Virtual Reality (2010) 14:241–253
DOI 10.1007/s10055-010-0174-6

ORIGINAL ARTICLE

Interactive multi-modal suturing

Shahram Payandeh · Fuhan Shi

Received: 28 April 2010 / Accepted: 23 September 2010 / Published online: 21 October 2010

Abstract We present a mechanics-based interactive multi-modal environment designed as part of a serious gaming platform. The specific objectives are to teach basic suturing and knotting techniques for simple skin or soft tissue wound closure. The pre-wound suturing target, skin, or deformable tissue is modeled as a modified mass-spring system. The suturing material is designed as a mechanics-based deformable linear object. Tools involved in a typical suturing procedures are also simulated. Collision management modules between the soft tissue and the needle, the soft tissue and the suture are analyzed. In addition to modeling the interactive environment of a typical suturing procedure, basics of the modeling approaches on the evaluation of a stitch formed by the user are also discussed. For example, if needle insertion points are too close from each other or to the edge of the wound, when the suture is pulled, the suture will tear the soft tissue instead of suturing the incision together. Experiment results show that our simulator can run on a standard personal computer and allow users to perform different suturing patterns with smooth graphics and haptic feedback.

Keywords Virtual suturing · Suture model · Wound closure · Tissue tearing · Haptic feedback · Surgical training environment · Serious games

S. Payandeh (✉) · F. Shi
Experimental Robotics and Graphics Laboratory,
Simon Fraser University, 8888 University drive,
Burnaby, BC V5A 1S6, Canada
e-mail: shahram@cs.sfu.ca

1 Introduction

In this paper, our intent is to design an interactive suturing and knotting environment with haptic feedback to offer a more realistic model that can be further developed as part of a serious gaming environment for training surgeons (Robert et al. 1999). Sensory feedback including both visual and force feedbacks are crucial requirements to make surgery simulations more realistic. To create more realistic models, we constructed our virtual objects based on our mechanics model of soft tissues and sutures to achieve a sense of haptic feedback. In the bimanual case studies of this paper, two Phantom Omnis haptic devices are utilized in our simulator to provide smooth force feedback and to allow the user to perform different suturing and knotting patterns during training.

There are a number of works that have made some contributions to the development of deformable linear objects (DLOs) simulation. Most of these previous models can be categorized as geometry-based models or physics-based models. Surgical training systems must enable users to feel the force feedback. Both external and internal forces must be considered to determine the force output, especially for suturing tasks, such as pulling and knotting. Some researchers have been focusing on knotting manipulation by robots. In Wakamatsu et al. (2005, 2006), established a model of DLOs based on an extension of differential geometry and proposed a planning method for knotting/unknotting of DLOs based on knot theory. If the initial and the objective states of the linear object are given, all possible knotting/unknotting plans can be derived and be executed by their system. However, their proposed models cannot simulate the DLOs dynamically in 3D space (Saha et al. 2006). In addition, their system does not allow any user interaction and cannot simulate the knotting/unknotting procedure in real time. In Pai (2002), Wang et al. (2003) and Cao et al. (2006), a Cosserat approach of modeling DLOs based on the Cosserat theory of elastic rods has been introduced. The Cosserat model is well suited for real-time applications because it needs less computation compared to finite elements models and provides a clear delineation between basic physical principles, material properties and mathematical approximations. However, it yields a set of ordinary differential equations to be solved. If two end points or multiple points along the length of a suture are specified (as in the procedure of knotting or unknotting with two hands), it is significantly more difficult to solve these equations. In addition, the "shooting" technique that is mentioned in Pai (2002) makes it very difficult to integrate external forces Grégoire and Schömer (2006) (Lenoir et al. 2002, 2004). A particle-based model of a rope is represented in Phillips et al. (2002) by overlapping spheres representing mass points, which are connected by simple springs. Each mass point can collide with other mass points as in the instantaneous elastic collision model, but the author only considers the linear spring forces and does not allow any user interaction. In Kahl and Henrich (2005), inner bending force and the gravity are taken into consideration. In Wang et al. (2005), the author mentioned gravity, stretch/compression force, forces from bending and twisting, dissipative friction, and contact forces with environment or to self-collision, but there is no detail about how to compute those forces. A mass-spring model for suturing in surgical training system has been built in LeDuc et al. (2003). Torsional spring, torsional damper, and viscous damper are mentioned in this paper, but the author did not use them in the simulation due to the complex computation. Furthermore, there is no discussion about collision detection and force propagation for haptic interaction between the user and the suture model.

Virtual suturing simulations integrate various techniques from different areas, such as modeling deformable objects, collision detection, and haptic and graphic rendering. Deformability of the objects, challenges of collision detection management, and high demanding of haptic rendering make this topic very difficult to model and simulate. Many surgical training environment have been designed with or without using haptic devices recently. A geometry-based suture model is presented in Brown et al. (2001) with a outline of modeling a suture simulation. Webster et al. (2001) describe a haptic simulation for teaching basic suturing skills for simple wound closure. During the simulation, needle holders, needle, sutures, and virtual skin are displayed and updated at real time. Marshall et al. (2005, 2006) present a haptic based suturing simulator operating on mass-spring surface meshes. The suture is built as a geometrical model and the "following the leader" is used to simulate displacements of suture nodes at each time step. A knot planner is also presented to let users tie a knot after the suturing. However, geometry models are less physically accurate and less realistic for dynamic simulations. Furthermore, the authors did not study the scenario about tissue tearing which can occur when stitches get over stretched, which can be one of the key metrics to evaluate the suturing skills. Berkley et al. (1999, 2004) show that real-time finite element method (FEM) can be used to simulate suturing in surgical training systems. The authors present a new real-time methodology based on linear FEM analysis and prove the constraint approach is well suitable for suturing simulations. The suturing task is broken down into 9 steps in Berkley et al. (2004). However, not all of these steps are implemented. Moreover, the computational model based on FEM is quite CPU intensive. In LeDuc et al. (2003), the authors present their initial works on simulating suturing using mass-spring models. Their suturing simulator shows that a wound could be closed by the suturing. However, it does not have any haptic interaction and the model does not allow the placement of arbitrary incision points around the wound. In addition, no models for the tissue reaction to excessive suture pulling forces are presented, which can result in tissue being ripped. A virtual suturing simulator with haptic feedback is presented in Lian and Chen (2006), in which the deformable tissue is modeled as a multi-layer mass-spring system. The authors show that their simulator could suture a pre-wound soft tissue. However, the paper does not address the knotting aspects which can also be one of the key elements in the surgical training environment with some key performance evaluation metric such as tissue tearing.

As the initial work of studying the force propagation between the suture and the soft tissue, this paper presents a method of simulating not only the detailed steps of one typical suturing procedures, but also a scenario when the tissue gets ripped when the needle insertion points are too close from each other or to the edge of the wound, which always happens to beginner trainees who lack suturing technique skills. Our suture model is built based on all the forces mentioned in LeDuc et al. (2003), and we provide a user interface to allow users tie an arbitrary knot. Also we analyze how the forces propagate along the suture during knotting and unknotting. With the virtual coupling technique Vafai and Payandeh (2010), we can provide very smooth force feedback to the user. The rest of this paper is organized as the following: Sect. 2 presents an overview of the mechanics-based modeling of components involved in a suturing task; Sect. 3 presents the implementation of multi-body collision detection management of the virtual suturing simulator; Sect. 4 gives details of interaction models involved in suturing procedure and stitch evaluation, and Sect. 5 presents experimental results followed by discussion section.

2 Model description

In this section, we present an overview of models for a skin or soft tissue, a needle, needle driver, and detailed modeling of the suturing task.

2.1 Soft tissue

Deformable objects can be modeled based on either physics-based or geometry-based modeling approaches. In Terzopoulos and Waters (1990), the authors have motivated the advantages of using physics-based approaches for animating physical characters and phenomenon. Among all the physics-based models, finite element method (FEM) as in Berkley et al. (1999, 2004), Chen and Zeltzer (1992), Bro-Nielsen and Cotin (1996), Bro-Nielsen (1998) and mass-spring system (MSS) as in Marshall et al. (2005, 2006), LeDuc et al. (2003), Hutchinson et al. (1996), Hui et al. (2004), Cha and Payandeh (2007) are two commonly used models for representing deformable objects. Payandeh and Azouz (2001) present an overview of haptic rendering using FEM and MSS. Lian and Chen (2006) have argued the benefits of using MSS in surgical simulations with haptic feedback.

