads us to consider techniques for tracking collision orientation, as well as techniques which cope with and correct any orientation inconsistencies that may arise.

3 - THE MECHANICAL MODEL

The main idea of our model is to integrate Newton's motion equation in a direct way to keep quickly evaluated time steps small and very frequent collision detection. More sophisticated and time consuming models based on global minimizations or Lagrangian dynamics formulations allowing higher time step would represent a waste of time. Thus, discontinuous responses such as collisions will be handled in an accurate way. Furthermore, this direct formulation allows us easy and precise inclusion of any nonlinear mechanical behavior. With such a model, we can also act directly on the position and speed of the elements, and thus avoid handling collisions through strong repulsion forces that perturb the simulation.

The animated deformable object is represented as a particle system by sets of vertices forming irregular triangles, thus allowing surfaces of any shape to be easily modeled and simulated.

3.1 - Description of the physical object

The object is considered to be isotropic and of constant thickness. Elastic properties of the object are mainly described by the standard parameters [23] that are:

- E the Young modulus
- ν the Poisson coefficient
- ρ the density
- T the thickness

Rough discretisation, however, alters the behavior of the surface. In particular, heterogeneous triangulations are "rigidifying" the whole surface, preventing easy buckling. These effects have to be corrected through tuning and adjustments of the mechanical parameters. In particular, textile easily buckles into double curvature, but buckle formation requires a change of area that increases with the size of the discretised elements [8]. To facilitate buckle formation on roughly discretised objects without loosing textile stretching stiffness, we use a variable Young modulus for reducing the stretching stiffness for compression and small extension.

3.2 - The motion equation

Using Newton's second law F=ma, the motion equation consists of a pair of coupled first-order differential equations for position and velocity. The system of equations is resolved using the second order (midpoint method) of the Euler-Cromer method [15].

The constraints implied in deformable object motion are divided in two categories:

* Continuous constraints including internal and some external ones such as wind and gravity.
* Discontinuous constraints resulting from collisions with other objects.

Discontinuous constraints induce instantaneous change in the state of the object. Considering the collision frequency, the interruption of the simulation every time a collision occurs would take much computation. Rather than considering complicated methods for solving differential equations with discontinuities [7] which may not be efficient for very complex collision situations, we prefer handling collisions separately.

The problem of solving the differential equations has been simplified considerably using a two phase process, similarly to House et al. [17]:

a - Considering only the continuous constraints, differential equations are solved using a time step that ensures mechanical stability for every vertices, computed from the acceleration of the vertices versus the length of their connected edges. However, we do not recompute acceleration at each step for vertices which do not require such a small time step.

b - Then, collisions are detected and discontinuous constraints are handled, through direct correction of position and speed complying momentum transfer laws.

3.3 - Internal strains

Internal strains are either in-plane, from planar extension and shearing, or out-of-plane, from bending and twisting. Considering the irregularity of the triangle mesh, the force evaluation should be independent from the size and shape of the triangles.

** Elastic and shearing strain*

A triangle is considered as a thin flat object in a plane stress situation. Each edge of the triangle is taken as a strain gauge giving strain measurement on the cloth surface (fig. 1). A set of three measurements, called "strain rosette" [30], is enough for completely evaluating the strain. The unit elongations given by each edge at an angle θ_i are related using: [25]

$$\varepsilon_{\theta_i} = \varepsilon_u \cos^2\theta_i + \varepsilon_v \sin^2\theta_i + 2\,\gamma_{uv} \sin\theta_i \cos\theta_i \qquad (1)$$

We compute the unit elongations (ε_u, ε_v) and shear (γ_{uv}) in an arbitrary, conveniently defined (u,v) coordinate system. Then, the Hook law for a uniform isotropic material [30] directly gives the stress components:

$$\sigma_{u,v} = \frac{E}{1-\nu^2}\left[\varepsilon_{u,v} + \nu\varepsilon_{v,u}\right] \qquad \tau_{uv} = G\,\gamma_{uv} = \frac{E}{2(1+\nu)}\gamma_{uv} \qquad (2)\ (3)$$

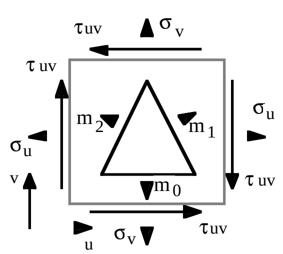

Fig. 1 : Stress evaluation in a triangle

The stress components on a triangle are convert into in-plane forces along its edges. The force applied on the edge j of the triangle i is:

$$\mathbf{F}_i^j = T\,L_j \left([\mathbf{m}_j\cdot\mathbf{u}\,\sigma_u - \mathbf{m}_j\cdot\mathbf{v}\,\tau_{uv}]\mathbf{u} + [\mathbf{m}_j\cdot\mathbf{v}\,\sigma_v + \mathbf{m}_j\cdot\mathbf{u}\,\tau_{uv}]\mathbf{v} \right) \qquad (4)$$

where $\mathbf{m}_j$ is a unit vector perpendicular to the edge j in the triangle plane. This force is then equally distributed on the two extremity vertices.

