Octree-based Virtual Dental Training System with a Haptic Device

H. T. Yau[1], L. S. Tsou[2] and M. J. Tsai[3]

[1]National Chung Cheng University, imehty@ccu.edu.tw
[2]National Chung Cheng University, lstsou@cad.me.ccu.edu.tw
[3]National Chung Cheng University, joybe@cad.me.ccu.edu.tw

ABSTRACT

The virtual dental training system using a haptic device is becoming a very important training tool for dental education. We propose a new approach that utilizes the so-called *EP* (Edge Proportion) value, local material stiffness and implicit function to develop a novel dental training system which has useful characteristics such as efficient collision detection, stable haptic interaction and accurate sculpting simulation. In the efficient collision detection approach, the oriented bounding box tree and *EP* value can be quickly updated for collision detection. The *EP* value is very important in our approach, because it can also be used to compute haptic force and update the tooth model. In the stable haptic interaction approach, we use the spring-damper force model and a simple force filter to obtain stable force feedback. Finally, in the precision sculpting simulation, an implicit function is employed to exactly represent various kinds of dental tools so that accurate calculation of cutter-surface intersection can be realized. The proposed system is highly applicable to virtual dental training and practices. Examples of tooth preparations such as inlay and bridge preparation are demonstrated.

Keywords: Dental Simulation, Haptic Rendering, Collision Detection

1. INTRODUCTION

Tooth preparation is a very important skill for dentists because it establishes the foundation for whatever restoration is being placed. Dental students traditionally use artificial teeth to practice these procedures, but the artificial models cannot supply the level of detail and material properties of realistic tooth preparation. Thus, Dental students must learn these skills with an experienced dentist while practicing on real patients. Both training types are time-consuming, expensive and error-prone. Computer simulation using a haptic device becomes a feasible training system since it provides touch sensation, which is important to any realistic tooth preparation training. The amount of applied force is very important since too much force will damage the tooth surface enamel [1]. Thus, using appropriate force is an essential skill in dental practices. Fortunately, a haptic system is a kinesthetic link between a human operator and a virtual environment, so a force feedback dental simulator can provide realistic touch sensation for students to practice these skills.

If a haptic dental training system is to replace the traditional systems to help dental students practice tooth preparation, the stability, reality and accuracy of this system become very important functional requirements. Based on these requirements, an octree-based haptic dental training system is proposed with the following characteristics:

- *Efficient Collision Detection:* The ability to efficiently detect collisions is an important aspect for computing responsive force and sculpting tooth model. Efficient and exact interference detection can enable the system to approximate real touch sensations more realistically. In this paper, to save memory space while preserving sufficient modeling accuracy, a tooth model is represented by an adaptive octree data structure. The trajectory of the cutting tool is generated by immediate and arbitrary tool motion conducted by the user. Thus, we can use the oriented bounding box (OBB) tree to express the boundary of a cutting tool. Utilizing the fast overlap test for OBBs [3], one can efficiently and exactly detect which voxels of the tooth model contact with the cutting tool.
- *Stable Haptic Interaction:* The responding force is calculated by a function that takes into account all the collision voxels, and each force of voxels is calculated by three axial forces. Utilizing this approach we obtain more precise and stable force than using only one point of the virtual tool for collision detection and force computation. Even so, the force rendering may still lose continuity in touch sensation. Due to this reason, we apply a filter to produce smoother force signal before sending it to the haptic device. Thus, a more stable force response is generated.

- *Accurate Sculpting Simulation:* Dental cutting tools come with many different shapes. Commercial haptic systems often have problems dealing with generalized cutters. We use an implicit function to represent the dental tool because a cutter with various geometric shape and orientation can be exactly represented by implicit algebraic equations. This really helps the system become more accurate in sculpting and more realistic in simulation.

The content of this paper is organized as follows. In Section 2, we give a brief survey of related work. Section 3 expounds on our system architecture and methodology. In Section 4, we describe the implementation of the dental training system and the results. At the end, a conclusion is made in Section 5.

2. RELATED WORK

2.1 Medical Haptic Simulation/Training System

A common application of these devices is to simulate touch sensation for learning physical skill, for example dental training. Thomas et al. [1] developed a dental surgical simulator, and they describe the software and implementation of the prototype system. Wang et al. [2] used a piecewise contact force model to approximately describe the cutting process and used force control to maintain stability of the haptic device. Kim et al [4] integrated visual, auditory and haptic sensations to develop a dental training system and used volumetric implicit surface for surface modeling and haptic rendering while sculpting. A haptics-based high-resolution augmented virtual reality system which displays and manipulates three-dimensional data for training and simulation was developed by Luciano et al. [5]. Petersik et al. [6] present a haptic rendering algorithm, based on a multi-point collision detection approach which provides realistic tool interactions.

