Simulating Surgical Incisions without Polygon Subdivision

Yogendra BHASIN, Alan LIU, Mark BOWYER
National Capital Area Medical Simulation Center
ybhasin@simcen.usuhs.mil
http://simcen.usuhs.mil

Abstract. Modeling cuts, bleeding and the insertion of surgical instruments are essential in surgical simulation. Both visual and haptic cues are important. Current methods to simulate cuts change the topology of the model, invalidating pre-processing schemes or increasing the model's complexity. Bleeding is frequently modeled by particle systems or computational fluid dynamics. Both can be computationally expensive. Surgical instrument insertion, such as intubation, can require complex haptic models. In this paper, we describe methods for simulating surgical incisions that do not require such computational complexity, yet preserve the visual and tactile appearance necessary for realistic simulation.

1. Introduction

Simulation is increasingly used for surgery training. Surgical simulators allow users to improve their skills prior to operating on a patient. These systems can assist surgical training by providing different training scenarios and quantifying user performance. Making surgical incisions is a fundamental task of most procedures. Simulating incisions require the portrayal of visual and tactile effects like cutting, bleeding and instrument insertion. These cues provide important feedback. Cuts change the visual appearance of the tissue. The incision's length, depth, and orientation vary by procedure. Blood can obscure the surgeon's view, and make surgery difficult. Tissue consistency and resilience affects the amount of pressure to apply on the scalpel. Due to its importance, cutting, bleeding and instrument insertion are frequently simulated surgical effects. A number of methods have been proposed to address them. In this section, we review current research.

Many methods simulate cuts by modifying the topology of the model. Some simulators simply remove model elements from the object along the cut path [1]. This creates an uneven appearance and unrealistic gaps in the model. Subdivision methods implement cuts by splitting the polygons along its boundaries or path of incision in real-time. [2] proposed cutting surface-based meshes while re-meshing the cut to achieve the required level of smoothness. [3] used volumetric models in the form of tetrahedral mesh for cutting. While such methods can produce visually pleasing cuts, they increase the size of the mesh, resulting in increased computational burden. In addition, some methods for real-time deformation modeling rely on pre-processing the model's geometry. Cuts change the model's topology and can invalidate these schemes [4].

Bleeding is commonly modeled using mathematical representations of fluid flow or particle systems. Computational fluid dynamics methods based on Navier-Stokes equations provide accurate fluid flow models. Although some simplified forms of these equations [5] have been developed, they can be difficult to implement in real-time simulations. [6] introduced particle systems as a technique for a class of fuzzy objects. Both particle-based

Lagrangian approach [5] and grid-based Eulerian approach [7] have been used to simulate fluids. Although easy to animate, these require many particles for realism and impose high computational overhead. [8] developed a particle based texture animation method to simulate arterial bleeding. Other methods include temporal texture synthesis [9], or video capture [10]. Such schemes can be inefficient, or provide limited realism.

Modeling instrument insertion is difficult because of interactions with complex anatomical models. Many simulators focus on specific aspects of the surgical task and assume that the instruments are already inserted in the operation site [8, 11]. Some simulators rely on mannequins for the actual instrument insertion [12]. The actual process of inserting the instruments hasn't received much attention, yet this task is crucial in some procedures, such as diagnostic peritoneal lavage [13].

Many surgical procedures such as cricothyroidotomy, chest tube insertion, and diagnostic peritoneal lavage, require only small incisions be made. The cuts are small, and often do not bleed profusely. When surgical instruments such as tubes or needles are inserted, resistance is felt due to anatomical structures beneath the skin however these structures cannot be seen. In these cases, it is possible to provide realistic visual and haptic feedback without the computationally intensive methods outlined above.

2. Method

We propose an alternative way to model some surgical incisions. Rather than using polygon subdivision, we use animated texture maps and local haptic models. The underlying patient model is untouched. We have implemented this approach on a prototype cricothyroidotomy simulator currently under development [14]. Cricothyroidotomy is an emergency procedure that is performed when the patient's airway is blocked, and less invasive attempts to clear it have failed. The simulator enables medical students to practice making the incision on the throat, then inserting surgical instruments to widen and maintain the airway. In this section, we describe our method for modeling the visual and tactile effects of small surgical incisions.

