10th International Symposium of Robotics Research
November 9-12, 2001. Lorne, Victoria, Australia.

Towards a Realistic Medical Simulator using Virtual Environments and Haptic Interaction

C. Laugier, C. A. Mendoza, K. Sundaraj
INRIA[a] Rhône-Alpes & GRAVIR[b] , SHARP Project,
ZIRST-655, Avenue de l'Europe, 38330 Montbonnot Saint Martin, FRANCE.
Email: [Christian.Laugier, César.Mendoza-Serrano, Kenneth.Sundaraj]@inrialpes.fr

Invited Paper for ISRR 2001

Abstract

This paper presents our experience towards the conception of a virtual reality medical simulator coupled with haptic interaction aimed at training surgeons. This area of research has a long history and a wide variety of approaches have been used. Generally, human tissue can be considered as a deformable body of viscoelastic material. To enable dynamic simulation of these bodies, we have patched three well known physical models onto their geometrical model : mass-spring networks which is more of a discrete object model, finite element method (FEM) based on continuum mechanics and recently long element method (LEM) which we believe to be more promising. We make some comparisons between these models. We also present some numerical resolution methods for simulation of deformable bodies. As far as real-time interactions are concerned, we present our work on collision detection, haptic interaction and topology modifications. In the haptic system, we separate the physical simulation and the haptic interaction to ensure stability; the link between the two process is acheived by means of a local model which will be eloborated. We present some experimental results to highlight these works.

Keywords

Medical simulators, physical models, numerical methods, deformable objects, virtual reality, haptic interaction.

[a] Inst. Nat. de Rech. en Informatique et en Automatique.
[b] Lab. d'Informatique GRAphique, VIsion et Robotique.

Towards a Realistic Medical Simulator using Virtual Environments and Haptic Interaction

C. Laugier, C. A. Mendoza, K. Sundaraj
INRIA Rhône-Alpes & GRAVIR , SHARP Project,
ZIRST-655, Avenue de l'Europe, 38330 Montbonnot Saint Martin, FRANCE.
Email: [Christian.Laugier, César.Mendoza-Serrano, Kenneth.Sundaraj]@inrialpes.fr

Invited Paper for ISRR 2001

Abstract— **This paper presents our experience towards the conception of a virtual reality medical simulator coupled with haptic interaction aimed at training surgeons. This area of research has a long history and a wide variety of approaches have been used. Generally, human tissue can be considered as a deformable body of viscoelastic material. To enable dynamic simulation of these bodies, we have patched three well known physical models onto their geometrical model : mass-spring networks which is more of a discrete object model, finite element method (FEM) based on continuum mechanics and recently long element method (LEM) which we believe to be more promising. We make some comparisons between these models. We also present some numerical resolution methods for simulation of deformable bodies. As far as real-time interactions are concerned, we present our work on collision detection, haptic interaction and topology modifications. In the haptic system, we separate the physical simulation and the haptic interaction to ensure stability; the link between the two process is achelved by means of a local model which will be eloborated. We present some experimental results to highlight these works.**

Keywords— **Medical simulators, physical models, numerical methods, deformable objects, virtual reallty, haptic interaction.**

I. Introduction

COMPUTER assisted surgery and medical robots is an emerging area of research. Extensive research is being done on the application of computers and robots for surgery, planning and execution of surgical operations and in training of surgeons. Surgeons are trained through apprenticeship. The basic techniques are taught with simple training equipment, but the rest of the training is done by using books describing surgical procedures and techniques and in the operating room by watching and participating in actual operations. Although operating room training is essential and invaluable, it does not provide the optimal environment to try or practice new techniques and procedures due to the risks to the patient. This method of training also limits the diffusion of knowledge since only a limited number of people can be trained by one experienced surgeon.

Virtual environments present an alternative to this training scheme. With virtual environments, it is possible to create an interactive 3D simulator, where the surgeons, using a haptic interface, can touch, manipulate or cut models of organs and tissues simulated on a computer. Virtual reality provides an environment where there is no risk to a patient and therefore less stressfull. They are interactive and 3D in contrast to books, and they are less expensive compared to training in the operating room using animals or cadavers. Virtual environments also give a unique advantage, as it is possible to generate arbitrary anatomies and pathologies with which the surgeons can be trained for cases that they will encounter only a few times during their whole career, but nevertheless must be trained for. This way, it is also possible to standardize the training and accreditation in surgery. Already, a large variety of medical skills are being treated by virtual reality environments; laparocsopic surgery (Figure 1), endoscopic exams, echography tests, etc.

Due to these advantages, multiple research centers have been created to focus on the developement of medical simulators [1][2][3][4][5][6][7][8]. Most of the existing medical simulators use only the anatomic geometry ignoring the use of linear viscous elastic properties of tissues. Furthermore, surgical training simulators require deformable models of different human organs. However, the state of the art for interactive deformable object simulation is not sufficiently advanced to build realistic real-time simulators. Some work on realistic tissue simulations can be found in [9][10][11][12][13][14].

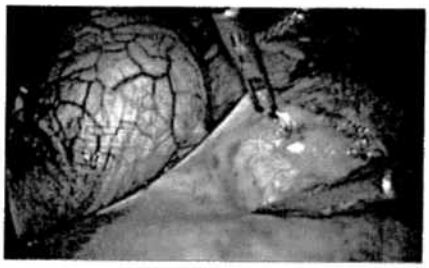


Fig. 1. Lapascopic surgery being performed on a human liver

Thus, the main challenge to achieve realism in medical simulators is to obtain deformable tissue models that are interactive, i.e. efficient enough to be simulated in real-time, visually and haptically realistic. These models are also required to be manipulated, cut and sutured. It is the aim of this paper to present our experince in the conception of a medical simulator. We begin by presenting two commonly used physical models before describing the construction of a new one in more detail. Next, to simulate forward integration in time, we present some numerical resolution methods and briefly touch on their complexity and stability problems. Following this, we lay out how we handle some aspects of real-time interaction that are inevitable in current medical simulators. We end this paper with some conclusions.