2.2 Haptic Interfaces

A haptic interface is a kinesthetic link between a human operator and a virtual environment. The reviews of haptics literatures can be found in [7,8]. Hayward and Astley [9] set out requirements and guidelines for performance measurement of haptic device. Adams and Hannaford [10] discussed the fundamental stability and performance of haptic interaction. Miller and Zeleznik [11] defined a set of robust haptic principles and techniques that can be applied to haptic interface.

2.3 Volume–based Sculpting

How to efficiently remove the materials according to the interference between the moving cutter and tooth is another issue that must be considered. McDonnell et al. [12] used a dynamic subdivision solids approach to develop a real-time interactive sculpting system. Galyean and Hughes [13] proposed a voxel model to represent a solid material and used marching-cube algorithm to convert a voxel model to a polygonal surface. The CT data can be rendered by volume visualization technique, and therefore Avila and Sobierajski [14] presented a haptic interaction method that was suitable for both volume visualization and modeling applications.

3. SYSTEM ARCHITECTURE

We integrated the PHANTOM Haptic interface with 3D stereo display to develop the octree-based haptic dental simulation system. The octree-based techniques can help achieve stability, reality and accuracy of this system. The system is composed of several components. An overview of our system architecture is given in Fig. 1. and we divide the system into three main loops of haptic loop, simulation loop and display loop.

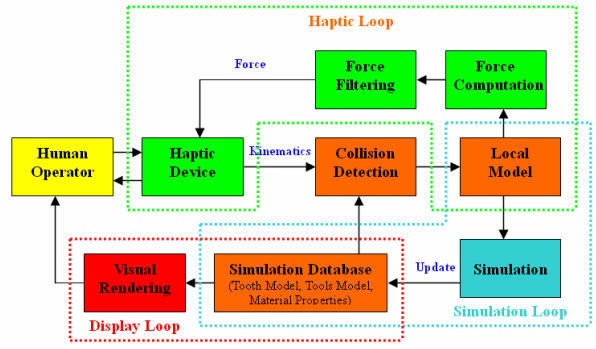

Fig. 1. Architecture of the virtual dental training system.

The haptic loop reliably maintains an approximate 1KHz force update rate to compute the stable interaction force between the haptic device and the virtual environment when the tool model contacts with the tooth model. The detail is described in Section 3.6 and 3.7. The simulation loop updates the tooth model by modifying the shape of the tooth when materials are removed form the surface of the tooth. The detail is described in Section 3.5. According to Fig. 1., the collision detection algorithm is the foundation of the haptic loop and the simulation loop. It is critical in our approach because it calculates all of the contact points, which are then utilized to compute the interaction force and update the tooth model. Furthermore, this algorithm must compute as fast as possible.
The display loop maintains a refresh rate at 30Hz to compute the surface mesh of the tooth shape for real-time rendering and display.

3.1 Tooth Representation

In this paper, we use a voxel data structure to represent the free form solid geometry of teeth because a voxel model has axis alignment and view-independent properties. At the same time the octree data structure is used to avoid creating a large number of voxels. Traditionally, since sculpting uses the uniform voxel data structure to represent an object, when precision increases, great quantity of voxel data will be produced to represent the object. This will reduce the sculpting performance because a lot of computer memory is needed.

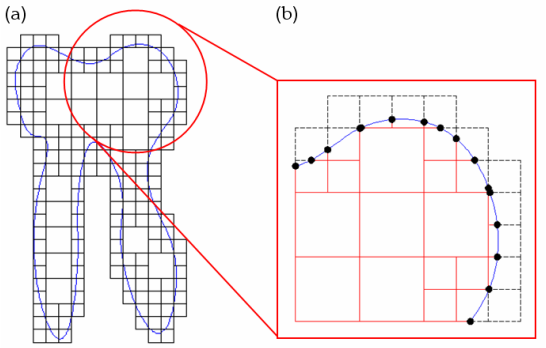

Fig. 2. Tooth representation. (a) Adaptive voxel model. (b) *EP* value.

