An interactive model of the human liver

F. Boux de Casson D. d'Aulignac
C. Laugier
GRAVIR/INRIA Rhône Alpes
38330 Montbonnot Saint-Martin, France
{Francois.Boux-de-Casson, Diego.d_Aulignac, Christian.Laugier}@inrialpes.fr

1. Introduction

Objectives With increasingly complex surgical procedures doctors need to learn new skills. However, to practice during a real intervention may be dangerous for the patient. It is our long-term goal to build a surgical simulator for hepatic procedures to be used in medical training. A cornerstone of such an endeavor is the underlying physical model of the liver. Very little strain/stress experimental data on the liver is currently available, and thus we want our model to be as realistic as possible in a qualitative way. Finally, since the simulation is to be used interactively real-time performance is of paramount importance.

Relevant mechanical characteristics of a human liver The liver is a very malleable body, its exact shape strongly depends on the contact interactions with the other organs located in its vicinity. It is composed of tree major parts :

1. the Parenchyma, which presents a mechanical behavior near to those of a sponge full of liquid;

2. a complex vascular network, irrigating the liver;

3. an elastic skin called Capsule of Glisson, which is quite elastic and stiffer than the Parenchyma.

To simplify the problem, we do not modelise the vascular network behavior. So, intuitively, one could coarsely represent the liver by a sponge filled with a liquid and covered by an elastic skin.

2. Outline of the Model

Because the above mentioned anatomic and biomechanical properties, a heterogeneous model is required for modeling the mechanical behavior of the liver. In the sequel, we show how we have modeled the liver using two main components : a 2D component for modeling the Capsule of Glisson and a 3D one for modeling the Parenchyma. Each of this models include a geometrical and a physical component.

2.1. Geometrical Component of the Model

The geometric component of the model is used for performing the display operations and for detecting the interactions. This model is also used as a spatial frame for constructing the physical component.

We have chosen to make use of 2D mesh of triangles (see fig.1), for representing the Capsule of Glisson, and a tetrahedric mesh for the Parenchyma. The triangular facets of the skin correspond to the external faces of the tetrahedra of the internal 3D mesh.

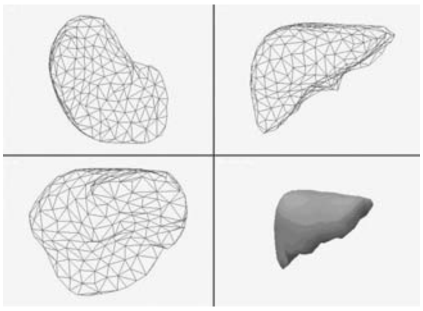

Figure 1. A 2D mesh, representing the Capsule of Glisson of the liver (the associated 3D mesh representing the Parenchyma is not shown in this picture).

2.2. Physical Component of the Model

The physical component of the model is used to compute the deformations of the liver resulting from the application of a set of external forces. These forces are either applied by the operator using the virtual tool (controlled using a haptic interface), or generated by the physical interactions with other virtual objects of the scene.

The physical component is constructed from the 2D and 3D meshes (geometrical component of the model), by associating point masses to the nodes of the previous meshes and by adding spring-damper connectors between appropriate local subsets of point masses (see [BdCL]). In order to model the non-linear mechanical behavior of the liver, without increasing the algorithmic complexity of the approach, we have chosen to associate the following behavior to each spring :

$$\vec{F}_{spring} = \left(\lambda_1 d^3 + \lambda_2 d\right) + \mu \dot{p} \tag{1}$$

where d is the relative deformation of the spring, that is to say $d = \frac{l-l_0}{l_0}$, with l is the length of the spring and l_0 its rest length, λ_1 and λ_2 are stiffness parameters, μ is a viscous parameter and $\dot{p}$ the relative velocity of the two particles connected by the spring. The choice of the first term of this law was directed by the fact that the few strain-stress data on the liver's mechanical behavior are showing linear behavior for small deformations (lower than 10%) and for bigger strain, the stress increases sharply. The sum of a linear and a cubic functions give a good approximation of this kind of behavior. The shape of the first term of this law is presented in the fig. 2. This non-linear relation make the springs to be relatively incompressible and unstrechable. The second term is a viscous term.

