

Knot-tying with Visual and Force Feedback for VR Laparoscopic Training

F. Wang, E. Burdet, R. Vuillemin and H. Bleuler

Abstract—Real-time simulation of thread and knot-tying with visual and force feedback is an essential part of virtual reality (VR) laparoscopic training. This paper presents a physics-based thread simulator that enables realistic knot tying at haptic rendering rate. The virtual thread follows Newton’s law and behaves naturally. The model considers main mechanical properties of the real thread such as stretching, compressing, bending and twisting, as well as contact forces due to self-collision and interaction with the environment, and the effect of gravity. The structure of the system has essential advantages over geometrically based approaches, as was illustrated in an implementation on the Xitact simulator.

I. INTRODUCTION

During the past decade, patients have benefited from significant advantages of minimal invasive surgery such as reduced pain and shortened recovery time. The most important change has been laparoscopy, which comprises advanced videolaparoscopes and redesigned surgical instruments. As a new modality, laparoscopy requires surgeons to manipulate the tools with limited range of motion while looking at the monitor in another direction. Laparoscopic procedures also require the operator to adapt to amplified tremor, diminished tactile sensation, loss of depth perception and proprioceptive-visual conflict issues from manipulating the elongated instruments with port fixation.

Videoscopic surgery requires ambidexterity, hand-eye coordination, and depth perception due to the disorientating, two-dimensional working environment. Conventional training methods for open surgery, which are trained on mocks, animals or cadavers, and by observing and assisting experienced colleagues, do not emphasize the acquisition of such skills. Furthermore, for safety, cost and ethical reasons (less training with animals), one cannot try all possible strategies. Due to the nature of laparoscopy and recent advances in computer technology, combined with the need for shortening learning curves in surgery training, it is expected that Virtual Reality (VR) simulators will enable more effective and systematic training, provide objective assessment of technical competence and facilitate the teaching of rare cases [1]. An increasing number of studies show that the basic

skills achieved by training with a VR laparoscopic simulator transfer to the operating room [2], [3], [4].

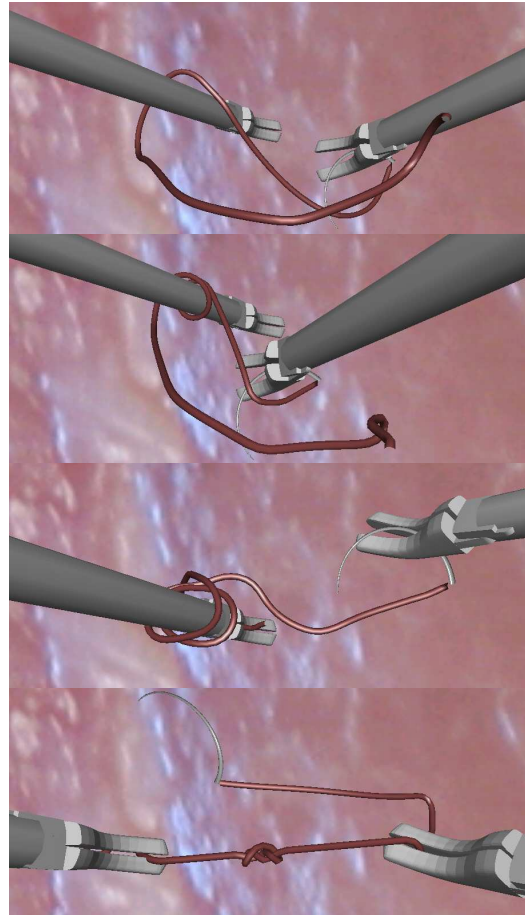


Fig. 1. Key steps in the tying of a surgical knot with visual and force feedback provided by a Xitact laparoscopy simulator

A major skill of (laparoscopic) surgery is knot tying, a bimanual task requiring skilful control of the nylon and knowledge of the correct topology [5]. Despite the introduction of mechanical means of tissue fixation such as staples and clips, it remains an essential component of surgical procedures.