A MSS can be applied to either volumetric elements such as tetrahedral or polygonal surface elements such as triangles. In order to reduce computation load during the real-time simulation, we chose a surface mass-spring model based on the model described in Hui et al. (2004). Although a home spring has been added to each node to maintain the shape of the deformable object, the adjacent edges in small triangles still could bend in any angle if their common node is dragged by needle or suture. As a result, we augmented the model for creating bending moments using coupled angular springs to each pierced vertex. We build our 3D models of pre-wound skin or soft tissue in 3D graphics applications such as Autodesk 3ds max and then export out the model as VRML file. Our simulator imports these VRML files to build the virtual objects. In this way, we can simulate different kinds of cuts for suturing tasks.

To reduce the computation load and to satisfy the high demand of haptic rendering, we have selected the simple iterative explicit Euler method to update the states of the soft tissue and the suture. Although the explicit method has some drawbacks comparing to the implicit method, our experimental results show that we can make the simulator stable by restricting the integration time step dt to be inversely proportional to the square root of the stiffness. In the explicit Euler method, the forces at time t_n contribute to the velocities at time t_{n+1}. Higher-order schemes (like Runge–Kutta) are better in terms of accuracy and smooth solutions. However, they may not be suitable for interactive haptic environment where constrains on the dynamical system change as a function of the status of contact between bodies. In this study and at each time step, we compute forces acting on each mass node at time step t_n. The explicit Euler integration scheme can be written as $\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{F}_i^n \frac{dt}{m}$ and $\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1} dt$ where i—discrete mass point, m—mass of each point, $\mathbf{x}_i$—position of point i, $\mathbf{v}_i$—velocity of point i, $\mathbf{F}_i^n$ is the net force acting on the node i at time step t_n (Provot 1995). If the node i is also the pierced node, it is also dragged by the needle or suture. In order to make the soft tissue deformation more smooth, we add torsional spring and torsional damper (Shi and Payandeh 2007).

2.2 Overview of suture model

As a one-dimensional element, we model our suture as mass-spring system that consists of a sequence of mass points laying on the centerline of the suture connected together by various types of spring (Shi and Payandeh 2007) (see (a) of Fig. 1). Let l_i be the segment length between P_i and P_{i+1}. To prevent the suture from being stretched too long or compressed too short, we set $l_{\max}$ and $l_{\min}$ as the maximum and minimum length of one suture segment, respectively.

During graphic rendering, we rendered cylinders as suture segments connecting two successive points. We use Euler method to integrate the dynamical model at each mass point in order to compute the overall shape of the suture. We first compute the total force acting at each point, P_i, and then update its position based on the computed force. Once the total force at each of the nodes has been calculated at a given time interval dt, we can then compute the velocity and position of each point.

Beside graphical visualization, a haptic device is used as a dynamic user interface device to provide both input and output to the virtual reality scene in the form of forces, positions, and velocity. However, a position-controlled impedance style haptic device, such as PHANTOM Omni from Sensable Tchenology (SensAble 2005), forces are not directly available as input variables into the model. We

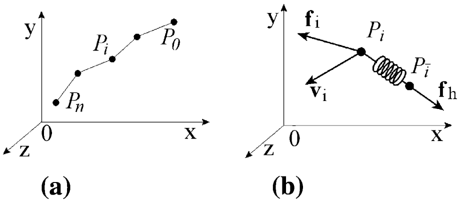


Fig. 1 **a** Suture Model. **b** A Virtual Coupling for Interactive Manipulation with Haptic Devise

used a virtual coupling technique (Colgate et al. 1995) to introduce an indirect layer of interaction between the mechanical device and the simulation by employing a spring-damper between a simulated body and the device end-effector (see Fig. 1b).

To capture the interaction model of the suture segments during the knotting (or unknotting) procedure, we assume that along the suture there are only two segments colliding with each other (see Fig. 2). Let μ be the friction coefficient, $\hat{\mathbf{e}}$ be the direction of the friction force, $\mathbf{n}$ be the repulsive force at the contact area. Then, the friction $\mathbf{f}_f$ can be described as: $\mathbf{f}_f = \mu||\mathbf{n}||\hat{\mathbf{e}}$. To compute the repulsion force $\mathbf{n}$, we introduce a spring-damper at the contact areas C and E on the two segments. We can then define $\mathbf{n} = (k_{rs}(2r - d) - k_{rd}(\mathbf{v}_r \cdot \hat{\mathbf{n}}))\hat{\mathbf{n}}$ where k_{rs} is a spring constant coefficient for the repulsion force, r is the radius of the suture model, d is the distance between point C to point E (Fig. 2b), k_{rd} is the damper constant for the repulsion force, $\mathbf{v}_r$ is the relative velocity of point C with respect to point $E, \hat{\mathbf{n}}$ is the unit vector from point E to point C. We also introduce linear interpolation function for computing the velocity of any point along each segments. For example (Fig. 2), $\mathbf{v}_c = (1 - a)\ \mathbf{v}_a + a\ \mathbf{v}_b$, where a is the fraction of point C along $\overrightarrow{P_aP_b}, \mathbf{v}_e = (1 - b)\mathbf{v}_c + b\mathbf{v}_d$, where b is the fraction of point E along $\overrightarrow{P_cP_d}$. Then, the relative velocity $\mathbf{v}_r = \mathbf{v}_c - \mathbf{v}_e$, and the friction direction vector $\hat{\mathbf{e}}$ is computed as follows: $\hat{\mathbf{e}} = \frac{(\mathbf{v}_r \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \mathbf{v}_r}{||(\mathbf{v}_r \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \mathbf{v}_r||}$.

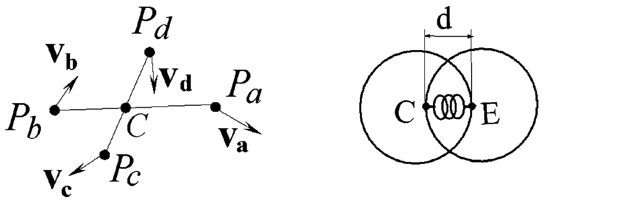
2.2.1 Internal forces

The spring force along the suture ($\mathbf{f}_s$) is computed by comparing the current length of the segment l_i defined between points, P_i and P_{i+1}, with the rest length of the segment l_r, projected along the direction from point P_i to P_{i+1}. Or we have, $l_i = ||P_{i+1} - P_i||, \Delta l = \frac{l_i - l_r}{l_r}$. Let $\hat{\mathbf{e}}_i$ be the unit vector from point, P_i to P_{i+1}, then we can write $\hat{\mathbf{e}}_i = P_{i+1} - P_i/||P_{i+1} - P_i||$, and $\mathbf{f}_s = k_l \Delta l \hat{\mathbf{e}}_i$. where k_l is the linear spring constant. We also define a viscous damping force defined as: $\mathbf{f}_d = k_d(v_{i+1} - v_i)\hat{\mathbf{e}}_i$. where $v_{i+1} = \mathbf{v}_{i+1} \cdot \hat{\mathbf{e}}_i, v_i = \mathbf{v}_i \cdot \hat{\mathbf{e}}_i, k_d$ is the linear damper constant. v_{i+1} and v_i are the norms of the components of the velocity of point P_{i+1} and P_i on the direction $\hat{\mathbf{e}}_i$.