II. Physical Models for Soft Tissue

MODELING and simulation of deformable objects for real-time applications is indeed a non-trivial task. A survey of deformable modeling was done by Gibson and Mirtich [15]. They describe much of the work done until 1997. In brief, they divided the work done on deformable objects into two parts: non-

physically based models and physically based models. Physically based models can further be divided into dicsrete object models and other models based on continuum mechanics. The latter is usually solved using the finite element method (FEM). Recently, our research work has brought us to the conception of a physical model : Long Element Method (LEM). We believe that this model is particularly interesting for soft tissue simulation. The following sub-sections are devoted to these models, with special attention to the LEM model.

A. Mass-Spring Model

Principle The method of using mass-spring networks [16][17][14] consists of a mesh of point masses connected by using elastic links. The mass-spring network is mapped onto the geometric mesh, i.e. the masses are the vertices and the springs are the edges of the mesh. This mass-spring network is used to discretize the equations of motion. For example the link connecting pairs of nodes allows the local description of the elastic properties and consequently the displacement of the tissue. Any change of length relative to the resting length causes an internal force to occur between the two connected nodes. The simulation of the deformable object is done by modeling the effect of inertial and damping forces. Then, each node N in the mesh is subject to an equation of the form :

$$m\ddot{p} + b\dot{p} + \Sigma_i F_i = \sum F^{ext} \tag{1}$$

where p is the position of node N, $\dot{p}$ and $\ddot{p}$ are its velocity and acceleration, m is the mass attached to N, b is the damping coefficient to model viscosity, F_i is the internal force exerted by a node N_i to which N is connected by a link (the sum Σ_i is taken over all such nodes N_i), and F_{ext} is the total force applied to N, for example, it can be the force exerted by a surgical tool or gravity. The force F_i is the viscous-elastic response of the spring connectors and is given by :

$$F = (-\lambda \Delta d - \mu \dot{d})k \tag{2}$$

where λ is the rigidity factor of spring connector, μ is a damping factor, Δd is the relative variation of the distance between the two connected particles, $\dot{d}$ is the relative speed between these two particles and $k = (b-a)/(|b-a|)$ where a and b are the connected particles.

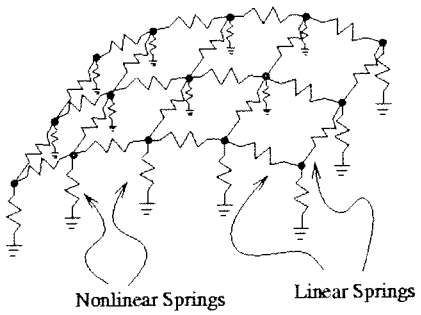


Fig. 2. Double layer non-linear mass-spring mesh of a human thigh

In Figure 2 we can see a double layer non-linear model of a human thigh and its mass-spring network representation. The vertices are the masses and the edges are the springs. We implemented this model in an echographic simulator [18], of a human thigh shown in Figure 9. The network is modeled by the following equations where Δx is the difference in the length of the springs with respect to their original resting length. This nonlinear spring response is chosen to model the incompressibility of the thigh after a certain deformation. The values of the surface elements are chosen uniform whereas the parameters of the nonlinear spring vary around the mesh to model the heterogeneous nature of the thigh.

$$\begin{aligned} F &= k\Delta x \quad \text{- Linear Springs} \\ F &= \Delta x/(m\Delta x + n) \quad \text{- Non-Linear Springs} \end{aligned} \tag{3}$$

Advantages & Limitations The mass-spring model is relatively easy to implement. The physics of the model is simple and well understood. These models can also be simulated in real-time without much difficulty [16][17], unlike other continuum models. Since interactions in this model are local between nodes, parallel computations are possible. Since we are concerned with deformable objects, the stiffness phenomenon rarely appears. It has been used as the physical model in numerous application (deformable objects), especially real-time applications, due to its rapidity. We have implemented a non-linear mass-spring network of a human liver in [19]. We find that this model generally conforms with reality and suitable for real-time interactions.

On the other hand, mass-spring models, as we have seen in the preceding sections, only takes into account two particles. When representing a volume using these binary connectors, the model can lead to several problems. Mass-spring connectors by itself has no volume. Of course, more springs will improve connectivity and thus produce a better approximation of the volume. Thus a volumetric object could perhaps be accurately modeled by infinite amount of particles and springs, but this is clearly not an option computationally speaking. For example, given a cube where the total of its mass is distributed on its eight corners and springs placed along the edges, it is impossible to enforce conservation of the volume. To remedy this problem, it has been proposed to add cross springs, thereby connecting opposing corners. However, this implies that the physical behavior of the object is intrinsically dependent on the connectivity of the springs. When aiming for physical realism this is clearly a handicap. Alternatively [20] proposed the use of angular and torsion springs, but again physical behavior is dependant on the topology and choice of the spring's parameters remains a black art.

B. Finite Element Method (FEM)

Principle Since the mass-spring model, which is a discrete model, suffers from certain drawbacks for certain applications, we decided to use a continuum model. The full continuum model of a deformable body gives the equilibrium of a body when subjected to external forces at any time. The model does this by minimizing the total potential energy of the system. In such a model, object deformation is described as material displacement. When the derivative of the potential energy of a body with respect to material displcement is zero, the continuous differential equilibrium criterion is satisfied.

