



A Layered Model of a Virtual Human Intestine for Surgery Simulation

Laure France, Julien Lenoir, Alexis Angelidis, P. Meseure, Marie-Paule Cani, François Faure, Christophe Chaillou

► To cite this version:

Laure France, Julien Lenoir, Alexis Angelidis, P. Meseure, Marie-Paule Cani, et al.. A Layered Model of a Virtual Human Intestine for Surgery Simulation. Medical Image Analysis, 2005. inria-00394478

HAL Id: inria-00394478

<https://inria.hal.science/inria-00394478>

Submitted on 11 Jun 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Layered Model of a Virtual Human Intestine for Surgery Simulation

L. France^a, J. Lenoir^b, A. Angelidis^c, P. Meseure^b,
M.-P. Cani^c, F. Faure^c, C. Chaillou^b

^a*Syscom, Université de Savoie, Chambéry, France*

^b*ALCOVE, INRIA Futurs, IRCICA-LIFL, Université de Lille I, Villeneuve
d'Ascq, France*

^c*GRAVIR, joint lab of CNRS, INRIA, INPG and Université J. Fourier,
Montbonnot, France*

Abstract

In this paper, we propose a new approach to simulate the small intestine in a context of laparoscopic surgery. The ultimate aim of this work is to simulate the training of a basic surgical gesture in real-time: moving aside the intestine to reach hidden areas of the abdomen. The main problem posed by this kind of simulation is animating the intestine. The problem comes from the nature of the intestine: a very long tube which is not isotropically elastic, and is contained in a volume that is small when compared to the intestine's length. It coils extensively and collides with itself in many places.

To do this, we use a layered model to animate the intestine. The intestine's axis is animated as a linear mechanical component. A specific sphere-based model handles contacts and self-collisions. A skinning model is used to create the intestine's volume around the axis. This paper discusses and compares three different representations for skinning the intestine: a parametric surface model and two implicit surface models. The first implicit surface model uses point skeletons while the second uses local convolution surfaces. Using these models, we obtained good-looking results in real-time. Some videos of this work can be found at:

<http://www-imagis.imag.fr/Publications/2004/FLAMCFC04/> .

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

PACS: 07.05.Tp

Email addresses: Laure.France@univ-savoie.fr (L. France),
[Julien.Lenoir|Philippe.Meseure|Christophe.Chaillou]@lifl.fr,
[Alexis.Angelidis|Marie-Paule.Cani|Francois.Faure]@imag.fr (M.-P. Cani).

1 Introduction

Laparoscopic surgery involves making small incisions in the body, through which surgical tools and a micro video camera are inserted. This technique avoids large cuts typical of conventional open surgery. It places the surgeon in peculiar working conditions, requiring among other skills a flawless hand-eye coordination. Training, therefore, is necessary, and could be done on a simulator.

Different endoscopic simulators are available commercially, such as the ones manufactured by Simbionix[35], Surgical Science[33] or Xitact[40]. These simulators usually handle the interaction of surgical tools on a single organ. Currently, there is no satisfying simulation solution for the viscera, which is made of easily deformed organs in contact, that cannot be simulated as totally independent components.

Being wound inside the abdominal cavity, the viscera must be pulled and folded with surgical tools to clear the area for treatment in such a way that they do return to their previous position. This manipulation is difficult for surgeons to perform correctly. Any mistake will prolong the procedure unnecessarily[11]. Since this clearing stage is necessary in almost all laparoscopic operations and surgeons have to train for a long time before acquiring the right technique, simulators will give them the opportunity to practice at a low cost.

Simulating a highly deformable and smooth organ such as the intestine is a challenge: the model must handle large displacements, contacts, and self-collisions in real-time. Moreover, the smooth aspect of the intestine must be adequately rendered. The goal of this work is to propose an intestine model well suited to the virtual surgical environment described above. To do this, we rely on a layered model, the components of which were described in two previously published conference proceedings[19,18]:

- a mechanical layer, which simulates the main intestine behavior and handles collisions with other objects as well as self-collisions,
- a smooth skin used at the rendering stage, for which three possible representations are discussed.

Obtaining a real-time simulation is a challenge with several temporal constraints: the first component (the mechanical layer) must be simulated at a high frequency ($100Hz$) to enable future user interaction via haptic devices [25], while the second component (the geometric skinning) can be updated at only $25Hz$ to display a continuous animation. Both simulations have to be performed in parallel.

This paper is organized as follows: Section 2 reviews related works. Sections 3 and 4 present our method of intestine simulation by describing the two components of the model. Section 5 contains the conclusion and discusses possible future refinements of our method.

2 Previous Work

We want to simulate the behavioral evolution and the visual aspect of the intestine under the action of surgical tools. In this section, we therefore present work related to two aspects of concern: soft tissue simulation and skinning.