2.2.2 Torsional mechanics

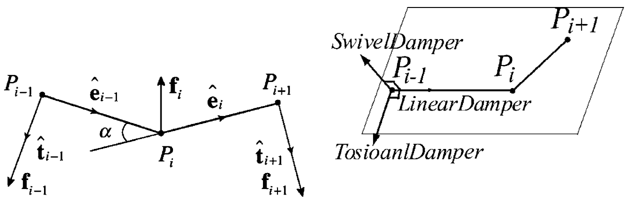
The torsional spring and damper are derived from the angle, α, between two connected segments of the suture. The basic idea is to model each two connected segment points as a vertex of triangle with a spring model between the two connecting edges. In this model, the force component orthogonal to each of the segment is used at each of the end vertices (Fig. 3a). Let $\hat{\mathbf{e}}_{i-1}$ and $\hat{\mathbf{e}}_i$ be the unit vectors with directions from point, P_{i-1} to P_i, and from P_i to P_{i+1}, respectively. Let $\hat{\mathbf{t}}_{i-1}$ and $\hat{\mathbf{t}}_{i+1}$ be the unit vectors with directions the same as the torsional force applied at the two end points and therefore, orthogonal to $\hat{\mathbf{e}}_{i-1}$ and $\hat{\mathbf{e}}_i$ respectively. Then, $\hat{\mathbf{t}}_{i+1} = \hat{\mathbf{e}}_i \times (\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i), \hat{\mathbf{t}}_{i-1} = \hat{\mathbf{e}}_{i-1} \times (\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i)$. If $\hat{\mathbf{e}}_{i-1} \cdot \hat{\mathbf{e}}_i \geq 0, \alpha = \arcsin(||\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i||)$. If $\hat{\mathbf{e}}_{i-1} \cdot \hat{\mathbf{e}}_i < 0, \alpha = \pi - \arcsin(||\hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i||)$. The torsional spring force can then be defined as: $\mathbf{f}_{i-1} = k_{ts}\frac{\alpha}{\pi||P_{i-1} - P_i||}\hat{\mathbf{t}}_{i-1}, \mathbf{f}_{i+1} = k_{ts}\frac{\alpha}{\pi||P_{i+1} - P_i||}\hat{\mathbf{t}}_{i+1}$ and $\mathbf{f}_i = -(\mathbf{f}_{i\text{-}1} + \mathbf{f}_{i+1})$, where k_{ts} is the torsional spring constant. The torsional damper works against the torsional spring to prevent any harmonic motion from accumulating. Similar to the linear damper, it also models the internal friction that resists bending in regular objects. Let v_{i-1}, v_{ib} be the norms of the velocity components of $\mathbf{v}_{i-1}$ and $\mathbf{v}_i$ on the direction of $\hat{\mathbf{t}}_{i-1}$, and let v_{i+1}, v_{ia} be the norms of the velocity components of $\mathbf{v}_{i+1}$ and $\mathbf{v}_i$ on the direction of $\hat{\mathbf{t}}_{i+1}$. Then, the torsional damper on the points, P_{i-1}, P_i and P_{i+1}, can be computed as: $\mathbf{f}_{i-1} = (\frac{(v_{i-1} - v_{ib})}{||P_{i-1} - P_i||} + \frac{(v_{i+1} - v_{ia})}{||P_{i+1} - P_i||})\frac{k_{td}\hat{\mathbf{t}}_{i-1}}{||P_{i-1} - P_i||}, \mathbf{f}_{i+1} = (\frac{(v_{i-1} - v_{ib})}{||P_{i-1} - P_i||} + \frac{(v_{i+1} - v_{ia})}{||P_{i+1} - P_i||})\frac{k_{td}\hat{\mathbf{t}}_{i+1}}{||P_{i+1} - P_i||}$, and $\mathbf{f}_i = -(\mathbf{f}_{i-1} + \mathbf{f}_{i+1})$. where k_{td} is torsional damper constant, $v_{i-1} = \mathbf{v}_{i-1} \cdot \hat{\mathbf{t}}_{i-1}, v_{ib} = \mathbf{v}_i \cdot \hat{\mathbf{t}}_{i-1}, v_{i+1} = \mathbf{v}_{i+1} \cdot \hat{\mathbf{t}}_{i+1}, v_{ia} = \mathbf{v}_i \cdot \hat{\mathbf{t}}_{i+1}$.

In general, vertex point, P_{i-1}, has a velocity relative to P_i. So far, damping elements have been introduced to the two components of the relative velocity. There still remains a component perpendicular to those two. Without the dampening, point P_{i-1} could indefinitely orbit the line formed by extending the edge connecting points P_{i+1} and



(a) *Two Sliding Segments* (b) *Two Segments Intersection*

Fig. 2 Model description for suture segment overlaps



(a) *Torsional Spring* (b) *Swivel Damper*

Fig. 3 Torsional Spring and Swivel Damper

point P_i (Fig. 3b). Let $\hat{\mathbf{s}}$ be the unit vector of the swivel dampers of point P_{i-1} and P_{i+1}, then, $\hat{\mathbf{s}} = \hat{\mathbf{e}}_{i-1} \times \hat{\mathbf{e}}_i$. The swivel dampers can be computed as: $\mathbf{f}_{i-1} = k_{sw} \frac{(\mathbf{v}_{i-1}-\mathbf{v}_i)\cdot\hat{\mathbf{s}}}{||P_{i-1}-P_i||}\hat{\mathbf{s}}, \mathbf{f}_{i+1} = k_{sw}\frac{(\mathbf{v}_{i+1}-\mathbf{v}_i)\cdot\hat{\mathbf{s}}||P_{i+1}-P_i||}{\hat{\mathbf{s}}}$, and $\mathbf{f}_i = -(\mathbf{f}_{i-1} + \mathbf{f}_{i+1})$ where k_{sw} is the swivel damper constant.

2.3 Needle and needle driver

Tools that are involved in a real surgical suturing task include needle, needle drivers, scissors, and tweezers. In this case study, we focus on modeling the needle and the needle driver as shown in Fig. 4. Figure 5a shows the needle model. We use arc *TE* to represent the needle. Point *O* is the center of the curvature of the needle and also is the origin of local coordinate of the needle. Point *T* and *E* are the tip point and the end point of the needle, respectively. *r* is the radius of the arc *TE*. $\hat{\mathbf{y}}$ is the unit tangent vector of arc *TE* at point *T*. Given any point *P* on the needle, we can obtain its position if the angle α from tip to *p* is given. *OP* can be derived as: $OP = OT\cos(\alpha) + r\sin(\alpha)\hat{\mathbf{y}}$, where $\hat{\mathbf{y}} = ((OT \times OE) \times OT)/(||OE||||OT||||OT||)$. Figure 5b shows the model of the needle driver. Triangle *AOB* is the open triangle of the needle driver.

2.4 Haptic force model along the suture

This section presents the force propagation model for the case when the user grasps the suture using one or two haptic devices for the one-handed or bi-manual manipulation. In ours, we need to compute the forces acting at each suture point starting from the grasp point propagating to the end point of the suture. Here, we define three distinct states based on the position solution of the suture point:

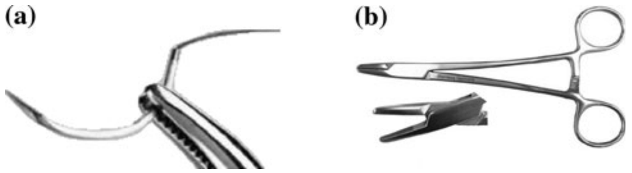


Fig. 4 **a** A real needle hold by a driver. **b** A real needle driver

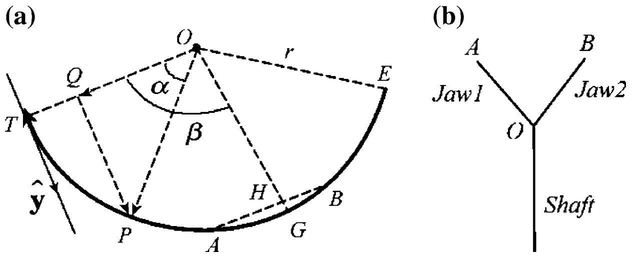


Fig. 5 **a** Needle model—arc *TE* represents the needle during simulation. **b** Model of the needle driver

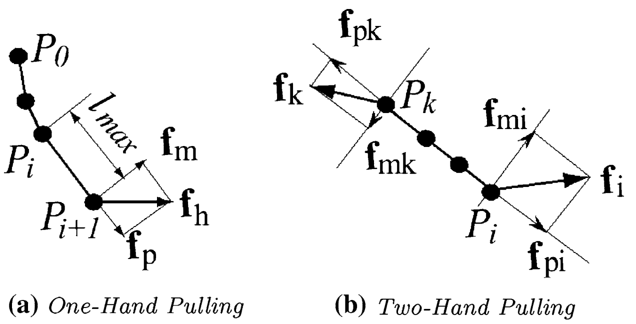
Case A—The user grasps point P_{i+1} with one hand. If $l_{\min} < l_i < l_{\max}$, there will not be any propagation of the user input force $\mathbf{f}_h$ from point P_{i+1} to P_i. All the external forces generated by the user are mapped to the motion of the suture points.