** Bending strain*

Curvature force are very weak compared to in-plane forces. As we intend to consider high deformation, the force evaluation must consider the possible case where the radius of curvature is less than the size of the triangles.

The edge between two triangles is used as a hinge for curvature manifestation (fig. 2a) providing information on single curvature only. It is known from the Mohr circle that it is always possible to decompose any twist strain into a combination of pure bending strains [8]. Considering the arbitrary orientation of the edges, even if we have no control on them, twist strain is taken into account via the additive property of curvature.

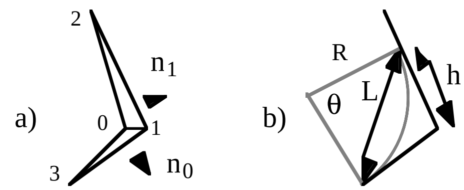

Fig. 2 : Curvature evaluation.

Using the angle between the normals (fig. 2a), we look for the maximum curvature radius (R) for which the corresponding arc fits inside the triangles. Referring to fig. 2b, h is less than or equal to the height of the triangles and L is greater than h*a, with a<1. This adaptation will allow R to reach values smaller than the size of the triangles. The local curvature (K) is the inverse of R.

To prevent K and the bending force from reaching infinity, we limit K to a maximal value. If the angle continues to grow, a specific high bending constraint handling will be performed.

The associated momentum in width units is:

$$M = D'DK = E\,D'\frac{T^3}{12(1-\nu^2)}K \qquad (5)$$

where D, the flexural rigidity, is associated with D' an the extra parameter which is needed to allow fine tuning of bending strain. Using D'=1 would be appropriated for continuous solid sheets but would not reflect the real comportment of textile [1]. The material is still isotropic. The force corresponding to M is obtained using the triangle dimensions and normal.

3.4 - Time effect

Textile material is not purely elastic and its response to stress depends on its straining history. Several phenomena, some of which are described in [23] and [21], yield time effects that appear as recovery behavior, creep, stress relaxation of stress, etc.

Integral and analytical theories aim to describe mathematically the macroscopic behavior of material. Since permanent or semi-permanent damage is closely related to the quality and straining history of fiber and textile's structure, an exact representation of them is impossible. Rather than trying to idealize the behavior with complex equations, we developed a simple empirical equation to model the consequences.

Basically, pure elastic behavior for a deformation "x(t)-x(0)" of an element at a time t is described by Hook's law:

$$F(t) = k\ (\ x(t) - x(0)\ ) \qquad (6)$$

which is added a linear viscoelastic response according to Newton's law:

$$F_v(t) = \gamma_x \frac{(\ x(t) - x(t\text{-}dt)\ )}{dt} \qquad (7)$$

where k and γ_x are elasticity and viscosity constants.

When deformation exceeds a given ratio, we switch from the viscoelastic behavior described above to a plastic behavior for which the equilibrium x(0) is moved to a new value, modeling permanent deformation. A relaxation time has also been defined, defining how fast the equilibrium evolves back to its original value, as soon as the deforming constraints are released.

3.5 - Collision management

We correct the non-constrained simulation by detecting and taking the collision effects into account.

Rather than computing "collision forces" through inverse kinematics from the momentum conservation law [10], we directly integrate the constraints by position and speed corrections on the concerned vertices accordingly to momentum conservation. Thus, we avoid dealing with high reaction forces that alter the mechanical simulation.

For instance, if a collision is detected between two elements, we compute the new positions of these elements that satisfy both the collision geometrical constraints and the mass center invariance. If some elements are implied into several collisions, we iterate the process until all the constraints are satisfied. Then, the speed of the elements is evaluated accordingly to momentum conservation, using perfectly inelastic collision for the normal speed along the collision plane, and coulombian friction for the tangential speed.

This technique ensures very fast collision response computation, each collision being handled independently and by avoiding high reaction forces. However, robustness is required for the collision detection algorithm, to maintain consistency in complex interdependent collisions situations where the collision response may not be able to solve completely all the constraints.