The thread can be simulated visually using a kinematic scheme such as “follow the leader” (*e.g.*, [6]), which works well in tissue where the thread follows the needle. However, it gives a false idea of the dynamics involved in manipulation of real nylon threads in air. Nylon thread manipulation exploits bending and twist, which cannot be realized using kinematic schemes. The motion can even look unnatural,

This work was supported by the University of Applied Sciences of Central Switzerland (HTA Luzern) and the National University of Singapore

F. Wang and R. Vuillemin are with Institute of Electronics, HTA Luzern, Technikumstrasse 21, CH-6048 Horw, Switzerland fwang@ieee.org, rvuillemin@hta.fhz.ch

E. Burdet is with Imperial College London, United Kingdom e.burdet@imperial.ac.uk

H. Bleuler is with the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland hannes.bleuler@epfl.ch

with the whole thread stopping as soon as the needle stops, even when the whole thread has not yet reached equilibrium. Knot-tying training procedures integrated into commercial laparoscopic simulators, e.g. [7], seem to use a similar kinematic approach.

A technique for thread simulation based on dynamics is presented in [8], which simulated knot tying with a spline of linear springs and control points at their ends. The focus was on handling self-collisions when the knot is being pulled and tied, while conserving energy, mass, and momentum of the system. This model does not contain bending or twisting force, and it seems to be far from suitable for real-time simulation [9]. Conservation of energy assumption may be inappropriate: real thread dynamics are dissipative, coming quickly to equilibrium after an impulse.

A haptic virtual thread for surgery training with force feedback needs to produce dynamics at an update at 500Hz. While the typical force level in microsurgery is below perception, experiencing the thread dynamics using scaled force as feedback may contribute to efficient learning. In particular, we want to test learning with haptic cues, which has been proven to provide efficient training in arm movements [10], [11] and robot-assisted rehabilitation [12].

In this context, this paper describes a novel approach for simulation of thread and knot-tying with real-time visual and force feedback (Fig. 1). This approach is based on second order dynamics, using physically plausible stretch/compression, bending, twist, friction, gravity and contact forces. At each time step the displacements of the parts grasped by the interacting tools are integrated into the thread dynamics, resulting forces computed at each node and the motion is integrated. Geometric constraints, in particular self-collisions, are expressed as forces within this single dynamics framework.

II. MODEL DYNAMICS

A. Thread Dynamics

The topological structure of the thread, shown in Fig. 2, is represented by a list of $K + 1$ nodes $i \in \{0, 1, \dots, K\}$ at positions $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)$ and at torsion angles q_i connected by K links with length l_i , $i \in \{1, 2, \dots, K\}$. Let \mathbf{u}_i represent the vectors pointing from node $i - 1$ to i .

A mass m_i and an inertia momentum I_i are assigned to each node i . A group of (massless) springs attached to each link produce stretching, compressing, bending and twisting between and on the nodes. The dynamics of each node i is

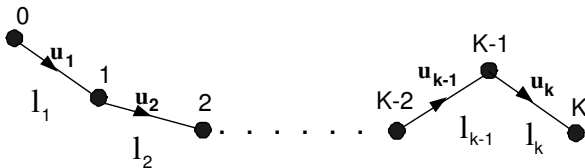


Fig. 2. Structure of the thread

determined by:

$$\begin{aligned} \mathbf{F}_i &= m_i \cdot \ddot{\mathbf{x}}_i(t) \\ \tau_i &= I_i \cdot \ddot{q}_i(t) \end{aligned} \quad (1)$$

where \mathbf{F}_i is the sum of external forces applied on mass m_i and τ_i the torsion force applied along the thread. The motion of each node is found by Euler integration:

$$\begin{aligned} \dot{\mathbf{x}}_i(t + dt) &= \dot{\mathbf{x}}_i(t) + \ddot{\mathbf{x}}_i(t) \cdot dt \\ \dot{q}_i(t + dt) &= \dot{q}_i(t) + \ddot{q}_i(t) \cdot dt \end{aligned}$$

$$\begin{aligned} \mathbf{x}_i(t + dt) &= \mathbf{x}_i(t) + \dot{\mathbf{x}}_i(t + dt) \cdot dt \\ q_i(t + dt) &= q_i(t) + \dot{q}_i(t + dt) \cdot dt \end{aligned} \quad (2)$$

In Eq. (1), \mathbf{F}_i is the sum of external forces applied on mass m_i :

$$\mathbf{F}_i = \mathbf{F}_i^s + \mathbf{F}_i^b + \mathbf{F}_i^t + \mathbf{F}_i^r + \mathbf{F}_i^c + m_i \mathbf{g} \quad (3)$$

where \mathbf{F}_i^s is the force resulting from stretching and compression of the spring links connected to node i , \mathbf{F}_i^b and \mathbf{F}_i^t are the forces resulting from bending and twisting of the link, \mathbf{F}_i^r is the dissipative friction force, \mathbf{F}_i^c is the contact force with the environment or between different parts of the thread due to self-collision, and $m_i \mathbf{g}$ is the gravitation force. Torsion force is due to stretch and friction:

$$\tau_i = \tau_i^s + \tau_i^r \quad (4)$$

We now describe how these forces are calculated.

