Simulation of a Minimally Invasive Surgery of Intestines

Laure France Julien Lenoir Philippe Meseure Christophe Chaillou

LIFL, Villeneuve d'Ascq 59655, France

[Laure.France|Julien.Lenoir|Philippe.Meseure|Christophe.Chaillou]@lifl.fr

Abstract

In this paper, we propose an approach to simulate a minimally invasive surgery of intestines. The difficulty of such a simulation comes from the animation of intestines. As a matter of fact, intestines are a very long tube that is not isotropically elastic, and that bends over itself in various spots, creating multiple self-contacts.

Our approach to model intestines is based upon a mechanical part reduced to their axis, a skinning model to define their volume, and a specific sphere-based model to manage collisions and self-collisions. A direct application of this simulation is the training of a typical surgical gesture, to move apart the intestines in order to reach certain areas of the abdomen.

Keywords: Surgical simulation, virtual reality, physically based simulation, real-time computation, medical application.

1 Introduction

Laparoscopic surgery is a surgical technique that performs small incisions in the body, through which surgical tools and a micro video camera are inserted. This technique avoids large cuts of open surgeries, however this leads to new working conditions, in particular the hand-eye coordination. Therefore, training is necessary, and surgeons' rehearsals of this practice could be done on a simulator.

Different endoscopic simulators have been developed, some of them commercialized, for instance those by Simbionix, Surgical Science or Xitact[1]. Usually, they consist in simulating the interaction of surgical tools with a single organ. Nevertheless, there are no current satisfying solutions for the viscera, that are very deformable objects, and that cannot be simulated separately. Laying and being wound round all together, the viscera must be separated to reach some areas to treat, leading to the challenge of simulating large displacements in a multiple interaction context.

The clearing of the area to treat consists of pulling and folding the intestines so that they do not come back to their previous position. The simulation of this clearing stage necessary before some surgeries such as the removal of the gallbladder or colon tumors, has not been performed yet (to our knowledge). Besides, this action puts complex behaviors at stake: Large displacements, contacts and self-collisions. Therefore, we propose a simulator to perform the previous stage of intestines displacement.

The paper is organized as follows: In section 2, existing models and their implemented techniques are presented. Our method of intestines simulation is then proposed in section 3, followed by some results developed in section 4.

2 Previous Work

We present work relative to two aspects that we are concerned with: Soft tissue simulation and collision detection for deformable bodies.

2.1 Soft Tissue Simulation

Different approaches have been proposed to physically model deformable tissue in the context of medical simulators. Nevertheless, few of them ensure real-time interaction. A first type of approaches relies on mass-spring nets that discretize the materials with points linked by springs and dampers. The material behavior is then governed by the Newton law [2, 8, 14]. These models provide large deformations and displacements in real-time, however the material behavior is greatly conditioned by the topological design of the mass-spring net: Different nets will generate different global behaviors. Besides, there is no guarantee that the tissue deformations resulting from the interaction with the surroundings and the user, is physically valid.

A second type of approaches stems from finite element methods, where shapes are described by a set of basic elements and shape functions with limited support, enabling continuous representations of objects. The material behavior may then be computed essentially statically, based on a stress-strain relation applied on the finite element model

[1] Simbionix: http://www.simbionix.com/
Surgical Science: http://surgical-science.com/
Xitact: http://www.xitact.com/

[4, 9, 16]. Linear elasticity is often used, as a good approximation for the behavior of deformable objects, in the limit of small displacements and deformations (typically less than 10% of the mesh size [16]). These models are physically more accurate, however they are expensive both in computation and memory, yielding to non interactive simulations. Some improvements have been proposed to reduce the computation time, for instance in performing pre-computations [5], or in using a condensation-based method that separates surface nodes from volume nodes [1], or based on a boundary element method that directly deals with the surface boundary of the object [11].

Some models attempt to link both approaches, either based on finite difference methods and adaptive mechanisms [7], or based on mass tensor techniques that allow the resolution of the finite elements with mass-spring equations. The latter was notably applied to non-linear elastic models [16], thus providing more complex behaviors, in particular large displacements of models.

