The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Modeling and Deforming Virtual Dense Elastic Object with Haptic Device PHANToM

Hiroshi TAKADA, Norihiro ABE

Kyushu Institute of Technology, 680-4 Kawazu, Iizuka-shi, Fukuoka, 820-8502, JAPAN
(Tel : 81-948-29-7776)
(takada@sein.mse.kyutech.ac.jp)

Yoshimasa KINOSITA

Munakata Suikokai General Hospital
341-1, Kamisaigou, Fukutsu-shi, Fukuoka, 811-3298, JAPAN

Tatsushi TOKUYASU

Oita National College of Technology
1666, Oaza-maki, Oita-shi, Oita, 870-0152, JAPAN

Hirokazu TAKI

Wakayama University
930, Sakaedani, Wakayama 640-8510, JAPAN

Shoujie He

VuCOMP
1231 West Campbell Road Richardson, TX 75080, USA

Abstract: In the recent years there have been various problems in medical treatments, of which the human error by the surgeon in an operation is one of the most serious of problems. In order to minimize the human error in an operation, we need a medical training system by which an inexperienced surgeon can try operating again and again to improve his skill. In this research, we construct the system of modeling a virtual dense elastic object and deforming the object using a haptic device called PHANToM. In the system which we construct, we use two PCs to distribute the process of calculation and SCRAMNet+ is used to connect each PC. PHANToM is used to operate the object and to express the force which is generated from the deformation of the object. We represent the dense object by using Voxels and Tetrahedrons, and the elastic object by using a spring-mass model. A virtual dense elastic object is obtained from CT or MRI to express each patient's organs.

Keywords: Virtual Reality, Simulation, Rendering, Deformation, Haptic Device

I. INTRODUCTION

In the recent years, there have been various problems in medical treatments, of which the human error by the surgeon in an operation is one of the most serious of problems. The major cause is considered to be the insufficient experience of surgeons. A great deal of experience is necessary in medical operation, and tactile or haptic sensation such as manual sensation becomes important to prevent mistakes. It is, however, impossible to use real human body for practicing medical operation. Therefore simulation of the medical operation with a sense of reality as in a real operation is required.

We have studied about the cutting operation using surface model (Koichi [1]), and deformation of surface model (Ryuichirou [2]) and synchronization between audiovisual and haptic feeling (Yoshihiro [3]) in previous research of our laboratory to construct the medical operation training system.

In this research, we approach the construction of the medical operation training system, by creating a virtual human organ model which is not a rigid object but a flexible object which is deformable. The data of the human organ is obtained from CT or MRI.

II. SYSTEM CONFIGURATION

This system consists of two PCs connected with SCRAMNet+: one PC (PC1) renders a virtual dense elastic object with Open GL and the other PC (PC2) calculates haptic feedback given to an operator through

PHANToM as shown in Fig.1. In this way, we can distribute the process of calculation.

The flowchart of entire process is shown in Fig.2. We use SCRAMNet+ to share the information between PC1 and PC2. Each PC writes and loads the information which is stored in the SCRAMNet.

PHANToM has the original coordinates against the coordinates of the virtual space created with OpenGL. Therefore, we have to multiply the position of PHANToM by the rotation matrix to match the coordinates of the virtual space with that of PHANToM.

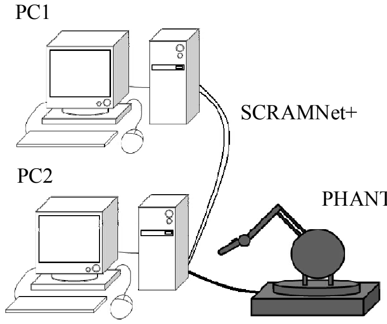

Fig.1. System configuration

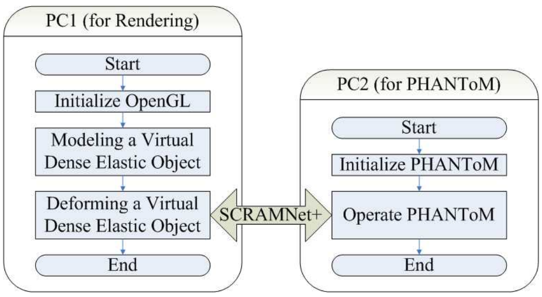

Fig.2. Flowchart of entire process

III. MODELING A VIRTUAL DENSE ELASTIC OBJECT

Rendering a virtual dense elastic object requires the system of drawing the inner tissue obtained from the CT or MRI. In order to construct the virtual dense elastic object, at first Voxels are obtained by dividing Geometry and then Tetrahedrons are generated from each Voxel as shown in Fig.3. We can create a virtual dense elastic object when we set color information (RGBA) to each Tetrahedron. We use CT or MRI data to express any human organs. In order to create the CT or MRI data in original format which can load in our system, we use OpenGL Volumizer.