2.1 Visual Feedback During Cutting

Animated and dynamic textures can allow interactive applications with little impact on the rendering performance [15]. Textures provide visually acceptable details of small scale structures. Our method is similar to [16], but has been extended to include the simulation of blood trails and oozing wounds. In our method, dynamic updates of the skin texture are implemented using the OpenGL's function glTexSubImage2D(). This function allows us to overwrite selective portion of a texture map by specifying the location of the rectangular sub-region to be replaced. Two types of textures are used, one for the wound, another for bleeding. The wound texture consists of a red spot in the center of a 5 x 5 image [Fig 1.1]. The texture is radially blurred using Adobe Photoshop so that the intensity of adjacent textures is identical at the edges. A cut is modeled by placing successive wound textures at the point of contact of the scalpel. This gives an appearance of a continuous cut [Fig 1.2].

Bleeding is modeled in two parts: the leading droplet, and the trail. Figure 1.3 shows a single blood droplet overwritten on skin. The droplet is expanded by successively replacing it with a longer drop texture [Fig 1.4]. As these drops flow down the skin, they leave behind a blood trail, giving an oozing effect. The region of the drop that joins the trail is Gaussian blurred so that the intensity of the two textures is consistent with each other and they blend together nicely [Fig 1.5].

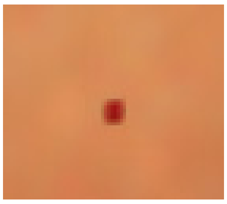

Figure 1.1. Single wound texture

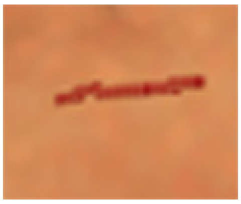

Figure 1.2. A Cut modeled using successive wound textures

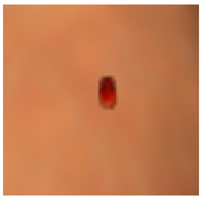

Figure 1.3. A blood droplet

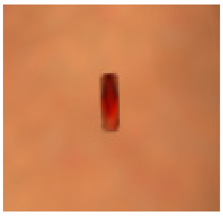

Figure 1.4. Expanded droplet

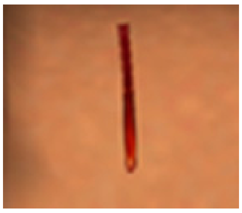

Figure 1.5. A blood drop with trail

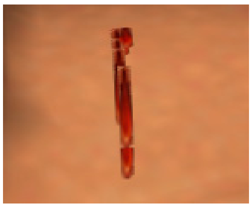

Figure 1.6. Overlapping drops with transparent regions

Blood drops and trails do not occupy the entire region of a rectangular image and have transparent regions. In the OpenGL implementation, subtextures do not blend with existing blood trace on the main texture but overwrites them. This produces undesirable effects as shown in the figure 1.6. So blood textures cannot be implemented using subtextures alone. To fix this problem, we model the droplets and the trails as RGBA images. The transparent regions of the images are identified using the alpha channel. These transparent regions are then tracked and manually rasterized using glDrawPixels() function so that the original texture is maintained on the main image [Fig. 3.1].

2.2 Haptic Feedback During Cutting

For small incisions a simplified haptic model can provide tactile fidelity comparable to more complex techniques. Unlike models that require change in the model's topology, our method uses a local model based on reaction forces as the blade is pressed into skin, and constrained motion in a plane. The haptic feedback during cutting is primarily governed by two effects. Reaction forces correspond to the tissue resistance felt as the scalpel is pressed deeper into the cut. The second effect corresponds to the blade's motion constrained by the surrounding tissue to move in the cut direction. In this section we describe how we implement these effects in our simulator.

