From measured physical parameters to the haptic feeling of fabric

P. Volino*, P. Davy*, U. Bonanni*, N. Magnenat-Thalmann*
G. Böttcher**, D. Allerkamp**, F.-E. Wolter**

* *MIRALab - University of Geneva*
** *Welfenlab - University of Hanover*

Abstract

Real-time cloth simulation involves many computational challenges to be solved, particularly in the context of haptic applications, where high frame rates are necessary for obtaining a satisfactory tactile experience.In this paper, we present a real-time cloth simulation system which offers a compromise between a realistic physically-based simulation of fabrics and a haptic application with high requirements in terms of computation speed. We give emphasis on architecture and algorithmic choices for obtaining the best compromise in the context of haptic applications. A first implementation using a haptic device demonstrates the features of the proposed system and leads to the development of new approaches for the haptic rendering using the proposed approach.

Keywords: *haptics, force feedback, real-time, cloth-simulation, elastic deformation*

1. Introduction

A key factor for achieving convincing haptic interaction is the successful integration of realistic graphical rendering and adequate haptic rendering techniques. If the haptic interaction occurs with deformable objects belonging to everyday life, the simulation accuracy of the object's behavior plays a fundamental role, and physically-based simulation is necessary. At first glance, the advantage of physically-based simulation in the context of haptics seems obvious: The mechanical models driving the visual simulation could easily be used for computing the resulting forces for the haptic rendering in a direct way. However, the graphics rendering loop has different requirements compared to the haptic rendering loop in terms of refresh frequencies. While in graphics a refresh rate of 30 fps is quite acceptable, in haptics a response frequency of 300-1000 Hz is needed to ensure accurate interaction. For this reason, a good approach for the development of Virtual Reality (VR) Systems for the visual and haptic rendering of deformable objects is to consider a multi-layer architecture for what concerns the physical-based models driving the simulation [1]. The first layer consists of a coarse model representing the large-scale features of the deformable object (for the graphics rendering), while the second layer is a finer local model defining the surface characteristics close to the point of interaction dedicated to meet the haptic requirement.

This paper presents the visual and haptic rendering techniques proposed in the context of the HAPTEX Project, which aims at developing a Virtual Reality System for the multimodal perception of textiles [2]. The aim of the project is to reproduce the feeling of touching a cloth surface with two finger-tips, simulating both the large-scale motion of a square of fabric accurately described by its mechanical properties (strain-stress curves), and the small scale properties (texture and roughness) for the tactile perception.

In this paper, we will first examine the related work for this area with an emphasis on the way to measure the physical parameters. Then, the large-scale mechanical model simulating the macroscopic deformation of cloth will be presented. Finally, a first implementation of haptic interaction model will be presented.

2. Related Work

2.1 Understanding and Modeling the Properties of Fabrics

The process of handling fabrics to understand their properties and structure is called “fabric hand” [3]. Modeling the behavior of textiles is a complex task because of its dependency on several parameters such as flexibility, compressibility, elasticity, resilience, density, surface contour (roughness, smoothness), surface friction and thermal character [4]. For the large scale simulation, fabrics are approximated as thin 2D surfaces, thus we can avoid taking into consideration pressure-thickness, surface friction and surface roughness. However, we can use objective measurement methods to acquire other parameters such as tensile, shearing or bending variables.

An objective assessment of the fabric hand is achieved through hardware equipments, such as tensile testers or the Kawabata measurement systems. These equipments are able to test for textile properties and extract the strain-stress curves for weft and warp elongation, shear deformation, bending, pressure-thickness, surface friction and surface roughness. The values of the extracted parameters vary depending on the fiber type or fabric type and dimension. The data obtained from measurements on fabrics serves as an input to the mechanical model driving our cloth simulation system.

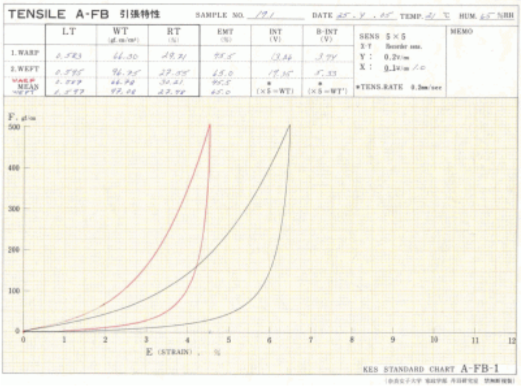


***Figure 1**: Strain stress curve from Kawabata tensile test*

An interesting consideration at this point is the possibility to go also in the opposite direction: through the use of an accurate cloth simulation scheme it is also possible to study new fabrics. We can develop and model non-existing textiles, which look and behave in a realistic way and have specific properties. Such virtual textiles would not base on any “real” counterparts, but could be produced by the textile industries after successful simulation.

2.2 Cloth Simulation

In computer graphics, cloth objects are represented by polygonal meshes. During cloth animation, the vertices of the cloth mesh are driven by the laws of the underlying physical model. The resulting equations are solved using numerical methods such as the semi-implicit Backward Euler method, which was first used by Baraff et al. [5] in the context of cloth simulation. Since then, semi-implicit integration has been widely used for cloth animation, as it provides better results in terms of stability and speed compared to other integration schemes [6]. An exhaustive overview of the state of the art in cloth animation research is given by [7]. In recent years, several research activities have been carried on in the field of cloth simulation, focusing on different aspects ranging from physical based models [8], [9] to integration schemes [5], [10] and collision response [11], [12].

In the following, we briefly describe existing simulation schemes underlying mechanical models for cloth simulation.

