A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation

Hervé Delingette, Stéphane Cotin, Nicholas Ayache
Projet Epidaure
INRIA Sophia-Antipolis
2004 Route des Lucioles
06902 Sophia-Antipolis, France

Abstract

In this paper, we describe the basic components of a surgery simulator prototype developed at INRIA. After a short presentation of the geometric modeling of anatomical structures from medical images, we insist on the physical modeling components which must allow realistic interaction with surgical instruments. We present three physical models which are well suited for surgery simulation. Those models are based on linear elasticity theory and finite elements modeling. The first model pre-computes the deformations and forces applied on a finite element model, therefore allowing the deformation of large structures in real-time. Unfortunately, it does not allow any topology change of the mesh therefore forbids the simulation of cutting during surgery. The second physical model is based on a dynamic law of motion and allows to simulate cutting and tearing. We called this model "tensor-mass" since it is analogous to spring-mass models for linear elasticity. This model allows volumetric deformations and cuttings, but has to be applied to a limited number of nodes to run in real-time. Finally, we propose a method for combining those two approaches into a hybrid model which may allow real time deformations and cuttings of large enough anatomical structures. This model has been implemented in a simulation system and real-time experiments are described and illustrated.

1. Introduction

The most recent major advance in the craft of surgery was the development of laparoscopic surgery. In this type of surgery, abdominal operations such as hepatic surgery are accomplished through small incisions rather than a large one that might be a foot long. The abdomen is blown up with gas so that there is open space inside. A video camera is introduced into the abdomen through one of the small incisions. The video image is magnified and transmitted to a high resolution monitor, allowing the surgeon to see the abdominal anatomy with great clarity. The surgical operation is then performed inside the abdomen using long and narrow scissors and clamps that are introduced through the other incisions. Thus, laparoscopic surgery allows surgeons to perform less traumatizing operations, the drawback of this technique being essentially for the surgeon who needs to learn and adapt himself to this new type of surgery. In this context, surgical simulation systems could be a great help in the training process.

There are several key problems in the development of a surgical simulator [2]. First of all, a model of the target organ(s) is required. This model should define both geometrical and physical characteristics of the organ(s). The geometry is usually obtained from various medical images modalities, while the deformable nature of the soft tissues are determined – when it is possible – through biomechanical studies. However, the computation of the shape and deformable behavior of an organ is not sufficient. Another very important requirement in surgery simulation concerns real-time interaction. Real-time interaction requires that any action from the operator generates an instantaneous response from the stimulated organ, whatever the complexity of its geometry. It means that we must be able to interactively deform or cut a virtual organ and eventually feel its reaction in real-time by the introduction of force feedback devices. A good balance between surgical realism and interactive rates of simulation is one of the most challenging problems in surgical simulation.

After a presentation of different techniques used to represent deformable objects, we will focus on the following problems: real-time computation methods and topology change. Both problems will be discussed in the framework of linear elasticity theory. After a short recall of a previously published [14] pre-computed real-time linear elastic model, we introduce a formulation of an elastic model

allowing topology changes. Finally, we present a hybrid model combining both approaches and we conclude with an application on hepatic surgery.

2. Deformable models

Terzopoulos [32], Waters [34] and Platt [28] showed the advantages of the physically-based models on previous computer animation techniques. In surgery simulation, a great interest has been given to mass-spring models due to their simplicity of implementation and their relatively low computational complexity [6, 27, 24, 25]. For instance, Kuehnapfel and Neisius [24] present a simulation of endoscopic surgery based on a surface mass-spring model. Although in this case the interactions are driven by instruments with motion sensors, no force feedback is used. The simulation system developed by Gibson *et al.* [20] takes into account the volumetric nature of organs with a deformation law derived from a mass-spring model. While these approaches allow interactive rates, they exhibit a lack of realism since they represent a solid as a discrete set of masses.

On the other hand, the continuum mechanics theory establishes a set of relationships between the shape of a body, constraints and internal deformations within this solid and the external forces applied to it. In particular, the use of elastic solids is widely described in the literature [4, 30, 11, 31]. Generally, various modifications and simplifications – e.g. linearisation – are introduced to reduce computation time or to induce a particular behavior, as it is usually the case in surgical simulation. Indeed, non-linear elastic models have been used recently for solving inverse problems [33], but they are not well suited for real-time computation. Actually, whatever the approach taken, it usually reflects a need for defining compromises between medical realism and real-time constraints. For instance, Bainville [4] defines the evolution of a set of rigid and deformable solids under the influence of various forces. In this case, the deformation law is represented by a hyper-elastic quasi-static model, associated to a finite element method for the numerical resolution. Unfortunately, the computation times makes this approach unpractical for real-time surgical simulation.

Our method is, to some extent, based on the opposite reasoning. We will start with a simple model - in order to obtain optimal computation times - and then we will propose some improvement in order to take into account additional characteristics and medical aspects. Since we don't know the exact nature of the deformation law of most soft tissues, we have based our method on a linear elastic modeling of the soft tissues. Such a model is indeed a good approximation of a large number of materials when the range of displacements remains small.

The problem of computing time reduction has been studied by Bro-Nielsen and Cotin [9] using a condensation technique [35] applied to a finite element method. With such an approach, the computation time required for the deformation of a volumetric model can be reduced to the computation time of a model only involving the surface nodes of the mesh. In [14] we have also proposed a method for real-time interaction with a volumetric deformable model of an organ. This method, based on a set of pre-computed equilibrium solutions is very efficient while no topology changes are performed.

But topology changes usually occur when the tissue model is submitted to cutting or tearing operations. These changes make useless the methods based on any kind of pre-computing of the inverse of the stiffness matrix since the mesh modification induces a modification of this matrix [13]. The challenge is then to be able to propose a formulation of a deformable elastic model allowing real-time deformations *and* cutting. A few number of researchers have focused on tissue cutting in the framework of surgery simulation. Song and Reddy [30] have described a technique for cutting linear elastic objects defined as finite element models. However, this technique was only applied to very simple two dimensional objects. Another approach, well suited to this kind of problem, consists in using mass-spring models. By construction, these models can be easily modified to allow topology changes. However, they exhibit non realistic behavior during deformation or cutting operation. Finally, geometric models defined implicitely [10] have been proposed for the simulation of tearing but without any real-time constraints.

3. Linear elasticity

A key issue that a surgical simulator must address is to realistically model the behaviour of soft tissues in real-time. A survey of soft tissue modeling can be found in [17]. However, we briefly describe the simplest biomechanical model: the *linear elastic* model.

