

Improving Realism of a Surgery Simulator : Linear Anisotropic Elasticity, Complex Interactions and Force Extrapolation

Guillaume Picinbono, Jean-Christophe Lombardo, Hervé Delingette, Nicholas Ayache

▶ **To cite this version:**

Guillaume Picinbono, Jean-Christophe Lombardo, Hervé Delingette, Nicholas Ayache. Improving Realism of a Surgery Simulator : Linear Anisotropic Elasticity, Complex Interactions and Force Extrapolation. [Research Report] RR-4018, INRIA. 2000, pp.37. inria-00072623

HAL Id: inria-00072623
https://inria.hal.science/inria-00072623

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Improving Realism of a Surgery Simulator: Linear Anisotropic Elasticity, Complex Interactions and Force Extrapolation

Guillaume Picinbono — Jean-Christophe Lombardo — Hervé Delingette —

Nicholas Ayache

N° 4018

Septembre 2000

THÈME 3



ISSN 0249-6399 ISRN INRIA/RR--4018--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Improving Realism of a Surgery Simulator: Linear Anisotropic Elasticity, Complex Interactions and Force Extrapolation

Guillaume Picinbono , Jean-Christophe Lombardo , Hervé Delingette , Nicholas Ayache

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Epidaure

Rapport de recherche n° 4018 — Septembre 2000 — 37 pages

Abstract: In this article, we describe the latest developments of the minimally invasive hepatic surgery simulator prototype developed at INRIA. The goal of this simulator is to provide a realistic training test bed for performing laparoscopic procedures. Therefore, its main functionality is to simulate the action of virtual laparoscopic surgical instruments for deforming and cutting tridimensional anatomical models. Throughout this paper, we present the general features of this simulator including the implementation of several biomechanical models and the integration of two force-feedback devices in the simulation platform. More precisely, we describe three new important developments that contribute to improve the overall realism of our simulator. First, we have developed bio-mechanical models, based on linear elasticity and finite element theory, that include the notion of anisotropic deformation. Indeed, we have generalized the linear elastic behavior of anatomical models to "transversally isotropic" materials, i.e. materials having a different behavior in one given direction. We have also added to the volumetric model an external elastic membrane standing for the "liver capsule", a quite stiff skin surrounding the liver and creating a kind of "surface anisotropy". Second, we have developed new contact models between surgical instruments and soft tissue models. For instance, after detecting a contact with an instrument, we define specific boundary constraints on deformable models to represent various forms of interactions with a surgical tool, such as sliding, gripping, cutting or burning. In addition, we compute the reaction forces that should be felt by the user manipulating the force-feedback devices. The last improvement is related to the problem of haptic rendering. Currently, we are able to achieve a simulation frequency of 25Hz (visual real-time) with anatomical models of complex geometry and behavior. But to achieve a good haptic feedback requires a frequency update of applied forces typically above 300Hz (haptic real-time). Thus, we propose a force extrapolation algorithm in order to reach haptic real-time.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 —Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Key-words: Simulation, elasticity, anisotropy, finite element method, deformable model, real-time, force feedback, extrapolation.

Améliorer le réalisme d'un simulateur de chirurgie: élasticité linéaire anisotrope, interactions complexes et extrapolation des forces

Résumé : Dans cet article, nous décrivons les derniers développements du prototype de simulateur de chirurgie hépatique minimalement invasive développé à l'INRIA. Le but de ce simulateur est de fournir un outil d'entraînement réaliste permettant de réaliser des opérations en laparoscopie. Sa principale fonctionnalité est donc de simuler la déformation et la découpe de modèles anatomiques tridimensionnels à l'aide de deux instruments de chirurgie laparoscopique virtuels. Tout au long de cet article, nous présentons les caractéristiques générales du simulateur, comprenant le développement de plusieurs modèles biomécaniques et l'intégration à la plate-forme de simulation de deux systèmes à retour d'effort. Nous décrivons plus précisément trois importants nouveaux développements qui contribuent à améliorer le réalisme global du simulateur. Premièrement, nous avons développé des modèles biomécaniques, basés sur les théories de l'élasticité linéaire et des éléments finis, qui intègrent la notion de déformation anisotrope. En effet, nous avons généralisé le comportement élastique linéaire de nos modèles anatomiques aux matériaux "transversallement isotropes", c'est à dire les matériaux qui ont un comportement différent dans une direction donnée. Nous avons aussi ajouté au modèle volumique une membrane élastique externe qui représente la "capsule de Glisson", une peau relativement rigide qui entoure le foie et qui crée une sorte d'"anisotropie de surface". Deuxièmement, nous avons développé de nouveaux modèles de contact entre les instruments chirurgicaux et les modèles de tissus mous. Par exemple, après avoir détecté un contact avec un instrument, nous définissons un ensemble de contraintes aux bords sur le modèle deformable afin de représenter différentes sortes d'interactions avec un instrument chirurgical, comme glisser, agripper, couper ou brûler. De plus, nous calculons les forces de réactions qui devraient être ressenties par l'utilisateur qui manipule les instruments à retour d'effort. La dernière amélioration est liée au problème du rendu haptique. Actuellement, nous sommes capable d'atteindre une fréquence de simulation de 25Hz (temps réel visuel) avec des modèles anatomiques à la géométrie et au comportement complexes. Mais réaliser un bon rendu haptique nécessite une fréquence de remise à jour des forces appliquées supérieure à 300Hz (temps réel haptique). Ainsi, nous proposons un algorithme d'extrapolation des forces permettant d'atteindre le temps réel haptique.

Mots-clés : Simulation, élasticité, anisotropie, méthode des éléments finis, modèle deformable, temps réel, retour d'effort, extrapolation.

Contents

1 Introduction **5**

2 Previous work **7**

3 Deformable models **8**
3.1 Linear elasticity 8
3.2 Anisotropic elasticity 9
3.3 Finite element formulation 12
3.4 Tensor/Mass model [17] 14
3.4.1 Numerical integration 14
3.4.2 Simulation of cutting 15
3.5 Pre-computed [13] and Hybrid models [17] 16
3.6 Results 16
3.6.1 Modeling of tubular structures 16
3.6.2 Introduction of anisotropic fibers into isotropic materials 18
3.7 Surface anisotropy 21
3.7.1 External spring-mass membrane 21
3.7.2 Results 22

4 Biomechanical validation **24**

5 Detection and definition of contacts with surgical instruments **24**
5.1 Collision detection: LCN method 25
5.2 Modeling contact with surgical instruments 25
5.3 Interesting features 28

6 Force feedback **30**
6.1 Force extrapolation 31
6.2 Results 31

7 Results: simulation of liver surgery **32**

8 Conclusion **34**

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 accomplished through small incisions. The abdomen of the patient 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. Special instruments are introduced through the other incisions, allowing surgeons to perform less invasive operations. The advantages for the patient are obvious: less pain and less strain for the organism, faster recovery, and shorter illness time. The drawback of this technique is 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 of a great help in the learning and training processes, allowing the surgeon to acquire this difficult hand-eyes coordination, to repeat several times the most difficult gestures or to choose the best surgical procedure for each pathological case.



Figure 1: *Surgery simulator prototype developed at INRIA*

There are several key problems in the development of a surgical simulator [2, 22]. To simulate a surgical gesture, it is necessary to build geometrical and physical models of the anatomical structures described in a simulator. Organs geometry may be extracted from tridimensional medical images (CT scan images for instance) of given patients. It may be useful to build a "standard" model of a given organ, in particular in the context of teaching surgical gestures. On the contrary, when the simulator is used for the rehearsing of a given surgical procedure, the anatomy of a given patient must be extracted from his medical images (see [24, 30] for an example on liver surgery).
Several mathematical models have been proposed in the literature [18, 23], for representing

the bio-mechanical behavior of biological soft tissues. The choice of a bio-mechanical model must take into account two contradictory elements : the simulation realism and the computation cost for implementing this model. The latter basically includes three distinct stages (figure 2). First, collisions between an anatomical structure and any surgical instrument must be detected. Then, boundary constraints on deformable models must be computed in order to realistically represent the action of surgical instruments. Finally, the geometry of soft tissue models must be updated and reaction forces must be sent to force-feedback devices.

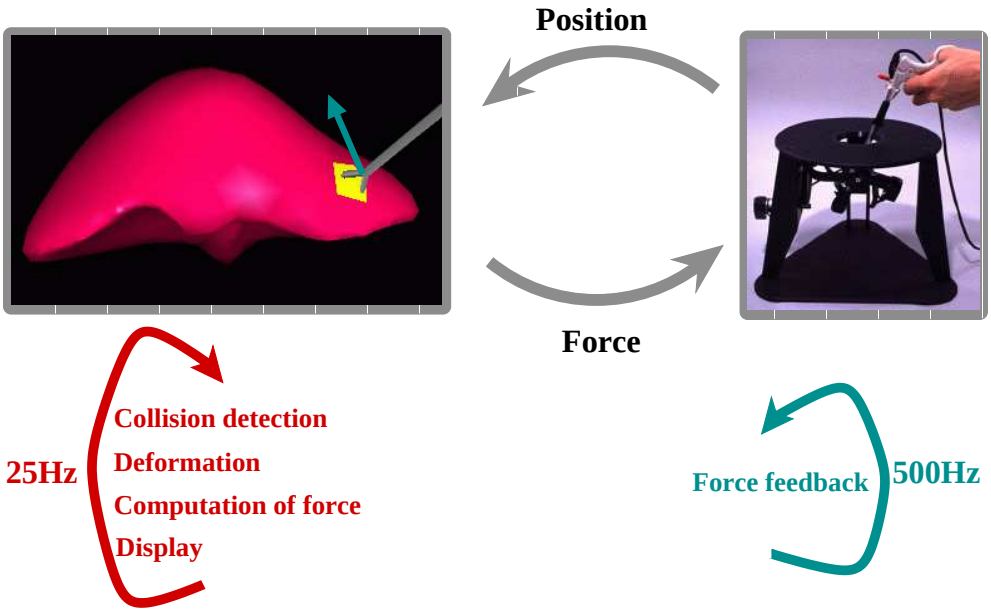


Figure 2: *Surgery simulator simplified work flow.*

This article focuses on the development of a laparoscopic surgery simulator prototype (figure 1) based on linear elasticity and finite elements theory and permitting the use of force feedback. Here we give a general description of three key components of this simulator. First, we describe a generalization of "tensor-mass" models [17] which can simulate visco-elastic soft tissue and which are well-suited for the simulation of cutting. Our contribution has been to improve these tensor-mass models by including "transversely anisotropic" behaviors of soft tissues. This is of high interest in a surgical simulator since most biological tissues behave as anisotropic material. We have also introduced another kind of anisotropy: a lot of human organs are surrounded by an elastic membrane or skin, which can be very stiff. We model this external stiffness by adding a spring-mass mesh at the surface of our volumetric model.

Second, we describe new contact models between laparoscopic surgical tools and a deformable soft tissue model. It mainly consists in applying specific boundary constraints where collisions have been detected, in order to represent common tools interactions such as sliding, gripping, cutting or burning.

Finally, we deal with the problem of haptic rendering: while up to date physically based deformable models are able to run at sufficient frequencies to provide a visual real time (about 25Hz), this is not the case for haptic rendering. It is commonly admitted that to give good haptic sensations, the forces must be refreshed at more than 300Hz for soft objects. We propose to extrapolate the forces produced by the simulation at a visual rate to feed the force feedback device at the appropriate frequency (typically 500Hz).