3.6 - Stability control

As we are dealing with nonlinear models put into widely varying conditions, some situations (for example, deformation caused by collisions) might lead to numerical instability. Once we detect increasing instability by monitoring local mechanical energy variations, we artificially distribute kinetic energy through momentum transfers in the neighborhood of the concerned elements. This transfer accelerates the propagation and the fairing of the perturbation. This technique increases the global robustness of the system for difficult conditions.

4 - COLLISION DETECTION AND HANDLING

For dealing with complex collision situations such as crumpling, we need efficient collision and self-collision detection, as well as a robust collision handling. We are now discussing these two aspects.

4.1 - Fast self-collision detection

Collision and particularly self-collision detection is often the bottleneck of simulation applications in terms of calculation time, because of the scene complexity that involves a huge number of geometrical tests for determining which elements are colliding.

In our case, the problem is complicated further because we are handling discretised surfaces that may contain thousands of polygons. We also are considering general situations where we cannot make any hypotheses about region proximities. Finally, we have to efficiently detect self-collisions within the surfaces. This prevents the use of standard bounding box algorithms because potentially colliding regions of a surface are always touching each other by adjacency.

We have developed a very efficient algorithm for handling this situation [32]. This algorithm is based on hierarchisation and takes advantage of the adjacency which, combined with a surface curvature criteria, let us skip large regular regions from the self-collision detection. We then get a collision evaluation time that is roughly proportional to the number of colliding elements, and independent of the total number of elements that compose our deforming surfaces.

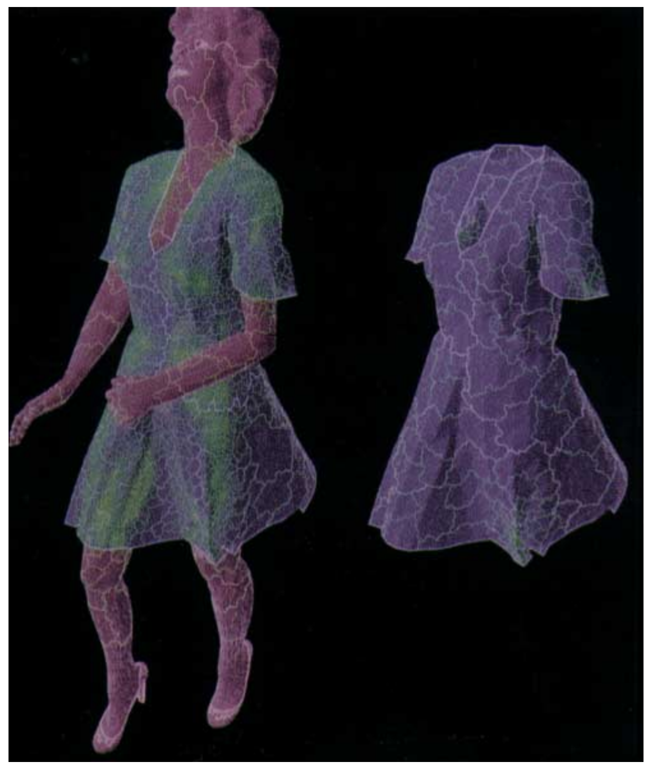

Fig. 3 : Hierarchical collision and self-collision detection on cloth. Less than 5% of the detection time is spent for self-collisions.

4.2 - Handling different kinds of collisions

Once the possible colliding triangles of our surfaces are located, we extract different types of geometrical collisions:

** Proximities*

They are represented by couples of elements that are closer than a threshold distance. That may be triangle-to-vertex, edge-to-edge, and more marginally edge-to-vertex and vertex-to-vertex proximities. They illustrate collision interaction. They are used for computing collision response.

** Interferences*

They are represented by edges-triangle couples that are crossing each other. They illustrate situations where two surfaces are interpenetrating. They reveal inconsistent collision situations that have to be corrected.

4.3 - Collision consistency

Collision response implies the correction of position and velocity to prevent contact and crossing. However, this problem cannot be efficiently resolved in complicated situations such as interaction between multiple collisions. It may occasionally happen that some vertices move to "the wrong side" of a colliding surface, a situation with which the collision response must cope.

Usually, the "right side" of a vertex from a triangle is determined by "inside-outside" orientation assumptions for the surfaces, made possible for some simple collision situations or geometrical contexts (for example, the vertices of the cloth have to be pushed outside the body, and colliding wrinkles have the same surface orientation). However, as we intend to simulate cloth or any other deformable surface in any situation, such orientation information is not available, and we cannot deduce from vertex-triangle proximity at which side of the triangle the vertex should be.