However, these types of methods are not applicable in our case, since they rely on a reference (rest) position of the object, and there is no such reference position for the intestines.

2.2 Collision Detection for Deformable Bodies

[13] proposed an interesting method to detect collisions in real time, between a deformable organ and a rigid surgical tool, by the means of graphics hardware. Indeed, the authors use the concept of viewing frustum, that they locate in the rigid tool. The hardware then "renders" what intersects this viewing volume, and thus what may be in collision with the rigid tool. Nevertheless, this method cannot be generalized to the collision detection between deformable objects.

Many approaches for collision detection are based on hierarchies of bounding volumes or spatial decompositions to address this problem. Only few of them are or may be adapted to handle deformable objects. They consist of building hierarchies of bounding volumes that ensure the best trade-off between a best approximation of objects and a straightforward checking of volume overlaps, for instance by using k-dops [12] or AABB trees [20]. Nonetheless, these methods provide no interpenetration processing.

Another approach is proposed by [9], based on the estimation of the penetration depth between deformable polyhedral objects, defined as finite elements with distance values. The positions of the volume mesh nodes are deduced from the resolution of finite elements. A fast marching level set method computes the depth of each node from the surface nodes of null depth. The depth values provide then an amount of intersection for the computation of the penetration potential energy used by penalty-based methods. Nevertheless, this method relies on tetrahedral volume meshes of objects.

In the context of interactive surgical simulation, we want to provide a fast collision detection method, common to all types of objects of the scene.

3 Model of intestines

We want to simulate the evolution of the behavior and the visual aspect of intestines under the action of surgical tools, and to return an appropriate haptic feedback via the user interface. We are particularly concerned by a more efficient detection and processing of collisions and prolonged contacts between intestines, surgical tools and other tissues in the abdominal cavity, and between parts of intestines.

Our approach relies on the following observation: We aim at simulating the displacement of the object, we thus focus on the displacement of its (virtual) skeleton[2]. In the case of intestines, this skeleton can be represented as a curve with mechanical properties. We then provide a skinning in order to represent and display the shape of the intestines (but with no mechanics at this level). Finally, since the object is able to interact and collide with its environment and some surgical tools, we use a detection model based on the mechanical model to compute these collisions and their response.

Our model thus consists of three components: A mechanical model to define its physical behavior; a skinning model to represent and display its shape; a model for self-collisions between its parts and collisions with other objects.

3.1 Mechanical Model

3.1.1 Modeling

We decide to define the intestines axis (skeleton) as a cubic Catmull-Rom [3] segmented spline. A 1D spline defines points $\mathbf{P}$ that are a linear combination of $n+1$ control points $\mathbf{q_k}$ and basis functions b_k:

$$\mathbf{P}(\mathbf{s},\mathbf{t}) = \sum_{k=0}^{n} \mathbf{q_k}(\mathbf{t}) b_k(s) \tag{1}$$

with t the time and $s \in [0,1]$ being the parametric abscissa along the entire curve. The corresponding velocity thus can be simply expressed as:

$$\dot{\mathbf{P}}(\mathbf{s},\mathbf{t}) = \sum_{k=0}^{n} \dot{\mathbf{q_k}}(\mathbf{t}) b_k(s) \tag{2}$$

[2] The same way as the human motion may be computed from the skeleton motion.

This definition allows the user to control the shape of the spline just by modifying the control points.

3.1.2 Mechanical Formalism

In order to animate the spline, we use the Lagrangian formalism that takes into account the continuity of the object and thus enables a continuous mass distribution along the curve. Moreover, it allows external actions and/or constraints to occur anywhere along the spline. Our work is inspired on the Lagrangian dynamic splines described in [18] for textile models. However, we show here that this formalism can be adapted to real-time simulations of a curve.