FEM divides the object into a set of volumic elements and approximates the continuous equilibrium equation over each element. In our medical simulator, we chose to represent an elementary volume by a tetrahedron, that is a set of four points in three-dimensional space. We may express the deformation of this volume with respect to its original shape, using the Green-Lagrange tensor [21], which has the nice property of being invariant to rotation or translation. Given a position a of a point in the undeformed volume and let x be the position of the same point in the deformed configuration, then the deformation tensor is :

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial x}{\partial a_i} \cdot \frac{\partial x}{\partial a_j}\right) - \delta_{ij} \tag{4}$$

where δ_{ij} is :

$$\delta_{ij} = \begin{cases} 1 : i = j \\ 0 : i \neq j \end{cases} \tag{5}$$

Since the internal stress of the volume is proportional to the deformation (or strain) we may calculate the forces on the particles if the stress-strain relationship is known. For Hookean (linearly elastic) and isotropic (identical elasticity along all axes) materials this relationship can be condensed into two parameters, also known as the Lamé coefficients.

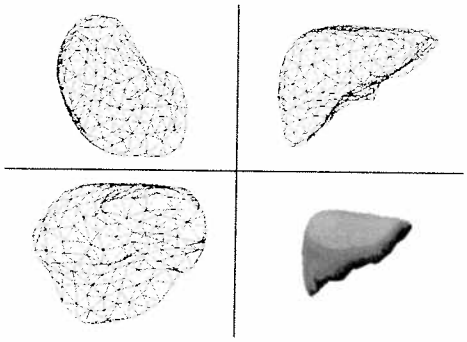


Fig. 3. Volumetric Mesh for a finite element model of a human liver

Once the forces on the particles are known, the following second order differential equation can be solved using a numerical integrator for the node displacements where M, C, and K are the mass, damping and stiffness matrices respectively for the deformable body. F is the externally applied forces matrix and U is the displacement vector matrix. In Figure 3 we can see a volumtric tissue of a human liver and its finite elements representation using tetrahedras.

$$M\ddot{U} + C\dot{U} + KU = \sum F^{ext} \tag{6}$$

Advantages & Limitations In contrast to the mass-spring model, FEM has a solid physical and mathematical foundation. The PDE's approximates exactly the deformation of an elementary element. Thus, FEM produces a more realistic physical simulation than mass-spring models with fewer nodal points. If the problem is of type static linear elasticity and linear geometry, the computational process of finding a solution amounts to solving a set of linear system of type $KU = F^{ext}$, where K has size $3N \times 3N$.

However, this model also has some drawbacks. The computational time spent on calculating the force of a nodal point is significantly longer when compared to a mass-spring model. Hence this is not the optimal choice for real-time applications. If the topology of the object changes during the simulation, or if there is a large deformation, the mass and stiffness matrix must be re-evaluated during the simulation. The choice of the deformation tensor is arbitary and generally depends on the application intended for. Also, FEM was intended for linear systems: non-linear systems can be simulated but at the loss of accuracy. Thus large deformations are generally not allowed.

C. The LEM Model (LEM)

Principle We now present LEM [22][23], a new physical model suitable for soft tissue simulation. The two basic principle that describe this model is Pascal's Principle and the conservation of volume. Unlike the previous two models, we use bulk variables such as pressure, density, volume, stress and strain to represent our object.

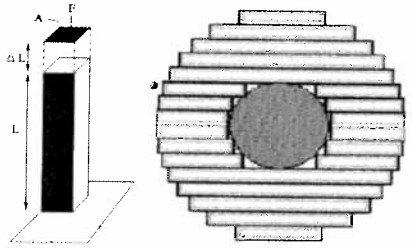


Fig. 4. A Long Element(right) and Modeling of non-homogenous elements(left) : soft tissue with rigid nucleus in the center

Consider the long element shown in Figure 4. The force per unit area is defined as pressure.

$$P = F/A \tag{7}$$

However, deformation produced by this pressure induces stress. For small applied forces, the stress ε, in a material is usually related to its deformation (i.e. elongation in our long elastic element). By introducing elasticity E, we get :

$$\varepsilon = E\Delta L/L \tag{8}$$

Force related to elongation in the well known form is :

$$F = K\Delta L \tag{9}$$

where $K = AE/L$ (K is dependent on L). The static solution produced by our model requires that the external pressure P_{ext}, equal the internal pressure P_{int}. The external pressure on the surface is affected by the atmospheric pressure P_{atm}, and by the stress due to elongation :

$$P_{ext} = P_{atm} + E\Delta L/L \tag{10}$$

The surface tension and environmental contact also contribute to the external pressure and we will integrate these factors into our model later. If the object is filled by fluid of density ρ, the internal pressure at a distance δ (from the upper part of the fluid) is due to pressure of fluid P_{fluid} and the effect of gravity g :

$$P_{int}|_{\delta} = P_{fluid} + \rho g\delta \tag{11}$$

Applying these equations to the group of N long elements and applying Pascal's Principle which gives constant ΔP, we then obtain the following continuous equation :

$$E_i \Delta L_i / L_i - \Delta P = d_i g h_i \tag{12}$$

where $\Delta P = P_{fluid} - P_{atm}$ and $i = 1..N$. Since the fluid is incompressible, we can add another constraint to our set of continuous equations. By using conservation of volume, we get :

$$\sum A_i \Delta L_i = 0 \tag{13}$$

where $i = 1..N$. Finally, we add the surface tension P_{st}. These terms are of the form $P = kx/A$ where x is the difference between the deformation of an element and its neighbors and k is a local spring constant :

$$P_{st} = \sum k_i^j (\Delta L_i - \Delta L_j)/A_i \tag{14}$$

for each neighboring j of element i. We now have $N+1$ equations and $N+1$ unknowns : ΔL from $1..N$ and ΔP. These $N+1$ equations can be written as a problem of type $Ax = B$ and solved using standard numerical methods :

$$E_i \Delta L_i / L_i + P_{st} - \Delta P = d_i g h_i \tag{15}$$

Advantages & Limitations Unlike the previous models, LEM uses bulk variables such as pressure, density and volume to model the object. These parameters are relatively more easy to be identified as compared to a mass of an element. Furthermore, the concept of an object being filled by an uncompressible fluid seems appropriate for soft tissue. By discretizing the object into columns of fluid, we get a system with much less variables leading to fewer equations and resulting in faster computation. These colums can also be used to represent non-homogeneous materials or even composite materials. No pre-calculation or condensation is required in the implementation of this model. The satic solution produced by this model is sufficient for soft tissue simulation as these kind of materials are known to be well-damped. The complexity of this model is of $\mathcal{O}(3N^2)$ which is generally one order of magnitude less than the finite element method. The system is numerically solved by using any standard numerical method.