We use the haptic interaction model provided by Reachin API [17]. Define the haptic device position as $P_{fing(t)}$ where time t is the instance after the first contact. $P_{fing(t)}$ moves freely in the space and can penetrate object surfaces. In our simulator, $P_{fing(t)}$ corresponds to the scalpel's tip. When $P_{fing(t)}$ penetrates the model, the ReachIn API defines an additional point, $P_{prxy(t)}$. As shown in the figure 2.1, this is the nearest point on the surface of the object to $P_{fing(t)}$. $P_{fing(0)}$ and $P_{prxy(0)}$ are correspondingly defined at the instance of the first contact.

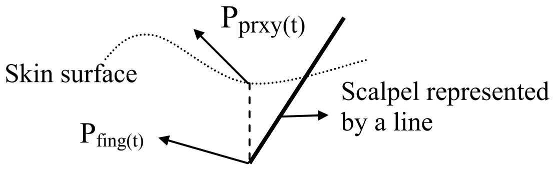

Figure 2.1: Reachin's haptic interaction model when a scalpel is penetrated inside the skin surface

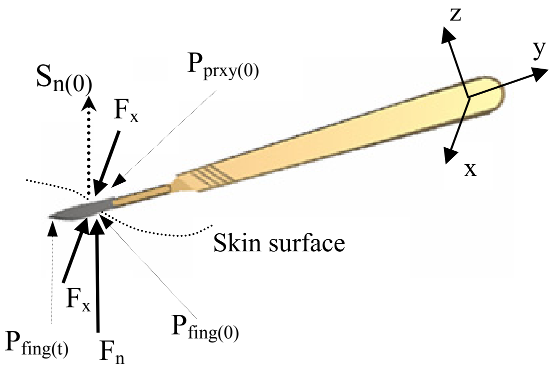

Figure 2.2: Reflected forces on the scalpel during cutting

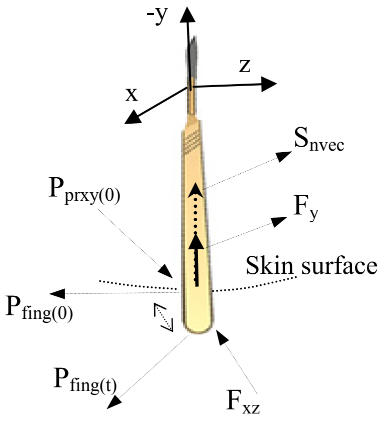

Figure 2.3: Reflected forces on the scalpel's handle during incision enlargement

Let, $S_{x(t)}$ and $S_{y(t)}$ be the unit vectors at time t along the scalpel's local x and y axis respectively. Let K_x, K_n, K_y, K_{xz} and K_d be constants set according to the desired tissue properties.

Implementing the reaction effect. Reaction forces model the resistance of the cut tissue. The user feels greater resistance as the scalpel is penetrated deeper into the skin. This force is proportional to the depth of penetration of the scalpel along the unit vector from $P_{fing(t)}$ to $P_{prxy(t)}$ denoted by $S_{n(t)}$. Thus,

$$F_{n(t)} = -K_n * [\ (P_{fing(t)} - P_{fing(0)}\ )\ .\ S_{n(t)}\ ] * S_{n(t)},\text{ where } S_{n(t)} = (P_{prxy(t)} - P_{fing(t)}\ ) / |\ P_{prxy(t)} - P_{fing(t)}\ |$$

Implementing the constrained motion effect. Once the scalpel is within the incision, the blade cannot move easily in a direction perpendicular to the cut. We implement this effect by constraining the motion of the blade in a plane containing the cut. A correction force is applied whenever the scalpel drifts from the plane. This force $F_{x(t)}$ acts in the direction perpendicular to the blade (along scalpel's x-axis) at time t. It is proportional to the distance of the scalpel from the cut-plane and is computed by projecting this distance onto the x-axis. Figure 2.2 illustrates. Thus,

$$F_{x(t)} = -K_x * [\ (P_{fing(t)} - P_{fing(0)}\ )\ .\ S_{x(0)}\ ] * S_{x(0)}.$$

