INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Real-time elastic deformations of soft tissues for surgery simulation

Stéphane Cotin , Hervé Delingette, Nicholas Ayache

N° 3511

Octobre 1998

THÈME 3

Rapport de recherche

ISSN 0249-6399

Real-time elastic deformations of soft tissues for surgery simulation

Stéphane Cotin *, Hervé Delingette**, Nicholas Ayache***

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

Rapport de recherche n° 3511 — Octobre 1998 — 28 pages

Abstract: In this paper, we describe a new method for surgery simulation including a volumetric model built from medical images and an elastic modeling of the deformations. The physical model is based on elasticity theory which suitably links the shape of deformable bodies and the forces associated with the deformation. A real-time computation of the deformation is possible thanks to a pre-processing of elementary deformations derived from a finite element method. This method has been implemented in a system including a force feedback device and a collision detection algorithm. The simulator works in real-time with a high resolution liver model.

Key-words: Surgery simulation, deformable models, real-time, force feedback

(Résumé : tsvp)

* E-mail: Stephane.Cotin@sophia.inria.fr
** E-mail: Herve.Delingette@sophia.inria.fr
*** E-mail: Nicholas.Ayache@sophia.inria.fr
**** http://www.inria.fr/Equipes/EPIDAURE-eng.html

Unité de recherche INRIA Sophia Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex (France)
Téléphone : 04 93 65 77 77 – Télécopie : 04 93 65 77 65

Déformations élastiques temps-réel de tissus mous pour la simulation de chirurgie

Résumé : Dans ce rapport de recherche, nous décrivons une nouvelle méthode permettant de simuler la chirurgie de structures anatomiques tridimensionnelles modélisées à partir d'images médicales. Le modèle physique de ces structures anatomiques est fondé sur la théorie de l'élasticité linéaire qui établit le lien entre les forces appliquées sur ces corps mous et la déformation résultante. Un calcul en temps-réel de la déformation est possible grâce à un pré-calcul des déformations élémentaires obtenues par la méthode des éléments finis. Cette méthode à été mise en oeuvre dans un simulateur chirurgical incluant un système à retour d'effort ainsi qu'un algorithme de détection de collisions. Nous présentons des exemples de simulation de la chirurgie hépatique.

Mots-clé : simulation de chirurgie, modèles déformables, éléments finis, imagerie médicale, retour d'effort, élasticité linéaire

1 Introduction

With the development of laparoscopic techniques which reduce operating time and morbidity, surgical simulation appears to be an essential element of tomorrow's surgery. Indeed, most unexperienced surgeons and students have to practice this novel technique. Nowadays they practice on an endotrainer, on living animals or cadavers. The lack of realism in the first solution and the ethical problems linked to the other solutions show a real need for simulated surgery.

This article presents new ideas for developing a real-time surgery simulation system incorporating the following characteristics: a volumetric deformable model, a force feedback device, and real-time deformations based on a quasi non-linear biomechanical behavior.

1.1 Simulation in surgery

Visual-realism and real-time interactions are essential in surgery simulation. Real-time interaction requires that any action from the operator generates an instantaneous response from the stimulated organ, whatever the complexity of its geometry. Moreover, since all the organs in the human body are not rigid, their shape may change during an operation. Consequently, the realism of the deformations is another key point in surgery simulation. This realism can be enhanced by the introduction of devices which allow for a better immersion in the virtual world. In surgery simulation, the integration of force feedback systems to generate such sensations is of prime importance, almost as important as visual feedback. When coupled with precise computations of the forces, it may be possible for the surgeon to feel haptic sensations close to reality.

1.2 Related works

Terzopoulos *et al.* [TPBF87], Waters [Wat92], and Platt and Barr [PB88] have shown the advantages of physical models over kinematic models for computer animation. Among these physical models, elastic[1] models have been extensively described in the literature [Spe92, CZ92, SR95, BCC95]. The particular shape of an elastic body is a function of both the internal stress and strain within the object and the external forces applied to it. Generally, some modifications or simplifications are made to the elasticity theory in order to give a particular behavior to the deformable body.

Among the representations used for deformable surfaces/volumes, the most widely used are *parametric models* with B-spline representation and *finite element models* introduced for computer engineering [Zie77] and applied to computer animation by Terzopoulos, Gourret, etc. For example, Gourret *et al.* [GMTT89] have described a system for modeling the human hand with a finite element volume meshed around bones. They formulate and solve a set of static equations for skin deformation based on bone kinematics and hand/object contact points in a grasping task. Other possible models are *mass-spring models* with the works of Miller [Mil88], Chadwick *et al.* [CHP89], Norton *et al.* [NTBG91], Luciani *et al.* [LJFC91], Joukhadar [Jou95] and *implicit surfaces* with, for example, the works of Desbrun and Gascuel [DG95].

In surgery simulation, scientists have mainly focus on the mass-spring methods due to their simplicity of implementation and their relatively low computational complexity [BG96, MC97, KN93]. For example, Kuehnapfel and Neisius [KN93] present a simulation of endoscopic

[1] This term includes elastic, linear elastic and visco-elastic models.

surgery based on a surface mass-spring model. Although in this case the interactions are driven by instruments with sensors, no force feedback is used. The simulation environment of Gibson *et al.* [GSMF97] takes into account the volumetric nature of the organs with a deformation law derived from a mass-spring model. Cover *et al.* [CEO93] have also developed a model based on thin plate splines for laparoscopic gall bladder surgery simulation. We can also cite the work of Koch *et al.* [KGCvB96] and Keeve *et al.* [KE96] on facial tissue modeling. Finite element models are less widely used due to the difficulty of their implementation and their larger computing time. Nevertheless, Sagar *et al.* [SBMH94] have proposed a method for simulating features of the human eye with a complex behavior (large incompressible 3D elastic deformations). However, no details are given about the implementation of the finite element method used to solve the equations. Another example of eye surgery was given by Le Tallec *et al.* [Le 93]. In this work, the authors have achieved a very precise study of the deformation law of the eye when touched by a trepan. However, the computing times should be accelerated by a factor of 1,000,000 to allow real-time simulation with force feedback.

Computing time reduction has been studied by Bro-Nielsen and Cotin [BNC96] using a condensation technique [Zie77]. With this method, the computation time required for the deformation of a volumetric model can be reduced to the computation time of a model only involving the surface nodes of the mesh. Song and Reddy [SR95] have described a technique for cutting linear elastic objects defined as finite element models. This technique was only applied to very simple two dimensional objects. We can also cite a method for free-form cutting in tomographic volume data [PTH95] based on voxel operations for cutting and visualization.

In our approach, we have tried to integrate all the requirements for a realistic simulation (i.e. real-time modeling of elastic tissue, real-time visual and haptic feedback). The static equations of the elastic model are solved by a modified finite element method that takes into account particular boundary conditions. The solution of these equations gives not only the deformed mesh but also the forces to be sent to a force feedback device according to the actual deformation. Finally, real-time interaction is possible thanks to a pre-processing of elementary deformations coupled with a speed up algorithm. The linear elastic deformations, computed in real-time, give a first approximation of reality and this has allowed the development of a first version of a simulator [CDCT96]. This linear model has been enhanced by taking into account biomechanical results on soft tissues and now gives a more realistic behavior.

The following sections detail the main concepts of our approach. Section 2 points out the problem of segmentation: how to build a model of the liver from 3D medical images. Section 3 describes two approximations (linear and quasi non-linear models) of the deformation law of soft tissues and in section 4, we present a method allowing real-time computation of the deformation and the forces. The collision detection algorithm used in the simulator is briefly described in section 5. Section 6 presents experimental results and section 7 concludes with perspectives.

2 3D reconstruction

The choice of a specific geometric representation is of prime importance for simulators. This choice should be governed by the resulting trade-off between realism and interaction. In our method, the realistic aspect of the organs is linked, firstly, to an accurate segmentation of a volumetric medical image and, secondly, to an appropriate representation of the segmented organ's surface. In the following, we will focus on the *liver*. The liver has a very important function in the metabolism and its surgery is complex. Moreover, the development of new laparoscopic techniques for hepatic surgery are of growing interest. To this end, we are working in collaboration with the medical team of Pr. Marescaux at IRCAD[2], specialized in hepatic surgery.

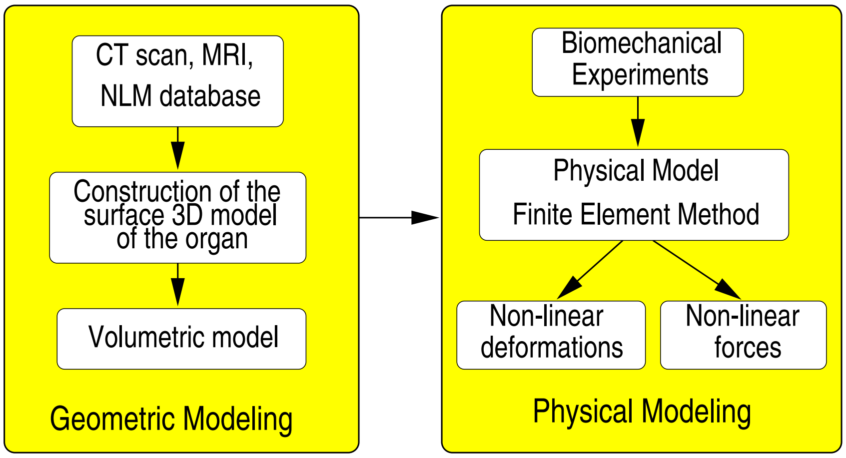

Figure 1: Creation of the deformable model. A geometric model is extracted from a 3D medical image. Then a deformation law is added.

