# Development of a Dental Training System Based on Point-based Models

Hong-Tzong Yau[1] and Chien-Yu Hsu[2]

[1,2]National Chung-Cheng University, [1]imehty@ccu.edu.tw, [2]apache@pcc-server.ccu.edu.tw

**ABSTRACT**

Recently, a new point-based geometric modeling approach using surfels (surface elements) has been introduced into the field of computer graphics. This new type of 3D model can be used to render complex geometry with less geometric and topological constraints. This paper presents a dental training system which uses surfel models and carries out realistic cutting simulation using a haptic device. We use surfel models to render dental tools and virtual teeth models to get better visual quality with less memory cost. By using surfel models to render dental tools, we can model cutters of many different shapes. The Boolean operation of the surfel model is implemented with local update; the modeling and visual update frequency of the system is about 30Hz. Collision detection and force computation are implemented on an Octree box to simulate physical interactions between a virtual dental tool and a tooth model. Dental training systems based on surfel models can work on any cutter shape. We add the material property to every surfel to simulate different force feedbacks when there is contact with different materials. Adaptive collision detection is performed so that haptic rendering update occurs under 1 kHz. Finally, we use a filter to make the force feedback smooth.

**Keywords:** dental training system, surfel model, haptics device.

## 1. INTRODUCTION

Haptic dental simulation allows dental students to practice cavity preparation and other procedures by providing realistic touch sensations. Most haptic-based dental training systems are developed based on the voxel model and use marching cubes to display a model. They use a lot of memory and do not produce good visual quality. Because of the collision algorithm, the dental tools rendered in most haptic-based dental training systems are very simple. Recently, a new point-based geometric modeling approach using surfels (surface elements) has been introduced into the field of computer graphics. This new type of 3D model can be used to render complex geometry with less geometric and topological constraints. It can be applied to implementing a painting system [10] or a sculpting system [15] using a haptic device. In this paper, we use surfel models to render dental tools and virtual teeth models to get better visual quality and render rate.

A dentist uses different cutter shapes to sculpture teeth according to different requirements. We use surfel models to model any cutter shape and Boolean operation [6] to determine whether points have to be deleted or added. We also use an Octree local update method to speed up the Octree data update rate. The modeling and visual update frequency of the system is about 30Hz. Collision detection of two surfel models is simplified so that it is Octree box collision detection. But if the Octree is subdivided into too many levels, the collision of two objects cannot be detected within 1ms. We use adaptive collision detection to speed up the collision time.

Realistic touch sensations in training are very important. We assign a property coefficient in every surfel. When we sculpt teeth, we can simulate the force in cutting and the force of contact with different materials. The force feedback that is calculated according to adaptive collision detection will result in a discontinuity. We use a 2$^{nd}$-order "bi-quadratic" digital filter to reduce the noise and get a smoother result.

The system architecture is shown in Fig 1. We use a surfel model to model a dental tool and a virtual teeth model to get better visual quality and render rate. In the modeling loop, the Boolean operation of the surfel model is used to update the surfel model of the teeth, and the local data update is used to update the Octree data of the teeth. In the

force loop, adaptive collision detection is used to check the intersection of the Octree box. Force computation is used to generate the force of contact with material properties.

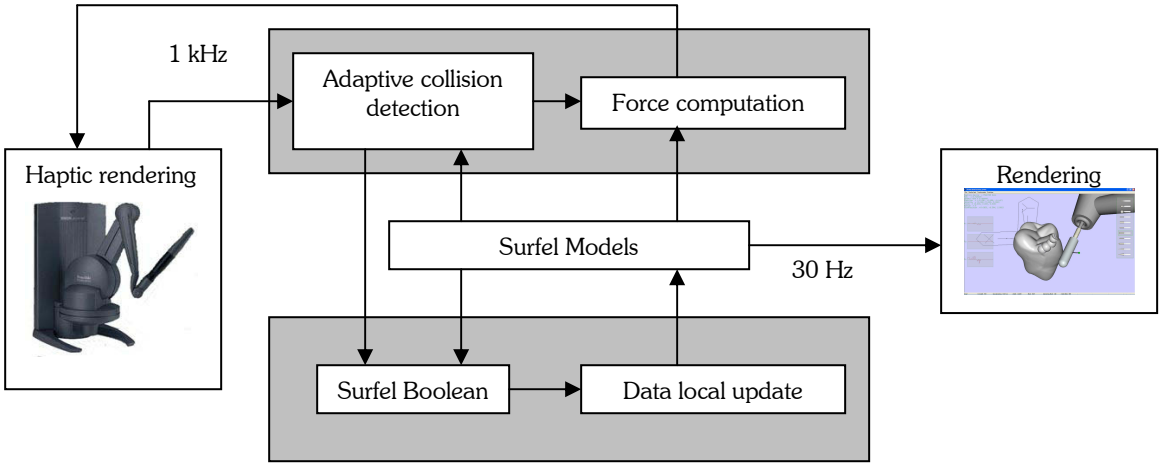



Fig 1. System architecture.

