

Non-linear anisotropic elasticity for real-time surgery simulation

Guillaume Picinbono, Hervé Delingette,* and Nicholas Ayache

INRIA, Epidauré project, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex, France

Received 14 November 2000; received in revised form 14 May 2002; accepted 31 March 2003

Abstract

In this paper, we describe the latest developments of the minimally invasive hepatic surgery simulator prototype developed at INRIA. A key problem with such a simulator is the physical modeling of soft tissues. We propose a new deformable model based on non-linear elasticity, anisotropic behavior, and the finite element method. This model is valid for large displacements, which means in particular that it is invariant with respect to rotations. This property improves the realism of the deformations and solves the problems related to the shortcomings of linear elasticity, which is only valid for small displacements. We also address the problem of volume variations by adding to our model incompressibility constraints. Finally, we demonstrate the relevance of this approach for the real-time simulation of laparoscopic surgical gestures on the liver.

© 2003 Elsevier Science (USA). All rights reserved.

Keywords: Surgery simulation; Non-linear elasticity; Large displacement; Finite element method; Deformable model; Real-time; Anisotropy; Incompressibility constraints

1. Introduction

A major and recent evolution in abdominal surgery has been the development of laparoscopic surgery. In this type of surgery, abdominal operations such as hepatic resection are performed through small incisions. A video camera and special surgical

* Corresponding author. Fax: +33(0)4-92-38-76-69.

E-mail addresses: Herve.delingette@sophia.inria.fr (H. Delingette), Nicholas.Ayache@sophia.inria.fr (N. Ayache).

tools are introduced into the abdomen, allowing the surgeon to perform a procedure less invasively. A drawback of this technique lies essentially in the need for more complex gestures and in the loss of direct visual and tactile information. Therefore the surgeon needs to learn and adapt himself to this new type of surgery and in particular to a new type of hand-eye coordination. In this context, surgical simulation systems could be of great help in the training process of surgeons.

Among the several key problems in the development of a surgical simulator [1,2], the geometrical and physical representation of human organs remain the most important. The deformable model must be at the same time very realistic (both visually and physically) and very efficient to allow real-time deformations. Several methods have been proposed: spring-mass models [3,4], free-form deformations [5], adaptive sampling of non-linear elastic models [6], or various finite element methods [7–12].

In this paper, we propose a new real-time deformable model based on non-linear elasticity and a finite element method. We first introduce the linear elasticity theory and its implementation through the finite element method, and we then highlight its shortcomings when the “small displacement” hypothesis does not hold. Then we focus on our implementation of St. Venant–Kirchhoff elasticity and incompressibility constraints.

2. Shortcomings of the linear elasticity model

Linear elasticity is often used for the modeling of deformable materials, mainly because the equations remain quite simple and the computation time can be optimized.

The physical behavior of soft tissue may be considered as linear elastic if its displacement and deformation remain small [13,14] (typically less than 10% of the mesh size). We represent the deformation of a volumetric model from its rest shape $\mathcal{M}_{\text{initial}}$ with a *displacement vector* $\mathbf{U}(x, y, z)$ for $(x, y, z) \in \mathcal{M}_{\text{initial}}$ and we write $\mathcal{M}_{\text{deformed}} = \mathcal{M}_{\text{initial}} + \mathbf{U}(x, y, z)$.

From this displacement vector, we define the linearized *Green–St. Venant strain tensor* (3×3 symmetric matrix) E_1 and its principal invariants l_1 and l_2 :

$$E_1 = \frac{1}{2}(\nabla\mathbf{U} + \nabla\mathbf{U}^t), \quad l_1 = \text{tr} E_1, \quad l_2 = \text{tr} E_1^2, \quad (1)$$

where $\nabla\mathbf{U}$ is the 3×3 gradient matrix and $\text{tr} E_1$ is the trace of the matrix E_1 .

The linear elastic energy W_{Linear} , for homogeneous isotropic materials, is defined by the following formula (see [15]):

$$W_{\text{Linear}} = \frac{\lambda}{2}(\text{tr} E_1)^2 + \mu \text{tr} E_1^2 = \frac{\lambda}{2}(\text{div} \mathbf{U})^2 + \mu \|\nabla\mathbf{U}\|^2 - \frac{\mu}{2} \|\text{rot} \mathbf{U}\|^2, \quad (2)$$

where λ and μ are the *Lamé coefficients* characterizing the material stiffness, $\text{div} \mathbf{U}$ is the divergence of the vector field $\mathbf{U}(x, y, z)$, $\|\nabla\mathbf{U}\|^2$ is the square norm of the gradient matrix, and $\text{rot} \mathbf{U}$ is the rotational matrix of the vector field $\mathbf{U}(x, y, z)$.

Eq. (2), known as *Hooke’s law*, shows that the elastic energy of a deformable object is a quadratic function of the displacement vector.

2.1. Finite element method

Finite element method is a classical way to solve continuum mechanics equations. It is a mathematical framework allowing to discretize a continuous variational problem [16]. We chose to use P_1 finite elements, where the elementary volume is a tetrahedron with a node defined at each vertex (Fig. 1). At each point $\mathbf{M}(x, y, z)$ inside tetrahedron \mathbf{T}_i , the displacement vector is expressed as a function of the displacements \mathbf{U}_k of vertices \mathbf{P}_k . For P_1 finite elements, interpolation functions Λ_k are linear ($\{\Lambda_k; k = 0, \dots, 3\}$ are the barycentric coordinates of \mathbf{M} in the tetrahedron):

$$\mathbf{U}(x, y, z) = \sum_{j=0}^3 \mathbf{U}_j \Lambda_j(x, y, z), \quad \Lambda_j(x, y, z) = \boldsymbol{\alpha}_j \cdot \mathbf{X} + \beta_j$$

$$\boldsymbol{\alpha}_j = \frac{(-1)^j}{6V(\mathbf{T}_i)} (\mathbf{P}_{j+1} \times \mathbf{P}_{j+2} + \mathbf{P}_{j+2} \times \mathbf{P}_{j+3} + \mathbf{P}_{j+3} \times \mathbf{P}_{j+1}),$$

where \times stands for the cross product between two vectors and $V(\mathbf{T}_i)$ is the volume of the tetrahedron.