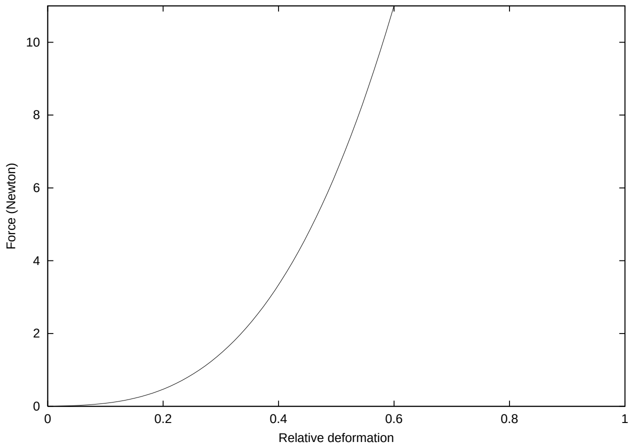

Figure 2. Plot of the springs force, as a function of their relative deformation, with $\lambda_1 = 50$ and $\lambda_2 = 0.35$.

In order to model the heterogeneousness of the liver, the connectors of the 2D mesh and of the 3D mesh are parameterized in such a way that they exhibit different mechanical characteristics :

- the Capsule of Glisson is modelised using high stiffness parameters (λ_1 and λ_2 of the Eq. 1) and low viscous parameter, allowing us to obtain a rigid elastic behavior which tend to bring back the Capsule of Glisson to its initial shape when no force is applied on it.
- for the Parenchyma, the stiffness parameters are tuned to be small in front of the viscous one, giving a plastic behavior (the inner material can easily be deformed, but does not go back quickly to its initial shape).

The combination of the two previous models (the elastic skin and the viscous volumetric internal material) gives us qualitatively and experimentally the required global behavior for the virtual liver (the behavior of a "sponge full of liquid covered by an elastic skin").

2.3. Integration of the Dynamic Equations

The integration method used in this simulator is the well-known explicit Newton-Euler. Using this method, point i of an object has the following update formulae.

$$\begin{aligned} {v_i}^{t+\Delta t} &= {v_i}^t + \Delta t {a_i}^t \\ {x_i}^{t+\Delta t} &= {x_i}^t + \Delta t {v_i}^t \end{aligned} \tag{2}$$

where Δt is the time step used and x_i, v_i and a_i are respectively the position, the speed and the acceleration of the particle. This integration method,

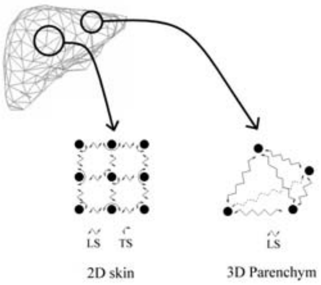

Figure 3. The hybrid mesh approach. A 2D and a 3D spring-damper meshes are used to model respectively the Capsule of Glisson and the Parenchyma.

known for it small time step problem when dealing with rigid objects, is good enough for our application, because we are dealing with relatively soft object. If object are stiffer, we can use implicit integration as demonstrated in [dLC99].

2.4. Collision Detection

Further, to detect the interaction between objects in a scene, a collision detection algorithm must be used. It is well known that collision detection is a computationally expensive task and a potential bottleneck in every application that aims to achieve real-time performance. For this reason it is essential to optimize these functions to their maximal extent. [JWLne] proposes an algorithm that obtains, in linear time, the points of contact on two deformable concave objects, as well as the direction of the contact and the volume of interpenetration. This approach has the advantage that it may be used for two or more objects of any shape, but depending on their number and complexity may not guarantee real time performance.

However, for the special case of collision detection between a rigid object which has the shape of a parallelepiped and a deformable object of any shape the approach [LCF99] may be used. It makes use of the OpenGL hardware to detect the polygons within a bounding box. If this bounding is superimposed onto the rigid object, the polygons in interaction will be detected. This approach has the advantage of being very fast due to hardware acceleration and is used in this application.

Processing physical interactions Once a collision has been detected an appropriate response must be computed. There exist several collision response models but for deformable objects the penalty method seems to be the most appropriate (see [DJL98] for more details). Note that the formulation presented is widely used in mechanics and has been verified physically. The force applied at a given point of an object where a collision has taken place is given by

$$\vec{F_c} = \begin{cases} (-\lambda v - \mu \dot{v} v)\vec{k} & \text{if } v < 0 \\ \vec{0} & \text{otherwise} \end{cases} \tag{3}$$

where λ is the rigidity factor of the collision, μ is a damping factor (which represents the dissipation of energy), v the volume of inter-penetration, and $\vec{k}$ the contact direction.

2.5. Haptic Interaction

As described in Section 2.4 the interaction force is calculated with respect to the interpenetration distance of two colliding bodies. This force is then used by the physical model to compute its update in state. Thus the rate at which force values can be supplied is limited by the execution time of collision detection and, thereafter, the update of the physical model.