## 2. RELATED WORK

**Surfel rendering.** The word "surfel" is a combination of two words: "surface" and "element" [1]. A surfel model can be used to render complex geometry with less geometric and topological constraints. Surface splats were first proposed for rendering purposes by Zwickeret al.[5]. In order to bridge the gaps between neighboring point samples, Pauly et al. presented a splat clipping method to render the sharp features of surface splats[4]. Guennebaud et al. discussed the capability of today's GPU for hardware-accelerated surface splatting[3]. This per-pixel normalization can efficiently be done by rasterizing a window-sized quad. In Botsch et al.[2], using the pixel shaders of current graphics hardware allows the rasterization of elliptical splats by rendering.

**Surfel Boolean geometry.** Constructive Solid Geometry (CSG) is a very common technique for building complex models using Boolean combinations of simpler models. Such a volumetric surface representation can easily be derived from a point-sampled surface by observing the shift vector induced by the MLS projection operator relative to the oriented normal vector. Since the intersection of two smooth objects may result in sharp feature curves, these objects have to be sampled properly in order to avoid aliasing artifacts. Pauly et al. [8][9] proposed a simple technique, based on a local Newton-type iteration method, that snaps nearby sample points to the sharp features. The sharp features are then represented by clipped splats. Another very efficient method for interactive CSG computations was proposed by Adams and Dutr´e [6]. In a following paper they also implemented their approach on the GPU in order to exploit hardware acceleration for CSG [7].

**Dental training systems.** Research has been done in the field of haptic-based dental training systems. Ranta et al. [11] introduced a dental training system using Phantom to allow students to practice cavity preparation. Thomas et al. [12] developed a dental training system which enables a user to practice the detection of carious lesions. But their systems only focus on haptic contact simulation. Kim et al. [16] presented a dental training system through a multi-model interface using visual, auditory, and haptic sensations. And their system is developed based on a voxel model like FreeForm [17]. The collision detection and force computation of their system are based on an offset surface. Wang et al. [13] presented a dental preparation surgery simulation system based on the use of triangle mesh to realize stable and realistic cutting simulations using an impedance display haptic device. Novint also developed the VRDTS (Virtual reality dental training system) prototype [18].

## 3. SYSTEM OVERVIEW

**User interface.** We developed a user interface which enables a dentist to manipulate dental tools and sculpture virtual teeth. In our dental training system, the virtual dental tool is positioned using a six DOF input device, such as the PHANToM Desktop (Sensable technologies), which provides haptic feedback to the user. The dentist can choose

different dental tools, such as a round cutter or a cylinder cutter, to sculpture the virtual teeth. A Surfel Boolean method [6] is used to simulate the sculpting of teeth.

**Virtual dental tools.** We model virtual dental tools using point-sampled surfaces, as shown in Fig 2. A dentist will use different cutter shapes to sculpture virtual teeth according to different requirements. We use a surfel model that can model any cutter shape. The Boolean operation and collision detection of the surfel models are very efficient. The virtual tools are parametric. We can input size, tolerance, and shape to change the cutter. The virtual dental tool is rendered using a surfel model with good visual quality and fast rendering speed.

**Sculpture model.** We use Boolean operation [6] to determine whether points have to be deleted or added. We also use an Octree local update method to speed up the Octree data update rate. After sculpting the model, we only update the Octree node that the surfel data has changed. The modeling and visual update frequency of the system is about 30Hz.

**Haptic Display.** To guarantee the required 1 kHz update frequency of the haptic device, we decouple the force computation from the rest of the application. Only operations that are necessary to simulate the dynamic behavior of the dental tools are performed. All other operations, such as Octree box intersection and surfel Boolean and data updates, run at the 30 Hz display frequency. We use adaptive Octree collision detection to compute the force, and choose the level according to the moving speed of the dental tool. If the dental tool moves too fast, we use the box in the first level to check for collision. Otherwise, we use the lower level of the Octree box.