2.1 Creation of an anatomical model of the liver

In order to produce a model of the liver with anatomical details, a dataset provided by the *National Library of Medicine* was used. This dataset consists of axial MRI images of the head and neck and longitudinal sections of the rest of the body. The CT data consists of axial scans of the entire body taken at 1 mm intervals. The axial anatomical images are scanned pictures of cryogenic slices of the body. They are 24 bit color images of size 2048 by 1216 pixels. These anatomical slices are also at 1 mm interval and coincide with the CT axial images. There are 1878 cross-sections for each modality.

To extract the shape of the liver from this dataset, we used the anatomical slices (cf. figure 2), which gives a better contrast between the liver and the surrounding organs. The dataset concerning the liver can be reduced to about 180 slices. After contrast enhancement, we apply an edge detection algorithm to extract the contours of the image, and then using a simple thresholding technique, we retain the stronger ones. Next, we use semi-automatic deformable models for two-dimensional contour extraction to generate a set of two-dimensional binary images (cf. figure 2). The slices generated are then stacked to form a three-dimensional binary image [MD97] (cf. figure 3).

2.2 Simplex meshes

In order to capture the shape of the external surface of the liver we could use a sub-voxel triangulation provided by the marching cubes algorithm [LC87], however the number of triangles generated is too large for further processing. Moreover, a smoothing of the surface is necessary

[2]Institute for Research Against Cancers of the Digestive tract.

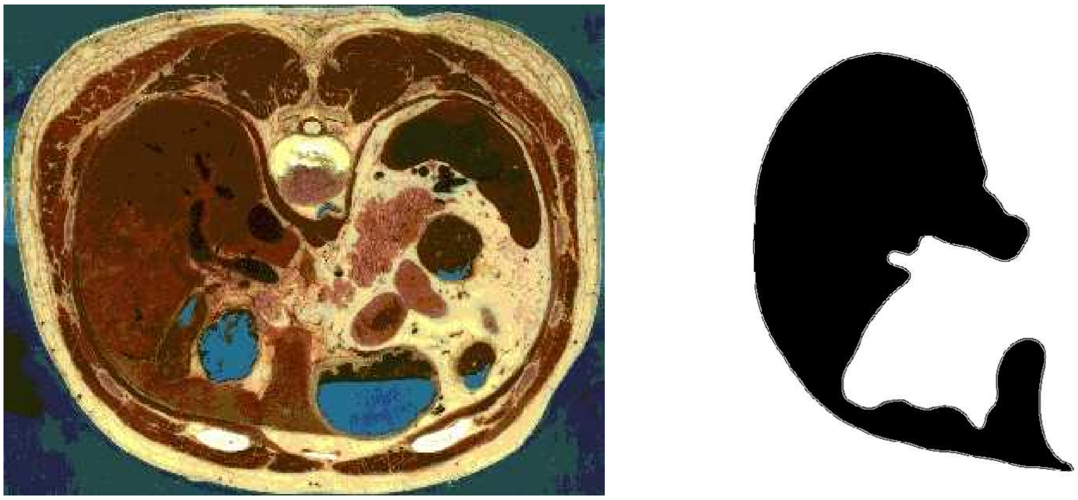

Figure 2: Segmentation of the liver, slice by slice. The initial data (left) is high resolution photography of an anatomical slice of the abdomen. The binary image (right) corresponds to the segmented liver cross-section.

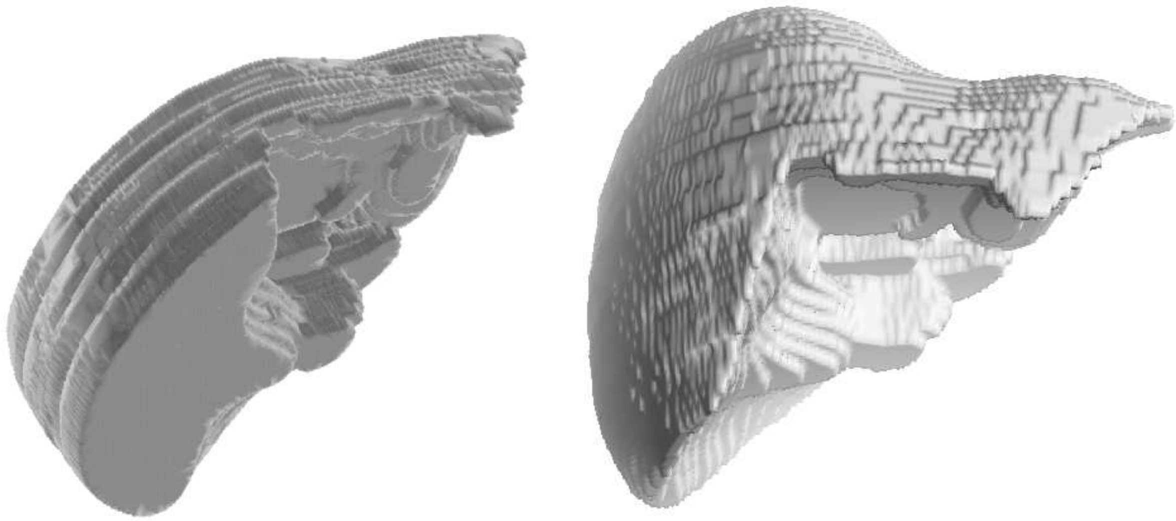

Figure 3: After segmentation, the binary images are stacked (left) to give a 3D binary image. We see the step-effect on the shape of the liver (right) when extracted using the marching-cubes algorithm.

to avoid staircase effects (cf. figure 3). A possible solution consists in decimating an iso-surface model by using a mesh simplification tool. However, for more flexibility, both in the segmentation and simplification processes, we have used "simplex meshes".

Simplex meshes are an original representation of three-dimensional objects developed by Delingette [Del94a, Del94b]. A simplex mesh can be seen as an extension of a snake [KWT88] in 3D and therefore is well suited for generating geometric models from volumetric data. A simplex mesh can be deformed under the action of regularizing and external forces. Additional properties like a constant connectivity between vertices and a duality with triangulations have been defined. Moreover, simplex meshes are adaptive, for example by concentrating vertices in areas of high curvature (thereby realizing an optimal shape description for a given number of vertices). The mesh may be refined or decimated depending on the distance of the vertices from the dataset. The decimation can also be interactively controlled. Figure 4 shows the effect of the mesh adaptation and we see the vertices are nicely concentrated at highly curved parts of the liver.

By integrating simplex meshes in the segmentation process, we have obtained smoothed triangulated surfaces, very close to an iso-surface extraction, but with fewer faces to represent the shape of the organs. In our example, the model of the liver has been created by fitting a simplex mesh to the tridimensional binary image previously described. Thanks to the adaptation and decimation properties of the simplex meshes, this model is composed of only 14,000 triangles, whereas the marching cubes algorithm gave 94,000 triangles (cf. figures 3 and 4).

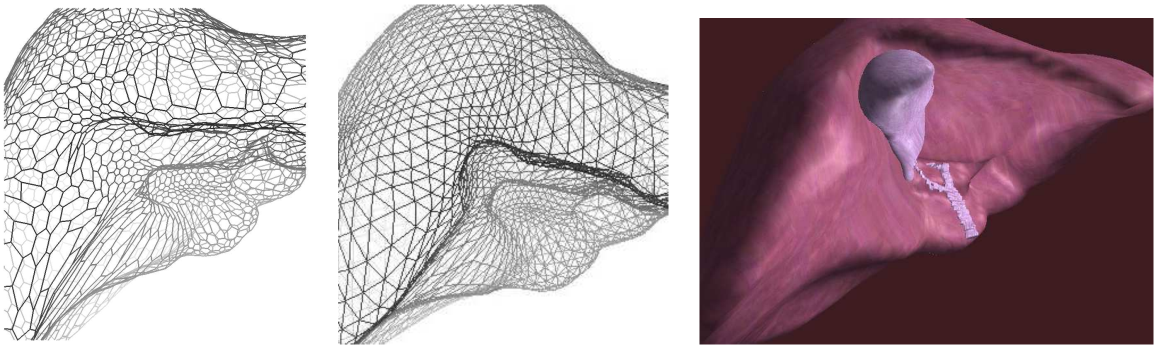

Figure 4: Different representations of the geometric liver model. The simplex mesh fitting the data (left) with a concentration of vertices in areas of high curvature, the triangulated dual surface (right) and a texture-mapped model with anatomical details (gall bladder and ducts) from an endoscopic viewpoint (bottom).

Although this approach is very useful for building a "generic" liver model, it is essential to integrate "patient-based" models in the simulator. In the framework of this research project, Montagnat and Delingette [MD97] are working on a method for extracting liver models from CT scan images. This method is based on simplex meshes with additional shape constraints.

2.3 Creating the volumetric model

Since 3D elasticity equations consider a solid body, a volumetric meshing of the object is required. The volume delimited by the previously extracted shape has to be divided into a set of finite elements of similar size. Due to the shape irregularity of anatomical models, a decomposition of the volume into tetrahedral elements is the best solution. Thanks to the duality property of simplex meshes, we can easily create a triangulation of the liver's surface. Then, the volume is decomposed into a set of tetrahedra such that the original triangulation is not modified. In order to achieve this, we used a product, $Simail^{TM}$[3], which is based on a Delaunay-Voronoi algorithm.

3 Physical modeling

Our model is based on elasticity theory, however our approach differs from related surgery simulation work on several points. First, the integration of a force feedback system in the simulation loop requires additional constraints on the model. Second, biomechanical studies about soft tissues have been taken into account in the deformation process although the underlying law remains linear. Third, the solution of the differential equations is optimized to handle fast deformations of complex models.

