Anisotropic Elasticity and Forces Extrapolation to Improve Realism of Surgery Simulation

G. Picinbono, J.C. Lombardo, H. Delingette, N. Ayache

Epidaure Project
I.N.R.I.A.
06902 Sophia-Antipolis Cedex, BP 93, France

Abstract

In this paper, we describe the latest developments of the hepatic surgery simulator prototype developed at INRIA. The goal of this simulator is to provide a realistic training testbed for performing laparoscopic procedures. Therefore, its main functionality is to allow to cut and deform tridimensional anatomical models with the help of two virtual laparoscopic surgical instruments. Thoughout this paper, we present the general features of the simulator including the creation of anatomical models from medical imaging, the implementation of different biomechanical models based on linear elasticity and finite element theory and the integration of two force-feedback devices in the simulation platform. More precisely, we describe two new important developments that improve the overall realism of the simulator. First, we can create biomechanical models 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 one privileged direction of deformation. The second 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 sensation requires a frequency update of the applied forces typically above 300Hz (haptic real-time). Thus, we propose an extrapolation algorithm of the forces computed by the deformable model in order to reach haptic real-time.

1 Introduction

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

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

lems in surgical simulation. To study all the prob-

Figure 1: *surgery simulator prototype*

lems related to surgery simulation, INRIA gathered six teams in a joined action AISIM [1]. A laparoscopic surgery simulators prototype has been developed (figure 1), based on linear elasticity theory, finite element method and utilization of force feedback devices. In this article, we will give a general description of the two key components of the simulator: the implementation of real-time physical models and the integration of force feedback devices. The first component is centered on the "Tensor/Mass" model [10] which allows real-time deformations and topology changes, and we describe the generalization of linear elasticity to anisotropic materials. The second component deals with the problem of haptic rendering. Using a force feedback device within the framework of deformable object simulation remains challenging because of the trade-off between the computation time needed by realistic physical models and the high update frequencies required for real-time haptic rendering. Indeed, it is well-known that the update frequency for haptic feedback is at least ten times greater than for visual rendering (25Hz for visual real-time and more than 300Hz for haptic real-time). To solve this trade-off, we propose to extrapolate contact forces such that we can use visual rendering frequency update without a great loss in the quality of haptic feedback.

2 Deformable models

Little bibliography about deformable models

2.1 Linear elasticity

The physical behavior of a soft tissue may be considered as linear elastic if the displacements applied to it remain small [11, 12] (less than 10% of the mesh size); as displacements increase, the linear elastic approximation becomes less and less valid. In particular, several biological materials are nearly incompressible since they are mainly composed of water. Such behavior cannot be modeled with linear elasticity for large displacements.

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 :

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

From the principal invariants of E :

$$\begin{cases} l_1 & = & tr\, E \\ l_2 & = & tr\, E^2 \end{cases} \tag{2}$$

we can express the linear elastic energy W_{Elastic}, for homogeneous isotropic materials, by the following formula (see [9]) :

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

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

2.2 Anisotropic elasticity

Isotropic behavior is a strong restriction to model human tissues. In fact a lot of anatomical structures like muscles, tendons, ligaments, blood vessels, are strongly anisotropic. That is why we are particularly interested in models of materials with one privileged direction, which are called "transversally isotropic" materials.

Bibliography about anisotropic elasticity (Kaiss-Le Tallec, Spencer, Weiss-Maker-Govindjee)

For such 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. We consider an elementary cube 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 *components of strain* corresponding respectively to the relative elongations (ϵ_x, ϵ_y and ϵ_z) in the three directions of the cube and the relative changes of angles (γ_{xy}, γ_{xz} and γ_{yz}) between the faces (shear) of the cube. For *small displacements*, these six numbers are related to the derivatives of $\mathbf{U}(x, y, z)$ by :

$$\epsilon_x = \frac{\partial u}{\partial x} \qquad \epsilon_y = \frac{\partial v}{\partial y} \qquad \epsilon_z = \frac{\partial w}{\partial z} \tag{4}$$
$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \quad \gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \quad \gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}$$

The strain tensor E can then be written as :

$$E = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} = \begin{bmatrix} \epsilon_x & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{xy} & \epsilon_y & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{xz} & \frac{1}{2}\gamma_{yz} & \epsilon_z \end{bmatrix} \tag{5}$$

