HORA 2019
www.horacongress.com

International Congress on Human-Computer Interaction, Optimization and Robotic Applications
July 5-7, 2019, Urgup, Nevşehir, Turkey

SETSCI Conference Proceedings
www.set-science.com
4(5): 15 - 19, 2019

Development of a Software Platform Based on Haptic Device for Surgery Simulators

Nazmiye Ebru Duysak[1*], Emre Dandıl[2] and Alpaslan Duysak [3]

1Computer Engineering Department/Institute Of Science, Bilecik Şeyh Edebali University, Turkey
2Computer Engineering Department/Faculty of Engineering, Bilecik Şeyh Edebali University, Turkey
[3]Department/Faculty Of Engineering, Dumlupınar University, Turkey
[*](edduysak@gmail.com)

Abstract – Haptic devices are used in a variety of virtual reality applications in order to allow user interaction with virtual objects. These devices transfer user comments such as rotations, transitions or deformations to the virtual environment and also transfer any reactions such as force feedback to the user. The force feedback which user receives is actually considered as feeling of touch.

In this work, we developed a user interface software suitable for haptic interactions. We also developed an embedded software to drive haptic device for generating realistic force feedback. Developed software traces the tip of the medical tool (attached to the haptic device) and determines any possible collision. Based on the depth of the collision the algorithm calculates the deformations on the organs and the force feedback for haptic device. The algorithm drives the three motors (one for each axes) on a haptic device to generate the necessary torque for realistic feelings of touch. The developed user interface is responsible for communications between haptic device and virtual environment, for running haptic and simulation loops and for updating and visualization of the virtual scene.

We successfully applied to develop user interface software and embedded software to a surgical simulator, where the user can manipulate and deform virtual organs and can feel any reactions from the scene.

Keywords – Haptic interaction, collision detection, deformation simulation

I. Introduction

Developing surgical simulators [1],[2],[3] becoming a very popular research area, especially with the technological development on haptic devices [4],[5]. Haptic devices are considered as robotic tools for allowing user interaction with virtual scene. The user can manipulate a surgical tool attached to the haptic device whose movements are transferred to virtual surgical tool in the simulation environment. Therefore the user can move, rotate, cut, deform or perform any surgical operations on tissue/organ models with this virtual surgical tools.

Due to the interaction between virtual tools and tıssues/ organs, some reaction forces are generated. The haptic device is working in both ways and transforms these forces to the user. The feedback forces generated by the haptic device are responsible for touch feelings. The user feels different forces when the surgical tools is touching the bone, liver or fatty tissue. Therefore, surgical simulators with haptic devices are considered more realistic instruments to be used as a part of classical medical education curriculum [6].

This work develops a user interface software which can be used for haptic interactions. The developed software includes collision detection between the tip of the virtual tool and tissue/organ, generating reaction forces, driving the haptic device and deformation algorithms.

II. Materials and Method

A block diagram of a surgical simulator is given in Figure 1. A typical simulator consists of several different units as given in the figure. Visualization module is responsible for 3D modelling, color, materials, texture mapping and updating the scene in real time. Simulation module applies deformation algorithms and performs the deformation simulation of soft tissue/organs due to any interaction. These modules are not in the scope of this paper.

Collision module is another important part of any surgıcal simulator. This module basically consists of three parts; collision detection, reaction forces and control unit. Once a collision is detected, a reaction force is generated and send to the haptic device and deformation information is sent to the deformation algorithms.

A. Collision Detection

The role of the collision detection algorithm is to detect any interaction between the tip of the vırtual tool and tissue/organ. Virtual tool is actually representing the real medical tool attached to the haptic device. Collision detection procedure firstly finds which object in the scene is in touch with the virtual tool. Then, the algorithm is responsible for finding the exact point of touch and its depth.

DOI: 10.36287/setsci.4.5.004

There are basically two types of collision. A self-collision occurs when the elements of the object are colliding among themselves. İntra-collision happens when objects are interacting with each other or with a medical tool.

