SURGICAL THREAD SIMULATION

Julien Lenoir[1], Philippe Meseure[1], Laurent Grisoni[1] and Christophe Chaillou[1]

Abstract. In this paper, we propose a surgical thread model for suture training and in a more general way thread manipulation. This model is based on dynamic Lagrangian splines constrained by Lagrangian multipliers and penalty-based self-collisions, allowing tying knots simulation in interactive time.

Keywords: Surgical simulation, animated curve, physically-based animation, skinning, suture.

Résumé. Dans cet article, nous proposons un modèle de fil pour l'apprentissage de certains gestes chirurgicaux comme la suture d'une plaie ou la réalisation de noeuds. Ce modèle est basé sur l'animation d'une spline par la dynamique Lagrangienne contrainte par les multiplicateurs de Lagrange et dont les auto-collisions sont traitées par des méthodes à pénalité. Cela permet de simuler la création de noeuds en temps interactif.

Mots clés: Simulation chirurgicale, animation de courbe basée sur la physique, habillage, suture.

INTRODUCTION

Surgical simulation is constantly in progress and stakes are important, opening a new way in surgical gesture training. As presented in [Del98], surgical simulators are subject to many developments in graphics domain. They have to deal with many issues such as visual representation, physical model, collisions and force feedback.

In surgical simulators, we find models for organs but rarely for surgical threads. However, surgeons frequently have to make knots with a surgical thread for sewing the tissues. This procedure requires precision and control for the practitioner. More generally, the surgeon must be familiar with the manipulation of a thread. As a result, it is useful to provide a tool that allows surgeons to train for thread manipulation using a virtual model that is as realistic as possible, including the critical case of knot creation for suture.

This paper presents a model of surgical thread built within an existing platform [JQD+00]. The simulation is done at interactive rate in order to make the platform able to control a force feedback device like Phantom (developped by SensAble Technologies) ([DMC02]).

This paper is structured as follows: In section 1, we present the previous work, then we propose in section 2 the modeling of the thread and different ways to draw it. In section 3, we expose the mechanical law applied to the thread which animates the curve and permits collision detection to interact with it. In section 4, we develop some constraints by Lagrangian multipliers which may impose some behaviour to the curve. Section 5 presents some results before concluding.

---

[1] LIFL, Université Lille I, Villeneuve d'Ascq.

[lenoir|meseure|grisoni|chaillou]@lifl.fr

1. Previous Work

Many developments in the past offer a precious framework for the future researches. Surgical Science[1] proposes a model of thread which behaviour is discontinuous. A physically-based model for regular behaviour is used and when a knot is about to be completed, change in the topology occurs brutally causing simulation discontinuities which may create instability. We aim at offering a model based only on a physical simulation, without discontinuities. To our knowledge, only one study aims at simulating surgical thread completely ([Pai02]). However, this model is static and fix constraints on the model's extremity and the implementation is quiet complex. Nevertheless, some works based on n dimension models can be adapted to the one dimension case (see [TPBF87] and [RNN00]).

Deformable objects are inherently subject to deformation energies. Terzopoulos et al. [TPBF87] introduce a continuous energy for volume, surface and curve by considering a metric tensor, which gives an expression of the geometric deformation. Theory of physical deformation based on the Cauchy tensor can be also used as described in [DDCB01], [PDA01], [NR01], since this tensor expresses the deformation of the material with physical parameters. A possible mechanical model for the thread is a spring chain which articulates the curve. Despite the works of [GW97], on the fact that exact simulation by springs is impossible, this model is realistic enough in our case. However, some problems occur when the mechanic is held by only some points: the control of the curve becomes awkward and makes the knot creation very difficult. In order to resolve both problems, a solution consists in using a continuous formalism. The Finite Element Method ([PDA01]) is a usual approach. However, this model is not adapted to a surgical thread simulation, since it is based on a rest form of the objects, that a surgical thread does not have. We choose to use the Lagrangian formalism introduced by Remion et al. in [RNG99] for physically-based simulation of splines. This formalism permits to consider continuous mass and energy, it also allows the application of any force or constraint on any point of the curve.

2. Thread Modeling

A convenient approach to represent curves is to use splines. These models offer an easy manipulation of the shape of the final curve by control points. Besides, a subclasse of splines like Catmull-Rom or BSpline, can limit the incidence of control points to a local area. In practice, we use Catmull-Rom splines for their interpolation property and uniform cubic BSpline for their better continuity which makes them better for simulation. These splines are defined by $n + 1$ control points q_k and $n + 1$ basis functions b_k by the equation:

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

with t the time and $s \in [0, 1]$ the parametric abscissa along the entire curve.