The physical behavior of a soft tissue model may be considered as linear elastic if the displacements applied to it remain small [19, 26] (less than 10% of the mesh size); as the displacements increase, the linear elastic approximation becomes less and less valid. In particular, several biological materials are nearly incompressible since they are mainly comprised of water. Such behavior cannot be modeled with linear elasticity, the integration of force feedback in the simulation limits the range of deformations to small deformations. This is because the force in the surgeon's hand will increase as he increases the deformation, thus preventing large deformations. Consequently, the deformation remains reasonably small. Another interest of linear elasticity is the possibility to compute any mesh deformation from the knowledge of a finite set of elementary deformations, as we

will see in the next section.

First, we define a reference volumetric anatomical model $\mathcal{M}_{\text{initial}}$ corresponding to its rest position. Under external constraints, for instance a surgical instrument, the anatomical model $\mathcal{M}_{\text{initial}}$ is deformed. We represent the deformation of a volumetric model from its rest shape with a *displacement vector* $\mathbf{U}(x, y, z)$ for $(x, y, z) \in \mathcal{M}_{\text{initial}}$ and we write $\mathcal{M}_{\text{deformed}} = \mathcal{M}_{\text{initial}} + \mathbf{U}(x, y, z)$. The displacement vector $\mathbf{U}(x, y, z)$ has three components:

$$\mathbf{U}(x, y, z) = \begin{cases} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{cases}$$

The displacement vector $\mathbf{U}(x, y, z)$ does not characterize the deformation of the anatomical model. For instance, under a translation $\mathbf{T}$ of the model $\mathcal{M}$, the displacement vector is $\mathbf{U}(x, y, z) = \mathbf{T}$, but the model does not yield any deformation. For a linear elastic material, the elastic energy $W_{Elastic}$ measuring the amount of deformation of $\mathcal{M}_{\text{deformed}}$, is defined as [12]:

$$W_{\text{Elastic}} = \frac{\lambda}{2}(trE)^2 + \mu trE^2 \tag{1}$$

where

- the 3×3 symmetric matrix E (known as the Green-St Venant strain tensor) is defined as :

$$E = \frac{1}{2}(\nabla \mathbf{U} + \nabla \mathbf{U}^T) \tag{2}$$

- λ and μ are the Lamé coefficients characterizing the stiffness of a material.

Equation 1, known as *Hooke's law*, shows that the elastic energy of a deformable object is a quadratic function of the displacement vector.

In the following sections, we will consider the framework of finite elements and assume that the object is represented by a conformal tetrahedral mesh. It is then possible to compute at each node i a force $\mathbf{F}_i$ corresponding to the derivation of the elastic energy with respect to the node position $\mathbf{P}_i$:

$$\mathbf{F}_i = \frac{\partial W_{\text{Elastic}}}{\partial \mathbf{P}_i} \tag{3}$$

Because the elastic energy is quadratic with respect to the displacement vector, the forces $\mathbf{F}_i$ are linear functions of the displacement vectors of each node $\mathbf{P}_j$. Therefore, it can be shown that minimizing the elastic energy of a three-dimensional object requires the solution of a linear system of the form :

$$[\mathbf{K}]\mathbf{u} = \mathbf{f}$$

where :

- $[\mathbf{K}]$ is the *rigidity* matrix representing the topology and stiffness of the discretized object
- $\mathbf{u}$ represents the displacement of all nodes
- $\mathbf{f}$ combines all external forces and boundary conditions.

In sections 4 and 5, we study successively a *quasi-static* and a *dynamic* model to compute the deformation of an elastic object and the resulting forces. Finally, we show in section 6 an hybrid approach which combines the advantages of each model for surgery simulation.

4. Quasi-static pre-computed linear elastic model

As stated in the introduction, surgical simulation requires visual feedback and, eventually, force feedback, i.e. an update frequency of about 60Hz for the display and 300Hz for the forces. When solving a problem of linear elasticity with a finite element method, the number of mesh vertices has a direct impact on the size of the matrices involved in the linear system $[\mathbf{K}]\mathbf{u} = \mathbf{f}$. This implies that even using more powerful computers, only deformable models with a small number of vertices could be simulated. However, most anatomical structures have a rather complex geometry and cannot be realistically described with such a limited number of vertices.

In order to speed up the interaction rate, we take advantage of the following properties: the linearity and the superposition principle. We give here a very general description of the method, an extensive description can be found in [14]. We first introduce a volumetric deformable model with the following properties:

1. The model follows the linear elastic biomechanical model.
2. This model deforms under some boundary conditions expressed in terms of imposed displacements and forces.
3. The model evolves in a quasi-static state: the position of the model at time $t + 1$ is the solution of the static problem with boundary conditions given at time t. This assumption of quasi static evolution, made in many situations [3, 23, 7, 21], considers as negligeable the effect of the acceleration and speed in the computation of the deformation. This assumption has several advantages, in particular a simplification of the problem to be solved but also a suppression of the oscillations in the vicinity of the equilibrium as well as a reduction of the complexity of the processing of the contacts between objects.

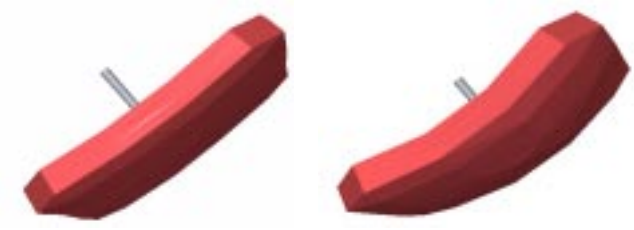

Figure 1. Deformation of plate represented as a pre-computed linear elastic model : (left) initial position (right) deformed position

Then, interactive rates of deformation can be obtained in a two-step method:

1. A pre-processing stage is required in order to to compute a set of elementary deformations of the model.

2. The deformation can be computed in real-time as a linear combination of elementary deformations. The computation time is a linear function of the number of *surface* nodes.

We use an iterative method (conjugate gradient) to solve each linear system. During the simulation, very limited computations are performed to get the exact deformation of the anatomical object. An update rate of 500Hz has been reached with a mesh having nearly 8000 tetrahedra. We have developed an hepatic surgery simulator [15, 1] including a force feedback device, based on this linear elastic soft tissue model.

The pre-processing stage can take between a few minutes and several hours depending on the size of the model and on the desired precision. This is not a problem since this preprocessing is done once for all.