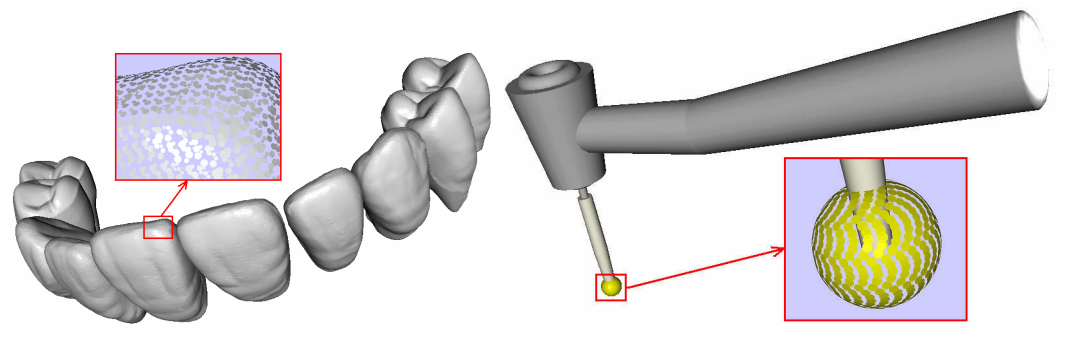


Fig 2. A virtual teeth model and virtual dental tools are rendered using point-sampled surfaces.

## 4. VIRTUAL DENTAL TOOLS

The cutter is rendered using point-sampled surfaces with different shapes and sizes. We use three standard geometric shapes for the surfel model to generate the cutter. We can generate different shapes of the surfel model with the same point density.

### 4.1 Sphere

Assume that the tolerance is $\varepsilon$ and the radius of the cutter is $R$. The center point of the bottom of the cutter is $(X_c, Y_c, Z_c)$. First, we use equation (2) to get the differential angle $dA$ that is related to the tolerance. We use $dA$ to get the differential radius $dR$. The position $(X, Y, Z)$ of the surfel is as follows:

$$\begin{cases} X = n \cdot dR \cdot \cos(m \cdot dA) + X_c \\ Y = n \cdot dR \cdot \sin(m \cdot dA) + Y_c \\ Z = R \cdot \cos(n \cdot dA) + Z_c \end{cases} \tag{1}$$

$$dA = \frac{\varepsilon}{R} \tag{2}$$

$$dR = R \cdot \cos(n \cdot dA) \tag{3}$$

where *m* and *n* are the parameters of the sphere in the *u* and *v* direction, respectively.

### 4.2 Cylinder

Assume that the tolerance is $\varepsilon$, the overshot tolerance is $os$, the radius of the cutter is $R$, and the height is $H$. The center point of the bottom of the cutter is $(X_c, Y_c, Z_c)$. The cylinder can be divided into two parts. First, the position $(X_d, Y_d, Z_d)$ of the surfel on the disc is as follows:

$$\begin{cases} X_d = n \cdot dr \cdot \cos(m \cdot dA) + X_c \\ Y_d = n \cdot dr \cdot \sin(m \cdot dA) + Y_c \\ Z_d = Z_c \pm H/2 \end{cases} \tag{4}$$

$$dr = \frac{R + os - 0.8\varepsilon}{n} \tag{5}$$

$$dA = \frac{dr}{r} \tag{6}$$

where *m* and *n* are the parameters of the disc in the *u* and *v* direction, respectively.

The second part is the vertical surface of the cylinder. The position $(X_v, Y_v, Z_v)$ of the surfel on the vertical surface is as follows:

$$\begin{cases} X_v = R \cdot \cos(m \cdot dA) + X_c \\ Y_v = R \cdot \sin(m \cdot dA) + Y_c \\ Z_v = Z_c \pm n \cdot dH \end{cases} \tag{7}$$

$$dH = \frac{H + 2os - 1.6\varepsilon}{\varepsilon} \tag{8}$$

where *m* and *n* are the parameters of the cylinder in the *u* and *v* direction, respectively.

### 4.3 Cone

Assume that the tolerance is $\varepsilon$, the overshot tolerance is $os$, the radius of the cutter is $R$, and the height is $H$. The center point of the bottom of the cutter is $(X_c, Y_c, Z_c)$. The cone can be divided into two parts. The disc part is similar to the cylinder. We can use equation (4) to create the disc surface of the cone. The position $(X, Y, Z)$ of the surfel on the cone is as follows:

$$\begin{cases} X_v = n \cdot R/H \cdot \cos(m \cdot dA) + X_c \\ Y_v = n \cdot R/H \cdot \sin(m \cdot dA) + Y \\ Z_v = Z_c + n \cdot dH \end{cases} \tag{9}$$

$$dH = \frac{H}{\sqrt{H^2 + R^2}} \frac{(\sqrt{H^2 + R^2} + os - 1.6\varepsilon)}{n} \tag{10}$$

$$dA = \frac{\varepsilon \cdot H}{n \cdot R} \tag{11}$$

where *m* and *n* are the parameters of the cone in the *u* and *v* direction, respectively.