2.2.1. Continuum Mechanics

In another approach, mechanical equations of continuum mechanics are expressed along the curved surface, and then discretized for their numerical resolution. Such accurate schemes are however slow and not sufficiently versatile for handling large deformations and complex geometrical constraints (collisions) properly. Finite Element methods express the mechanical equations according to the deformation state of the surface within well-defined elements (usually triangular or quadrangular). Their resolution also involves large computational costs [13]. Another option is to construct a model based on the interaction of neighboring discrete points of the surface. Such particle systems allow the implementation of simple and versatile models adapted for efficient computation of highly deformable objects such as cloth.

2.2.2. Spring-Mass Models

The simplest particle system one can think of is a spring-mass system. In this scheme, the only interactions are forces exerted between neighboring particle couples, similarly as if they were attached by springs (described by a force/elongation law along its direction, which is actually a rigidity coefficient and a rest length in the case of linear springs) [14]. Spring-mass schemes are very popular methods, as they allow simple implementation and fast simulation of cloth objects. There has also been recent interest in this method as it allows quite a simple computation of the Jacobian of the spring forces, which is needed for implementing semi-implicit integration methods.

The simplest approach is to construct the springs along the edges of a triangular mesh describing the surface. This however leads to a very inaccurate model that cannot describe accurately the anisotropic strain-stress behavior of the cloth material, and also not the bending. More accurate models are constructed on regular square particle grids describing the surface. While elongation stiffness is modeled by springs along the edges of the grid, shear stiffness is modeled by diagonal springs and bending stiffness is modeled by leapfrog spring along the edges (Figure 2). This model is still fairly inaccurate because of the unavoidable cross-dependencies between the various deformation modes relatively to the corresponding springs. It is also inappropriate for nonlinear elastic models and large deformations. More accurate variations of the model consider angular springs rather than straight springs for representing shear and bending stiffness, but the simplicity of the original spring-mass scheme is then lost. The model detailed in Part 3 alleviates these issues by combining the simplicity and speed of particle systems with the accuracy of a model that evaluates strain and stress on the true deformed surface of cloth.

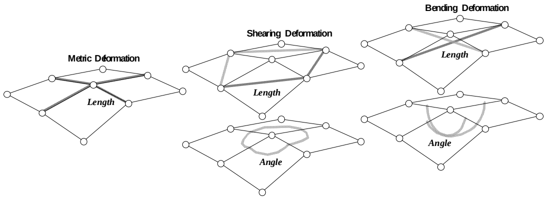


Figure 2: *Using length or angle springs for simulating cloth with a square particle system grid.*

2.3 Toward Haptic Rendering of Textiles

Haptic Rendering algorithms are responsible for reproducing the forces acting on the simulated objects and transmit them to the haptic interface. In the case of static, non deformable objects, the computation of these forces is limited to reproducing the object's stiffness according to the user's position and interaction speed. When simulating dynamic deformable surfaces such as textiles, however, this issue grows in complexity, as the simulated surface has its own behavior which influences the haptic feedback and therefore has to be taken into consideration.

There has been very little research on the haptic rendering of textiles so far [15] [16] [17]. In all cases, projects dealing with haptic perception of textiles targeted very specific scenarios and were severly limited in terms of visual rendering (no real-time animation), used hardware (low-cost commercial products or expensive ad-hoc devices) or targeted realism (static, non deformable textiles).

A system which provides both force feedback and tactile perception of virtual textiles through a haptic device, and is synchronized with a real-time, physical based graphical simulation of the fabric has still not been developed yet. To achieve this challenging goal, several advances in all related fields are necessary: from the techniques of fabric measurements, to the development of novel haptic devices and the improvement of physical-based models for real-time cloth simulation.

3. A Mechanical Model for Real-Time Cloth Simulation

3.1. Key Techniques for Real-Time Cloth Simulation

The most important challenge is to simulate in real-time cloth materials which are accurately described by nonlinear anisotropic mechanical models (weft, warp and shear strain-stress curves).

Most existing real-time simulation systems consider drastic simplifications in the mechanical model (isotropic linear elasticity) and even change the mechanical nature of the material (for example, reduce stiffness by simulating very deformable materials) for getting more computation speed. However, our application requires the simulation of precise mechanical behavior of cloth.

Going away from traditional grid-based spring-mass approaches, we propose a very accurate particle system which reaches the accuracy of first-order nonlinear finite elements while retaining the simplicity of particle systems. This model uses a fast computational method for evaluating precisely the strain state of a triangle of the cloth surfaces, and may then turn the corresponding stress back into particle forces.

Real-time constraints however impose a severely limited complexity of the mechanical system. Here are the main simplifications that we consider:

- Simulation of a coarse grid, typically limited to a thousand of triangle elements. What matters for the computation time is not only the number of elements to be computed, but also the size of these elements (the smaller they are, the stiffer the numerical system will be).
- Neglecting bending stiffness. Bending stiffness is usually very low in usual fabric materials, and it usually does not make any sense simulating bending effects producing millimeter wrinkles with centimeter mesh elements. Bending properties are also quite costly to compute.
- Not performing self-collision within the cloth. Self-collision techniques are usually quite costly in computational resources, and processing the detected collisions is also complex. Given the simple context of our animation (a hanging piece of cloth), we restrict collision processing to detecting contacts between the cloth object and the haptic virtual fingertips.

Large simulation time steps should be considered for the numerical integration of the system, and this is obtained through the use of implicit numerical integration methods. This however slightly alters the dynamic behavior of the cloth (numerical damping). Experimental tests can help finding the best time-steps producing animation of cloth objects with damping in par with the damping observed in real cloth materials.

3.2. Mechanical Modeling of Cloth

The mechanical behavior of cloth (approximated as a thin surface) is decomposed in in-plane deformations (the 2D deformations along the cloth surface plane) and bending deformation (the 3D surface curvature).