5. Dynamic linear elastic model or *tensor-mass* system

5.1. Motivations

The main drawback of the previous approach is the impossibility to change the topology of the mesh in real-time since it induces a modification of the stiffness matrix and by the way requires a new pre-processing stage. Moreover, without pre-processing, it is no longer possible to solve exactly and in real-time the equation $[\mathbf{K}]\mathbf{u} = \mathbf{f}$ with a time-varying matrix $[\mathbf{K}]$, even with meshes of a few hundred nodes.

An alternative we propose in this section is to consider the evolution of a physical dynamic model which can be approximately solved by an efficient real-time iterative approach. This approach allows to deal with meshes of reasonable size (more than 1000 nodes for a frame-rate simulation), and provides a good approximation to the exact static equation.

The physical model is based on the Newtonian law of motion of each mesh point $\mathbf{P}_i$:

$$m_i \frac{d^2\mathbf{P}_i}{dt^2} = \gamma_i \frac{d\mathbf{P}_i}{dt} + \mathbf{F}_i \tag{4}$$

where $\mathbf{F}_i$ is obtained from the derivation of the elastic energy given by equation 3.

In the sequel of this section, we present the computation of $\mathbf{F}_i$ within the framework of finite element modeling and the numerical integration scheme. We then compare our approach with the spring-mass formulation, and present simulation results.

5.2. Definition of a dynamic linear elastic finite element model

We assume that we have a conformal tetrahedral mesh (as defined in finite element theory) describing the geometry of the considered anatomical structure. We denote $\mathcal{M}_{\text{initial}}$ the mesh at its rest position and $\mathbf{P}_i^0$ the initial position of each vertex. We then proceed in 3 steps :

1. We first define the interpolation equations giving the displacement vector at a point (x, y, z) inside a tetrahedron T_i as a function of the 4 displacement vectors of the 4 vertices of T_i.

2. We write the elastic energy of a tetrahedron as a function of the 4 displacement vectors of the 4 vertices of T_i.

3. We compute the elastic force $\mathbf{F}_i$ acting on a vertex $\mathbf{P}_i$

5.2.1 Displacement vector equation

Given a deformed model $\mathcal{M}_{\text{deformed}}$, we define the *displacement vector* for each point of the mesh by linearly interpolating the displacement $\mathbf{P}_i^0\mathbf{P}_i$ of the vertices inside each tetrahedron.

More precisely, if we write T_i the tetrahedron defined by the four vertices $T_i(j)$, $j = 0, \ldots, 3$, in their **rest position**, then the vector displacement is defined as :

$$\mathbf{U}_{T_i}(x, y, z) = \sum_{j=0}^{3} b_j(x, y, z)\mathbf{P}^0_{T_i(j)}\mathbf{P}_{T_i(j)}$$

where $b_j(x, y, z)$ are the barycentric coordinates of the point (x, y, z) inside the tetrahedron T_i. Since the barycentric coordinates $b_j(x, y, z)$ are linear with respect to the coordinates (x, y, z), the matrix $E(x, y, z)$, defined in equation 2, is constant for each point of the tetrahedron T_i since

it is related to the derivatives of the displacement vector. Appendix **??** provides more details on the formulation of the shape functions $b_j(x, y, z)$.

5.2.2 Elastic energy

If we associate with each tetrahedron T_i its linear elastic properties, i.e. the two Lamé coefficients λ_i and μ_i, then using equation (1), we can express the elastic energy $W_{Elastic}(T_i)$ of the tetrahedron T_i as a quadratic function of the coordinates of $\{\mathbf{P}_{T_i(j)}\}$ (see appendix **??** for more details). The total elastic energy $W_{\mathrm{Elastic}}(\mathcal{M}_{\mathrm{deformed}})$ required to deform $\mathcal{M}_{\mathrm{initial}}$ into to $\mathcal{M}_{\mathrm{deformed}}$ is the sum of the elastic energy of each tetrahedron.

5.2.3 Linear Elastic Force

Given the expression of the elastic energy, we derive the force $\mathbf{F}_i$ applied on a vertex $\mathbf{P}_i$ with the relation:

$$\mathbf{F}_i = -\frac{\partial W_{Elastic}(\mathcal{M}_{\mathrm{deformed}})}{\partial \mathbf{P}_i} = \sum_{T_j \in L(i)} -\frac{\partial W_{Elastic}(T_j)}{\partial \mathbf{P}_i}$$

where $L(i)$ is the set of tetrahedra adjacent to vertex $\mathbf{P}_i$.

Within the tetrahedron T_i, the force $\mathbf{F}_{T_i(j)}$ applied on vertex $\mathbf{P}_{T_j(j)}$ takes the following form:

$$\mathbf{F}_{T_j(j)} = \sum_{k=0}^{3} [\mathbf{K}_{\mathbf{jk}}^{\mathbf{T_i}}] \mathbf{P}^0_{T_i(k)} \mathbf{P}_{T_i(k)}$$

where $[\mathbf{K}_{\mathbf{jk}}^{\mathbf{T_i}}]$ are 3×3 stiffness matrices that can be computed as follow:

Given a tetrahedron T_i and its four vertices $\mathbf{P}^0_{T_i(0)}, \mathbf{P}^0_{T_i(1)}, \mathbf{P}^0_{T_i(2)}, \mathbf{P}^0_{T_i(3)}$, we compute the six tensors $[\mathbf{K}_{\mathbf{jk}}^{\mathbf{T_i}}]$ as follows:

$$[\mathbf{K}_{\mathbf{jk}}^{\mathbf{T_i}}] = \frac{\lambda_i \mathbf{M}_k \mathbf{M}_j^T + \mu_i \mathbf{M}_j \mathbf{M}_k^T + \mu_i (\mathbf{M}_j \mathbf{M}_k) [Id_{3\times 3}]}{36 V(T_i)} \tag{5}$$

with

$$[Id_{3\times 3}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Moreover, equation (5) induces the following relationship: $[\mathbf{K}_{\mathbf{kj}}^{\mathbf{T_i}}] = [\mathbf{K}_{\mathbf{jk}}^{\mathbf{T_i}}]^T$

It is important to notice that these stiffness matrices only depend on the material characteristics within a tetrahedron – through the Lamé coefficients λ_i and μ_i – and the geometry of the tetrahedron T_i at its rest shape.