Using this equation for the displacement vector \mathbf{U} leads to the finite element formulation of linear elastic energy in the tetrahedron \mathbf{T}_i [8]:

$$W_{\text{Linear}}(\mathbf{T}_i) = \sum_{j,k=0}^3 \mathbf{U}_j^t [\mathcal{B}_{jk}^{\mathbf{T}_i}] \mathbf{U}_k, \quad (3)$$

$$\mathcal{B}_{jk}^{\mathbf{T}_i} = \frac{\lambda}{2} (\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k) + \frac{\mu}{2} [(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_j) + (\boldsymbol{\alpha}_j \cdot \boldsymbol{\alpha}_k) \text{Id}_3],$$

where $[\mathcal{B}_{jk}^{\mathbf{T}_i}]$ is the tetrahedron contribution to the stiffness tensor of the edge $(\mathbf{P}_j, \mathbf{P}_k)$ (or of the vertex \mathbf{P}_j if $j = k$), $\{\boldsymbol{\alpha}_j, k = 0, \dots, 3\}$ are the shape vectors of the tetrahedron and \otimes stands for the tensor product of two vectors, which gives a (3×3) matrix: $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^t$. Id_3 is the (3×3) identity matrix.

Finally, to obtain the force $\mathbf{F}_p^{\mathbf{T}_i}$ applied by the tetrahedron \mathbf{T}_i on the vertex \mathbf{P}_p , we derive the elastic energy with respect to the vertex displacement \mathbf{U}_p :

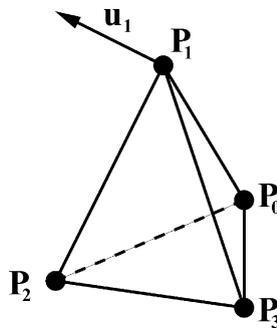


Fig. 1. P_1 finite element.

$$\mathbf{F}_p^{\mathbf{T}_i} = 2 \sum_{j=0}^3 [\mathcal{B}_{pj}^{\mathbf{T}_i}] \mathbf{U}_j. \quad (4)$$

We have been using this linear elasticity formulation for several years through two deformable models, the *pre-computed model* [7] and the *tensor-mass model* [8,17]. Furthermore, it can be extended to anisotropic linear elasticity [9,18], which allows to model fiber-reinforced materials, very common within biological tissues (tendons, muscles, etc.), or other anatomical structures like blood vessels.

2.2. The problem of rotational invariance

The main limitation of the linear model is that it is not invariant with respect to rotations. When the object undergoes a rotation, the elastic energy increases, leading to a variation of the volume (see Fig. 2). In the case of a global rotation of the object, we could solve the problem with a specific change of the reference frame.

But this solution proves itself to be ineffective when only one part of the object undergoes a rotation (which is the case in general). This case is presented by the cylinder of Fig. 3: the bottom face is fixed and a force is applied to the central top vertex. Arrows show the trajectory of some vertices, which are constrained by the linear model to move along straight lines. This results in the distortion of the mesh. Furthermore, this abnor-

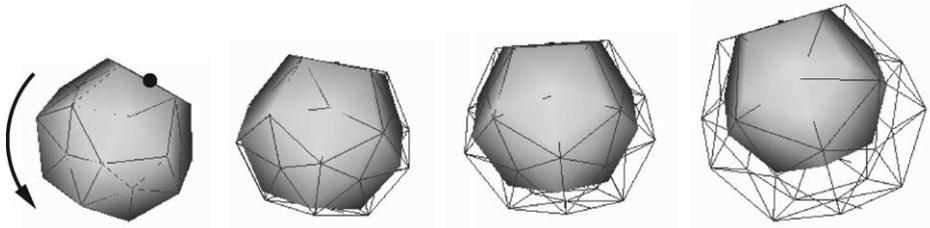


Fig. 2. Global rotation of the linear elastic model (wireframe).

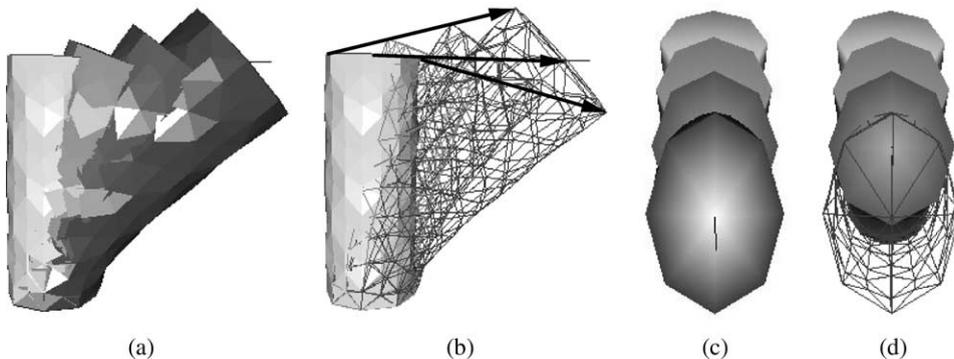


Fig. 3. Successive deformations of a linear elastic cylinder. (a) and (b): side view. (c) and (d): top view.

mal deformation is not the same in all directions since the object only deforms itself in the rotation plane (Fig. 3(c) and (d)). This unrealistic behavior of the linear elastic model for large displacements led us to consider different models of elasticity.

3. St. Venant–Kirchhoff elasticity

A model of elasticity is considered as a large displacement model if it derives from a strain tensor which is a quadratic function of the deformation gradient. Most common tensors are the *left* and *right Cauchy–Green strain tensors* (respectively $B = \nabla\phi\nabla\phi^t$ and $C = \nabla\phi^t\nabla\phi$, ϕ being the deformation function).

The St. Venant–Kirchhoff model is a generalization of the linear model for large displacements, and is a particular case of hyperelastic materials. The basic energy equation is the same (Eq. (2)), but now E stands for the complete *Green–St. Venant strain tensor*:

$$E = \frac{1}{2}(C - I) = \frac{1}{2}(\nabla\mathbf{U} + \nabla\mathbf{U}^t + \nabla\mathbf{U}^t\nabla\mathbf{U}). \tag{5}$$

Elastic energy, which was a quadratic function of $\nabla\mathbf{U}$ in the linear case, is now a polynomial of order four with respect to $\nabla\mathbf{U}$:

$$\begin{aligned} W &= \frac{\lambda}{2}(\text{tr}E)^2 + \mu\text{tr}E^2 \\ &= \frac{\lambda}{2}\left[(\text{div}\mathbf{U}) + \frac{1}{2}\|\nabla\mathbf{U}\|^2\right]^2 + \mu\|\nabla\mathbf{U}\|^2 - \frac{\mu}{2}\|\text{rot}\mathbf{U}\|^2 \\ &\quad + \mu(\nabla\mathbf{U} : \nabla\mathbf{U}^t\nabla\mathbf{U}) + \frac{\mu}{4}\|\nabla\mathbf{U}^t\nabla\mathbf{U}\|^2, \end{aligned} \tag{6}$$

$$W = W_{\text{Linear}} + \frac{\lambda}{2}(\text{div}\mathbf{U})\|\nabla\mathbf{U}\|^2 + \frac{\lambda}{8}\|\nabla\mathbf{U}\|^4 + \mu(\nabla\mathbf{U} : \nabla\mathbf{U}^t\nabla\mathbf{U}) + \frac{\mu}{4}\|\nabla\mathbf{U}^t\nabla\mathbf{U}\|^2,$$

where W_{Linear} is given by Eq. (2), and $A : B = \text{tr}(A^tB) = \sum_{i,j} a_{ij}b_{ij}$ is the dot product of two matrices.

