A Suture Model for Surgical Simulation

Julien Lenoir[1], Philippe Meseure[2], Laurent Grisoni[1], and Christophe Chaillou[1]

[1] ALCOVE, INRIA Futurs, IRCICA-LIFL, UMR CNRS 8022, University of Lille 1, 59655 Villeneuve d'Ascq Cedex, FRANCE,
[2] SIC, FRE CNRS 2731, Bât SP2MI, Bld Marie et Pierre Curie, BP 30179, 86962 Futuroscope Cedex, FRANCE
Julien.Lenoir@lifl.fr,
Videos can be found on : http://www.lifl.fr/~lenoir

Abstract. In this paper, we propose a surgical thread model in order for surgeons to practice a suturing task. We first model the thread as a spline animated by continuous mechanics. The suture is simulated via so-called "sliding point" constraints, which allow the spline to move freely while constrained to pass through specific piercing points. The direction of the spline at these points can also be imposed. Moreover, to enhance realism, an adapted model of friction is proposed, which allows the thread to remain fixed at the piercing point or slides through it. Our model yields to good results showing realistic behavior, robust computation and interactive rates.

1 Introduction

Suturing is a fundamental surgical gesture that any practitioner has to acquire and improve. This technique is useful for example during an ablation of an organ or for stitching a wound up. Besides, thanks to the growing power of computers, we are able to offer interactive surgery simulations of realistic yet complex models. Consequently, it seems convenient to use surgical simulators to practice suturing in difficult contexts such as endoscopic surgeries [3]. We therefore want to design a dynamic surgical thread model for interactive simulation with which a surgeon can train the suturing. This model must be computed at 30Hz at least, for good visual effects.

This paper is organized as follows: First, we discuss some previous work in the domain. Then, we explain our basis model and the new constraints we propose for suturing. Finally, we present some results before concluding.

2 Previous Work

Surgical simulation needs both models of organs and interactions. Thus, the tools that the practitioners use and their effects must be modeled too. However, most researches have concentrated on organs simulation only, whereas the modeling of certain tools remains an issue. Among these, the simulation of surgical threads has been studied only recently. The simplest approach to model a thread is to

use a mass/spring chain [10, 4]. This technique is generally used in commercial simulators[1]. To avoid using a high stiffness for the stretching, some models even rely on a chain of rigid links [2]. Pai [9] proposes a static simulation of a curve based on the cosserat theory. Moving a Frenet frame along the thin solid, they obtain a specific energy term measuring stretching and twisting deformation.

Some algorithms have been proposed to handle suturing [4, 2] and knot tying [10] with such models. These however heavily rely on their discrete nature. For suturing, these methods do not result from any physical equations. At each time step, a point of the thread is linked to a point on the organ (considered as the piercing point). The concerned thread point can change to simulate sliding, but this modification is an arbitrary choice. Moreover, if the thread must pass through several piercing points, many thread points must be fixed and it is not clear if the resolution of the model remains stable. What is more, the overall movement results in "step-by-step" sliding. To hide this effect (visually and haptically), the distance separating two successive points on the curve must be small, which induces a fine discretization and a penalized computation time. Moreover, this discretization loses the continuous property of the curve, which could have been kept by other approaches [14].

In a way similar to organ models, we want to design a model of threads, where all physical parameters remain continuous. Since during a suturing task, any point of the curve can potentially be constrained in the pierced holes, it is also desirable that the constraints should apply everywhere along the curve. For that purpose, we propose new types of constraints which allows the thread to slide with frictions through a point in a specific direction. The knot tying is out of the scope of this article and we only focus on sewing.

3 Physical simulation of thread

In this section, we briefly present our model of thread. However, more details can be found in [6]. We model the thread as a spline with few control points:

$$\mathbf{P}(s,t) = \sum_{i=1}^{n} \mathbf{q}_i(t) b_i(s). \tag{1}$$

where s is the parametric abscissa, t the time, $\mathbf{q}_i$ the control point positions, n the number of control points, and b_i the basis functions specific to the spline type. We provide the curve with physical properties such as a continuous mass distribution and animate it using the Lagrange equation of motion. In this model, the coordinates q_i^α with $\alpha \in \{x, y, z\}$ of the control points of the curve are the degrees of freedom.