The tensile behavior of cloth is described by relationships relating, for any cloth element, the stress $\boldsymbol{\sigma}$ to the strain $\boldsymbol{\varepsilon}$ (for elasticity) and its speed $\boldsymbol{\varepsilon'}$ (for viscosity) according to the laws of viscoelasticity. For cloth materials, strain and stress are described relatively to the weave directions weft and warp following three components: weft and warp elongation (**uu** and **vv**), and shear (**uv**). Thus, the general viscoelastic behavior of a cloth element is described by strain-stress relationships as follows:

$$\begin{aligned}
&\sigma_{uu}(\varepsilon_{uu},\varepsilon_{vv},\varepsilon_{uv},\varepsilon_{uu}{}',\varepsilon_{vv}{}',\varepsilon_{uv}{}')\\
&\sigma_{vv}(\varepsilon_{uu},\varepsilon_{vv},\varepsilon_{uv},\varepsilon_{uu}{}',\varepsilon_{vv}{}',\varepsilon_{uv}{}')\\
&\sigma_{uv}(\varepsilon_{uu},\varepsilon_{vv},\varepsilon_{uv},\varepsilon_{uu}{}',\varepsilon_{vv}{}',\varepsilon_{uv}{}')
\end{aligned} \qquad (1)$$

Assuming to deal with an orthotropic material (usually resulting from the symmetry of the cloth weave structure relatively to the weave directions), there is no dependency between the elongation components (**uu** and **vv**) and the shear component (**uv**). Assuming null Poisson coefficient as well (a rough approximation), all components are independent, and the fabric elasticity is simply described by three independent elastic strain-stress curves $\sigma_{uu}(\varepsilon_{uu})$, $\sigma_{vv}(\varepsilon_{vv})$, $\sigma_{uv}(\varepsilon_{uv})$ (weft, warp, shear), along with their possible viscosity counterparts.

In the same manner, viscoelastic strain-stress relationships relate the bending momentum to the surface curvature for weft, warp and shear. With the typical approximations used with cloth materials, the elastic laws are only two independent curves along weft and warp directions (shear is neglected), with their possible viscosity counterparts.

3.3. A Fast and Accurate Particle System Model

Our initial goal is to design an efficient mechanical simulation scheme for simulating the large-scale behavior of the cloth.

Because of the real need of representing accurately the anisotropic nonlinear mechanical behavior of cloth in applications, spring-mass models are inadequate, and we need to find out a scheme that really simulates the viscoelastic behavior of actual surfaces. For this, we have defined a particle system model that relates this accurately over any arbitrary cloth triangle through simultaneous interaction between the three particles which are the triangle vertices. Such a model integrates directly and accurately any strain-stress model using polynomial spline approximations of the strain-stress curves, and remains accurate for large deformations. Details on this approximation of strain-stress curves are given in appendix A.2.

3.3.1. Assumptions

Cloth is typically represented by curved surfaces composed of polygonal meshes, being either triangular or quadrangular, and regular or irregular. In order to enrich these geometrical surfaces with the above mentioned mechanical properties, we need to define a specific mechanical model.

We consider the following constraints for designing the large-scale simulation system:

- The geometrical description of the cloth surface should be any triangular mesh.
- The mechanical properties of the cloth material is described by strain-stress curves, approximated analytically as independent piece-wise polynomial splines (weft, warp, shear).
- Bending stiffness is neglected, as it does not make sense to spend time computing accurately their very weak effect comparatively to the scale required for real-time simulation of the model (the size of the elements are much more likely to limit the sharpness of the wrinkle patterns than the bending stiffness of the cloth).

3.3.2. Description of the Model

In this model, a triangle element of cloth is described by 3 2D coordinates (**ua**, **va**), (**ub**, **vb**), (**uc**, **vc**) describing the location of its vertices **A**, **B**, **C** on the weft-warp coordinate system defined by the directions **U** and **V** with an arbitrary origin. They are orthonormal on the undeformed cloth (figure 2). Out of them, a pre-computation process evaluates the following values:

$$
\begin{aligned}
&Rua = d^{-1}\left(vb - vc\right) && Rva = -d^{-1}\left(ub - uc\right)\\
&Rub = -d^{-1}\left(va - vc\right) && Rvb = d^{-1}\left(ua - uc\right)\\
&Ruc = d^{-1}\left(va - vb\right) && Rvc = -d^{-1}\left(ua - ub\right)
\end{aligned}
$$

$$
\textit{with}
$$

$$
d = ua\left(vb - vc\right) + ub\left(vc - va\right) + uc\left(va - vb\right) \qquad (2)
$$

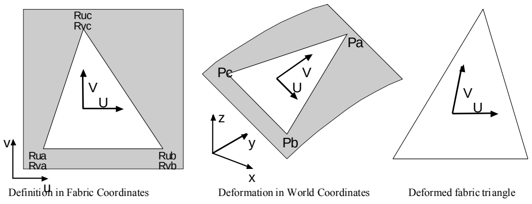
During the computation process, the current deformation state of the cloth triangle is evaluated using the current 3D direction and length of the deformed weft and warp direction vectors **U** and **V**. They are computed from the current positions **Pa**, **Pb**, **Pc** of its supporting vertices as follows:

$$
\begin{aligned}
U &= Rua\ Pa + Rub\ Pb + Ruc\ Pc\\
V &= Rva\ Pa + Rvb\ Pb + Rvc\ Pc
\end{aligned} \qquad (3)
$$

The current in-plane strains ε of the cloth triangle is then computed with the following formula:

$$
\begin{aligned}
&\varepsilon_{uu} = |U| - 1 \qquad \varepsilon_{vv} = |V| - 1\\
&\varepsilon_{uv} = \frac{|U+V|}{\sqrt{2}} - \frac{|U-V|}{\sqrt{2}}
\end{aligned} \qquad (4)
$$

We have chosen to replace the traditional shear deformation evaluation based on the angle measurement between the thread directions by an evaluation based on the length of the diagonal directions. The main advantage of this is a better accuracy for large deformations (the computation of the behavior of an isotropic material under large deformations remains more axis-independent).