Our aim is to provide force feedback through means of a haptic interface of type PHANToM[1]. These interfaces require a very high update rate for the force and typically this frequency is around 1KHz. However, the physical model is not able to provide force values at such rates.

Previously, to increase the frequency of the physical simulation [AH98] proposed a multi-resolution approach where only areas of haptic interest are simulated in detail. However, for complex objects this may not be sufficient and the resulting simulation still too slow. While [MRF+96] already suggested decoupling the haptic servo loop from the main application in the context of rigid bodies, we have extended this reasoning to deformable objects.

Our approach consists in making a first order approximation of the collision forces which can be calculated at a much higher frequency. The approximation is made through the use of a *local model* of the contact; in the case of a rigid tool interacting with a deformable object we make the (false) assumption that the surface of the deformable object remains in the same position between two updates of the physical model. The surface of the object that is in interaction with the tool is approximated by a simple geometric primitive such as a sphere or a plane. Thus the interpenetration distance between the tool and the object can be found at a much higher frequency (i.e. simpler and thus faster distance computation) and, therefore, the force values supplied at an increased rate. This, even though only gross approximations have been used, leads to much smoother haptic interaction (Figure 8) [dBL]. Thus we may divide this process into two categories:

- Model update Using the information about distance and the derivative of the distance, a simple *local model* of the objects surface is constructed. Thus this approximation is updated at the frequency of the physical model, and remains static in between those updates.

- Haptic loop Once the approximation of the surface is in place the distance between the haptic position and the surface is minimized analytically using Lagrangian multipliers. This distance multiplied by the stiffness constant yields the force values.

[1] see http://www.sensable.com

2.6. Changing the topology

An algorithm to tear the model has been implemented, it is presented in [BL00]. The global idea is to separate the elements of simulation (i.e. the tetrahedra) when they are stretched above a given threshold. This approach gives better results than when elements are removed, because the discretization is often coarse in real-time models. Thus when big elements are removed, one can see matter disappearing. Subdivision methods give more accurate results, but are not compatible with interactive constraint, because the number of elements to be simulated increases at each topology change, and the real-time constraint cannot always be assured.

Nevertheless, the limitation of our approach is that the topology changes are constrained by the initial topology of the model.

3. Implementation and Experimental Results

3.1. Architecture

The machine used for these tests was a biPentiumII 300, algorithms have been implemented in C++.

We have used as input a 2D mesh of the liver which has been pre-processed in order to reduce the number of facets, to smooth it, and to obtain the internal tetrahedric mesh (which has been calculated using the GHS3D[GB98] software of INRIA). The final 2D mesh we used is compound by 370 facets, and the 3D one by 1151 tetrahedra.

Then, we have used this 2D and 3D meshes to generate the spring-dampers network (see §2.2). For the purpose of the dynamic simulation, the model of the liver has been placed in an empty space, without gravity, but with a slight environmental viscosity. The four particles of a tetrahedron which is in the middle of the model are fixed. A virtual tool, simulated by a rigid object controlled in position by the operator, makes it possible to apply forces to the model of the liver, and to follow compliantly the external boundaries of the virtual liver.

3.2. Validation of the deformation model

To validate that the non-linear mechanical behavior of each connector leads to a global non-linear behavior, we have used a cylindrical model for virtual mechanical traction tests. The top of the cylindrical model (fig. 4) is fixed, and different forces are applied to all the bottom point-masses. The figure 5 presents the results of the tests. We can see that the behavior of the global cylindrical model presents a similar shape to those of each of its connectors. So, the non-linear behavior of the connectors is faithfully reproducted on the global model.

3.3. Liver model

The model of the liver can be updated at a 150Hz frequency, and the local model gives a good haptic feedback. In the experiments, the simulations of the liver responses to various actions of the operator has shown qualitatively realistic behaviors : the liver was locally deformed under the effects of the forces applied using the rigid virtual tool and it went back to its initial shape

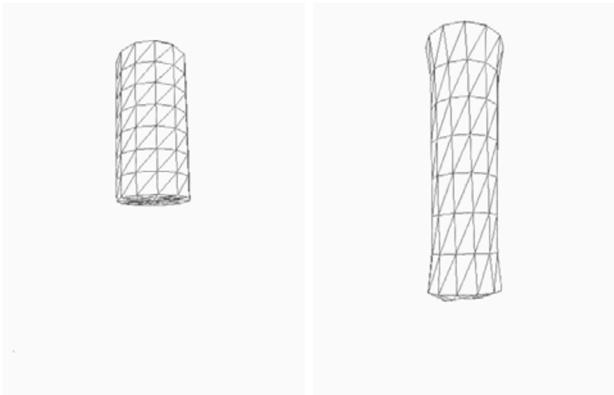

Figure 4. Cylindrical model at rest (left) and submitted to a six newton force (right).