Virtual objects, in this work, consists of triangles and vertices. Detecting collision implies any collision between the tip of the medical tool and these triangles. In order to speed up the detection process, there are many algorithms developed [7],[8]. The computational speed of this algorithm is very important because this module needs to work in 1kHz speed [9].

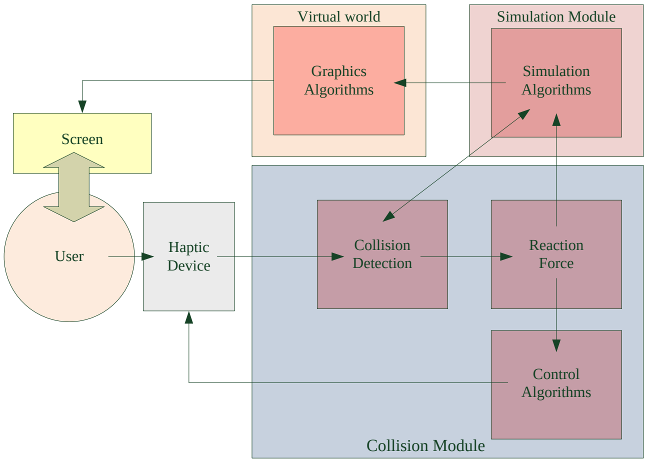


Fig.1. A typical simulator system.

Each object in the virtual scene is surrounded by a bounding box as shown in Figure 2. The axis-aligned bounding box (AABB) algorithm then basically divides the object two parts based on its long axis. Bounding boxes are assigned to the each part of these object. These two parts are again divided along the long axis. This process goes on until one element is left in each box [10].

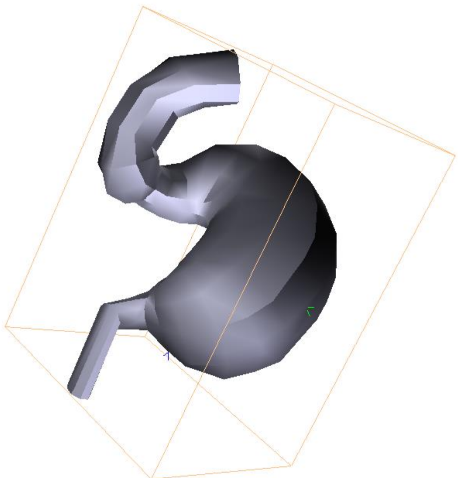
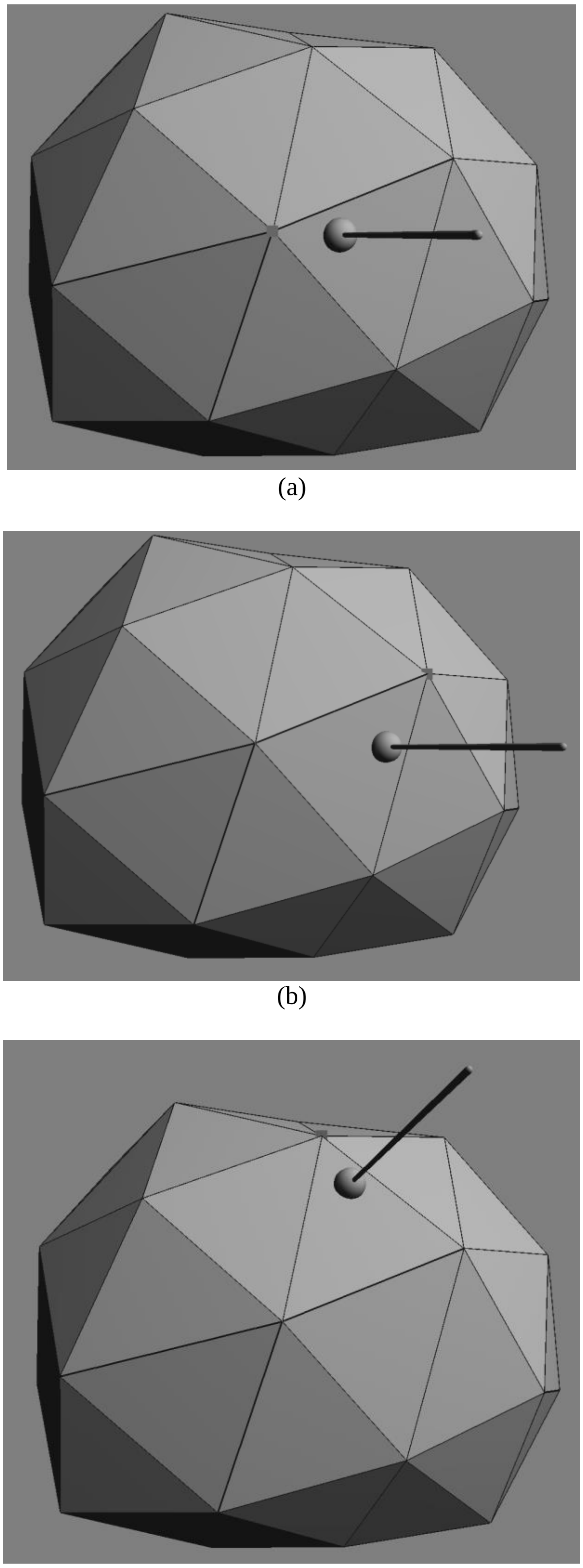