Definition in Fabric Coordinates

Deformation in World Coordinates

Deformed fabric triangle

Figure 3: A triangle of cloth element defined on the 2D cloth surface (left) is deformed in 3D space (center) and its deformation state is computed from the deformation of its weft-warp coordinate system (right).

For applications that model internal in-plane viscosity of the material, the "evolution speeds" of the weave direction vectors are needed as well. They are computed from the current triangle vertex speeds **Pa'**, **Pb'**, **Pc'** as follows:

$$\begin{aligned} U' &= Rua\,Pa' + Rub\,Pb' + Ruc\,Pc' \\ V' &= Rva\,Pa' + Rvb\,Pb' + Rvc\,Pc' \end{aligned} \tag{5}$$

Then, the current in-plane strain speeds **ε'** of the triangle is computed:

$$\varepsilon_{uu}{}' = \frac{U \cdot U'}{|U|} \qquad \varepsilon_{vv}{}' = \frac{V \cdot V'}{|V|}$$
$$\varepsilon_{uv}{}' = \frac{(U+V)\cdot(U'+V')}{|U+V|\sqrt{2}} - \frac{(U-V)\cdot(U'-V')}{|U-V|\sqrt{2}} \tag{6}$$

At this point, the in-plane mechanical behavior of the material can be expressed for computing the stresses **σ** out of the strains **ε** (elasticity) and the strain speeds **ε'** (viscosity) using the curves (1). Finally, the force contributions of the cloth triangle to its support vertices computed from the stresses **σ** as follows:

$$\begin{aligned} Fa &= -\frac{d}{2}\left((Rua\,\sigma_{uu} + Rva\,\sigma_{uv})\frac{U}{|U|} + (Rua\,\sigma_{uv} + Rva\,\sigma_{vv})\frac{V}{|V|} \right) \\ Fb &= -\frac{d}{2}\left((Rub\,\sigma_{uu} + Rvb\,\sigma_{uv})\frac{U}{|U|} + (Rub\,\sigma_{uv} + Rvb\,\sigma_{vv})\frac{V}{|V|} \right) \\ Fc &= -\frac{d}{2}\left((Ruc\,\sigma_{uu} + Rvc\,\sigma_{uv})\frac{U}{|U|} + (Ruc\,\sigma_{uv} + Rvc\,\sigma_{vv})\frac{V}{|V|} \right) \end{aligned} \tag{7}$$

It is important to note that when using semi-implicit integration schemes, the contribution of these forces in the Jacobian **∂F/∂P** and **∂F/∂P'** can easily be computed out of the curve derivatives **∂σ/∂ε** and the orientation of the vectors **U** and **V**.

3.3.3. Numerical Integration

The equations resulting from the mechanical formulation of particle systems do usually express particle forces **F** depending on the state of the system (particle positions **P** and speeds **P'**). In turn, particle accelerations **P"** is related to particle forces **F** and masses **M** by Newton's 2nd law of dynamics. This leads to a second-order ordinary differential equation system, which is turned to first-order by concatenation of particle position **P** and speed **P'** into a state vector **Q**. Some details about how the differential equation system is set are given in appendix A.1. A vast range of numerical methods has been studied for solving this kind of equations.

The most widely-used method for cloth simulation is currently the semi-implicit Backward Euler method, which was first used by Baraff et al. [5] in the context of cloth simulation. As any implicit method, it alleviates the need of high accuracy for the simulation of stiff differential equations, offering convergence for large timesteps rather than numerical instability (a step of the semi-implicit Euler method with "infinite" timestep is actually equivalent to an iteration of the Newton resolution method) [6]. Many variations of this method are available, through various computational simplification approaches [18][19][8].

The formulation of a generalized implicit Euler integration is the following:

$$Q_{(t+dt)} - Q_{(t)} = Q'_{(t+\alpha\,dt)}\,dt \tag{8}$$

The derivative value is not known at a moment after **t**, and is then extrapolated from the value at moment **t** using the Jacobian, leading to the semi-implicit expression which requires the resolution of a linear system:

$$Q_{(t+dt)} - Q_{(t)} = \left(I - \alpha \left.\frac{\partial Q'}{\partial Q}\right._{(t)} dt \right)^{-1} Q'_{(t)}\,dt \tag{9}$$

We have introduced the coefficient **α** so as to modulate the "implicitiy" of the formula[20]. Hence, **α = 1** is the regular implicit Backward Euler step (stable), whereas **α = 0** is the explicit Forward Euler step (unstable), and **α = 1/2** is the 2nd-order implicit Midpoint step (most accurate, at the threshold of stability).

The **α** parameter is a good handle for adjusting the compromise between stability and accuracy. While maximum robustness is obviously observed for large values, reducing its value increases accuracy (reduces numerical damping) at the expense of stability, and speeds up the computation as well (better conditioning of the linear system to be resolved). Hence, for our real-time application, we can modulate this value according to the simulation and interaction requirements to offer the best accuracy and robustness. The requirement of high robustness of haptic applications also makes higher-order methods [21] [10] less adequate.

3.4. Collision Processing

Due to constraints for real-time performance, self-collision detection on the cloth object is ignored. The only task in this field is to detect collisions between solid objects and the cloth surface. This task is performed through an adapted bounding-volume hierarchy algorithm, which uses a constant Discrete-Orientation-Polytope hierarchy constructed on the mesh [12] [22]. The solid object is itself embedded in a polytope, and collision detection is done through traversal of the colliding nodes of the hierarchy.