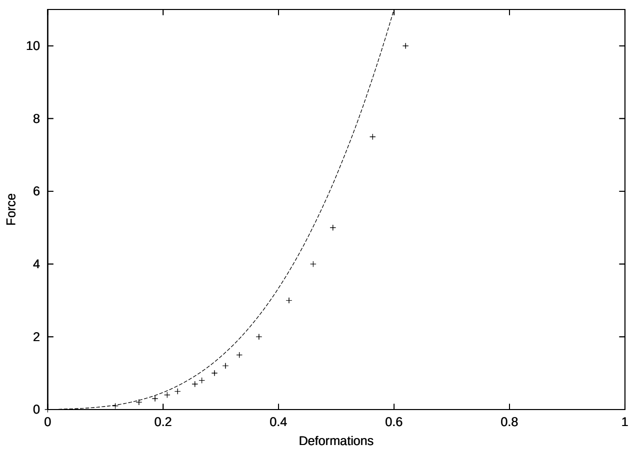

Figure 5. The dots curve is the mechanical law of the connectors (Eq. 1) and the cross were obtained by the virtual mechanical traction tests on the cylindrical model. Both curves presents the same non-linear shape.

as soon as no forces is applied to it. It was also possible to smoothly slide along its external surface, using the virtual tool.

Furthermore, if the operator, after having strongly pressed the liver, withdraws the tool, the liver didn't return quickly to its initial shape. This is due to the strong viscosity of the connectors of the Parenchyma, and simulates the malleable characteristic of the liver.

This model presents *qualitatively* a mechanical behavior which is closed to that of a real liver. It remains to find the exact numerical values of the parameters of elasticity and viscosity of the Parenchyma and the Capsule of Glisson. However, no force/displacement physical data is available yet, because

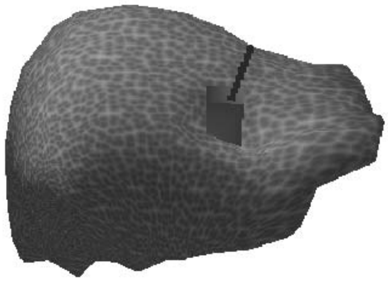

Figure 6. Interactive manipulation of the liver.

of the difficulty in making the measurements *in vivo* (the liver being made up mainly of blood, its dynamic behavior is very different when not irrigated, because blood coagulates quickly).

3.4. Haptic feedback

Figure 7 shows how the PHANToM force-feedback device is used to interact with the model of the liver. Position and orientation of the stylus in the real world determine the location of the virtual probe. The simulation sends back the forces due to interaction with the deformable model.

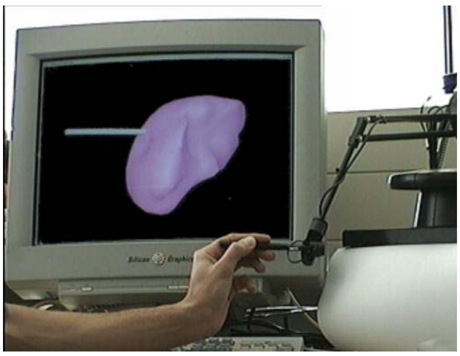

Figure 7. The liver model, being manipulated using the force feedback device.

Since the simulation frequency (Section 3.3) is insufficient to guarantee smooth haptic interaction the approach detailed in Section 2.5 is used. Figure 8 shows the difference in the force feedback, for the interaction with a deformable model, with and without the local approximation. The harsh peaks in force without the local model propel the user (quite violently) away from the object thereby loosing contact, i.e. forces return to nil).

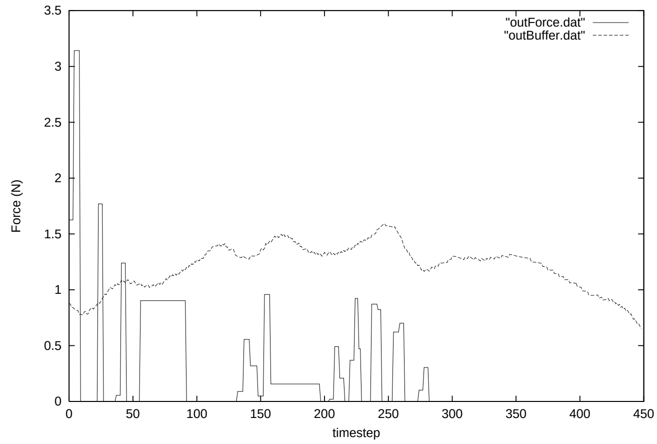

Figure 8. Evolution of the force over a time interval using the haptic local model (dashed), and without (solid).