For a given vertex $\mathbf{P}_i$, the elastic force $\mathbf{F}_i$ is therefore the sum of all contributions $\mathbf{F}_{T_i(j)}$ from all adjacent tetrahedron T_i. We can therefore express the elastic force $\mathbf{F}_i$ as:

$$\mathbf{F}_i = [\mathbf{K}_{\mathbf{ii}}] \mathbf{P}_i^0 \mathbf{P}_i + \sum_{j \in N(\mathbf{P}_i)} [\mathbf{K}_{\mathbf{ij}}] \mathbf{P}_j^0 \mathbf{P}_j \tag{6}$$

where $[\mathbf{K}_{\mathbf{ii}}]$ is the sum of tensors $[\mathbf{K}_{\mathbf{ii}}^{\mathbf{T_j}}]$ associated to the tetrahedra adjacent to $\mathbf{P}_i$, $[\mathbf{K}_{\mathbf{ij}}]$ is the sum of tensors $[\mathbf{K}_{\mathbf{ij}}^{\mathbf{T_j}}]$ associated to the tetrahedra adjacent to edge (i, j) and $N(\mathbf{P}_i)$ is the list of $\mathbf{P}_i$ neighbors.

5.3. Data structure

Given a tetrahedral mesh of a solid – in our case an anatomical structure – we build a data structure incorporating the notion of vertices, edges and tetrahedra. For each vertex, we store *the adjacency with tetrahedra*, the current position $\mathbf{P}_i$, the rest position $\mathbf{P}_i^0$ and the tensor $[\mathbf{K}_{\mathbf{ii}}]$. For each edge, we store the two adjacent vertices as well as the tensor $[\mathbf{K}_{\mathbf{ij}}]$. Finally for each tetrahedron, we store a reference to its four vertices and its six edges as well as the Lamé coefficients λ_i, μ_i and the four vectors $\mathbf{M}_i$ defined in equation **??** (see appendix **??**).

5.4. Numerical integration

We use a Newtonian differential equation (4) as the equation governing the motion of our linear elastic model. This equation is related to the differential equation found in continuum mechanics [5] :

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \tag{7}$$

Following finite elements theory, the mass $\mathbf{M}$ and damping $\mathbf{C}$ matrices are sparse matrices that are related to the stored physical properties of each tetrahedron. In our case, we consider that $\mathbf{M}$ and $\mathbf{C}$ are diagonal matrices, i.e. that mass and damping effects are concentrated at vertices. This simplification called *mass-lumping* decouples the motion of all nodes and therefore allows to write equation 7 as the set of independent differential equations (4) for each vertex.

Furthermore, we choose an *explicit integration scheme* where the elastic force is estimated at time t in order to compute the vertex position at time $t + 1$:

$$\left(\frac{m_i}{\Delta t^2} - \frac{\gamma_i}{2\Delta t}\right) \mathbf{P}_i^{t+1} = \mathbf{F}_i + \frac{2m_i}{\Delta t^2} \mathbf{P}_i^t - \left(\frac{m_i}{\Delta t^2} + \frac{\gamma_i}{2\Delta t}\right) \mathbf{P}_i^{t-1} \tag{8}$$

The key advantage of this explicit integration scheme is that no rigidity matrix inversion is required for updating each vertex. Therefore, after modifying the mesh topology, equation 8 is used to update the vertex position without any additional computation apart from the update of local tensors $[K_{ii}]$ (see section 5.5). On the contrary, using an implicit integration scheme would have entailed the inversion of the

rigidity matrix every time the mesh topology is altered. A discussion about explicit versus implicit schemes has been also proposed in [8].

Explicit schemes are only *conditionnaly stable* are therefore they tend to converge more slowly than implicit schemes. In order to obtain a stable scheme, the time step used for the numerical integration must be small enough. The critical time step is related to the highest eigenvalue of the rigidity matrix and the local mass and damping values (see [5] for more details). To optimize the time step, we use constant values of the mass and damping values. Indeed, in [8], Bro-Nielsen proposed to use mass and damping values proportional to the volume of each tetrahedra. This approach leads to ill-conditionned iterative scheme (with low time-step) when dealing with meshes having tetrahedra of different size. In our case, the choice of the time-step is only related to the stiffness of the soft tissue material independently of the elements size.

Finally, we use a fourth order Runge-Kutta method [29] for discretizing the time domain instead of the Euler Method of equation 8. This method requires to evaluate four times the forces applied to P_i in order to compute the next position $P_i(t+1)$. However, we have compared this approach with the Euler method which only requires one evaluation of forces. Our conclusion is that the Runge-Kutta method allows the use of larger time steps – about ten times larger – than the Euler method. Consequently, the fourth order Runge-Kutta method leads to a speed-up factor of about two. This is particularly interesting when deforming stiff material requiring small time steps.

5.5. Simulation of cutting and tearing

One of the basic task in surgery simulation consists in cutting and tearing soft tissue. With the dynamic linear elastic model, those task can be achieved in real-time.

We simulate the action of an electric scalpel – a bipolar cauttery instrument – on soft tissue by successively removing tetrahedra at places where the instrument is in contact with the anatomical model. This approach implies that for realistic simulation, the tetrahedra must be relatively small at the regions where the cutting may occur. Furthermore, in order to keep the mesh conformal, additional tetrahedra may be automatically removed after checking the local vertex and edge adjacency.

When a collision between the instrument and a tetrahedron is detected, the local deformation tensors associated with the tetrahedron are computed and then subtracted to the current deformation tensors at the edges and vertices of the tetrahedron. Since the update of the tensors is only local, this is performed in a very efficient manner. For instance, when removing the tetrahedron T_i, ten update operations are performed :

$$[K_{jj}] = [K_{jj}] - [K_{jj}^{T_i}] \qquad [K_{jk}] = [K_{jk}] - [K_{jk}^{T_i}]$$

Finally, we update the list of displayed triangles if the tetrahedron is located at the border of the volumetric model. By locally updating the tensors, the tissue has exactly the same behavior as if we had removed the corresponding tetrahedron at its rest position. Because of the volumetric continuity of finite element modeling, the deformation of the tissue remains very natural during the cutting.

In addition to cutting, we can simulate the tearing of soft tissue. The tearing occurs at places where normal stress and shearing are too high. The basic algorithm is to compute a local deformation criterion measure for each tetrahedron. If this criterion is greater than a threshold, then the tetrahedron is automatically removed and the tensors are locally updated. We have implemented three geometric criteria for detecting highly deformed tetrahedra. Those criteria are the relative variation of volume, the mean relative elongation of the six edges and the maximum relative elongation of the edges.