### 4.4 Surfel Models of the Dental Tools

A dentist will use different cutter shapes to sculpture the teeth according to different requirements. We use surfel models to model any cutter shape. As shown in Fig 3, we use spheres, cylinders, and cones to create several cutters that are usually used, such as a round cutter, a round end cylinder cutter, a needle cutter, an X'mas cutter, and a disc cutter. We can also change the size of these cutters.

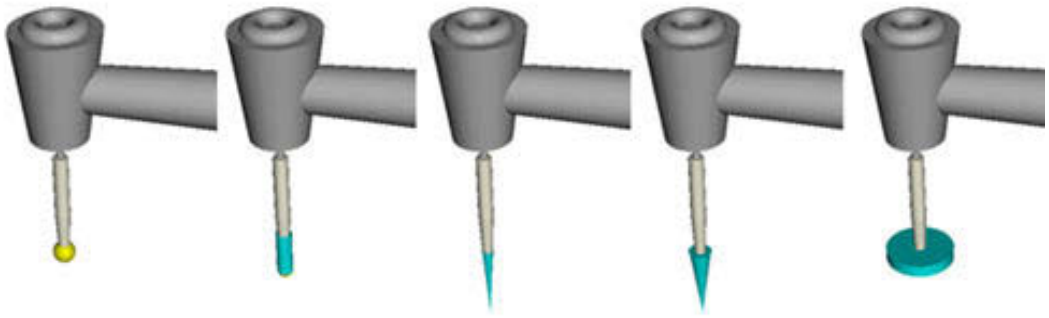


Fig 3. High-speed dental tools of different shapes.

## 5. SURFEL BOOLEAN

We use the Boolean operation [6] to determine whether points have to be deleted or added. As shown in Fig 4, the dental tool and the tooth model are surfel models. When the two models collide, if the points of the tooth model are inside the cutter, the points need to be deleted from the tooth model. If the points of the cutter are inside the tooth, the points need to be added to the tooth model. The Boolean operation of the surfel model used in [6] only deals with the efficiency of a one-time Boolean operation but did not have an efficiency update method. Sculpting of the model continues from the updated model that was last sculpted. The update speed is very important in order for the simulation of the sculpting to be smooth. We use an Octree local update method to speed up the data update rate. After sculpting the model, we only update the Octree node that the surfel data had changed.

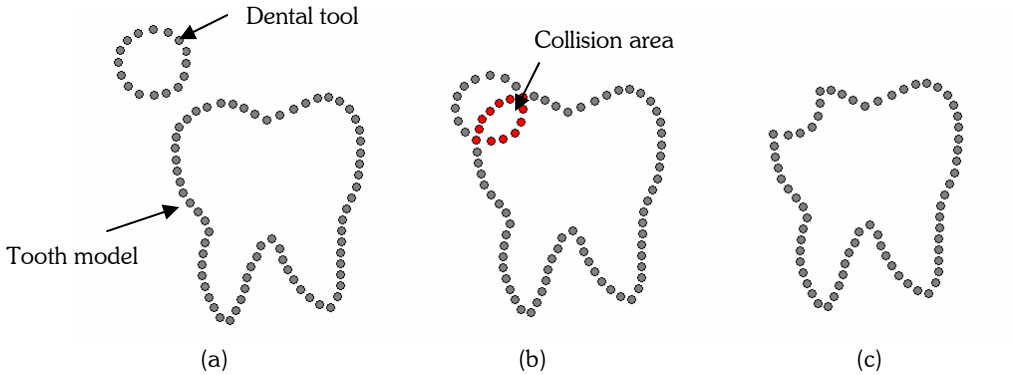



Fig 4. (a)the dental tool and the tooth are represent as a point-sample surface. (b) two models are collided. (c) the result after sculpting.

**Local update.** The surfel data are not changed very much after every sculpting action. The Boolean operation [6] can update the surfel data efficiently. But the Boolean operation is according to the Octree data. The Octree data have to be updated before the next sculpting action. Most of the Octree data do not change after sculpting. It would waste too much time to reconstruct all of the Octree data. We only update the Octree node that the surfel inside the node has changed. First, we process the subdivision part. As shown in Fig 5 (b), the Octree node at the bottom left is not the lowest level. After the Boolean operation, some surfels are added into this node, and the node becomes a boundary node. We subdivide this node first, using the inside-outside test to check whether the subdivided node is inside or outside. On the other hand, some boundary nodes after the Boolean operation become empty nodes. We use the inside-outside test to check the empty node first. Then we check whether the neighboring box is the same to determine if this level of the Octree should be deleted.