Stretching and compression force F^s

All nodes undergo stretching and compression of the springs bound to them according to:

$$\mathbf{F}_{i-1}^s = \kappa_i^s \cdot \Delta_i \cdot \mathbf{u}_i - \kappa_{i-1}^s \cdot \Delta_{i-1} \cdot \mathbf{u}_{i-1} \quad (5)$$

where $\Delta_i = l_i - r_i$ is the difference between a link's current length and resting length, and κ_i^s is its stiffness constant.

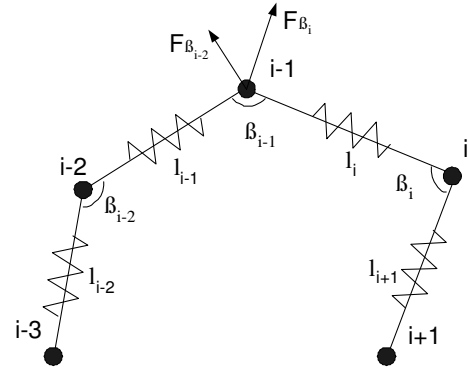


Fig. 3. Bending force at node i .

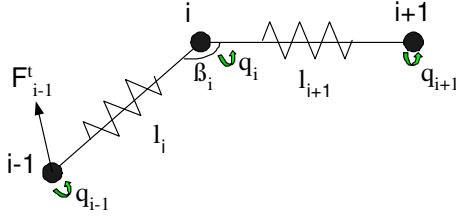


Fig. 4. Twisting effect

Bending force F^b

Fig. 3 illustrates the bending force \mathbf{F}_{i-1}^b computed from two angular springs at nodes $i-2$ and i , given by:

$$\begin{aligned}\mathbf{F}_{i-1}^b &= \mathbf{F}_{\beta_{i-2}} + \mathbf{F}_{\beta_i}, \\ \mathbf{F}_{\beta_{i-2}} &= \kappa_{i-2}^b \cdot (\pi - \beta_{i-2}) \cdot \mathbf{u}_{i-1} \times (\mathbf{u}_{i-2} \times \mathbf{u}_{i-1}) \\ \mathbf{F}_{\beta_i} &= \kappa_i^b \cdot (\pi - \beta_i) \cdot \mathbf{u}_i \times (\mathbf{u}_i \times \mathbf{u}_{i+1})\end{aligned}\quad (6)$$

where κ_{i-2}^b (for $\mathbf{F}_{\beta_{i-2}}$) is the stiffness constant of the angular spring, the bending angle is $(\pi - \beta_{i-2})$, and the orientation of the bending force is specified by $\mathbf{u}_{i-1} \times (\mathbf{u}_{i-2} \times \mathbf{u}_{i-1})$.

Torsion τ^s and twisting force F^t

In Fig. 4, an angular spring is associated to each spring link, producing the torque

$$\tau_{i-1}^s = \kappa_{i-1}^{T_1} \cdot (q_i - q_{i-1}) \quad (7)$$

Torsion of link $i+1$ produces a twisting force \mathbf{F}_{i-1}^t on the mass m_{i-1} given by

$$\mathbf{F}_{i-1}^t = \kappa_{i+1}^t \cdot (q_{i+1} - q_i) \cdot \sin(\pi - \beta_i) \cdot (\mathbf{u}_{i+1} \times \mathbf{u}_i) \quad (8)$$

where κ_{i+1}^t is the stiffness constant of the angular spring, $(q_{i+1} - q_i)$ is the torsion angle of spring link L_{i+1} and $(\mathbf{u}_i \times \mathbf{u}_{i+1})$ gives the orientation vector of the twisting force acted on mass m_{i-1} .