One of the main interest of such a model, is that all the physical properties are defined in a continuous way. These include external forces or constraints, which can therefore be applied on any point of the curve and not only on the

[1] See Surgical Science web site: http://surgical-science.com

control points (which somehow do not lie on the curve). The generalized form of the forces applied at a point P are expressed as $Q_i^\alpha = \mathbf{F}.\frac{\partial \mathbf{P}}{\partial q_i^\alpha}$ and are thus automatically distributed among the control points. External forces include gravity, viscosity, etc.

To control the deformations of the thread, we have to consider an internal energy that aims at physically structuring the spline. We can choose a set of springs regularly dispatched along the curve. The springs link two points of the curve (not necessarily control points) and are handled via the above-mentioned generalized forces computation. The use of various springs (including rotational springs) enable the control of stretching, bending and twisting. However, for the stretching energy only, a continuous approach exists [8].

To perform the physical constraints needed by a suturing task, we need a method which satisfies at most the constraints with no discontinuity in the physical simulation. Instead of using unstable projection methods [11], we rely on the Lagrange multipliers method for its robustness. We obtain a linear system:

$$\begin{pmatrix} M & 0 & 0 & -L_x^T \\ 0 & M & 0 & -L_y^T \\ 0 & 0 & M & -L_z^T \\ L_x & L_y & L_z & 0 \end{pmatrix} \begin{pmatrix} \mathbf{A_x} \\ \mathbf{A_y} \\ \mathbf{A_z} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{B_x} \\ \mathbf{B_y} \\ \mathbf{B_z} \\ \mathbf{E} \end{pmatrix}. \tag{2}$$

where $L = (L_x L_y L_z)$ is the constraint matrix, $\mathbf{A}$ the acceleration of the degrees of freedom, $\mathbf{B}$ a vector that sums the different contributions of all forces, $\mathbf{E}$ a vector coding the intensity of the violation of the different constraints. λ are the Lagrange multipliers and each of them links a constraint to the degrees of freedom. The three symmetric matrices M form the generalized mass matrix M_g and are computed in the following way (see [6]):

$$\forall (i,j) \in \{1..n\} \times \{1..n\}, \qquad M_{ij} = m. \int_{\mathbb{R}} b_i(s) b_j(s)\, ds. \tag{3}$$

with m the mass of the spline. For n control points and c constraints, the overall system consists in $3n + c$ equations.

4 Constraints for suture simulation

We need several constraints to simulate the suture correctly. The first constraint is a sliding point constraint which allows our spline to pass through a specific point of an organ surface and to slide through it. Moreover, the thread direction on the contact point is controlled to be orthogonal to the organ surface or directed by the needle inserted. This offers a more realistic simulation in which the thread does not turn freely around the contact point. To enhance realism, a local friction model is introduced on the contact point to reproduce both kinetic friction phenomena which brake the slipping and static friction sticking.

4.1 Sliding point constraint

The sliding point constraint allows the spline to slide through a specific point. We can consider that this constraint is similar to a fixed point one with a dynamic abscissa parameter. It derives into three expressions (one for each coordinate) which we note $\mathbf{g}$ and are written as:

$$\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t, s(t)) = \mathbf{P}(s(t), t) - \mathbf{P}_0. \tag{4}$$

The equation $g = 0$ imposes that some point of the spline must be at the position $\mathbf{P}_0$, that we suppose fixed for now. Since it is a dynamic system, $s(t)$ changes over time in order to be the right point of the curve that minimizes the energy of the constrained system.