3.1 Force feedback

Recent work [BH95] has shown that in virtual environments the sense of presence is highly correlated with the degree of immersion in that environment. In particular, the sensation of forces (haptic feedback) and the sensation of textures are very important in medical applications. In minimally invasive surgery the surgeon's hands remain outside of the patient's body, thus tactile feedback is not needed. Since force sensing depends on the deformable nature of the virtual model, the realism of the force feedback is highly correlated to the model's physical realism.

The information flow in the simulator should form a closed loop [JR95]: the model deforms according to the motion induced by the surgeon with the force feedback device. This deformation allows us to compute the contact force and finally the loop is closed by generating this force through mechanical actuators.

When adding force feedback in a simulation, the main difficulty is related to the real-time constraint imposed by such a system: to remain realistic, the forces must be computed at a very high frequency, at least equal to 300 Hz [Bur96]. This frequency can vary according to the tissue stiffness and the kind of interaction. Deformations and graphics rendering need to be performed at about 24 Hz, that corresponds to the frequency of the human persistence of vision. Finite element methods or mass-spring models are computationally prohibitive. To guarantee interactive rates with large meshes, we use a speed-up algorithm presented in section 4.

3.2 Biomechanical behavior of soft tissues

It is hard to quantify the realism of a deformable model since very little information is available regarding the deformability of human tissue. Recent publications in the field of surgical simulators make use of thin-plate deformable surfaces, mass-spring models, or volumetric linear elasticity. With such models, the real-time constraint is difficult to achieve without detriment to the realism. Conversely, research in the field of biomechanics has shown that the fairly realistic model for soft tissues is probably a visco-elastic non-linear model [Fun93]. The major disadvantage of such a model remains its high computational complexity.

[3] Simulog S.A. - 1, rue James Joule - 78286 Guyancourt Cedex - France

3.3 3D linear elasticity as a first approximation of the deformation law for soft tissue

In a previous work [CDA96], we have shown the interest of linear elasticity as a reasonable approximation for soft tissue deformation. Here we recall the statement of the elasticity problem. Let Ω be the configuration of an elastic body before deformation. Under the action of a field of volumetric forces f_Ω and surface forces f_Γ, the elastic body is deformed and takes a new configuration Ω^*. The problem is then to determine the displacement field u which associates with the position $\mathbf{p}_0$ of any particle of the body before deformation, its position $\mathbf{p}$ in the final configuration. In order to solve this problem numerically, we use a classical finite elements approach, i.e., with Lagrange elements of type P_1 [Bat96]. Using this class of elements implies the decomposition of the domain Ω into a set of tetrahedral elements as mentioned in section 2.3. Through variational principles, the elasticity theory problem's solution becomes equivalent to the solution of a linear system:

$$[\boldsymbol{K}]\,\boldsymbol{u} = \boldsymbol{f} \tag{1}$$

where $[\boldsymbol{K}]$ is the stiffness matrix and is symmetric, positive definite, and sparse; $\boldsymbol{u}$ is the unknown displacement field and $\boldsymbol{f}$ the external forces. The size of this matrix $[\boldsymbol{K}]$ is $3\,n \times 3\,n$ where n is the number of mesh vertices (or nodes). We immediately see that the mesh's size (represented by the number of mesh nodes) is an important parameter influencing the computation time. The use of a good geometric representation and the possibility of simplifying it by decimation (cf. section 2.2) are consequently of prime importance, even if this condition is not sufficient to insure real-time interactivity (cf. section 4).

In a general approach, a set of external forces are applied to the surface of the solid while some mesh nodes are fixed (otherwise a translation, and not a deformation, would occur). When using a force feedback device for the interactions with the deformable model it is impossible to measure the forces exerted by the operator. The endoscope's extremity position is the only information transmitted by the device. Consequently, the deformation must be driven by specific constraints *in displacement* and *not in force*. Hence, our boundary conditions are mainly the contact points displacements between the surgical tool and the body. We can then deduce both the forces exerted on the end effector of the tool and the global deformation by applying the linear elasticity equations (cf. figure 5). The computed forces are finally set into the force feedback system in order to provide a mechanical resistance to the surgeon's hands: the loop is closed.

In order to include the new constraints, we replace the linear system $[\boldsymbol{K}]\,\boldsymbol{u} = \boldsymbol{f}$ by the new system (cf. Appendix A):

$$\left[\begin{array}{c|c} \boldsymbol{K} & \overline{\boldsymbol{K}} \\ \hline \overline{\boldsymbol{K}}^{\boldsymbol{T}} & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u} \\ \hline \boldsymbol{\lambda} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f} \\ \hline \boldsymbol{u}^{\star} \end{array}\right] \tag{2}$$

where $[\overline{\boldsymbol{K}}]$ is a matrix composed of vectors $\mathbf{e}_i$ with a 1 in the ith position and zeroes elsewhere. This system results from the *Lagrange multipliers* method which is used for the imposition of specified values for the solution variables. The values λ_i of $[\boldsymbol{\lambda}]$ obtained after solving the system (2) are equal to the opposite of the force that needs to be applied at the degree of freedom[4] u_i in order to impose the displacement $u_i = u_i{}^{\star}$.

[4] In our case, we have only three degrees of freedom, corresponding to the components of the displacement vector.

In linear theory, the behavior of the deformable model is physically correct only for small displacements (about 10% of the mesh size), it is less realistic for larger deformations. Although it is a major disadvantage of linear elasticity, the integration of force feedback in the simulation limits the range of deformations to small deformations. This is because the force in the surgeon's hand will increase as he increases the deformation, thus preventing large deformations. Consequently, the deformation remains reasonably small. Another interest of linear elasticity is the possibility to compute any mesh deformation from the knowledge of a finite set of elementary deformations. This point will be detailed in section 4.

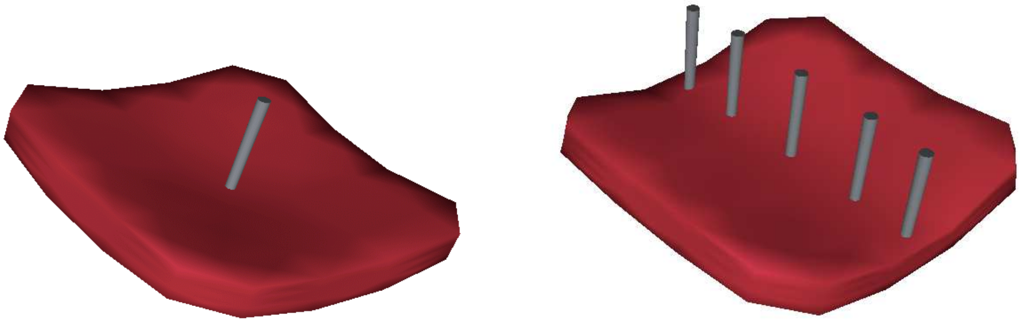

Figure 5: Deformation of a plate. The nodes at corners are rigidly fixed. The plate is deformed under a simple contact (left). Several constraints are combined to give the final deformation (right) by using the superposition principle.

3.4 Improvement of the physical model to simulate quasi non-linear elasticity

Recent publications in biomechanics tend to show that a reasonable model for soft tissue is non-linear visco-elasticity [Fun93]. Recent experiments realized by Chinsei [CM97] on cylindrical samples of the brain (very soft tissue with a similar behavior to the liver but with different parameters) have established the following properties: the response under compression depends on the loading speed (there is a strong stress/strain rate dependence) and is non-linear (cf. figure 6). The repeatability tests have shown that the response remains approximately the same under identical conditions.

From these experimental results, it is possible to plot the curves $f(x)$ and $d_r(x)$, i.e., the *loading force* and the *radial displacement* as functions of the *axial displacement* (cf. figures 7 and 8). These diagrams are only related to the loading phase and not the stress relaxation phase.

Assuming that such curves are available for the organ we consider (it is not yet the case for the liver), a least squares minimization is used to find the polynomial function $P(x)$ approximating $d_r(x)$ and the polynomial function $Q(x)$ approximating $f(x)$. Depending on the precision required for the approximation and the shape of the experimental curve, the degree of the polynomial functions ranges from 4 to 10.

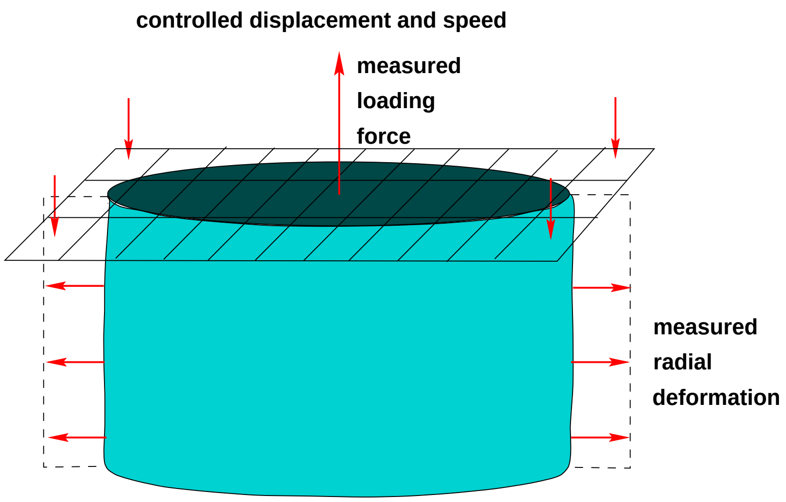

Figure 6: A sample of soft tissue is compressed at a fixed speed while the axial force, axial displacement, and radial deformation are measured.