Damping (F^r, τ^r)

Dissipation enters the system in the form of friction forces

$$\begin{aligned}\mathbf{F}_{i-1}^r &= -\kappa^{ex} \cdot \dot{\mathbf{x}}_{i-1}(t) - \kappa^{rx} \cdot (\dot{\mathbf{x}}_i(t) - \dot{\mathbf{x}}_{i-2}(t)) \\ \tau_{i-1}^r &= -\kappa^{eq} \cdot \dot{q}_{i-1}(t) - \kappa^{rq} \cdot (\dot{q}_i(t) - \dot{q}_{i-2}(t))\end{aligned}\quad (9)$$

Internal dynamical damping is through κ^{rx} and κ^{rq} , while κ^{ex} and κ^{eq} represent contact with the environment.

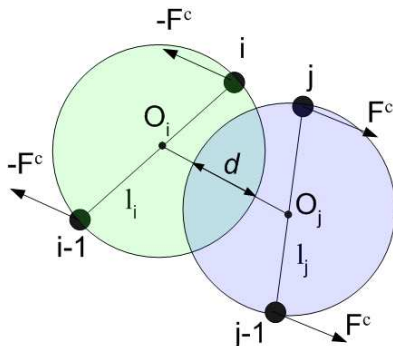


Fig. 5. Reaction force in self-collision

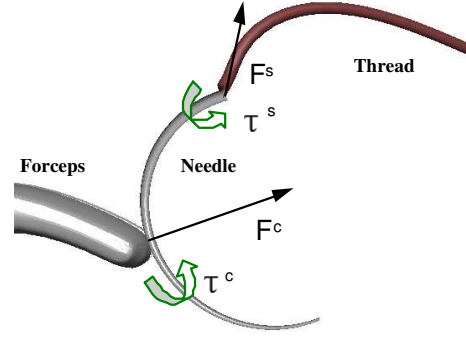


Fig. 6. Forces applied on the needle

Contact force F_i^c

Contact of the thread with itself or other objects such as surgery tools or skin is handled through a repelling force depending on separation. In self-collision, each link is bounded by a spherical volume (Fig. 5). The reaction force between two colliding links, acting on the four neighboring nodes, is computed as

$$\mathbf{F}_i^c = \kappa^c e^{\kappa^o d} \cdot \mathbf{o}_{ij} \quad (10)$$

where \mathbf{o}_{ij} is a united vector pointing from O_i to O_j . Section III describes how contact points are detected. For self-collisions of the thread, the distance d between two links in contact is determined from the penetration of the associated spheres, as shown in Fig. 5.

B. Needle Dynamics

The needle is modelled as an additional node with mass m and inertia momentum τ (Fig. 6). The needle end is connected to the first node of the thread, such that the needle dynamics are influenced by the force and torque from the thread. The dynamics of the needle is determined by:

$$\begin{aligned}\mathbf{F} &= \mathbf{F}^s + \mathbf{F}^r + \mathbf{F}^c + m\mathbf{g} \\ \tau &= \tau^s + \tau^r + \tau^c\end{aligned}\quad (11)$$

The total force \mathbf{F} applied on needle is composed of stretching force from thread \mathbf{F}^s , friction \mathbf{F}^r , the force \mathbf{F}^c in reaction to contact with the environment, and gravity. The total torsion τ is due to torsion transferred from thread τ^s , friction τ^r and torsion imposed by external force τ^c .

Grasping involves the interaction between the tool tips and needle. However it is difficult to reach the equilibrium in grasping since the needle has an irregular and solid shape. The needle will be bounced between the tool tip and finally jump out when grasped. Here we set the needle state to be "grasped" as soon as it is in contact with both inner sides of the tool tip, then it will be translated and rotated by the contacting tool until it is released.

III. DETECTION OF COLLISION

Collision detection is a key component in 3D virtual reality, with extensive research devoted to checking collisions between rigid objects. Less effort has addressed collisions

between deformable objects, especially self-collisions of a deformable object. In surgical simulation, a suture often collides with itself as well as other objects (needle, forceps, organs). Every collision must be detected at every time step, so that the reaction force (10) can be correctly set to prevent the objects from passing through.