2.1 *Soft Tissue Simulation*

Most previous deformable models for surgical simulation were dedicated to the simulation of volumetric objects (e.g. the liver). Some models rely on mass-spring nets [7,16,26] and others on finite elements [13,17,30]. Specific methods, such as condensation [6], reduction of the mesh to only surface elements [23] or adaptive simulation [15] were proposed to obtain real-time performance.

Unfortunately, volumetric models are not very well suited to represent the intestine, because the intestine is not isotropically elastic. Indeed, the intestine can withstand tight curvatures without any noticeable resistance, and it does not resume its shape after deformation (contrary to other organs like the liver). Since the geometry of the intestine is similar to a linear object, it seems more convenient to use a linear model (1D model), which provides better control of the strain, curvature, and twisting deformations. Therefore, to simulate the displacement of the intestine, we focus on the displacement of its virtual skeleton¹, represented as a curve with mechanical properties.

A good way of simulating the dynamic behavior of a linear object is presented by Remion et al. in [32]. The authors use a Lagrangian model to animate a 1D spline where the physical properties are distributed continuously along the curve. We rely on this approach for our mechanical component of the intestine. However, the solution in [32] must be adapted to meet the real-time constraints of our future goal to provide a simulator for haptic devices. Moreover, an adequate deformation energy must be proposed to emulate as best as possible the intestine's behavior. Several works can be used to define this energy. We have experimented with both spring-based energy as discussed by Provot in

¹ The same way as the human motion may be computed from the skeleton motion.

[31] and the continuous deformation energy as covered by Terzopoulos et al. in [36].

In addition, in the specific case of the intestine, we must handle multiple interactions and collisions, by taking into account the intestine's thickness. The interaction between those parts of the intestine in continuous contact with themselves and with the intestine and its environment, consisting of surgical tools, other organs or tissues, can be decomposed into two tasks: the collision detection, and the computation of resulting interaction forces or displacements.

To address this problem, many approaches for collision detection are based on hierarchies of bounding volumes or spatial decompositions. Only a few of them are adaptable enough to handle deformable objects. They do this by building hierarchies of bounding volumes that ensure an optimum trade-off between the best approximation of objects and a straightforward checking of volume overlaps, for instance by using spheres [22], k-dops [24] or AABB trees [39].

With regard to the collision response, different approaches exist to prevent the detected interpenetrations. Impulses methods [28] or analytical approaches [2] are however restricted to rigid bodies. Instead, penalty methods [29] do not try to prevent overlap, but they apply forces, depending on the penetration depth or volume to decrease this overlap. This has the advantage of speed. In addition, the problem of numerous sections of the intestine being in contact with one another is quite similar to the simulation of multi-body collision detection. Yet, Milenkovic et al. in [27] show that a method based on interpenetration is usually more efficient in this case.

To be able to provide interactive simulations, we prefer to use a simple and fast algorithm for both the collision detection and response, based on a set of approximating spheres which interpenetration generates penalty forces. This technique is a good approximation and provides good results for the considered virtual scenes.

2.2 *Skinning*

Skinning is meant to provide an elaborate visualization of an object based on a simpler model than that of the object itself, namely the skeleton, which emulates the object's behaviour. Skinning builds complex surfaces from a small set of skeleton model control points. Skinning is usually purely passive, attempting to restore skeleton deformations, without managing any supplementary deformations (radial deformations).

In [20], Haristis uses a series of circles placed regularly along a curve to display an “inside” view of the still colon. Our approach uses the same idea, but provides instead an animated tessellated surface for the external view of the intestine. We want to define a volume around the skeleton-axis of the intestine. Some methods are better adapted to build a tubular volume based on a linear skeleton.

A classical approach is to use parametric, generalized cylinders, based, for instance, on spline surfaces, as in [12]. A generalized cylinder is a solid whose axis is a 3D space curve. At any point on the axis, a closed cross section is defined, usually normal to the axis. The volume is then described by a bounded surface cross-section, which is swept along the 3D curve.

An alternative is to rely on implicit surfaces [3]. Implicit surfaces particularly suit organic shapes, since they define a volume around a skeleton, enabling realistic curved shapes and geometric contacts between surfaces. An implicit surface is defined by the set of points p at which a field function f has a constant value e (called iso-value): $S = \{p \in R^3 / f(p) = e\}$. Two kinds of implicit surfaces are of interest for this work:

- Distance surfaces [5,8]:

Given a skeleton primitive, the field it generates at a point p is the field generated by the closest point on the skeleton primitive. By summing all the individual fields of many skeleton primitives, a blending of them is obtained. The resulting implicit surface is a smooth blending of the individual surfaces, and may create bulges near the junctions [4].

- Convolution surfaces [4]:

Convolution surfaces have an advantage over distance surfaces for defining tubular shapes. This representation avoids the bulges in regions connecting the segments, and allows the number of segments to change on the fly without creating jumps of the implicit surface (as long as newly introduced vertices are smoothly translated).

For a single convolution segment-skeleton, the field value f at a point p is the sum of the fields of all the point-skeletons s along the segment S : $f = \int_{s \in S} f(p, S) ds$.