2 Previous work

A number of deformable models have been proposed in the framework of soft object simulation. In the particular case of real-time surgery simulation, the main challenge is to define a compromise between surgical realism and computation time.

Various computer animation techniques have been used in surgical simulators, mainly selected for their computational efficiency. For example Basdogan *et al.*[3] use "Free Form Deformations" in their laparoscopic simulator, the efficiency of this deformable model allowing them to use force feedback device under good conditions. Suzuki *et al.*[33], defined a particle system for their liver surgery simulator, coupled with their three fingers force feedback device. Unfortunately, even if these deformable models can be very fast, they are not mechanically realistic.

In surgery simulation, a great interest has been given to spring-mass models due to their simplicity of implementation and their relatively low computation complexity. For instance, Neumann *et al.*[25] present a vitrectomy simulator based on vectorial spring-mass model (in the case of vectorial springs forces not only depend on the variation of length of the springs, but also on the variation of angle between spring directions). Although in this case the user interactions are driven by surgical instruments (including tissue cutting), no force feedback is used. The simulator of knee arthroscopic surgery developed by Gibson *et al.*[19], takes into account the volumetric nature of the organs with a deformation law derived from spring-mass model. While these approaches allow interactive rates, they exhibit a lack of realism since they represent a solid as a discreet set of masses.

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 materials along with finite element method is widely described in the literature [35, 34, 29, 6, 5, 13]. Generally, various modifications and simplifications - e.g. linearization - are introduced to reduce computation time or to induce a particular behavior, as it is usually the case in surgical simulation. For instance, non-linear elastic models have been used recently for solving inverse problems [35] or for gynaecological laparoscopy simulation [34], but they are not well suited for real-time computation, or require very sophisticated and powerful parallel computation [29]. Even when using linear elasticity,

finite element based deformable models require optimization methods to reach real-time computation. Bro-Nielsen and Cotin [6, 5] have studied the problem of computing time reduction using a condensation technique. Cotin *et al.*[13] also proposed a method based on a set of pre-computed equilibrium solutions which allows real-time interactions with a volumetric deformable model of an organ. The main drawback of these two models is that they do not allow topology changes.

Indeed, topology changes usually occur when the tissue model is cut into pieces. These changes discard any method based on pre-computation of the inverse of the stiffness matrix since they induce a modification of this matrix [12]. The challenge is then to propose a formulation of a deformable elastic model allowing real-time deformations *and* cutting. Song and Reddy [31] 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 for this kind of problem, consists in using spring-mass models. By construction, these models can be easily modified to allow topology changes. However, they exhibit non realistic behavior during deformation or cutting operations. Finally, geometric models defined implicitly [9] have been proposed for the simulation of tearing but without any real-time constraints.

This work is a part of an INRIA joint action: to study all the problems related to surgery simulation, INRIA gathered six teams in the AISIM project [1]. To simulate the deformation of the organ, AISIM's members followed several approaches, including spring-mass models [15], finite element methods [13, 17], and hierarchical multi-resolution finite volume models [16]. To evaluate the validity of the simplifications made on real-time physical models, a non-linear incompressible model with an external shell [36] fitted with bio-mechanical results was developed. A new method was designed to detect collisions between a deformable object (the organ) and a rigid tool passing though a fixed point ([21]). Finally, the rendering processes, both visual and haptic were addressed. This led to a new technique to map undistorted textures on complex geometries [26]. Our solution to the haptic rendering issue [27] is described later in this paper.

3 Deformable models

3.1 Linear elasticity

The physical behavior of soft tissue may be considered as linear elastic if its displacement and deformation remain small [18, 23] (typically less than 10% of the mesh size). To describe a linear elastic model, we first need to 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:

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

With this displacement vector, we define the linearized *Green-St Venant strain tensor* (3×3 symmetric matrix) E by:

$$E = \frac{1}{2}\left(\nabla\mathbf{U} + \nabla\mathbf{U}^t\right). \tag{1}$$

From the principal invariants of E:

$$l_1 = tr\ E \qquad l_2 = tr\ E^2, \tag{2}$$

we can express the linear elastic energy $W_{Elastic}$, for homogeneous isotropic materials, by the following formula (see [11]):

$$W_{\text{Elastic}} = \frac{\lambda}{2}\,(trE)^2 + \mu\ trE^2, \tag{3}$$

where λ and μ are the *Lamé coefficients* characterising the stiffness of the material.

Equation 3, known as *Hooke's law*, shows that the elastic energy of a deformable object is a quadratic function of the displacement vector.

3.2 Anisotropic elasticity

Isotropic behavior is too restrictive when modeling human tissue. Indeed, most of anatomical structures such as muscles, tendons, ligaments, blood vessels, are strongly anisotropic. That is why we are interested in simulating materials having a different behavior in a given direction, which are called **transversally isotropic** materials.

A.J.M. Spencer gives a detailed theoretical description of **transversally isotropic** materials in [32]. In the literature we can find applications for modeling human organs such as the knee ligaments [38] and the eye [20], but without any real-time constraints.

We propose to apply the transversal isotropy to linear elasticity in the framework of real-time deformable models. For these materials, the elastic energy of equation 3 must be modified in order to account for the anisotropy. First, we explain in more details, the meaning of the strain tensor E.

$$E = \frac{1}{2}\begin{bmatrix} 2\,\epsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{xy} & 2\,\epsilon_y & \gamma_{yz} \\ \gamma_{xz} & \gamma_{yz} & 2\,\epsilon_z \end{bmatrix} \tag{4}$$

We consider an elementary cube (figure 3(a)) in $\mathcal{M}_{\text{initial}}$ and then look at its shape after applying the displacement $\mathbf{U}(x,y,z)$. The local cube deformation is characterized by six

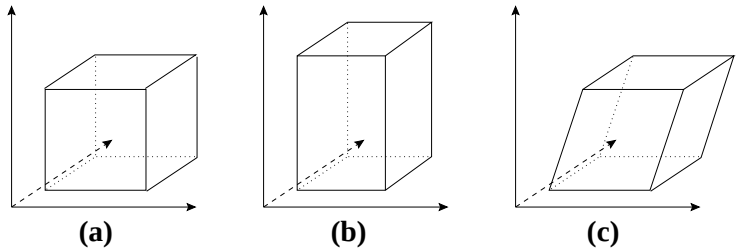


Figure 3: *Deformations of an elementary cube (a): elongation in cube direction (b) and change of angle between cube faces (c)*

components of strain corresponding to the relative elongations (figure 3(b)) ($\epsilon_x = \partial u/\partial x$, ϵ_y, and ϵ_z) in the three cube directions and the relative changes of angle ($\gamma_{xy} = \partial u/\partial y + \partial v/\partial x$, γ_{xz}, and γ_{yz}) between the cube faces (figure 3(c)).

The isotropic elastic energy of equation 3 can be written as a function these components of strain :

$$\begin{aligned} W_{Elastic} = & \quad \frac{\lambda}{2}\left(\epsilon_x + \epsilon_y + \epsilon_z\right)^2 + \mu\left(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\right) \\ + & \quad \frac{\mu}{2}\left(\gamma_{xy}^2 + \gamma_{xz}^2 + \gamma_{yz}^2\right). \end{aligned} \tag{5}$$

This energy is isotropic since the same weight is given to each direction of stretch and shear. For transversally isotropic materials, it is necessary to define two sets of Lamé constants:

- (λ^L, μ^L): **Longitudinal** Lamé constants in a given direction having unitary vector $\mathbf{a}_0$;
- (λ^T, μ^T): **Transverse** Lamé constants in the plane transversal (i.e. orthogonal) to $\mathbf{a}_0$;
- $\Delta\lambda = \lambda^L - \lambda^T$ and $\Delta\mu = \mu^L - \mu^T$.

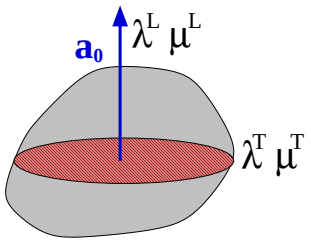


Figure 4: *Transversally isotropic material*

For instance, if the z axis is the direction of anisotropy $\mathbf{a}_0 = (0 \quad 0 \quad 1)$, then we need to add to the isotropic energy of equation 5, the anisotropic contribution ΔW_{Aniso}.

This anisotropic contribution is composed of two different parts : the **anisotropic stretch** and the **anisotropic shear**.

- **Anisotropic stretch :** In this case we must modify the weight of the terms containing the z axis elongation, that is to say the terms containing ${\epsilon_z}^2$, $\epsilon_x\epsilon_z$ or $\epsilon_y\epsilon_z$. We obtain two anisotropic contributions :

$$\Delta W_{\mathrm{A}niso-Stretch} = \left(\frac{\Delta\lambda}{2} + \Delta\mu\right) {\epsilon_z}^2 \quad (6)$$

$$\Delta W_{\mathrm{A}niso-Cross-Stretch} = \Delta\lambda\ (\epsilon_x\epsilon_z + \epsilon_y\epsilon_z) \quad (7)$$

  In the general case, these anisotropic elastic energies can be simply written with the introduction of the new invariant l_4 linked to the strain tensor E and direction $\mathbf{a}_0$ by :

$$l_4 = {\mathbf{a}_0}^t\, E\, \mathbf{a}_0 \quad (8)$$

  Then the anisotropic stretch energies can be written as:

$$\Delta W_{\mathrm{A}niso-Stretch} = (\frac{\Delta\lambda}{2} + \Delta\mu)\ {l_4}^2 \quad (9)$$

$$\Delta W_{\mathrm{A}niso-Cross-Stretch} = \Delta\lambda\ \left(l_1 l_4 - {l_4}^2\right) \quad (10)$$

- **Anisotropic shear :** If we want to take into account the shear part of the anisotropic deformation, we must also modify the terms containing γ_{xz} and γ_{yz}, by adding another anisotropic contribution :

$$\Delta W_{\mathrm{A}niso-Shear} = \frac{\Delta\mu}{2}\ ({\gamma_{xz}}^2 + {\gamma_{yz}}^2). \quad (11)$$

  If we define another invariant l_5 :

$$l_5 = {\mathbf{a}_0}^t\, E^2\, \mathbf{a}_0 \quad (12)$$

  this anisotropic shear contribution becomes :

$$\Delta W_{\mathrm{A}niso-Shear} = \frac{\Delta\mu}{2}\ \left(l_5 - 4\,{l_4}^2\right) \quad (13)$$

Therefore, the anisotropic elasticity formulation can be decomposed into two stretch terms and one shear term. By weighting differently the stretch and shear components in the anisotropic energy, we can model several mechanical behaviors (see part 7 for more results).