In [9,18], we have generalized linear elasticity to materials having a different behavior in one given direction. These materials, called “transversally isotropic” materials, can also be modeled with St. Venant–Kirchhoff elasticity by adding to the isotropic elastic energy of Eq. (6), an anisotropic contribution which penalizes the material stretch in the direction given by unit vector \mathbf{a}_0 :

$$W_{\text{Trans_iso}} = W + \left(\frac{\lambda^L - \lambda}{2} + \mu^L - \mu\right)(\mathbf{a}_0^t E \mathbf{a}_0)^2, \tag{7}$$

where λ^L and μ^L are the Lamé constants along the direction of anisotropy \mathbf{a}_0 .

3.1. Finite element modeling

With the notations introduced in Section 2.1. we express the St. Venant–Kirchhoff elastic model with finite element theory as:

$$W(\mathbf{T}_i) = \sum_{j,k} \mathbf{U}_j^t [\mathcal{B}_{jk}^{\mathbf{T}_i}] \mathbf{U}_k + \sum_{j,k,l} (\mathbf{U}_j \cdot \mathcal{C}_{jkl}^{\mathbf{T}_i}) (\mathbf{U}_k \cdot \mathbf{U}_l) + \sum_{j,k,l,m} \mathcal{D}_{ijklm}^{\mathbf{T}_i} (\mathbf{U}_j \cdot \mathbf{U}_k) (\mathbf{U}_l \cdot \mathbf{U}_m), \quad (8)$$

where the terms $\mathcal{B}_{jk}^{\mathbf{T}_i}$, $\mathcal{C}_{jkl}^{\mathbf{T}_i}$, and $\mathcal{D}_{ijklm}^{\mathbf{T}_i}$, called “stiffness parameters,” are given by:

- $\mathcal{B}_{jk}^{\mathbf{T}_i}$ is a (3×3) symmetric matrix (which corresponds to the linear component of the energy):

$$\begin{aligned} \mathcal{B}_{jk}^{\mathbf{T}_i} &= \frac{\lambda}{2} (\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k) + \frac{\mu}{2} [(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_j) + (\boldsymbol{\alpha}_j \cdot \boldsymbol{\alpha}_k) \text{Id}_3] \\ &\quad + \left(\frac{\lambda^L - \lambda}{2} + \mu^L - \mu \right) (\mathbf{a}_0 \otimes \mathbf{a}_0) (\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k) (\mathbf{a}_0 \otimes \mathbf{a}_0), \end{aligned}$$

- $\mathcal{C}_{jkl}^{\mathbf{T}_i}$ is a vector:

$$\begin{aligned} \mathcal{C}_{jkl}^{\mathbf{T}_i} &= \frac{\lambda}{2} \boldsymbol{\alpha}_j (\boldsymbol{\alpha}_k \cdot \boldsymbol{\alpha}_l) + \frac{\mu}{2} [\boldsymbol{\alpha}_l (\boldsymbol{\alpha}_j \cdot \boldsymbol{\alpha}_k) + \boldsymbol{\alpha}_k (\boldsymbol{\alpha}_j \cdot \boldsymbol{\alpha}_l)] \\ &\quad + \left(\frac{\lambda^L - \lambda}{2} + \mu^L - \mu \right) (\mathbf{a}_0 \otimes \mathbf{a}_0) (\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k) (\mathbf{a}_0 \otimes \mathbf{a}_0) \boldsymbol{\alpha}_l, \end{aligned}$$

- and $\mathcal{D}_{ijklm}^{\mathbf{T}_i}$ is a scalar:

$$\begin{aligned} \mathcal{D}_{ijklm}^{\mathbf{T}_i} &= \frac{\lambda}{8} (\boldsymbol{\alpha}_j \cdot \boldsymbol{\alpha}_k) (\boldsymbol{\alpha}_l \cdot \boldsymbol{\alpha}_m) + \frac{\mu}{4} (\boldsymbol{\alpha}_j \cdot \boldsymbol{\alpha}_m) (\boldsymbol{\alpha}_k \cdot \boldsymbol{\alpha}_l) \\ &\quad + \left(\frac{\lambda^L - \lambda}{8} + \frac{\mu^L - \mu}{4} \right) (\mathbf{a}_0 \cdot \boldsymbol{\alpha}_j) (\mathbf{a}_0 \cdot \boldsymbol{\alpha}_k) (\mathbf{a}_0 \cdot \boldsymbol{\alpha}_l) (\mathbf{a}_0 \cdot \boldsymbol{\alpha}_m). \end{aligned}$$

- The last term of each stiffness parameter represents the anisotropic behavior of the material.

The force applied at each vertex \mathbf{P}_p inside a tetrahedron is obtained by derivation of the elastic energy $W(\mathbf{T}_i)$ with respect to the displacement \mathbf{U}_p :

$$\mathbf{F}^p(\mathbf{T}_i) = 2 \underbrace{\sum_j [\mathcal{B}_{pj}^{\mathbf{T}_i}] \mathbf{U}_j}_{\mathbf{F}_1^p(\mathbf{T}_i)} + \underbrace{\sum_{j,k} 2(\mathbf{U}_k \otimes \mathbf{U}_j) \mathcal{C}_{jkp}^{\mathbf{T}_i} + (\mathbf{U}_j \cdot \mathbf{U}_k) \mathcal{C}_{pjk}^{\mathbf{T}_i}}_{\mathbf{F}_2^p(\mathbf{T}_i)} + 4 \underbrace{\sum_{j,k,l} \mathcal{D}_{ijklp}^{\mathbf{T}_i} \mathbf{U}_l \mathbf{U}_k^t \mathbf{U}_j}_{\mathbf{F}_3^p(\mathbf{T}_i)}. \quad (9)$$

The first term of the elastic force ($\mathbf{F}_1^p(\mathbf{T}_i)$) corresponds to the linear elastic case presented in Section 2.1. The next part of the paper deals with the generalization of the tensor-mass model to large displacements.