5.6. Spring-Mass model versus tensor-mass model

In a classical approach, a vertex $\mathbf{P}_i$ in a spring-mass system, is submitted to an elastic force:

$$\mathbf{F}_i = \sum_{j \in N(\mathbf{P}_i)} k_{ij}(\|\mathbf{P}_i\mathbf{P}_j\| - l_{ij}^0)\frac{\mathbf{P}_i\mathbf{P}_j}{\|\mathbf{P}_i\mathbf{P}_j\|} \qquad (9)$$

where $N(\mathbf{P}_i)$ is the set of vertices $\mathbf{P}_j$ adjacent to $\mathbf{P}_i$, k_{ij} is the stiffness coefficient between vertices $\mathbf{P}_i$ and $\mathbf{P}_j$, l_{ij}^0 is the rest length between $\mathbf{P}_i$ and $\mathbf{P}_j$.

We have shown that a vertex in our dynamic linear elastic model is submitted to the force of equation 6. Because of the similarity of the two approaches, we have coined the word **Tensor-mass** in order to describe this dynamic linear elastic model.

By comparing equations 6 and 9, it is clear that both dynamic models have the same computational complexity which is linear in the number of edges. In practice, we were able to reach an update frequency of 40 Hz with a tetrahedral mesh having 760 vertices and about 4000 edges[1] and similar results were obtained for a spring-mass system. The linear elastic model does not require any square root evaluation, but slightly more information must be pre-computed than for spring-mass systems.

However, both approaches substantially differ in terms of biomechanical modeling. Spring-mass systems constitute a discrete representation of an object and their behavior strongly depend on the topology of the spring network.

[1] These computing times where obtained on a 233 Mhz DEC Alphastation.

When a spring is removed or added it may change drastically the elastic behavior of the whole system.

Conversely, our finite elastic model is a continuous representation of the object and its behavior is independent of the mesh topology (it mostly depends on the mesh resolution). This implies that when the mesh is cut, continuous and natural behavior of the tissue is simulated.

Because all biomechanical data related to biological soft tissue are formulated as parameters of the continuum mechanics (such as Young modulus or Poisson coefficients), it is *a priori* very difficult to model realistic soft tissue deformations with a spring-mass system. However, several authors [22, 18] have developed genetic or simulated annealing algorithms to identify springs parameters (stiffness and damping) given the deformation of an object.

Finally as previously mentioned, the linear elastic model is only valid for small displacements. For instance, if a rigid transformation is applied to the rest shape $\mathcal{M}_{initial}$, then the forces applied to all vertices will not be null. On the contrary, a spring model under the same displacement would not deform, since the length of the springs are preserved under a rigid transformation.

We summarize the comparison between those three soft tissue models in table 1.

	Pre-computed	Tensor-Mass	Spring-Mass
Efficiency	+++	+	+
Realism	+	+	-
Cutting Simulation	-	++	+
Large Displacements	-	-	+

Table 1. Comparison between the three soft tissue models : pre-computed quasi-static, tensor-mass and spring-mass models

5.7. Examples

We present two examples of tensor-mass deformation. In figure 2 we show a volumetric plate being cut and fixed at its four corners. The cutting consists in removing interactively tetrahedra. As more and more tetrahedra are removed, one can see that the deformation of the overall plate is modified. The plate model consists of 759 vertices and 2212 tetrahedra.

In figure 3, we show a cylinder that is deformed under the action of gravity. Based on the relative change of volume, the tearing algorithm removes automatically the tetrahedra where the relative change of volume becomes greater than a given threshold. The cylinder consists of 248 vertices and 889 tetrahedra.

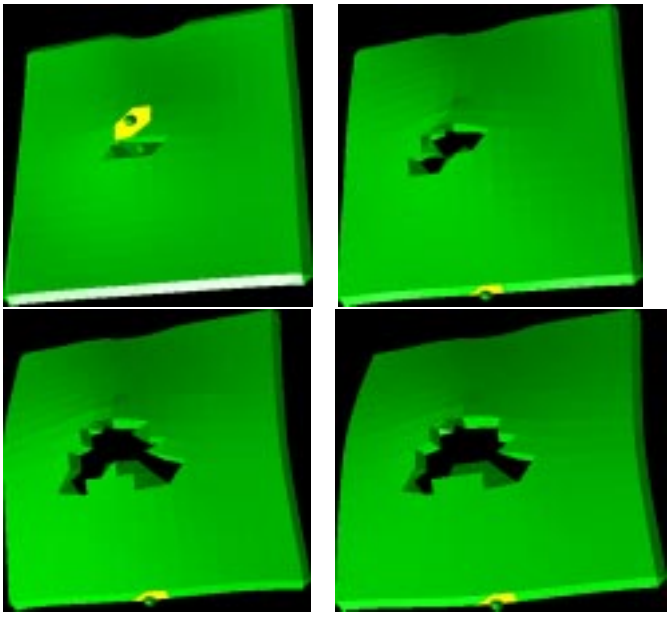

Figure 2. Cutting of a plate by removing tetrahedra. The plate deforms itself during the cutting operations.

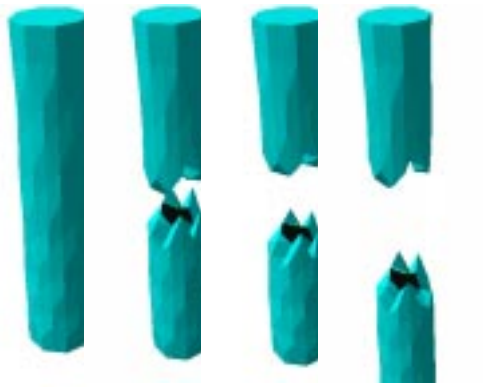

Figure 3. Fracture of a cylinder under the action of gravity

6. Hybrid elastic model for surgery simulation

6.1. Motivations

We have previously described two *linear elastic models* that have the following properties :

1. The *quasi-static* pre-computed elastic model is extremely efficient but does not allow topology change (cutting, tearing) (see section 4).

2. The *dynamic* elastic model (or tensor-mass model) requires more computation but authorizes topology change (see section 5).

In this section, we propose to combine those two approaches in order to optimize the trade-off between computation time and visual realism of the simulation.

The key idea consists in considering two different types of anatomical models inside a surgical simulator :

- The anatomical structures where the surgery occurs. On these structures, tearing and cutting need to be simulated. In many cases, surgical procedures are standardized and therefore, it is possible to foresee where the surgery should occur. In general, it corresponds to pathological structures and only represent a small subset of the structures that need to be visualized inside a simulator.

- The anatomical structures that only needs to be visualized or deformed that greatly contribute to the realism of the simulation but where surgery does not occur.