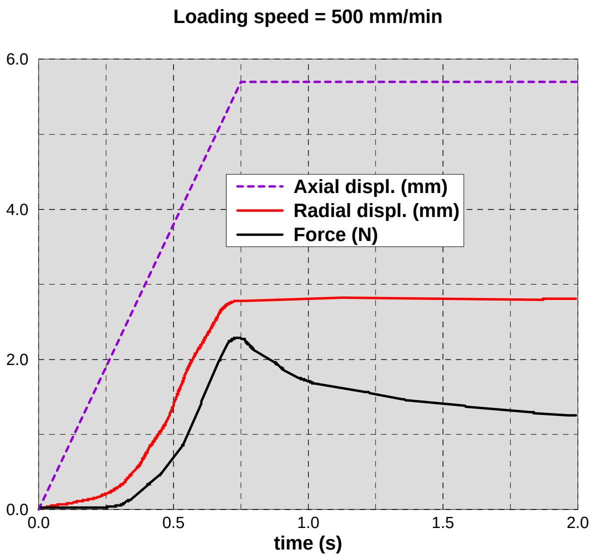

Figure 7: The curves obtained at the end of the experiment, for a compression speed of 500mm/min.

To extend these results to 3D while preserving a deformation law based on linear elasticity, we make the following assumptions. First, the computation of a quasi non-linear deformation[5] will be based on a linear deformation result using the same mesh with the same constraints. Consequently, $d_r(x)$ (resp. $f(x)$) has to be expressed as a function of the linear displacement $d_r^L(x)$ (resp. linear force $f^L(x)$). Second, the variation in *amplitude* of the radial displacement or radial force vectors is constrained to be similar to the experimental curves. Therefore, the displacement vector $\mathbf{u}_n$ associated to a node $\mathbf{n}$ is decomposed into two vectors $\mathbf{u}_n^t$ and $\mathbf{u}_n^r$, one collinear and one orthogonal to the direction of the constraint. Such a decomposition of the motion is required because experimental results provide only information on the radial deformation according to a axial constraint applied on the top of the cylindrical sample. Consequently, only the non-linear variation of the amplitude of the orthogonal component is known. Thus the amplitude of $\mathbf{u}_n^r$ is modified to follow the curve $P(x)$, while $\mathbf{u}_n^t$ continues to have linear variation.

In order to express the curve $d_r(x)$ (resp. $f(x)$) as a function of the corresponding linear deformation $d_r^L(x)$ (resp. $f^L(x)$), empirical elastic parameters (E, ν) are set for the tissue (cf. figure 8). For a cylindrical sample of height h and radius r, the theoretical radial deformation, is $d_r^L(x) = \nu \dfrac{rx}{h}$ while the force is $f(x) = \dfrac{sEx}{h}$ where s is the surface of the top of the cylinder. Hence, we can deduce the functions $P(d_r^L(x))$ (resp. $Q(f^L(x))$) for particular elastic properties (cf. figure 9).

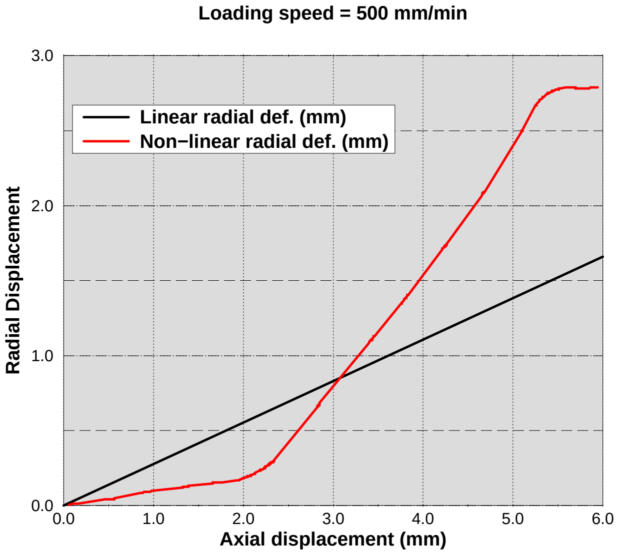

Figure 8: Linear and non-linear radial displacements as functions of the axial displacement.

Now, if we consider a particular mesh node $\mathbf{n}$ with linear displacement $\mathbf{u}_n = \mathbf{u}_n^t + \mathbf{u}_n^r$, the corresponding non-linear displacement u_n is:

$$u_n = \mathbf{u}_n^t + u_n^r \tag{3}$$

[5]The underlying deformation model is still linear elasticity, so the term non-linear deformation would be incorrect.

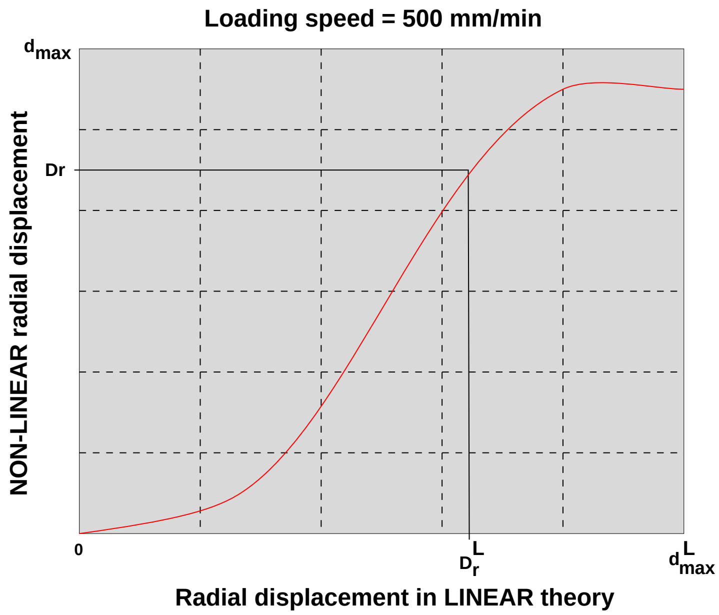

Figure 9: Non-linear displacement expressed in terms of the linear displacement.

with:

$$u_n^r = P(||\mathbf{u}_n^r||)\frac{\mathbf{u}_n^r}{||\mathbf{u}_n^r||} \tag{4}$$

The amplitude of the forces is modified in the same way except that it is the collinear component $\mathbf{f}_n^t$ that will be modified according to $Q(x)$. Thus, the non-linear force f_n occurring at a node $\mathbf{n}$ is:

$$f_n = f_n^t + \mathbf{f}_n^r \tag{5}$$

with:

$$f_n^t = Q(||\mathbf{f}_n^t||)\frac{\mathbf{f}_n^t}{||\mathbf{f}_n^t||} \tag{6}$$

4 Reduction of computing time

The degree of realism required in surgical simulation requires a complex model of the organ. The number of mesh vertices has a direct impact on the size of the matrices involved in the linear system $[\boldsymbol{K}]\boldsymbol{u} = \boldsymbol{f}$. The computation time required for solving the system is too high for real-time deformation of the mesh. In order to speed up the interactivity rate, we take advantage of the following properties: the linearity and the superposition principle.

4.1 Pre-computation of elementary deformations

The pre-processing algorithm can be described as follows.

- Specify a set of mesh nodes that remain fixed during the deformation. Specified values $u_i^* = 0$ of the solution $\boldsymbol{u}$ are not necessarily set for the three degrees of freedom (*dof*) of these nodes. Sometimes only one or two *dof* need to be set.

- For each *dof* k_i of each "free" node $\mathbf{k}$ on the surface of the mesh, an "elementary" displacement constraint u_i^* is set and the following quantities are computed:
  - the displacement of every free node $\mathbf{n}$ ($\mathbf{n} \neq \mathbf{k}$) in the mesh. It is stored as a set of 3×3 tensors $[\boldsymbol{T}_{\boldsymbol{nk}}^{\boldsymbol{u}}]$ expressing the relation between the displacement of node $\mathbf{n}$ in the mesh and the elementary displacement $\mathbf{u}_k^* = [u_{k_1}^*, u_{k_2}^*, u_{k_3}^*]^T = [\delta_1, \delta_2, \delta_3]^T$ imposed at node $\mathbf{k}$. The applied displacement is the same for every node.
  - the components of the elementary force λ at node $\mathbf{k}$. It is stored as a 3×3 tensor $[\boldsymbol{T}_{\boldsymbol{k}}^{\boldsymbol{f}}]$.

If m is the number of free nodes, the linear system

$$\left[\begin{array}{c|c} \boldsymbol{K} & \overline{\boldsymbol{K}} \\ \hline \overline{\boldsymbol{K}}^T & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u} \\ \hline \boldsymbol{\lambda} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f} \\ \hline \boldsymbol{u}_{\boldsymbol{k}}^* \end{array}\right]$$

has to be solved $3m$ times. We use an iterative method (conjugate gradient) to solve each linear system. The pre-processing stage can last anywhere from a few minutes to several hours depending on the size of the model and on the desired precision. For example, the pre-processing time required for a mesh with 193 vertices and 725 tetrahedra takes 7 min on a Dec AlphaStation 400 Mhz. When the mesh size increases to 1735 vertices and 8124 tetrahedra, the computation time reaches about 9 hours. Finally, these results are stored in a file. Since they depend on the fixed nodes and elasticity parameters, it is possible to generate a different set of files that correspond to a particular behavior, with specified boundary values for a given geometry.