## 6. HAPTIC DISPLAY

The algorithm used to generate force depends on the whether a sculpting or a non-sculpting mode is used. In the sculpting mode, the tooth object can be deformed. In the non-sculpting mode, the tooth object cannot be deformed, and a user can only touch the tooth object. The haptic rendering algorithm in the sculpting mode generates a force

proportional to the force used to cut the volume and a force proportional to the speed. We assume a property coefficient $w_i$ in every surfel. When the sculpted surfel number is N, we can get the average property coefficient $w_m$.

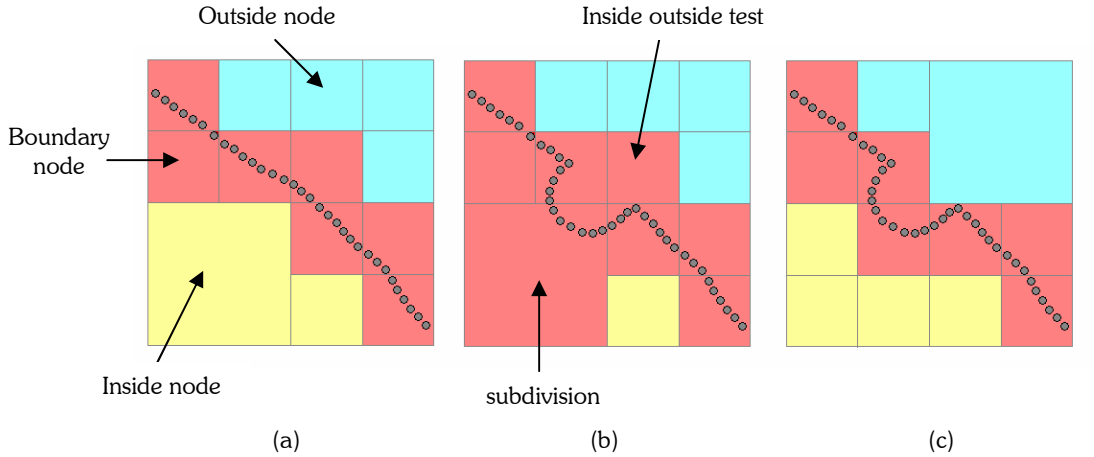



Fig 5. (a) The original Octree data. (b) The Octree data after the model having changed. (c) Update the Octree data only on the node that surfel data has changed.

We can use the following equation to calculate the force in the sculpting mode:

$$F_m = k_m w_m \vec{r} \tag{12}$$

$$w_m = \frac{1}{N}\sum_{i=0}^{N} w_i \tag{13}$$

where $k_m$ is a constant and $\vec{r}$ is the vector that the Phantom uses from the previous position to the current position. The viscous force is as follows:

$$F_v = k_v \vec{v} \tag{14}$$

where $k_v$ is a constant and $\vec{v}$ is the velocity of the Phantom. The force feedback $F_s$ for sculpting is the combination of the material force and viscous force:

$$F_s = F_m + F_v \tag{15}$$

The force generated in the non-sculpting mode is calculated from the Octree box intersection. Octree box intersection can quickly check whether these two objects have collided. But if the Octree is subdivided into too many levels, collision detection of two objects cannot be finished within 1ms. We use adaptive collision detection to speed up the collision time.

**Adaptive collision detection.** When we check the Octree box for collision, we can govern the level that will be calculated, as shown in Fig 6. The level is according to the moving speed of the cutter. If the cutter moves too fast, we only consider level 2 of the Octree. If the cutter moves slowly, we can calculate the collision with the lowest level of the Octree. The level is determined by the following equation (16):

$$L = \begin{cases} L_{\max} & if \left|\vec{v}\right| < v_{\min} \\ k_l \cdot \left|\vec{v}\right| & otherwise \\ 2 & if \left|\vec{v}\right| > v_{\max} \end{cases} \tag{16}$$

Where $L_{\max}$ is the maxima level of the Octree, $k_l$ is a constant, and $v_{\max}$ and $v_{\min}$ are the threshold of the moving velocity. If $L = k_l \cdot \left|\vec{v}\right|$, the value $L$ will inside the interval $2 < L < L_{\max}$.