The former type of anatomical structures are good candidates for being modeled as tensor-mass models whereas the latter should be modeled with a pre-computed linear model.

However, this approach is really optimal when different parts of the same anatomical structure can be modeled either as a pre-computed elastic model or a tensor-mass model. This allows to decrease the number of tensor-mass elements to its minimum and therefore to increase the interactivity of the overall simulation.

In the next sections, we describe how to connect those two linear elastic models in order to represent a global deformable object. We have called *hybrid elastic model* the combined model.

6.2. Hybrid elastic model

We consider a hybrid elastic model $\mathcal{M}_{hybrid}$ whose elements are of two different types. We denote as $\mathcal{M}_{dynamic}$ the set of tensor-mass elements and we denote as $\mathcal{M}_{quasi-static}$ the set of pre-computed linear elastic elements. Therefore a hybrid elastic model may be composed

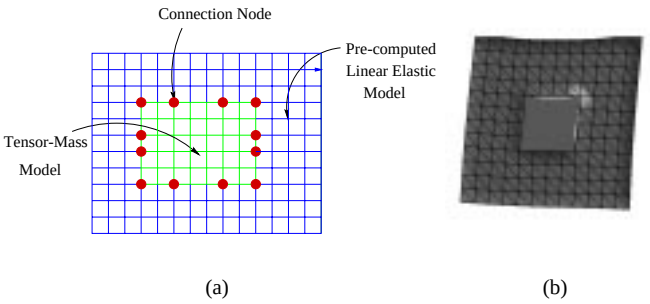

Figure 4. (a) The definition of the connection nodes in a hybrid elastic model; (b) A hybrid elastic model with only 4 connection nodes.

of several pieces of tensor-mass models each corresponding to a structure directly involved in surgery. Since $\mathcal{M}_{dynamic}$ is connected to $\mathcal{M}_{quasi-static}$ they share common vertices or *connection nodes*. Those connection nodes constitute additional boundary conditions for both elastic models. As seen in figure 6.2, the two models may not be completely connected along their common boundaries. In fact, in order to reduce the number of tensor-mass elements, it is possible to associate a fine pre-computed elastic model with a coarse tensor-mass model. As shown in figure 6.2 (b), this incomplete connection entails some visual artifacts due to the non-continuity between the two parts. However, if the connection part between the two elastic models is never seen by the user, a different mesh resolution may be used.

Since both linear elastic models follow the same physical law, the combination of those 2 models should behave exactly as a global linear elastic model. To achieve this goal, the additional boundary conditions imposed at the connection nodes must be consistent in terms of forces and displacements for both elastic models.

Boundary Conditions	Pre-computed	Tensor-Mass Model
Applied Forces	No	Yes
Constrained displacement	Yes	Yes

Table 2. Natural boundary conditions for the quasi-static and dynamic linear elastic models

We have shortly summarized the "natural" boundary conditions for both elastic models in table 2. Since the pre-computed model only supports displacement boundary conditions, $\mathcal{M}_{dynamic}$ must impose the displacements of the connection nodes whereas $\mathcal{M}_{quasi-static}$ imposes the

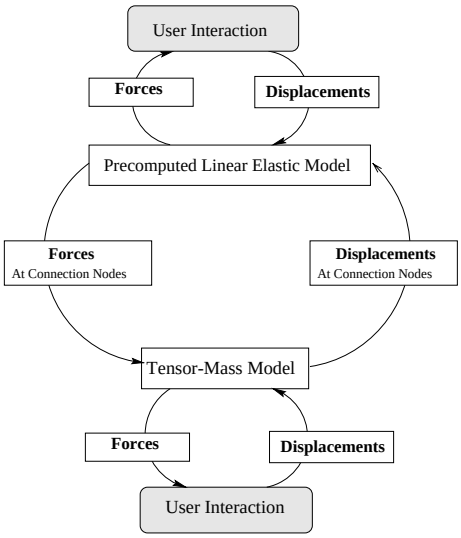

Figure 5. The interaction loop for a hybrid elastic model.

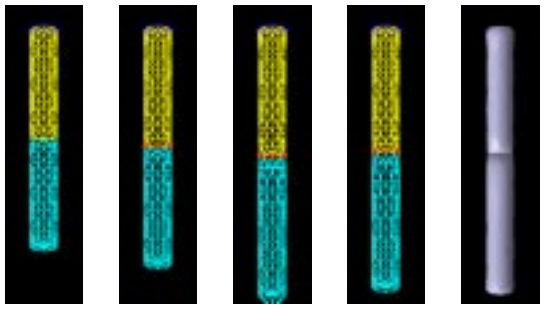

Figure 6. Deformation of a hybrid elastic model under gravity forces : the upper cylinder consists of a pre-computed linear elastic model whereas the lower part consists of a tensor-mass model. The leftmost figure corresponds to the initial position of the mesh and the rightmost figure to the steady state.

force acting on the connection nodes.

Figure 5 summarizes the computation loop of a hybrid model. The pre-computed elastic model $\mathcal{M}_{quasi-static}$ is updated based on the imposed displacements on its boundary. The imposed displacement may come from the user interaction (action of a surgical tool for instance) and from the connection nodes with a tensor-mass model. At this stage, the displacement of all surface nodes and the force applied on nodes where a displacement has been imposed are quickly computed on $\mathcal{M}_{quasi-static}$. In particular, the forces applied to each connection node are computed.

After updating $\mathcal{M}_{\mathrm{quasi-static}}$, we update $\mathcal{M}_{\mathrm{dynamic}}$ based on the forces imposed on the connection nodes and the displacements imposed by the user interaction. The position and forces are computed for all nodes including the connection nodes. Again, the new positions of the connection nodes constitute new displacement constraints for $\mathcal{M}_{quasi-static}$.

In figure 6, we show an example of a hybrid cylinder model deforming under the action of gravity forces. The figure shows the different steps of deformation and, on the right, the steady state reached by the model. At the equilibrium the forces applied to all connection nodes are zero and the displacement vector stabilizes to a constant value.

In this example, both quasi-static and dynamic models have exactly the same Lamé coefficients. We have verified that the steady state reached by the hybrid model is the same as the steady state that would have reached a single quasi-static or dynamic elastic model.