For instance, to model a complete anisotropic model, with both stretch and shear contributions, we must add the following energy :

$$\Delta W_{\mathrm{A}niso} = \Delta\lambda\, l_1 l_4 + 2\Delta\mu\, l_5 - (\frac{\Delta\lambda}{2} + \Delta\mu)\, l_4^2. \quad (14)$$

Finally, the total elastic energy of a transversally isotropic material is:

$$W_{\mathrm{T}rans-Iso} = W_{\mathrm{E}lastic} + \Delta W_{\mathrm{A}niso}. \quad (15)$$

3.3 Finite element formulation

Our deformable models are based on a finite elements model consisting of a conformal tetrahedral mesh with first order interpolation (P_1 element). 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 the vertices $\mathbf{P}_k$:

$$\mathbf{U}(x,y,z) = \sum_{k=0}^{3} \Lambda_k(x,y,z)\ \mathbf{U}_k$$

$$\text{with } \mathbf{O}M = \sum_{k=0}^{3} \Lambda_k(x,y,z)\ \mathbf{O}P_k$$

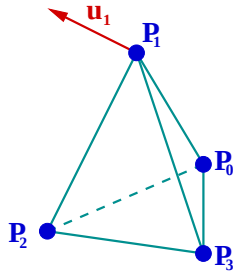


Figure 5: P_1 *finite element*

where Λ_k are the barycentric coordinates of $\mathbf{M}$ in the tetrahedron.

First, we can write the two invariants used in elastic energy formula, ${l_1}^2$ and l_2, with the finite element method :

$${l_1}^2 = (tr\ E)^2 = \sum_{k,l} \mathbf{U}_k^t\ (\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l)\ \mathbf{U}_l \tag{16}$$

$$l_2 = tr\ E^2 = \sum_{k,l} \mathbf{U}_k^t\ [(\boldsymbol{\alpha}_l \otimes \boldsymbol{\alpha}_k) + (\boldsymbol{\alpha}_k\ .\ \boldsymbol{\alpha}_l) Id_3)]\ \mathbf{U}_l \tag{17}$$

where:

$$\left\{ \boldsymbol{\alpha}_k = \frac{(-1)^k}{6V(\mathbf{T}_i)} (\mathbf{P}_{k+1} \times \mathbf{P}_{k+2} + \mathbf{P}_{k+2} \times \mathbf{P}_{k+3} + \mathbf{P}_{k+3} \times \mathbf{P}_{k+1}), k = 0, ..., 3 \right\}$$

are the shape vectors of the tetrahedron, $\times$ is the cross product, and $\otimes$ is the tensor product of two vectors which gives a matrix: $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\ \mathbf{v}^t$.

We can then write the **isotropic linear elastic energy** of tetrahedron $\mathbf{T}_i$ as a function of its vertex displacements:

$$W_{Elastic} = \sum_{k=0}^{3} \sum_{l=0}^{3} \mathbf{U}_k^t [K_{kl}^{\mathbf{T}_i}] \mathbf{U}_l, \tag{18}$$

where

$$[K_{kl}^{\mathbf{T}_i}] = \lambda\ (\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l) + \mu\ (\boldsymbol{\alpha}_l \otimes \boldsymbol{\alpha}_k) + \mu\ (\boldsymbol{\alpha}_k\ .\ \boldsymbol{\alpha}_l)\ Id_3$$

is the tetrahedron contribution to the stiffness tensor of edge $(\mathbf{P}_k, \mathbf{P}_l)$ or of vertex $\mathbf{P}_k$ if $k = l$. Likewise, we obtain a similar equation for the anisotropic invariants ${l_4}^2$, $l_1 l_4$ and l_5 :

$$\begin{aligned}
{l_4}^2 &= \left(\mathbf{a_0}^t\, E\, \mathbf{a_0}\right)^2 = \sum_{k,l} \mathbf{U}_k^t \left(\mathbf{a_0} \otimes \mathbf{a_0}\right) \left(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l\right) \left(\mathbf{a_0} \otimes \mathbf{a_0}\right) \mathbf{U}_l \\
l_1 l_4 &= tr\, E(\mathbf{a_0}^t\, E\, \mathbf{a_0}) = \sum_{k,l} \mathbf{U}_k^t \left(\mathbf{a_0} \otimes \mathbf{a_0}\right) \left(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l\right) \mathbf{U}_l \qquad (19) \\
l_5 &= \mathbf{a_0}^t\, E^2\, \mathbf{a_0} = \sum_{k,l} \mathbf{U}_k^t \left[\left(\mathbf{a_0} \otimes \mathbf{a_0}\right) \left(\boldsymbol{\alpha}_l \otimes \boldsymbol{\alpha}_k\right) + \left(\boldsymbol{\alpha}_l \otimes \boldsymbol{\alpha}_k\right) \left(\mathbf{a_0} \otimes \mathbf{a_0}\right)\right. \\
&\qquad \left. + \left(\boldsymbol{\alpha}_l . \boldsymbol{\alpha}_k\right) \left(\mathbf{a_0} \otimes \mathbf{a_0}\right) + \left(\mathbf{a_0} \otimes \mathbf{a_0}\right) : \left(\boldsymbol{\alpha}_l . \boldsymbol{\alpha}_k\right) Id_3\right] \mathbf{U}_l \qquad (20)
\end{aligned}$$

where : is the dot product of two matrices, which is defined, for two matrices A and B by: $A : B = \sum_{i,j} A_{ij} B_{ij}$.

For a complete anisotropic model (stretch and shear), the elastic energy of tetrahedron $\mathbf{T}_i$ is :

$$\Delta W_{\mathrm{A}niso} = \sum_{k=0}^{3} \sum_{l=0}^{3} \mathbf{U}_k^t [A_{kl}^{\mathbf{T}_i}] \mathbf{U}_l, \qquad (21)$$

$$\begin{aligned}
A_{kl}^{\mathbf{T}_i} &= \Delta\lambda (\mathbf{a_0} \otimes \mathbf{a_0})(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l) - (\frac{\Delta\lambda}{2} + \Delta\mu)(\mathbf{a_0} \otimes \mathbf{a_0})(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l)(\mathbf{a_0} \otimes \mathbf{a_0}) \\
&+ \frac{\Delta\mu}{2} [(\mathbf{a_0} \otimes \mathbf{a_0})(\boldsymbol{\alpha}_l \otimes \boldsymbol{\alpha}_k) + (\boldsymbol{\alpha}_l \otimes \boldsymbol{\alpha}_k)(\mathbf{a_0} \otimes \mathbf{a_0}) + (\boldsymbol{\alpha}_k \,.\, \boldsymbol{\alpha}_l)(\mathbf{a_0} \otimes \mathbf{a_0}) \\
&\qquad + (\mathbf{a_0} \otimes \mathbf{a_0}) : (\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_l) Id_3] \qquad (22)
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{F}_k^{\mathbf{T}_i} &= 2 \sum_{l=0}^{3} [K_{kl}^{\mathbf{T}_i} + A_{kl}^{\mathbf{T}_i}] \mathbf{U}_l \\
&= [G_{kk}^{\mathbf{T}_i}] \mathbf{U}_k + \sum_{l=0; l \neq k}^{3} [G_{kl}^{\mathbf{T}_i}] \mathbf{U}_l \quad \text{where} \quad G_{kl}^{\mathbf{T}_i} = K_{kl}^{\mathbf{T}_i} + A_{kl}^{\mathbf{T}_i} \qquad (23)
\end{aligned}$$

This equation gives only the force applied by **one** tetrahedron on the vertex $\mathbf{P}_k$. We obtain the global force applied on vertex a $\mathbf{P}_k$ by adding the contributions of all the tetrahedra sharing this vertex:

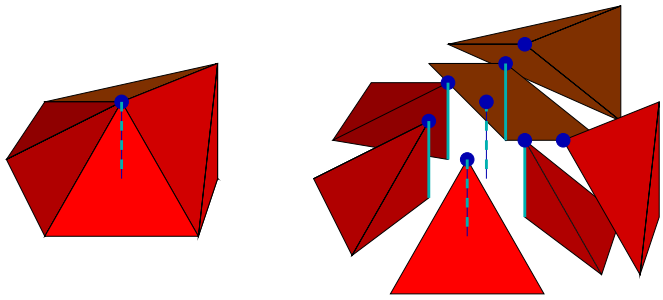


Figure 6: *Accumulation of the stiffness tensors on the vertices and the edges*

$$\mathbf{F}_k = [G_{kk}]\mathbf{U}_k + \sum_{\mathbf{P}_l \in N(\mathbf{P}_k)} [G_{kl}]\mathbf{U}_l, \tag{24}$$

$$G_{kk} = \sum_{\mathbf{T}_i \in N(\mathbf{P}_k)} G_{kk}^{\mathbf{T}_i} \quad : \quad \text{is the global stiffness tensor of vertex } \mathbf{P}_k$$

$$G_{kl} = \sum_{\mathbf{T}_i \in N(\mathbf{P}_k, \mathbf{P}_l)} G_{kl}^{\mathbf{T}_i} \quad : \quad \text{is the global stiffness tensor of edge } \mathbf{P}_k\mathbf{P}_l.$$

Thus, $\mathbf{F}_k$ are linear functions of the displacement vectors of node $\mathbf{P}_k$ and its mesh neighbors $N(\mathbf{P}_k)$.

3.4 Tensor/Mass model [17]

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 its neighboring tetrahedra, its current position $\mathbf{P}_k$, its rest position $\mathbf{P}_k^0$, and its tensor $[G_{kk}]$. For each edge, we store its two vertices as well as its tensor $[G_{kl}]$. Finally for each tetrahedron, we store its four vertices, its six edges as well as the Lamé coefficients $\lambda_k^L, \mu_k^L, \lambda_k^T, \mu_k^T$, the direction of anisotropy a_0, and the four shape vectors $\boldsymbol{\alpha}_k$.

3.4.1 Numerical integration

We use a Newtonian differential equation:

$$m_i \frac{d^2\mathbf{P}_i}{dt^2} = \gamma_i \frac{d\mathbf{P}_i}{dt} + \mathbf{F}_i \tag{25}$$

as the equation governing the motion of our linear elastic model. This equation is related to the differential equation found in continuum mechanics [4]:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}. \tag{26}$$

Following finite elements theory, the mass **M** and damping **C** matrices are sparse matrices that are related to the stored physical properties of each tetrahedron. In our case, we consider that **M** and **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 equation 26 as the set of independent differential equations (25) 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 $t+1$:

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

3.4.2 Simulation of cutting

One of the basic tasks in surgery simulation consists in cutting soft tissue. With the dynamic linear elastic 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.

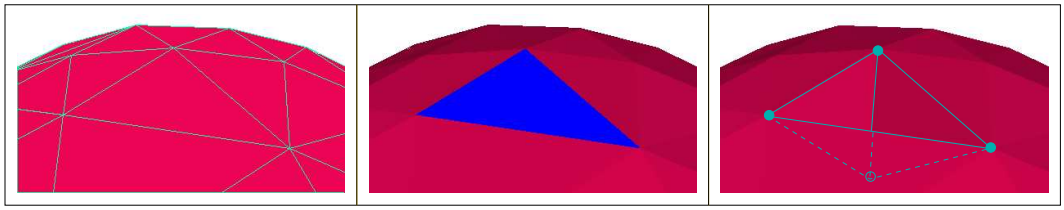


Figure 7: *Possibility to change mesh topology "on line"*

When a collision between the instrument and a tetrahedron is detected, local deformation tensors associated with the tetrahedron are subtracted from the current deformation tensors at the tetrahedron edges and vertices (figure 7). Since the tensors are only updated locally, this is performed in a very efficient manner. For instance, when removing tetrahedron T_i, 10 update operations are performed:

$$[G_{jj}] = [G_{jj}] - [G_{jj}^{T_i}] \qquad [G_{jk}] = [G_{jk}] - [Gx_{jk}^{T_i}].$$

Finally, we update the list of displayed triangles located on the mesh surface. By locally updating tensors, 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 is realistic during the cutting.