The Lagrangian mechanism is based on the equations:

$$\forall i, \quad \frac{\partial \frac{\partial K}{\partial \dot{q}_i^\alpha}}{\partial t} + \frac{\partial K}{\partial q_i^\alpha} = Q_i^\alpha + \frac{\partial E}{\partial q_i^\alpha} \tag{3}$$

where q_i^α represent the degrees of freedom of the object, $\dot{q}_i^\alpha$ the velocities of these degrees of freedom, K the kinetic energy of the object, E the potential energy and Q_i^α the work of conservative forces. A description of equation (3) can be found in [18], we detail however the left part in order to show an optimization important for real-time simulation purposes.

The expression of the velocity $\dot{\mathbf{P}}$ of equation (2) depending only on the velocity of the degrees of freedom (and not on the degrees of freedom), we obtain:

$$\forall i, \quad \frac{\partial K}{\partial q_i^\alpha} = 0 \tag{4}$$

$$\forall i, \quad \frac{\partial \frac{\partial K}{\partial \dot{q}_i^\alpha}}{\partial t} = m \sum_{k=0}^{n} \left(\int_0^1 b_i(s) b_k(s)\ ds \right) \ddot{q}_k^\alpha(t) \tag{5}$$

The left part of the Lagrangian equations (3) can be re-written as a matrix vector product of the mass matrix $\mathcal{M}$ and the acceleration vector $\mathbf{A}$:

$$\mathcal{M}\mathbf{A} = \begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{pmatrix} \begin{pmatrix} \mathbf{A^x} \\ \mathbf{A^y} \\ \mathbf{A^z} \end{pmatrix}$$

with $A_i^\alpha = \ddot{q}_i^\alpha$ where A_i^α is an element of A^α and $\alpha \in \{x, y, z\}$. The right part of equations (3) combines the other energies, such as the gravity, deformation energy, viscosity friction, and collision forces (details can be found in [18]). It can be noticed that such forces can be applied on any point of the spline, not only on control points, thanks to the Lagrangian formalism.

3.1.3 Deformation Energy

Concerning the deformation energy in particular, which is aimed at structuring the model, we use two methods that can be combined in equation (3).

The first method is based on springs to induce an internal potential energy in the system. We can simulate a strain energy by considering mechanical points of the spline that are consecutively linked. We are also able to simulate a curvature energy by joining a point to its next neighbors like in [17] (see Figure 1).

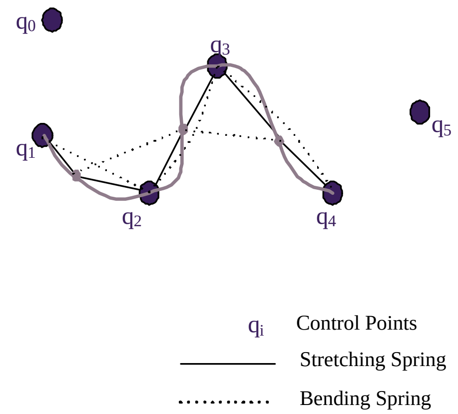


Figure 1: Spring distribution along the skeleton axis.

The second method provides a strain energy by considering the continuity of the spline. The continuous deformation energy considered in [15], based on the St Venant-Kirchhoff tensor, requires time-consuming computation, which is not conceivable for interactive time purposes. On the other hand, [19] introduced three terms for the computation of a continuous deformation energy for curves: One for the stretching energy, one for the curvature energy and one for the twisting energy. An approximation of the stretching energy relative to a degree of freedom is:

$$\hat{E}(t) = \frac{1}{2} k (l(t) - l_0)^2$$

where k is the curve stretching, l_0 the length of the rest position of the curve and l the length of the curve, namely the sum of the lengths of the spline sub-sampled segments. The variation of this energy with respect to a degree of freedom, necessary to the Lagrangian law, is approximated by a finite difference:

$$\frac{\partial \hat{E}}{d q_i^\alpha} = \hat{E}_{q_i^\alpha + \delta} - \hat{E}_{q_i^\alpha - \delta}$$

where $\hat{E}_{q_i^\alpha \pm \delta}$ represents the stretching energy with an infinitesimal variation of q_i^α.

With the method of [19], we ensure a global control of the stretching, while with springs, we ensure a local but looser control, both for the stretching and the curvature.