6.3. Surgery simulation on hybrid elastic models

We now demonstrate the efficiency of our approach for performing surgery simulation. We have chosen to simulate an hepatectomy consisting in removing an anatomical segment of a liver (in this case the segment number 6 as defined by Couinaud[16]) . The complete mesh of the liver contains 1537 vertices and 7039 tetrahedra (see figure 7 right). About 18% (280 vertices and 1260 tetrahedra) of the liver hybrid mesh is modeled as tensor-mass system (see figure 7 left) and the remaining as pre-computed linear elastic model.

The surgery simulator consists of two force-feedback systems simulating the elongated surgical instruments used in laparoscopy. Those force-feedback devices are driven by a PC computer that is linked to a powerful computer (SGI Onyx2 Infinite Reality with 2 processors). On that computer, the collision detection is performed as well as the animation of the hybrid elastic model.

In figure 8, we show different stages of the hepatectomy simulation. With those two virtual instruments, it is possible to push any part of the liver (either represented with a quasi-static or dynamic model) and to cut any elements of the dynamic model located in the bottom part of the screen. The first six pictures show the deformation of the model when the tool collides with the dynamic model. Since both models have the same elastic characteristics, it is not possible to visually distinguish the interface between the two different elastic models. The last six pictures show the cutting of the liver segment by removing additional tetrahedra. One can notice that each part of the hybrid model naturally deforms itself during the resection simulation.

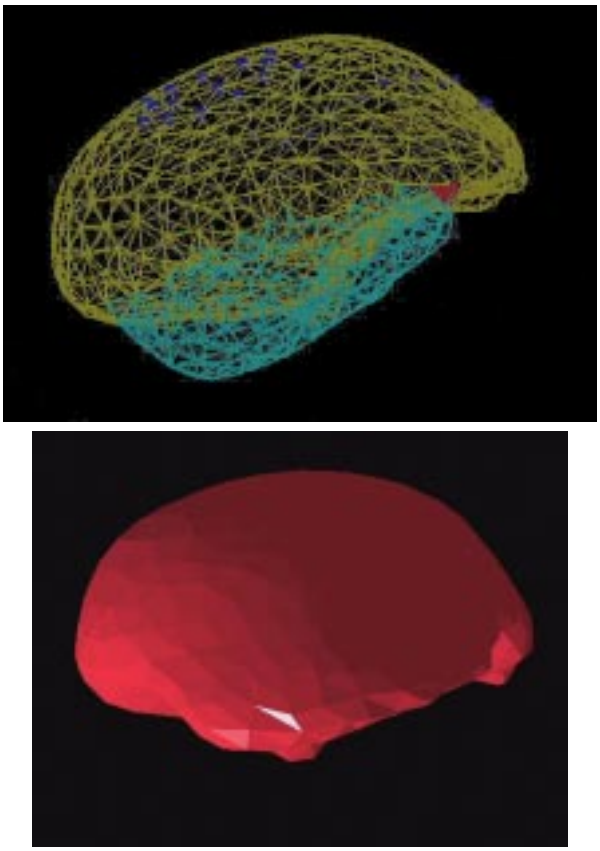

Figure 7. Top : the hybrid liver model seen in wireframe. The upper mesh corresponds to the pre-computed quasi-static elastic model whereas the bottom mesh corresponds to the tensor-mass model. Bottom : the hybrid liver model seen in flat shading. The connection nodes ensure the visual continuity between the two elastic models.

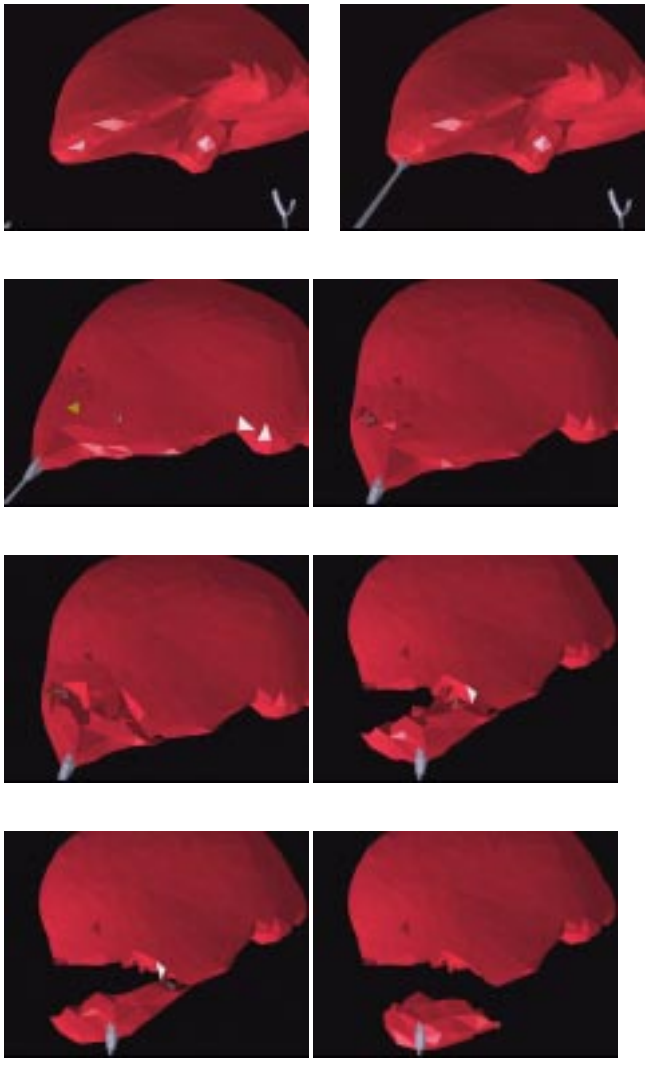

Figure 8. Deformation and cutting of the hybrid elastic model.

7. Conclusion

We have presented three different soft tissue models based on linear elasticity. The first model, introduced in [14], is extremely efficient but do not allow the simulation of cutting. The second model called "tensor-mass model" allows to simulate the dynamics of soft tissue, similarly to spring-mass models. Tensor-mass models are well suited for the simulation of tearing and cutting, but a limited number of elements (around one thousand) is allowed for real-time simulation. Finally, we have proposed hybrid elastic models that combine both previously described elastic models therefore enabling to cut and deform large anatomical structures.

We are currently improving the biomechanical model by modeling anisotropic and non-linear elastic behavior under large displacements. We are also investigating several speed-ud algorithms based on parallel-computation and mesh adaptation. To enhance visual and haptic realism, it will also be necessary to take into account the contact with neighboring organs and to obtain precise biomechanical characteristics of the liver.