3.2. Non-linear tensor-mass model

The main idea of the tensor-mass model is to split, for each tetrahedron, the force applied to a vertex in two parts—forces created by the vertex displacement and forces produced by the displacements of its neighbors:

$$\mathbf{F}_1^p(\mathbf{T}_i) = 2[\mathcal{B}_{pp}^{\mathbf{T}_i}] \mathbf{U}_p + 2 \sum_{j \neq p} [\mathcal{B}_{pj}^{\mathbf{T}_i}] \mathbf{U}_j. \tag{10}$$

This way we can define for each tetrahedron a set of *local stiffness tensors* for vertices ($\{\mathcal{B}_{pp}^{\mathbf{T}_i}; p = 0, \dots, 3\}$) and for edges ($\{\mathcal{B}_{pj}^{\mathbf{T}_i}; p, j = 0, \dots, 3; p \neq j\}$). By doing this for every tetrahedron, we can accumulate on vertices and edges of the mesh the corresponding contributions to the *global stiffness tensors*:

$$\mathcal{B}^{pp} = \sum_{\mathbf{T}_i \in N(\mathbf{v}_p)} \mathcal{B}_{pp}^{\mathbf{T}_i}, \quad \mathcal{B}^{pj} = \sum_{\mathbf{T}_i \in N(\mathbf{E}_{pj})} \mathcal{B}_{kl}^{\mathbf{T}_i}.$$

These *stiffness tensors* are computed when creating the mesh and are stored for each vertex and edge of the mesh.

The same principle can be applied to the quadratic term ($\mathbf{F}_2^p(\mathbf{T}_i)$ of Eq. (9)) and the cubic term ($\mathbf{F}_3^p(\mathbf{T}_i)$) (see Appendix A). The former brings *stiffness vectors* for vertices, edges, and triangles, and the latter brings *stiffness scalars* for vertices, edges, triangles, and tetrahedra. Table 1 summarizes the stiffness parameters stored on each geometrical primitive of the mesh.

Given a tetrahedral mesh of a solid—in our case an anatomical structure—we build a data structure incorporating the notion of vertices, edges, triangles, and tetrahedra, with all their neighbors. For each vertex, we store its current position \mathbf{P}_p , its rest position \mathbf{P}_p^0 , and its stiffness parameters. For each edge and each triangle, we store stiffness parameters. Finally for each tetrahedron, we store the Lamé coefficients λ and μ (and λ^L , μ^L , and \mathbf{a}_0 if the model is anisotropic), the four shape vectors α_k , and the stiffness parameters.

During the simulation, we compute forces for each vertex, edge, triangle, and tetrahedron, and we use a Newtonian differential equation to update the vertex positions:

$$m_i \frac{d^2 \mathbf{P}_i}{dt^2} = \gamma_i \frac{d \mathbf{P}_i}{dt} + \mathbf{F}_i. \tag{11}$$

Table 1
Storage of the stiffness parameters on the mesh

Stiffness parameters distribution	Tensors	Vectors	Scalars
Vertex \mathbf{V}_p	\mathcal{B}^{pp}	\mathcal{C}^{ppp}	\mathcal{D}^{pppp}
Edge \mathbf{E}_{pj}	\mathcal{B}^{pj}	\mathcal{C}^{ppj} \mathcal{C}^{ijp}	\mathcal{D}^{ijpp} \mathcal{D}^{pjip} \mathcal{D}^{ijjp} \mathcal{D}^{ijpp}
Triangle \mathbf{F}_{pjk}		\mathcal{C}^{jkp} \mathcal{C}^{kjp} \mathcal{C}^{pjk}	\mathcal{D}^{jkpp} \mathcal{D}^{jpkp} \mathcal{D}^{jkjp} \mathcal{D}^{kjkp} \mathcal{D}^{pjkp} \mathcal{D}^{kjjp} \mathcal{D}^{ikkp}
Tetrahedron \mathbf{T}_{pjkl}			\mathcal{D}^{ijkl} \mathcal{D}^{klij} \mathcal{D}^{jlkp} \mathcal{D}^{lkjp}

This equation is related to the differential equations of continuum mechanics [19]:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{F}(\mathbf{U}) = \mathbf{R}. \quad (12)$$

Following finite element theory, the mass \mathbf{M} and damping \mathbf{C} matrices are sparse matrices which 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 us to write Eq. (12) as the set of independent differential equations 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 $t + 1$:

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

One of the basic tasks in surgery simulation consists in cutting soft tissue. With our deformable model, this task can be achieved efficiently. We simulate the action of an electric scalpel on soft tissue by successively removing tetrahedra at places where the instrument is in contact with the anatomical model.

When removing a tetrahedron, 280 floating numbers update operations are performed to suppress the tetrahedron contributions to the stiffness parameters of the surrounding vertices, edges, and triangles (see Appendix B). By locally updating stiffness parameters, the tissue has exactly the same properties as if we had removed the corresponding tetrahedron at its rest position. Because of the volumetric continuity of finite element modeling, the tissue deformation remains realistic during the cutting.

4. Incompressibility constraint

Living tissue, which is essentially made of water, is nearly incompressible. This property is difficult to model and leads in most cases to instability problems. This is the case with the St. Venant–Kirchhoff model: the material remains incompressible when the Lamé constant λ tends towards infinity. Taking a large value for λ would force us to decrease the time step and therefore to increase the computation time. Another reason to add an external incompressibility constraint to our model is related to the model itself: the main advantage of the St. Venant–Kirchhoff model is to use the strain tensor E which is invariant with respect to rotations. But it is also invariant with respect to symmetries, which could lead to the reversal of some tetrahedra under strong constraints.

We choose to penalize volume variation by applying to each vertex of the tetrahedron a force directed along the normal of the opposite face \mathbf{N}_p (see Fig. 4), the norm of the force being proportional to the square of the relative volume variation:

$$\mathbf{F}_{\text{incomp}}^p = \text{sign}(V - V_0) \left(\frac{V - V_0}{V_0}\right)^2 \vec{\mathbf{N}}_p. \quad (13)$$

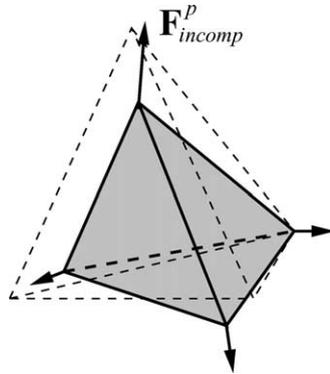


Fig. 4. Penalization of the volume variation.