Case B—The user grasps point P_{i+1} with one hand. If the expected solution for the length of the segment $l_i' > l_{\max}$ or $l_i' < l_{\min}$, geometrically, the length of the segment is adjusted to $l_{\max}$ or $l_{\min}$ (Fig. 6a). For example, let $\mathbf{f}_p$ be the component of the input force $\mathbf{f}_h$ along the segment direction and $\mathbf{f}_m$ is the component of the input force which propagates to point P_i from point P_{i+1}. $\mathbf{f}_p$ and $\mathbf{f}_m$ can be obtained as: $\mathbf{f}_p = (\mathbf{f}_h \cdot \hat{\mathbf{e}}_i)\hat{\mathbf{e}}_i, \mathbf{f}_m = (\mathbf{f}_h \cdot \hat{\mathbf{e}}_m)\hat{\mathbf{e}}_m$. where $\hat{\mathbf{e}}_m = \frac{\hat{\mathbf{e}}_i \times \mathbf{f}_h}{||\hat{\mathbf{e}}_i \times \mathbf{f}_h||} \times \hat{\mathbf{e}}_i, \hat{\mathbf{e}}_i$ is a unit vector pointing from P_i to P_{i+1}. Using a similar approach, we can derive the user input force propagated at each point of the suture.

Condition C—In this condition, the user grasps two points, P_k and P_i, of the suture. The force propagation model is similar as stated in the Condition B. But the computation needs to be repeated first starting from point P_i and then starting from point P_k Fig. 6b.

3 Collision detection and management

One of the important components of the interactive virtual suturing is the notion of collision detection management. For example, for collision detection between the suture and its environment (such as self-collision and contact with the grasping tools), similar to (Brown et al. 2004), we have constructed a bottom-up bounding-volume hierarchy (BVH) of the suture at successive levels of detail (Fig. 7). In order to determine self-collisions of the suture, we search two copies of the BVH from top down. Whenever two BVHs (one from each copy) are not overlapping, the two segments are not in colliding state and we do not explore the segment representation of their contents. When two leaf spheres overlap, the distance between the two



(a) *One-Hand Pulling* **(b)** *Two-Hand Pulling*

Fig. 6 Force Propagation

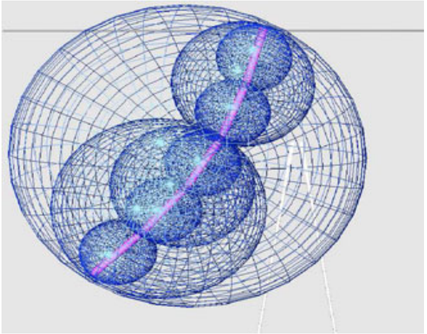


Fig. 7 Bounding-Volume Hierarchy

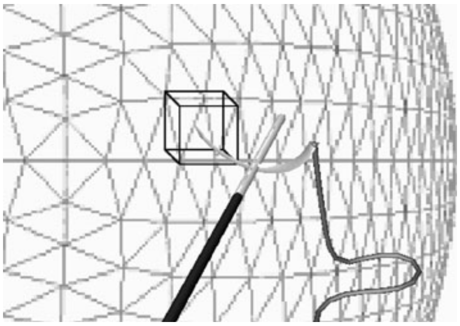


Fig. 8 Bounding box of the needle tip

centers of the nodes is computed. If it is less than the node diameter, $2r$, then the two segments are reported to be colliding.

The collisions between the suture and grasper are determined by modeling the graspers as line segments with a given radius, and check whether the BHV of the suture has any overlaps with the line segments of the grasper. When two suture segments are detected to be at a distance $d < 2r$ from each other, an equal (but opposite) displacement vector is applied to each segment $r - d/2 + \varepsilon/2$. At the instant of collision, we also need to compute new velocities of mass points of the corresponding suture segment which are involved in the collision. Similarly to the method presented in Bridson et al. (2002), we apply impulse to the colliding points of these two segments. Figure 2 shows the case where point C with relative position a along the segment $\overrightarrow{P_aP_b}$ interacts with point E with relative position b along the segment $\overrightarrow{P_cP_d}$. Let $\mathbf{i}$ be the impulse, then $\mathbf{i} = \mathbf{n}\Delta t$ where $\mathbf{n}$ is the repulsion force that we have defined in Sect. 2.2. The new velocities due to the impulses can be defined as: $i' = \frac{2\|\mathbf{i}\|}{(a^2+b^2+(1-a)^2+(1-b)^2)}$, $\mathbf{v}_a^{new} = \mathbf{v}_a + (1-a)\frac{i'}{m}\hat{\mathbf{n}}, \mathbf{v}_b^{new} = \mathbf{v}_b + a\frac{i'}{m}\hat{\mathbf{n}}, \mathbf{v}_c^{new} = \mathbf{v}_c - (1-b)\ \frac{i'}{m}\hat{\mathbf{n}}$, and $\mathbf{v}_d^{new} = \mathbf{v}_d - b\frac{i'}{m}\hat{\mathbf{n}}$. where m is the mass of each mass point P_a, P_b, P_c, and P_d. $\hat{\mathbf{n}}$ is the unit vector from point E to point C.

3.1 Needle drivers and needle

When needle is grasped, it should follow the movements of the stylus of the haptic device. Here, we represent the needle arc into six connected line segments. Let, for example, the line segment AB be one of these needle segments (Fig. 5a). First, we check collision of the needle driver's open triangle with each needle segment. At the instant of collision, the intersection point G on the needle can be obtained through it projection point H which can be computed through the angle β which is the angle between OT and OH.

3.2 Needle and soft tissue

In order to determine the collision between the needle and the soft tissue, we define a bounding box around the needle tip using the length of the longest edge spring of the soft tissue model (Fig. 8). This bounding box is always following the movement of the needle. Whenever a mesh node is found inside the bounding box, we mark all the vertices of its adjacent polygons as candidate collision points with the needle tip.

3.3 Suture and soft tissue

In suturing task, the pierced node of the organ model should follow the movements of the suture in cases when the spring force acting on the node is less than the suturing friction force. Otherwise, the node should be sliding along the suture. We call each of these new pierced node as a soft constraint. Three kinds of soft constraints for the suture are defined based on the location of the pierced node: *Top Constraint*, *Bottom Constraint*, and *Groove Constraint*. Figure 9 shows the schematic of a suturing pattern. Node C and N are top constraints. Node D and E are bottom constraints.

We name the constraints in an order along the suture. At any instances, no more than one constraint is defined at the

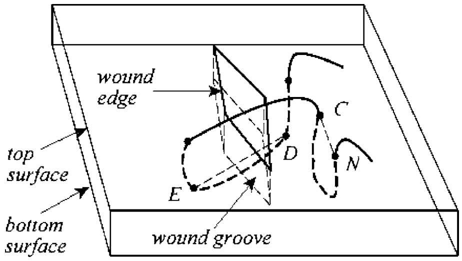


Fig. 9 Schematic of a suturing pattern. Node C and N are top constraints. Node D and E are bottom constraints

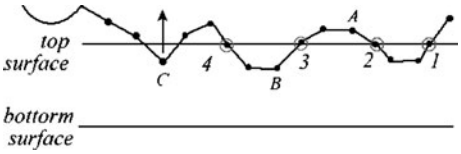


Fig. 10 Collisions between suture and soft tissue

a suture node. To implement the collision detection between the suture and the soft tissue, we first determine whether the suture nodes are inside or outside of the soft tissue. This is determined by a permutation algorithm. We enumerate how many constraints have passed a given suture node since all constraints start from the needle and have to pass from the needle to the suture in order. If the number is even, the suture node should be outside the soft tissue; Otherwise, it should be inside. For example, referring to Fig. 10, two constraints have passed the suture node *A*; therefore, *A* should be outside of the tissue. In addition, there are three constraints that have passed node *B*, so *B* should be inside. Using this approach, we know that suture node *C* should be outside; therefore, we must move node *C* to the outside of the soft tissue during the collision management.