3.1.4 Real-time Simulation

After adding all these terms into the Lagrangian equations (3), this leads to the resolution of a system of the form $\mathcal{M}\mathbf{A} = \mathbf{B}$, namely the resolution of $\mathbf{A} = \mathcal{M}^{-1}\mathbf{B}$. It can be pointed out that $\mathcal{M}$ is a diagonal block matrix with identical diagonal elements M:

$$M_{ij} = m \int_0^1 b_i(s) b_j(s) \; ds$$

In [18], the authors remark that M is a symmetric, time-independent matrix since the basis functions used are commutative and time independent. This allows the pre-computation of the inverse matrix, yielding in a faster resolution of the equation system.

In this paper, we take benefit of an additional property provided by the cubic Catmull-Rom segmented spline: For any cubic spline, the matrix M is band and its band width w is proportional to the spline locality l (the locality is the number of segments on which a control point influences). Indeed, we have the following relationship: $w = 2l - 1$. Therefore, for a cubic spline of locality $l = 4$, the band width of the matrix M is 7. This property permits a new pre-computation of the banded matrix M by LU decomposition, fading out the resolution complexity from $O(n^2)$ to $O(nw)$. Since w is fixed ($w = 7$), the system resolution complexity becomes $O(n)$.

Once this equation system resolved, we get the acceleration of the degrees of freedom. We use an explicit integration method, Runge-Kutta 4, to compute the new velocities and new positions of the degrees of freedom.

3.2 Skinning Model

Once the intestines skeleton is mechanically computed at each time step of the simulation, we need a representation of the intestines in order to visualize them. The object representation must depend upon the position and the placement of its skeleton. In other words, the skeleton describing the general motion of the object, the skinning has to follow it, providing the resulting shape of the object.

Different methods exist to define a volume, some of them are more adapted to a tubular volume based on a linear skeleton. Indeed, surface models may be appropriate, especially for the modeling of cavernous tissues such as vessels, gallbladder, intestines. The idea is to create a volume delimited by a surface around the skeleton. We use a method similar to the one describing a colon model in [10]. However, intestines are more complex than the colon, since more curvatures and inflexion points occur. Besides, intestines undergo large displacements, whereas the colon is quite immobile.

We define the intestines surface as a parametric surface defined by a generalized cylinder with a spline skeleton associated to a circular section with a varying radius (see Figure 2). The Frenet basis represents a simple and efficient method to define a local frame on the axis curve. However, according to the configuration of the spline, important rotations of consecutive Frenet frames may occur, leading to unwanted twists. Hence, the solution we use is slightly different: For each parameter s subdividing a segment of the spline curve and $\mathbf{t_s}$ the tangent vector of the curve in this point, we construct the new coordinate frame $(\mathbf{t_s}, \mathbf{k_s}, \mathbf{b_s})$ based on the previous one $(\mathbf{t_{s-\delta s}}, \mathbf{k_{s-\delta s}}, \mathbf{b_{s-\delta s}})$ (δs being defined as the axis parameter step used for tesselation) such that:

$$\begin{aligned} \mathbf{b_s} &= \mathbf{t_s} \wedge \mathbf{k_{s-\delta s}} \\ \mathbf{k_s} &= \mathbf{b_s} \wedge \mathbf{t_s} \end{aligned}$$

The initial frame may be the Frenet frame for instance. For each other segment, $(\mathbf{t_0}, \mathbf{k_0}, \mathbf{b_0})$ is identical to $(\mathbf{t_1}, \mathbf{k_1}, \mathbf{b_1})$ of the previous segment. This method gives good results in practice.

We then define a certain number of points in the plane perpendicular to the curve, so that they discretize a circle of center the curve point and of radius obtained by $R(s) = R_{min} + cos^2(\pi s)\Delta R$, in order to get an approximate shape of the intestines (see Figure 2). We use $\Delta R = R_{max} - R_{min}$, and R_{min} and R_{max} respectively the maximum and the minimum radii allowed, and $s \in [0, 1]$ between two control points of the spline. Finally, we join points between circles to create facets. This representation allows a certain display of the intestines at interactive rates.