Since the volume V is proportional to the height of each vertex its opposite triangle, when V is greater than V_0 then the force \mathbf{F}_{incomp}^p tends to decrease V by moving each vertex along the normal of its opposite triangle. These forces act as an artificial pressure inside each tetrahedron. This method is closely related to Lagrange multipliers, which are often used to solve problem of energy minimization under constraints.

5. Results

In the first experiment, we wish to highlight the contributions of our new deformable model in the case of partial rotations. Fig. 5 shows the same experience as the one presented for linear elasticity (Section 2.2, Fig. 3). On the left we can see that the cylinder vertices can now follow trajectories different from straight lines (Fig. 5(a)), leading to much more realistic deformations than in the linear (wire-frame) case (Fig. 5(b) and (c)).

The second example presents the differences between isotropic and anisotropic materials. The three cylinders of Fig. 6 have their top and bottom faces fixed, and

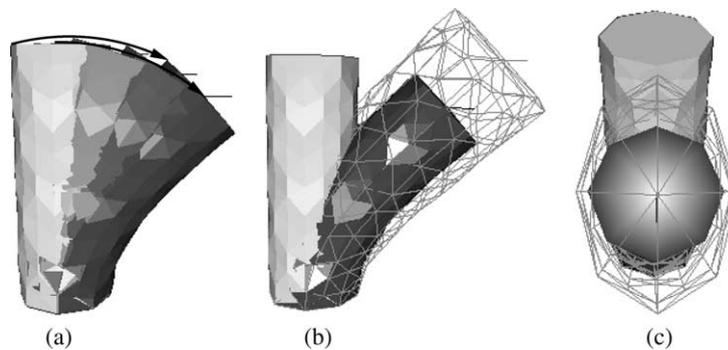


Fig. 5. (a) Successive deformations of the non-linear model. Side (b) and top (c) view of the comparison between linear (wireframe) and non-linear model (solid rendering).

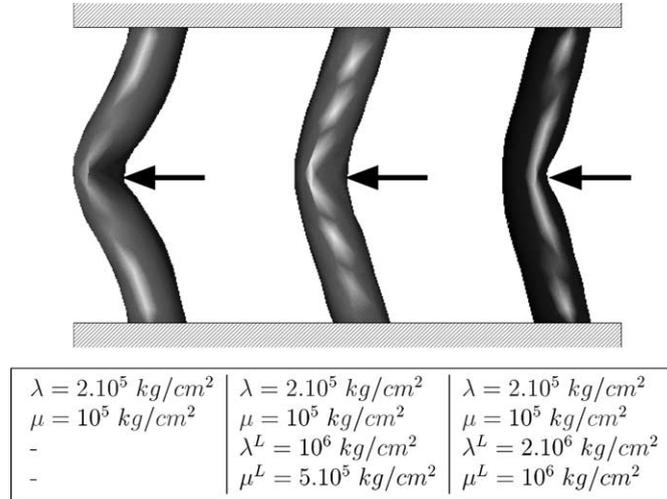


Fig. 6. Deformation of tubular structures with non-linear transversally isotropic elasticity.

are submitted to the same forces. While the isotropic model on the left undergoes a “snake-like” deformation, the last two, which are anisotropic along their height, stiffen in order to minimize their stretch in the anisotropic direction. The rightmost model, being twice as stiff as the middle one in the anisotropic direction, starts to squeeze in the plane of isotropy because it cannot stretch anymore.

In the third example (Fig. 7), we apply a force to the right lobe of the liver (the liver is fixed in a region near the center of its back side, and Lamé constants are: $\lambda = 4 \times 10^4 \text{ kg/cm}^2$ and $\mu = 10^4 \text{ kg/cm}^2$). Using the linear model, the right part of the liver undergoes a large (and unrealistic) volume increase, whereas with non-linear elasticity, the right lobe is able to partially rotate, while undergoing a much more realistic deformation.

Adding the incompressibility constraint on the same examples decreases the volume variation even more (see Table 2), and also stabilizes the behavior of the deformable models in strongly constrained areas.

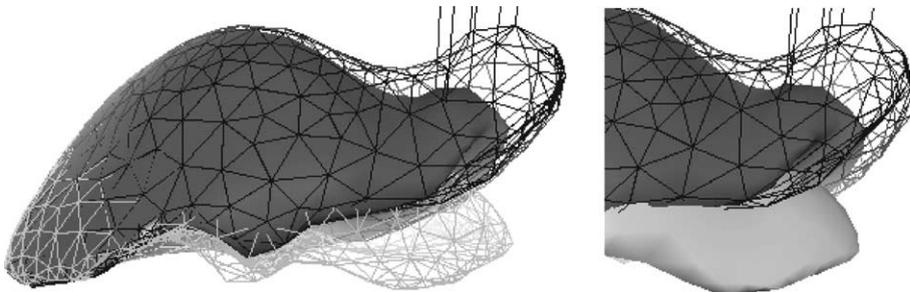


Fig. 7. Linear (wireframe), non-linear (solid) liver models, and rest shape (bottom).

Table 2

Volume variation results. For the cylinder: left, middle, and right stand for the different deformations of Figs. 3 and 5(a)

Volume variations (%)	Linear	Non-linear	Non-linear incomp.
Cylinder left middle right	7 28 63	0.3 1 2	0.2 0.5 1
Liver	9	1.5	0.7

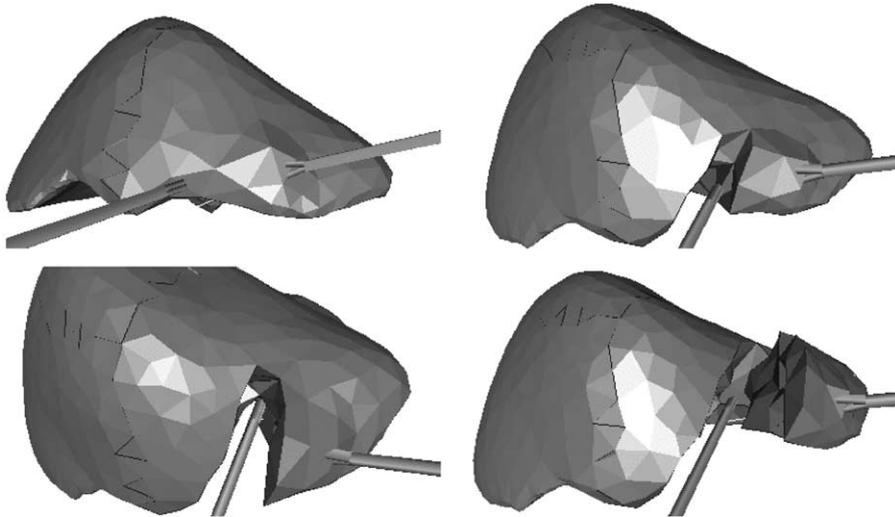


Fig. 8. Simulation of laparoscopic liver surgery.