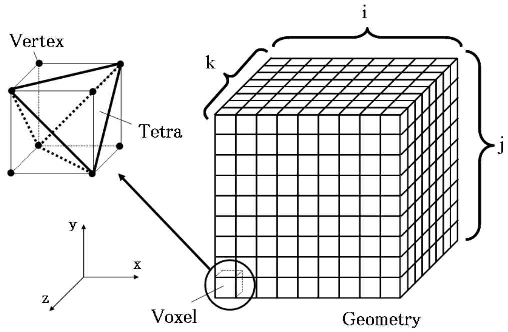

Fig.3. Virtual dense elastic object

1. High Performance of Rendering Process

We set RGBA value to each Tetrahedron as color information to render a virtual dense elastic object. If the partition number of Voxel increases, the number of Tetrahedron also increases and the process of rendering all Tetrahedrons results in bad performance. Therefore we render only the visible part of a virtual dense elastic object from user's viewpoint to improve the performance. The inner object is invisible to the user, so it is not necessary to render. We make a list of visible Tetrahedrons, and we enable high performance of the rendering process by using the list.

2. Spring-Mass Model

A spring-mass model is a model which is a set of the massless spring and the point mass. As shown in Fig.4, we apply the spring-mass model to each Tetrahedron (Koichi [4]).

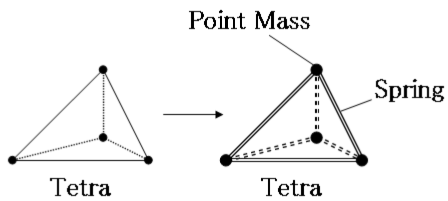

Fig.4. Spring-mass model

We replace each side of Tetrahedron by spring and each Vertex of Tetrahedron by point mass. Tetrahedrons share their Vertices and if the Vertices move, the Tetrahedron deform. So we can represent the deforming process of a virtual dense elastic object as the movement of the Vertices. A spring-mass model is used to realize the deformation based on mechanics.

IV. DEFORMING A VIRTUAL DENSE ELASTIC OBJECT

A virtual dense elastic object must be deformed when it is cut or pushed with a medical tool and the effect must be given to the operator. A medical tool is represented as a rigid stick by PHANToM.

1. Deforming Process

The information such as position, acceleration and force are stored on each Vertex. When a Vertex is moved with a medical tool, a force of spring and damper is induced by the movement of Vertex and the force acts on the connection between Vertices. From the velocity and the displacement between Vertices connected with springs and dampers, each force of the Vertex can be expressed as an equation (1). Here an operator is given the force which is determined from the equation (1) through PHANToM.

$$F_i = \sum \left(\frac{l_{ij}}{|l_{ij}|} k_{ij} \left(|l_{ij}| - |l_{0ij}| \right) + c_{ij} v_i \right) \tag{1}$$

Then we compute the motion equation to obtain each position of the Vertex using the force obtained from the equation (1). Euler method is used to solve dynamically characteristics of a spring-mass model using a motion equation. Then information stored on each Vertex adjacent to the moved Vertex is recomputed and the entire object will deform.

2. Collision Detection of PHANToM

Collision is detected between a Vertex and PHANToM in order to choose the Vertex which is moved with PHANToM. However if we use the position of PHANToM to detect a collision, we have only to detect collision at just one point. The medical operation training system, however, requires operation using a medical tool such as scissors or scalpel. So we make a medical tool as a line made from the position of PHANToM and the angle of rotation of PHANToM. If a deformable dense object is moved with a rigid stick such as a medical tool controlled with PHANToM, the same computation is conducted recursively for all Vertices on the object colliding with the medical tool. Here we explain about collision detection between a Vertex and a medical tool made from PHANToM.

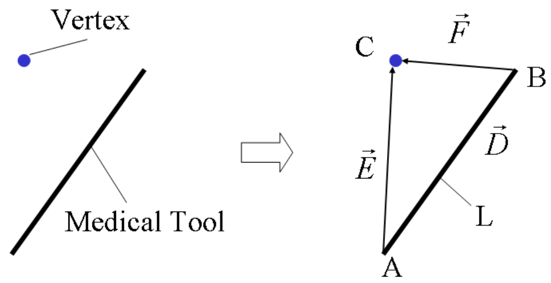

Fig.5. Vectorization of the collision detection model

We use a vector to detect the collision. As shown in Fig.5, we label the edge of a medical tool as *AB* and length of the medical tool as *L*. Also we label the Vertex as *C*. Then we can express the vector *AB*, *AC*, and *BC* as $\vec{D}$, $\vec{E}$ and $\vec{F}$ respectively. Collision is detected when *C* is on the segment *AB*. It is possible to judge whether *C* is on the line *AB* or not by checking whether $\vec{D}$ and $\vec{E}$ are parallel or not using a cross product of the vector. A normal vector can be obtained from a cross product. If the result of the normal vector is zero, then $\vec{D}$ and $\vec{E}$ are parallel, that is to say *C* is on the line *AB*. Next, if *C* is on the line *AB*, check the length of $\vec{E}$ and $\vec{F}$, and if they are less than or equal to *L*, *C* is on the segment *AB*.