Nevertheless, there are some limitations in this model aslo. LEM is only valid for small deformations since the matrix A and its inverse A^{-1} changes for large defromations. Hence, it needs to be re-evaluated in the case of large deformations.

D. Construction of Models

The construction of these models can be divided into two phases : geometrical and physical. These phases are explained below :

Geometrical This phase deals with the 3D reconstruction. Typically, the data set is obtained form slices of MRI medical images. Several database of images exist; one of them being the *Visible Human Project* [1]. We obtain our surface mesh in our simulator from project *EPIDAURE* at INRIA Sophia-Antipolis[2] A volumic mesh can then be generated accordingly. We use *GHS-3D*, a software by INRIA to obtain a volumic triangulated mesh. For the LEM model, the surface mesh is treated by our method presented in [23] to derive the columns of long elements filled with fluid. A liver modeled using LEM is shown in Figure 5.

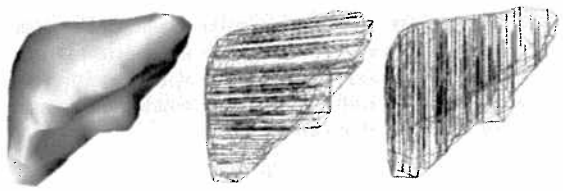


Fig. 5. A human liver discretized into long elements(blue lines)

Physical In this phase, we map a physical model onto the geometrical model. For the mass-spring model, masses and springs are mapped onto vertices and edges respectively. In the FEM model, we use tetrahedras as an elementary volume and lump the mass at the nodes of the tetrahedra. What remains to be done is to find the respective physical parameters. For the human liver, we took the physical parameters from [24]. The parameters for the human thigh was obtained using a robot as shown in Figure 9.

III. Numerical Resolution

REAL-TIME interaction is inevitable in a domain such as surgical training. A system which is computationally expensive is generally not acceptable. For any physical model, integrating it forward in time can be most difficult in terms of time and stability. Generally, system of particles/elements is represented as :

$$\begin{aligned} M\ddot{x} + C\dot{x} + Kx &= \sum F^{ext} \\ \dot{v} &= M^{-1}(-Cv - Kx + \sum F^{ext}) \\ \dot{x} &= v \end{aligned} \tag{16}$$

where M, C and K are the $3n \times 3n$ mass, damping and stiffness matrices. These matrices are typically quite sparse, M anc C being diagonal or banded depending on the representing function. K is generally banded. F^{ext} is the total external forces. An analytical solution is not possible for such a system because of its complexity. So we have to search for a numerical one. This system of equations can give a dynamic resolution or a static resolution depending on the formulation.

A. Dynamic Resolution

Dynamic resolution can be obtained using *explicit* or *implicit* integration.

Explicit Integration Once the force acting on a particle is known it becomes possible to evaluate its change in velocity and position. The simplest way to do so is using the Newton-Euler integration :

$$\begin{aligned} {v_i}^{t+\Delta t} &= {v_i}^t + \Delta t {a_i}^t \\ {x_i}^{t+\Delta t} &= {x_i}^t + \Delta t {v_i}^t \end{aligned} \tag{17}$$

However, if the object we are simulating is not *highly* elastic the resulting ordinary differential equations are stiff resulting in poor stability and requiring the numerical integrator to take very small time steps.

[1] http://www.nlm.nih.gov/research/visible/visible_human.html

[2] http://www-sop.inria.fr

Implicit Integration To get around the problem of stiff springs [25] proposed to use implicit integration. Then velocity and position is obtained as follows :

$$\begin{aligned} {v_i}^{t+\Delta t} &= {v_i}^t + \Delta t {a_i}^{t+\Delta t} \\ {x_i}^{t+\Delta t} &= {x_i}^t + \Delta t {v_i}^{t+\Delta t} \end{aligned} \tag{18}$$

where if we let y be any required state and $f(y)$ the derivative, then we have the following for a general linear system :

$$\begin{gathered} f(y^t) = \lambda y^t \\ y^{t+\Delta t} = y^t + \Delta t f(y^{t+\Delta t}) = y^t + \Delta t \lambda y^{t+\Delta t} \\ y^{t+\Delta t} = \frac{y^t}{1-\lambda \Delta t} \end{gathered} \tag{19}$$

where λ is a constant. For linear differential equations, the iteration cannot diverge. When applied to non-linear differential equations, there is a possibility of divergence. Using this method, the update of a state over a time step requires solving a linear system the size of the number of degrees of freedom of all particles. Since the system is sparse a conjugate gradient can be used for solving.

B. Static Resolution

If the dynamics of our system are not essential like for soft tissue applications, we may decrease the complexity by neglecting the velocity. Since at equilibrium the external and internal forces are perfectly balanced we may use the principle of virtual work. Hence we find the displacements u using the tangent stiffness matrix K for a given external force. Recall that a similar expression is obtained when LEM is used based on equilibrium of internal and external pressure. Hence, the same resolution method can be applied.

$$Ku = f_{ext} \tag{20}$$

However, this is only valid for *small* displacements, since the stiffness matrix changes for larger displacements, and the solution of the system is non-linear beyond that. We use a Newton-Raphson iterative scheme for this purpose; initially the displacements are calculated as in the linear case, but the residual internal force is reevaluated at the new configuration. The system is then solved again and again until convergence. This, of course, implies evaluating the stiffness matrix at each iteration. In the modified Newton-Raphson scheme the tangent stiffness matrix in *not* updated at each iteration, normally at the expense of an increased number of iteration. In our experience it is, however, a computationally interesting alternative.