The last example is the simulation of a typical laparoscopic surgical gesture on the liver. One tool is pulling the edge of the liver sideways while a bipolar cautery device cuts it. During the cutting, the surgeon pulls away the part of the liver he wants to remove. This piece of liver undergoes large displacements and the deformation appears fairly realistic with this new non-linear deformable model (Fig. 8).

Obviously, the computation time of this model is larger than for the linear model because the force equation is much more complex (Eq. (9)). With our current implementation, simulation frequency is five times slower than with the linear model. Nevertheless, with this non-linear model, we can reach a frequency update of 25 Hz on meshes made of about 2000 tetrahedra (on a PC Pentium PIII 500 MHz). This is sufficient to reach visual real-time with quite complex objects, and even to provide a realistic haptic feedback using force extrapolation as described in [9].

6. Optimization of non-linear deformations

We have shown in this paper that non-linear elasticity allows us to simulate much more realistic deformations than linear elasticity as soon as the model undergoes

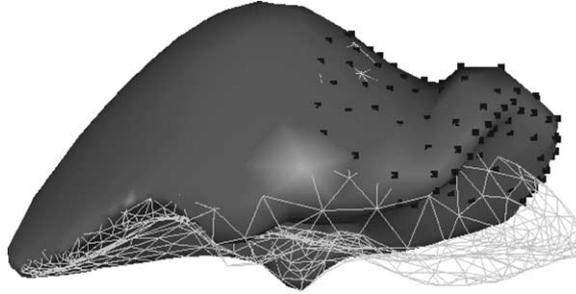


Fig. 9. Adaptable non-linear model deformation compared with its rest position (wireframe).

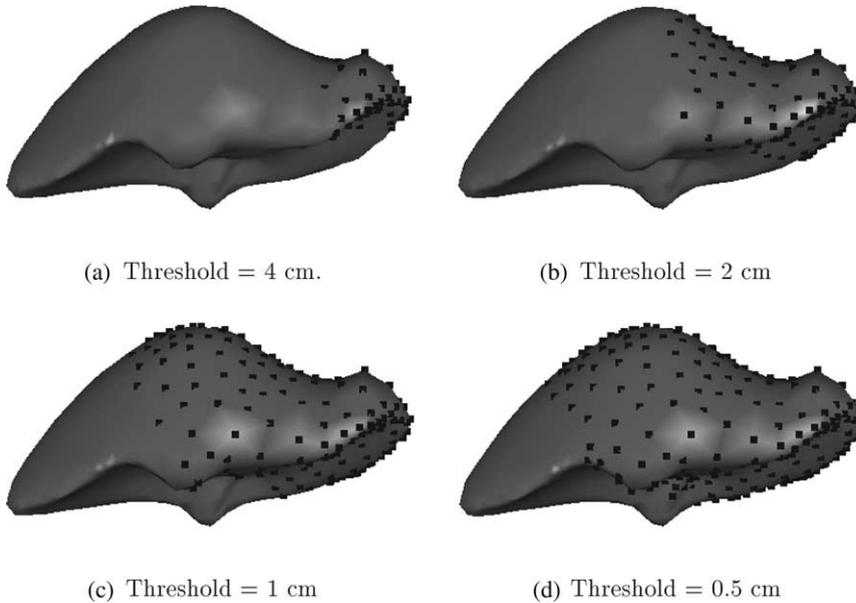


Fig. 10. Deformation of the adaptable non-linear model for several values of the threshold.

large displacements. However, non-linear elasticity is more computationally expensive than linear elasticity. Since non-linear elastic forces tend to linear elastic forces as the maximum vertex displacement decreases to zero, we propose to use non-linear elasticity only at parts of the mesh where displacements are larger than a given threshold, the remaining part using linear elasticity. Thus, we have modified the force computation algorithm in the following manner: for each vertex, we first compute the linear part of the force, and we add the non-linear part only if its displacement is larger than a threshold.

Fig. 9 shows a deformation computed with this optimization (same model as in Fig. 7). This liver model is made of 6342 tetrahedra and 1394 vertices. The threshold is set to 2 cm while the mesh is about 30 cm long. The points drawn on the surface

identify vertices using non-linear elasticity. With this method, we reach an update frequency of 20 Hz instead of 8 Hz with a fully non-linear model.

The same deformation is presented in Fig. 10 for different values of the threshold. With this method, we can choose a trade-off between the bio-mechanical realism of the deformation and the updating frequency of the simulation. The diagram on Fig. 11 shows the update frequencies reached for each value of the threshold, in comparison with the fully linear and the fully non-linear models. Even when the threshold tends towards infinity, the adaptable model is slower than the linear model, because the computation algorithm of the non-linear force is more complex. Indeed, the com-

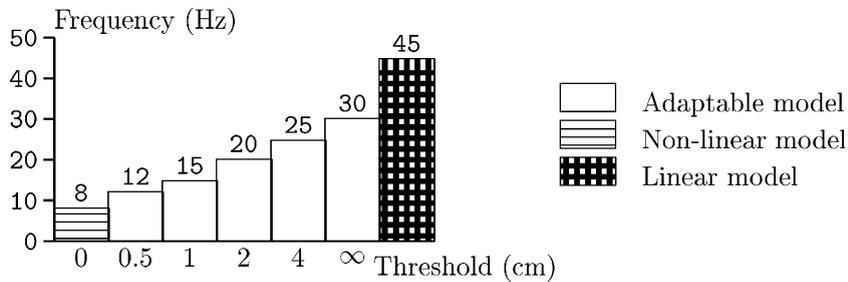


Fig. 11. Updating frequencies of the adaptable model for several values of the threshold.