In [4], there are no analytical methods to compute this integral, and time consuming computations were proposed. In [9,21,34], other point-skeleton kernel functions were proposed, for which the integral has a closed-form solution. The solution proposed in [9] is the fastest, and is expressed as:

$$f(p) = \frac{\sin \alpha_1 - \sin \alpha_2}{d^2(p, H)} \quad (1)$$

Where $d(p, H)$ is the distance between p and its projection H on the skeleton segment, and α_1 and α_2 are the signed angles between this axis of projection and the lines linking p to the extremities of the skeleton segment.

3 Mechanical Model

As shown in section 2, a dynamic spline seems to be a good approach to compute the global dynamic motion of the intestine. This section details the way we use this model and explains how we extend it to handle the non-zero thickness of the intestine for collision processing.

3.1 Modeling

We define the intestine's axis (skeleton) as a cubic Catmull-Rom [10] 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 :

$$\mathbf{P}(\mathbf{s}, \mathbf{t}) = \sum_{k=0}^n \mathbf{q}_k(\mathbf{t}) b_k(s) \quad (2)$$

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

$$\dot{\mathbf{P}}(\mathbf{s}, \mathbf{t}) = \sum_{k=0}^n \dot{\mathbf{q}}_k(\mathbf{t}) b_k(s) \quad (3)$$

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

3.2 Lagrangian dynamic splines

To animate the spline, we use the Lagrangian formalism proposed by Remion et al. in [32]. This takes into account the continuity of the object and thus enables a continuous mass distribution along the curve. It allows external actions and/or constraints to occur anywhere along the spline. We show here that this formalism can be adapted to real-time simulations of a curve.

The Lagrangian mechanism is based on the equations:

$$\forall i, \quad \frac{\partial \frac{\partial K}{\partial \dot{q}_i^\alpha}}{\partial t} + \frac{\partial K}{\partial q_i^\alpha} = Q_i^\alpha + \frac{\partial E}{\partial q_i^\alpha} \quad (4)$$

where q_i^α represent the degrees of freedom of the object, \dot{q}_i^α the velocities of these degrees of freedom, K the kinetic energy of the object, E the potential

energy (deformation energy and gravity) and Q_i^α the non-conservative generalized forces (collisions forces). A description of equation (4) can be found in [32]. We detail, however, the left part to show an optimization that is important for real-time simulation purposes.

The expression of the velocity $\dot{\mathbf{P}}$ of equation (3), depends only on the velocity of the control points (and not on their position). Given this, we obtain:

$$\forall i, \quad \frac{\partial K}{\partial \dot{q}_i^\alpha} = 0 \quad (5)$$

$$\forall i, \quad \frac{\partial \frac{\partial K}{\partial \dot{q}_i^\alpha}}{\partial t} = m \sum_{k=0}^n \left(\int_0^1 b_i(s) b_k(s) ds \right) \ddot{q}_k^\alpha(t) \quad (6)$$

The left part of the Lagrangian equations (4) 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 (4) combines the other energies, such as the gravity, deformation energy, viscosity friction, and collision forces (details can be found in [32]). It will be noticed that such forces can be applied on any point of the spline, not only on control points, thanks to the Lagrangian formalism.

3.3 Deformation Energy

To define the deformation energy E , which is aimed at structuring the model (see equation (4)), we combine two methods.

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 immediate neighbors as presented by Provot in [31] (see Figure 1). The deformation energy is expressed as $E = \sum_{i=1}^N \frac{1}{2} k^i (l^i - l_0^i)^2$, where N is the number of springs, k their stiffness, l their current length and l_0 their rest length.

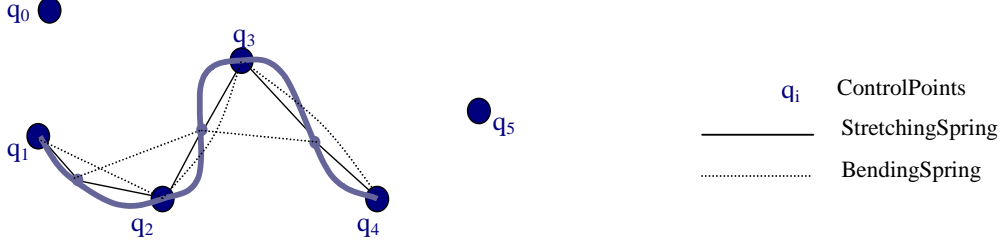


Fig. 1. Spring distribution along the skeleton axis.

The second method provides a strain energy by considering the continuity of the spline. Terzopoulos et al. in [36] 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:

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

with $k = \frac{\epsilon s l_0}{4}$

where k is the curve stretching, ϵ the Young modulus, s the object section, l_0 the curve length at the rest position and l the curve length, that is, 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:

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

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

In our experiments, we found that the continuous energy [36] is good for ensuring a global control of the stretching, but is not sufficient when used alone to handle local folds of the intestine. In practice, we thus combine the two energy terms: with the method described by [36], we ensure a global control of the stretching, while with springs, as in [31], we ensure a local but looser control of both the stretching and the curvature.