The representation of the thread can be performed by drawing cylinders so that the surgeon is able to apprehend the thread manipulation through its thickness. Another solution is to skin the curve with a generalized cylinder, like in [Bin87], [Blo90]. We propose both solutions in our system.

3. Mechanical Simulation of the thread

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 by the implementation of Lagrangian theory described in [RNG99] for textile models. However, we aim at applying this formalism for real-time simulations of a thread.

---

[1] surgical-science.com

The Lagrangian mechanism is based on the equation:

$$\frac{\partial \frac{\partial K}{\partial \dot{q}_i^\alpha}}{\partial t} + \frac{\partial K}{\partial q_i^\alpha} = Q_i^\alpha + \frac{\partial E}{\partial q_i^\alpha} \qquad (2)$$

where q_i^α represent the degrees of freedom of the object, $\dot{q}_i^\alpha$ the partial derivative $\frac{\partial q_i^\alpha}{\partial t}$. , K the kinetic energy of the object, E the potential energy and Q_i^α the work of conservative forces applied to the curve.

A description of the equation (2) can be found in [RNG99]. However, we detail the left part in order to show a possible optimization, important for real-time simulation purposes.

Considering the position of a point along the curve given in equation (1), the corresponding velocity can be simply expressed as: $\dot{\mathbf{P}}(\mathbf{s}, \mathbf{t}) = \sum_{k=0}^n \dot{\mathbf{q}}_{\mathbf{k}}(\mathbf{t}) b_k(s)$. The kinetic energy K of the system is then computed by summing the kinetic energies of all points of the curve, giving: $K(t) = \int_0^1 \frac{1}{2} m(s,t) || \sum_{k=0}^n \dot{\mathbf{q}}_{\mathbf{k}}(\mathbf{t}) b_k(s) ||^2 ds$

We assume that the mass is constant along the curve. It thus depends neither on the time nor on the parametric abscissa, leading to the calculation simplification of the kinetic energy. After simplification, we obtain for the left-hand parts of equation (2) :

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

These equations can be represented using matrix notations, involving product of the mass matrix $\mathcal{M}$ and the acceleration vector $\mathbf{A}$:

$$\mathcal{M}\mathbf{A} = \left(\begin{array}{ccc} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{array} \right) \left(\begin{array}{c} \mathbf{A}^{\mathbf{x}} \\ \mathbf{A}^{\mathbf{y}} \\ \mathbf{A}^{\mathbf{z}} \end{array} \right) \qquad (3)$$

with $A^\alpha = (A_0^\alpha, A_1^\alpha ..., A_n^\alpha)^T$, where $A_i^\alpha = \ddot{q}_i^\alpha$ and $\alpha \in \{x, y, z\}$, and M is composed by $M_{ij} = m \int_0^1 b_i(s) b_j(s) \ ds$.

On the other hand, the right part of equation (2) combines the other energies, such as gravity, deformation energy, viscosity friction, and collision forces (details can be found in [RNG99]). 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.

In order to take into account the interactions with the surgeon and the environment, we have to detect the collisions with other objects present in the scene. This is performed by approximating all objects by spheres (cf. [DMC02] for more details). In the case of the thread, spheres are created by a sampling along the curve spline. The collision detection is then reduced to spheres collision, leading to the response to be calculated by a penalty method and directly integrated in the system equations (2).

Real-time Simulation:
After adding all the energies terms into the Lagrangian equation (2), 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 blocks M.

In [RNG99], 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 system.

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

Once this system is solved, we have the acceleration of the degrees of freedom namely the control points's coordinates. Then, we use a Runge Kutta 4 numerical integration or an implicit method of numerical integration provided in [HMC01] to estimate the new velocities and new positions of the degrees of freedom.

Deformation Energy :
Concerning the deformation energy aimed at structuring the model, we use two methods that can be combined in equation (2).

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 in a similar way than [Pro95] (see Figure 1).