Therefore, the tooth model with adaptive octree data structure is created by subdivision and complete distance field representation (CDFR) approach [15] with specified resolution. We use the CDFR approach to compute a high quality distance field that facilitates accurate surface characterization. Fig. 2(a). shows adaptive voxels to created a tooth model by subdivision and Fig. 2(b). shows distance fields computed by CDFR. If the voxel vertex is located outside the tooth model, the potential value is positive. On the contrary, if it is located inside the tooth model, the potential value is negative. Therefore, the black circles are the points of tooth surface. Nevertheless, we do not record the distance fields at vertexes in our voxel data, but record the proportion of voxel edge instead. The proportion of a voxel edge that is called the *EP* value in this paper can easily be computed by a linear interpolation equation in Eqn. (1).

$$F(I) = F(I_s) + (F(I_e) - F(I_s))(\frac{I - I_s}{I_e - I_s}) \tag{1}$$

where *Is* and *Ie* are potential values at the start and end points of a voxel edge, and *I* is also a potential value at the interpolation point. *F(Is)*, F(*Ie)* and *F(I)* are the *EP* value at start, end and interpolation point respectively as shown in Fig. 3.

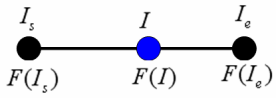

Fig. 3. Linear interpolation of *EP* values.

Then, we can define the *EP* value at *Is* as zero and at *Ie* as one. Furthermore the potential value at the interpolation

point is always zero. Thus, Eqn. (1) can be reduced to Eqn. (2). In order to distinguish between the outside edge and inside edge, we define the positive sign to express the inside edge and negative sign the outside edge. The sign can be checked by the potential values at the start point (*Is*). If *Is* is positive, then the *EP* value is negative; otherwise, the *EP* value is positive.

$$EP = \frac{-I_s}{I_e - I_s} \tag{2}$$

Recording the proportion of voxel edge has some significant benefits as follows:

- Extensively fit for any kind of implicit function when the cutter changes size or type.
- Easy to exactly compute the triangle mesh of tooth geometry for virtual rendering by using a marching cubes algorithm.
- Highly efficient for computing the stable interaction force between haptic device and virtual tooth model.

Fig. 4. shows the tooth representation in adaptive voxel model and its polygonal model.

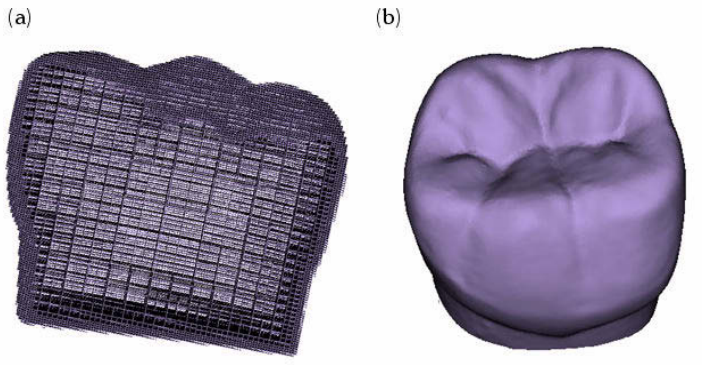

Fig. 4. Tooth representation. (a) Adaptive voxel model. (b) STL model extracted from (a).

3.2 Tool Representation

We use implicit functions to represent the various dental tools [16] because a cutter can be easily and exactly represented by implicit algebraic equations, and judging whether the cutter keeps in contact with the tooth model is also easy. The cutter's position and orientation change very quickly in dental training system. Therefore, when the cutter is represented by a geometric model, such as an STL model, the translation and orientation transformations must be applied to the cutter or the tooth model at all times. This would be computationally expensive as in 5-axis NC simulation. Instead, in this system, we use implicit functions to represent a cutter. When a user moves or rotates the cutter, only the implicit functions get changed and it is highly efficient and accurate.