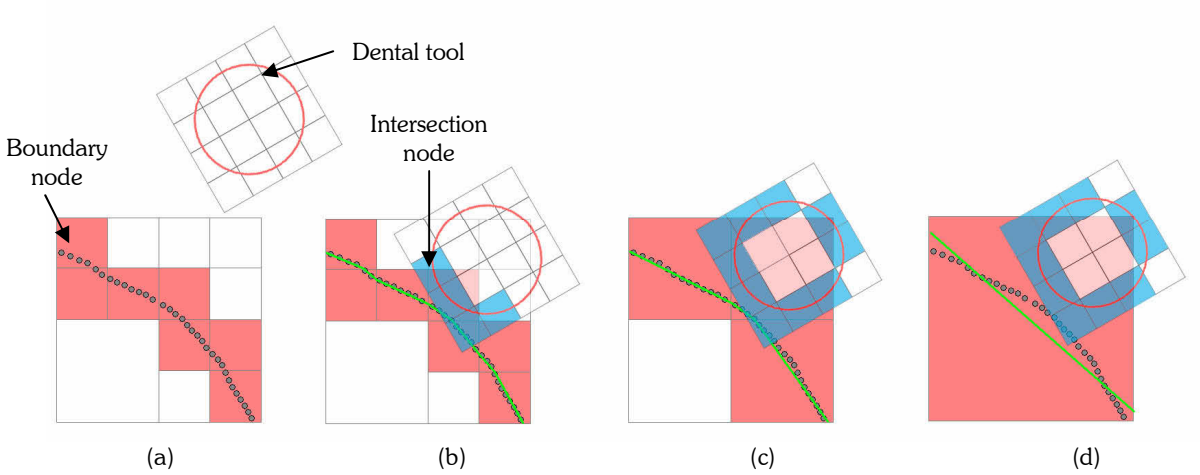



Fig 6. Adaptive collision detection. (a) Two objects do not collide. (b) We calculate the collision detection to level 2 of the Octree. (c) We calculate the collision detection to level 1 of the Octree. (d) We calculate the collision detection to level 0 of the Octree.

We can get the information on whether nodes of the cutter intersect with the nodes of the teeth after using adaptive collision detection. We use the intersecting relationships to compute the collision state. A simplified method is used to calculate the intersected distance [14]. Too much time would be wasted to compute the projected distances of every surfel in the intersected node. Every node can obtain an average normal vector and average plane. We use the average normal vector of the cutter's nodes to compute the distance of the intersection with the average plane of the teeth's nodes. If the intersected point is inside the node of the teeth, this distance is an effective distance $d_i$. If the intersected point is outside the node of the teeth, we do not consider this the intersected distance. When the intersected point is in the opposite direction of the cutter's average normal vector, the surface of the cutter is inside the teeth model. And the intersected distance will less than zero ( $d_i < 0$ ). As shown in Fig 7 (a), if the intersected points are outside the average normal vector of the cutter's nodes, the surface of the cutter is very close to the teeth model. As shown in Fig 7(b), the surface of the cutter is inside the teeth. The intersected points are in the opposite direction of the cutter's average normal vector. We can get information quickly about nodes of the cutter has surfel inside the teeth. Finally, the spring-damper system and the material properties $w_i$ and $b_i$ are used to calculate the force. We only consider the nodes with $d_i < 0$ . The force equation is as follows:

$$\vec{F}_c = \sum_{i=0}^{N} d_i \cdot w_i \cdot \vec{n}_i - b_i \cdot \vec{v}_i \tag{17}$$

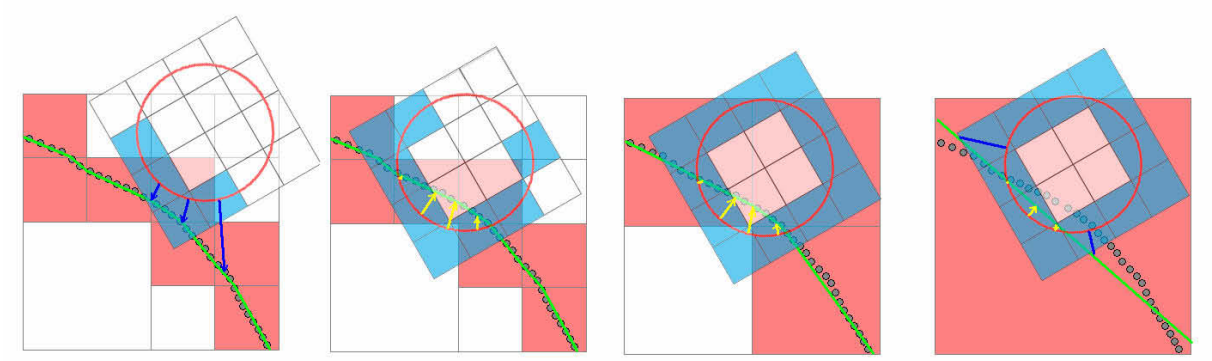
$$\vec{v}_i = \vec{n}_i(\vec{n}_i \cdot \vec{v}_p) \tag{18}$$

where $\vec{v}_p$ is the velocity of the Phantom, $\vec{v}_i$ is the velocity projected to the node's average vector, and N is the number of the cutter node with $d_i < 0$