Acknowledgments

This project was partially funded by a contract with IRCAD with the collaboration of professor J. Marescaux, director of IRCAD and the help of Y. Russier and J.M. Clément. The authors are grateful for their collaboration in this project and their valuable advices. The author also thank J-C. Lombardo, G. Picinbonno and all the members of the INRIA incentive action AISIM for stimulating discussions.

References

[1] N. Ayache, S. Cotin, and H. Delingette. Surgery simulation with visual and haptic feedback. In Y. Shirai and S. Hirose, editors, *8th International Symposium on Robotics Research*, pages 311–316. Springer, Japan, Oct. 1997.

[2] N. Ayache, S. Cotin, H. Delingette, J.-M. Clement, J. Marescaux, and M. Nord. Simulation of endoscopic surgery. *Journal of Minimally Invasive Therapy and Allied Technologies (MITAT)*, 7(2):71–77, July 1998.

[3] E. Bainville. *Modélisation géométrique et dynamique d'un geste chirurgical*. PhD thesis, Université Joseph Fourier, mars 1996.

[4] E. Bainville, P. Chaffanjon, and P. Cinquin. Computer generated visual assistance during retroperitoneoscopy. *Computers in biology and medecine*, 2(25):165–171, May 1995.

[5] K.-L. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.

[6] R. Baumann and D. Glauser. Force Feedback for Virtual Reality based Minimally Invasive Surgery Simulator. In *Medecine Meets Virtual Reality*, volume 4, San Diego, CA, Jan. 1996.

[7] M. Bro-Nielsen. *Medical Image Registration and Surgery Simulation*. PhD thesis, IMM Technical University of Denmark, Lingby, Denmark, Mar. 1996. IMM-PHD-1996-25.

[8] M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE : Special Issue on Surgery Simulation*, pages 490–503, Apr. 1998.

[9] M. Bro-Nielsen and S. Cotin. Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. In *Proceedings of Eurographics'96 - Computer Graphics Forum*, volume 15, pages 57–66, 1996.

[10] M.-P. Cani-Gascuel and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1), Mar. 1997.

[11] D. T. Chen and D. Zeltzer. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. In E. E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 89–98, July 1992.

[12] P. G. Ciarlet. *Mathematical elasticity Vol. 1: Three-dimensional elasticity*. , Amsterdam, 1987. ISBN 0-444-70259-8.

[13] S. Cotin. *Modèles anatomiques déformables en temps-réel : Application à la simulation de chirurgie avec retour d'effort*. PhD thesis, Université de Nice-Sophia Antipolis, 1997.

[14] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions On Visualization and Computer Graphics*, 1998. to appear.

[15] S. Cotin, H. Delingette, J.-M. Clément, V. Tassetti, J. Marescaux, and N. Ayache. Geometric and Physical Representations for a Simulator of Hepatic Surgery. In *Proceedings of Medecine Meets Virtual Reality IV*, pages 139–151. IOS Press, Jan. 1996.

[16] Couinaud. *Le foie, études anatomiques et chirurgicales*. Masson, 1957.

[17] H. Delingette. Towards realistic soft tissue modeling in medical simulation. *Proceedings of the IEEE : Special Issue on Surgery Simulation*, pages 512–523, Mar. 1998.

[18] O. Deussen, L. Kobbelt, and P. Tucke. Using simulated annealing to obtain a good approximations of deformable bodies. In *Proc. Eurographics Workshop on Animation and Simulation*, 1995.

[19] Y. C. Fung. *Biomechanics - Mechanical Properties of Living Tissues*. Springer-Verlag, second edition, 1993.

[20] S. Gibson, J. Samosky, A. Mor, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, and N. McKenzie. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback . In J. Troccaz, E. Grimson, and R. Mosges, editors, *Proceedings of the First Joint Conference CVRMed-MRCAS'97*, volume 1205 of *Lecture Notes in Computer Science*, pages 369–378, Mar. 1997.

[21] J. P. Gourret, N. Magnenat-Thalmann, and D. Thalmann. Simulation of Object and Human Skin Deformations in a Grasping Task. In *Computer Graphics (SIGGRAPH'89)*, volume 23 No 3, pages 21–31, Boston, MA, USA, July 1989.

[22] D. C. Jean Louchet, Xavier Provot. Evolutionary identification of cloth animation model. In *Workshop on Computer Animation and Simulation (Eurographics'95)*, pages 44–54, 1995.

[23] Kaiss, M. and Le Tallec, P. La Modélisation numérique du contact œil-trépan. *Revue Européenne des Eléments Finis*, 5(3):375–408, Feb. 1996.

[24] U. Kuehnapfel and B. Neisius. CAD-Based Graphical Computer Simulation in Endoscopic Surgery. *End. Surg.*, 1:181–184, 1993.

[25] A. Luciani, S. Jimenez, J. Florens, C. Cadoz, and O. Raoult. Computational physics: a modeler simulator for animated physical objects. In *Eurographics Workshop on Animation and Simulation*, pages 425–437, Vienna, 1991.

[26] W. Maurel, Y. Wu, and N. M. T. D. Thalmann. *Biomechanical Models for Soft Tissue Simulation*. ESPRIT Basic Research Series. Springer-Verlag, 1998.

[27] P. Meseure and C. Chaillou. Deformable Body Simulation with Adaptative Subdivision and Cuttings. In *Proceedings of the WSCG'97*, pages 361–370, Feb. 1997.

[28] J. C. Platt and A. H. Barr. Constraint Methods for Flexible Models. In *Computer Graphics (SIGGRAPH'88)*, volume 22 No 4, pages 279–288, 1988.

[29] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge Press, 1991.

[30] G. J. Song and N. P. Reddy. Tissue Cutting In Virtual Environment. In *Medecine Meets Virtual Reality IV*, pages 359–364. IOS Press, 1995.

[31] T. H. Speeter. Three Dimensional Finite Element Analysis of Elastic Continua for Tactile Sensing. *The International Journal of Robotics Research*, 11 No 1:1–19, Feb. 1992.

[32] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 205–214, July 1987.

[33] L. Tsap, D. Goldgof, and S. Sarkar. Efficient nonlinear finite element modeling of nonrigid objects via optimization of mesh models. *Journal of Computer Vision and Image Understanding*, 69(3):330–350, Mar. 1998.

[34] K. Waters. A physical model of facial tissue and muscle articulation derived from computer tomography data. In *Visualization in Biomedical Computing (VBC'92)*, volume 574, Chappel Hill, NC, 1992.

[35] O. Zienkiewicz. *The finite element method*. McGraw-Hill, London, 3 edition, 1977.