3.4 Collision Detection Algorithm

The collision detection algorithm detects collisions between the tooth model and cutter model and is also the foundation to compute force feedback and update the tooth model. Fig. 5. shows its flow chart, and the detail is described as follows:

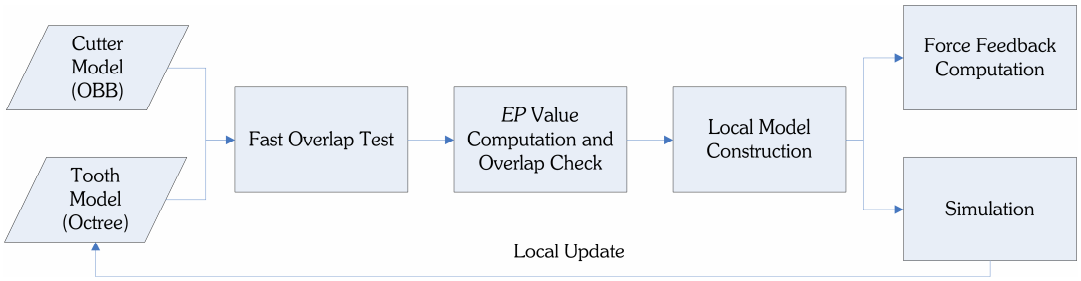

Fig. 5. Flow chart of collision detection algorithm.

I. The first step is to quickly detect the voxels that do not have contact with the cutter. Because the cutter's axis changes constantly, we use the octree and the cutter's bounding box to efficiently eliminate the non-contact voxels. Then, a fast overlap test is used to obtain voxels that may have contact with the cutter.

II. The second step is to utilize the implicit function of the cutter to compute the *EP* value. This value can correctly find which voxels are inside the cutter. Based on the potential values at *Is* and *Ie*, there are four kinds of contact situations between the voxel edge and the cutter as follows:
   - When *Is* and *Ie* are positive, the *EP* value of this voxel edge is set as one. It means that the voxel edge is located outside the cutter so this edge must be reserved as shown in Fig. 6(a).
   - When *Is* and *Ie* are negative, the *EP* value is set as zero. It means that the edge is located inside the cutter so this edge must be removed as shown in Fig. 6(b).
   - When *Is* is positive and *Ie* negative, the *EP* value is computed by Eqn. (2) and its sign is positive. It means that the partial edge is located outside the cutter so the proportion of this edge must be reserved as shown in Fig. 6(c).
   - When *Is* is negative and *Ie* positive, the *EP* value is also computed by Eqn. (2) and its sign is negative. It means that the partial edge is located inside the cutter so the proportion of this edge must be removed as shown in Fig. 6(d).

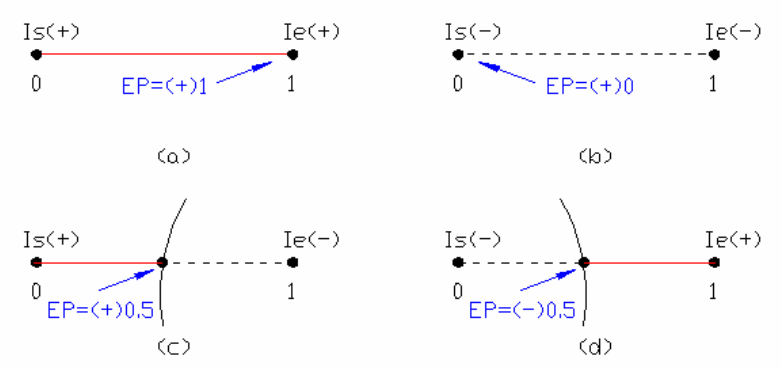

Fig. 6. Contact situations between cutter and voxel edge.

III. The final step is to construct the local model using all contact voxels. The local model is used to compute the force feedback and update the tooth model.

3.5 Simulation Algorithm

Fig. 7. shows the flow chart of the simulation algorithm. When the local model is constructed by the collision detection algorithm and the material removed, the first step of the simulation loop is to check the resolution of the local voxel model as shown in Fig.8(a). Then, the subdivision will be executed until the resolution is satisfied as shown in Fig.8(b). Finally, the local model's current *EP* value is utilized to modify the tooth geometry and the tooth model has been sculpted at the same time as shown in Fig.8(c).

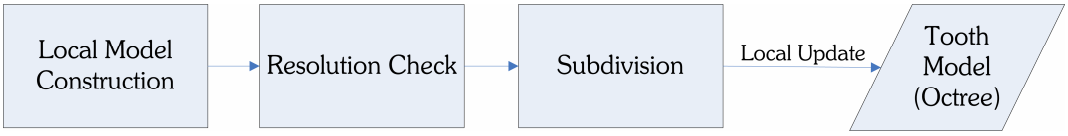

Fig. 7. Flow chart of the simulation algorithm.