4 Suturing procedure and stitch evaluation

We have divided one complete suturing task into five states (Marshall et al. 2005; Shi and Payandeh 2007): (1) needle pierces the object; (2) soft constraints slid on the needle; (3) soft constraints slid on the suture; (4) application of tension on the suture in order to close the wound or incision; and (5) knotting. After we have found a contact point between the needle and the soft tissue, we subdivide the contact polygon into several smaller polygons (Hui et al. 2004; Marshall et al. 2005) (Fig. 11). To prevent any two connected springs from bending easily to any undesirable angle, we have added torsional springs and dampers to each of these newly created springs. Because the torsional spring and damper should be configured on two connected springs as a pair, we have to consider two connected springs simultaneously. Referring to Fig. 11b, for example, suppose point *O* is the pierced node on the soft tissue surface. We add torsional springs and dampers to springs *FO* and *OR* as a pair,to springs *DO* and *OP*, and to springs *EO* and *OQ*.

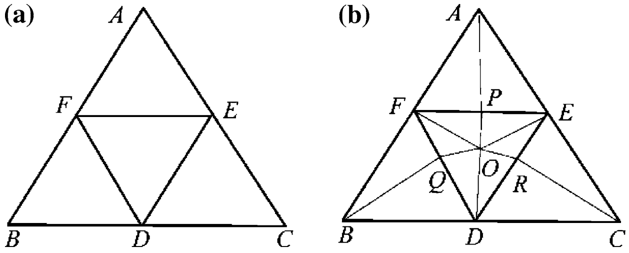


Fig. 11 **a** Triangles before subdivisions. **b** Triangles after subdivisions

4.1 Tissue tearing

To make the suturing simulation more realistic, in addition to modeling and simulating basic suturing task, one also needs to consider some examples of unwanted events which the user may create. Ripping of the tissue can be one of these unwanted circumstances that can occur during the training of a novice surgeons or medical students. It is also one of the key metrics to evaluate the trainee’s performance during the training procedure using this serious gaming configuration.

Based on the assumption that the stitch can go through from the top surface to the bottom surface of the soft tissue, we define two types of tearing: *tear-into* and *tear-through* that are similar to the cut-into and the cut-through described in Hui et al. (2004). We also define *top start constraint* to be the constraint where the tearing starts on the top surface and *top end constraint* to be the constraint where the tearing ends on the top surface. For tear-through case, we define *bottom start constraint* and *bottom end constraint* with the same idea. Figure 12a shows that point *C* is a top start constraint in tear-into case, point *D* is a groove constraint. Figure 12b shows that point *C* is a top start constraint and point *D* is a bottom start constraint.

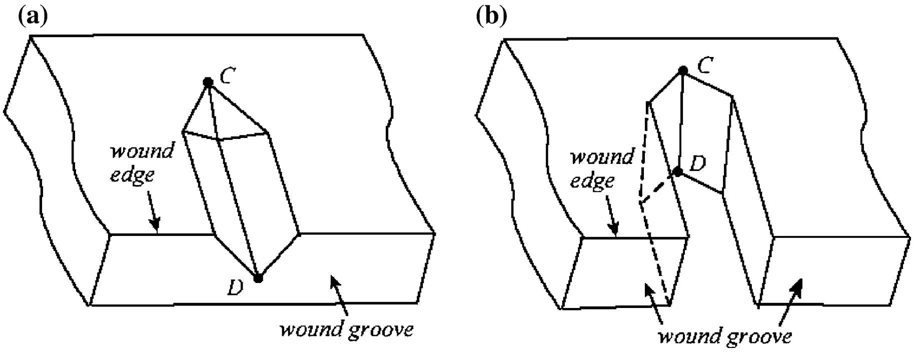


Fig. 12 **a** Tear-into the soft tissue. *C* is the top start constraint and *D* is a groove constraint **b** Tear-through the soft tissue. *C* is the top start constraint and *D* is the bottom start constraint

4.1.1 Tear-into

To simulate the tearing, we followed a similar approach that we have used in simulation of virtual cutting algorithms as described in Hui et al. (2004) and Cha and Payandeh (2007). First, we divide the tearing of one polygon into two states: *Start State* and *Termination State*. Two additional states are also defined for each of these two states: *tear-through vertex* and *tear-through edge* that depend on the intersection point between the tearing path and the teared polygon. For our case, the start state of first tearing polygon and the termination state of the last tearing polygon are always tear-through vertex (Fig. 13).

To determine the state of the middle polygon along the tearing path, we need to find out the intersection points between each of these polygons and the tearing path. In order to mask the subdivision due to the deformation of the tissue model, we simply map the home positions of each mass node on the top surface onto a plane in order to find out the intersections points between the mapped tearing path and each mapped edge, and then we inverse map the virtual intersection points back to the real surface of polygons. Figure 14 shows that C is the top start constraint and N is the top end constraint. C' and N' are the mapped nodes to C and N, respectively.

Two cases need to be resolved depending on whether the tearing path crosses the wound or not. Figure 12a shows a half of the section of a tearing across the wound. Figure 15 shows the polygon subdivision of a tearing which does not cross the wound. For the former case, we use the groove constraint to set the depth of the tearing groove (point D in Fig. 12a). For the case that the tear does not cross the wound, we have selected a default tearing depth.

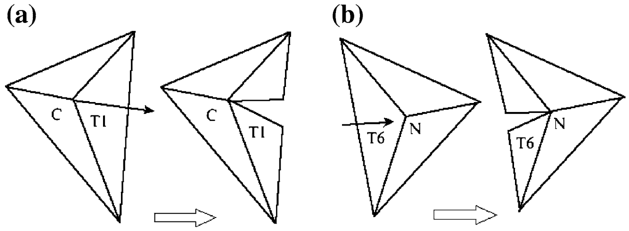


Fig. 13 **a** Start state of the first tearing polygon. **b** Termination state of the last tearing polygon

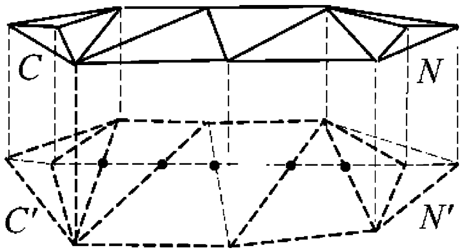


Fig. 14 Mapping the surface polygons onto a plane to find out the intersection points

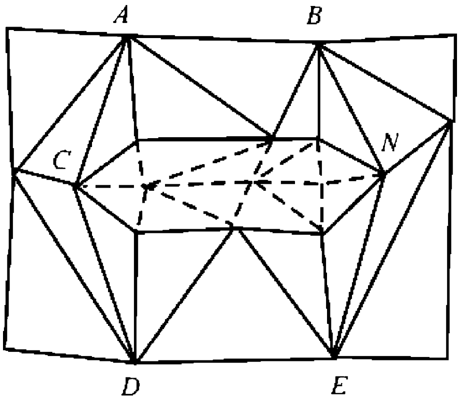


Fig. 15 Surface polygon subdivision in tear-into condition where the tearing path is not across the wound

4.1.2 Tear-through

In this condition, both the bottom intersection points along with the bottom tearing path and the top intersection point along the top tearing path need to be determined. We used similar mapping approach as in tear-into case to map the top surface and the bottom surface of the soft tissue separately. Then, we subdivide the bottom surface and the top surface at the same time. After the subdivision, we use groove polygons to connect the bottom and the top up. Figure 12b shows a tear-through case.