FIGURE 1. Spring distribution over a spline

The second method provides a strain energy by considering the continuity of the spline. The continuous deformation energy presented in [NR01], based on the St Venant-Kirchhoff tensor, requires time-consuming computations, which is not conceivable for interactive time purposes. On the other hand, we can approximate the stretching energy proposed by [TPBF87] with:

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

where $k = \frac{Esl_0}{4}$, l_0 the length of the rest position of the curve, l the length of the curve, namely the sum of the lengths of the spline sub-sampled segments, E the Young's modulus and s the section area of the thread. The variation of this energy with respect to a degree of freedom, necessary to the Lagrangian law, is approximated by a finite difference:

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

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

With the second method, we ensure a global control of the stretching, while with the first method, we ensure a local but looser control, both for the stretching and the curvature.

4. Constraints and Knot creation

Self-collisions of the thread are treated the same way as collisions, by detecting collisions on the thread's spheres and computing forces with the penalty method. This allows the formation of knots but that are unstable and breakable.

Actually, practitioner has to set several thread points to an organ in order to sew up a wound. This can be realised by imposing constraints to the system. Setting constraints can be done by various approaches ([MW88]): Penalty methods provide approximative solutions that are accurate enough for collisions and self-collisions response but are unsuitable for mechanically fixing a point; More complex methods based on the Lagrangian multipliers [RNG99] consist of introducing new unknowns in the system. Each of these unknowns has

to preserve one atomic constraint. For example, a fixed-point constraint is formed by three atomic constraints, one for each axis. The Lagrangian equation system (3) extended to the Lagrangian multipliers leads to the following matrix system:

$$\begin{pmatrix} M & 0 & 0 & -Lx^T \\ 0 & M & 0 & -Ly^T \\ 0 & 0 & M & -Lz^T \\ Lx & Ly & Lz & 0 \end{pmatrix} \begin{pmatrix} \mathbf{A^x} \\ \mathbf{A^y} \\ \mathbf{A^z} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{B^x} \\ \mathbf{B^y} \\ \mathbf{B^z} \\ \mathbf{E} \end{pmatrix}$$

where L constitute the constraints coefficients, $\mathbf{E}$ correspond to the constant terms, and $\boldsymbol{\lambda}$ are the Lagrange multipliers.

This new system can be solved by the method presented in [Rem00]. It separates the acceleration into a tendency acceleration which gives the system acceleration without constraints and into a correction acceleration which expresses the links between the acceleration and the constraints. Unfortunately, the addition of constraints makes the resolution heavier since its complexity grows to $O(cn^2 + c^2n + c^3)$ with c the number of constraints, this implies that only a limited number of constraints can be simulated at interactive rate.

5. Results

FIGURE 2. Curves Examples

(a) Constraints illustration   (b) Knot formation   (c) Curves interactions
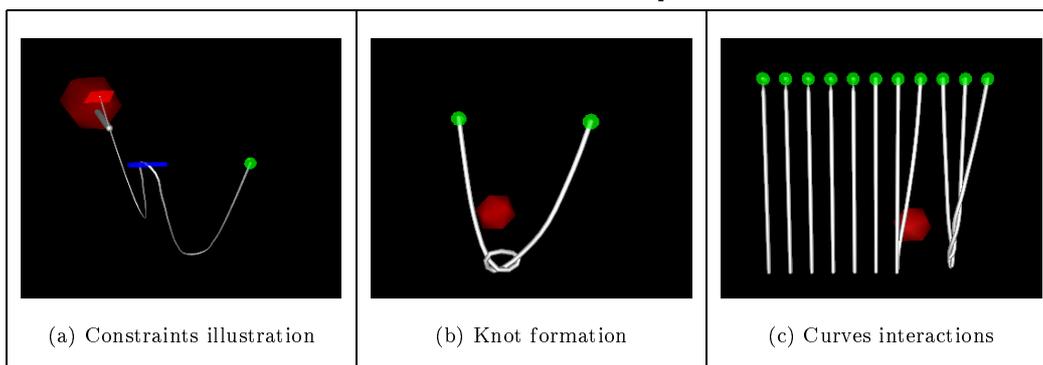
The figure (2a) illustrates different constraints that can be applied to a spline curve: A fixed point (modeled by the green sphere), a point constrained to an axe (modeled by the blue cylinder) or to a plane (modeled by the red plane). The user manipulates a virtual probe represented by the red sphere, via the Phantom. Interaction between the probe and the curve is shown by the white cylinder linking the user to the curve. To show the knot formation, we take an initial configuration of the curve initiating a knot and we fix the two extremities of the curve. The sole action of the weight clamps the knot (figure 2b which is a BSpline of 13 control points and 6 constraints). The self-collision being treated by the penalty method, a knot can be broken easily by large and fast gestures. In general, spline curve can make contact with others objects. This last point is illustrated on figure (2c which is 11 BSplines of 10 control points and 3 constraints per curve). In our system, the mechanical simulation is separated from the visualisation, we thus can indicate the computation time of the mechanical computation for a simulated time of 1 ms.