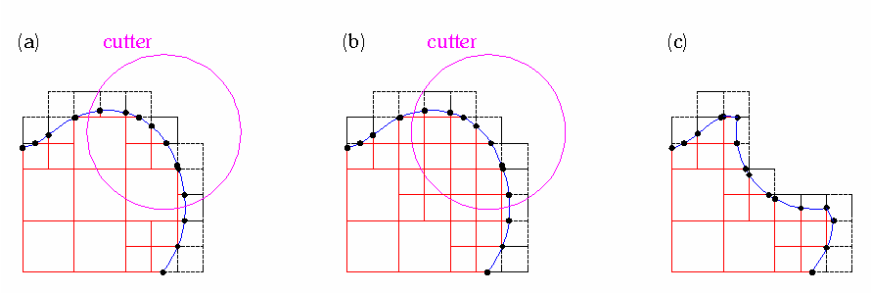

Fig. 8. Simulation algorithm. (a) The cutter contacts the tooth model. (b) Subdivision. (c) Local update.

The local update algorithm is described as follows. Fig. 9. shows all of the contact situations between the tooth model's old *EP* value (*EP1*) and the local model 's new *EP* value (*EP2*) on the same voxel edge. Fig. 9(a)-(h) shows the case of *EP1* being positive and Fig. 9(i)-(p) shows the case of *EP1* being negative. The arc in Fig. 9. represents the current cutter and *EP2* is computed by this cutter and the voxel edge. Thus, Fig. 9(a)-(d),(i)-(l). shows the cutter moves from the end point to start point; on the contrary, Fig. 9(e)-(h), (m)-(p). shows the cutter moves from the start point to end point.

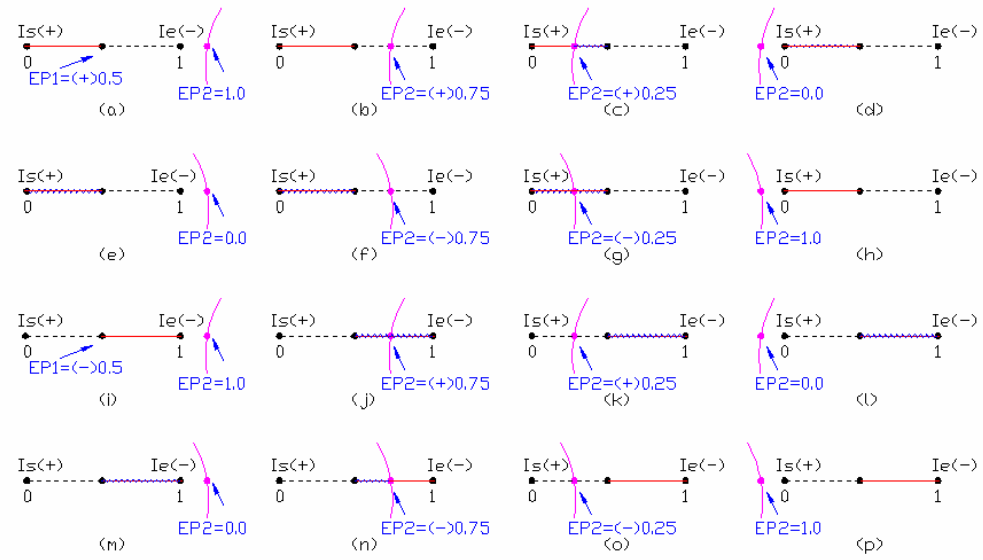

Fig. 9. *EP* value update.

Although there are many contact situations that need to update the tooth model, these situations can be classified into three conditions, described as follows:

- Condition 1: When the *EP2* is zero as shown in Fig. 9(d)(e)(l)(m)., it means that the edge has to be cut by cutter, and this edge of tooth model must be updated. Therefore, its *EP* value has to be set zero.
- Condition 2: When the *EP1*'s sign is different form the *EP2*'s sign as shown in Fig. 9(f)(g)(j)(k)., we set the *EP* value to be zero. In Fig. 9(f)(k), the *EP* value has to be set zero because the red segment of the voxel edge is inside the cutter. However, when the red segment is not inside the cutter completely as shown in Fig. 9(g)(j), we deliberately set the *EP* value to be zero because we assume the segment is small enough to be ignored when the tooth model has high resolution.
- Condition 3: When *EP1* or *EP2* does not meet the above conditions as shown in Fig. 9(a)-(c)(h)(i)(n)-(p)., the partial red segment is inside the cutter. Thus, the *EP* values also have to be updated. The update rule is to compare the *EP1* and the *EP2* values and choose the small one.