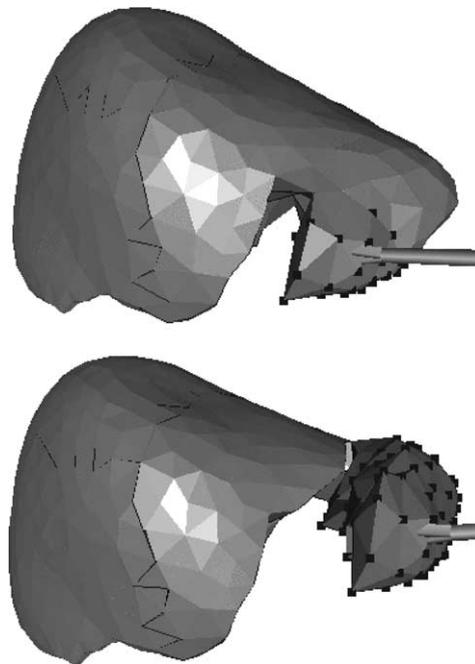


Fig. 12. Surgery simulation using adaptable model.

putation of non-linear forces requires to visit all vertices, edges, triangles, and tetrahedra of the mesh, whereas only vertices and edges need to be visited for the linear model.

For the simulation example of Fig. 8, this optimization allows to reach update frequencies varying between 50 and 80 Hz, depending on the ratio of points using non-linear elasticity (Fig. 12). The minimal frequency of 50 Hz is reached at the end of the simulation, when all vertices of the resected part of the liver are using large displacement elasticity (on the right of Fig. 12).

In general, two strategies can be used to set the value of this threshold. In the first strategy, the threshold is increased until a given update frequency is matched as demonstrated previously. The second strategy is physically motivated and sets the threshold to 10% of the typical size of the mesh since it corresponds to the extent of displacement where linear elasticity is considered as a valid constitutive law.

7. Validation

The liver is filled with blood (near 50% of its global weight), which implies that *ex vivo* liver tissues should not behave like *in vivo* tissues. Therefore, to validate our biomechanical liver model, we would have to compare it against *in-vivo* deformations of a real liver. Currently, such an experiment raises several technical and ethical difficulties. For the same reasons, the elastic properties of the liver (Young modulus and Poisson ratio) must be measured *in vivo*, or using complex *in vitro* experimental protocols. In both cases, the results obtained by several teams [20–22] have shown a strong variability in the parameters estimation.

In the framework of real-time surgery simulation, we essentially need to build soft tissue models that behave realistically. Therefore, we are currently relying on the feedback from surgeons experimenting the simulator to validate the behavior of our liver model. They consider that this St. Venant–Kirchhoff non-linear elastic modeling of soft tissues is sufficient for most simulation purposes.

8. Conclusion

We have proposed in this paper a new deformable model based on large displacement elasticity, a finite element method, and a dynamic explicit integration scheme. It solves the problem of rotational invariance and takes into account the anisotropic behavior and the incompressibility properties of biological tissues. Furthermore we have optimized the computation time of this model by computing the non-linear part of the force only for the parts of the mesh which undergo “large displacements.”

Including this model into our laparoscopic surgery simulator prototype improves its bio-mechanical realism and thus increases its potential use for learning and training processes.

Our future work will focus on improving the computation efficiency with a new formulation of large displacement elasticity. Also, we plan to include vessels net-

works inside hepatic parenchyma and introduce, besides our geometric and physical models, a preliminary physiological model.

Appendix A. Non-linear force equation

Eq. (9) gives the force applied by tetrahedron \mathbf{T}_i on vertex \mathbf{P}_p . To build the corresponding tensor-mass model, we need to split this equation (i.e., to split the sums) into different forces, respectively, computed on vertex \mathbf{P}_p , on edges $(\mathbf{P}_p, \mathbf{P}_j)$, on triangles $(\mathbf{P}_p, \mathbf{P}_j, \mathbf{P}_k)$, and the tetrahedron \mathbf{T}_i . Then, we add the contributions of all adjacent tetrahedra, to obtain the equation which gives the force applied on vertex \mathbf{P}_p as a function of the displacements of its neighbors:

$$\begin{aligned}
 \mathbf{F}^p = & \underbrace{2[\mathcal{B}^{pp}] \mathbf{U}_p + [2(\mathbf{U}_p \otimes \mathbf{U}_p) + (\mathbf{U}_p \cdot \mathbf{U}_p) \text{Id}_3] \mathcal{C}^{ppp} + 4\mathcal{D}^{pppp} \mathbf{U}_p \mathbf{U}_p^t \mathbf{U}_p}_{\text{(Vertex contribution)}} \\
 & + \sum_{\text{edges}(p,i)} \left\{ 2[\mathcal{B}^{pj}] \mathbf{U}_j + 2[(\mathbf{U}_j \otimes \mathbf{U}_p) + (\mathbf{U}_j \cdot \mathbf{U}_p) \text{Id}_3] \mathcal{C}^{ppj} \right. \\
 & \quad + 2(\mathbf{U}_p \otimes \mathbf{U}_j) \mathcal{C}^{jpp} + 2(\mathbf{U}_j \otimes \mathbf{U}_j) \mathcal{C}^{jip} + (\mathbf{U}_j \cdot \mathbf{U}_j) \mathcal{C}^{pji} \\
 & \quad + 4[\mathcal{D}^{jppp} (2\mathbf{U}_p \mathbf{U}_p^t \mathbf{U}_j + \mathbf{U}_j \mathbf{U}_p^t \mathbf{U}_p) + \mathcal{D}^{jppp} \mathbf{U}_p \mathbf{U}_j^t \mathbf{U}_j \\
 & \quad \left. + (\mathcal{D}^{jpjp} + \mathcal{D}^{pjpi}) \mathbf{U}_j \mathbf{U}_j^t \mathbf{U}_p + \mathcal{D}^{jjip} \mathbf{U}_j \mathbf{U}_j^t \mathbf{U}_j \right\} \\
 & \underbrace{\hspace{10em}}_{\text{(Edge contribution)}} \\
 & + \sum_{\text{faces}(p,j,k)} \left\{ 2[(\mathbf{U}_k \otimes \mathbf{U}_j) \mathcal{C}^{jkp} + (\mathbf{U}_j \otimes \mathbf{U}_k) \mathcal{C}^{kjp} + (\mathbf{U}_j \cdot \mathbf{U}_k) \mathcal{C}^{pjk}] \right. \\
 & \quad + 4[(\mathcal{D}^{pjkp} + \mathcal{D}^{jpkp}) (\mathbf{U}_j \mathbf{U}_k^t \mathbf{U}_p + \mathbf{U}_k \mathbf{U}_j^t \mathbf{U}_p) + 2\mathcal{D}^{jkkp} \mathbf{U}_p \mathbf{U}_j^t \mathbf{U}_k \\
 & \quad + (\mathcal{D}^{kjip} + \mathcal{D}^{jkjp}) \mathbf{U}_j \mathbf{U}_j^t \mathbf{U}_k + \mathcal{D}^{jkkp} \mathbf{U}_k \mathbf{U}_j^t \mathbf{U}_j \\
 & \quad \left. + (\mathcal{D}^{ikkp} + \mathcal{D}^{kjkp}) \mathbf{U}_k \mathbf{U}_k^t \mathbf{U}_j + \mathcal{D}^{kkjp} \mathbf{U}_j \mathbf{U}_k^t \mathbf{U}_k \right\} \\
 & \underbrace{\hspace{10em}}_{\text{(Face contribution)}} \\
 & + \sum_{\text{tetra}(p,j,k,l)} \left\{ 4[(\mathcal{D}^{iklp} + \mathcal{D}^{kjlp}) \mathbf{U}_l \mathbf{U}_j^t \mathbf{U}_k + (\mathcal{D}^{ilkp} + \mathcal{D}^{ljkp}) \mathbf{U}_k \mathbf{U}_j^t \mathbf{U}_l \right. \\
 & \quad \left. + (\mathcal{D}^{kljp} + \mathcal{D}^{lkjp}) \mathbf{U}_j \mathbf{U}_k^t \mathbf{U}_l \right\}. \\
 & \underbrace{\hspace{10em}}_{\text{(Tetrahedron contribution)}}
 \end{aligned}$$