3.5 Pre-computed [13] and Hybrid models [17]

The most common method to solve the elasticity problem formulated in equation 24 is to build the global linear system $[K]\ \mathbf{U} = \mathbf{F}$ which gives the displacement of all nodes as a function of boundary conditions and external forces. Unfortunately, such linear systems cannot be solved fast enough for complex objects. The main idea is to pre-compute unitary deformations of each node and to decompose, during the simulation, any deformation into a sum of unitary deformations. This method allows real-time simulation for complex objects like anatomical structures but cannot model any change of mesh topology. This drawback led us to develop a new model. This combination of the two previous deformable models have been proposed in [17]: a hybrid model combines a large pre-computed model with a smaller tensor-mass model allowing for topology changes.

3.6 Results

In this section, we give two examples of anisotropic material modeling. In the first example we give a direct application of our "transversally isotropic" deformable model to simulate the biomechanical behavior of tubular structures like blood vessels or tendons, and the second example highlights the effects of introducing anisotropic fibers into isotropic materials.

3.6.1 Modeling of tubular structures

In this first example, we consider two identical cylinders: one is isotropic (figure 8a and 8c) and the other is 5 times stiffer in the z direction (figure 8b and 8d). In this case, we have included the complete anisotropic energy with both stretching and shearing terms. First, the cylinders are pinched by applying forces perpendicular to the anisotropic direction. We can see that the range of the deformation is the same for the two models. Furthermore the deformation is more realistic in the second case because it is smoothed along the anisotropic direction (because of the shear part of anisotropic energy). On the second set of cylinders (figure 8c and 8d) we can see the deformation resulting from the application of the same force in the z direction on the top face of each cylinder (the bottom face is fixed). As expected, the model is much stiffer in the anisotropic direction, and the stretch is much smaller.

To highlight the difference between the stretch and shear parts of the anisotropic energy, we present another result on figure 10 : (a) is an isotropic cylinder and (b) is anisotropic with only the stretching term. The superimposition of the two deformations (figures 10(c) and 10(d)) shows that the deformation terms are equivalent for transversal constraints. On the other hand, the deformation of the last cylinder (10(e)), which is anisotropic in stretch and shear, is somewhat smoother. In fact, the shearing term of anisotropic energy penalizes the variation of angle between faces along the anisotropic direction. This leads to a smoother distribution of the shear along this direction (see figure 9).

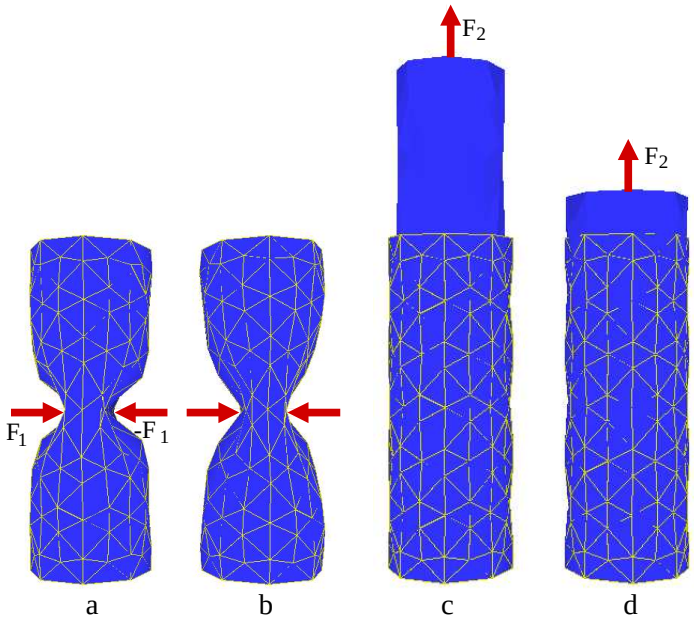


Figure 8: *Comparison between isotropic (a and c) and anisotropic (b and d) cylinders*

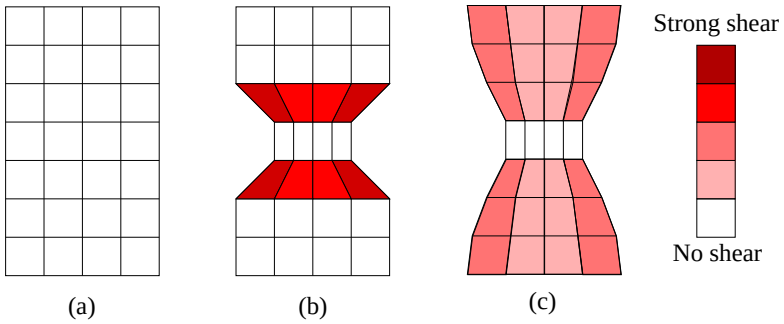
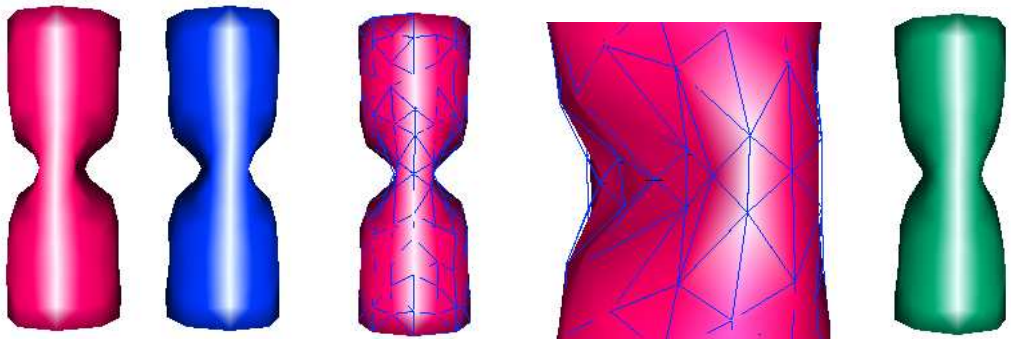


Figure 9: *Shear: variation of angle between edges. (a): rest shape. (b): with isotropic material, shear can be locally very strong. (c): with anisotropic shearing term, the shear is weaker and smoothly distributed along the anisotropic direction.*



(a) (b) (c) (d) (e)

Figure 10: *Comparison between isotropic (a), anisotropic in stretch (b) and fully anisotropic (stretch and shear) (e) elasticity*

3.6.2 Introduction of anisotropic fibers into isotropic materials

Besides the modeling of anisotropic anatomical structures, our "Transversally isotropic" tensor-mass model also allows to include such structures into isotropic materials, simply by adding the anisotropic terms only in some tetrahedra, the rest of the deformable model remaining perfectly isotropic. In the example shown on figure 11, we have defined an anisotropic strip (darker part) in an isotropic elastic material (we have chosen a simple geometrical model for this example in order to have a better understanding of the effects of the anisotropic parts on the overall deformation). This anisotropic part could model the presence of fibers (blood vessels, muscles or tendons), with the direction of anisotropy corresponding to the largest dimension of the stripe. First we have stretched the model along the direction of anisotropy. Figure 11(a) shows the result of this stretch and figure 11(b) focuses on the local shear at the frontier between isotropic and anisotropic materials. Then we have stretched the model in the direction perpendicular to the direction of anisotropy, to check that, in this direction, the stretch is the same as in the isotropic case (figure 11(c)). Figure 11(d) confirms that the anisotropic part hardly deforms in the fiber direction.

Finally, figure 12 shows the modeling of the blood vessels in the liver. In this example, the geometry of the vessels is built as a tree made of seven straight lines. Each tetrahedron containing on of these lines has a transversally isotropic behavior (figures 12(a) and 12(b)), the direction of the line being the direction of anisotropy (figure 12(c)). We obtain a deformable model of the liver that takes into account the anisotropic mechanical properties of blood vessels (figures 12(d), 12(e), and 12(f)).

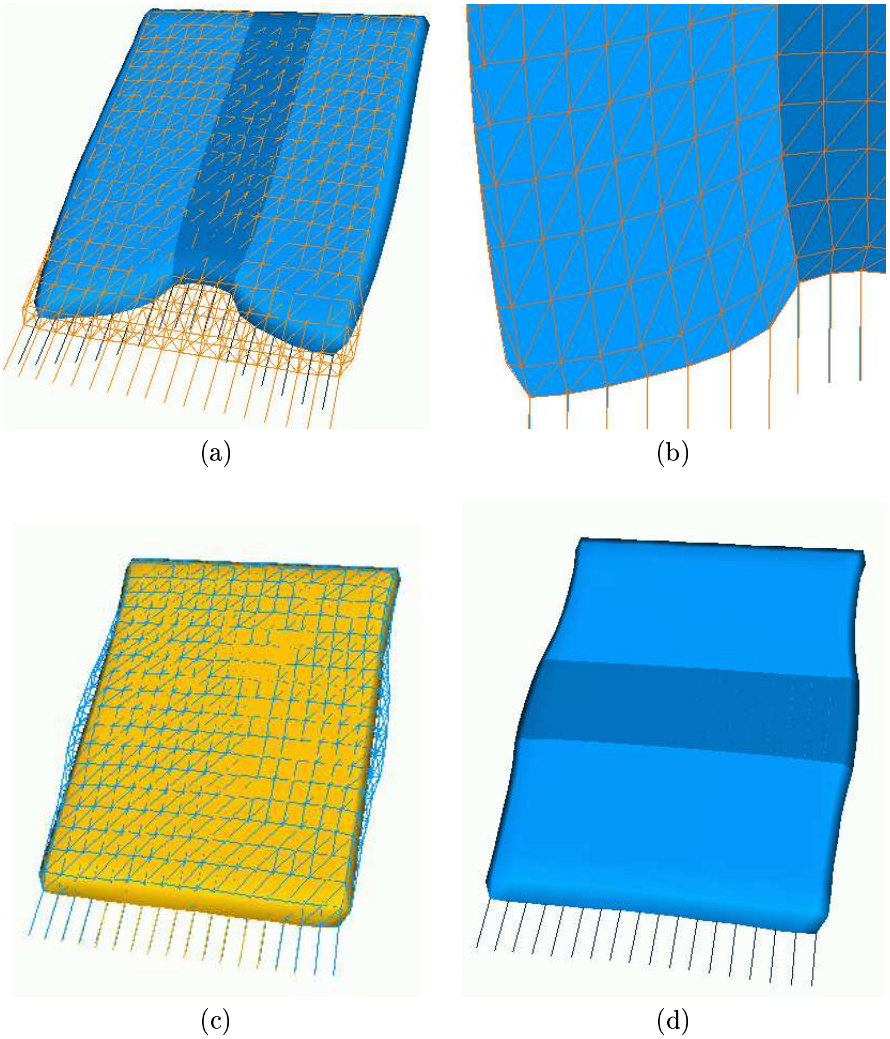


Figure 11: *Effects of the introduction of anisotropic fibers in isotropic materials. The darker part of the solid model (in blue) on (a), (b) and (c) is anisotropic with the direction of anisotropy corresponding to the largest dimension of the stripe. Its deformations along the direction of anisotropy ((a) and (b)) and orthogonally to this direction ((c) and (d)) are compared with an entirely isotropic model (in wireframe on (a) and solid rendering on (c))*