The isotropic elastic energy of equation 3 can be written as :

$$\begin{aligned} W_{\text{Elastic}} = & \quad \frac{\lambda}{2}\left(\epsilon_x + \epsilon_y + \epsilon_z\right)^2 + \mu\left(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\right) \\ + & \quad \frac{\mu}{2}\left(\gamma_{xy}^2 + \gamma_{xz}^2 + \gamma_{yz}^2\right) \; dx\, dy\, dz \end{aligned} \tag{6}$$

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 to $\mathbf{a_0}$.
- $\Delta\lambda = \lambda^L - \lambda^T$ and $\Delta\mu = \mu^L - \mu^T$

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 6, the anisotropic contribution ΔW_{Aniso} defined as :

$$\begin{aligned} \Delta W_{\text{Anisob}} = & \quad \frac{\Delta\lambda}{2}\left({\epsilon_z}^2 + 2\epsilon_x\epsilon_z + 2\epsilon_y\epsilon_z\right) \\ & +\Delta\mu\ {\epsilon_z}^2 + \frac{\Delta\mu}{2}\left({\gamma_{xz}}^2 + {\gamma_{yz}}^2\right) \\ = & \quad \Delta\lambda\ (\epsilon_x + \epsilon_y + \epsilon_z)\epsilon_z \\ & +2\Delta\mu\ \left({\epsilon_z}^2 + \frac{1}{4}{\gamma_{xz}}^2 + \frac{1}{4}{\gamma_{yz}}^2\right) \\ & -\left(\frac{\Delta\lambda}{2} + \Delta\mu\right)\ {\epsilon_z}^2 \end{aligned} \tag{7}$$

In the general case, this anisotropic elastic energy can be simply written with the introduction of two new invariants l_4 and l_5 linked to the strain tensor E and direction $\mathbf{a_0}$:

$$\begin{cases} l_4 = & {\mathbf{a_0}}^t\, E\, \mathbf{a_0} \\ l_5 = & {\mathbf{a_0}}^t\, E^2\, \mathbf{a_0} \end{cases} \tag{8}$$

Then the anisotropic energy can be written as :

$$\Delta W_{\text{Aniso}} = l_4 l_1 \Delta\lambda + 2 l_5 \Delta\mu - l_4^2\left(\frac{\Delta\lambda}{2} + \Delta\mu\right) \tag{9}$$

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

$$W_{\text{Trans-iso}} = W_{\text{Elastic}} + \Delta W_{\text{Aniso}} \tag{10}$$

2.3 Finite element formulation

All our deformable models are based on a finite element model consisting of a conformal tetrahedral mesh. At each point $\mathbf{M}(x, y, z)$ in the tetrahedron $\mathbf{T}_i$,

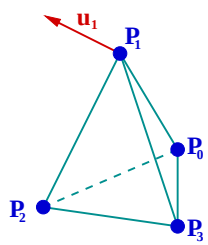

Figure 2: P_1 *tetrahedral finite element*

the displacement vector is expressed as a function of the displacements of the vertices $\mathbf{P}_k$:

$$\mathbf{U}(x, y, z) = \sum_{k=0}^{3} \mathbf{U}_k \lambda_k(x, y, z) \tag{11}$$

where $\{\lambda_k, k = 0, ..., 3\}$ are the barycentric coordinates of $\mathbf{M}$ in the tetrahedron.
We can then write the isotropic linear elastic energy as a function of the displacement of each vertex :

$$W_{\text{Elastic}} = \sum_{k=0}^{3}\sum_{l=0}^{3} \mathbf{U}_k^t [K_{kl}^{\mathbf{T}_i}] \mathbf{U}_l \tag{12}$$

where $[K_{kl}^{\mathbf{T}_i}] = \lambda\alpha_k\alpha_l^t + \mu\alpha_l\alpha_k^t + \mu(\alpha_k^t\alpha_l)Id_3$ is the tetrahedron contribution to the stiffness tensor of the edge $(\mathbf{P}_k, \mathbf{P}_l)$ (or of the vertex $\mathbf{P}_k$ if $k = l$). $(\alpha_\mathbf{k}, k = 0..3)$ are the shape vectors of the tetrahedron. Likewise, we obtain a similar equation for the anisotropic part of the elastic energy:

$$\begin{aligned} W_{\text{Aniso}} &= \sum_{k=0}^{3}\sum_{l=0}^{3} \mathbf{U}_k^t [A_{kl}^{\mathbf{T_i}}] \mathbf{U}_l && (13) \\ A_{kl}^{\mathbf{T}_i} &= \Delta\lambda(\mathbf{a}_0\mathbf{a}_0^t)(\alpha_k\alpha_l^t) && (14) \\ &+ \frac{\Delta\mu}{2}\left[(\mathbf{a}_0\mathbf{a}_0^t)(\alpha_l\alpha_k^t) + (\alpha_l\alpha_k^t)(\mathbf{a}_0\mathbf{a}_0^t)\right. \\ &\quad \left. +(\alpha_k^t\alpha_l)(\mathbf{a}_0\mathbf{a}_0^t) + (\mathbf{a}_0\mathbf{a}_0^t) : (\alpha_k\alpha_l^t)Id_3\right] \\ &- (\frac{\Delta\lambda}{2} + \Delta\mu)(\mathbf{a}_0\mathbf{a}_0^t)(\alpha_k\alpha_l^t)(\mathbf{a}_0\mathbf{a}_0^t) \end{aligned}$$

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

$$\begin{aligned} \mathbf{F}_k^{\mathbf{T}_i} &= \left(\frac{\partial(W_{\text{Elastic}} + \Delta W_{\text{Anisotropic}})}{\partial \mathbf{U}_k}\right)^t \\ &= 2\sum_{l=0}^{3}[K_{kl}^{\mathbf{T}_i} + A_{kl}^{\mathbf{T}_i}]\mathbf{U}_l \\ &= [G_{kk}^{\mathbf{T}_i}]\mathbf{U}_k + \sum_{l=0;l\neq k}^{3}[G_{kl}^{\mathbf{T}_i}]\mathbf{U}_l && (15) \end{aligned}$$

Construction of the tensor:
We obtain the global force applied on vertex $\mathbf{P}_k$ by adding the contributions of all the tetrahedra sharing this vertex:

$$\begin{aligned} \mathbf{F}_k &= [G_{kk}]\mathbf{U}_k + \sum_{\mathbf{P}_l \in N(\mathbf{P}_k)}[G_{kl}]\mathbf{U}_l && (16) \\ G_{kk} &= \sum_{\mathbf{T}_i \in N(\mathbf{P}_k)} G_{kk}^{\mathbf{T}_i} && (17) \\ G_{kl} &= \sum_{\mathbf{T}_i \in N(\mathbf{P}_k, \mathbf{P}_l)} G_{kl}^{\mathbf{T}_i} && (18) \end{aligned}$$

Because the elastic energy is quadratic with respect to the displacement vector, the forces $\mathbf{F}_k$ are linear functions of the displacement vectors of each node $\mathbf{P}_k$.

2.4 Pre-computed model

10 lines about pre-computed model with a link to [7]

- Construct the global stiffness matrix $[G]\ U = F$
- Solve this linear systeme (with conjugate gradient) for elementary forces applied on each node in each space direction, and store the results.
- Use the pre-computations during real-time simulation:
  Apply "general forces" = Apply $\sum$ "elementary forces".
- Results: very efficient simulation, but no topology changes

$\Longrightarrow$ another model which can handle cutting: no pre-computations.

2.5 Tensor/Mass model

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

2.5.1 Numerical integration

We use a Newtonian differential equation:

$$m_i\frac{d^2\mathbf{P}_i}{dt^2} = \gamma_i\frac{d\mathbf{P}_i}{dt} + \mathbf{F}_i \qquad (19)$$

as the equation governing the motion of our linear elastic model. This equation is related to the differential equation found in continuum mechanics [3] :

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \qquad (20)$$

Following finite elements theory, the mass $\mathbf{M}$ and damping $\mathbf{C}$ matrices are sparse matrices that are related to the stored physical properties of each tetrahedron. In our case, we consider that $\mathbf{M}$ and $\mathbf{C}$ are diagonal matrices, i.e. that mass and damping effects are concentrated at vertices. This simplification called *mass-lumping* decouples the motion of all nodes and therefore allows to write equation 20 as the set of independent differential equations (19) 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 $t+1$:

$$\left(\frac{m_i}{\Delta t^2} - \frac{\gamma_i}{2\Delta t}\right)\mathbf{P}_i^{t+1} = \mathbf{F}_i + \frac{2m_i}{\Delta t^2}\mathbf{P}_i^t - \left(\frac{m_i}{\Delta t^2} + \frac{\gamma_i}{2\Delta t}\right)\mathbf{P}_i^{t-1} \qquad (21)$$

2.5.2 Simulation of cutting

One of the basic task in surgery simulation consists in cutting. With the dynamic linear elastic model, this task can be achieved in real-time.