To avoid the drift due to numerical integration, we use the equations of the constraint $\mathbf{g}$ to formulate a second order differential equation. This provides a solution with a critical damping, known as a specific Baumgarte technique [12, 1]. This leads to the constraint equations:

$$\ddot{\mathbf{g}} + \frac{2}{\Delta t}\dot{\mathbf{g}} + \frac{1}{\Delta t^2}\mathbf{g} = 0. \tag{5}$$

where Δt is the time step of the simulation.

As s becomes a dynamic parameter, it also becomes a new unknown of the system which require a new equation. If we consider a perfect constraint, the Lagrange theory imposes that the virtual power of the strain due to this constraint must be equal to zero. This is written as:

$$\lambda . \frac{\partial \mathbf{g}}{\partial s} = 0. \tag{6}$$

This theoretical framework is explained in more details in [13].

We decide to consider a different equation which gives us a direct relation between s and λ to accelerate the resolution process. We allow that the effective work of the force generated by the Lagrange multipliers is not null and we represent it as an error. This error is due to an incorrect value of s and is applied to correct the dynamics of this parameter. This approach gives us an equation slightly different from Eq. 6:

$$\epsilon . \ddot{s} + \lambda . \frac{\partial \mathbf{g}}{\partial s} = 0. \tag{7}$$

where the factor ϵ is close to zero. The system becomes:

$$\begin{pmatrix} M & 0 & 0 & 0 & -L_x^T \\ 0 & M & 0 & 0 & -L_y^T \\ 0 & 0 & M & 0 & -L_z^T \\ 0 & 0 & 0 & \epsilon & -L_s^T \\ L_x & L_y & L_z & L_s & 0 \end{pmatrix} \begin{pmatrix} \mathbf{A^x} \\ \mathbf{A^y} \\ \mathbf{A^z} \\ \ddot{s} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{B^x} \\ \mathbf{B^y} \\ \mathbf{B^z} \\ 0 \\ \mathbf{E} \end{pmatrix}.$$

We decide to solve the system by decomposing the acceleration in two parts, one for *tendency* and another for *correction*: $\mathbf{A} = \mathbf{A}_t + \mathbf{A}_c$ [5]. The acceleration of tendency represents the acceleration without any constraint and the other acceleration is the correction due to the constraints. This leads us to this new equation system:

$$\begin{cases} M\mathbf{A}_t^x = \mathbf{B}^x \\ M\mathbf{A}_t^y = \mathbf{B}^y \\ M\mathbf{A}_t^z = \mathbf{B}^z \\ M_g\mathbf{A}_c = L^T\lambda \\ \epsilon\ddot{s} = L_s^T\lambda \\ L(\mathbf{A}_t + \mathbf{A}_c) + L_s\ddot{s} = \mathbf{E} \end{cases}$$

We replace the terms $\mathbf{A}_c$ and $\ddot{s}$ in the sixth equation by their expression respectively in the fourth equation and the fifth equation. These replacements yield an equation for λ:

$$\begin{cases} M\mathbf{A}_t^x = \mathbf{B}^x \\ M\mathbf{A}_t^y = \mathbf{B}^y \\ M\mathbf{A}_t^z = \mathbf{B}^z \\ \mathbf{A}_c = M_g^{-1}L^T\lambda \\ \epsilon\ddot{s} = Ls^T\lambda \\ LM_g^{-1}L^T\lambda + \frac{L_sL_s^T}{\epsilon}\lambda = \mathbf{E} - L\mathbf{A}_t \end{cases}$$

To simulate a complete suturing task, the spline must pass through several piercing points P_0^i which implies several new variables s^i. In practice, the algorithm works well and our system tolerates many sliding point constraints. The system is solved in $\mathcal{O}(c.n^2 + c^2.n + c_g.c^2)$ where c_g is the number of unknown s_i.

With only such constraints, the spline can freely slide through but also turn around all the piercing points without considering the organ surface. It appears necessary to avoid an inversion of the insertion type (that is, penetration in or exit out of the organ). For that purpose, it is convenient to design a new constraint which would insure the right direction of the piercing.