5 Experiment results

We conducted an experimental studies of interacting with the virtual suturing environment on a standard personal computer with Intel(R) Pentium(R) 4 3.00GHz CPU and 1G RAM. We have used two haptic devices, namely two Phantom Omni from Sensable Technology for the bi-manual suturing and knotting tasks proposed in this paper. The computed task forces are scaled accordingly to match with the required force output of the haptic devices. Figure 16 shows our experimental setup configuration.

5.1 Suturing

The following scenarios show the details of different steps in one suturing procedure. Figures 17 to 18 show the changes in the soft tissue surface before the needle pierces the surface and the after. As it can be seen from the wireframe screen shots, polygon changes show the results of the subdivision algorithm discussed in the previous section.

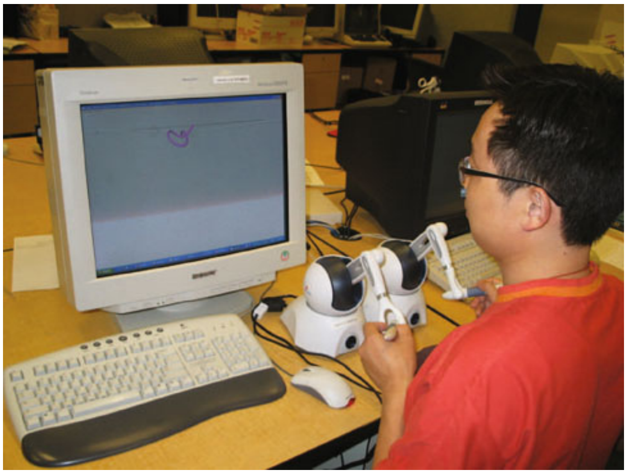


Fig. 16 The multi-modal experimental setup consisting of two haptic devices and graphical display

Figures 19 to 20 show the sample results when the constraint is sliding both on the needle and on the suture. Figure 21 shows a simple continuous suture pattern implemented in our simulator. Figure 22 shows a simple suturing pattern with a knot.

Haptic feedback plays an important role in any virtual interaction which requires the user to touch and manipulate objects. In a virtual training environment, it has been shown that haptic feedback can increase the sense of realism in virtual interaction (Moody et al. 2001). In particular, hypothesis of the study of (Vafai and Payandeh 2010) which was shown through user studies was that participants prefer haptic feedback while using a virtual dissection environment rather than no haptic feedback. As a result of the added haptic cues, it was predicted a better (lower) performance times for the tasks using haptic feedback than for not feedback. Through their preliminary user studies, they were able to support the hypothesis and effectiveness of presence of haptic feedback in interaction with virtual environment.

In order to compute the resultant interaction forces that are fedback to haptic devices, it is necessary to analyze the details of the forces propagation along the suture, between the suture and the soft tissue, inside the soft tissue and also track the forces acting on each soft constraint in the suturing procedure. In Marshall et al. (2006), we have studied the forces when the needle pierces the deformable meshes. In following section, we only present force

Fig. 17 Before the needle pierces tissue. **a** Screen shot **b** Wire frame

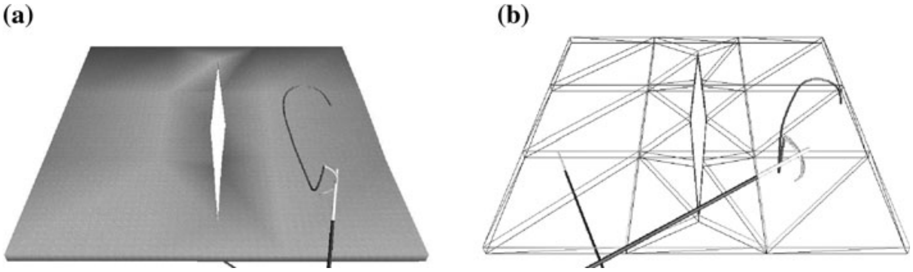


Fig. 18 After the needle pierces tissue. **a** Screen shot **b** Wire frame

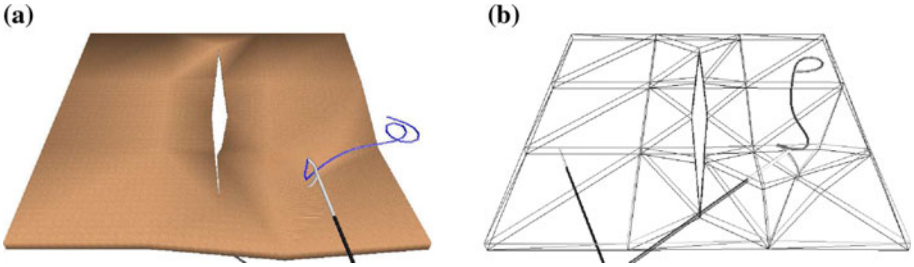
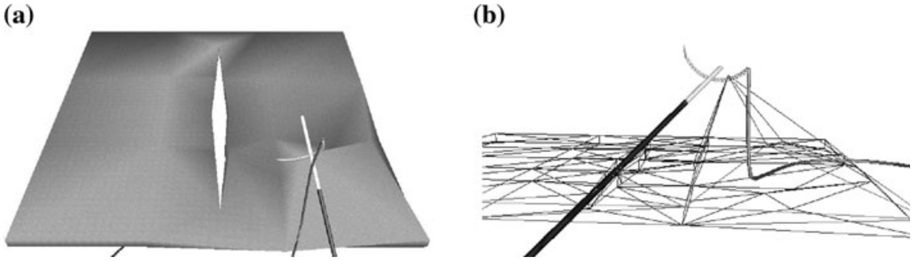


Fig. 19 Constraint slipping on the needle. **a** Screen shot **b** Wire frame



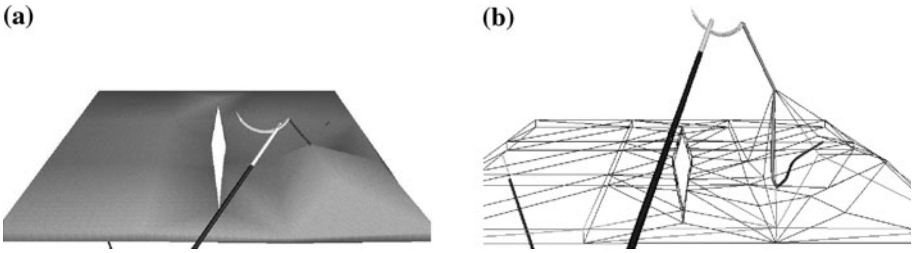


Fig. 20 Constraint slipping on the suture. **a** Screen shot **b** Wire frame

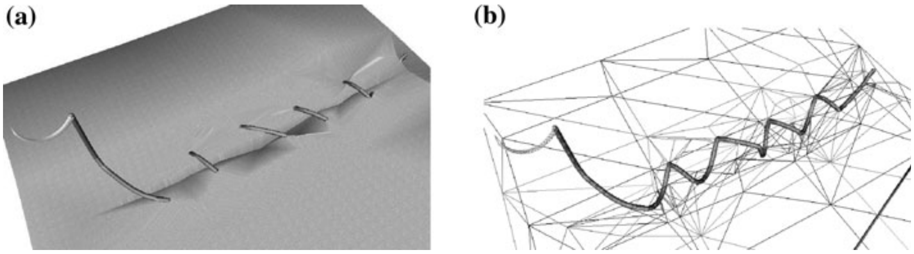


Fig. 21 A simple continuous pattern. **a** Screen shot **b** Wire frame

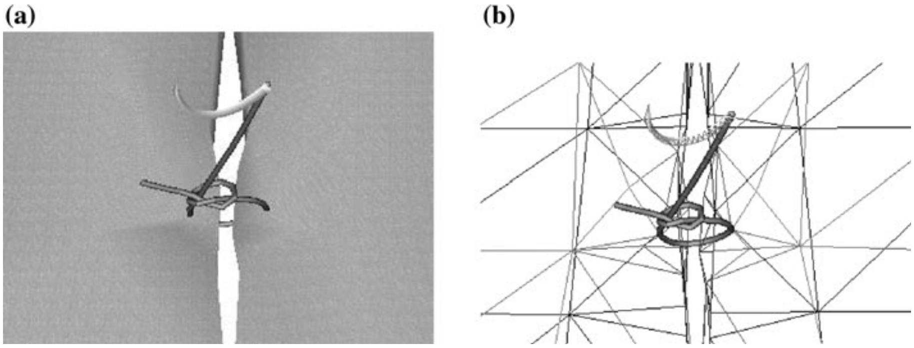


Fig. 22 A single stitch with a knot (Takamatsu et al. 2006). **a** Screen shot **b** Wire frame

changes at the pierced node when the suture can tear the soft tissue.