4.2 Real-time deformations with linear elasticity

The displacement $\mathbf{u}_n$ of a node $\mathbf{n}$ in the mesh (node on the surface or in the volume) induced by the constraint $\mathbf{u}_k^*$ applied at node $\mathbf{k}$ can be obtained by the following linear equation.

$$\mathbf{u}_n = [\boldsymbol{T}_{\boldsymbol{nk}}^{\boldsymbol{u}}] \frac{\boldsymbol{u}_{\boldsymbol{k}}^*}{\|\boldsymbol{u}_{\boldsymbol{k}}^*\|} \tag{7}$$

for any node $\mathbf{k} \neq \mathbf{n}$. In general, more than one node is moved on the surface of the mesh during a contact with another object. The total displacement of a node is the sum of all the displacements induced by the controlled nodes $\mathbf{u}_{k_l}^*, l = 1, .., m$. However, the superposition principle is not directly applicable since the controlled nodes are also influenced by this principle. For example, considering two nodes with controlled displacements $\mathbf{u}_{k_1}^*$ and $\mathbf{u}_{k_2}^*$, we can see on figure 10 that the final displacement at these nodes will be greater than the imposed displacement. In other words, the mesh would deform more than it should.

Consequently, the constraint $\tilde{\mathbf{u}}_{k_l}^*, l = 1, .., m$ that must be applied instead of $\mathbf{u}_{k_l}^*, l = 1, .., m$ can be determined from the knowledge of the previous tensors of deformation $[\boldsymbol{T}_{\boldsymbol{nk}}^{\boldsymbol{u}}]$. The following matrix represents the mutual influence of all the controlled nodes.

$$[\boldsymbol{M}] = \begin{bmatrix} [\boldsymbol{\delta}] & [\boldsymbol{T}_{\boldsymbol{21}}^{\boldsymbol{u}}] & [\boldsymbol{T}_{\boldsymbol{31}}^{\boldsymbol{u}}] & \cdots & [\boldsymbol{T}_{\boldsymbol{m1}}^{\boldsymbol{u}}] \\ [\boldsymbol{T}_{\boldsymbol{12}}^{\boldsymbol{u}}] & [\boldsymbol{\delta}] & [\boldsymbol{T}_{\boldsymbol{32}}^{\boldsymbol{u}}] & \cdots & [\boldsymbol{T}_{\boldsymbol{m2}}^{\boldsymbol{u}}] \\ \vdots & & & & \vdots \\ [\boldsymbol{T}_{\boldsymbol{1m}}^{\boldsymbol{u}}] & [\boldsymbol{T}_{\boldsymbol{2m}}^{\boldsymbol{u}}] & [\boldsymbol{T}_{\boldsymbol{3m}}^{\boldsymbol{u}}] & \cdots & [\boldsymbol{\delta}] \end{bmatrix}$$

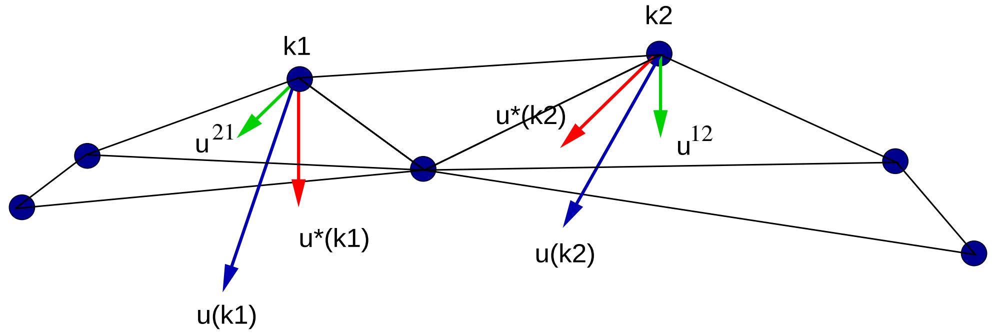

Figure 10: $\mathbf{u}^*_{k_1}$ and $\mathbf{u}^*_{k_2}$ are the initial imposed displacements. $\mathbf{u}^{12}$ and $\mathbf{u}^{21}$ are, respectively, the displacement induced by $\mathbf{u}^*_{k_1}$ on node k_2 and the displacement induced by $\mathbf{u}^*_{k_2}$ on node k_1. $\mathbf{u}_{k_1}$ and $\mathbf{u}_{k_2}$ are the displacements resulting from the application of the superposition principle.

with

$$[\boldsymbol{\delta}] = \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 \end{bmatrix}$$

The vector of modified constraints $[\tilde{\boldsymbol{u}}^*]$ that must be applied to the mesh is determined by:

$$[\tilde{\boldsymbol{u}}^*] = [\boldsymbol{M}]^{-1}[\boldsymbol{u}^*] \tag{8}$$

Consequently the superposition principle can be applied to compute the total displacement of each node **n**:

$$\mathbf{u}_n = \sum_{l=1}^{m} [\boldsymbol{T}^{\boldsymbol{u}}_{\boldsymbol{n}\boldsymbol{k}_l}][\tilde{\boldsymbol{u}}^*_{\boldsymbol{k}_l}] \tag{9}$$

The force associated to a controlled node **k** is determined by equation (10). This force is the one that should be applied to the node **k** to produce the displacement $\tilde{\mathbf{u}}^*$.

$$\mathbf{f}_k = [\boldsymbol{T}^{\boldsymbol{f}}_{\boldsymbol{n}\boldsymbol{k}}][\boldsymbol{u}^*] \tag{10}$$

4.3 Real-time quasi non-linear elastic deformations

The tensors of deformation $[\boldsymbol{T}^{\boldsymbol{u}}_{\boldsymbol{n}\boldsymbol{k}}]$ and force $[\boldsymbol{T}^{\boldsymbol{f}}_{\boldsymbol{n}\boldsymbol{k}}]$, pre-processed in the linear domain, allow the computation of the displacement and force vectors $\mathbf{u}_n$ and $\mathbf{f}_n$ associated with a node **n**. Consequently, we can deduce the orthogonal and collinear components $\mathbf{u}^r_n$, $\mathbf{f}^r_n$ and $\mathbf{u}^t_n$, $\mathbf{f}^t_n$, and equations (4), (6) can be applied at very low additional time cost.

A method based on the previous algorithm has been implemented to compute more realistic deformations taking into account biomechanical experimental results on soft tissue.

Computing times of table 1 have been obtained on a Dec Alpha 400 Mhz for a model of the liver with 1400 nodes and 6500 tetrahedra, i.e., the size of the matrix $[\boldsymbol{K}]$ is 4200×4200. The pre-computing step took approximatively 8 hours.

A method based on the previous algorithm has been implemented to compute more realistic deformations taking into account biomechanical experimental results on soft tissue.

	"normal" time	*"accelarated" time*
linear elasticity	15000 ms	7 ms
quasi non-linear elasticity	> 50000 ms	8 ms

Table 1. Computing times for a liver model.

5 Collision detection

Collision detection is a complex and well known problem in computer animation [LC92, GVP91, FS91, FVDFH90, Cam90, MW88]. When the real-time constraint is added, the difficulty is considerably increased. With physically-based models, most of the external forces are contact forces and in surgery simulation, the deformation is mainly driven by user interactions so an efficient collision detection algorithm is necessary.

The algorithm presented below considers a collision occurring between a simple rigid object and a complex deformable body. Moreover, in most cases the contact will happen when the deformable object is in its equilibrium configuration and the problem becomes equivalent to the contact between two rigid bodies. Consequently, our algorithm is based on the partitioning of the space in a regular grid [MW88]. With a data structure that can be updated very quickly, we can take into account the possibility of a collision with the deformed mesh.

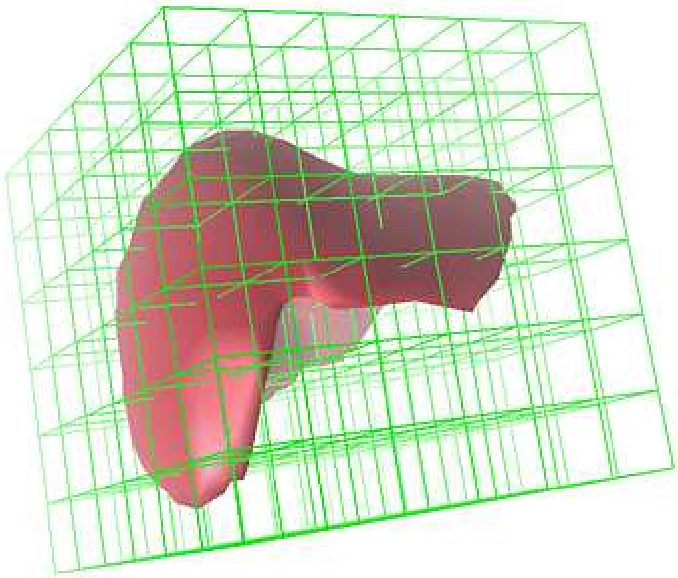

Figure 11: The bounding box of the mesh is decomposed into a set of buckets associated to a hash table.

The algorithm used in the simulator can be summarized as follows:

- First, the rigid moving object (a surgical tool in our application) is discretized into a set of points $\{\mathbf{p}_k\}$.

- The bounding box of the deformable object at equilibrium is divided in a set of parallelepipeds (also called *buckets* in the following). The number of buckets in the directions x, y and z is parameterizable with the condition that any bucket contains at least one node.
- A hash table is created and indexed on the coordinates of a node $\mathbf{n}$ located in the bucket. Thus, for a point $\mathbf{P}$ in the bounding box, we can easily determine the bucket including this point. By using the *code function* associated to the hash table, we can compute an entry in the table and find the list of nodes located at the same entry (which means that the nodes are in the same bucket as $\mathbf{P}$).
- For each point $\mathbf{p}_k$, a three-level search is performed:
  - for the set of nodes $\{\mathbf{n}_i\}$ located in the same bucket as $\mathbf{p}_k$, the Euclidean distances between $\mathbf{p}_k$ and each $\mathbf{n}_i$ is computed. Let d_{min} be the minimal distance. Compute the distance between $\mathbf{p}_k$ and each node $\mathbf{n}_j$ located in the 26-neighboring buckets. If a distance d is found to be less than d_{min} then $d_{min} = d$. The computing time is a function of the number of buckets and number of nodes per bucket. It is preferable to have a small number of nodes on average per bucket.
  - let $\mathbf{n}$ be the node associated to the minimal distance d_{min}. This node is shared by a set of triangles that can be very quickly determined when we associate an appropriate data structure to the triangulation.
  - finally, for each of these triangles, we check for an intersection with the segment defined by positions of node $\mathbf{p}_k$ at times t and $t-1$ (cf. figure 12).