4.2 Sliding direction constraint

We need a specific constraint to impose at a sliding point that the spline is orthogonal to the surface of the organ. We thus define a direction constraint linked to a sliding point constraint. We just need the wanted vector direction $\mathbf{T}_0$ and the sliding point $\mathbf{P}(s(t), t)$ on which the direction is imposed. We first determine a local frame $(\mathbf{P}(s(t), t); \mathbf{T}_0, \mathbf{u}, \mathbf{v})$ by computing $\mathbf{u}$ and $\mathbf{v}$ (vectors supporting the local plane tangent to the organ at $\mathbf{P}$). We create a constraint that forces the direction of the sliding point to be orthogonal to $\mathbf{u}$. Repeating this process on $\mathbf{v}$, we constrain the direction of the sliding point to be in one direction orthogonal to $\mathbf{u}$ and $\mathbf{v}$, it is thus in the direction $\mathbf{T}_0$:

$$c_1(\mathbf{q}, \dot{\mathbf{q}}, t, s(t), \mathbf{u}) = \frac{\partial \mathbf{P}}{\partial s}(s(t), t).\mathbf{u} = 0. \quad (8)$$

$$c_2(\mathbf{q}, \dot{\mathbf{q}}, t, s(t), \mathbf{v}) = \frac{\partial \mathbf{P}}{\partial s}(s(t), t).\mathbf{v} = 0. \quad (9)$$

We constrain the direction of the tangent $\mathbf{T}(s,t) = \frac{\partial \mathbf{P}}{\partial s}(s(t),t)$ but we let the intensity of this tangent free. Therefore, its norm will be set correctly by the energy minimization of the Lagrange equations.

We still have a simulation of a thread that can freely slides through a point of an organ and in a specific direction. However, we now need to control the sliding. Frictions appear to be the most physically correct approach.

4.3 Friction on sliding point constraint

At the point $\mathbf{P}(s,t)$, we compute the velocity of the point $\mathbf{V} = \frac{d\mathbf{P}}{dt}$ and its local tangente $\mathbf{T} = \frac{\partial \mathbf{P}}{\partial s}$. For any vector $\mathbf{x}$ (force or velocity), we can express its tangential component $\mathbf{x}_t = (\mathbf{x}.\mathbf{T})\mathbf{T}$ and its normal component $\mathbf{x}_n = \mathbf{x} - \mathbf{x}_t$.

To determine which friction model (static or kinetic) should be applied, we just have a check at the tangential velocity at the sliding point $\mathbf{V}_t$.

- **static case:** If the velocity $\mathbf{V}_t$ is below a given threshold, it is considered null, and we suppose that the frictions are static. In other words, the point is not supposed to move. Instead of computing the friction force which cancels the tangential forces (which would require a post-processing), we choose to replace the sliding point constraint by a fixed point one. The resolution of the system (cf. Eq. 8) gives the value of the Lagrange multipliers, which are related to the intensity of the forces which enforce the constraint. To respect the Coulomb friction model, we must check that the friction forces are in the friction cone. We compute both friction (λ_t) and constraint (λ_n) forces, and check if $||\lambda_\mathbf{t}|| < \mu_s||\lambda_\mathbf{n}||$, where μ_s is the static friction coefficient. If the test succeeds, the static friction hypothesis holds and the computation is over. Otherwise, the frictions are pseudo-static. In that case, the constraint is replaced by a sliding point constraint. We keep the normal component λ_n of the constraint force, and add a tangential force to the system:

$$F_{friction} = -\mu_s||\lambda_n||\frac{\lambda_t}{||\lambda_t||}$$

- **dynamic case:** If the velocity V_t is not null, the kinetic friction model is immediately applied. The system defined with the sliding point constraint without friction is solved. The Lagrange multipliers give the normal force λ_n. We then compute the kinetic friction force, based on the kinetic friction coefficient μ_k:

$$F_{friction} = -\mu_k||\lambda_n||.\frac{\mathbf{V_t}}{||\mathbf{V_t}||}.$$

This algorithm is applied to all the piercing points. We have been able to check if static, pseudo-static or kinetic frictions were to apply at each point. If frictions are static everywhere, the computation of the forces is over. If however one or more points are in the pseudo-static or kinetic case, the computed friction forces have to be injected into the system which is solved again (to allow the friction forces to be dispatched to the degrees of freedom of the curve). Thus, our algorithm generally requires two solving passes of the system of Eq. 8.