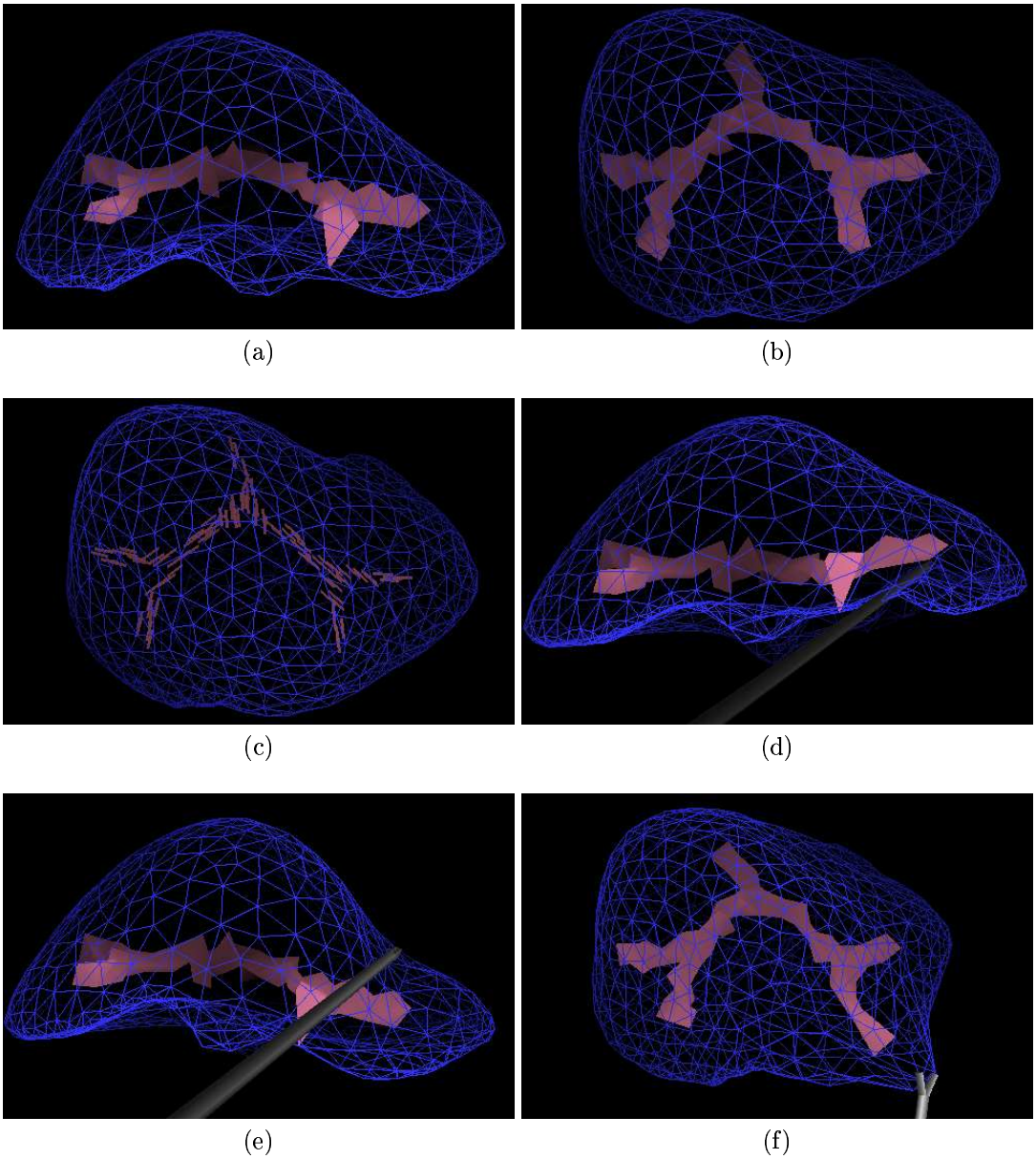


Figure 12: *Modeling of the main vessels of the liver using anisotropic elasticity in the tetrahedra containing these vessels*

3.7 Surface anisotropy

Several anatomical structures have an external skin or membrane. In particular, the liver has an external "capsule" or membrane. The goal of this section is to model this specific skin in order to improve the realism of the liver deformations.

3.7.1 External spring-mass membrane

The "liver capsule" is quite stiff, and plays an important role in the deformation of the liver ([15, 36]). As we do not want to slow down the simulation frequency, we choose to add to our tensor-mass model a spring-mass mesh on top of external vertices, edges and faces of the volumetric mesh. The same method is use by Boux de Casson *et al.* in [15], who define two different sets of springs, one for the liver paremchyma and the other, stiffer, for the liver capsule. The working scheme for spring-mass deformable models is very close from tensor-mass one. Each edge of the surface behaves like a spring, with a rest length l_0 and a stiffness coefficient k. At each time step we only have to compute, for each edge of the surface, its current length l and the forces this spring applies to its two vertices $\mathbf{V}_j$ and $\mathbf{V}_k$ (see figure 13).

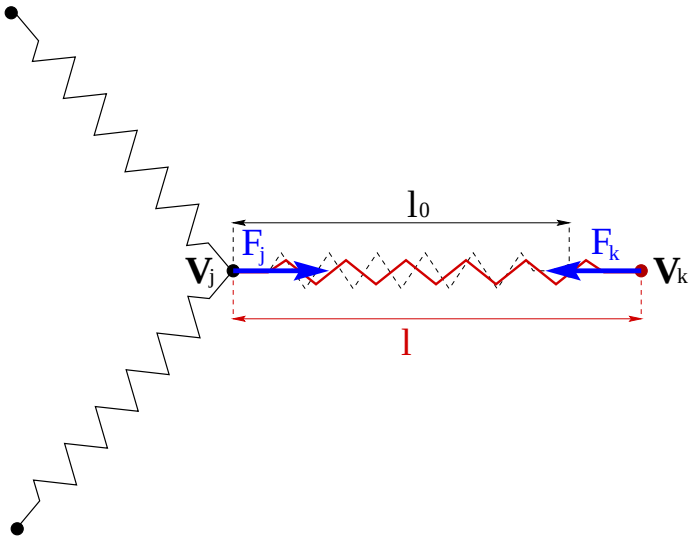


Figure 13: *spring-mass mesh*

These forces are derived from the elastic energy of a spring given by:

$$W_{spring} = \frac{1}{2}k(l - l_0)^2, \tag{27}$$

and only depend on the stiffness k and the length variation:

$$\mathbf{F}_j^{spring} = -\mathbf{F}_k^{spring} = k\left(\frac{l - l_0}{l}\right)\overrightarrow{\mathbf{V}_j\mathbf{V}_k}. \tag{28}$$

The goal of this external membrane being to penalize the stretch of the surface of the organ, we use a stiffness value k proportional to the sum of the altitudes (see figure 14) of the two triangles sharing out this edge. If the direction of the edge does not change too much, the force applied by the spring on its vertices is a good approximation of the variation of area of the two adjacent triangles (see figure 14):

$$\begin{aligned} k &= \alpha\, h \\ \mathbf{F}_j^{spring} &= k\,\Delta l\,\vec{T}_{jk} = \alpha\,(h\,\Delta l)\,\vec{T}_{jk} = \alpha\,\Delta S\,\vec{T}_{jk} \end{aligned} \tag{29}$$

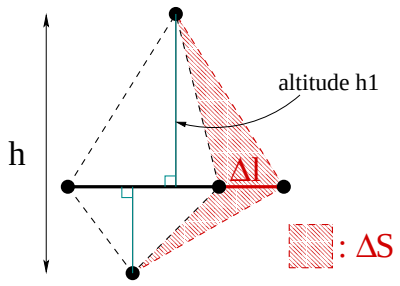
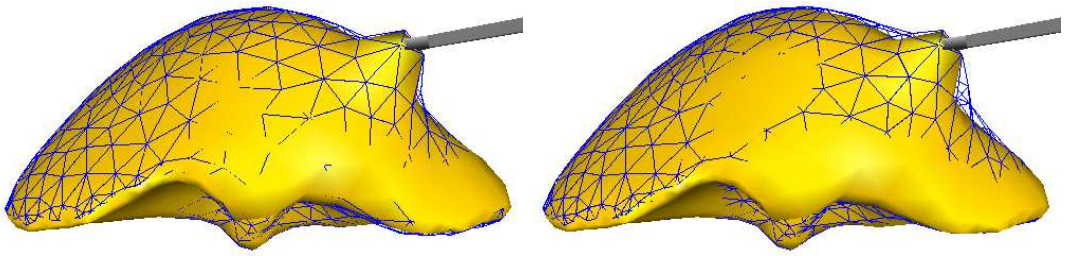


Figure 14: *Stiffness of each spring of the surface*

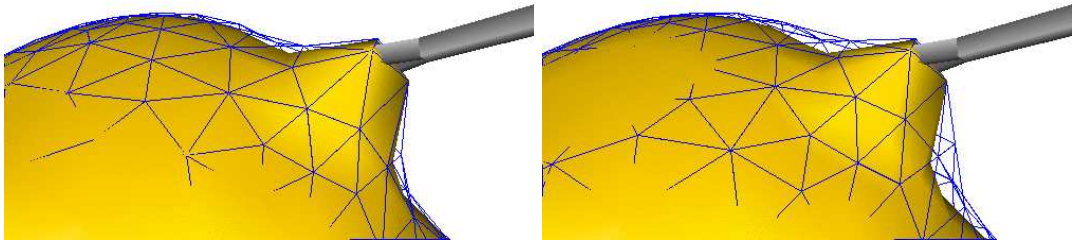
3.7.2 Results

We have implemented this additional spring-mass membrane on our tensor-mass model. The simulation frequency is almost unchanged (less than 4% decrease) since the additional computations are very simple and only concern the edges of the surface.

Figure 15 shows the comparison between an isotropic tensor-mass model (in solid rendering on figure 15(a) et (b)) and the same model on which we have added the spring-mass membrane (in wireframe rendering on figure 15(a) et (b)). Figures (a) and (b) of figure 15 show the local deformation differences brought by the external membrane (between left and right of figure 15(a) and (b), the stiffness of the membrane, i.e. the stiffness of the springs, was multiplied by 10). The introduction of the external spring-mass membrane around the liver improves the anatomical modeling of the liver, and increase the realism of the simulated deformations.



(a) The model with wire-frame rendering has an external spring-mass membrane, while the other (solid rendering) is isotropic. Between left and right figures, the stiffness of the springs of the membrane was multiplied by 10



(b) Close look on the experiments presented above

Figure 15: Comparison of the deformations between an isotropic model and the same model with an external membrane standing for the "liver capsule"

4 Biomechanical validation

The liver is filled with blood (near 50% of its global weight), which implies that ex-vivo liver tissue should not behave like an in-vivo tissue. 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 Poison ratio) must be measured in-vivo, or using complex in-vitro experimental protocols. In both cases, the results obtained by several teams (F. Carter *et al.* [10], V. Vuskovic [37], D. Dan [14]) have showed a strong variability in the parameter 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 behaviour of our liver model. Linear elasticity appears to be reasonable for small displacements. For large displacements, we are also working on a non-linear modeling of deformations [28].