The pseudocode for the *EP* -Update algorithm is described as follows:

```
FLOAT EP-Update(EP1, EP2)
{
    IF ( EP2 = 0 OR EP1* EP2 < 0 )    // Condition 1 & 2
        RETURN 0
    ELSE                              // Condition 3
    {
        IF ( EP1 < EP2 )
            RETURN EP1
        ELSE
            RETURN EP2
    }
}
```

Fig. 10. shows a dental simulation process by using different cutters.

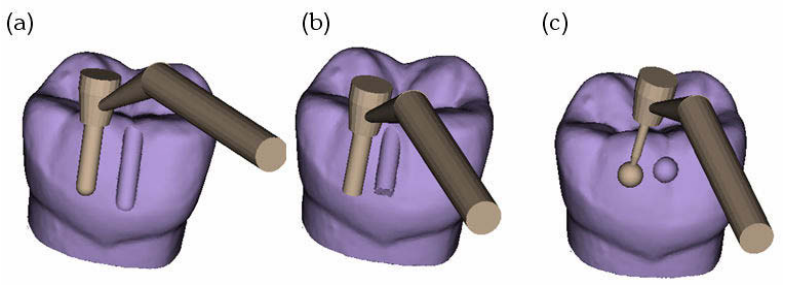

Fig. 10. Dental simulation. (a) Using ball endmill. (b) Using flat endmill. (c) Using dental burr.

3.6 Force Computation Algorithm

A spring system in one dimension is showed in Fig. 11. When the external force *F* is applied to the spring, the spring will compress from x_1 to x_2, so the spring force can be calculated by using the Hooke's law:

$$\vec{F}_{spring} = k \cdot \Delta\vec{d} = k \cdot (x_1 - x_2) \tag{3}$$

where

k : spring stiffness

$\Delta\vec{d}$: displacement

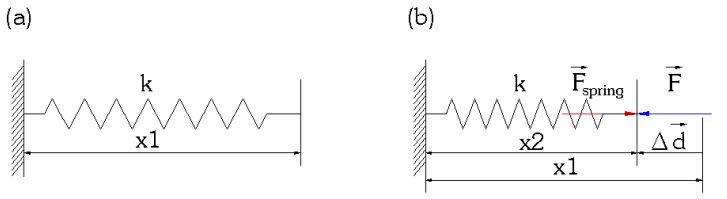

Fig. 11. One-dimension spring system.

In order to compute the spring force, the displacement ($\Delta\vec{d}$) must be computed. Thus, we also use the *EP* value to compute the displacement. In Fig. 9., there are sixteen situations which the cutter contacts the tooth model, but only ten situations require the force computation. These situations can be further classified into four conditions to compute the displacement:

- Condition 1: When *EP2* is zero as shown in Fig. 9(d)(e)(l)(m)., $\Delta\vec{d}$ equals *EP1*.

- Condition 2: When *EP1* and *EP2* have different signs as shown in Fig. 9(f)(g)(j)(k)., $\Delta\vec{d}$ also equals *EP1*.
- Condition 3: When *EP1* is bigger than *EP2* and *EP1* is positive as shown in Fig. 9(c)., $\Delta\vec{d}$ is the value of (*EP1 - EP2*).
- Condition 4: When *EP1* is bigger than *EP2* and *EP1* is negative as shown in Fig. 9(n)., $\Delta\vec{d}$ is the value of (*EP2 - EP1*).

Pseudocode for computing the displacement is as follows:

```
FLOAT Displacement (EP1, EP2)
{
    IF ( EP2 = 0 OR EP1*EP2 < 0 )    // Condition 1 & 2
        RETURN EP1
    ELSE
    IF ( EP1 > EP2 )
    {
        IF ( EP1 > 0 )
            RETURN ( EP1 – EP2 )    // Condition 3
        ELSE
            RETURN ( EP2 – EP1 )    // Condition 4
    }
}
```

In our 3D dental training system, it is assumed that all of the voxel edges inside the tooth model have a spring as shown in Fig. 12(a)., so different tissues, for example enamel, dentin and pulp can have local material stiffness. In the CT or MRI image data, different tissues have different grayscale value, so the local material stiffness can be assigned accordingly.