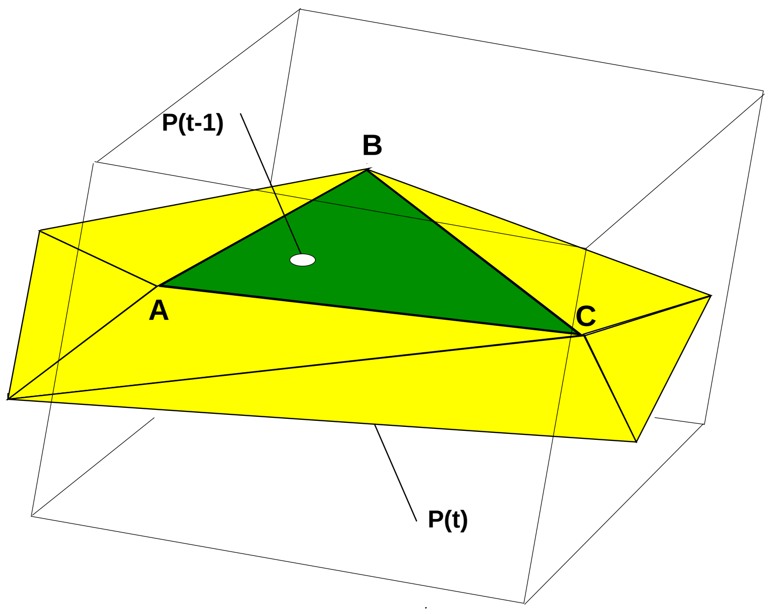

Figure 12: The collision point is computed as the intersection of the segment $[P_t, P_{t-1}]$ with the triangle (ABC).

The precision of the collision detection is consequently subtriangular. The barycentric coordinates of the intersection point are computed and used to breakdown the motion on the three vertices of the intersected triangle.

Let s_1, s_2 and s_3 be the three vertices of the intersected triangle and (d_x, d_y, d_z) the displacement of the intersection point with barycentric coordinates (k_1, k_2, k_3). The displacement applied at vertex v, $v = 1, 2, 3$ is:

$$d_x(s_v) = \alpha_v \, d_x$$
$$d_y(s_v) = \alpha_v \, d_y$$
$$d_z(s_v) = \alpha_v \, d_z$$

with

$$\alpha_v = (k_v + 1 - k_1^2 + k_2^2 + k_3^2) \quad v = 1, 2, 3$$

This algorithm is currently implemented with only one contact point located at the extremity of the surgical tool and works at about 500 Hz when no update of the bucket data structure is necessary. The collision detection with the deformed mesh is performed at about 300 Hz.

6 Experiments & validation

6.1 Validation of the non-linear model

To improve the previous results, we have meshed a cylinder of dimensions identical to the brain sample. The elastic parameters were chosen in the range of soft material (i.e. rubber). Then, the tensors of elementary deformations and forces were computed in the linear domain and finally this cylindrical mesh was deformed in real-time by applying a constant displacement field on its top.

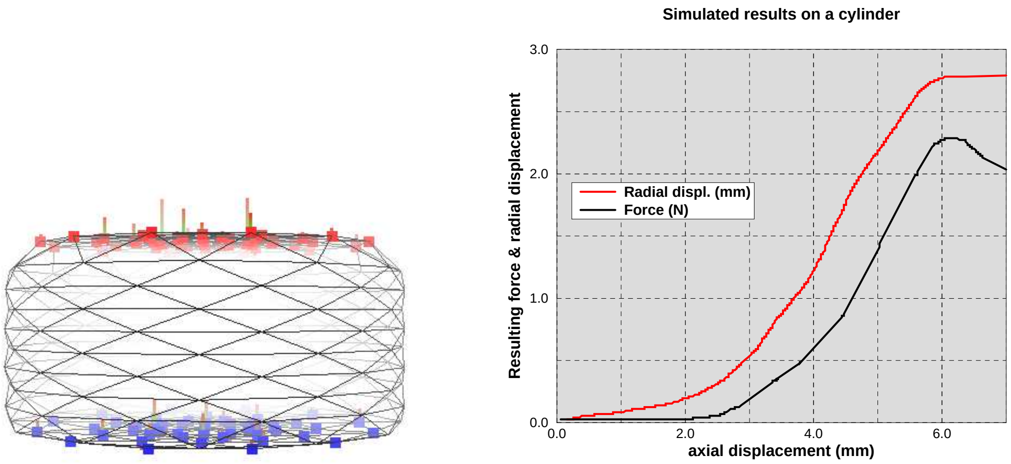

Figure 13: The curves describing the non-linear variation in force and radial displacement are measured, respectively, on the top of the cylinder and at nodes located on the circle at mid-height on the cylinder. The experimental biomechanical curves and the simulated one are exactly superimposed.

These experiments show that we can obtain a behavior for a synthetic cylindrical mesh very similar to a cylindrical sample of brain tissues. Nevertheless, the experimental curves issued from 2D stress/strain relationships are not sufficient to deduce a 3D behavior. We only assume that our results are correct since they give, at specified nodes, the same result as those expected and, at the other nodes, a global behavior that seems close to reality.

By generating this result for a given geometry we assume that we obtain a behavior close to the real one although most of the organs are not homogeneous and are made of tissues of variable stiffness (like the vessels in the liver).

6.2 Hepatic surgery simulation

The different "features" described previously (i.e., the geometrical and physical models, the collision detection algorithm, the speed-up algorithm and the force feedback interactions) have been integrated in our prototype of laparoscopic surgery simulator (see figure 16). These features run on a distributed architecture, based on a PC and a Dec Alpha station (see figure 14). The force feedback device (*Laparoscopic Impulse Engine*[6]) is connected to the PC (Pentium 166 Mhz) for technical reasons. The collision detection is also performed on the PC as well as the force evaluation since it is possible, thanks to our pre-processing algorithm, to split the computation of the forces and the computation of the deformation. The deformation is performed on the Alpha station (400 Mhz with 3D graphics hardware) as well as the display of the different parts of the scene (see figures 15 and 17) and the communication between the two machines is performed via an ethernet connection.

The data transmitted between the two computers is limited to the five degrees of freedom of the surgical tool plus some information issued from the collision detection algorithm. Consequently, we have a very high frequency ($>$ 300 Hz) in the simulation loop with a very little latency between force feedback and visual feedback.

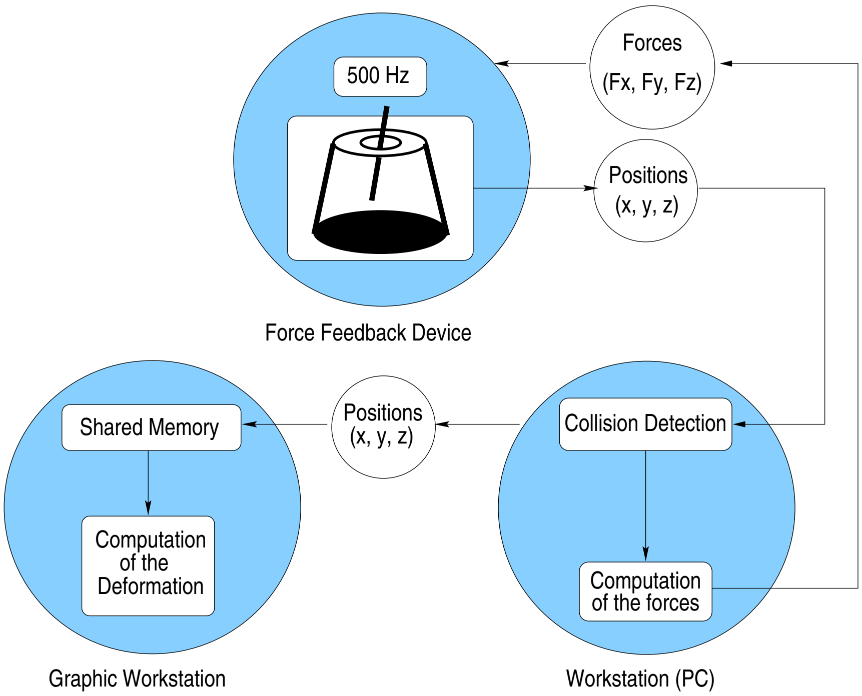

Figure 14: Description of the simulator architecture.

[6]The Laparoscopic Impulse Engine is a product from Immersion Coorporation Inc., California.

7 Conclusion & Perspectives

By the introduction of biomechanical properties in the deformation process, we have both enhanced the realism of the deformation/forces and the relevance of the model. The simulator has been tested by surgeons, specialized in laparoscopic surgery. It appears that the generated sensations are very close to reality, probably due to the addition of haptic feedback. Of course, the appreciation of the results is essentially qualitative, but our goal is not to compute an exact deformation of an organ. Besides, this objective seems very difficult to achieve given the actual knowledge in the field of soft tissues and the complexity of the interactions between the organs of the abdomen.

