



VIRTUAL-REALITY SIMULATOR FOR TRAINING IN MYRINGOTOMY WITH TUBE PLACEMENT

Caiwen Huang¹; Sumit K. Agrawal^{1,2}; Hanif M. Ladak¹⁻⁴

¹Department of Electrical and Computer Engineering, ²Department of Otolaryngology - Head and Neck Surgery, ³Department of Medical Biophysics, ⁴Biomedical Engineering Graduate Program, Western University, London, ON, Canada

INTRODUCTION

Myringotomy with tube insertion is a very common surgical procedure in which a tiny incision is made in the eardrum and a ventilation tube is inserted to treat middle-ear infections. As this is the first procedure that residents in Otolaryngology learn, there is great potential for damaging the delicate structures of the ear, resulting in hearing loss [1].

In the procedure, a surgeon uses one hand to hold a speculum, and uses the other hand to perform major operations such as removing wax, inserting a blade through the narrow ear canal, creating the incision and placing the tube. High levels of dexterity and experience are needed to operate in the small confines of the ear using a surgical microscope.

Surgical simulators based on virtual-reality (VR) technologies have been introduced as training tools in otolaryngology, especially for endoscopic sinus surgery [2] and for temporal bone drilling [3]. For myringotomy, several aspects of VR-based simulation were reported by the Auditory Biophysics Laboratory at Western University. A blade navigation system [4-5], deformation modeling and cutting of the eardrum [6] were implemented on different software platforms and were separately used as different training modules. The simulation of speculum and microscope movement and tube insertion have not been implemented. Therefore, a trainee still cannot practice myringotomy as a whole in one simulation.

In this paper, a newly designed VR-based myringotomy simulator, which can perform key operations such as surgical view adjustment by tilting the speculum and microscope, blade navigation through the speculum and ear canal,

cutting of a deformable eardrum model, and inserting a ventilation tube in the incision, is introduced. The cutting model is redesigned and improved, and tube insertion is implemented based on the redesigned deformation model of the eardrum.

DESIGN AND IMPLEMENTATION

Software and hardware integration

This simulator runs on a personal computer with Intel Core i7-CPU and Windows 7 operating system. A Phantom Omni haptic arm (Geomagic, Inc., Morrisville, NC) is employed for 3D positioning of the surgical instruments such as microscope, speculum, blade and forceps, and to provide force feedback, as shown in Figure 1. The simulation program is developed in Unity3D (Unity Technologies, San Francisco, CA), a leading commercial digital gaming engine. To incorporate the Omni haptic arm into Unity3D, a software plug-in [7] was adapted and imported into Unity3D.



Figure 1: Myringotomy simulator

Geometrical model

A high-resolution clinic CT is used for reconstruction of the geometric model of the ear (ear canal and eardrum only). The segmented

parts of the ear are imported into Geomagic Studio (Geomagic, Inc., Morrisville, NC) to smoothen the surfaces and then create the final model of the ear in Unity3D (Figure 2). The virtual surgical tools such as blade, forces, tube, and speculum are created in the Blender software (Blender.org, Amsterdam, The Netherlands) and input into Unity3D.

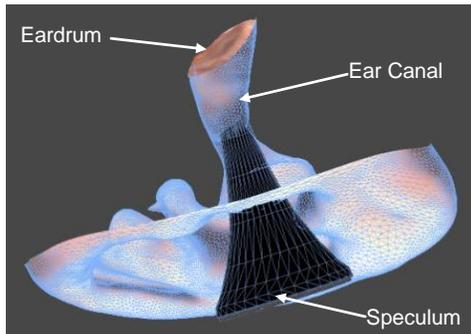


Figure 2: Geometrical model of the ear

Mass-spring system of the eardrum

To simulate the deformation of the eardrum, a mass-spring system is created (Figure3).

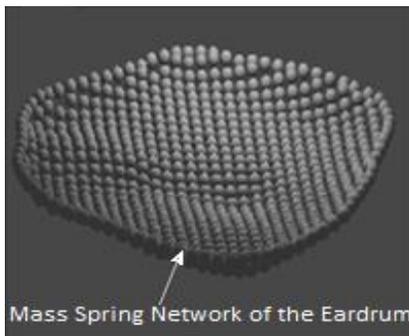


Figure 3: Mass-spring system of the eardrum

A mesh consisting of a 32 (width) × 32 (length) regular grid of triangular elements is constructed from CT data to represent the eardrum. Each node in the mesh is regarded as a point mass. To model deformations in Unity 3D, a rigid ball is placed at each node (Figure 3). The radius of the ball is slightly smaller than half of the distance between adjacent nodes. The balls are connected to each other using a “Spring Joint”, a Unity 3D component that groups together two rigid bodies and constrains them to move as if they are connected by a spring. The mass of each ball is set to a very small value because for the slow movements made during surgical procedures, inertial

effects can be ignored. The “Spring Joint” also has a damping coefficient and spring stiffness associated with it; these are set empirically by an experienced surgeon. The balls along the boundary of the spring network are fixed to simulate the physiological condition that the perimeter of the eardrum does not move.