Both the modified and the full Newton-Raphson schemes demand the solution of a linear system at each iteration. Due to the large size of the matrix for non-trivial objects a direct solution is not computationally feasible. We choose to solve the system iteratively using successive over-relaxation (SOR) due to its simple implementation and potential for parallelization.

Summarizing, we have two nested iterative schemes to solve the non-linear system. We use SOR to solve the linear system at each Newton-Raphson (full or modified) iteration in order to minimize the residual force. It is important to note that the Newton-Raphson iteration may experience convergence problems. In this case the correction to the incremental displacement vector should be weighted by a value < 1.

Fig. 6. A 'pull' and 'push' deformation by static resolution

IV. Real-Time Interactions

ULTIMATLY , the goal of a medical simulator is to allow real-time interactions and conform with reality. It is clear that the ability of a trainee surgeon to successfully grasp the skills of an operating procedure depends on the *learning* environment. It is our aim to provide the trainee the maximum possible. As fas as a simulator is concerned, it is well known that during a simulation, given *any* physical model, the most difficult aspects, in terms of computational time or updating data structures are collision detection, the different rates for haptic interaction between graphical updates and physical simulation and topology modification during specific surgical procedures. We address these problems in the following sections.

A. Collision Detection

Deformable bodies may become concave during deformation. Many algorithms obtained from collision detection of convex objects [26] can be used by dividing concave polyhedra into several set of convex polyhedras. To avoid this costly time computations, Baraff and Witkin [27] divided objects into convex sub-objects that can only obey first order deformation (a facet or an edge can not be curved), which guarantees that they stay convex during simulation. In some litterature, Open-GL hardware acceleration has been suggested [28]. We avoid this since its collision detection technique is highly depending of the configuration of the system, and performance can be lost while changing to another PC-class/MS Windows graphics environment [29]. These and other methods only give us the intersecting elements or primitives.

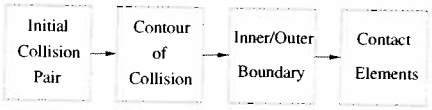


Fig. 7. Algorithm to localize all contact elements

However, computing the collision response requires us to try to evaluate the involved local deformation of the colliding objects; using a non-linear penalty method [20]. This can be done by determining the *fictitious interpenetration* of the objects. This means that the collision detection algorithm require not only the intersecting elements but also those elements that are interior to each other. Thus, we based our collision detection method on the bounding volume technique. We use Axis-Aligned Bounding Boxes (AABB) to bound our facets. By using a binary tree to represent the Bounding Box hierarchy, we apply the algorithm we have presented in [30] to localize all the interior facets.

This alogorithm requires only an initial colliding pair of facet. Following this, a collision contour is constructed whereby allowing us to decide the facets which are interior and exterior to the volume of collision. A simple recursive search will localize all interior elements. Figure 7 shows the these stages. The complexity of this algorithm is $\mathcal{O}(\mathcal{N})$, where N is the number of intersecting pairs. Figure 8 shows an example of our collision detection implementation.

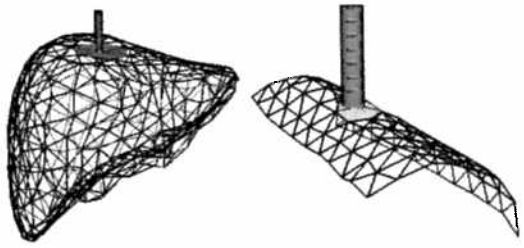


Fig. 8. Exact contact determination : liver(left) and thigh(right)

B. Haptic Interaction

Haptic systems gives people the sensation of touching objects in virtual environments. Including haptic technology improves the perception of a surgeon thus producing a deeper sense of immersion. Multiple problems arise in haptic applications interacting with deformable objects, for example, costly computational time, numerical instability in the integration of the body dynamics, time delays, etc. Lenghty computations are forbidden in haptic systems which need high simulation rates (about 1KHz) to obtain realistic force feedback. The update rates of the physical objects being simulated is normally of the order of 20 to 150 Hz. This difference in simulation rates can cause an oscillatory behaviour in the haptic device that can become highly unstable and inflict harm on the operator [31]. Several numerical approaches [32][33] have been proposed to solve the difference rate problem. However, due to the unknown nature of human behavior, there will always exists a small error that cause slight vibrations.

In our approach, instead of interacting with the complete physical model we use a simple intermediate representation : local models, as in [18][34][35][36] [37] which allows to compute the force feedback at the haptic required frequency. The force F, is calculated using Hooke's law and the minimal distance d, between the local model and the haptic tool :

$$F_{haptic} = \begin{cases} k\,(f_{model})\,d & \text{if } d < 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where the local stiffness k, depends on the physical model force f_{model} to give a more realistic sense of touch. To avoid small vibrations due to sudden changes in the values of f_{model} (updated at the simulation frequency), the haptic update procedure is done gradually in each haptic iteration until the new value is reached.

The computation of the minimal distance d, depends on the intermediate representation used. We have two main approaches which are explained below.