When the cutter contact the tooth model as shown in Fig. 12(b)., the springs of the green segments look like being compressed. Thus, we can easily compute the spring forces in x, y and z directions by using Eqn. (3) and then the overall resultant spring force is the summation of all forces as follows:

$$\vec{F}_{Total} = \sum k_i \cdot \Delta\vec{d}_i \tag{4}$$

where

k_i : local material stiffness

$\Delta\vec{d}_i$: displacement

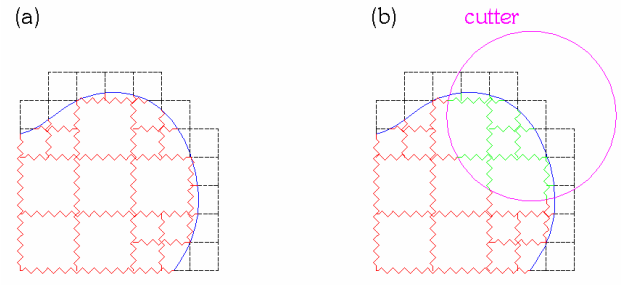

Fig. 12. Force model. (a) Spring system. (b) Force computation by using Hooke's law.

3.7 Force Filtering Algorithm

Several different approaches have been proposed to analyze the stability of the haptic system, but some of these approaches cannot be implemented due to the limit that the haptic force feedback must maintain an update rate at 1K Hz. Therefore, for practical use, we adopt the spring-damper approach and a simple filter to get more stabilized feedback response as shown in Fig. 13. The force equation, after damping is applied, becomes:

$$\vec{F} = \sum k_i \cdot \Delta\vec{d}_i - b \cdot \vec{v} \tag{5}$$

where

k_i : local material stiffness

$\Delta\vec{d}_i$: displacement

b : viscosity

$\vec{v}$: linear velocity

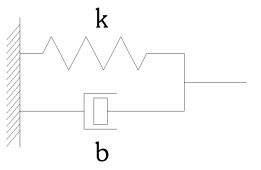

Fig. 13. Spring-damper system.

As regard to the force filter [2], we use the Eqn. (6) to produce smoother force signal before sending it to the haptic device.

$$\begin{cases} \vec{F}_t' = \vec{F}_{t-1} + \delta \cdot \Delta\vec{F} / \left\|\Delta\vec{F}\right\| & \left\|\Delta\vec{F}\right\| > \delta \\ \vec{F}_t' = \vec{F}_t & \left\|\Delta\vec{F}\right\| \le \delta \end{cases} \tag{6}$$

Where δ is a predefined threshold for the force change.

4. IMPLEMENTATION AND RESULTS

The dental training system has been implemented using a hardware consisting of Xeon 2.8-GHz CPU, 1G Byte memory and a NVida Quadro FX 1100 graphic card. Here, we demonstrate two kinds of tooth preparations using our system. One is inlay preparation as shown in Fig. 14. and the other is a bridge case as shown in Fig. 15.

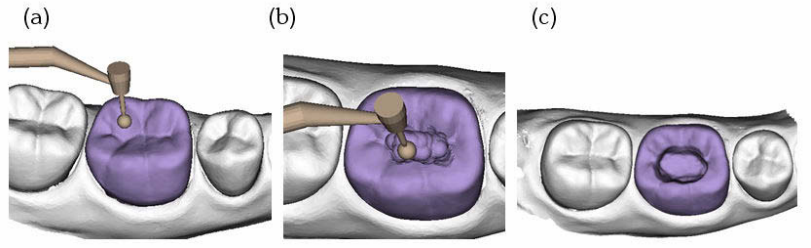

Fig. 14. Inlay preparation. (a) Tooth Model. (b) The process of inlay preparation. (c) Finish inlay preparation.

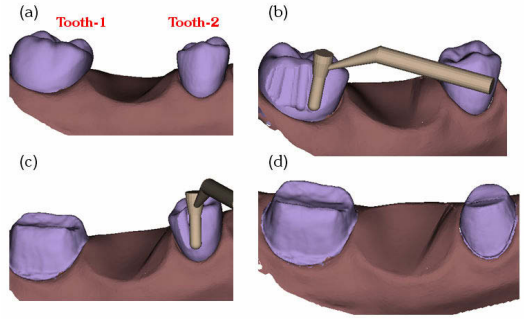

Fig. 15. Bridge case (a) Tooth Model. (b) The process of tooth-1. (c) The process of tooth-2 (d) Finish tooth preparation.