3.4 Real-time Simulation

Adding all these terms into the Lagrangian equations (4), leads to the resolution of a system in the form of $\mathcal{M}\mathbf{A} = \mathbf{B}$, that is, the resolution of $\mathbf{A} = \mathcal{M}^{-1}\mathbf{B}$, where \mathbf{A} is the variable. It should be noted 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 [32], Remion et al. remark that M is a symmetric, time-independent matrix because the basis functions used are commutative and time independent. We rely on this property to pre-compute the inverse matrix, yielding a faster resolution of the equation system.

In this paper, we benefit from an additional property provided by the cubic Catmull-Rom segmented spline: for any cubic spline, the matrix M is a band matrix, the width w of which is proportional to the spline locality l (the locality is the number of segments on which a control point influences). This creates 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 band 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 is 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.

3.5 Collision/Self-collision Model

As mentioned in section 2, penalty methods are highly appropriate for the provision of real-time response to collisions. Moreover, since only external forces are added, a penalty method can be easily combined with our dynamic spline skeleton by computing the corresponding generalized forces Q_i^α in equation (4).

We need a simple geometric model of interpenetration for computing these penalty forces. We currently approximate the intestine with a chain of spheres (see Figure 2). These spheres, of the same radius, are uniformly distributed

along the curve (see Figure 2). When a surgical tool interacts with the intestine, the intestine’s spheres detect the collision and provide penalty forces as reaction response, which yields the subsequent intestine movement or deformation.

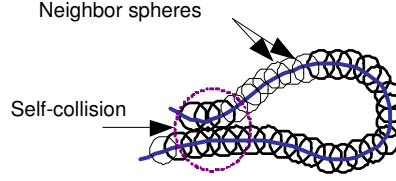


Fig. 2. Sphere distribution along the skeleton axis of the intestine.

The same approach is used for handling the intestine’s self-collisions. However, spheres that are neighbors along the skeleton (chain) do not figure in collision detection (see Figure 2).

Our method relies on a space grid used to accelerate the collision detection: only pairs of spheres located in the same voxel are considered. When two spheres belong to a same voxel, we compute the penetration distance from the spheres’ center and radius. We thus add a spring at the spheres’ center, to simulate the repelling action between them (see [14] for more details).

In practice, the collisions are checked at $50Hz$ (at least). This is fast enough not to miss collisions, since the surgeons’ gestures are relatively slow.

4 Skinning Model

The purpose of this section is to find a good way to compute the intestine surface display from the current position of its skeleton curve. We suggest, then analyze three different approaches.

4.1 Parametric Skinning

4.1.1 Method

We first define the intestine’s surface as a parametric surface defined by a generalized cylinder, with a spline skeleton associated with a circular section of a varying radius (see Figure 3). 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 at 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 tessellation) such that:

$$\begin{aligned}\mathbf{b}_s &= \mathbf{t}_s \wedge \mathbf{k}_{s-\delta s} \\ \mathbf{k}_s &= \mathbf{b}_s \wedge \mathbf{t}_s\end{aligned}$$

For example, the initial frame may be the Frenet frame, where for each other segment, the frame $(\mathbf{t}_0, \mathbf{k}_0, \mathbf{b}_0)$ is identical to the frame $(\mathbf{t}_1, \mathbf{k}_1, \mathbf{b}_1)$ of the previous segment. In practice, this method generates good results.

We then define a certain number of points in the plane perpendicular to the curve, so that they discretize a circle where the curve point is the center and where the radius is obtained by $R(s) = R_{min} + \cos^2(\pi s)\Delta R$, in order to get an approximate shape of the intestine (see Figure 3). We use $\Delta R = R_{max} - R_{min}$, and R_{min} and R_{max} respectively to represent the maximum and the minimum radii allowed, and $s \in [0, 1]$ between two control points of the spline. Finally, we join the points between the circles to create facets. This representation allows a certain type of display of the intestine at interactive rates.

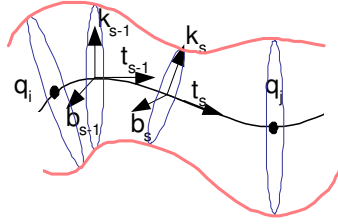
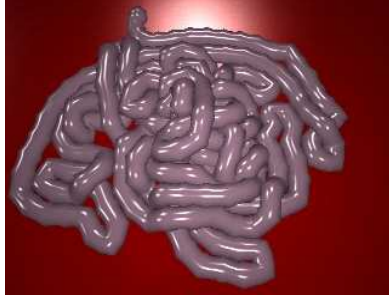


Fig. 3. Skinning of the intestine.

