

Haptic Technique for Simulating Multiple Density Materials and Material Removal

Tales Nereu Bogoni^{1,2,3}, Márcio Sarroglia Pinho^{1,2}

¹PontificalCatholicUniversityof Rio Grande do Sul
Porto Alegre – RS – Brazil

²INCT-MACC

Petrópolis – RJ – Brazil

³Mato Grosso StateUniversity
Colider – MT –Brazil

tales@unemat.br, pinho@puccrs.br

ABSTRACT

The term *haptic* refers to the tactile sensation perceived by the user when he is manipulating objects in a Virtual Environment. These sensations are extremely important in Virtual Reality simulators for training medical and dental procedures with the goal of increasing the level of motor skills. Many tasks trained, such as teeth and bones drilling, are held in rigid bodies and require parts of the virtual models to be removed. One way to manipulate these objects is by using voxel-based models, which divide the object into smaller parts that can be removed by the haptic device. This paper deals with a new method for haptic rendering with voxel-based rigid bodies that may be composed of various materials with different densities. The method describes the process for obtaining and storing the virtual objects. In addition it details the techniques used for collision detection, calculating the force feedback and removing parts of objects during the drilling process. The paper ends with an experiment that shows that the method provides stability, supports removal of voxels and different material densities properly.

Keywords

Virtual Reality; Haptic Rendering; Voxel-based Material Removal

1. INTRODUCTION

Virtual Reality (VR) has been increasingly used for training in various areas. One of these fields is health, which uses simulators to train surgeons or dentists, with the aim of increasing their motor skills before they have direct contact with patients. In order to do this, simulators must provide a high degree of realism, both visual and tactile.

For simulators to produce tactile sensation, they must contain haptic devices. These devices reproduce in the real environment the effects of the interactions happening in the virtual environment, causing reactions of the equipment that is being manipulated by the user. These reactions are called *haptic feedback*. The strategy used for calculating the force feedback is called *haptic rendering* and seeks to represent the resistance of the materials that compose the virtual objects during the interaction of a virtual instrument, which is being manipulated by the user with the other virtual objects.

One of the tasks that make use of haptic devices is sculpting rigid virtual objects. During this process,

the operator has to remove part of the volume of the object in order to change its original shape. In the healthcare area, the process of sculpting may be applied in systems that seek to train bones scraping and drilling, and also in dentistry to prepare teeth for filling or for endodontic treatments. Bones and teeth are composed of various types of tissues with different degrees of resistance to perforation. Thus, it is necessary to use virtual models able to represent objects made of multiple materials. In general, these models are made of voxels, even though there are approaches that use polygonal models. Among the various methods that perform haptic rendering using volumetric models, a great part does not deal with changes in the structure of the object such as voxel removal by drilling or sculpting, and are used for exploring the models only.

Considering this background, the aim of this research is to present a method that can be used for sculpting rigid objects composed of voxels with different degrees of hardness, so that they can be used to represent bones and teeth in medical and dentistry training systems. The method presented in this paper is able to keep the stability of the haptic device and represent realistically the force feedback that occurs during the interaction between the tool manipulated by the user and the object being sculpted.

In addition to this introduction, the paper is divided into four sections. Section III presents the commonly adopted strategies for generating force in haptic devices, and also the basic concepts related to haptic

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

rendering and similar research from a bibliographic review in healthcare articles that mention haptic devices and voxelized objects. Then, Section IV describes the method used to perform the experiment, including the modeling and the representation of the object to be sculpted and the tool, how collision detection and haptic rendering are performed, and finally the removal of voxels from the volume. Section V describes an experiment used for evaluating the method proposed in this paper. Finally, Section VI presents the conclusions obtained in this study and suggestions for future research.

2. STRATEGIES FOR GENERATING FORCE FEEDBACK IN HAPTIC INTERFACES

For the development of this research, a bibliographic review was carried out in order to identify the main strategies for obtaining the objects that compose the virtual environment, the data structures used for storing these objects, the methods applied to collision detection and the strategies adopted for calculating the force feedback that is sent to the haptic device. Finally, techniques developed for removing the object parts are described, all of them applied to systems related to the healthcare area.

Ways of obtaining the objects

Within the healthcare area, the objects represent parts of the human body and are obtained basically in 3 ways: tomographic images, magnetic resonance and simplified geometric models. In the first two cases, sets of segmented images are used to identify the different types of tissue and set the values of the tissue density to the voxels, based on the intensity of the voxels identified in the segmented images. In the last case, geometric models are converted into volumetric objects with density values arbitrarily defined for each type of material.