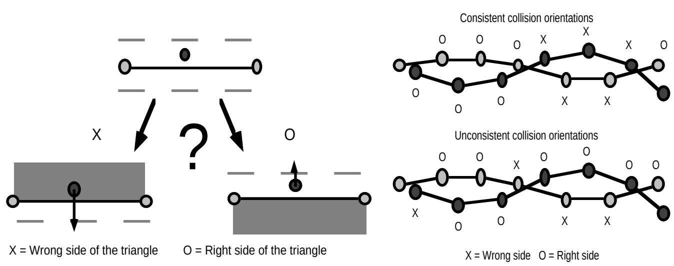

Fig. 4 : Orientation ambiguity and collision consistency.

Our contribution has been to create algorithms able to correctly orient the detected collisions so as to correct any wrong situation. We use a combination of techniques described below.

** Remnant proximities*

As we said, vertices may marginally cross the colliding surfaces, but the response must return them to the correct side, even if they are temporarily out of the scope of the collision detection.

For solving this problem, a proximity will be kept in memory for a certain time after its last detection, even when the concerned elements move far away from each other. During this time, the proximity will be geometrically updated with respect to the displacements of the objects. If this collision then gets detected again, its orientation is still known according to its "history".

** Cinematical tracking*

For each newly detected proximity, we compute the relative movement of the concerned elements, and we can know whether a crossing has happened just before detection or not. If not, the elements are still at right side of each other.

** Consistency checking and correction*

Even using the previously mentioned techniques, it is always possible that some collisions get incorrect orientation due to inaccurate response. This occurs mainly in complicated cases that lead to geometrically incorrect situations, and therefore erroneous detection.

Our algorithm should not be perturbed by such false situations, and should be able to correct the wrong collision orientations, whenever they happen.

Usually a contact region between two surfaces is represented by several collisions. The elements concerned by this group of collisions define "collision regions" on both surfaces. Our main

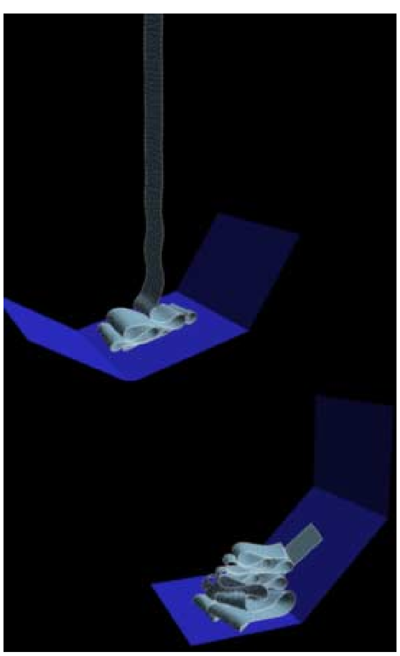

Fig. 5 : Complex collisions (Calc time : 5 hrs).

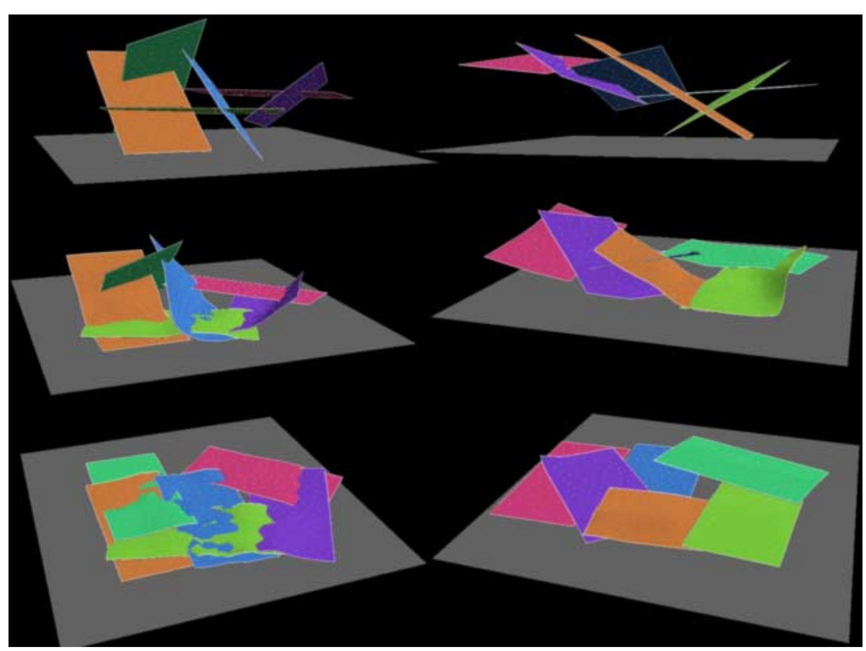

Fig. 6 : Falling without, and with collision consistency correction.

idea is to ensure global consistency of the collision behavior within and between these regions.