4.1.2 Results

Here, we present the results we obtained with a 4 meter model of intestine. In the first case, we simulated a fall onto a flat surface. The result was a real-time simulation (the computation time of a simulation step was $19ms$ for 250 control points on a Pentium IV $1.7GHz$ $512Mb$) that created a quite realistic placement of the intestine (see Figure 4a). The intestine display is performed at $90fps$. In the second case, the intestine was placed in the abdominal cavity in the same way, and surgical tools were used to move it, performing the medical gesture of clearing (see Figure 4b).

This representation allows an interactive display. Nevertheless, it suffers from a major drawback (see Figure 4b): the resulting shape is imperfectly rendered in high curvatures, due to tangent discontinuities and the fact that the thickness variations are only C^0 in these areas.



(a) Placement.



(b) In the cavity.

Fig. 4. Parametric intestine.

The remainder of this paper, therefore, presents two alternate methods based on implicit surfaces for the display of the intestine.

4.2 *Implicit Skinning using Point-Skeletons*

4.2.1 *Method*

We define the intestine’s surface as an implicit surface, which is generated by discrete point-skeletons positioned along the spline. The strength of their field value is tuned according to the radius of the intestine (see Figure 5 (a)). During the animation, the skeleton shape varies as the spline points move. However, if two points of the elastic spline are too far from each other, the corresponding surfaces will no longer blend, thus separating the object into two parts. In order to avoid these topology changes, the number of sampling points must change during the simulation. This is done by adaptively placing spheres along the curve at intervals, the range of which is a function of the spline segment length and the radius of the spheres. Next, we use a real-time implementation of the marching cubes algorithm proposed by [38] for the visualization of the implicit surface.

4.2.2 *Results*

One purpose of the blended surface is to provide a smooth shape for the intestine model (see Figure 5 (b)). However, it is particularly difficult to provide a constant radius. Indeed, the variation of the number of skeletons leads to fluctuations of the geometry (this could represent the spasms of the intestine, but we want to control them²). In addition, care must be taken to avoid blend-

² Even when the intestine is at rest, it can undergo periodic spasms (related to digestion and elimination), which we want to simulate. Although the implicit skinning

ing between non-consecutive parts of the intestine. To do this, we require the use of the blending graph. Unfortunately, in our implementation, the blending control cannot be done at a sufficient rate due to the marching cube method to display the surface [37]. We checked the same simulation that we used in the parametric case, that is, the intestine falling on a flat surface. The intestine display is performed at $19fps$ with no blend control and at $1.1fps$ with the blend control.

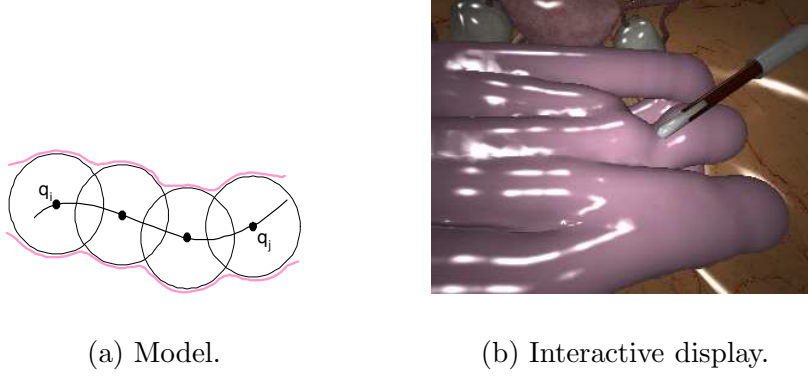


Fig. 5. Implicit intestine using point-skeletons.

4.3 Implicit Skinning generated by a Convolution Surface

To avoid the flaws of a varying discrete number of point-skeletons, we must use the whole skeleton curve or at least a linear approximation of this curve to generate the intestine geometry.

4.3.1 Method

A straightforward way of modeling a smooth, complex surface would then consist of using convolution surfaces, by summing all the individual field contributions of the simple segment-skeletons. This would not however prevent the surface from blending between non-consecutive parts. To prevent this, we use the controlled blending technique developed by Angelidis et al. in [1], which also ensures the display of surfaces at an interactive rate.

First, we need to describe the sampling of the implicit surface. We use the seed-based method of Cani et al. in [9] which benefits from the temporal coherence. Seeds migrate toward the surface according to the iso-surface. Initially, they start from the skeleton, and in the remainder of the simulation, they start from their previous positions. The discretization of the surface triangulation

using point-skeletons creates some spasm artifacts, they are not controlled.

is adaptive, allowing the surface to be rendered at different levels of detail. To produce this adaptive sampling, a rough polygonization is initially attached to a few sampling points, and then refined recursively by a uniform subdivision until a certain criterion is satisfied.

Finally, the unwanted blending is managed by local convolution [1] (see Figure 6 (a)). The display method then assigns to every point $p \in \mathbb{R}^3$ on the surface a parameter u along the skeleton, which corresponds to the attached point of the seed axis. A local portion of the skeleton around u is used to compute $f(p)$, which is a neighborhood of constant size D around u . The field value is then computed exactly as before, with the exception that only the useful part of the skeleton is considered in this case. A constraint on D is $D < 2R$ (where R is the intestine radius), since two points on the surface separated by a distance greater than $2R$ must not influence each other.