Fig.2. A virtual object in a bounding box.

The element left in each box is a triangle. The collision detection algorithm first needs to find which box is in collision with the tip of the virtual tool and then needs to find where the triangle is cut, exact collision point.

Ray triangle collision detection algorithms [11] are used to find this collision point. Once a collision point is found, the algorithm finds the closest vertex to collision point and marks that vertex as collision point.

Figure 3 shows a simple model interacting with a virtual tool. The model consists of 66 triangles and 45 vertices. As shown in the figure, for various touching points (a,b,c), collision vertices are marked with a small square.



(a)

(b)

(c)

Fig. 3. A simple model collision detection and marking collision vertices.

The other responsibilities of the collision module is to find collision depth and calculate the responses.

B. Response Forces

When the tip of the virtual tool hits the virtual object a collision occurs. If the tip point pushes/pulls the object, one needs to consider collision force [12]. This force is related to the depth of the collision. Figure 4 shows such situation.

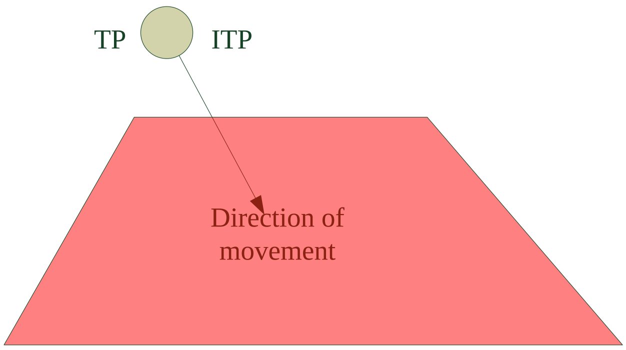
Initial tip point (TP) and ideal tip point (ITP) are shown on a 2D slice of a virtual object Fig. 4.a. When TP moves inside the object we need to consider two cases. İf the object is solid, no deformation allowed, TP should stay on the surface of the object where IPT is, as shown in figure 4.b. In this case reaction force should be at its maximum and TP moves on the location shown with ITP.

The second case considers deformable objects. When TP moves the inside the object, ITP does not stay on the surface of the object. This movement deforms the object and now ITP stays on top of deformed surface, figure 4.c.