Regions are efficiently computed by neighborhood walking and labelling through all detected collisions. We update incrementally the regions as the surfaces are moving through the detected collisions.

For a consistent collision situation between two surfaces:

* The collision orientation should remain constant within the collision regions of the two surfaces, so that the whole collision group behave consistently.
* The two regions should be oriented accordingly, so that the surfaces are repulsing each other if they are at the correct side of each other, or attracting if they are not.

For determining the region orientation, we use a statistical evaluation of all the collision orientations within the region, according to their reliability (whether they have been deduced from remnance or tracking). We then force all the collisions of the region to the same orientation.

Using this process whenever any inconsistency has been detected (using edge-to-triangle interference), we can efficiently correct the situation by forcing every detected collision to behave accordingly to the majority's choice.

4.4 - Incremental collision detection

In situations where large surfaces collide but where the deformations and relative movements remain small, and when the overall geometrical configuration remains constant, recomputing proximities which have simply evolved in position is meaningless.

We provide some incremental algorithms that will update the existing proximities between each animation step, using some quick geometric computations for each of them, instead of using the whole global detection computation.

Depending of the situation, different actions are provided:

* The proximity direction and distance is recomputed according to the displacement of the concerned elements.
* The proximity may evolve through neighboring elements (sliding).
* The proximity is forgotten (or remnant) when the concerned elements are not in the detection range.
* Some new proximities may appear from the neighboring elements of existing collisions.

We could imagine using exclusively incremental detection in situations involving only sliding of two permanently colliding surfaces, thus permitting very fast evaluation computation. However, new collision "zones" on topologically variable situations will not be detected. A good compromise is to alternate incremental detection with full detection, according to the simulation conditions.

5 - RESULTS

The mechanical simulation and collision detection algorithms have been implemented in the C language on Silicon Graphics workstations. An animation system has been designed to handle moving objects coming from various animation sources (fixed and frame-by-frame animated objects, objects transformed by mathematical transformations, and of course objects animated by mechanical simulation) that interact with the other objects.

Any object may be subjected to mechanical simulation (provided it is discretised into a triangular mesh). A cloth object is imported directly from existing panel design software, and is assembled using the same mechanical software. Once this process is finished and the cloth becomes a single object, it may be handled as any other mechanical object.

We have used this animation system for testing our algorithms. Mainly, the following tests have been performed.

5.1 - Mechanical properties

We have simulated sets of objects with different mechanical data which illustrate the different mechanical properties that can be handled:

In order to simulate some "exceptional" conditions where our cloth, or any kind of other object, is subject to variable interaction that will cause much random wrinkling and deformation, we put our objects in a rotating cylinder that animates them the same way a drying machine would.

The dryer is primarily a test which validates the efficiency and the robustness of the collision detection algorithm. Any kind of collision configuration may occur, and interaction between different collisions is high. Secondly, it is a good test for the validation of the mechanical model under the high deformations caused by collisions between deforming objects, as well as for collision response and friction. It also verifies the numerical stability of the model.

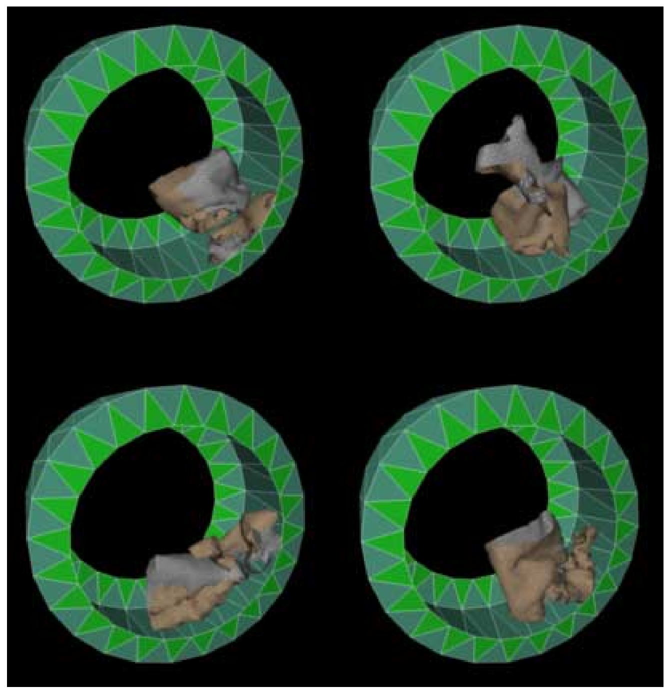

Fig. 7 : Crumpling garments in the dryer.
Calc time : 8 hrs for 1 min animation.