5.2 Tearing

For the tearing experiments, we manually set the end point of the suture to be always outside of the suture to prevent the suture from sliding out the soft tissue from the insertion points.

Figures 23 and 24 are screen shots of tear-into case. Figure 23 shows that two needle insertion are too close from each other and the tearing path does not cross the wound. Figure 24 shows that the needle insertion points are too close to the edge of the wound and the tearing path crosses the wound. Figures 25 and 26 are screen shots of tear-through case. Figure 25 shows that tearing path crosses the wound. Figure 26 shows that tearing path does not cross the wound.

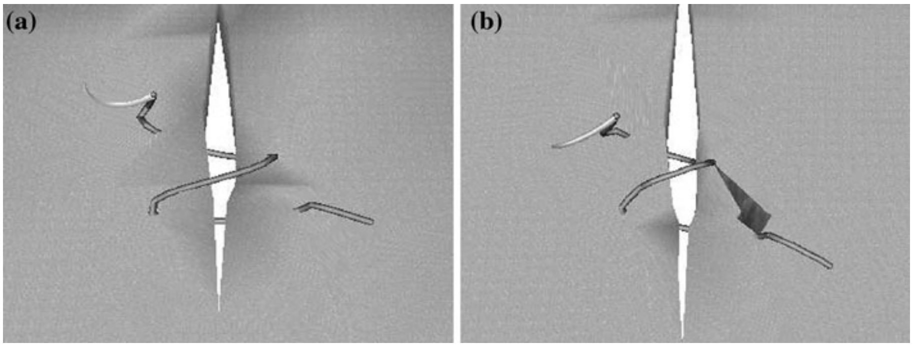


Fig. 23 Tear-into-tearing path not across the wound. **a** Before tearing **b** After tearing

Fig. 24 Tear-into-tearing path across the wound. **a** Before tearing **b** After tearing

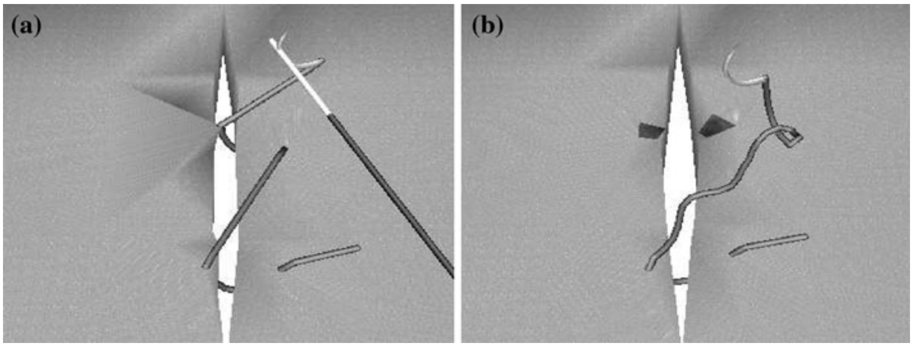


Fig. 25 Tear-through-tearing path across the wound. **a** Before tearing **b** After tearing

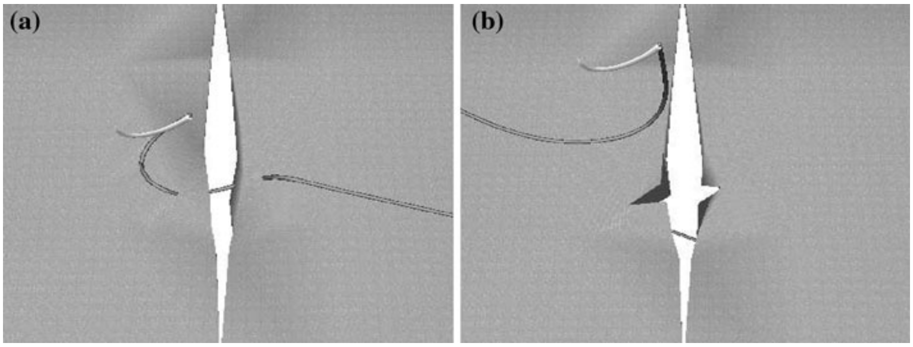


Fig. 26 Tear-through-tearing path not across the wound. **a** Before tearing **b** After tearing

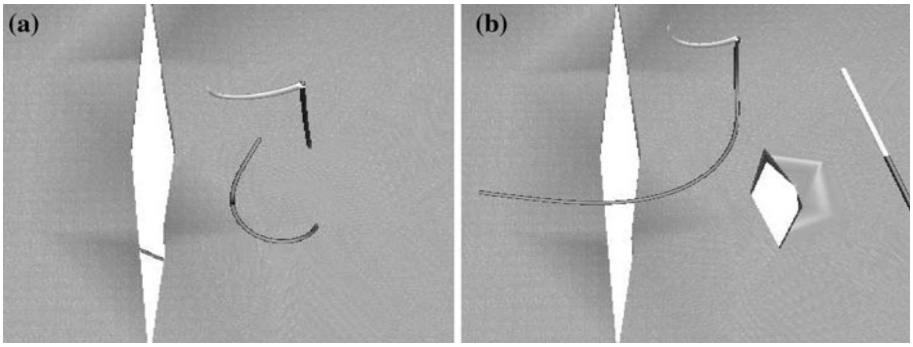


Fig. 27 Plot about the force acting on a pierced node when the suture tears the tissue from this node

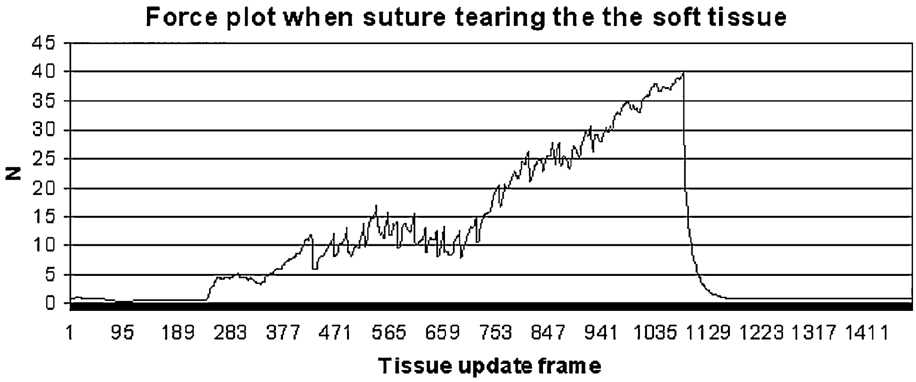


Figure 27 shows a sample plot of the interaction force when the suture exerted excessive force at the pierces node of the soft tissue. This computed force is then scaled and is fedback to the haptic device that can represent a continuous haptic cues throughout the manipulation.

6 Discussion and future work

We present a practical suturing simulator for surgical training environments based on physics models. The experiment results show that our simulator could suture a pre-wound soft tissue together with smooth force feedback and also allow the user to tie a knot. In our implementation, we have found that Implicit Euler's method has more advantages than simple explicit method. It does require more computational procedures to obtain the inverse of the large sparse stiffness matrix. Modern GPU computational environments may offer a suitable desktop computational platform for interactive virtual reality applications. In addition to the model of the needle deriver, tweezers and scissors are also used in the real suturing surgery, which are not implemented in our virtual reality environment. In the future, we will simulate a tweezers to allow the user to hold the soft tissue or the skin during suturing. Also, a scissor could be used for users to cut the suture into different parts after the suturing or during the suturing as in the simple interrupt suturing pattern. Our current subdivision method can cause many unwanted small triangles if the user repeatedly inserts at the same location through the soft tissue. To address this issue, one can track the movements of the needle and down sample the repeated incision points. For example, if the user pulls the needle back, the subdivided triangle should go back to its original un-subdivided configuration.