Analytic Based This representation is oriented to non-highly deformable objects [18] where d is computed by obtaining the distance to a sphere or to a single plane. To update them we compute its configuration parameters $u = [u_1 u_2 \dots]$ (e.g. the center and radius of a sphere) as follows :

Let d_{model} be the distance between the position of the virtual probe $x_{virtual}$, and the complete object. Let d be $\phi(x, u)$, where $x = [x_1 x_2 \dots x_n]$ are the variables that correspond to the configuration space dimensions. The goal is to find u_i that keeps $d_{model} = \phi(x, u)$. The algorithm is based in the first order approximations of Equation 22.

$$\begin{aligned} \Delta\phi(x,u) &\simeq [\partial\phi(x,u)/\partial u]\Delta u \\ \Delta\dot{\phi}(x,u) &\simeq [\partial\dot{\phi}(x,u)/\partial u]\Delta u \end{aligned} \tag{22}$$

If we let the Jacobian matrix to be J as below then we have the following :

$$J = \begin{bmatrix} \partial\phi(x,u)/\partial u \\ \partial\dot{\phi}(x,u)/\partial u \end{bmatrix} \text{ and } J\Delta u = \begin{bmatrix} \Delta\phi(x,u) \\ \Delta\dot{\phi}(x,u) \end{bmatrix} \tag{23}$$

Thus, the required parameters Δu are obtained using the pseudo-inverse of J ($J^{\dagger} = (J^T J)^{-1} J^T$) as shown in Equation 24.

$$\Delta u = J^{\dagger} \begin{bmatrix} \Delta\phi(x,u) \\ \Delta\dot{\phi}(x,u) \end{bmatrix} \tag{24}$$

This approach has been implemented in an echographic simulator of a human thigh shown in Figure 9. The data adquisition is done using a PUMA robot arm, as seen in Figure 9-a. We use a a two layer mass-spring model, as explained in Section . The physical parameters are adjusted using a non-linear least square estimation [10] and implicit integration is applied to solved the dynamics of the system. Due to the relatively small deformations of the thigh, planes or spheres are likely to be used as the intermediate representation, see Figure 9-b. Ecographic images are produced by the interaction between the deformable model and a virtual tool as shown in Figure 9-c. The virtual tool is a representation of the haptic device (PHANToM type) which sends the force feedback to the user.

Local Topology Based For highly deformable objects we propose a local model [35] based on the topology of the object. The computation of d is more complicated than the previous approach, however updating the intermediate representation is easier since it is constructed by using the colliding facets and its neighboring facets as a set of planes.

The minimal negative distance is calculated between haptic position, x_{haptic} and a point x_p, called proxy [36] lying on the surface of the planes. The proxy is constrained to never have a negative distance with respect to the other planes. Generally, the minimal negative distance problem is be written as :

$$\begin{aligned} &\min d = \| x_p - x_{haptic} \| \\ &\text{such that } P_i = 0 \end{aligned} \tag{25}$$

The expression $P_i = 0$ is the set of equations of the different planes that form the local surface where i is the number of planes which can be reduced to 3 using [37]. Lagrange multipliers λ, are used to solve the minimization problem [36] [37]. An ilustrating example of this solution to obtain x_p is shown in [35] where:

$$x_p = x_{haptic} - N^T \lambda \tag{26}$$

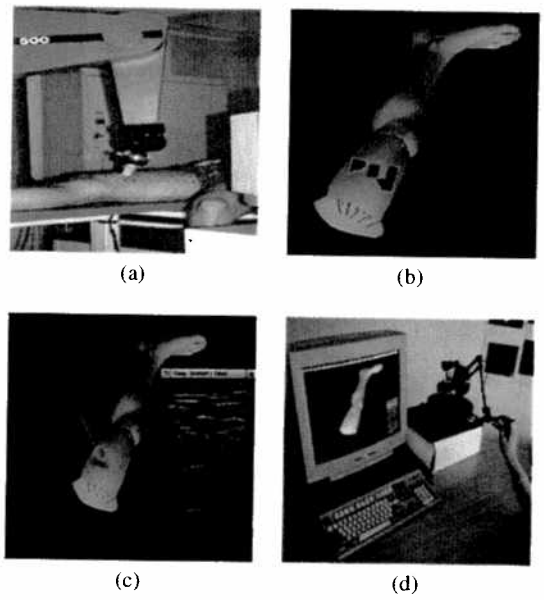


Fig. 9. (a) Data adquisition (b) Local Model : Plane (suitable for objects with small deformations) (c) Real time deformations with ecographic images (d) Haptic Interaction

where $\lambda = [N^T]^{-1}\ (x_{haptic} - N^{-1}\ D)$, $N = [n_1\ n_2\ n_3]^T$ is the combined set of normals belonging to each plane and $D = -[p_1 n_1\ p_2 n_2\ p_3 n_3]^T$ and p_i the baricentre point of the facet corresponding to plane i.

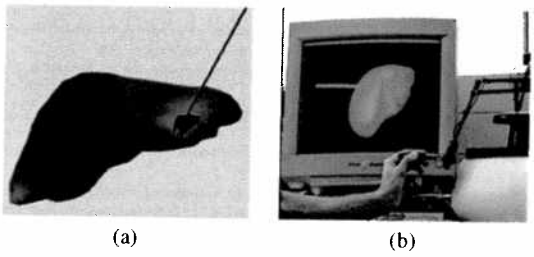


Fig. 10. Haptic interaction with the aid of a local model : (a) The topological local model and (b) Haptic interaction with a PHANToM device

This approach has been implemented for haptic interaction between different human organs like the liver or the vesicule which have a high degree of concativity meaning high deformability. The topological intermediate representation is shown in Figure 10-b where the virtual tool (PHANToM device) touches a human liver. For the dynamic simulation of the liver we have implemented and experimented the three types of physical models described in Section II. Notice that the intermediate representation takes the local shape of the liver.

C. Topology Modifications

One important feature in virtual reality medical simulators is the *cutting* phenomena. Cutting can be implemented either in mass-spring models [38][39][40] or finite element models [21]. In both implementations, cutting is carried out by simply removing an element or dividing it.

In our previous work [41] we have introduced a methodology to simulate real-time tearing phenomena. This approach is based on an interesting *separation* of the involved physical elements of the model instead of destroying or spliting them. If the elements (e.g. facets, tetrahedrons, etc) are destroyed the complexity of the simulation is decreased but the discretization of the mesh needs to be larger in order to maintain realism. However, this leads to a drawback in real-time interactivity. The subdivision approach increases the number of elements, making the simulation slower at each cutting operation. Recently we have extended this approach to the cutting phenomena using a mass-spring mesh. This works well for surface meshes (skin simulation cutting) and can be implemented for volumetric meshes. In [42] some research is presented using finite element models but it is still computationally expensive for real time interaction.