**Force filter.** The force feedback that is calculated according to adaptive collision detection will result in a discontinuity. We use a 2nd-order "bi-quadratic" digital filter to reduce the noise and obtain a smoother result. The 2nd-order "bi-quadratic" digital filter transfer function is as follows:

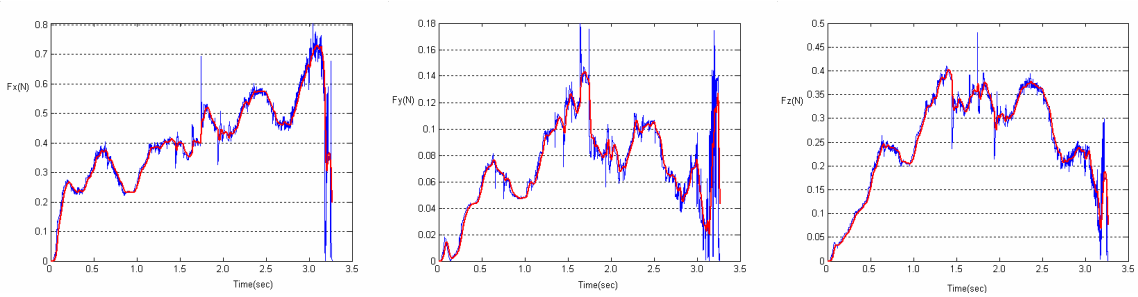
$$H(z) = H_d \cdot \frac{1 - b_1 z^{-1} - b_2 z^{-2}}{1 - c_1 z^{-1} - c_2 z^{-2}} \tag{19}$$

Hence in a filter consisting with a N cascaded bi-quad sections, only 4 coefficient $b_1$, $b_2$, $c_1$ and $c_2$ need to be specified for each of the N sections. The $H_d$ constant can be multiplied together with the $H_d$ constants of the other N-1 sections, result in only one gain "overall gain constant" for the entire filter.



(a) (b) (c) (d)

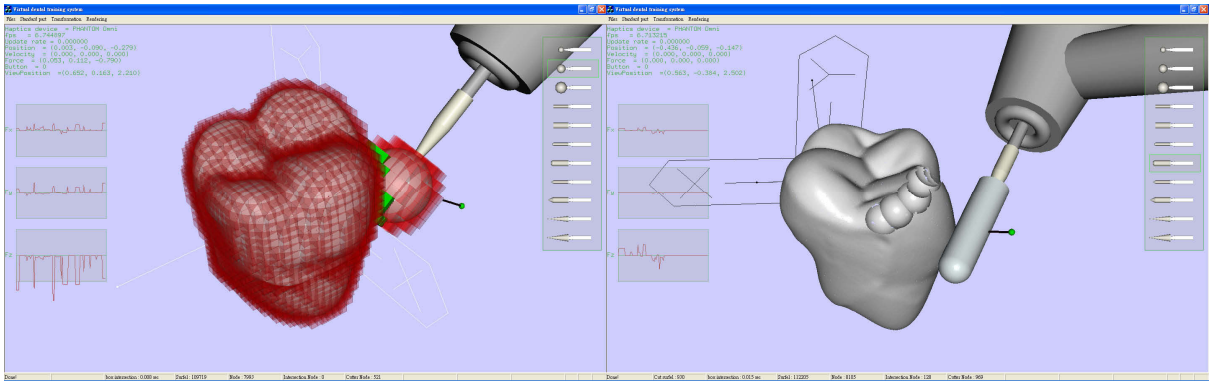
Fig 7. Force feedback. (a) The box intersection without surface collision. (b) The box intersected at level 2 box with surface collision. (c) The box intersected at level 1 box with surface collision. (d) The box intersected at level 0 box with surface collision.




(a) (b) (c)

Fig 8. Virtual force with filter. (a) The force in the x-direction. (b) The force in the y- direction. (c) The force in the z-direction.