We have also integrated our virtual suturing environment with two haptic devices. The interaction forces that are computed through out the suturing task are continuously fedback to the user which can offer dominant haptic cues. The motivation for the presence of the haptic cues also encouraged by our previous studies (Vafai and Payandeh 2010) that have demonstrated the effectiveness of haptic feedback for training and interaction of the user with the virtual training environment. These computed forces are used to display the haptic sensations during the prominent changes in the contact events such as tissue tearing.

Although the proposed suturing environment can offer a more realistic environment for training, the question of effectiveness of such environment remains a fundamental challenge. Could a suturing training environment be used to measure surgical skills or could trainees improve their surgical skills are two issues that need further investigation.

References

Baraff D (1997) Implicit method for differential equations. In: Physically based modelling: principles and practice, ACM SIGGRAPH course notes

Berkley J, Weghorst S, Gladstone H, Raugi G, Berg D, Ganter M (1999) Banded matrix approach to finite element modeling for soft tissue simulation. Virt Real Res Dev Appl 4:203–212

Berkley J, Turkiyyah G, Berg D, Ganter M, Weghorst S (2004) Real-time finite element modeling for surgery simulation: an application to virtual suturing. IEEE Trans Visual Comput Graph 10(3):314–325

Bridson R, Fedkiw R, Anderson J (2002) Robust treatment of collisions, contact and friction for cloth animation. In: Proceedings of the 29th annual conference on computer graphics and interactive techniques. pp 594–603

Bro-Nielsen M (1998) Finite element modeling in surgery simulation. Proc IEEE 86:490–503

Bro-Nielsen M, Cotin S (1996) Real-time volumetric deformable models for surgery simulation using finite element and condensation. EUROGRAPHICS'96 15(3):C57–C66

Brown J, Montgomery K, Latombe J-C, Stephanides M (2001) A microsurgery simulation system. In: Proceedings of medical image computing and computer aided interventions, The Netherlands

Brown J, Latombe J, Montgomery K (2004) Real-time knot tying simulation. The Visual Comput Int J Comput Graph 20(2): 165–179

Cao DQ, Dongsheng Liu, Charles WH-T (2006) Three dimensional nonlinear dynamics of slender structures: Cosserat Rod element approach. Int J solids Struct 43(3–4):760–783

Cha JJ, Payandeh S (2007) Interactive cross cutting. In: Proceedings of IEEE international conference on robotics and automation (ICRA '07), pp 2576–2581

Chen DT, Zeltzer D (1992) Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. Comput Graph 26(2):89–98

Colgate JE, Stanley MC, Brown JM (1995) Issues in the haptic display of tool use. In: Proceedings of IEEE intelligent robots and systems, pp 140–145

Grégoire M, Schömer E (2006) Interactive simulation of one-dimensional flexible parts. In: Proceedings of the 2006 ACM symposium on solid and physical modeling, pp 95–103

Hui Z, Payandeh S, Dill J (2004) On cutting and dissection of virtual deformable objects. In: Proceedings 2004 IEEE international conference on robotics and automation (ICRA '04), pp 3908–3913

Hutchinson D, Preston M, Hewitt T (1996) Adaptive refinement for mass/spring simulation. In: Proceedings of 7th eurographics workshop on animation and simulation, pp 31–45

Kahl B, Henrich D (2005) Manipulation of deformable linear objects: force-based simulation approach for haptic feedback. In: Proceedings of 12th international conference on advanced robotics (ICAR 2005), pp 115–122

LeDuc M, Payandeh S, Dill J (2003) Toward modeling of a suturing task. In: Proceedings of Graphics Interface (GI), pp 273–279

Lenoir J, Meseure P, Grisoni L, Chaillou C (2002) Surgical thread simulation. In: Proceedings of modelling & simulation for computer-aided medicine and surgery, (MS4CMS), pp 102–107

Lenoir J, Meseure P, Grisoni L, Chaillou C (2004) A suture model for surgical simulation. In: 2nd International symposium on medical simulation (ISMS'04), pp 105–113

Lian LL, Chen YH (2006) Haptic surgical simulation: an application to virtual suture. Computer-Aided Des Appl 3(1–4):203–210

Marshall P, Payandeh S, Dill J (2005) Suturing for surface meshes. In: Proceedings of the IEEE international conference on control applications, pp 25–30

Marshall P, Payandeh S, Dill J (2006) A study on haptic rendering in a simulated surgical training environment. In: Proceedings of the symposium on haptic interfaces for virtual environment and teleoperator systems, pp 241–247

Meyer M, Debunne G, Desbrun M, Barr AH (2000) Interactive animation of cloth-like objects for virtual reality. J Vizual Comput Animat 12(1):1–12

Moody L, Baber C, Arvanitis TN (2001) The role of haptic feedback in the training and assessment of surgeons using a virtual environment. In: Proceedings of EuroHaptics

Pai D (2002) Strands: interactive simulation of thin solids using cosserat models. Comput Graph Forum 21(3):347–352

Payandeh S, Azouz N (2001) Finite elements, Mass-Spring-Damper systems and haptic rendering. In: Proceedings of IEEE international symposium on computational intelligence in robotics and automation, pp 224–230

Phillips J, Ladd A, Kavraki LE (2002) Simulated knot tying. In: Proceedings of IEEE international conference on robotics and automation, pp 841–846

Provot X (1995) Deformation constraints in a mass-spring model to describe rigid cloth behavior. Graphics Interface, pp 147–154

Robert VO, Robert RP, Thomas MK, William CB, Nancy HC, Webb RR, Whitney JB, Marc R (1999) Measuring and developing suturing technique with a virtual reality surgical simulator. J Am Col Surg 189(1):114–127

Saha M, Isto (2006) Motion planning for robotic manipulation of deformable linear objects. In: Proceedings of IEEE international conference on robotics and automation, pp 2478–2484

SensAble Technologies, Inc. (2005) penHaptics Toolkit version 2.0 PROGRAMMER'S GUIDE

SensAble Technologies, Inc. (2005) OpenHaptics Toolkit version 2.0 API REFERENCE"

Shi HF, Payandeh S (2007) Real-time knotting and unknotting. In: Proceedings of IEEE international conference on robotics and automation (ICRA '07), pp 2570–2575

Takamatsu J, Morita T, Ogawara K, Kimura H, Ikeuchi K (2006) Representation for knot-tying tasks. IEEE Trans Robot 22(1): 65–78

Terzopoulos D, Waters K (1990) Physically-based facial modeling, analysis, and animation. J Visual Comput Animation B 1(2): 73–80

Vafai N, Payandeh S (2010) Toward the development of interactive dissection with haptic feedback. Virtual Real 14(2):85–103

Wakamatsu H, Takahashi K, Hirai S (2005) Dynamic modeling of linear object deformation based on differential geometry coordinates robotics and automation. In: Proceedings of the IEEE international conference on Publication, pp 1028–1033

Wakamatsu H, Arai E, Hirai S (2006) Knotting/unknotting manipulation of deformable linear objects. Int J Robot Res 25(4):371–395

Wang C, Richardson AMD, Liu D, Rosing R, Tucker R, De Masi B (2003) Construction of nonlinear dynamic MEMS component models using Cosserat Theory. In: Proceedings of SPIE design, test, integration & packaging of MEMS symposium

Wang F, Burdet E, Dhanik A, Poston T, Teo CL (2005) Dynamic thread for real-time knot-tying. In: Proceedings of the first joint eurohaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems

Webster RW, Zimmerman DI, Mohler BJ, Melkonian MG, Haluck RS (2001) A prototype haptic suturing simulaton. Stud Health Technol Inform 81:567–569