5 Detection and definition of contacts with surgical instruments

We use the deformable models presented above in our laparoscopic surgery simulator. An important stage in the simulation loop corresponds to the detection and processing of the contact between a laparoscopic instrument and soft tissue models. In this section, we first describe how the collision with an instrument can be efficiently detected using accelerated graphics hardware. Then, we present the different contact models that we have implemented, corresponding to the action of gripping, cutting, sliding or burning.

Figure 16 shows the modeling of a basic laparoscopic instrument which consists of a long and narrow cylinder, terminated with a hook or pliers, passing through a fixed point corresponding to the hole by which the instrument is inserted inside the patient abdomen.

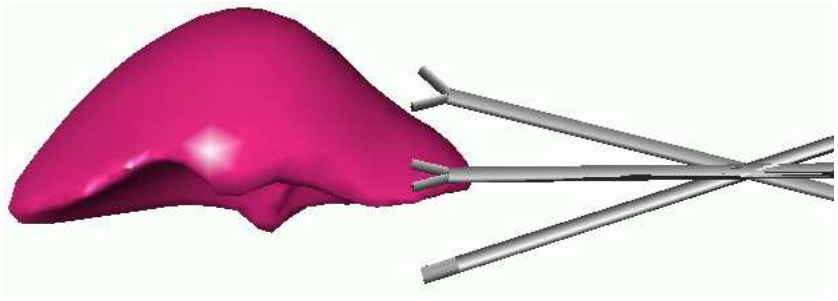


Figure 16: *Modeling of a basic laparoscopic instrument.*

5.1 Collision detection: LCN method

The main idea of this method is to use the OpenGL library to detect the intersection of a complex scene with a polyhedral object consisting of a very limited number of faces (limited by the number of clipping planes). The polyhedral object is represented by an OpenGL camera with a set of clipping planes, and the collision detection is performed with the OpenGL selection buffer technique (more details on this method can be found in [21]).

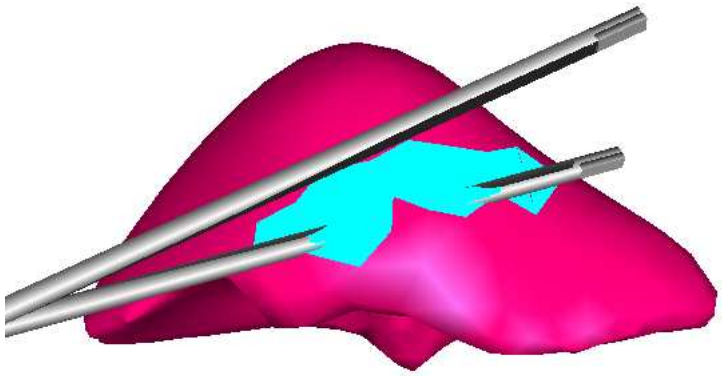


Figure 17: *Collision detection between a moving surgical instrument and a liver model.*

This approach is very well suited for detecting contacts between soft tissue models and laparoscopic surgical instruments since they can be represented as a single box. Furthermore, it has two main advantages. First, it is rather simple as it can be implemented with no more than twenty lines of OpenGL code. Second, this method is very efficient, especially when using dedicated (accelerated) graphics hardware. For instance, on an SGI Onyx2 workstation, the collision detection takes less than 0.2 ms. Even on a PC Pentium2 running at 333MHz under linux and using the Mesa 3D software library with no graphics hardware, the collision detection typically takes less than 3.0 ms ([21]).

5.2 Modeling contact with surgical instruments

The collision detection module outputs a set of triangles colliding with an instrument. The next task is to modify the geometry of these triangles in order to model the contact between the rigid laparoscopic instrument and a soft tissue model. A common technique described in the literature is the penalty method where a force proportional to penetration depth is applied on all triangles collided by the tool. The problem with this method is that it does not really respect the geometry of the contact. In fact, it does not guarantee that the

collided faces will exactly follow the displacement of the tool. Some faces can be penetrated by the tool during several time steps or can be pushed too far away, which creates some discontinuity and instability problems. Instead, we choose a geometric approach rather than a physical approach: we apply precise displacements to the colliding faces. The main idea in our approach is to compute these displacements so that all triangles stay in contact with the instrument. Indeed, this avoids the soft tissue model to oscillate inside and outside the instrument. Finally, we must keep in mind that the computation of these displacements must be simple enough to keep the simulation frequency as high as possible.

Our goal being to model the contact between the whole tool and the organ, we first consider the collision with the handle of the tool. In this case the problem is to move a set of adjacent triangles in such a way that they all contain the same straight line (the edge of the tool). The only solution is to project every vertex of the collided faces on a plane containing the straight line (see figure 18 and 20(a)).

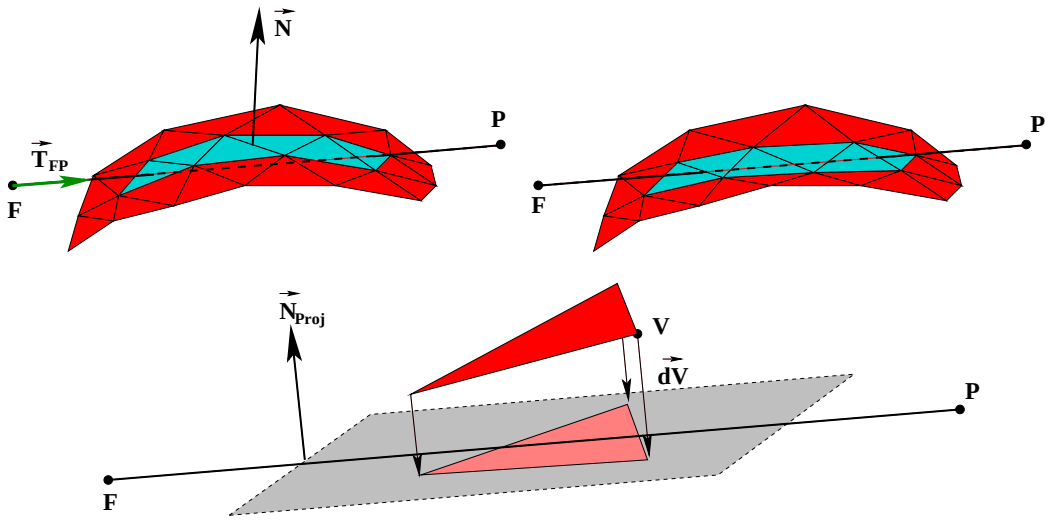


Figure 18: *Definition of the projection plane, and projection of the collided faces on it*

For that, we define the projection plane with the direction of the handle of the tool ($\vec{\mathbf{T}}_{FP}$ of figure 18) and the average normal of all the collided faces $\vec{\mathbf{N}}$. From these two vectors we define the normal to the projection plane by:

$$\vec{\mathbf{N}}_{Proj} = \left(\vec{\mathbf{T}}_{FP} \wedge \vec{\mathbf{N}}\right) \wedge \vec{\mathbf{T}}_{FP},$$

and we project the vertices orthogonally on this plane:

$$\mathbf{d}\vec{\mathbf{V}} = \left(\mathbf{P}\vec{\mathbf{V}} \,.\, \vec{\mathbf{N}}_{Proj}\right) \vec{\mathbf{N}}_{Proj}.$$

But there are two problems with this method. First it does not perform well when the tool is far from being parallel to the surface. In this case only the end of the tool collides the surface, involving an unrealistic deformation (see 2D example of figure 19). The second

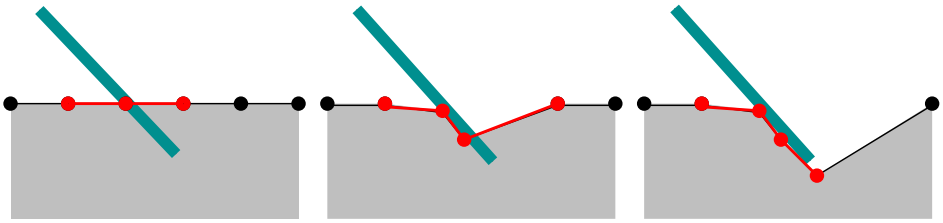
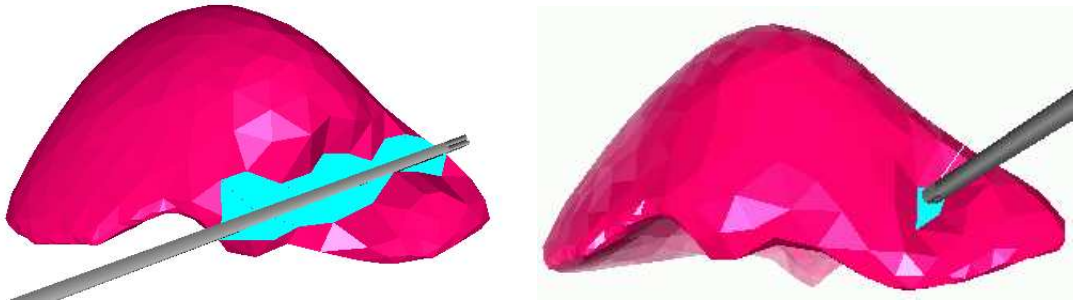


Figure 19: *Problem when colliding with the end of the tool*

problem is that when the tool is perpendicular to the surface, the projection plane is ill-defined. For these reasons, we have developed a specific algorithm for the end of the tool: we now project the collided faces on the plane defined by the end of the tool and the average normal of the collided faces (figure 20(b)).



(a) (b)

Figure 20: *Modeling of contact between the surgical tool and the liver soft tissue model.*

The problem was to split the set of collided faces in two sets, one for the end of the tool and the other for the handle. We took advantage of the efficiency of the collision detection algorithm by performing two collision detections at each simulation step. The first is restricted to the last centimeter of the tool and the second is done with the handle. In this way we can define very realistic boundary constraints for the soft tissue model deformable model (figure 21 and 22).

Furthermore, in order to slide on the surface, we do not apply any position constraint on projected vertices: these vertices are not fixed during the computation of the deformation.

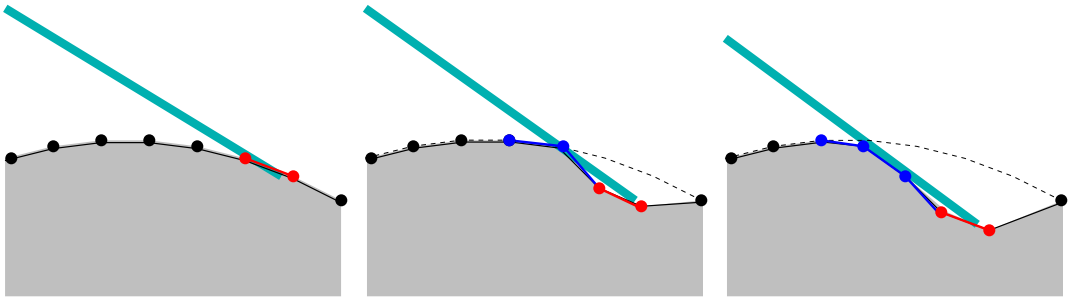


Figure 21: *Double collision detection (in red for the end of the tool and in blue for the handle) and corresponding projections*

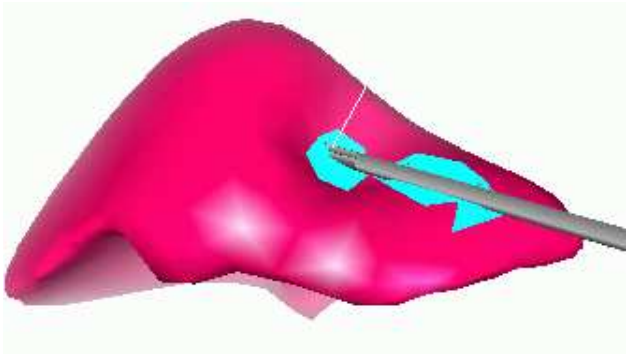


Figure 22: *Processing the contact with the whole tool and the liver soft tissue model*

In fact, the collided vertices are projected on the corresponding plane, and after each deformation step of the mesh, we re-project them on the same plane. This way, the vertices can move on the projection plane, allowing the triangles to slide under the tool, which means that we simulate a contact without any friction.