It can be noticed that, in the kinetic case, we start the computation step by considering a sliding point constraint which is biased by ϵ. Theoretically, the Lagrange multipliers give us a force orthogonal to the local tangent (i.e. $\lambda_t = 0$) due to Eq. 6. However, by considering Eq. 7, we allow the constraint to work. We thus compute the effective normal force by projection and the non-null tangential force is just considered as a supplementary force applied to the thread (like deformation forces, collisions...).

4.4 Interaction between the thread and the organ

Since the beginning of this section, we have always considered that the points P_0^i of the organ were static. To take the motion and deformation of the organ into account, a quasi-static approach is possible. We consider $\mathbf{P}_0^i$ as constant for a given time step (this assumption is reasonable, since the motions are small). After all the sliding points $\mathbf{P}_0^i$ have moved, the sliding point constraints may be violated, but the stabilization scheme of Eq. 5 attracts the curve toward all the points. For the suture to modify the movement and deformation of the organ, we use the action/reaction law. At each $\mathbf{P}_0^i$, the normal and friction forces are inverted and applied to the organ.

A more precise and robust approach consists in assembling the thread and the organ in a same system [5]. $(\mathbf{P}_0^i, \mathbf{u}^i, \mathbf{v}^i)$ are then considered as variables of the organ. The constraint equations Eq. 4 and Eq. 8 can then differentiate the expression of $(\mathbf{P}_0^i,\ \mathbf{u}^i,\ \mathbf{v}^i)$ according to the motion of the organ.

5 Results

Our implementation takes place in a framework of surgical simulators [7] offering tools for collision process and self collision detection. This platform also gives access to a large choice of numerical integration methods which are necessary to compute the motion of our model.

The simulations were performed on a 1.7GHz Pentium IV processor with 512 MB of memory and an implicit Euler numerical integration was used. The test (figure 1) simulates a suture controlled by a needle. We use a spline with 20 control points and two piercing points. The first figure presents the initial situation and the second one shows the simulation at some time later when the user pulled the needle. We can see that the thread has slid through the two piercing points and it is directed normally to the organ surface. The simulation takes about 30 ms which implies a simulation frequency of 34Hz.

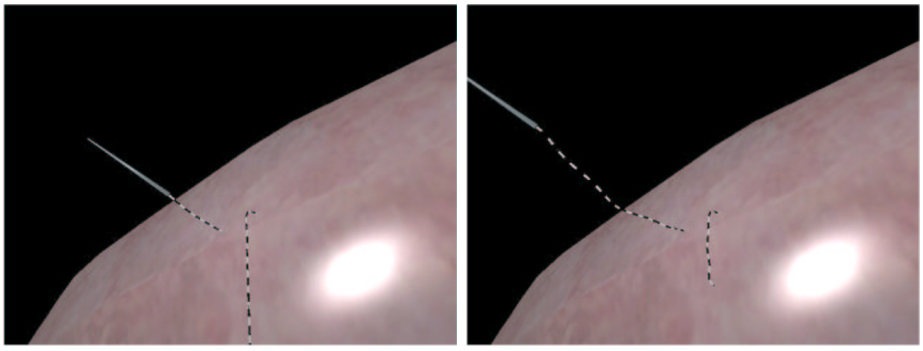

Fig. 1. Suturing task with a thread and a needle. The figure on the left shows a thread sliding along two piercing points. The figure on the right shows the state of the thread after the user has pulled it.