4.3.2 Results

With this model, the visual rendering of the intestine is adequate: there is no blending, nor are any bulges created (see Figures 6 (b), 7 and 8). However, the computation time is still too long if we want a fine discretization of the surface object. To obtain the same simulation used in the parametric case, the intestine falling on a flat surface, we must perform the intestine display at $3.9fps$.

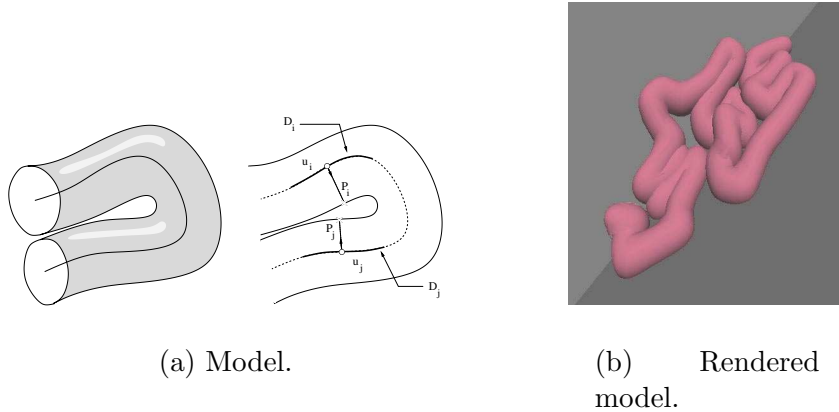


Fig. 6. Implicit intestine generated by a subdivision curve.

4.4 Discussion

We used three different models to represent the intestine shape. On one hand, the parametric model is the faster to display, but presents some visual artifacts. On the other hand, the implicit models provide satisfying visual results,

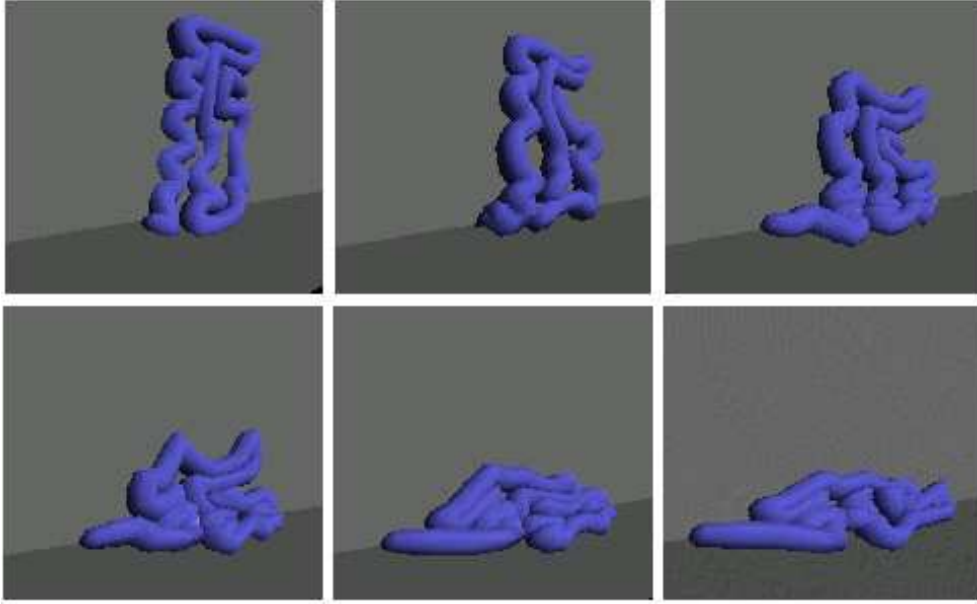
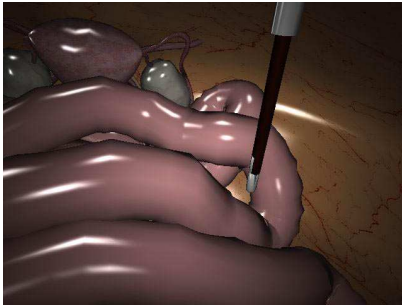


Fig. 7. Display of intestine only animated with gravity.



(a) View 1.



(b) View 2.

Fig. 8. Implicit intestine in the abdominal cavity.

since they naturally can represent organic shapes. Nonetheless, due to their respective complexity, they can hardly be displayed at interactive rate. In particular, the first model with point-skeletons needs a control via the blending graph, that makes it much slower than the model based on convolution surface.

Therefore, the choice of the kind of display can rely on what we want to simulate: if the intestine is not the main object of the simulation but just an element of it, the parametric model may be reasonably good. On the contrary, if the intestine is the focused object, for instance undergoing an operation, the implicit model based on convolution surface may be more appropriate.