The bounding-volume hierarchy method (BVH) [6], [13], [14] constructs a hierarchical bounding representation based on spheres. The key to its efficiency is a simple search routine that uses the bounding representation to ignore most possible pairs of components outside the bound. It is best at representing geometric proximity inside an object at various levels of detail. However, geometric proximity is not invariant in a deforming object, and constant rebuilding of a BVH to maintain it can be very costly.

As the linking sequence of our thread model is fixed, we can build a thread BVH with an invariant topology so that when the thread deforms only the locations and volumes of each bounding sphere need to be updated. As in Fig. 5, one can use minimal spheres enclosing each two-node link and representing the volume of the thread pieces. The spheres are then input to the BVH as its leaves (the lowest level), after which it can be built bottom-up by bounding pairs of successive spheres to new spheres at each higher level.

The resulting BVH is a binary tree. Each intermediate sphere tightly bounds its two children and also encloses all the leaf spheres below it. The root sphere encloses the entire thread and all the bounding spheres. To find the collision between root spheres T_1 and T_2 , we use a recursive algorithm defined as follows:

```

Algorithm FindCollision ( $T_1, T_2$ )
If ( $T_1$  and  $T_2$  have null intersection)
    Return;
Else
    If ( $T_1$  and  $T_2$  are both leaf spheres)
        If ( $T_1$  and  $T_2$  are neither same one nor neighbor)
            Save  $T_1$  and  $T_2$  as a collision pair;
        Return;
    If ( $T_1$ 's level higher than  $T_2$ 's)
        Switch  $T_1$  and  $T_2$ ;
    FindCollision ( $T_1, T_2 \rightarrow \text{left\_item}$ );
    FindCollision ( $T_1, T_2 \rightarrow \text{right\_item}$ );
    Return;

```

To find self-collision in the thread with root sphere T , we use function **FindCollision**(T, T) after updating the BVH tree. Then all the collision pairs are stored in a list, which will be used to set the reaction force as described in Fig. 5. An arbitrary object can be prepared and loaded into a balanced binary tree by ‘‘Adaptive Medial-Axis Approximation’’ [15]. The BVH topology is pre-computed once at the beginning of the simulation and remains fixed. Each simulation loop updates only the center and radius of the bounding spheres, taking $O(n)$ for each loop since it is done bottom up. (The BVH method takes $O(n^{\frac{4}{3}})$ at each query in the worst case [14].) This is only slightly less efficient than the grid method in detecting self-collisions [16]. It is very efficient

on average, especially when other objects are involved.

IV. ALGORITHM IMPLEMENTATION

The thread dynamics are computed at each time step while the laparoscopic tools are manipulated. Let C denote the finite set of all nodes by which the operator is controlling the thread or needle. For example, if one moves the thread by grasping one end, C contains only this end node; in knot tying, where the surgeon is typically grasping two ends (of threads emerging from tissue) with surgical tools, C is composed of these two nodes. The poses of such control nodes are determined by measurement with the VR laparoscopic devices, so only the poses of the other nodes $\bar{C} = \{0, 1, 2, \dots, K+1\} - C$ need to be computed by the algorithm. This is performed in the following steps:

- 1) Read the position and orientation of all the control nodes C from the laparoscopic device.
- 2) Compute the total external force and torque applied on each node. This force can be fed back to control nodes through the haptic device.
- 3) Read the time step dt between the two consecutive loops and integrate the position and torsion angle at all nodes.

The second step, corresponding to (3,4), computes the force at each node from the novel pose $(\mathbf{x}_i, q_i)(t+dt)$ at control node $i \in C$ and old pose $(\mathbf{x}_i, q_i)(t)$ at other nodes $i \in \bar{C}$. It computes the forces at each node using the forces defined in section II:

$$(\mathbf{F}_{i-1}, \boldsymbol{\tau}_{i-1}) = \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i-2}, \mathbf{x}_{i-3}, q_{i+1}, q_i, q_{i-1}) \quad (12)$$

The third step, corresponding to (1,2), integrates the motion equation at each node $i \in \bar{C}$:

$$(\mathbf{x}_i, q_i)(t+dt) = \mathbf{p}(\mathbf{F}_i, \boldsymbol{\tau}_i) \quad (13)$$

A 120-node virtual thread was implemented on a desktop with 3.0G P4 CPU. Xitact LS 500 laparoscopy simulator [17] was used to simulate the surgical tools. The dynamics of the system was updated at 500Hz. There is no lag noticed while manipulating the simulator. Force feedback is transferred back to the simulator through the contacting node of the thread or the needle, which is simply the total force and torque applied on the contacting nodes. As Fig. 1 shows, the simulation provides realistic knot tying to help surgeons in training to tie knots.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a physics-based knot-tying model that updates at haptic rendering rate. The model follows Newton’s laws and considers main properties of the real thread, such as stretching, compressing, bending and twisting, as well as contact forces due to self-collision and interaction with the surgery tools.