The spline interpolation allows the use of a limited number of control points. However, during a suture, it is important to give enough degrees of freedom for the curve to deal with all the constraints. We are currently working on a multi-resolution method which can not be described here due to space limitation. This method can locally increase the number of control points while removing others in low curvature areas. In our first experiment, we see that, even for complex geometric configurations (such as a knot), very few degrees of freedom are added and the number of control points remains quite the same. This shows that 20 control points is in practice sufficient.

6 Conclusion and Perspectives

We propose in this paper a dynamic model of surgical threads with specific constraints permitting the suturing task on objects like organs. These constraints are enabled to force the thread to pass through a sliding point, to impose the direction of the thread at this point, and to integrate frictions between the thread and the organ.

Nevertheless the complete suturing simulation still requires a lot of work. First, it is essential to identify our model to the properties of existing threads to get a realistic simulation. For such purpose, the adaptation of the Cosserat energy seems inescapable [9]. Besides, the knot tying was out of the scope of this article. It requires the multi-resolution process that we have mentioned above. It is still under development but show promising results. In this context, we are also working on the cutting of the thread. We are confident that our thread model will become a powerful tool for future simulators.

Acknowledgment

The authors would like to thanks Y. Rémion (LERI, Reims) for its precious help on the dynamic splines formalism and the free-variable constraints resolution.

References

1. J. Baumgarte. Stabilization of constraints and integrals of motion. *Computer Meth. Appl. Mech. Eng.*, 1:1–16, 1972.
2. J. Brown, K. Montgomery, J.-C. Latombe, and M. Stephanides. A microsurgery simulation system. In *MICCAI01*, Utrecht, October 2001.
3. H. Delingette. Towards realistic soft tissue modeling in medical simulation. *Proceedings of the IEEE, pages 512-523*, April 1998.
4. M. LeDuc, S. Payandeh, and J. Dill. Toward modeling of a suturing task. In *Graphics Interface'03 Conference*, pages 273–279, Halifax, June 2003.
5. J. Lenoir and S. Fontenau. Mixing deformable and rigid-body mechanics simulation. In *Computer Graphics International*, Crete, June 2004.
6. J. Lenoir, P. Meseure, L. Grisoni, and C. Chaillou. Surgical thread simulation. In *Modelling and Simulation for Computer-aided Medecine and Surgery (MS4CMS)*, volume 12, pages 102–107, Rocquencourt, November 2002. INRIA, EDP Sciences.
7. P. Meseure, J. Davanne, L. Hilde, J. Lenoir, L. France, F. Triquet, and C. Chaillou. A physically-based virtual environment dedicated to surgical simulation. In N. Ayache and H. Delingette, editors, *International Symposium on Surgery Simulation and Soft Tissue Modeling (IS4TM)*, volume 2673 of *Springer Verlag LNCS*, pages 38–47, Juan–les–Pins, June 2003.
8. O. Nocent and Y. Rémion. Continuous deformation energy for dynamic material splines subject to finite displacements. In *EUROGRAPHICS workshop on Computer Animation and Simulation*, pages 87–97, Manchester (UK), September 2001.
9. D. Pai. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum (Proceedings of EUROGRAPHICS'02)*, 21(3), September 2002.
10. J. Phillips, A. Ladd, and L. Kavraki. Simulated knot tying. In *IEEE International Conference on Robotics and Automation*, pages 841–846, Washington, May 2002.
11. J.C. Platt and A.H. Barr. Constraint methods for flexible models. *Computer Graphics (Proceedings of ACM SIGGRAPH 88)*, 22(4):279–288, August 1988.
12. Y. Rémion, J.M. Nourrit, and D. Gillard. Dynamic animation of spline like objects. In *WSCG'1999 Conference*, pages 426–432, Plzen, February 1999.
13. Yannick Rémion. Prise en compte de "contraintes à variables libres". Technical Report 03-02-01, LERI, February 2003.
14. D. Terzopoulos, J. Platt, A. Barr, and K. Fleisher. Elastically deformable models. *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, 21(4):205–214, July 1987.