4. Haptic Rendering

A first implementation of haptic simulation has shown a few limitations. Due to the bidirectional nature of haptic, adding its support can lead to considerations on two different forces. First, in section 4.1 we present the method used to send the force to the user. Then, in section 4.3. we discuss the forces sent to the mechanical model. Furthermore one of the main issues in haptic rendering is to find a solution for the problem of different update rates. Indeed, the user communicates with the virtual environment through a probe, the "haptic cursor". This cursor is used to transmit forces to the surface and to the user. A high update rate of forces (1 kHz) is needed to give the user a satisfying experience of touch and feel of virtual objects. Since the simulation of the mechanics being synchronized with the visual rendering loop runs at a much lower frequency (<40 Hz) linking the force rendering directly to the physical model leads to problems highlighted in the section 4.2.1. Our solution to this problem is the introduction of an intermediate layer responsible for the simulation of the small section of the surface being in contact with the haptic cursor. This local geometry is described in the following section 4.2.2.

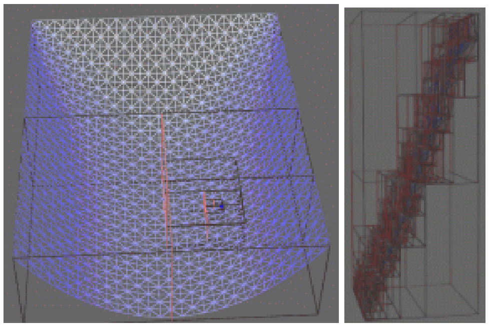


Figure 4: *Screenshots of a list of touched boxes for a point collision detection and of the full hierarchy of bounding boxes.*

4.1 Computing Forces Sent to the User

Different methods are available in the haptic community to define how the force could be sent from the model. A point based inter-action is currently considered. The force is calculated by detecting and generating responses for the collision between the haptic cursor and the surface.

4.1.1Constrain- Based Method

To detect the collision between the surface and the haptic cursor, a global coarse level collision is computed as described in section 3.4, then the exact collision is computed for the set of triangles in the selected bounding box.

At the moment when a collision occurs, the HIP (Haptic Interface Point) is penetrating the surface and the IHIP (Ideal Haptic Interface Point) is forced to stay on the surface. Then the force sent to the haptic device is calculated based on the relative position of the HIP and the IHIP, as it can be seen on figure 5. It can be computed from the laws of Hooke and Coulomb[23].

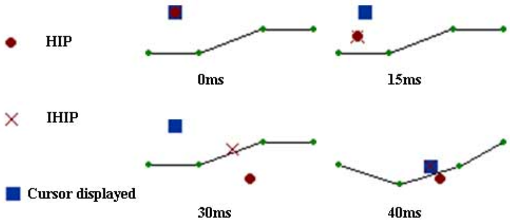


Figure 5*: Penetration of the haptic cursor; at first the HIP and the IHIP share the same place then the IHIP is projected on the closest point of the surface.*

The history of the haptic cursor path is necessary to determine if we are on collision. To do so, an efficient pseudo test method [24] has been used. The result achieved can be seen on figure 6.

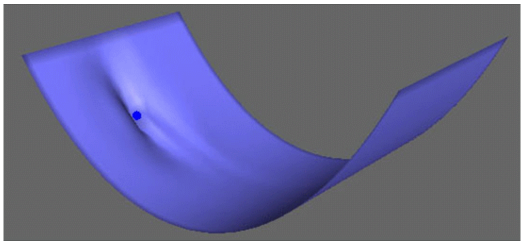


Figure 6: Screenshot of the deformed surface

4.1.2 Continuous force generation on discrete polygonal surface

For our application of displaying a smooth surface, the previous algorithm is insufficient. The edges of the triangular mesh became perceivable as the mesh is a rough approximation by the particle-system. To provide the user with an impression of a smooth transition between triangles, several approaches are presented.

An approach with a low complexity presented in [24], [25] is called "Force Shading". Likewise the Gouraud shading in graphical rendering, we define a vector field being a substitute of the face normal at an arbitrary point on the triangle. An interpolation between the vertex normals described in barycentric coordinates is used for the evaluation. Although this method is a convenient way to make the normal vector continuous at the edges, it neglects the accurate curvature of the corresponding surface.

In contrast to computing the vertex normals by averaging the adjacent face normals, an alternative approach [26] considers the in-plane mechanics to control the vertex normals. A triangular Beziér-patch interpolating vertex position and normal is used to smooth the surface. The approximation is tangent continuous on a single patch. It is not necessarily tangent continuous between patches. However, it gives an improved force feedback of the surface compared to the triangular mesh.

4.2 Local Refinement

4.2.1 Need of a quick simulation thread

For the implementation of the rendering system, two threads were used: the haptic thread and the graphic thread where the simulation was running. While the state of the haptic is updated every millisecond, the state of the simulation was updated every 40 milliseconds (assuming a frequency of 25Hz). This is illustrated by figure 7.

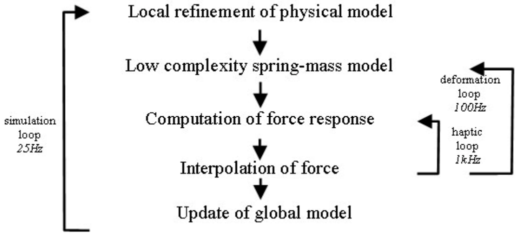


Figure 7: *Flowchart of the haptic rendering, showing synchronization of the different components.*

This difference in period leads to synchronization issues and since we assume in haptic a latency of 300 – 1000 Hz, the user would feel the discontinuities when the simulation is updated. We propose to solve this problem by introducing an intermediate layer responsible for the simulation of a small section of the surface being in contact with the haptic cursor. This simulation makes use of simplifications resulting in computations which are not as complex as in the main simulation but it is sufficiently fast for the haptic rendering.