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

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

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

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

2.6 Results

2.7 hybrid model

10 lines about hybrid model with a link to [10]

3 Force feedback

Thanks to retina persistence a visual sensation of continuity is provided with relatively low frequencies (about 25Hz). But haptic rendering requires update frequencies ranging from 300Hz for soft objects to 10kHz for rigid contact.
In the literature, two main approaches to obtain haptic real-time are developed:

- computing forces empirically [4], 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 from purely geometric constraint.
- using a simplified physical model. The simplification can be done in two ways, either by decreasing the size of the model [6] or by performing as much pre-computation as possible [7]. In the latter case, the model cannot undergo any topological change. A combination of these two possibilities have been proposed in [8, 10] where an hybride model combines a large model where unitary deformations have been pre-computed, with a smaller model allowing topology changes.

We propose to enhance the hybrid model by using force interpolation. Indeed, it has been shown ([5]) that if haptic rendering is very precise (we can feel vibrations until 10kHz, and force variations between 30 and 300 Hz), the gesture is slower (from 1Hz for the answer to an unexpected signal and 10Hz for a reflex action). Thus, the applied forces must be updated at a high rate, but, because it is related to user's action, their evolution is quite slow. The idea is to estimate the force between two time steps of the deformable model simulation.

First, we will describe our simulator architecture. Then we present various force extrapolation schemes we tried.

3.1 Simulator architecture

As the update rates of visual rendering and haptic rendering are quite different, it is natural to divide the simulator in two component. As shown in figure 3, the former component manages the force feedback loop while the latter component manages the object deformation loop. In our setup, the two force feedback devices (*Laparoscopic Impulse Engines*[1]) are driven by dedicated computer. The communication between the simulation workstation and the force feedback workstation is performed via a classical *Ethernet* connection, using *UDP sockets*.

3.2 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 series of parameters $(t_n, \mathbf{P}_n, \mathbf{F}_n)$ representing

[1] http://www.immerse.com

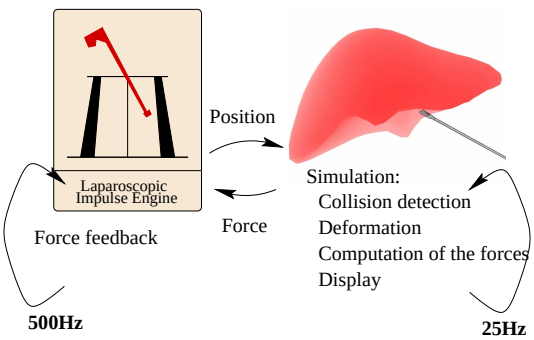

Figure 3: *Hepatic surgery simulator*

the force $\mathbf{F}_n$ applied to the tool in position $\mathbf{P}_n$ at time t_n. The time step between two successive t_n is about 0.04s (visual real-time) and is not necessarily constant. Good quality force feedback can be reached by an update of the force at about 500Hz. So, we must choose an extrapolation function $\mathbf{F}(t)$ providing an estimation of the force to apply to the tool 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 feedback forces. This has several advantages. First, it does not need any additional forces. Then, as the applied force results from the deformation computation, such an extrapolation scheme ensures that only valid forces are applied without risking to damage 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 noticeable. But we must face a new problem, as the applied forces are not the ones that the simulation of the deformable model computes, they can be arbitrarily large. These force amplitude peaks occur especially when the time step increases.

The force changes are mainly due to the tool movement. Furthermore, it is possible to query at a very high rate the position of the tool during the extrapolation. These observations lead us to develop a force estimator based on the tool position: **linear extrapolation over position**. We project the current tool

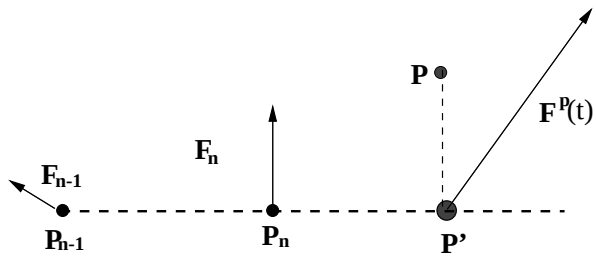

Figure 4: Tool position projection for extrapolation over position

position $\mathbf{P}$ in $\mathbf{P}$' on the line defined by the two previous tool position $\mathbf{P}_{n-1}$ and $\mathbf{P}_n$ (figure 4). We can then consider the norm ratio for extrapolation:

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

We can 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.