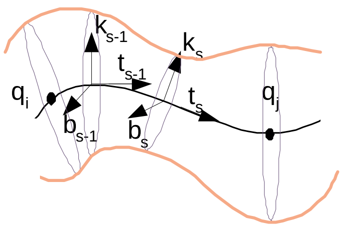


Figure 2: Skinning of the intestines.

3.3 Collision/Self-collision Model

The interaction between a soft tissue and its surroundings, such as surgical tools, other organs or tissues, can be decomposed into two tasks: The collision detection and the computation of resulting interaction forces or displacements. For many surgical instruments, the interaction with the tissue does not only occur at their extremity, but also along their length. Therefore, it is necessary to consider the surgical instruments as a whole. Similarly, it is necessary to consider all other objects in their totality, since they may interact together in various spots.

3.3.1 Collision Spheres

We choose penalty methods to cope with interaction between deformable bodies. Indeed, in the case of intestines, overlaps inevitably occur, and collision detection thus must measure the amount of interpenetration.

Surgical simulators require interactive simulations. Nevertheless, the simulation increases in complexity and in time-consumption while dealing with the collision detection between objects or parts of a same object. In order to be able to provide interactive simulations, we choose to use a simple and fast algorithm for the collision detection, that may be approximate but accurate enough for the considered scenes.

Our approach is based on the concept of collision spheres, where every object of the scene is approximated by spheres. We then use a regular 3D grid that discretizes the 3D space, and which voxels are filled with collision spheres of all objects at each simulation step. Each time a new sphere has to be added in a voxel, we check if there are already spheres contained in this voxel, via a sphere list maintained for each voxel, at each time step. In this case, for each sphere already present, we compute its distance to the new sphere. If actually there is a collision, a penalty force proportional to the penetration is added to the force vector of both spheres. When all the present spheres are processed, the new sphere is added in the sphere list of that voxel. A scan of all the voxels is never needed at any time (see [6] for more details).

3.3.2 Application to intestines

In the case of intestines, it consists of placing spheres of the same radius along the curve, uniformly according to the parametric abscissa (see Figure 3). When a surgical tool interacts with intestines, the intestines spheres ensure the detection of the collision and provide penalty forces as reaction response, yielding in the movement and/or deformation of the intestines and forces returned by the feedback device.

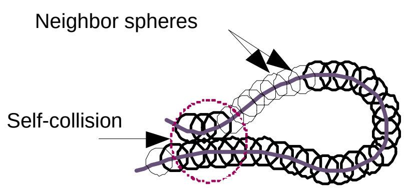


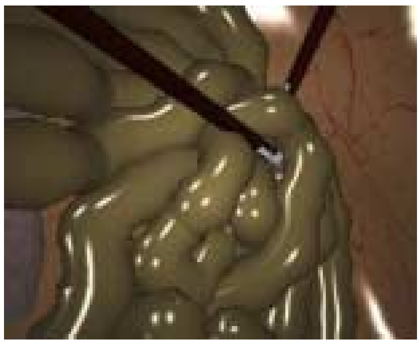
Figure 3: Sphere distribution along the skeleton axis of the intestines.

Concerning the management of self-collisions between parts of intestines, this is directly performed by the previous collision detection with the voxel method. Nevertheless, in the detected collisions, we must discard the spheres that are naturally neighbors (see Figure 3).

4 Results

We present results obtained with a 4 meter model of intestines. In the first case, we let it fall on a plane ground. We got a real-time simulation ($12ms$ for 250 control points on a bi-athlon $1.2Go$) that ended up in a quite realistic placement of the intestines (see Figure 4a). The intestines display is performed at $40fps$. In the second case, the intestines were placed in the abdominal cavity the same way, and surgical tools were moving them, performing the medical gesture of clearing (see Figure 4b)[3].

(a) Placement.



(b) In the cavity.

Figure 4: Simulation of intestines.