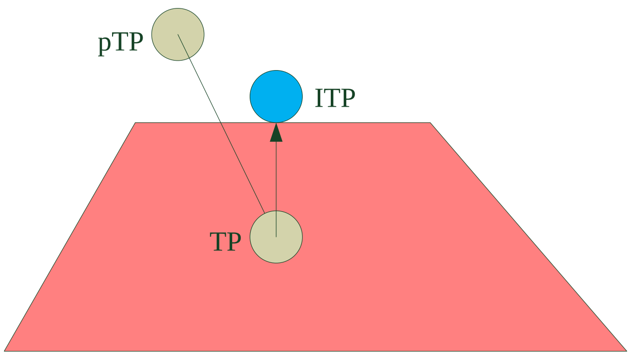
The amount of deformation is determined based on the collision depth. The depth information, figure 4.b, is used to calculate the collision force [13]. A virtual spring between ITP and TP is used to calculate this force as,

$$f_c = -K_c.X$$

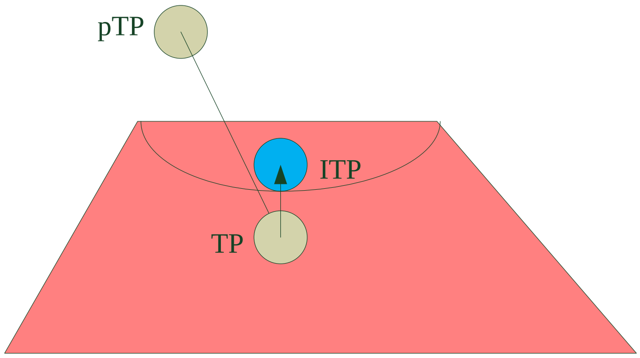
where $X = |TP - ITP|$ and K_c is stiffness of the virtual spring.



(a)

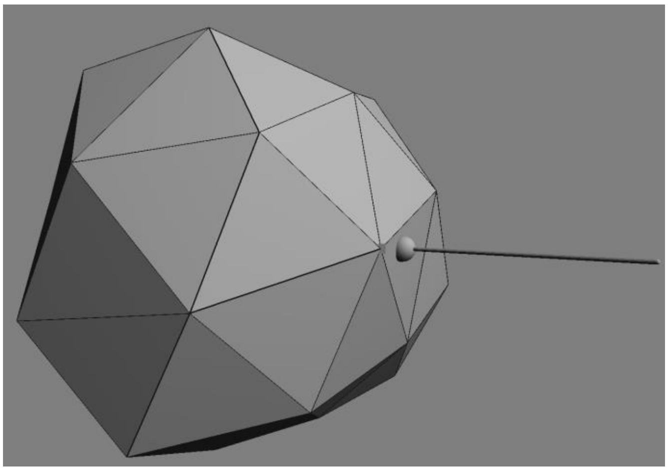


(b)

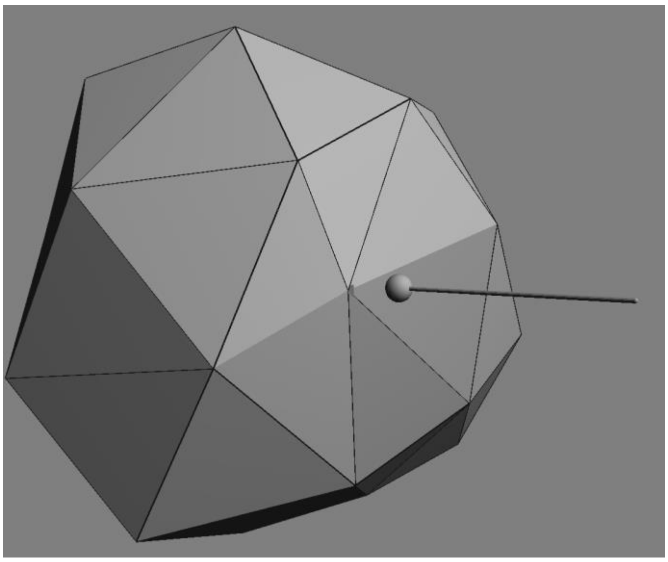


(c)