3.5. Changing the topology

Our current cutting algorithm works only on membrane models. It is possible to cut a dynamic model, using the PHANToM device to control the position and orientation of a virtual tool. The model is cut exactly on the tool trajectory (figure 9).

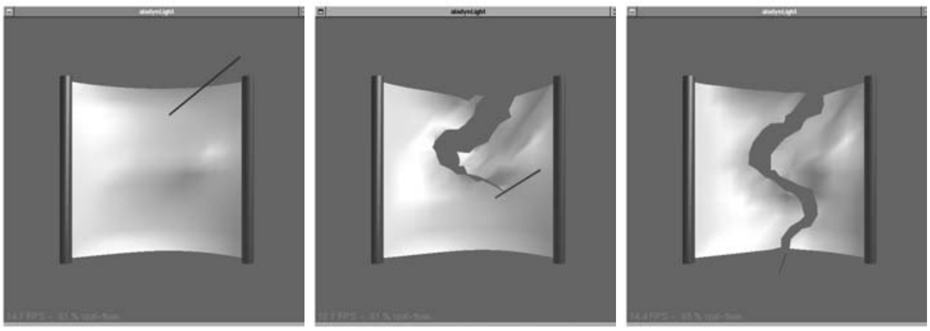

Figure 9. Interactive cut of a membrane model.

4. Conclusion and Perspectives

An heterogeneous, non-linear volumetric model of the liver based on a mass-spring model has been presented. In conformity with the reality, the model presents two different dynamic behaviors for the Parenchyma and the Capsule of Glisson. The optimized collisions detection algorithm makes possible to interact on the model in haptic real-time, using a virtual tool controlled in position.

However, it is important to remember that the elasticity and viscosity parameters of our model were chosen intuitively, to obtain a good *qualitative* behavior at a good simulation frequency, but real strain/displacement information is imperative to adjust the model.

References

[AH98] Astley (O.R.) et Hayward (V.). – Multirate haptic simulation achieved by coupling finite element meshes through norton equivalents. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation.*

[BdCL] Boux de Casson (F.) et Laugier (C.). – Modeling the dynamics of a human liver for a minimally invasive simulator. *In : Proc. of the Int. Conf. on Medical Image Computer-Assisted Intervention*, pp. 1156–1165. – Cambridge (GB), September.

[BL00] Boux de Casson (F.) et Laugier (C.). – Simulating 2d tearing phenomena for medical surgery simulators. *In : Computer Animation.*

[dBL] d'Aulignac (D.), Balaniuk (R.) et Laugier (C.). – A haptic interface for a virtual exam of the human thigh. *In : Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2452–2457. – San Francisco, CA (US), April.

[DJL98] Deguet (A.), Joukhadar (A.) et Laugier (C.). – Models and algorithms for the collision of rigid and deformable bodies. *In : Robotics: the algorithmic perspective*, éd. par Agarwal (P. K.), Kavraki (L. E.) et Mason (M. T.), pp. 327–338. – A K Peters, 1998. Proc. of the Workshop on the Algorithmic Foundations of Robotics. Houston, TX (US). March 1998.

[dLC99] d'Aulignac (D.), Laugier (C.) et Cavusoglu (M. C.). – Towards a realistic echographic simulator with force feedback. *In : Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 727–732. – Kyongju (KR), October 1999.

[GB98] George (P.L.) et Borouchaki (H.). – *Delaunay Triangulation and Meshing - Applications to Finite Elements.* – Éditions Hermès, 1998.

[JWLne] Joukhadar (A.), Wabbi (A.) et Laugier (C.). – Fast contact localisation between deformable polyhedra in motion. *In : Proc. of the IEEE Computer Animation Conf.*, pp. 126–135. – Geneva (CH), June june.

[LCF99] Lombardo (J.-C.), Cani (M.-P.) et F.Neyret. – Real-time collision detection for virtual surgery. *In : Computer Animation.* – Geneva Switzerland, May 26-28 1999.

[MRF+96] Mark (W.R.), Randolph (S.C.), Finch (M.), Van Verth (J.M.) et Taylor (R.M.). – Adding force feedback to graphical systems: Issues and solutions. *In : Computer Graphics* Proceedings, Annual Conference Series. – ACM SIGGRAPH.