Stability. The ReachIn API employs an event-driven paradigm to update system variables. For efficiency reasons, updates may not occur immediately after an event occurs. This occasionally results in inconsistent values for $P_{fing(t)}$ and $P_{prxy(t)}$ and in turn $S_{n(t)}$. Instead of a smooth reaction force, the user may feel occasional jerkiness. To reduce this effect, $S_{n(t)}$ is averaged over the past n iterations as,

$$S_{navg(t)} = \sum_{i=t-n}^{t} S_{ni} / |\sum_{i=t-n}^{t} S_{ni}\ |$$

Occasionally the user might feel certain vibrations in the system because of lower update rates. This is overcome by applying a damping force $F_{d(t)}$ proportional to the velocity $V_{(t)}$ of $P_{fing(t)}$ and is computed as, $F_{d(t)} = K_d * V_{(t)}$.

2.3 Haptic Feedback During Instrument Insertion

In cricothyroidotomy, once an incision is made, the handle of the scalpel is used to enlarge the incision so that the tube can be inserted in the trachea. As shown in the figure 2.3, the tactile feedback during instrument insertion is similarly modeled using reaction and constraining forces. We describe them in more detail.

Implementing the reaction effect. The reaction force $F_{y(t)}$ is proportional to the depth of penetration of the instrument along $S_{n(t)}$. $F_{ort(t)}$ is used to model the resistance of the instrument depending on the orientation of the handle. The instrument can be easily inserted inside the incision if it is orientated along the cut direction. However, the user feels more resistance if it is inserted in other directions. It is computed as the normalized scalar product of the handle's orientation and the cut direction. The reaction force is given by,

$$F_{y(t)} = -K_y * [(P_{fing(t)} – P_{fing(0)}) . S_{navg(t)}] * S_{navg(t)} * F_{ort(t)}$$

Implementing the constrained motion effect. The instruments inserted inside the incision site cannot cut the skin surface. However they can expand the soft tissue to a limited extent when penetrated deeper. This behavior is modeled using a constraining force that is proportional to the distance of the instrument from its initial contact position and is computed as,

$$F_{xz(t)} = K_{xz} * \{ [(P_{prxy(t)} - P_{fing(t)}) - (P_{prxy(0)} - P_{fing(0)})] . (P_{prxy(t)} - P_{fing(t)}) \} * (P_{prxy(0)} - P_{fing(0)})$$