Appendix B. Update operations when removing a tetrahedron from the mesh

When we remove a tetrahedron from the mesh, we need to update stiffness parameters of 4 vertices, 6 edges, and 4 faces, which means 280 floating number operations (see Table 1):

$$4(1 \text{ tensor} + 1 \text{ vector} + 1 \text{ scalar}) + 6(1 \text{ tensor} + 4 \text{ vectors} + 5 \text{ scalars}) \\ + 4(3 \text{ vectors} + 9 \text{ scalars}) = 280 \text{ real numbers.}$$

References

- [1] N. Ayache, S. Cotin, H. Delingette, J.-M. Clement, J. Marescaux, M. Nord, Simulation of endoscopic surgery, *Journal of Minimally Invasive Therapy and Allied Technologies (MITAT)* 7 (2) (1998) 71–77.
- [2] J. Marescaux, J.-M. Clément, V. Tassetti, C. Koehl, S. Cotin, Y. Russier, D. Mutter, H. Delingette, N. Ayache, Virtual reality applied to hepatic surgery simulation: the next revolution, *Annals of Surgery* 228 (5) (1998) 627–634.
- [3] F. Boux de Casson, C. Laugier, Modeling the dynamics of a human liver for a minimally invasive surgery simulator, in: *Second International Conference on Medical Image Computing and Computer Assisted Intervention—MICCAI'99*, Cambridge UK, 1999, pp. 1156–1165.
- [4] D. Lamy, C. Chaillou, Design, Implementation and evaluation of an haptic interface for surgical gestures training, in: *International Scientific Workshop on Virtual Reality and Prototyping*, Laval, France, June 1999, pp. 107–116.
- [5] C. Basdogan, C. Ho, M.A. Srinivasan, S.D. Small, S.L. Dawson, Force interaction in laparoscopic simulation: haptic rendering soft tissues, in: *Medecine Meets Virtual Reality (MMVR'6)*, San Diego CA, January 1998, pp. 28–31.
- [6] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space and time adaptive sampling, in: *Proceedings of SIGGRAPH'01 SIGGRAPH 2001*, August 2001.
- [7] S. Cotin, H. Delingette, N. Ayache, Real-time elastic deformations of soft tissues for surgery simulation, *IEEE Transactions On Visualization and Computer Graphics* 5 (1) (1999) 62–73.
- [8] H. Delingette, S. Cotin, N. Ayache, A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation, in: *Computer Animation*, Geneva, Switzerland, IEEE Computer Society, May 26–28, 1999, pp. 70–81.
- [9] G. Picinbono, J.C. Lombardo, H. Delingette, N. Ayache, Anisotropic elasticity and force extrapolation to improve realism of surgery simulation, in: *IEEE International Conference on Robotics and Automation: ICRA 2000*, San Francisco, CA, April 2000, pp. 596–602.
- [10] Morten Bro-Nielsen and Stephane Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation, in: *Eurographics'96*. ISSN 1067-7055, Blackwell Publishers, 1996, pp. 57–66.
- [11] G. Székely, Ch. Brechbühler, R. Hutter, A. Rhomberg, P. Schmid, Modelling of soft tissue deformation for laparoscopic surgery simulation, *Medical Image Analysis* 4 (2000) 57–66.
- [12] G. Székely, M. Bajka, C. Brechbühler, J. Dual, R. Enzler, U. Haller, J. Hug, R. Hutter, N. Ironmonger, M. Kauer, V. Meier, P. Niederer, A. Rhomberg, P. Schmid, G. Schweitzer, M. Thaler, V. Vuskovic, G. Troster, in: *Virtual Reality Based Simulation of Endoscopic Surgery*, 9(3), Presence, MIT press, 2000, pp. 310–333.
- [13] Y.C. Fung, *Biomechanics—Mechanical Properties of Living Tissues*, second ed., Springer, Berlin, 1993.
- [14] W. Maurel, Y. Wu, N.M. Thalmann, D. Thalmann, *Biomechanical Models for Soft Tissue Simulation*, Springer, Berlin, 1998.
- [15] P.G. Ciarlet, *Mathematical Elasticity Vol. 1: Three-dimensional Elasticity*, Elsevier Science Publishers B.V., Amsterdam, 1988.
- [16] O. Zienkiewicz, *The Finite Element Method*, third ed., McGraw-Hill, London, 1977.
- [17] S. Cotin, H. Delingette, N. Ayache, A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation, *The Visual Computer* 16 (8) (2000) 437–452.
- [18] G. Picinbono, J.-C. Lombardo, H. Delingette, N. Ayache, Improving realism of a surgery simulator: linear anisotropic elasticity, complex interactions and force extrapolation, *Journal of Visualisation and Computer Animation* 13 (2002) 147–167.

- [19] K.-L. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [20] F.J. Carter, Biomechanical testing of intra-abdominal soft tissue, in: *International Workshop on Soft Tissue Deformation and Tissue Palpation*, Cambridge, MA, October 1998.
- [21] V. Vuskovic, M. Kauer, G. Székely, M. Reidy, Realistic force feedback for virtual reality based diagnostic surgery simulators, in: *IEEE International Conference on Robotics and Automation, ICRA 2000*, San Francisco, CA, April 2000, pp. 1592–1598.
- [22] D. Dan, *Caractérisation mécanique du foie humain en situation de choc*. PhD thesis, Université Paris 7, September 1999.