3.3 Results

The first experiment was simply performed by using our surgery simulator with one of the three extrapolation schemes activated. The constant extrapolation gives us the sensation of touching a rough surface. The extrapolation according to time is an improvement, but the device sometimes send some unexpected large forces. As soon as the tool movement is slow enough, the sensation given by the extrapolation over position are smooth. Of course, this is a very subjective evaluation, and we tried to compare the three methods more objectively.

The time, the tool position, and the force computed by the simulation of the deformable model were recorded during several surgery simulation sessions. We interpolate the force *a posteriori* to have a reference for the computation of the errors. For an easier understanding of the figures, we plot 2D experiments. We also prefer a polar representation of the forces.

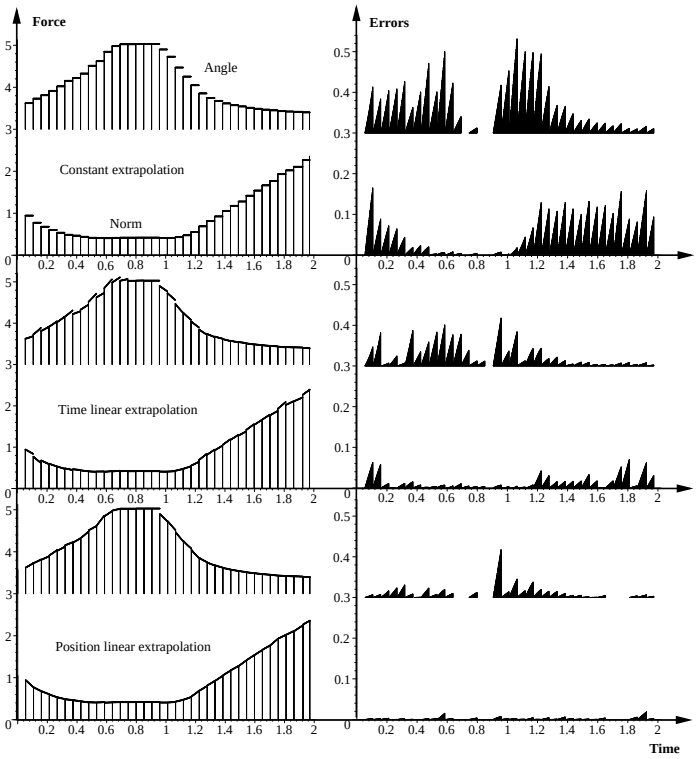

Figure 5: Evaluating the different extrapolation methods

The left column shows the original data set with impulses and the extrapolated one with a line. The difference between the extrapolated and the interpolated forces, which is taken as a measure of the error, is plotted, also in a polar fashion, in the right column of the figure 5. Theses plots shows the extrapolation results for a input data set at 20Hz.

We note that the linear extrapolation over position gives very interesting results (very few discontinuities and no singular force). We tried the same type of experiment with different simulation frequencies and with different tool movements. The position linear extrapolation always gave the best results, which is confirmed by sensation received during simulation.
Other tests were performed with different speeds for the tool movement. They show that if the toll motion is too fast or if the update frequency is too low, the error becomes important. Studies have shown that the surgeon's gesture is performed at about 0.01 $m.s^{-1}$. With such a speed and a simulation running at a visual real-time rate (about 20Hz), the linear extrapolation over position gives very good results.

4 Conclusion

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

[4] C. Bosdogan, C. Ho, M. A. Srinivasan, S. D. Small, and S. L. Dawson. Force Interaction in Laparoscopic Simulation: Haptic Rendering Soft Tissues. *Proceedings of the Medecine Meets Virtual Reality (MMVR'6) Conference*, pages 28–31, January 1998.

[5] T.L. Brooks. Telerobotic response requirements. *Proceedings of the IEEE International Conference on systems, Man and Cybernetices*, pages 113–120, 1990. Los Angeles.

[6] 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), Atlanta, Georgia*, pages 190–197, March 1998.

[7] 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.

[8] S. Cotin, H. Delingette, and N. Ayache. Efficient linear elastic models of soft tissues for real-time surgery simulation. Technical Report RR-3510, INRIA, 1998.

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

[10] H. Delingette, S.Cotin, and N.Ayache "A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation," *Computer Animation*, 1999.

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

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