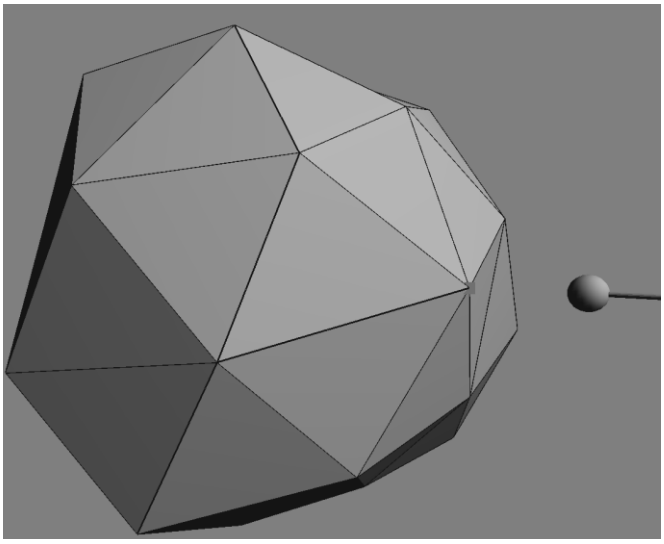
Fig.4. Collision and its responses.



(a)



(b)



(c)

Fig.5. Collision detection deformation simulation.

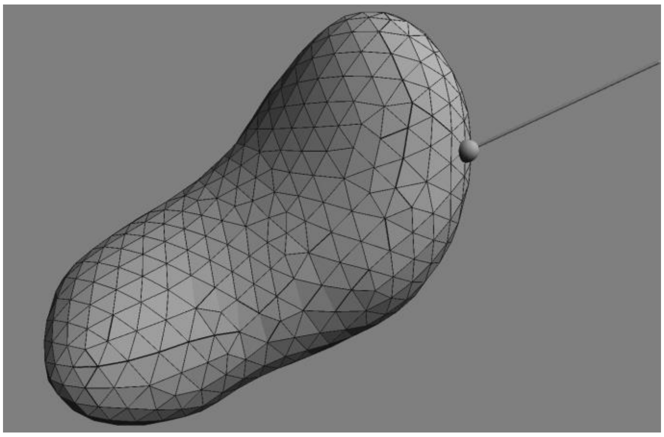
.

III. RESULTS

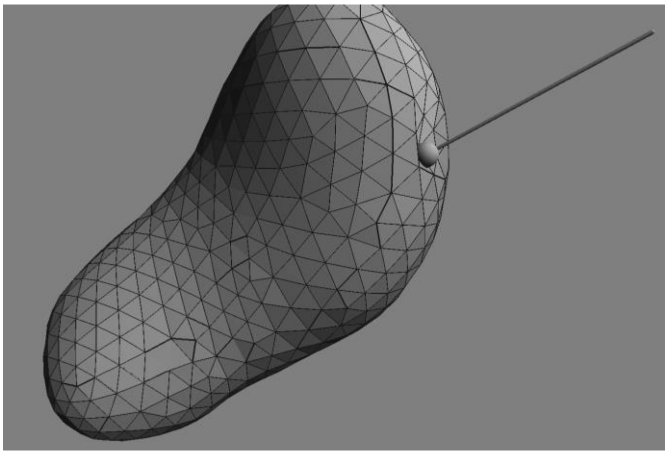
The developed haptic user interface is used with a haptic device and with a several virtual deformable objects. A simple model is interacting with a simple object is shown in figure 5. Figure 5.a shows collision point while figure 5.b and 5.c shows pushing and pulling action respectively.

A more complicated object which has 1196 triangles and 600 vertices is shown in figure 6. Figure 6.a shows start of the collision while figure 6.b shows collision with deformation. Collision detection and deformation simulation is performed in real time in these experiments.