Collision detection

To implement collision detection in Unity 3D, two types of collision detectors are designed: a ball-shaped collision detector and a mesh collision detector. Both are illustrated in Figure 4. The ball-shape detector is associated with every ball of the mass-spring network. When the blade touches a ball, the ball will move and lead to deformation of the whole eardrum mesh.

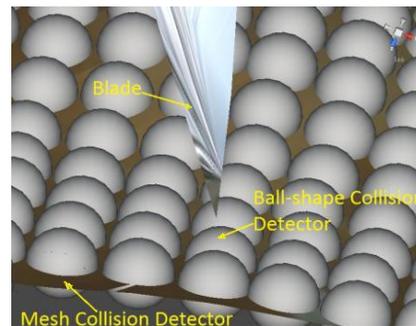


Figure 4: Collision models

The mesh collision detector is placed on the surface of the eardrum mesh, as shown in Figure 4. It is used to activate the cutting algorithm. While the ball-shaped detectors are responding dynamically and causing immediate deformation, the mesh collision detector is updated on-the-fly, and aligns to the deformed mesh.

During a simulated operation, if the blade touches a ball before penetrating the eardrum mesh, it causes a local deformation delivering an important visual cue that the eardrum has been touched. When the blade moves further, the balls are pushed away, and the blade touches the eardrum mesh collision detector and activates the cutting algorithm. During the cutting, the blade interacts with the ball-shaped collision detector and the eardrum surface detector at the same time. The balls along the cutting path successively collide with the blade causing progressive deformation of the eardrum



mesh; meanwhile, an incision path is calculated when the blade touches the eardrum mesh collision detector.

Cutting design

The subdivision cutting (SC) algorithm [8] is selected for cutting because it accurately represents the actual cutting curve. The basic SC approach splits the triangles along the sampled cutting path. For instance, in Figure 5, polyline ABCEDFG is the sampled cutting path, and the blue curve is the actual cutting curve formed by all collision points between the virtual blade and surface. The triangle LMN is split by segment AB and re-meshed by shaded triangles 1, 2, and 3. The end points in this polyline are the approximated intersections between the edges of the triangles and actual cutting curve.

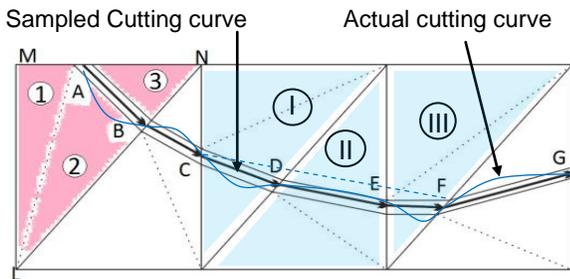


Figure 5: Illustration of subdivision cutting

The number of collision points calculated along the actual cutting curve is related to the cutting speed. If the mesh triangles are small, even at normal cutting speed, it always occurs that more than one triangle is nested between a pair of successive points along the sampled cutting curve. For example, in Figure 5, assuming the sampled cutting path is polyline ABCFG. C and F are a pair of successive end points. The points D, E are not included since there are no collision points calculated between C and F. The shaded triangles I, II, and III are nested between C and F but need to be split. To solve the situation, a cutting plane is used to intersect these nested triangles in one shot rather than splitting every nested triangle by invoking the basic SC algorithm recursively as described in Ref. [8]. This cutting plane is determined by the coordinates of the point C and a normal vector \mathbf{n} , where \mathbf{n} is defined as:

$$\mathbf{n} = \mathbf{CF} \times (\mathbf{V}_C + \mathbf{V}_F) / 2$$

where \mathbf{V}_C and \mathbf{V}_F are two unit vectors representing the axis of the blade at point C and at point F, and \mathbf{CF} is a unit vector pointing from C to F. The introduction of such a cutting plane reduces the calculation time and also maintains a smooth and reasonable cutting path, as shown in Figure 6, and this meets the requirements of real-time performance.

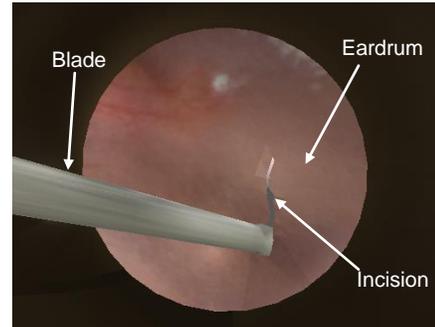


Figure 6: Demonstration of the cutting model

Speculum and microscope

Before the cutting is performed, the view of the surgical focus can be adjusted by rotating the speculum or tilting the microscope. In this simulator, the movements of the speculum and of the microscope are simulated by moving the handle of a haptic arm in 3D space. The simulated microscope can move in a small bounded box. The user feels friction effects when using the microscope. The rotation of the simulated microscope is limited to within 15 degrees. Also, the user can rotate the virtual speculum around a selected pivot to see different parts of the eardrum, as shown in Figure 8.