We consider the virtual cutting phenomenon as an interaction between a deformable virtual object and a rigid body or tool. In this process the deformable model is cut and its geometrical topology changes. Three important criteria are considered : (1) preserving the main physical properties, (2) obtaining realistic visual effects and (3) satisfying real-time constraints. Then, rather than dividing the facets of the initial mesh into smaller ones, we *separate* them. Separating the facets provides us a way to avoid matter from disappearing and to avoid an important growth in the complexity of the model. In Figure 11-b, a 2D mesh is cut using our algorithm. This becomes important while simulating skin cutting which is an important feature towards a realistic laparoscopic simulator. The steps to model the cutting phenomena at time t once a collision between the rigid body (virtual tool) and the deformable object has been detected is presented below.

Step 1 : Assigning cutting elements Let $S(t)$ be the closest vertex to the contact point $C_p(t)$. Then a cut is carried out between $S(t-1)$ and $S(t)$ if the following conditions are verified :

- $C_p(t)$ and $C_p(t-1)$ belong to different facets.
- $S(t-1) \neq S(t)$

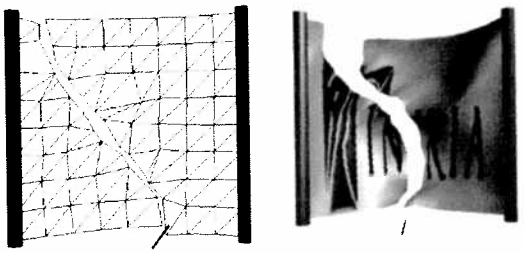
Step 2 : Finding the cutting path In [19] we propose an algorithm to determine the edges that have to be split between vertices $S(t-1)$ and $S(t)$. Then, vertex $S(t-1)$ is placed in the position of $C_p(t-1)$ and vertex $S(t)$ in the position of $C_p(t)$. Finally, the vertices belonging to the cutting path are orthogonally projected to the line between $S(t-1)$ and $S(t)$ as seen in Figure 11-a.

Step 3 : Remeshing along the cutting path We take the following steps to remesh along the cutting path between vertices $S(t-1)$ and $S(t)$:

- A new particle is created, at the same position as the split one, with the same velocity, and half of its mass.
- The rest-length of the broken spring is updated to be its actual length.
- A new spring is created, with the same elasticity and viscosity as the broken one.

V. Conclusion

This paper has presented our experience towards the conception of a realistic virtual medical simulator. We have touched areas on physical models, numerical resolution techniques and real-time interactions.



(a) (b)

Fig. 11. (a) Global idea of the cutting process (b) Cutting a virtual 2D skin mesh using a haptic device (type PHANToM) to manipulate the virtual tool

The physical models presented covers discrete object models to full continumm mechanics models. These models have been used in various application and they have been highlighted. We have also described LEM in detail which has complexity one order of magnitude less than FEM. We have also proposed the use of a static resolution instead of a dynamic one. Very essential real-time interactions have been presented together with some results. We use bounding volume techniques for collision detection which we believe to be suitable for deformable objects like virtual organs in medical simulators. We have also presented a methodology to solve the unstability problem of the update rate difference between the haptic and the physical simulation. The virtual reality 2D cutting phenomena has been described. We believe that the ideas can be exported to a 3D cases. Other real-time topology modifications will be implemented as well, e.g. suture operation.

To end the paper, we compare the three physical models presented in this paper aginst some important criterias.

	Mass-Spring	FEM	LEM
Rapidity	★★	★	★★★
Realism	★	★★★	★★
Implementation	★★★	★	★★
Soft Tissue Model	★	★★	★★★
Updating Topology	★★	★	★★★

VI. Acknowledgements

This work is within the framework of project CAESARE by the French National Institute for research in Computer Science and Automation. It has also been supported by the National Council of Science and Technology of Mexico (CONACYT). We would like to thank I. F. Costa and R. Balaniuk for their contribution. K. Sundaraj would like to acknowledge the financial grant from the French Ambassy for his doctoral thesis.

References

[1] "Virtual environments for surgical training and augmentation, (vesta)." Tech. Rep., Berkeley-UC. http://robotics.eecs.berkeley.edu/ mdownes/surgery/surgstm.html.

[2] "Center for medical robotics and computer assisted surgery." Tech. Rep., Carnegie, Mellon University. http://www.mrcas.ri.cmu.edu.

[3] "Image guided therapy program." Tech. Rep., Harvard. http://splweb.bwh.harvard.edu:8000.

[4] "Nsf: Enginnering research center for computer-integrated surgical systems and technology." Tech. Rep., John Hopkins University. http://cisstweb.cs.jhu.edu.

[5] "Image guided surgery." Tech. Rep., MIT. http://www.ai.mit.edu/projects/medical-vision/index.html.

[6] "National center for biocomputation." Tech. Rep., Stanford/NASA. http://www-biocomp.stanford.edu.

[7] "Center for advance technology in surgery at stanford." Tech. Rep., Stanford Medical School. http://catss.stanford.edu.

[8] "Gestes médico-chirurgicaux assistés par ordinateur (gmcao)." Tech. Rep., TIMC-CHU Grenoble. http://www-timc.imag.fr/gmcao.

[9] H Delingette. "Rapport de recherche, no.3506, towards realistic soft tissue modeling in medical simulation." Tech. Rep., INRIA, Sophia-Antipolis, Sept. 1998.

[10] D. d'Aulignac, C. Laugier, and M.C. Cavusoglu, "Modeling the dynamics of a human thigh for a realistic echographic simulator with force feed-back," in *Medical Image Computing and Computer Assisted Intervention - MICCAI'99 Proceedings*, 1999, pp. 1191–1198.