These results have been obtained on a Pentium 4 1.7GHz processor with 512Mb RAM. The average computation time is 5.7ms for the knot formation (figure 2b), 16.3ms for the (figure 2a) and 16ms for the (figure 2c).

Movies can be found at *http://www.lifl.fr/~lenoir/MS4CMS*.

6. Conclusion

In this paper, we presented a model of surgical thread. This model is based on a spline curve animated by the Lagrangian formalism. Internal constraints are treated by Lagrangian multipliers while external constraints and self-collisions are handled by a penalty method.

The results show that our model is fast enough for interactive time applications, like surgical simulation. The constraints offer many opportunities to the system (fixing any coordinates of any curve's point) and permit us to treat fixed points correctly. This leads to numerically more stable simulations that are interactive. This allows the surgeon to learn correctly the manipulation of a surgical thread.

However, the penalty method for the self-collision is insufficient and makes the knots very breakable. A future could consist in replacing it by more accurate techniques like Lagrangian multipliers. Another drawback is the limited number of degrees of freedom which makes the control of the curve too coarse close to the knot. This could be improved by localy refining the curve and proposing a multiresolution thread simulation.

Currently, our model deals with elasticity without considering the plastic domain. An improvement coul consist in adding this property to the system. Finaly, we could consider another form of energy as the spring cannot bear the thread's weight. In fact, the torsion of the curve should be added since it is an important part of the behaviour. It can be considered with special springs which induce another discrete energy or with cosserat, an energy theory presented in [Pai02], which proposes a continuous energy.

Acknowledgments

References

[Bin87]   T.O. Binford. Generalized cylinders representation. *In S.C. Shapiro, p. 321-323. Wiley, New York*, 1987.

[Blo90]   J. Bloomenthal. Calculation of reference frames along a space curve. *Graphics Gems vol. 1, Academic Press*, 1990.

[DDCB01]   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.

[Del98]   H. Delingette. Towards realistic soft tissue modeling in medical simulation. *Proceedings of the IEEE: Special Issue on Surgery Simulation, pages 512-523*, April 1998.

[DMC02]   J. Davanne, P. Meseure, and C. Chaillou. Stable haptic interaction in a dynamic virtual environment. *IROS'02*, 2002.

[GW97]   A. Van Gelder and J. Wilhelms. Simulation of elastic membranes with triangulated spring meshes. *Journal of Graphics Tools*, July 3, 1997.

[HMC01]   L. Hilde, P. Meseure, and C. Chaillou. A fast implicit integration method for solving dynamic equations of movement. *Proceedings of VRST'01, Banff, Canada*, 15th-17th november 2001.

[JQD+00]   A.C. Jambon, D. Querleu, P. Dubois, C. Chaillou, P. Meseure, S. Karpf, and C. Géron. Spic: Pedagogical simulator for gynecologic laparoscopy. *Proceedings of the 8th Medecine Meets Virtual Reality Conference pp 139-145*, 2000.

[MW88]   M. Moore and J. Wilhelms. Collision detection and response for computer animation. *Computer Graphics*, 1988.

[NR01]   O. Nocent and Y. Remion. Continuous deformation energy for dynamic material splines subject to finite displacements. *Eurographics CAS'2001*, 2001.

[Pai02]   Dinesh K. Pai. Strands: Interactive simulation of thin solids using cosserat models. *Eurographics*, 2002.

[PDA01]   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.

[Pro95]   X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface*, 1995.

[Rem00]   Y. Remion. Animation dynamique : moteur lagrangien généraliste et applications. *Université de Reims, Champagne-Ardenne. Habilitation à diriger des recherches*, december 2000.

[RNG99]   Y. Remion, J.M. Nourrit, and D. Gillard. Dynamic animation of spline like objects. *WSCG*, 1999.

[RNN00]   Y. Remion, J.M. Nourrit, and O. Nocent. Dynamic animation of n-dimensional deformable objects. *WSCG*, 2000.

[TPBF87]   D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, july 1987.