5.2 - Cloth assembly and simulation

To define cloth objects, we use existing software for designing 2D panels, as described in [36]. The cloth is assembled according to its seaming borders by a 3D mechanical simulation where "elastics" provide forces to join the seaming lines together.

We have considerably improved the seaming process by using well-tuned viscoelastic forces, which also simulate transversal viscosity and damping for directing the panel borders straight to the destination.

Once the seaming lines are close enough, we engage a "hard" seam by topologically merging together the corresponding vertices and edges to obtain a single surface. As there is no constraint with respect to the discretisation of our surfaces, we can imagine building any kind of object using this process.

Designed this way, a cloth object is handled in the simulation system the same way as any other mechanical object. There are no constraints according to the geometrical context (the cloth may not be worn by a body). We have complete freedom to decide how to manipulate this object.

6 - CONCLUSION AND FUTURE WORK

We have developed an efficient set of techniques that allows us to simulate any kind of deformable surface in various mechanical situations. Our main contribution was to design algorithms that could handle very general, context-free situations: First by implementing a robust process for mechanical simulation that can cope with difficult situations involving high deformations and numerous collisions, and secondly by linking it to a powerful collision detection system that, in addition to good performance for collision and self-collision detection, is able to deal with the lack of geometrical context for correctly orienting collision response.

Our main idea was to keep the system, which was basically designed for cloth simulation, as general and versatile as possible, by not restricting simulations to special contexts, or objects to certain shapes.

Besides providing performance and robustness improvements, this powerful tool can be used to build applications involving garments in some very particular situations, like grasping and folding. Because of the versatility of the simulation algorithm, we can further imagine not only simulating the cloth worn by the body, but also trying to realistically reproduce an actor grasping his/her clothes and dressing himself/herself.

ACKNOWLEDGMENTS

This work is supported by the Swiss National Research Foundation and the European ESPRIT HUMANOID2 project. Thanks to all the people that have contributed to it, and particularly to Jean-Claude Mousally for his assistance in preparing some illustrations and Hans-Martin Werner for reviewing the English text.

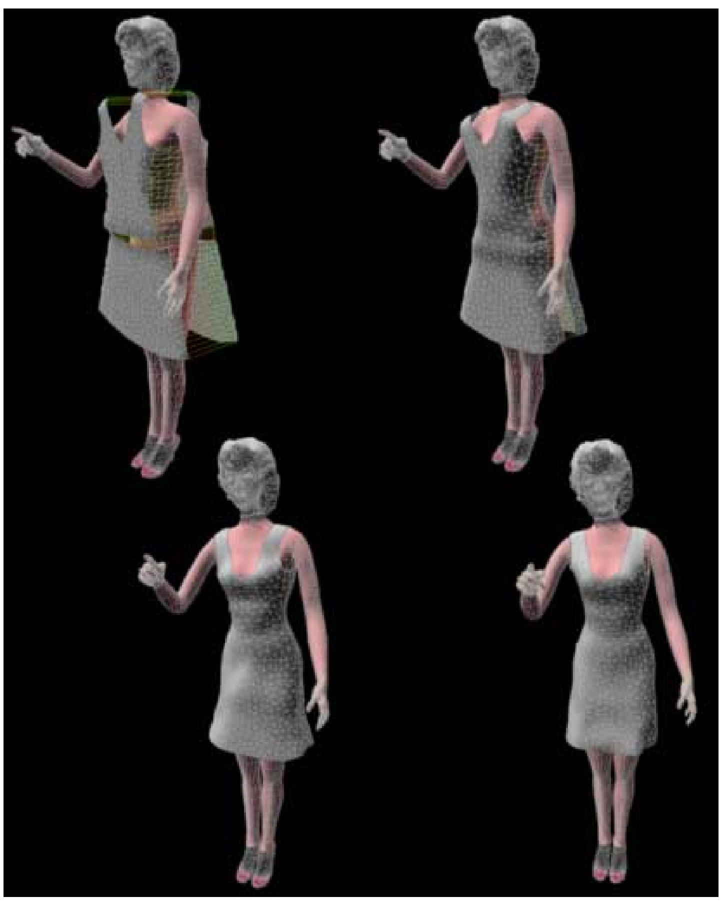

Fig. 8 : Seaming garment panels around tho body (Calc time : 10 min).