4.2.2 Local Refinement of the Physical Model

Maintaining the update rate requires that we restrict ourselves to a small section of the surface for physically accurate interactions. By taking into account the position of the cursor and its finite velocity the motion area is well defined. Hence, we can define a proximity region given by a bounding volume, where the cursor and the surface may possibly have contact. Restricting our local consideration and computation to be performed in the haptic loop to the parts of the surface in the bounding volume allows us to reduce our computation effort to a minimum. This so-called local geometry is used for our computations in the haptic loop.

Zhang et al. [27] proposed to apply a geometric subdivision scheme on a mesh in conjunction with a spring-mass-system to model elastic deformations. The latter method is well established in Computer Graphics, as it is a useful tool to create approximately smooth surfaces from polygon meshes. Although creating refinements of the mesh with the geometric algorithms is straightforward, the determination of the new physical parameters of our mechanical model is challenging. Inserting new particles in the model requires a redistribution of masses covered by the originally involved particles representing the surface. Furthermore, the increased amount of the particles in the model yields a higher computational load within the haptic loop. To overcome the latter issue, a simplification of the mechanical model has to be made. If we assume that the user is normally exploring the mechanical surface at a low velocity, we can neglect the second order terms of the strain-stress relationship. Due to the small deformations occurring in a time frame of milliseconds, the mechanics can be described sufficiently well by the laws of elasticity neglecting external forces like gravity. As a result we can use the simpler spring-mass-model to model the local geometry under deformation without loosing much accuracy compared to the physically precise mechanical model.

To prevent the local geometry, i.e. the small section of the surface being in contact with the haptic cursor, from diverging too far from the main mechanical simulation we assume constant velocity at its border. This implies that no forces affect the particles on the border of the local geometry.

4.3 Force sent to the Mechanical Model

A global force is sent by the user to deform the mechanical model. And due to its nature of particle system, different forces have to be sent to a set of particles.

4.3.1 First Approach and Limitation

The first approach for this distribution has been to distribute the force on the particles forming a triangle according to their barycentric coordinates. But this approach was insufficient because it does not give the same profile of a surface depending of the position of the haptic cursor. It is illustrated on figure 8.

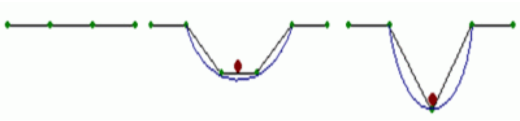


Figure 8: Barycentric coordinates distribution does not give the same profile

4.3.2 Transmission of Forces

While we have discussed the force rendering strategies extensively, we have to address the deformation of the local geometry and its integration into the simulation. From the physical point of view, a deformation can be seen as result of induced energy. In our case the induced energy may be understood as being partly contained in the elastic material as tensile energy of the stretched textile. The deformation model in 3.2.2 so far has been used to describe the deformation of the textile that will take place in case the mass particles in that model are exposed to extrinsic forces during some time interval. The model may be used also for explaining to what extend the deformation energy will be stored in the stretched textile. For an accurate physical model one has also to consider the dissipation of energy to be defined by the inelastic behavior of the textile. In the latter case the deformation energy may perhaps be partly lost in a plastic deformation process taking place in the woven material during the tensile stress of the textile.

Having a closer look at the point contact case, we assume that an external mass particle p_e with mass m_e and speed v_e hits the textile mesh in a surface point p_s with mass m_s and speed v_s, defined by the particles of the associated triangle using barycentric coordinates. Then we assume here now that the impulse of the external particle is transferred and the particles are now starting to travel together and may be considered as one particle of mass m_e+m_s moving with a new velocity v_{es}. Some part possibly even all of the kinetic energy obtained from the coupled mass particle with velocity v_{es} must now to be absorbed by the textile, e.g. it may be transformed into elastic energy stored in the subsequent stretching of the mesh. That stretching may be modeled as being performed in timesteps until the kinetic energy of the recent impulse is transferred into new strain energy that has been induced into the textile mesh during the subsequent timesteps.

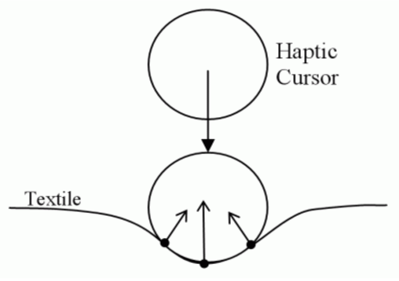


***Figure 9**: Force components contributing to force response.*

Considering an arbitrary shape of the haptic cursor, we can transfer the energy in a similar way. By observing the contact in the continuous time domain and focusing on a movement perpendicular to the surface we get in the moment of contact with the textile only one contact point on the surface. Moving forward in time the region of contact will increase around the point. Moreover, the initial contact point will be displaced by the cursor through its motion. Due to the displacement of the surface points from their rest position, we stretch the textile, by the latter displacement inducing energy into the system. The response force of the textile against the movement of the cursor deforming the surface will be defined by the deformation model in 3.3.2. . While we increase the tensile strain of the textile we also increase the internal force between the particles. The component of the internal force being perpendicular to the surface at each surface point is responsible for the relaxation of the textile into its rest position. As the component of the force is also in opposite direction of the force applied onto the haptic cursor, the summation of the perpendicular components over the surface points being in contact with the cursor will be the response of the textile against the deformation.

In the discretization of time being the main issue in haptic rendering, we have to deal with the penetration of the haptic cursor into the textile as we cannot assume to detect the initial contact and its displacement. What we will find is a situation where the haptic cursor is either on one side of the textile or intersecting with the textile. In the latter case we have to correct the position of the surface points of the textile to be in contact with the cursor. Such a correction has to be done with respect to the needed kinetic energy. That means the acceleration and the velocity has to be corrected as well.