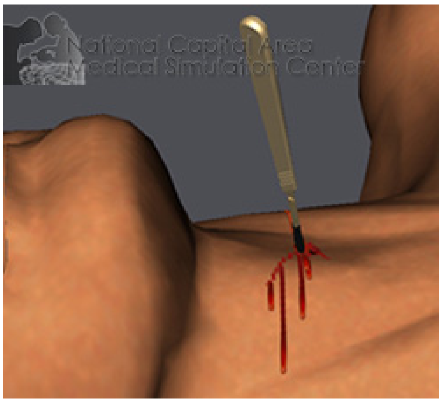

Figure 3.1. Cricothyroidotomy simulator showing bleeding due to an incision

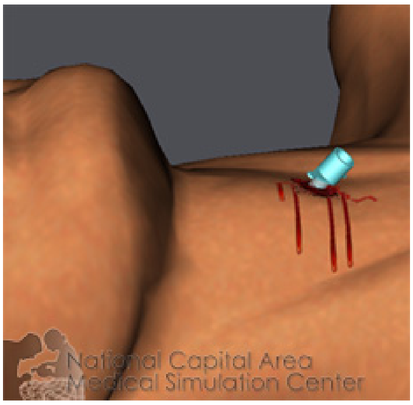

Figure 3.2. Endotracheal tube inserted inside the trachea

3. Results

We have implemented the techniques on a cricothyroidotomy simulator. In our simulator, incisions can be made at arbitrary locations. The incisions open and can bleed freely, with blood trails developing and flowing down the neck in a convincing fashion [Fig. 3.1]. The enlargement of the incision can be done using the handle of the scalpel. If an incision is made at the correct location, the insertion of a breathing tube can be performed [Fig. 3.2]. The user experiences the same tactile feedback as in the actual procedure. Our method makes no changes to the underlying model's topography. Bleeding is modeled by dynamically updating the skin with wound and blood sub-textures. Tactile feedback during incisions is modeled using reaction and constraining forces. As an initial test of validity, surgeons familiar with the procedure were invited to test the system. Many have responded favorably to the tactile and visual realism of the simulation.

4. Discussion and Conclusion

For many surgical procedures, accurate physical modeling of cutting and instrument insertion may be unnecessary. We have developed techniques that do not require computationally intensive algorithms, yet achieve similar visual and tactile effects. Initial feedback from surgeons familiar with the procedure suggests that our techniques are a viable approach. Further studies are necessary to quantify the degree of acceptance. While our method works well for small, straight-line incisions, it may not work as well for more complex procedures.

5. Acknowledgements

This work is supported by the USAMRMC under contract no. DAMD17-03-C-0102. The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

References

[1] M. Bro-Nielsen, “Finite element modeling in medical VR”, Journal of the IEEE, Vol. 86, No. 3, pp. 490-503, 1998.
[2] F. Ganovelli, C. O’Sullivan, “Animating cuts with on-the-fly re-meshing”, Eurographics, pp. 243–247, 2001.
[3] A. B. Mor, T. Kanade, “Modifying soft tissue models: Progressive cutting with minimal new element creation”, MICCAI, pp. 598-607, 2000.
[4] G. Song, N. Reddy, “Tissue cutting in virtual environments,” Interactive Technology and the New Paradigm for Healthcare, Studies in Health Technology and Informatics, IOP Press, pp. 359-364, 1995.
[5] M. Kass, G. Miller, “Rapid, stable fluid dynamics for computer graphics”, Computer Graphics, Vol. 24, No. 4, pp. 49-57, 1990.
[6] W. T. Reeves, “Particle systems - A technique for modeling a class of fuzzy objects”, Computer Graphics, Vol. 17, No. 3, pp. 359-376, 1983.
[7] J. Stam, “Stable fluids”, ACM SIGGRAPH, pp. 121-128, 1999.
[8] U. Kühnapfel, H. K. Çakmak, H. Maass, “Endoscopic surgery training using virtual reality and deformable tissue simulation”, Computers and Graphics, pp. 671-682, 2000.
[9] G. Doretto, S. Soatto, “Editable dynamic textures”, ACM SIGGRAPH Sketches and Applications, 2002.
[10] P. Oppenheimer, A. Gupta, S. Weghorst, R. Sweet, J. Porter, “The representation of blood flow in endourologic surgical simulations”, Medicine Meets Virtual Reality, pp. 365-371, 2001.
[11] LapSim, Surgical Science, www.surgical-science.com.
[12] AccuTouch Endoscopy Simulator, Immersion Medical, www.immersion.com.
[13] A. Liu, C. Kaufmann, T. Ritchie, “A computer-based simulator for diagnostic peritoneal lavage”, Medicine Meets Virtual Reality, 2001.
[14] A. Liu, Y. Bhasin, M. Bowyer, “A Haptic-enabled Simulator for Cricothyroidotomy”, To appear in Medicine Meets Virtual Reality, 2005.
[15] S. Soatto, G. Doretto, and Y. Wu, “Dynamic textures”, International Journal of Computer Vision, Vol. 51, No. 2, pp. 91-109, 2003.
[16] F. Neyret, R. Heiss, F. Sénégas, “Realistic rendering of an organ surface in real-time for laparoscopic surgery simulation”, The Visual Computer, Vol. 18, No. 3, pp. 135-149, 2002.
[17] Reachin Technologies AB, www.reachin.se.