In these experiments the correctness of collision detection (on simple model) and force feedback (with the haptic device) are tested. Visually correct deformation simulations in real time are obtained. Collision detection and force generation works well.



(a)



(b)

Fig.6. Collision detection and deformation simulation of an organ, spleen.

IV. DISCUSSION

In this work we implemented AABB method in order to speed up the collision detection process. Oriented Bounding Box (OBB) is not preferred because it requires more computations then AABB. On the other hand, if rotations are needed OBB should be used. There are different algorithms for ray-triangle intersections. In this work a fast implementation is employed. Since surgery simulators require around 1 kHz haptic rate, faster algorithms are chosen in this work.

V. CONCLUSION

This work develops a haptic user interface software for surgery simulators. Collision detection algorithm and resultant force generation method have implemented with success.

The developed software is used with a haptic device. The software is successfully transfer any user manipulation to the virtual scene and at the same time transfers, with the help of haptic device, force feedback to the user.

The detected collision also starts some deformation on the virtual organ. These deformations are also successfully simulated.

REFERENCES

[1] L. Yanping, W. Xudong, W. Fule, C. Xiaojun, W. Chengtao and S. Guofang, Development and validation of a surgical training simulator with haptic feedback for learning bone-sawing skill, Journal of Biomedical Informatics, vol. 48, pp. 122-129. 2014.

[2] N.R. Saber, V. Menon, J.C. St-Pierre, T. Looi, J.M. Drake and X. Cyril, “Development of a patient-specific surgical simulator for pediatric laparoscopic procedures.”, Stud Health Technol Inform, vol.196, pp. 360-4, 2014.

[3] X. Chen, P. Sun and D. Liao, “A patient-specific haptic drilling simulator based on virtual reality for dental implant surgery.”, Int J Comput Assist Radiol Surg, 13.11 , pp.1861-1870, 2018.

[4] T. Wang, B. Pan, Y. Fu, S. Wang and Y. Ai, “Design of a new haptic device and experiments in minimally invasive surgical robot”, Computer Assisted Surgery,vol. 22, pp.240-250, 2017.

[5] D. Escobar-Castillejos, J. Noguez, L. Neri, A. Magana and B. Benes, “A Review of Simulators with Haptic Devices for Medical Training”, Journal of Medical Systems, 40.104, 2016.

[6] J. T. Mills, H. Y. Hougen, D. Bitner, T. L. Krupski and N. S. Schenkman, “Does robotic surgical simulator performance correlate

with surgical skill?." Journal of surgical education, 74(6), (2017), pp.1052-1056.

[7] S. Kockara, T. Halic, K. Iqbal, Senior Member, IEEE, C. Bayrak, and Richard Rowe, "Collision Detection: A Survey". IEEE International Conference on Systems, Man and Cybernetics, pp.4046-4051, 2007

[8] R. R. Man, D. S. Zhou, Q.Zhang, "A Survey of Collision Detection." Applied Mechanics and Materials, vol.538, pp. 360-363, 2014.

[9] J. Hu, C. Chang, Y., N. Tardella, J. Pratt, and J. English, Effectiveness of haptic feedback in open surgery simulation and training systems. Stud. Health Technol. Inform, vol.119, pp. 213-218, 2006.

[10] A. Majercik, C. Crassin, P. Shirley, M. McGuire, A Ray-Box Intersection Algorithm and Efficient Dynamic Voxel Rendering, Journal of Computer Graphics Techniques Vol. 7, 2018.

[11] T. Möller and B.Trumbore, "Fast, minimum storage ray-triangle intersection," Journal of Graphics Tools, 2(1), pp. 21-28, 1997.

[12] H. Maass, B. B. A. Chantier, H. K. Çakmak, C. Trantakis and U. G. Kuehnapfel, "Fundamentals of Force Feedback and Application to a Surgery Simulator," Computer Aided Surgery, 8.6, pp. 283-291, 2003.

[13] P. Kadleček, Haptic rendering for 6/3-DOF haptic devices, Master Thesis, 2014.