3. High Performance of Deforming Process

The process of deforming a virtual dense elastic object requires high computational power when we calculate all Vertices. If the number of Tetrahedron increases, same as rendering process, the process of deformation results in a bad performance. Therefore we calculate only the Vertices of the Tetrahedron which has the color information. We make the list of Tetrahedrons which has the color information and use the list for calculation. Additionally, in order to reduce the computational load of the collision detection process, we search only the Vertices of the Tetrahedron which construct the surface of the object. We use the visible Tetrahedron list which we make in the rendering process for collision detection.

V. EXECUTION RESULT

1. Modeling of a Virtual Dense Elastic Object

A virtual dense object restored from CT is shown in Fig.6 and 7, where the Geometry is (1.0, 1.0, 1.0) and the number of partition of Voxel is (120,120,70). The entrails appear as shown in Fig.6 if the threshold amount is changed. This fact shows that a virtual dense object is successfully implemented. Fig.7 shows rendering the data of the blood vessel and it also shows that it is able to rotate.

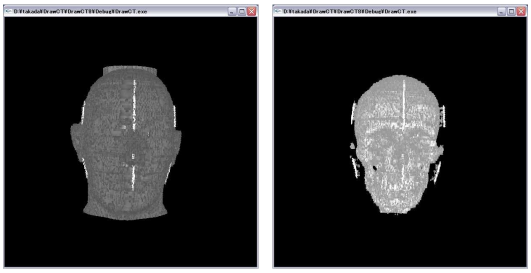

Fig.6. Rendering head

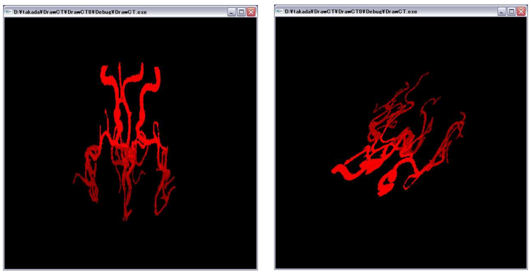

Fig.7. Rendering blood vessel

2. Deformation of a Virtual Dense Elastic Object

Fig.8 shows the situation in which a blood vessel fixed at both ends is deformed to see the invisible object behind the vessel, where Geometry is (0.1, 1.0, 0.1) and the number of partition of Voxel is (1, 10 ,1). The vessel is deformed with a medical tool controlled with PHANToM. In this case, we confirm that the adequate haptic feedback is returned to operator's hand. Additionally, we confirm that the object gets back to its original shape.

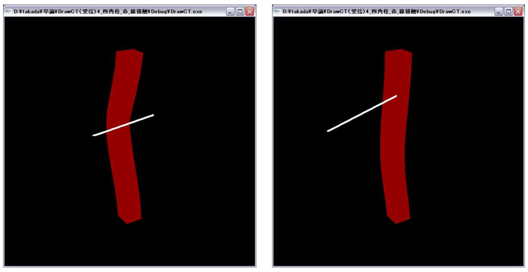

Fig.8. Deforming blood vessel

VI. CONCLUSION

We construct the system of rendering a virtual dense elastic object obtained from CT or MRI data and expressing deformation of tissue in this research to achieve the construction of the medical operation training system. Then we confirm the force generated from deformation of tissue and the real time rendering of a virtual dense elastic object.

However, the size of a virtual dense elastic object which can be deformed is too small to operate as a real medical operation. More the number of Tetrahedrons to be rendered, lesser is the frame rate. In a real medical operation, as the target tissue is only a portion of whole organ, it is sufficient to transform only the target portion into a set of Tetrahedrons and retain the rest as it is or model it roughly to reduce computational load. Also, in this research, we just confirm the force obtained from computing the amount of deformation. In the future, we would like to give a real haptic feeling to an operator. At present, cutting operation with a scalpel or scissors are not implemented. We would like to develop a medical training system in which an inexperienced surgeon is allowed to operate again and again by using a virtual model obtained from each patient's CT or MRI.

ACKNOWLEDGMENTS

We greatly appreciate the aid of the Grant-in-Aid for Scientific Research (S) and (A).

REFERENCES

[1] Koichi Y, Kazuaki T, Norihiro A, et al. (1999), Cutting operation of virtual object and it application to medical simulation.. ICAT'99. 161-165.
[2] Ryuichirou M, Norihiro A, Yoshimasa K, et al. (2007), Simulation of ICSI Procedure Using Virtual Haptic Feedback Model. CME2007. 176-181.
[3] Yoshihiro T, Norihiro A, Hirokazu T, et al. (2008), Synchronization between Audiovisual and Haptic Feeling for Constructing Edutainment Systems. Journal of Virtual Reality. Vol.12, No.1, 27-36.
[4] Koichi S (2006), Introductory Book of Making Dynamic Animation with OpenGL (in Japanese). Morikita Publishing Co, Tokyo, pp 126-152