5.3 Interesting features

The interaction between a laparoscopic tool and soft tissues (the liver for example) is in reality very complex, and therefore very difficult to simulate realistically. But, for a surgery simulator, the quality of this interaction strongly influences the overall realism of the simulation. In that context, we have tried to model some of the most important gestures used in surgery :

- **Sliding on the surface :** The surface of most abdominal organs is wet and slippery. The surgeons often use this property to insert a tool between two organs, or to move a part of an organ in order to ease an other gesture. In some cases, this sliding is used by surgeons during a palpation process in order to identify some stiffer structures that can reveal the presence of lesions. In our model, the tool handle is always sliding on the liver. However, the tool end may have different actions depending on the nature of the surgical tool chosen.

- **Gripping a piece of the liver :** Even if the liver surface is fairly slippery, some part of the liver can be gripped by the jaws of pliers. In most cases, during surgery, a surgeon grips the edge of a liver lobe in order to reach a remote part of the liver or to ease the removal of a liver segment. We model this gesture as follows (see figure 23) : if the tool collides the liver with jaws of pliers wide open (see figure 23(a) and 23(b)), and when the jaws are half-closed (figure 23(c)), we consider that the corresponding triangles stick to the extremity of the tool (figure 23(d)).



(a) (b) (c) (d)

Figure 23: *The tool grips a part of the liver*

- **Removing tetrahedra with the jaws or with an electrical or ultrasound tool :** One important advantage of the tensor-mass model presented in section 3.4 is that it is well suited for the interactive simulation of cutting. In liver surgery, cutting is performed by removing pieces of parenchyma which we model by removing tetrahedra. We have implemented two cutting gestures, corresponding to the most common surgical methods. The former consists in "chewing" the liver with the jaws of the tool. We use the same approach as for representing the gripping gesture, but when the jaws are completely closed, we remove all tetrahedra located inside the jaws (figure 24).

  The latter gesture is related to another surgical tool often used for removing parenchyma : the cavitron (ultrasound lancet). The tool extremity creates ultrasound waves which explode the haptic cells at the vicinity of the tool. The main advantage of this method, when the ultrasound frequency is suitably chosen, is that it only removes the parenchyma without damaging the blood vessels. We model this gesture simply by removing all tetrahedra in contact with tool end. An example of this method is shown in part 7 (figure 27).

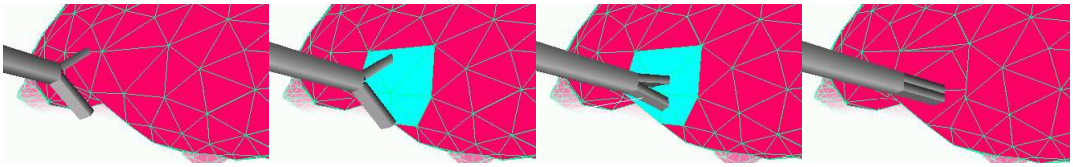


Figure 24: *The tool jaws cut a part of the liver*

From these contact model between soft tissues and laparoscopic tools, we compute the organ deformation and the reaction forces applied to the tools. These reaction forces are sent to the corresponding force-feedback devices in order to provide a haptic feedback to the user.

6 Force feedback

Laparoscopic surgery is much more difficult for surgeons than classical surgery. Surgeons loose the third dimension and use two long tools instead of their ten fingers. So they must use all the clues to understand and reconstruct the scene in their mind. Force feedback is one of these clues, and its introduction in our surgery simulator highly improves the overall realism of the simulation.

But haptic rendering requires a high update rate (ranging from 300Hz for soft objects to 10kHz for rigid contact) to give the user a proper sense of touch.
To achieve a satisfactory haptic feedback, two approaches have been proposed:

- computing forces empirically [3], for example by using a force proportional to the penetration depth of the tool in the object. In this case the force is not computed from a physical deformation, but only from geometric constraints.
- using a simplified physical model. The simplification can be performed in two ways, either by decreasing the mesh size [8] or by performing as much pre-computation as possible [13].

We propose another solution based on human characteristics. Indeed, it has been shown [7] that if haptic rendering is very precise (we can feel force discontinuities above 300 Hz), human gesture can be sampled at a much lower frequency (from 1Hz for the answer to an unexpected signal to 10Hz for a reflex action). Thus, all applied forces must be updated at a high rate, but, because it is related to the user's hand gesture, their evolution is quite slow. The idea is to estimate forces between two time steps of the deformable model simulation with an extrapolation. A more detailed presentation of this study can be found in [27].

6.1 Force extrapolation

Our aim is to generate forces at a rate of 500Hz from forces computed by the deformable object simulation at a rate of about 30Hz. The simulation loop gives us a discrete time series of parameters $(t_n, \mathbf{P}_n, \mathbf{F}_n)$ representing the force $\mathbf{F}_n$ applied to the tool in position $\mathbf{P}_n$ at time t_n. Good quality force feedback can be reached by a force update at about 500Hz. So, we must choose an extrapolation function $\mathbf{F}(t)$ providing an estimate force applied at time t $(t_n \leq t < t_{n+1})$ according to already known data $(t_i, \mathbf{P}_i, \mathbf{F}_i)$, $i = 0..n$.

The hardware of the *Laparoscopic Impulse Engine* assumes a **constant extrapolation** between two received forces. This has several advantages. First, it does not require any additional computation. Second, as the applied force results from the deformation computation, such an extrapolation scheme ensures that only valid forces are applied without the risk of damaging the device. The main problem with this method is the discontinuity of the applied force which gives the sensation of touching a rough surface as soon as the update rate becomes too low (under about 300Hz).

Another way to estimate the current value of a signal changing over time is to extrapolate it over time. As our deformable model sketches a linear elastic behavior, we only consider **linear extrapolation over time**. This method gives better results than the previously described one. The force discontinuities are less important. But we must face a new problem, since the applied forces do not originate from the simulation of deformable models, they can be arbitrarily large. These force amplitude peaks occur especially when the time step increases.

$$\mathbf{F}^p(t) = \mathbf{F}_n + \frac{t - t_n}{t_n - t_{n-1}}(\mathbf{F}_n - \mathbf{F}_{n-1}) \qquad t_n \leq t < t_{n+1} \tag{30}$$

The force changes are mainly due to the tool movement. In addition, it is possible to query at a very high rate the position of the tool during the extrapolation. These observations led us to develop a force estimator based on the tool position: **linear extrapolation over position**. We project the current tool position $\mathbf{P}$ onto the line defined by the two previous tool position $\mathbf{P}_{n-1}$ and $\mathbf{P}_n$ to obtain $\mathbf{P}$' (figure 25). We can then consider the norm ratio for extrapolation:

$$\mathbf{F}^p(t) = \mathbf{F}_n + \frac{\|\mathbf{P}' - \mathbf{P}_n\|}{\|\mathbf{P}_n - \mathbf{P}_{n-1}\|}(\mathbf{F}_n - \mathbf{F}_{n-1}) \qquad t_n \leq t < t_{n+1} \tag{31}$$

We notice that the error induced by the tool position projection is null when $\mathbf{P}_{n-1}$, $\mathbf{P}_n$, and $\mathbf{P}$ are aligned, in other words when the tool trajectory is a line.

All of these three extrapolation methods were implemented in our surgery simulator. In order to compare and to evaluate them, several experiments were performed. The results are presented in the next section.

6.2 Results

To evaluate and compare the three extrapolation methods, time of acquisition , tool position, and computed reaction forces were recorded during several surgery simulation sessions. We

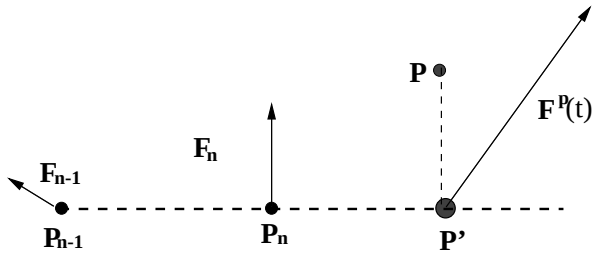


Figure 25: *Tool position projection for extrapolation over position*

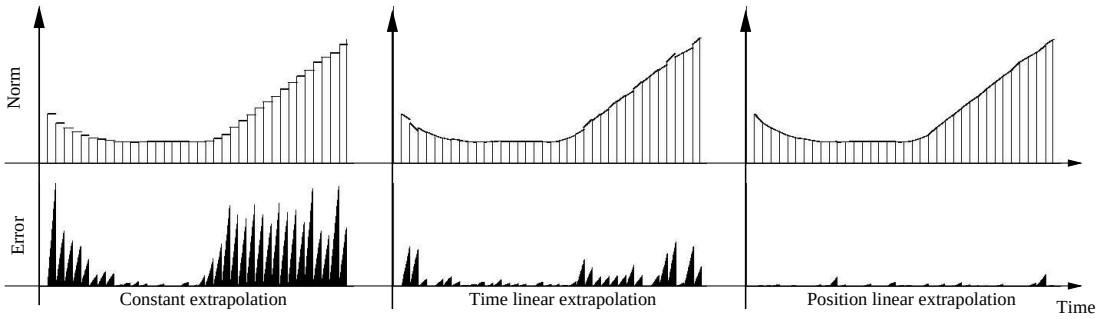


Figure 26: *Evaluating the different extrapolation methods*

have interpolated the force *a posteriori* to have a reference for the computation of the errors. We present results on the norm of the force. The top row of figure 26 shows the original data set with the impulses and the extrapolated one as a line. The norm of the difference between the extrapolated and the interpolated forces, which is taken as a measure of the error, is plotted on the bottom row.

We note that the linear extrapolation over position gives very interesting results (very few discontinuities and no singular forces). We tried the same type of experiments with different simulation frequencies and with different tool movements. The position linear extrapolation always gave the best results, which is confirmed by the sensation perceived during simulation.

7 Results: simulation of liver surgery

In this section, we present an example of the typical gesture we want to simulate: figure 27 shows a simulation of laparoscopic liver surgery during which the user pulls the right part of the deformable model with one tool while cutting with another tool. The interaction algorithm we have developed allows us to simulate several procedures, while the force ex-

trapolation method provides a realistic haptic feedback despite a low simulation frequency (about 30Hz). Our linear elastic deformable model reaches this frequency with a mesh size of up to 8000 tetrahedra, which allowed us to model reasonably complex organs.