[11] Stéphane Cotin, *Modèles anatomiques déformables en temps-réel*, Ph.D. thesis, Epidaure-Sophia Antipolis, 1997.

[12] 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 1999.

[13] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," in *Proc. Eurographics'96*, 1996.

[14] S. Gibson, "3d chainmail: A fast algorithm for deforming volumetric objects," in *Proc. Symp. on Interactive 3D Graphics, ACM SIGGRAPH*, 1997, pp. 149–154.

[15] F. Gibson and B. Mirtich, "A survey of deformable models in computer graphics," Tech. Rep., MERL, Cambridge, MA. http://www.merl.com/reports/TR97-19/index.html, 1997.

[16] F. Boux de Casson and C. Laugier, "Modelling the dynamics of a human liver for a minimally invasive simulator," in *Medical Image Computing & Computer Assisted Intervention-MICCAI'99 Proceedings*.

[17] A. Delingette, G. Subsol, and S. Cotin, "A craniofacial surgery simulation testbed," in *Proc. Visualization in Biomedical Computing (VBC)*, Oct. 1994.

[18] D. d'Aulignac, R. Balaniuk, and C. Laugier, "A haptic interface for a virtual exam of the human thigh," in *Int. Conf. on Robotics and Automation, San Francisco, USA*, Apr. 2000.

[19] F. Boux-de Casson, *Simulation Dynamique de Corps Biologiques et Changements de Topologie Interactifs*, Ph.D. thesis, INRIA Rhone-Alpes and Universite de Savoie, 2000.

[20] A. Deguet, A. Joukhadar, and C. Laugier, "Models and algorithms for the collision of rigid and deformable bodies," in *Robotics: the algorithm perspective, Proc. of the Workshop on the Algorithmic Foundations of Robotics, Houston, USA*, Mar. 1998.

[21] James O'Brien and Jessica Hodgins, "Graphical models and animation of brittle fracture," in *SIGGRAPH 99 Conference Proceedings*.

[22] I. Costa and R. Balaniuk, "Static solution for real time deformable objects with fluid inside," in *ERCIM News*, Jan. 2001, pp. 44–45.

[23] K. Sundaraj, C. Laugier, and Costa I. F., "An approach to lem modelling : Construction, collision detection and dynamic simulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001.

[24] P. Davies, F. Carter, D. Roxburgh, and A. Cuschieri, "Mathematical modelling for keyhole surgery simulations: spleen capsule as an elastic membrane," in *Journal of Theoretical Medicine*, 1999.

[25] D. Baraff and A. Witkin, "Large steps in cloth simulation," in *Computer Graphics (Proc. SIGGRAPH)*, 1998, pp. 43–54.

[26] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between objects in 3d space," in *IEEE Int. Conf. on Robotics and Automation*, 1988.

[27] D. Baraff and A. Witkin, "Dynamic simulation of non-penetrating flexible bodies," in *Computer Graphics*, July 1992.

[28] J. Lombardo, M. Cani, and Neyret F., "Real-time collision detection for virtual surgery," in *Computer Animation, Switzerland*, May 1999.

[29] "High performance graphics hardware design requirements," Tech. Rep., Linas Vepstas. http://linas.org/linux/graphics.html.

[30] K. Sundaraj and C. Laugier, "Fast contact localisation of moving deformable polyhedras," in *Proc. of the Int. Conf. on Control, Automation, Robotics and Visinn,Singapore*, Dec. 2000.

[31] R. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," in *IEEE Transactions on Robotics & Automation 99*.

[32] E. Ellis, N. Sarkar, and M. Jenkins, "Numerical methods for the force reflection of contact," in *ASME Transactions of Dynamic Systems Measurement and Control*, 1997, pp. 768–774.

[33] M. Cavusoglu and F. Tendick, "Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments," in *Int. Conf. on Robotics and Automation '00*.

[34] W. Mark, S. Randolph, and Finch, "Adding force feedback to graphics systems: Issues and solutions," in *Computer Graphics Proc. SIGGRAPH, New Orleans, Lousiana*, Aug. 1996.

[35] C. Mendoza and C. Laugier, "Realistic haptic rendering for highly deformable virtual objects," in *IEEE Virtual Reality VR 2001, Yokohama JAPAN*, Mar. 2001.

[36] D. Ruspini, K. Kolarov, and Khatib O., "Haptic interaction in virtual environments," in *Proceedings of Int. Conf. on Intelligent Robots and Systems 1997, Grenoble-FR*.

[37] C. Zilles and K. Salisbury, "A constraint-based god-object method for haptic display," in *ASME, Haptic Interfaces for Virtual Environment and Teleoperator Systems, Dynamics Systems and Control, Chicago, Illinois*, Nov. 1994, vol. 1, pp. 146–150.

[38] D. Terzopoulos and Fleischer K., "Modeling inelastic deformation: Viscoelasticity, plasticity, fracture," in *Computer Graphics, Proceedings of SIGGRAPH'88, Atlantu, Georgia*.

[39] A. Norton, G. Turk, B. Bacon, J. Gerth, and P. Sweeney, "Animation of fracture by physical modeling," in *The Visual Computer*, 1991, vol. 7, pp. 210–217.

[40] D. Bielser, V. A. Maiwald, and M. H. Gross, "Interactive cuts through 3-dimentional soft tissue," in *EUROGRAPHICS'99*, 1999, vol. 18, pp. C–31–C38.

[41] F. Boux de Casson and C. Laugier, "Simulating 2d tearing phenomena for interactive medical surgery simulators," in *Computer Animation 2000, Pensilvania, USA*, May 2000.

[42] H. Nienhuys and F. Vanderstappen, "Combining finite element deformation with cutting for surgery simulations," in *EUROGRAPHICS 2000*, 2000.