5 Conclusion

In this paper, we presented a method to simulate in interactive time the intestine in the abdominal cavity, for surgical training purposes. The intestine is a very deformable object, and during a surgery, it may undergo large displacements and multiple interactions. This paper focuses on two contributions.

We presented first a method based on a skeleton defined by a spline to compute the mechanical motion of the intestine, from the Lagrange formalism, which we optimized to obtain a fast simulation. We then used collision spheres for fast collision detection and response.

Next, we compared different methods of skinning this model. First, we created a skinning using parametric surfaces around the skeleton, but this produced rough visual effects associated with the curvatures of the intestine, however it was obtained in real-time. We then compared two implicit solutions that offer quite good results as far as visual effects are concerned, providing good shape and deformations of the intestine³, but requiring more computation time.

Finally, we obtained a reasonable basic framework to allow intestine surgery simulations, by showing, in the context of a surgical simulator, that skinning by implicit surfaces can provide an attractive model when the surgical procedure focuses on the intestine; otherwise, the skinning by parametric surfaces method may be more appropriate. Some videos showing our results can be found at <http://www-imagis.imag.fr/Publications/2004/FLAMCFC04/>.

Nevertheless, several improvements can be envisaged. For example, it may be desirable to let the simulation switch between the two methods, providing better resolution for up close work and faster rendering for more general overviews. The challenge in doing this will be to create a simulation that lets us display the area of focus using the implicit technique, while simultaneously displaying the non-focused areas using the parametric technique; and, which lets us switch between the two display techniques, as the area of focus changes, seamlessly.

Moreover, only the discretization of the surface triangulation is currently adaptive. Future work might focus on dynamically adapting the discretization of the skeleton according to the curvature as it varies over time. The introduction of adaptive methods would be especially beneficial, since this would allow

³ We restore the skeleton deformation as much as possible. When the skeleton curves itself, so does the surface, therefore, the surface is deformed. Nevertheless, the radial deformations of the intestine are not taken into account since they are independent of the skeleton.

the animation to be displayed at different levels of detail, thereby obtaining interactive rates on any workstation.

Self-collisions could also be detected more precisely by taking into account the cylindrical aspect of the intestine or the information provided by the implicit surface. It may also be possible to add the contact surfaces presented by Angelidis et al. in [1] so that the contact is better handled.

We should, nevertheless, obtain real-world data as input for our model. This will allow us to prove our model against real-life examples.

Finally, it should be noted that we can apply this model to other tubular objects in the human body, such as the Fallopian Tubes, as shown in Figure 9.

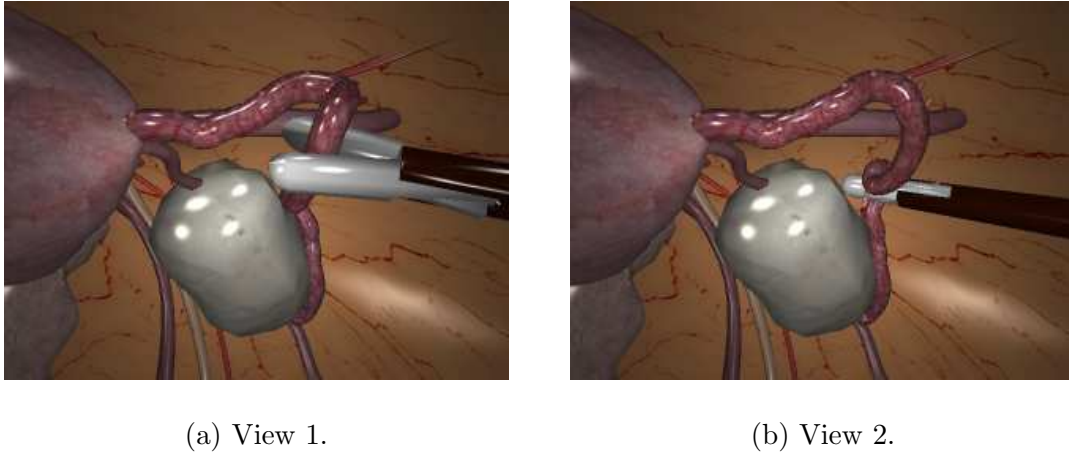


Fig. 9. Fallopian Tubes modeled by our three-component method.

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 Intestine Surgery Simulator). The authors would like to thank people of IRCAD (French Digestive Cancer Research Institute) for their constructive discussions, Bruce Cane for the English proofreading of the paper, and the reviewers for their helpful comments.