5. Conclusion

Achieving a convincing virtual textile simulation requires a good compromise to be reached between the need of accuracy in the material representation and the need of speed for obtaining visually realistic simulation framerates compatible with real-time perception. These factors have to be considered both in the visual and the haptic fields. A dedicated structure has to be proposed for adapting the different framerates required by the mechanical simulation and the haptic rendering computations. Hence, two separate computations threads were implemented: The first is a low-frequency thread for dealing with the complex large-scale simulation of the whole cloth surface, and which an accurate particle system representation integrated with state-of-the-art numerical methods for achieving quantitative accuracy of the nonlinear anisotropic behavior of cloth in real-time. The second is a high-frequency thread for computing the local data necessary for haptic rendering and for sending accurately haptic forces back to mechanical simulation.

Using this framework, we are now looking forward to integrating tactile rendering of the fabric material, so as to enhance the real perception of the cloth object to a new level.

6. Acknowledgements

Funding for this research is provided by the project "HAPtic sensing of virtual TEXtiles" (HAPTEX), supported by the Sixth Framework Programme (FP6) of the European Union (Contract No. IST-6549). The funding agency is the Future and Emerging Technologies (FET) area of the Information Society Technologies (IST) department.

APPENDIX

A.1 Explanation of the Differential Equation System

On a general level: We consider simultaneously (the parameterized) family of vector valued coupled differential equation systems covering the motions of all mesh vertex points $\boldsymbol{p_{alm}}$ (index *alm* parameterizes the mesh)

$$M_{alm} \cdot p''_{alm}(t) = F(p_{alm}(t), p'_{alm}(t))$$

(Being appropriately supplemented by non tangential and exterior forces)

The latter formula derives from Newton's second law of motion, it describes the force F (being determined by the particle mass and acceleration) by the non-linear function *F*. That function *F* incorporates the strain stress functions σ_{ij} who are depending on the strain functions ε_{ij}, being dependent on $\boldsymbol{p_{alm}}$. Furthermore σ_{ij} is also depending on the strain speeds ε'_{ij} who themselves are depending on $\boldsymbol{p_{alm}}$ and $\boldsymbol{p'_{alm}}$. In addition *F* is depending on geometric functions (cf. equation 2) that are depending on $\boldsymbol{p_{alm}}$. Considering all these continuously differentiable dependencies then it is possible to compute $\partial_{\mathbf{p}}F$ and $\partial_{\mathbf{p}'}F$. The latter generalized partial derivatives are built up by the partial derivatives of the force function *F(**p*** $F(\mathbf{p}_{alm}, \mathbf{p}'_{alm})$ (some vertex **p**, having velocity **p**´) with respect to the coordinates x,y,z of a space point and the partial derivatives of $F(\mathbf{p}_{alm}, \mathbf{p}'_{alm})$ with respect to the coordinates x`, y`, z` of the vertex speed vector **p**´. Computing the latter partial derivatives heavily employs the chain rule. In this context we obtain the partial derivatives of the stress functions σ_{ij} respective the relevant strain parameter ε_{ij} from the derivatives of spline curves approximating the function $\sigma_{ij}(\varepsilon_{ij})$, cf. A.2.

The matrix containing all those partial derivatives evaluated at $F(\mathbf{p}_{alm}, \mathbf{p}'_{alm})$

$$\left(\partial_x F \quad \partial_y F \quad \partial_z F \quad \partial_{x'} F \quad \partial_{y'} F \quad \partial_{z'} F\right) = \left(\partial_p F \quad \partial_{p'} F\right)$$

describes a linear mapping being the derivative (also called Jacobian) of the force vector $F(\mathbf{p}_{alm}, \mathbf{p}'_{alm})$ evaluated at a vertex point **p** moving with speed **p'**.

This function $F(\mathbf{p}_{alm}, \mathbf{p}'_{alm})$ assigns to $(\mathbf{p}_{alm}, \mathbf{p}'_{alm})$ the planar (tangential) vector describing the (tangential) force (cf. equation 7) that the mesh point with index *alm*, velocity $\mathbf{p}'_{\mathrm{alm}}$ and position $\mathbf{p}_{\mathrm{alm}}$ will experience via the stretch of the mesh.

The solution of the differential equation system belonging to the mesh vertex point $\mathbf{p}_{\mathrm{alm}}$ describes its trajectory $\mathbf{p}_{\mathrm{alm}}(\mathbf{t})$. As stated above the differential equations systems are coupled, thus all these systems have to be solved simultaneously. Let **n** be the number of vertices in the mesh. We could interpret the **n** trajectories as a single trajectory in 3**n**-dimensional space. Indeed this is how the systems are solved as described in section 3.3.3. In the latter setting we view the system of **n** coupled 6-dimensional first order differential equation systems as being one 6**n**-dimensional first order differential equation system.

A.2 Approximation of Strain-Stress Curves

We use polynomial spline approximations for the three independent strain-stress functions in the following form:

$$\sigma(\varepsilon) = A \cdot \varepsilon + \sum_i \sum_j B_{i,j} \cdot (\varepsilon - E_i)_+^j$$

This is a kind of truncated power basis spline with basis functions defined as follows:

$$(x - E_i)_+^j = (\max\{x - E_i, 0\})^j$$

A strain-stress measurement of a fabric results in a set of points $(\varepsilon_k, \hat{\sigma}(\varepsilon_k))$ which should be approximated by $\sigma(\varepsilon)$ minimizing the sum of squares error:

$$\sum_k (\hat{\sigma}(\varepsilon_k) - \sigma(\varepsilon_k))^2$$

We want to use the spline approximation also to approximate the derivative $\sigma'(\varepsilon)$. Therefore we require $\sigma(\varepsilon)$ to be continuously differentiable. This is easily achieved by requesting $B_{i,0} = B_{i,1} = 0$. Furthermore, we limit the order of the curve (j < 4) and the number of segments (i < 3) for optimizing computation speed.