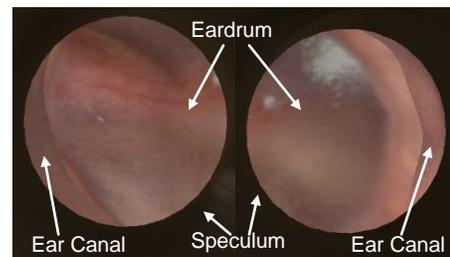


Figure 8: Focusing on different parts of eardrum

Tube insertion

Tube insertion involves grasping a ventilation tube using a forceps and navigating the tube down the ear canal to the incision in the eardrum. When grasping a ventilation tube, the opening and closing actions of the forceps

are controlled by pressing a button on the handle of the haptic arm. Both the forceps and the tube are modeled as rigid bodies. An animation is designed to simulate the forceps grasping the tube. The movement of the virtual forceps is controlled by moving the haptic arm in 3D space.

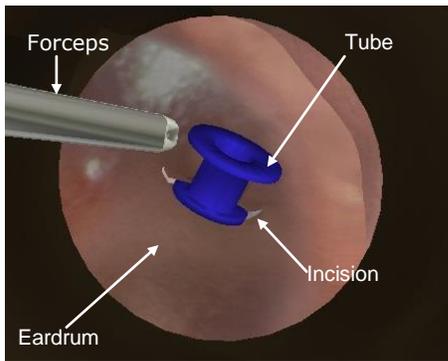


Figure 9: Demonstration of the tube insertion

The second part of the simulation involves inserting a tube into the incision after moving the tube down the ear canal. In the simulation, when the tube touches the incision, the ball-shape collision detectors interact with the tube and then cause the eardrum to deform. While the tube squeezes into the incision, the user can see the incision splaying and entering the incision, as shown in Figure 9. The simulations of the incision splaying and the interactions between the tube and eardrum are based on the eardrum physics model that is discussed above. After insertion, the user can use a simulated needle to adjust the tube into the final position. Once user uses a needle to push the tube, it rotates and slides into the incision along a pre-calculated route.

RESULTS

Qualitative feedback from instructing ear surgeons indicates that the models are anatomically accurate and contain all landmarks necessary for surgical training. Moreover, surgeons agreed that speculum placement, microscope positioning, eardrum cutting and tube insertion were simulated with sufficient realism for training purposes. All operations can be performed in real time, and the simulator developed in this work is anatomically and biomechanically realistic for surgical training.

CONCLUSIONS

A complete simulator has been now implemented. Two types of validation studies are being considered to quantify the simulator: (1) face and content validity and (2) discriminant validity.

ACKNOWLEDGEMENTS

Funding for this work was provided by the Natural Sciences and Engineering Research Council, the Ontario Research Fund and Medtronic Canada. The authors would like to thank Cavan Fyans who provided the source code to help integrate the Omni haptic arm into Unity3D [7].

REFERENCES

- [1] S. Baer, H. Williams, and A. McCOMBE, "A Model for Instruction in Myringotomy and Grommet Insertion," *Clinical Otolaryngology Allied Sciences*, vol. 15, no. 4, pp. 383-384, 1990.
- [2] S. Weghorst, C. Airola, P. Oppenheimer, C. V. Edmond, T. Patience, D. Heskamp, and J. Miller, "Validation of the Madigan ESS Simulator." *Studies in Health Technology Informatics*, vol. 50, pp. 399-405, 1998.
- [3] G. J. Wiet, P. Schmalbrock, K. Powell, and D. Stredney, "Use of Ultra-High-Resolution Data for Temporal Bone Dissection Simulation," *Otolaryngology -- Head and Neck Surgery*, vol. 133, no. 6, pp. 911-915, 2005.
- [4] L. J. Sowerby, G. Rehal, M. Husein, P. C. Doyle, S. Agrawal, and H. M. Ladak, "Development and Face Validity Testing of a Three-Dimensional Myringotomy Simulator with Haptic Feedback." *Journal of Otolaryngology: Head and Neck Surgery*, vol. 39, no. 2, pp. 122-129, 2010.
- [5] B. Wheeler, P. C. Doyle, S. Chandarana, S. Agrawal, M. Husein, and H. M. Ladak, "Interactive Computer-Based Simulator for Training in Blade Navigation and Targeting in Myringotomy." *Computer Methods Programs in Biomedicine*, vol. 98, no. 2, pp. 130-139, 2010.
- [6] A. K. Ho, H. Alsaffar, P. C. Doyle, H. M. Ladak, and S. K. Agrawal, "Virtual Reality Myringotomy Simulation with Real-Time Deformation: Development and Validity Testing," *The Laryngoscope*, vol. 122, no. 8, pp. 1844-1851, 2012.
- [7] A.C. Fyans and G. McAllister, "Creating Games with Feeling." [Website]. Available: <http://www.cfyans.com/papers/Creating Games With Feeling.pdf> [Accessed: Jan 10, 2012].
- [8] C. D. Bruyns, and S. Senger, "Interactive Cutting of 3D Surface Meshes," *Computers Graphics*, vol. 25, no. 4, pp. 635-642, 2001.