5. CONCLUSION

In this paper, we propose a novel approach to develop a dental training system for simulating tooth preparation. For smooth and realistic force interaction with the haptic device, which is critical to meaningful tooth preparation simulation, we develop an octree-based adaptive voxel model connected by a matrix of spring network. Local material stiffness can be simulated by assigning different spring constants. Therefore, stable and realistic contact forces can be simulated to approximate the touch sensation of different tooth materials such as enamel, dentin, and pulp. For precision sculpting simulation, an implicit function is employed to exactly represent various kinds of dental tools so that accurate calculation of cutter-surface intersection can be realized. The proposed system is highly applicable to virtual dental training and practices. Examples of tooth preparations such as inlay and bridge preparation are demonstrated. Future work includes the comparison of the simulation results to standard or experienced preparation scanned by a 3D laser scanner. Such error comparison can help dental school professors analyze and grade the students' preparation work and provide objective feedback to their learning. Another interesting future work is the simulation of implant surgery, which will be a natural extension of this research work.

6. REFERENCES

[1] Thomas, G., Johnson, L., Dow, S. and Stanford, C., The Design and Testing of a Force Feedback Dental Simulator, *Computer Methods and Programs in Biomedicine*, Vol. 11, No. 64, 2001, pp 53-64.

[2] Wang, D., Zhang, Y., Wang, Y., Lee, Y.-S, Lu, P. and Wang, Y., Cutting on Triangle Mesh Local Model-Based Haptic Display for Dental Preparation Surgery Simulation, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, No. 6, 2005, pp 671-683.

[3] Gottschalk, S., Lin, M. C. and Manocha, D., OBB-Tree: A Hierarchical Structure for Rapid Interference Detection, *Computer Graphics*, Vol. 30, 1996, pp 171--180.

[4] Kim, L., Hwang, T., Park, S. H. and Ha, S., Dental Training System using Multi-model Interface, *Computer-Aided Design & Applications*, Vol. 2, No. 5, 2005, pp 591-598.

[5] Luciano, C., Banerjee, P., Florea, L. and Dawe, G., Design of the ImmersiveTouch™: a High-Performance Haptic Augmented Virtual Reality System, *11th International Conference on Human-Computer Interaction*, 2005.

[6] Petersik, A., Pflesser, B., Tiede, U., Hohne, K.-H and Leuwer, R., Realistic Haptic Interaction in Volume Sclpting for Surgery Simulation, *Lecture Notes in Computer Science*, Vol. 2673, 2003, pp 194-202.

[7] Srinivasan, M. A. and Basdogan, C., Haptics in virtual environments: taxonomy, research status, and challenges, *Computers & Graphic*, Vol. 21, No. 4, 1997, pp 393-404.

[8] Salisbury, K., Conti, F., Barbagli. F., Haptic Rendering: Introductory Concepts, *IEEE Computer Graphics and Applications*, Vol. 24, No. 2, 2004, pp 24-32.

[9] Hayward, V. and Astley, O. R., Performance Measures for Haptic Interface, *Robotics Research: The 7th Int. Symposium*, 1996, pp 195-207.

[10] Adams, R. J. and Hannaford, B., Stable Haptic Interaction with Virtual Environments, *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 3, 1999, pp 465-474.

[11] Miller, T. and Zeleznik, R. C., The design of 3D haptic widgets, *ACM Symposium on Interactive 3D Graphics*, 1999, pp 97–102.

[12] McDonnell, K. T., Qin, H. and Wlodarczyk, R. A., Virtual Clay: A Real-time Sculpting System with Haptic Toolkits, *ACM Symposium on Interactive 3D Graphics*, 2001, pp 179-190.

[13] Galyean, T. A. and Hughes, J. F., Sculpting: An Interactive Volumetric Modeling Technique, *Computer Graphics*, Vol. 25, No. 4, 1991, pp 267-274.

[14] Avila, R. S. and Sobierajski, L. M., A Haptic Interaction Method for Volume Visualization, *IEEE Visualization*, 1996, pp 197-204.

[15] Huang, J., Crawfis, R., Lu, S. C. and Liou, S. Y., A complete distance field representation, *IEEE Visualization*, 2001, pp 247-254.

[16] Yau, H. T., Tsou, L. S. and Tong, Y. C., Adaptive NC Simulation for Multi-axis Solid Machining, *Computer-Aided Design & Applications*, Vol. 2, No. 1-4, 2005, pp 95-104.