We substitute $A = B_{0,1}, E_0 = 0$ leading to a simplified representation:

$$\sigma(\varepsilon) = \sum_i \sum_j B_{i,j} \cdot (\varepsilon - E_i)_+^j$$

The main problem is that we do not only have to compute the coefficients $B_{i,j}$ but also the knots $E = (E_1, E_2, \ldots)$. It is noticeable that for given knots E the coefficients $B_{i,j}$ are uniquely determined by the corresponding linear least squares problem leading to:

$$\sigma(\varepsilon) = \sum_i \sum_j B_{i,j}(E) \cdot (\varepsilon - E_i)_+^j$$

Thus, the knot vector E is the true independent variable which is computed by solving the remaining non-linear least squares problem with a numerical standard algorithm for minimizing functions.

References

[1] Astley, O. R., Hayward, V., “Multirate haptic simulation achieved by coupling finite element meshes through Norton equivalents”, Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on Volume 2, 16-20 May 1998 Page(s):989 - 994 vol.2

[2] Salsedo, F. et al., “Architectural Design of the HAPTEX System”; submitted to the Proceedings of this Conference.

[3] Hui, C. L., Lau, T. W., Ng, S. F., Chan, K. C. C., “Neural Network Prediction of Human Psychological Perceptions of Fabric Hand”, Textile Research Journal, May 2004.

[4] Mäkinen, M. et al., “Influence of physical parameters on fabric hand”; submitted to the Proceedings of this Conference.

[5] Large Steps in Cloth Simulation, Baraff D., Witkin A.; Computer Graphics 32, Annual Conference Series (1998), 43–54.

[6] Comparing efficiency of integration methods for cloth animation, Volino P., Magnenat-Thalmann N.; In Proceedings of the Conference on Computer Graphics International (CGI-01).

[7] Simulation of Clothes for Real-time Applications, Magnenat-Thalmann N., Cordier F., Keckeisen M., Kimmerle S., Klein R., Meseth J.; in: Proc. of Eurographics 2004, Tutorial 1, 2004.

[8] Stable but responsive cloth, Choi K., Ko H; In Proceedings of ACM SIGGRAPH (2002), pp. 604–611.

[9] Simulation of clothing with folds and wrinkles, Bridson R., Marino S., Fedkiw R.; In Proc. ACM/Eurographics Symposium on Computer Animation (2003), pp. 28–36.

[10] A high performance solver for the animation of deformable objects using advanced numerical methods, Hauth M., Etzmuss O.; In Proc. Eurographics 2001 (2001), Chalmers A., Rhyne T.-M., (Eds.), vol. 20(3) of Computer Graphics Forum, pp. 319–328.

[11] Untangling cloth, Baraff D., Witkin A., Kass M.; ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) 22, 3 (2003), 862–870.

[12] Implementing fast Cloth Simulation with Collision Response, Volino, P.; Magnenat-Thalmann, N; Computer Graphics International Proceedings, IEEE Computer Society, pp 257-266, 2000

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

[14] Eberhardt, P., Weber, A., Strasser, W., 1996. A Fast, Flexible, Particle-System Model for Cloth Draping, Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications), IEEE Press, pp 52-59, Sept. 1996.

[15] Sensing the fabric: To simulate sensation through sensory evaluation and in response to standard acceptable properties of specific materials when viewed as a digital image, P. Dillon, W. Moody, R. Bartlett, P. Scully, R. Morgan, C. James (2001). In Proceedings of Haptic Human-Computer Interaction, J. Lecture Notes in Computer Science. 2058, BT85B, 205 –217.

[16] Huang, G., Metaxas, D., and Govindaraj, M. 2003. Feel the "fabric": an audio-haptic interface. In *Proceedings of the 2003 ACM Siggraph/Eurographics Symposium on Computer Animation* (San Diego, California, July 26 - 27, 2003). Symposium on Computer Animation. Eurographics Association, Aire-la-Ville, Switzerland, 52-61.

[17] Haptic Simulation of Fabric Hand, M. Govindaraj, A. Garg, A. Raheja, G. Huang and D. Metaxas, Eurohaptics Conference 2003.

[18] Desbrun, M., Schröder, P., Barr, A., 1999. Interactive Animation of Structured Deformable Objects; Proceedings of Graphics Interface, A K Peters, 1999.

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

[20] Volino, P., Magnenat-Thalmann, N., 2005. Accurate Garment Prototyping and Simulation, Computer-Aided Design & Applications, 2(1-4), CAD Solutions.

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

[22] Metzger, J., Kimmerle, S., Etzmuss, O., 2003. Hierarchical Techniques in Collision Detection for Cloth Animation, Journal of WSCG, 11 (2) pp 322-329.

[23] Salisbury, J. K., Brock, D., Massie, T., Swarup, N., Zilles, C., Haptic rendering: Programming touch interaction with virtual objects. Proceedings of the ACM Symposium on Interactive 3D Graphics, 1995.

[24] Ho, C.-H., Basdogan, C., Srinivasan, M.A., Efficient Point-Based Rendering Techniques for Haptic Display of Virtual Objects, Presence 8(5), pp 477-491, 1999

[25] Morgenbesser, H.B., Srinivasan, Force Shading for Haptic Shape Perception. *Proceedings of the ASME Dynamic Systems and Control Division,* 58, 407-412., MA, 1996

[26] Oshita, M., Makinouchi, A. Real-time Cloth Simulation with Sparse Particles and Curved Surfaces, Proceedings of Computer Animation, 2001

[27] Zhang, J., Payandeh, S., and Dill, J. Haptic subdivision: an approach to defining level-of-detail in haptic rendering. *Proceedings. 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp 201-208, March 2002.