We implemented this dental training system with a double Xeon 2.8GHz CPU, a 1G byte RAM, and a Quadro FX 1100 graphics card. The Boolean operation of the surfel model is implemented with local update, which speeds up the simulation speed significantly. The visual update frequency of the system was about 30Hz. The haptic rendering update was under 1 kHz. Using adaptive collision detection resulted in some noise when the velocity of the dental tool changed too much. The force filter reduced the noise and obtained a smoother force.



(a) (b)

Fig 9. (a) The direction of force feedback.(b) The surface after sculpting.

## 7. CONCLUSIONS

In this paper, we presented a dental training system based on the surfel model. This system allows a dental student to learn dental procedures and to use dental tools with tactile feeling. The virtual tooth model and virtual dental tools are

rendered using point-sampled surfaces. Using surfel model costs less memory than using volumetric representation. Haptic rendering is implemented based on Octree adaptive collision detection. This Octree data structure is also used to sculpt the surfel model. We use a local update algorithm to update the Octree data structure in the sculpting loop. In the future, we plan to enhance the dental training system by adding more realistic feeling for the user when the user sculpts different properties and by adding interior properties to a virtual tooth model that is rendered using a surfel model. When tooth enamel is sculpted, the user will feel its hardness, and when the dentine area is sculpted, the user will feel its softness.

## 8. REFERENCES


[1] Pfister, H., Zwicker, M., van Baar, J. and Gross, M., Surfels: Surface elements as rendering primitives, *Proc. of ACM SIGGRAPH 00*, 2000, pp. 335–342.

[2] Botsch, M. and Kobbelt, L., High-quality point-based rendering on modern GPUs, *Proc. of Pacific Graphics 03*, 2003.

[3] Guennebaud, G. and Paulin, M., Efficient screen space approach for hardware accelerated surfel rendering. *Proc. of Vision, Modeling, and Visualization03*, 2003.

[4] Zwicker, M., R¨as¨anen, J., Botsch, M., Dachsbacher, C. and Pauly, M., Perspective accurate splatting, *Proc. of Graphics Interface 04*, 2004.

[5] Zwicker, M., Pfister, H., van Baar, J. and Gross, M., Surface Splatting , *Proc. of ACM SIGRAPH 01*, 2001, pp 371-378.

[6] Adams, B. and Dutré, P., Interactive boolean operations on surfel-bounded solids. *Proc. of ACM SIGGRAPH 03*, 2003, pp 651–656.

[7] Adams, B. and Dutré, P., Boolean Operations on Surfel-Bounded Solids Using Programmable Graphics Hardware, *Proc. of Symp. On Point-Based Graphics 04*, 2004.

[8] Pauly, M., Keiser, R., Kobbelt, L. and Gross, M., Shape modeling with point-sampled geometry. *Proc. of ACG SIGGRAPH 03*, 2003.

[9] Zwicker, M., Pauly, M., Knoll, O. and Gross, M., PointShop 3D: An interactive system for point-based surface editing. *Proc. of ACM SIGGRAPH02*, 2002.

[10] Adams, B., Wicke, M., Dutré, P., Gross, M., Pauly, M. and Teschner, M., Interactive 3D painting on point-sampled objects. *Proc. of Symp. On Point-Based Graphics 04*, 2004, pp 57–66.

[11] Ranta, J. F. and Aviles, W. A., The virtual reality dental training system-simulating dental procedures for the purpose of training dental students using haptics, *Proc. Fourth PHANTOM users group workshop*, Nov.1999.

[12] Thomas, G., Johnson, L., Dow, S. and Stanford, C., The design and testing of a force feedback dental simulator, *Computer methods and programs on biomedicine*, 2000.

[13] 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.

[14] Renz, M., Preusche, C., Potke, M., Kriegel, H. P. and Hirzinger, G., Stable haptic interaction with environments using an adapted voxmap-pointshell algrithm, *Proc. Of the Eurohaptics*, 2001.

[15] Kinashi, S., Sugisaki, Y., Kanao, H., Fujisawa, M. and Miura, K. T., Development of a geometric modeling device with haptic rendering, C*omputer-Aided Design & Application*, Vol. 2, No. 1-4, 2005, pp 283-290.

[16] Kim, L., Hwang, Y., Park. S. H. and Ha, S., Dental training system using multi-model interface, *Computer-Aided Design & Application*, Vol. 2, No. 5, 2005, pp 591-598.

[17] SensAble Technologies Inc. FreeForm modeling system, 1999, http://www.sensable.com/freeform.

[18] Virtual reality dental training system(VRDTS), http://www.nonvint.com/VRDTS.htm