Nevertheless, several improvements can be made in the simulation process. One is adding a visco-elastic behavior (since the speed of endoscope has an importance in the deformation) and taking large displacements into account. Another important improvement is based on the possibility to simulate tissue cutting. Since the pre-computations allowing real-time interactions are dependent on the geometry of the mesh, it seems impossible to use only finite element models in the simulator. Actually, to take into account changes of mesh topology (due to tissue cutting), several tasks should be performed: remeshing of the cutting area, recalculation of the stiffness matrix and new pre-processing stage. These modifications cannot be done in real-time and a new approach should be used. Consequently, we are now working on a dynamic tissue cutting simulation technique that could be merged with the current model.

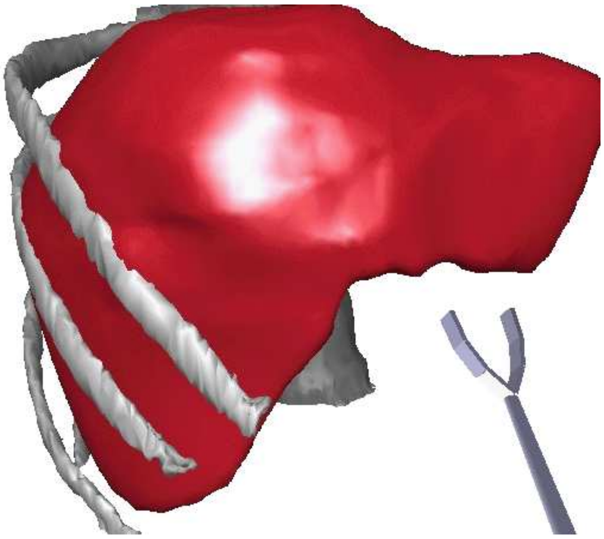

Figure 15: For a more realistic representation of the operative field, other elements can be introduced. Here the ribs are just introduced for visual feedback but we plan to add their interaction with the deformed liver for visual and haptic feedback.

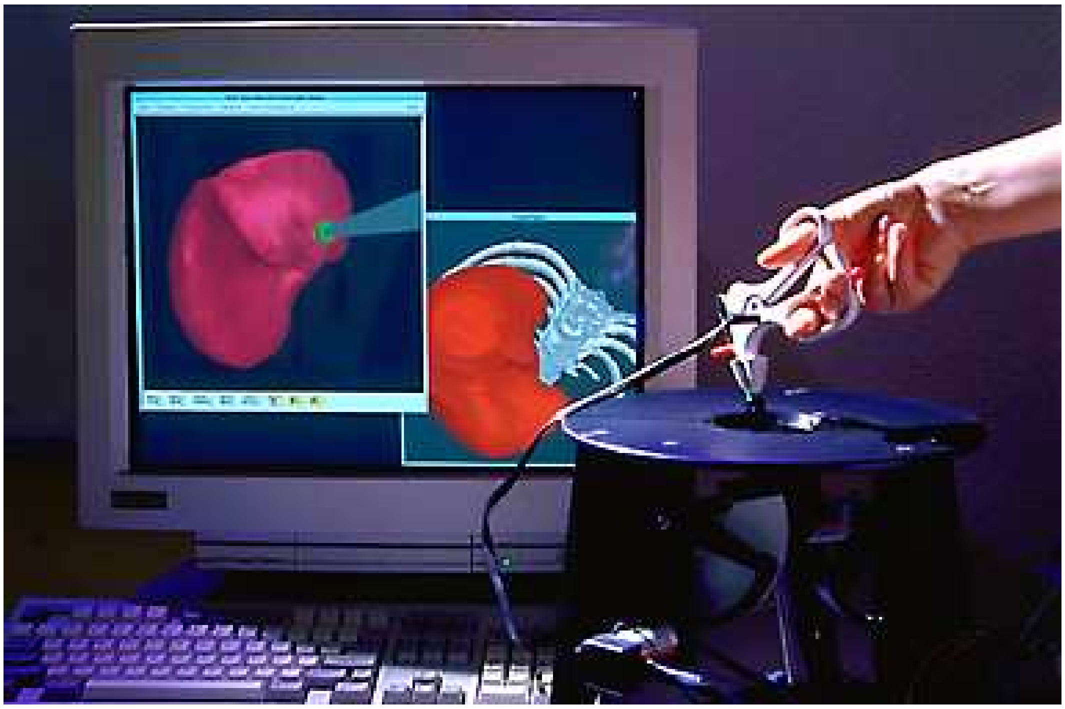

Figure 16: The surgeon manipulates the force feedback device. If a collision is detected with the surface of the virtual organ, the mesh is deformed in real-time and a non-linear reaction force is computed and sent back to the force feedback system.

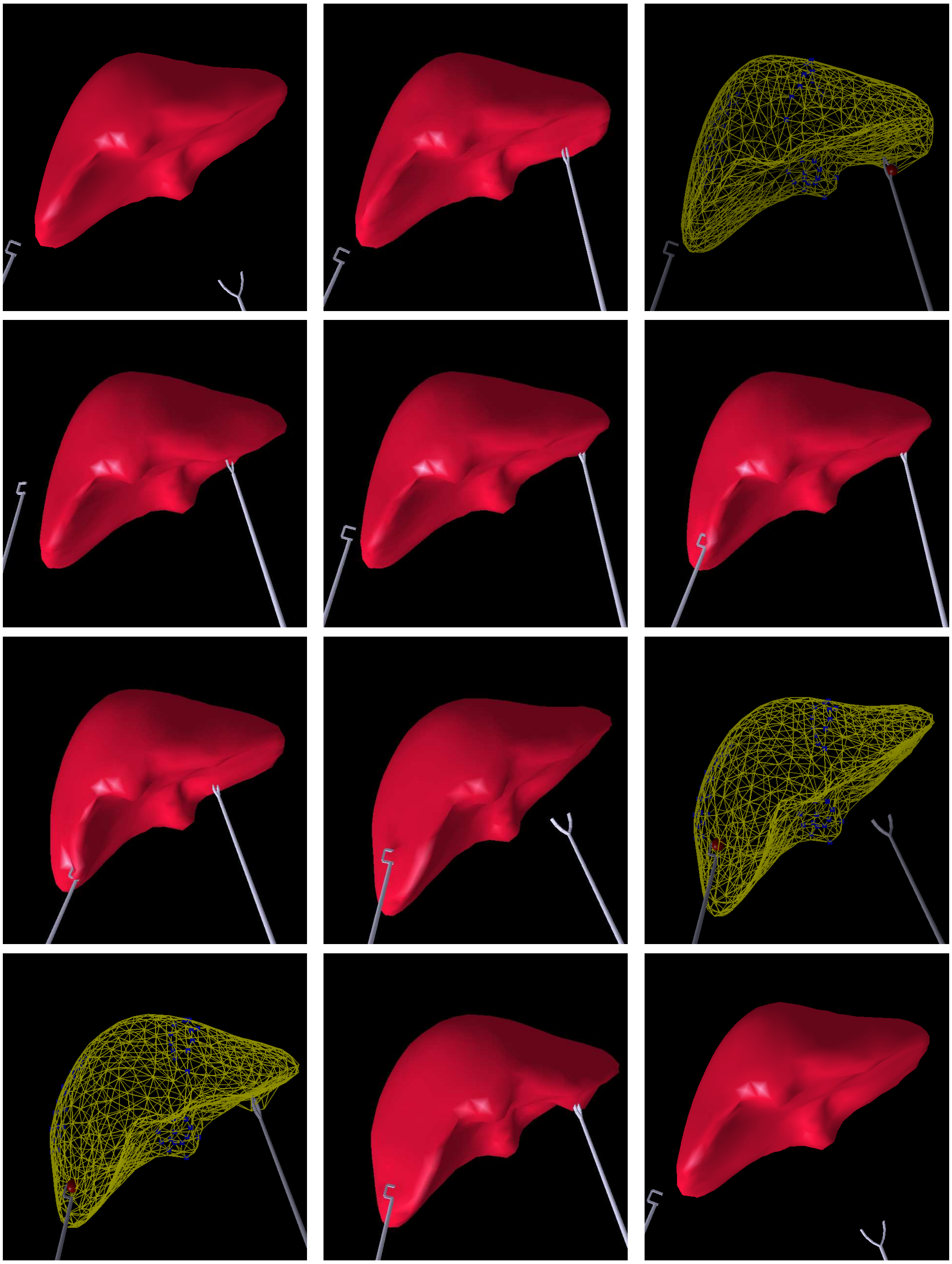

Figure 17: A sequence of images issued from a simulation. The liver model contains about 1500 nodes. The simulation was performed at 50Hz for the visual feedback and 300Hz for force feedback.

Acknowledgments

This project was partially funded by a contract with IRCAD with the collaboration of professor J. Marescaux, director of IRCAD and the help of Y. Russier and J.M. Clément. The authors are grateful for their collaboration in this project and their valuable advice. We also acknowledge Morten Bro-Nielsen for the interesting discussion we had during the development of the simulator and we thank Janet Bertot for proof reading this article.