[3]Some videos are available at: http://www.lifl.fr/∼france/vric02.html

5 Conclusion

In this article, we presented a method to simulate in interactive time intestines in the abdominal cavity, for surgical training purposes. Intestines are very deformable objects, and during a surgery, they may undergo large displacements and multiple interactions.

We thus presented a method based on a skeleton defined by a spline to compute the mechanical motion of the intestines, from the Lagrange formalism that we optimized to ensure an accurate and fast simulation. We then proposed a skinning based on parametric surfaces around the skeleton, that offers quite good results in terms of visual effects concerning the movements and deformations of the intestines. Besides, we used collision spheres for a fast collision detection, providing real-time feedback via the haptic device. Finally, we obtained a reasonable basic framework to allow intestines surgery simulations.

However, several improvements can be envisaged. First of all, the visual realism could be improved by using implicit surfaces that particularly suit to organic shapes, and that perform more accurate surface deformations managed by self-collisions, but still time-consuming. On the other hand, self-collisions could be detected more precisely by taking into account the cylindrical aspect of intestines. Finally, we should obtain real parameters as inputs for our model, to be able to compare it with reality.

6 Acknowledgments

This work is supported by INRIA (French National Institute for Research in Computer Science and Control) as part of the ARC SCI (research action for Intestines Surgery Simulator). The authors would like to thank people of iMAGIS lab and IRCAD (French Digestive Cancer Research Institute) for their constructive discussions.

References

[1] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum 15(3):57–66*, 1996.

[2] H. Cakmak and U. Kuhnapfel. Animation and simulation techniques for vr-training systems in endoscopic surgery. *Eurographics Workshop on Animation and Simulation, pp. 173-185*, 2000.

[3] R. Catmull and R. Rom. A class of local interpolating splines. *Computer Aided Geometric Design, Academic Press*, 1974.

[4] S. Cotin, H. Delingette, and N. Ayache. Real time volumetric deformable models for surgery simulation. *Visualization in Biomedical Computing, Proceedings, volume 1131 of Lecture Notes in Computer Science. Springer Verlag*, 1996.

[5] 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.

[6] J. Davanne, P. Meseure, and C. Chaillou. Stable haptic interaction in a dynamic virtual environment. *Submitted to IROS'02*, 2002.

[7] G. Debunne, M. Desbrun, M.P. Cani, and A.H. Barr. Dynamic real-time deformations using space and time adaptive sampling. *Computer Graphics*, August 2001.

[8] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. A craniofacial surgery simulation testbed. *Visualization in Biomedical Computing, pp 607–618*, 1994.

[9] S. Fisher and M. Lin. Fast penetration depth estimation for elastic bodies using deformed distance fields. *IROS*, 2001.

[10] A. Haristis, D. Gillies, and C. Williams. Realistic generation and real time animation of images of the human colon. *Computer Graphics Forum 1992, vol. II, 3:367–379*, 1992.

[11] D.L. James and D.K. Pai. ARTDEFO: Accurate real time deformable objects. *Proc. ACM SIGGRAPH'99, pp 65–72*, 1999.

[12] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume of k-dop's. *TVCG(4), pp 21–36*, 1998.

[13] J.C. Lombardo, M.P. Cani, and F. Neyret. Real-time collision detection for virtual surgery. *Computer Animation'99*, May 1999.

[14] P. Meseure and C. Chaillou. Deformable body simulation with adaptive subdivision and cuttings. *Proceedings of the WSCG'97 Conference, pp 361–370*, 1997.

[15] O. Nocent and Y. Remion. Continuous deformation energy for dynamic material splines subject to finite displacements. *Eurographics CAS'2001*, 2001.

[16] G. Picinbono, H. Delingette, and N. Ayache. Non-linear and anisotropic elastic soft tissue models for medical simulation. *ICRA2001: IEEE International Conference Robotics and Automation, Seoul Korea*, May 2001.

[17] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface'95*, 1995.

[18] Y. Remion, J.M. Nourrit, and D. Gillard. Dynamic animation of spline like objects. *WSCG*, 1999.

[19] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, july 1987.

[20] G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools, 2(4):1–13*, 1997.