The simulator integrates internal and external forces within a single framework, overcomes limitations of a non-physical simulation systems [18] while supporting force feedback, the option of haptic cues for learning, and more realistic visual display with dynamic effect. With this approach, collision

management becomes simple and computationally efficient, obviating the need to consider geometry when a knot is tightened.

We will add the suturing simulation into the framework, which include the interaction among the needle, thread and soft tissues. Other forces will be investigated, for example surface tension holding the thread to a wet surface, often relevant in real surgery and particularly important in microsurgery.

ACKNOWLEDGMENTS

We thank Lindo Duratti, Roger Gassert and the staff of Xitact for technical support with the implementation on the hardware.

REFERENCES

- [1] Coleman J, NDuka CC, Darzi A (1994) Virtual reality and laparoscopic surgery. *Br J Surg* 81: 1709C11.
- [2] Seymour N, Gallagher AG, OBrienM, Roman S, Andersen D, Satava RM. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Ann Surg.* 2002;236:458-64.
- [3] Hyltander A, Liljegren E, Rhodin PH, Lonroth H: The transfer of basic skills learned in alaparoscopic simulator to the operating room. *Surg Endosc.* 2002 Sep. 16(9):1324-28.
- [4] Grantcharov TP, Kristianson VB, Bendix J, Bardram L, Rosenerg J, Funch-Jensen P. Randomized clinical trial of virtual reality simulation for laparoscopic skills training. *Br J Surg.* 2004;91:146-50.
- [5] Mori T, Hatano N, Maruyama S, Atomi Y: Significance of 'hands-on training' in laparoscopic surgery. *Surg Endosc* 1998;12:256-260.
- [6] J Brown, J-C Latombe and Kevin Montgomery (2004), Real-time Knot Tying Simulation, *Visual Computer* 20(2,3):165-79.
- [7] <http://www.immersion.com/medical/products/laparoscopy/modules/basic.php>
- [8] J Phillips, A Ladd and LE Kavraki (2002), Simulated Knot Tying, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*841-6.
- [9] http://www.cs.duke.edu/~jeffp/physical_simulation.html
- [10] E Burdet, R Osu, DW Franklin, T Milner and M Kawato (2001), The central nervous system stabilizes unstable dynamics by learning optimal impedance, *Nature* 414:446-449.
- [11] R. Osu, E. Burdet, D. W. Franklin, T. E. Milner, and M. Kawato, Different Mechanisms Involved in Adaptation to Stable and Unstable Dynamics *J Neurophysiol*, November 1, 2003;90(5):3255-69.
- [12] Reinkensmeyer DJ, Emken JL, Cramer SC, Robotics, motor learning, and neurologic recovery. *Annu Rev Biomed Eng.* 2004;6:497-525. Review.
- [13] Lotan, I., Schwarzer, F., Halperin, D., and Latombe, J.C. (2002), Efficient maintenance and self-collision testing for kinematic chains. *Proc. 18th ACM Symposium on Computational Geometry*, 43-52.
- [14] L. Guibas, A. Nguyen, D. Russel, L. Zhang (2002), Collision detection for deforming necklaces, *Proc. ACM symposium on computational geometry*, 33-42.
- [15] G Bradshaw, and C Sullivan (2004), Adaptive Medial-Axis Approximation for Sphere-Tree Construction, *ACM Transactions on Graphics* 23(1):1-26.
- [16] Overmars, M.H., Halperin, D. (1998). Spheres, molecules, and hidden surface removal. *Computational Geometry: Theory and Applications* 11, 83-102.
- [17] <http://www.xitact.com/>
- [18] F. Wang, E. Burdet, A. Dhanik, T. Poston, C.L. Teo (2005), Dynamic Thread for Real-Time Knot-Tying. *Proc. First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'05)*, 507-8.