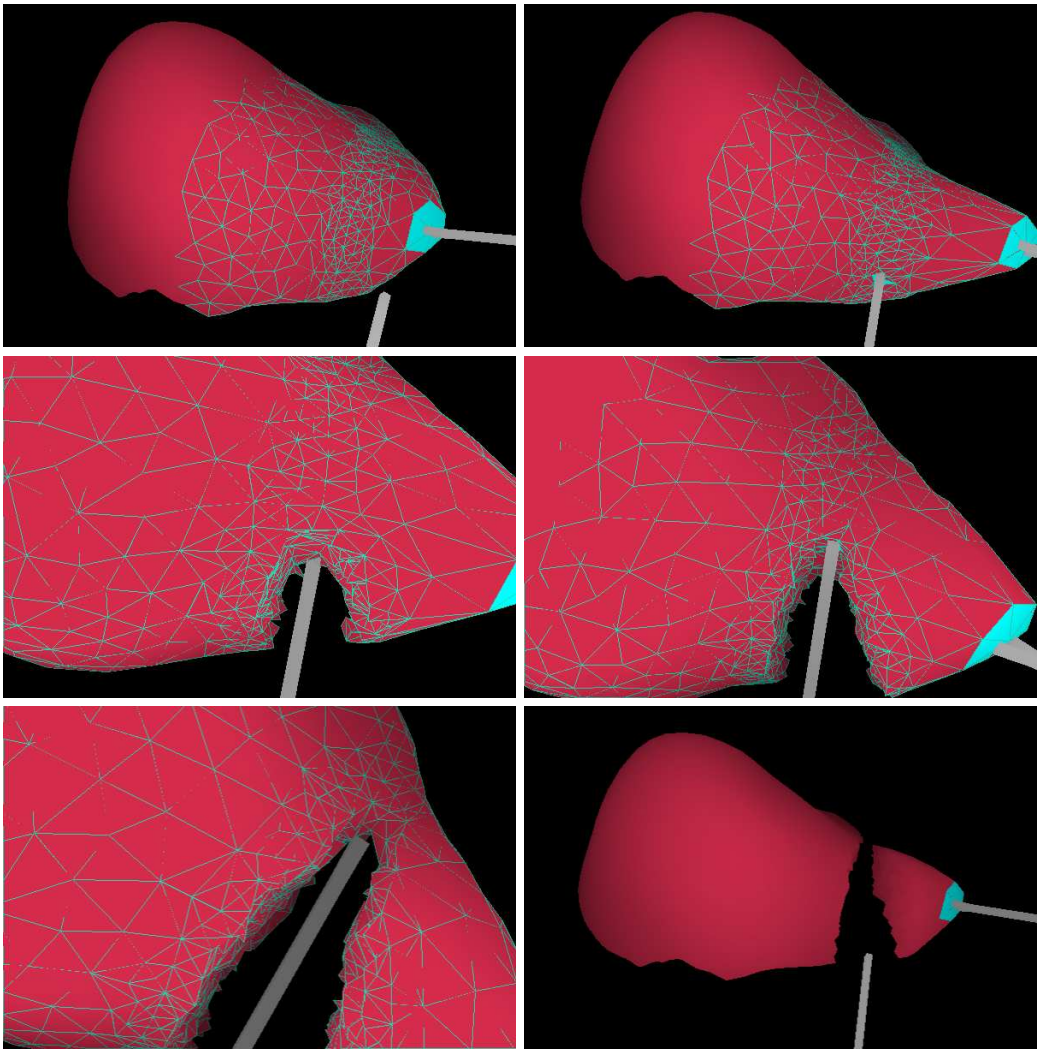


Figure 27: *Simulation of hepatectomy*

8 Conclusion

We have described, in this paper, several improvements to our minimally invasive surgery simulator. First, we have enhanced the bio-mechanical behavior of our deformable model by adding the possibility to simulate the deformations of anisotropic material. Indeed, we can now model materials having a different behavior in one given direction, thanks to the generalization of linear elasticity to transversally isotropic materials. We have also modeled the skin surrounding some organ by adding an external spring-mass mesh to our 3D tensor-mass model. Second, we have defined complex interactions between the organ and a laparoscopic tool, which can slide on the surface of the organ, grip, cut, or burn the organ. Finally we have proposed a method of extrapolation of the forces applied on the laparoscopic tools providing a realistic haptic feedback with only visual frame rate.

Our future work will focus on modeling more complex bio-mechanical behavior, with the introduction of non-linear elastic models. Another improvement will be the modeling of the hepatic vessels networks which constitute one of the main problems during liver surgery.

References

[1] Action Incitative SIMulation de chirurgie. http://www.inria.fr/epidaure/AISIM.

[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] C. Basdogan, C. Ho, M. A. Srinivasan, S. D. Small, and S. L. Dawson. Force Interaction in Laparoscopic Simulation: Haptic Rendering Soft Tissues. In *Medecine Meets Virtual Reality (MMVR'6)*, pages 28–31, San Diego CA, january 1998.

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

[5] M. Bro-Nielsen, D. Helfrick, B. Glass, X. Zeng, and H. Connacher. VR Simulation of Abdominal Trauma Surgery. In *Medicine Meets Virtual Reality (MMVR'98)*, pages 117–123, January 1998.

[6] 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*, pages 57–66. Blackwell Publishers, 1996.

[7] T.L. Brooks. Telerobotic response requirements. In *IEEE International Conference on systems, Man and Cybernetics*, pages 113–120, 1990.

[8] G. Burdea, G. Patounakis, V. Popescu, and R.E. Weiss. Virtual Reality Training for the Diagnosis of Prostate Cancer. In *IEEE International Symposium on Virtual Reality and Applications (VRAIS'98)*, pages 190–197, Atlanta, Georgia, March 1998.

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

[10] F. J. Carter. Biomechanical testing of intra-abdominal soft tissue. In *International Workshop on Soft Tissue Deformation and Tissue Palpation*, Cambridge, MA, October 1998.

[11] P. G. Ciarlet. *Mathematical elasticity Vol. 1: Three-dimensional elasticity.* Elsevier Science Publishers B.V., 1988.

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

[13] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions On Visualization and Computer Graphics*, 5(1):62–73, January-March 1999.

[14] D. Dan. *Caractérisation mécanique du foie humain en situation de choc.* PhD thesis, Université Paris 7, September 1999.

[15] F. Boux de Casson and C. Laugier. Modeling the Dynamics of a Human Liver for a Minimally Invasive Surgery Simulator. In *MICCAI'99*, Cambridge UK, 1999.

[16] G. Debunne, M. Desbrun, A. Barr, and M.-P. Cani. Interactive multiresolution animation of deformable models. In *10th Eurographics Workshop on Computer Animation and Simulation (CAS'99)*, September 1999.

[17] H. Delingette, S.Cotin, and N.Ayache. A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation. In *Computer Animation*, Geneva, Switzerland, May 26-28 1999.

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

[19] S. Gibson, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, A. Mor, S. Nakajima, H. Ohkami, R. Osborne, J. Samosky, and A. Sawada. Volumetric object modeling for surgical simulation. *Medical Image Analysis*, 2(2):121–132, 1998.

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

[21] J.-C. Lombardo, M.-P. Cani, and F. Neyret. Real-time collision detection for virtual surgery. In *Computer Animation*, Geneva Switzerland, May 26-28 1999.

[22] J. Marescaux, J-M. Clément, V. Tassetti, C. Koehl, S. Cotin, Y. Russier, D. Mutter, H. Delingette, and N. Ayache. Virtual Reality Applied to Hepatic Surgery Simulation : The Next Revolution. *Annals of Surgery*, 228(5):627–634, 1998.

[23] W. Maurel, Y. Wu, N. M. Thalmann, and D. Thalmann. *Biomechanical Models for Soft Tissue Simulation.* Springer-Verlag, 1998.

[24] J. Montagnat and H. Delingette. Globally constrained deformable models for 3d object reconstruction. *Signal Processing*, 71(2):173–186, December 1998. ftp://ftp-sop.inria.fr/epidaure/Publications/Montagnat/SPAIJ.ps.gz.

[25] P.F. Neumann, L.L. Sadler, and J. Gesser M.D. Virtual Reality Vitrectomy Simulator. In William M. Wells et al., editors, *First International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI'98*, number 1496 in Lecture Notes in Computer Science, pages 910–917, Cambridge MA, USA, October 11-13 1998. Springer Verlag.

[26] F. Neyret and M.-P. Cani. Pattern-based texturing revisited. In *SIGGRAPH'99*, Los Angeles CA, August 1999.

[27] G. Picinbono and J.-C. Lombardo. Extrapolation: a Solution for Force Feedback? In *International Scientific Workshop on Virtual Reality and Prototyping*, pages 117–125, Laval, France, June 1999.

[28] G. Picinbono, J.C. Lombardo, H. Delingette, and N. Ayache. Non-Linear Anisotropic Elasticity for Real-Time Surgery Simulation. Technical report, INRIA, October 2000.

[29] A. Rhomberg, C. Brechbühler, G. Székely, and G. Tröster. A parallel architecture for interactive fem computations in a surgery simulator. In *Proceedings of the Parallel Computing conference, ParCo99*, Delft, The Netherlands, August 1999. the non-profit foundation ParCo Conferences. accepted for publication.

[30] L. Soler, H. Delingette, G. Malandain, J. Montagnat, N. Ayache, J.-M. Clement, C. Koehl, O. Dourthe, D. Mutter, and J. Marescaux. A fully automatic anatomical, pathological and functionnal segmentation from CT-scans for Hepatic Surgery. In *Medical Imaging 2000*, San Diego, February 2000.

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

[32] A. J. Spencer. *Continuum Theory of Fiber-Reinforced Composites.* Springer-Verlag, New York, 1984.

[33] N. Suzuki, A. Hattori, A. Takatsu, T. Kumano, A. Ikemoto, Y. Adachi, and A. Uchiyama. Virtual Surgery System Using Deformable Organ Models and Force Feedback System with Three Fingers. In William M. Wells et al., editors, *First International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI'98*, pages 397–403, Cambridge MA, USA, October 11-13 1998. Springer Verlag.

[34] G. Székely, Ch. Brechbühler, R. Hutter, A. Rhomberg, and P. Schmid. Modelling of Soft Tissue Deformation for Laparoscopic Surgery Simulation. In William M. Wells et al., editors, *First International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI'98*, number 1496 in Lecture Notes in Computer Science, pages 550–561, Cambridge MA, USA, October 11-13 1998. Springer Verlag.

[35] L.V. Tsap, D.B.Goldgof, S. Sarkar, and W.-C. Huang. Efficient Nonlinear Finite Element Modeling of Nonrigid Objects via Optimization of Mesh Models. *Computer Vision and Image Understanding*, 69(3):330–350, March 1998.

[36] M. Vidrascu. Simulation of the deformations of the liver by domain decomposition. In *mini-symposim Decomposition de domaines de USNCCM99*, Boulder, August 1999.

[37] V. Viskovic, M. Kauer, G. Székely, and M. Reidy. Realistic Force Feedback for Virtual Reality Based Diagnostic Surgery Simulators. In *IEEE International Conference on Robotics and Automotion: ICRA 2000*, pages 1592–1598, San Francisco, CA, April 2000.

[38] Jeffrey A. Weiss, Bradley N. Maker, and Sanjay Govindjee. Finite element implementation of incompressible, transversely isotropic hyper-elasticity. *Computer Methods in Applied Mechanics and Engineering*, 135:107–128, 1996. U.C. Berkely Report UCB/SEMM-95/07.



Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399