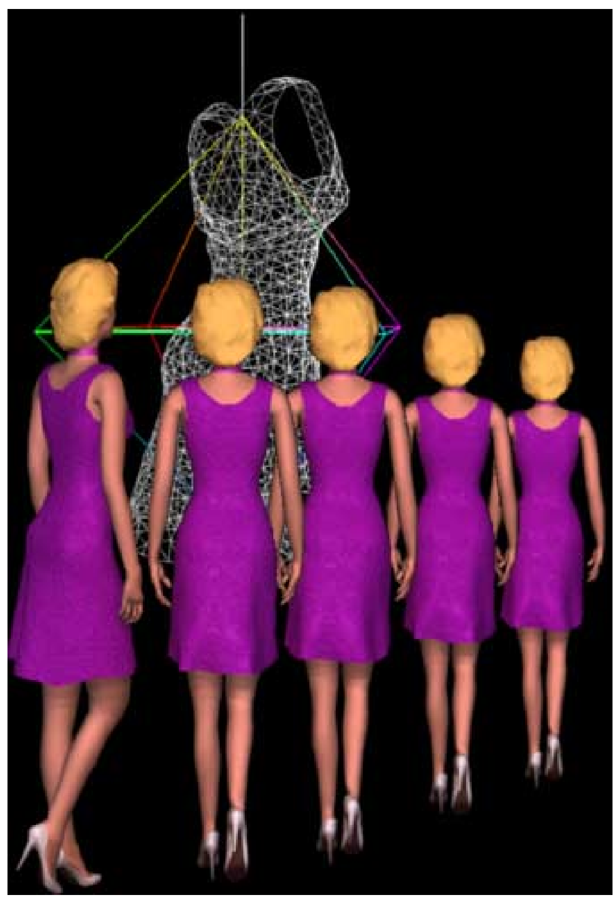

Fig. 9 : Calc time : 5 hrs for 15 sec animation.

BIBLIOGRAPHY

[1] : **J. Amirbayat, J.W.S. Hearle,** *"The Complex Buckling of Flexible Sheet Material - Part1 : Theoretical Approach"*, Int. J. Mech. Sci., 28(6), pp 339-358, 1986.

[2] : **M. Aono,** *"A Wrinkle Propagation Model for Cloth"*, Computer Graphics International Proc., Springer-Verlag, 1990.

[3] : **D. Baraff,** *"Curved Surfaces and Coherence for Non-Penetrating Rigid Body Simulation"*, Computer Graphics (proc. SIGGRAPH'90), 24(4), pp 19-28, 1990.

[4] : **D. Baraff, A. Witkin,** *"Dynamic Simulation of Non-Penetrating Flexible Bodies"*, Computer Graphics (proc. SIGGRAPH'92), 26(2), pp 303-308, 1992.

[5] : **D. Baraff, A. Witkin,** *"Global Methods for Simulating Flexible Bodies"*, Computer Animation Proc., Springer-Verlag, pp 1-12, 1994.

[6] : **D.E. Breen, D.H. House, M.J. Wozny,** *"Predicting the Drape of Woven Cloth using Interacting Particles"*, Computer Graphics (proc. SIGGRAPH'94), .28(4), pp 365-372, 1994.

[7] : **R.Barzel**, *"Physically-Based Modeling for Computer Graphics"*, Academic Press, 1992.

[8] : **C. R. Calladine,** *"Theory of Shell Structures"*, Cambridge University Press, 1983.

[9] : **J.F. Canny, D. Manocha,** *"A new approach for Surface Intersection"*, International journal of Computational Geometry and Applications, 1(4), pp 491-516, 1991.

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

[11] : **E. F. Denby,** *"The Deformation of Fabrics during Wrinkling - A Theoretical Approach"*, Textile Reserch Journal, Landcaster PA., 46, pp 667-670, 1976.

[12] : **S.G. Dhande, P.V.M. Rao, S. Tavakkoli, C.L. Moore,** *"Geometric Modeling of Draped Fabric Surfaces"*, IFIP Trans. Graphics Design and Visualisation, North Holland, pp 349-356, 1993.

[13] : **T. Duff,** *"Interval Arithmetic and Recursive Subdivision for Implicit Functions and Constructive Solid Geometry"*, Computer Graphics (proc. SIGGRAPH'92), 26(2), pp 131-138, 1992.

[14] : **K.L. Gay, L. Ling, M. Damodaran,** *"A Quasi-Steady Force Model for Animating Cloth Motion"*, IFIP Trans. Graphics Design and Visualisation, North Holland, pp 357-363, 1993.

[15] : **H. Gould, J. Tobochnik,** *"An introduction to computer simulation methods: Applications to physical systems"*, Reading Mass., Addison-Wesley, 1988.

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

[17] : **D. E. Breen, D. H. House, P. H. Getto,** *"A Physically-based Particle Models of woven cloth"*, The Visual Computer, 8(5-6), Springer-Verlag, Heidelberg, pp 264-277, 1992.