Appendix A: Lagrange multipliers

The Lagrange multipliers operate on the variational formulation of the problem under consideration. The variational formulation of the elasticity problem can be written

$$\Pi = \frac{1}{2}u^T K u - u^T f \tag{11}$$

with the conditions $\dfrac{\partial \Pi}{\partial u_i} = 0$ for all i. Assume that we want to impose the displacement at the degree of freedom (a component of the node's position) u_i with:

$$u_i = {u_i}^\star$$

In the Lagrange multipliers method we modify the right hand side of (11) to obtain

$$\Pi^\star = \frac{1}{2}u^T K u - u^T F + \lambda(u_i - {u_i}^\star) \tag{12}$$

where λ is an additional variable, and invoke that $\delta\Pi^\star = 0$, which gives

$$\delta u^T K u - \delta u^T F + \lambda \delta u_i + \delta\lambda(u_i - {u_i}^\star) = 0 \tag{13}$$

Since δu and $\delta\lambda$ are arbitrary, we obtain

$$\left[\begin{array}{c|c} \boldsymbol{K} & \overline{\boldsymbol{K}} \\ \hline \overline{\boldsymbol{K}}^{\boldsymbol{T}} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u} \\ \hline \boldsymbol{\lambda} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f} \\ \hline \boldsymbol{u}^\star \end{array}\right] \tag{14}$$

where $[\overline{\boldsymbol{K}}]$ is a matrix composed of vectors $\mathbf{e}_i$ with all entries equal to zero, except the ith entry equal to one.

Contents

References

[Bat96] K.-J. Bathe. *Finite Element Procedures.* Prentice Hall, 1996. ISBN 0-13-301458-4.

[BCC95] E. Bainville, P. Chaffanjon, and P. Cinquin. Computer generated visual assistance during retroperitoneoscopy. *Computers in biology and medecine*, 2(25):165–171, May 1995.

[BG96] R. Baumann and D. Glauser. Force Feedback for Virtual Reality based Minimally Invasive Surgery Simulator. In *Medecine Meets Virtual Reality*, San Diego, CA, January 1996.

[BH95] W. Barfield and C. Hendrix. *Interactive Technology and the New Paradigm for Healthcare*, chapter 4: Factors Affecting Presence and Performance in Virtual Environments, pages 21–28. IOS Press and Ohmsha, 1995.

[BNC96] M. Bro-Nielsen and S. Cotin. Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. In *Proceedings of Eurographics'96 - Computer Graphics Forum*, pages 57–66, 1996.

[Bur96] G. Burdea. *Force and Touch Feedback for Virtual Reality.* John Wiley & Sons, New York, August 1996. ISBN 0-471-02141-5.

[Cam90] S. Cameron. Collision detection by four-dimensional intersection testing. *IEEE Transactions on Robotics and Automatons*, 6(3):291–302, 1990.

[CDA96] S. Cotin, H. Delingette, and N. Ayache. Real Time Volumetric Deformable Models for Surgery Simulation. In K. Hohne and R. Kikinis, editors, *Vizualisation in Biomedical Computing*, pages 535–540, Springer, 1996.

[CDCT96] S. Cotin, H. Delingette, J-M. Clément, V. Tassetti, J. Marescaux, and N. Ayache. Geometric and Physical Representations for a Simulator of Hepatic Surgery. In *Proceedings of Medecine Meets Virtual Reality IV*, pages 139–151, IOS Press, January 1996.

[CEO93] S. A. Cover, N. F. Ezquerra, and J. F. O'Brien. Interactively Deformable Models for Surgery Simulation. *IEEE Computer Graphics and Applications*, 13(6):68–75, 1993.

[CHP89] J. Chadwick, D. Haumann, and R. Parent. Layered construction of deformable animated characters. In *Computer Graphics (SIGGRAPH'89)*, pages 243–252, July 1989.

[CM97] K. Chinsei and K. Miller. Compression of Swine Brain Tissue - Experiment *In Vitro*. *Journal of Mechanical Engineering Laboratory*, 106–115, 1997.

[CZ92] D. T. Chen and D. Zeltzer. Pump it up: Computer animation of a biomechanically based model of the muscle using the finite element method. *Computer Graphics (SIGGRAPH'92)*, (26):89–98, July 1992.

[Del94a] H. Delingette. *Simplex Meshes: a General Representation for 3D Shape Reconstruction.* Technical Report 2214, INRIA, March 1994.

[Del94b] H. Delingette. Simplex meshes: a general representation for 3d shape reconstruction. In *Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, USA, 1994.

[DG95] M. Desbrun and M.-P. Gascuel. Animating soft substances with implicit surfaces. In *Computer Graphics (SIGGRAPH '95)*, pages 287–290, august 1995.

[FS91] M.C. Forgue and M. Shinya. Real-time impact dynamics simulation. *Compugraphics*, 427–435, 1991.

[Fun93] Y. C. Fung. *Biomechanics - Mechanical Properties of Living Tissues.* Springer-Verlag, second edition, 1993.

[FVDFH90] J.D. Foley, A. Van Dam, S. Feiner, and J. Hughes. *Computer Graphics - Principles and Practice.* Addison Wesley, 1990.

[GMTT89] J. P. Gourret, N. Magnenat-Thalmann, and D. Thalmann. Simulation of Object and Human Skin Deformations in a Grasping Task. In *Computer Graphics (SIGGRAPH'89)*, pages 21–31, Boston, MA, USA, July 1989.

[GSMF97] S. Gibson, J. Samosky, A. Mor, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, and N. McKenzie. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback . In J. Troccaz, E. Grimson, and R. Mosges, editors, *Proceedings of the First Joint Conference CVRMed-MRCAS'97*, pages 369–378, March 1997.

[GVP91] M.-P. Gascuel, A. Verroust, and C. Puech. A modeling system for complex deformable bodies suited to animation and collision processing. *Journal of Visualization and Computer Animation*, 2(3):82–91, August 1991. A shorter version of this paper appeared in *Graphics Interface'91.*

[Jou95] A. Joukhadar. Energy based adaptive time step and inertia-matrix based adaptive discretization for fast converging dynamic simulation. In *Proc. of the Int. Workshop on Visualisation and Mathematics*, Berlin (D), may 1995.

[JR95] B.G. Jackson and L.B. Rosenberg. *Interactive Technology and the New Paradigm for Healthcare*, chapter 24: Force Feedback and Medical Simulation, pages 147–151. IOS Press, 1995.

[KE96] Girod B. Keeve E., Girod S. Craniofacial surgery simulation. In *Proceedings of the 4th International Conference on Visualization in Biomedical Computing VBC'96*, pages 541–546, Hamburg, Germany, Octobre 1996.

[KGCvB96] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser, and Y. Parish. Simulating facial surgery using finite element methods. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, pages 421–428, ACM SIGGRAPH, Addison Wesley, August 1996.

[KN93] U.G. Kuehnapfel and B. Neisius. CAD-Based Graphical Computer Simulation in Endoscopic Surgery. *End. Surg.*, 1:181–184, 1993.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface reconstruction algorithm. 21(4), July 1987.

[LC92] M. Lin and J. Canny. Efficient collision detection for animation. In *Third Eurographics Worshop on Animation and Simulation*, Cambridge, England, Octobre 1992.

[Le 93] Le Tallec, P. and Rahier, C. and Kaiss, A. Three Dimensional Incompressible Viscoelasticity in Large Strains. *Computer Methods in Apllied Mechanics and Engineering*, 109:233–258, 1993.

[LJFC91] A. Luciani, S. Jimenez, J.L. Florens, C. Cadoz, and O. Raoult. Computational physics: a modeler simulator for animated physical objects. In *Eurographics Workshop on Animation and Simulation*, pages 425–437, Vienna, 1991.

[MC97] P. Meseure and C. Chaillou. Deformable Body Simulation with Adaptative Subdivision and Cuttings. In *Proceedings of the WSCG'97*, pages 361–370, February 1997.

[MD97] J. Montagnat and H. Delingette. Volumetric Medical Images Segmentation using Shape Constrained Deformable Models. In J. Troccaz, E. Grimson, and R. Mosges, editors, *Proceedings of the First Joint Conference CVRMed-MRCAS'97*, Springer, March 1997.

[Mil88] G. Miller. The motion dynamics of snake and worms. In *Computer Graphics (SIGGRAPH'88)*, pages 169–173, Atlanta (USA), aug 1988.

[MW88] M. Moore and J. Wilhelms. Collision Detection and Response for Computer Animation. In *Computer Graphics (SIGGRAPH'88)*, pages 289–298, August 1988.

[NTBG91] A. Norton, G. Turk, B. Bacon, J. Gerth, and P. Sweeney. Animation of Fracture by Physical Modeling. *The Visual Computer*, 7:210–219, 1991.

[PB88] J. C. Platt and A. H. Barr. Constraint Methods for Flexible Models. In *Computer Graphics (SIGGRAPH'88)*, pages 279–288, 1988.

[PTH95] B. Pflesser, U. Tiede, and Hoehne. Towards realistic visualization for surgery rehearsal. In *Computer Vision, Virtual Reality and Robotics in Medecine*, pages 487–491, Springer, April 1995.

[SBMH94] M. A. Sagar, D. Bullivant, G. Mallinson, P. Hunter, and I. Hunter. A Virtual Environment and Model of the Eye for Surgical Simulation. In *Computer Graphics (SIGGRAPH'94)*, pages 205–212, 1994.

[Spe92] Thomas H. Speeter. Three Dimensional Finite Element Analysis of Elastic Continua for Tactile Sensing. *The International Journal of Robotics Research*, 11 No 1:1–19, February 1992.

[SR95] G. J. Song and N. P. Reddy. Tissue Cutting In Virtual Environment. In *Medecine Meets Virtual Reality IV*, pages 359–364, IOS Press, 1995.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleisher. Elastically Deformable Models. In *Computer Graphics (SIGGRAPH'87)*, pages 205–214, July 1987.

[Wat92] Keith Waters. A physical model of facial tissue and muscle articulation derived from computer tomography data. In *Visualization in Biomedical Computing (VBC'92)*, Chappel Hill, NC, 1992.

[Zie77] O.C. Zienkiewicz. *The finite element method.* McGraw-Hill, London, 3 edition, 1977.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399