References

- [1] A. Angelidis, P. Jepp, and M.-P. Cani. Implicit modeling with skeleton curves: Controlled blending in contact situations. *Shape Modeling International*, 2002.
- [2] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *ACM SIGGRAPH Computer Graphics*, v.23 n.3, p.223-232, July 1989.
- [3] J. Bloomenthal, C. Bajaj, J. Blinn, M.-P. Cani, A. Rockwood, B. Wyvill, and G. Wyvill. *Introduction to Implicit Surfaces*. Morgan Kaufmann, July 1997.
- [4] J. Bloomenthal and K. Shoemake. Convolution surfaces. *Computer Graphics*, 25(2):251-256, July 1991.
- [5] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(2):109-116, March 1990.
- [6] 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.
- [7] H. Cakmak and U. Kohnapfel. Animation and simulation techniques for vr-training systems in endoscopic surgery. *Eurographics Workshop on Animation and Simulation*, pp. 173-185, 2000.
- [8] M.-P. Cani and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 1(3), March 1997. Published under the name M.-P. Cani-Gascuel.
- [9] M.-P. Cani and S. Hornus. Subdivision curve primitives: a new solution for interactive implicit modeling. In *Shape Modelling International*, May 2001.
- [10] R. Catmull and R. Rom. A class of local interpolating splines. *Computer Aided Geometric Design*, Academic Press, 1974.
- [11] Personal communication of Luc Soler. Virtual-surg manager at IRCAD. http://www.virtual-surg.com/English/index_en.htm, 2002.
- [12] S. Coquillart. Computing offsets of b-spline curves. *Computer Aided Design*, Vol. 19, No. 6, p305-309, 1987.
- [13] S. Cotin, H. Delingette, and N. Ayache. Real time volumetric deformable models for surgery simulation. *Visualization in Biomedical Computing*, volume 1131. Springer Verlag, 1996.
- [14] J. Davanne, P. Meseure, and C. Chaillou. Stable haptic interaction in a dynamic virtual environment. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [15] 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.

- [16] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. A craniofacial surgery simulation testbed. *Visualization in Biomedical Computing*, pp 607–618, 1994.
- [17] S. Fisher and M. Lin. Fast penetration depth estimation for elastic bodies using deformed distance fields. *IROS*, 2001.
- [18] L. France, A. Angelidis, P. Meseure, M.-P. Cani, J. Lenoir, F. Faure, and C. Chaillou. Implicit representations of the human intestines for surgery simulations. *Conference on Modelling and Simulation for Computer-aided Medicine and Surgery*, 2002.
- [19] L. France, J. Lenoir, P. Meseure, and C. Chaillou. Simulation of a minimally invasive surgery of intestines. *Virtual Reality International Conference*, pp 21–27, 2002.
- [20] 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.
- [21] S. Hornus, A. Angelidis, and M.-P. Cani. Implicit modelling using subdivision-curves. *The Visual Computer*, 2002.
- [22] P.M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–209, July 1996.
- [23] D.L. James and D.K. Pai. ARTDEFO: Accurate real time deformable objects. *Proc. ACM SIGGRAPH*, pp 65–72, 1999.
- [24] 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.
- [25] D. Lamy and C. Chaillou. Design, implementation and evaluation of an haptic interface for surgical gesture training. *Virtual Reality and Prototyping Conference, 7mes journées du Groupe de Travail Ralit Virtuelle, Laval*, pages 107–116, June 1999.
- [26] P. Meseure and C. Chaillou. Deformable body simulation with adaptive subdivision and cuttings. *Proc. WSCG Conference*, pp 361–370, 1997.
- [27] V. Milenkovic and H. Schmidl. Optimization-based animation. *SIGGRAPH’01 Conference Proceedings, Computer Graphics annual conference series, Los Angeles*, pages 37–46, August 2001.
- [28] B. Mirtich and J. Canny. Impulse-based dynamic simulation. *Proceedings of Workshop on Algorithmic Foundations of Robotics*, February 1994.
- [29] M. Moore and J. Wilhelms. Collision detection and response for computer animation. *SIGGRAPH’88 Conference Proceedings, Computer Graphics, Atlanta*, 22(4):289–298, August 1988.
- [30] G. Picinbono, H. Delingette, and N. Ayache. Non-linear and anisotropic elastic soft tissue models for medical simulation. *IEEE International Conference Robotics and Automation*, May 2001.

- [31] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface*, 1995.
- [32] Y. Remion, J.M. Nourrit, and D. Gillard. Dynamic animation of spline like objects. *WSCG*, 1999.
- [33] Surgical Science. <http://surgical-science.com/>.
- [34] A. Sherstyuk. Kernel functions in convolution surfaces: a comparative analysis. *The Visual Computer*, 15(4), 1999.
- [35] Simbionix. <http://www.simbionix.com/>.
- [36] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, July 1987.
- [37] F. Triquet, L. Grisoni, P. Meseure, and C. Chaillou. Realtime visualization of implicit objects with contact control. *International Conference on Computer Graphics and Interactive Techniques in Australia and South East Asia*, February 2003.
- [38] F. Triquet, P. Meseure, and C. Chaillou. Fast polygonization of implicit surfaces. *WSCG (2):283–290*, February 2001.
- [39] G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.
- [40] Xitact. <http://www.xitact.com/>.