[18] **T.L. Kunii, H.Gotoda,** *"Modeling and Animation of Garment Wrinkle Formation processes"*, Computer Animation Proc., Springer-Verlag, pp 131-146, 1990.

[19] : **B. Lafleur, N. Magnenat Thalmann, D. Thalmann,** *"Cloth Animation with Self-Collision Detection"*, Proc. of the IFIP conference on Modeling in Computer Graphics (proc. SIGGRAPH'91), Springer, pp 179-187, 1991.

[20] : **M.C. Lin, D. Manocha,** *"Interference Detection between Curved Objects for Computer Animation"*, Computer.Animation Proc., Springer-Verlag, pp 43-55, 1993.

[21] : **A.H. Manich, M.D. De Castellar,** *"Elastic Recovery of Polyester Staple Fiber Rotor Spun Yarn"*, Textile Research Journal, Landcaster PA., 62, pp 196-199, 1992.

[22] : **M. Moore, J. Wilhelms,** *"Collision Detection and Response for Computer Animation"*, Computer Graphics (proc. SIGGRAPH'88), 22(4), pp 289-298, 1988.

[23] : **W.E. Morton, J.W.S. Hearle,** *"Physical properties of textile fibers"*, Manchester and London, The textile institute, Butterworths, 1962.

[24] : **L.F. Palazzi, D.R. Forsey,** *"A Multilevel Approach to Surface Response in Dynamically Deformable Models"*, Computer Animation Proc., Springer-Verlag, pp 21-30, 1994.

[25] : **D. Rosenthal,** *"Resistance and Deformation of Solid Media"*, N-Y, Pergamon Press, 1974.

[26] : **M. Shinya, M.C. Forgue,** *"Interference Detection through Rasterisation"*, The journal of Visualisation and Computer Animation, J. Wiley & Sons, 4(2), pp 132-134, 1991.

[27] : **J.M. Snyder, A.R. Woodbury, K. Fleisher, B. Currin, A.H. Barr,** *"Interval Methods for Multi-Point Collisions between Time-Dependant Curved Surfaces"*, Computer Graphics annual series, pp 321-334, 1993.

[28] : **D. Terzopoulos, J.C. Platt, H. Bar,** *"Elastically Deformable Models"*, Computer Graphics (proc. SIGGRAPH'87), 21, pp 205-214, 1987.

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

[30] : **S. Timoshenko, J. N. Goodier,** *"Theory of Elasticity"*, 3rd ed., N-Y, McGraw-Hill, 1970.

[31] : **B. Von Herzen, A.H. Barr, H.R. Zatz,** *"Geometric Collisions for Time-Dependant Parametric Surfaces"*, Computer Graphics (proc. SIGGRAPH'90), 24(4), pp 39-48, 1990.

[32] : **P. Volino, N. Magnenat Thalmann,** *"Efficient Self-Collision Detection on Smoothly Discretised Surface Animations using Geometrical Shape Regularity"*, Computer Graphics Forum (EuroGraphics Proc.), 13(3), pp 155-166, 1994.

[33] : **R.C. Webb, M.A. Gigante,** *"Using Dynamic Bounding Volume Hierarchies to improve Efficiency of Rigid Body Simulations"*, Computer Graphics International Proc., Springer-Verlag, pp 825-841, 1992.

[34] : **R.C. Webb, M.A. Gigante,** *"Distributed, Multi-Person, Physically-Based Interaction in Virtual Worlds"*, Computer Graphics International Proc., Springer-Verlag, pp 41-48, 1993.

[35] : **J. Weil,** *"The synthesis of Cloth Objects"*, Computer Graphics (proc. SIGGRAPH'86), 4, pp 49-54, 1986.

[36] : **H.M. Werner, N. Magnenat Thalmann, D. Thalmann,** *"User Interface for Fashion Design"*, Graphics Design and Visualisation, IFIP Trans. North Holland, pp 197-204, 1993.

[37] : **A. Witkin, W. Welch,** *"Fast Animation and Control of Non-Rigid Structures"*, Computer Graphics (proc. SIGGRAPH'90), 24, pp 243-252, 1990.

[38] : **F. Yamaguchi,** *"An Unified Approach to Interference Problems using a Triangle Processor"*, Computer Graphics (proc. SIGGRAPH'85), 19, pp 141-149, 1985.

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

[40] : **M. Zyda, D. Pratt, W. Osborne, J. Monahan,** *"Real-Time Collision Detection and Response"*, The journal of visualisation and Computer Animation, 4(1), J. Wiley & Sons, pp 13-24, 1993.