Data storage structures

Both in the area of medicine [5, 10, 13] and in dentistry [1, 11, 15], most systems make use of three-dimensional matrices for storing the object to be sculpted. This is a common approach, because each voxel can store the density of the represented tissue (used for calculating the force feedback in the haptic device) regardless of the neighbor structures. Besides, because they are obtained from tomography and magnetic resonance, the representation through voxels is more natural.

In addition to the objects to be sculpted, the systems have virtual tools that are manipulated by the user to sculpt the object. Such tools may be represented in polygonal [2, 5, 13], volumetric [9, 22], or hybrid [1, 10, 15] ways.

Methods for collision detection

Bounding boxes are used with geometrically modeled tools, but when volumetrically modeled tools are used, collision detection is accomplished through an occupancy map [12]. In this structure, the matrices that represent the object and the tool are superposed, which makes it possible to identify the voxels with collision.

With tools modeled in a hybrid way, the tool is wrapped around bounding boxes, and when it collides with the object's bounding box, the cutting part of the tool (drill bit) is transformed into voxels and superposed to the voxels of the object.

Strategies for calculating force feedback

After the collision detection, it is necessary to simulate the force vectors resultant from the contact between the tool and the object being sculpted and send them to the haptic device to perform the haptic rendering, in order to make the user exert more or less force to move, penetrate or break the object. This simulation uses data intrinsic to the objects in the virtual environment, such as hardness, texture and friction coefficients, in addition to vectors related to the movement performed by the haptic device and the force exerted by the user. Due to its complexity, maintaining a suitable refresh rate in this simulation is the greatest challenge faced by haptic systems. In order to allow an interaction without vibrations or abrupt movements for the user, the devices must be refreshed about 1000 times per second (1 KHz)[4], which makes it necessary to perform collision detection and force feedback calculation in less than 1 ms.

In the real word, when there is an interaction with a rigid object by using a tool, this tool remains in the surface of the object when the contact occurs. On the other hand, when a haptic device is used, the *Haptic Interface Point* (HIP), which is responsible for telling the system which point is being manipulated in the device, can penetrate the virtual object between two processing cycles, depending on the speed of the movement performed by the user. In those cases, the haptic device must apply a force that acts against the movement performed by the user, making the HIP return to the surface of the virtual device. The point that defines the feedback position is called proxy or god-object [23].

Determining the correct position of the proxy is a hard task, and is a decisive issue to make the system able to keep the stability of the haptic rendering. During the execution of each haptic rendering loop, it is necessary to identify the position of the HIP, find the new position of the proxy and calculate the force that will be sent to the haptic device.

Figure 1 shows the haptic rendering process. It is possible to observe that the HIP and the proxy have

the same coordinates while there is no collision with the object, which can be noticed in $t-3$, $t-2$, $t-1$. In time t the HIP penetrates the object and the proxy is kept on its surface. Then, in time $t+1$, a new position is calculated for the proxy when the HIP is moved to a new position inside the object, with the aim of keeping it on the surface of the object, but following the HIP movement.

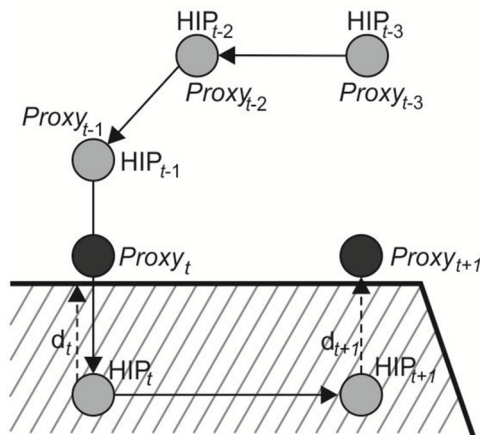


Figure 1–Haptic Rendering (Adapted from [6])

As to polygonal models, the normal vectors of the planes that compose the object are used to indicate the position of the proxy [17, 23].

With regard to volumetric objects, however, these normal vectors do not exist explicitly. For these cases, the literature presents two approaches. In the first one, these normal vectors are calculated from the coordinates of the voxels in the surface of the object [18], and then the same techniques described for Figure 1 are used. Another solution is to apply methods that determine the new proxy position directly from the voxels, using the previous proxy coordinates and the current HIP, as presented by Vlasov et al. [19], who use a Ray Casting algorithm between the previous position of the proxy and the current position of the HIP to calculate the new position of the proxy, which will be close to the surface of the object. In this technique, the first step is to cast a ray from the current position of the proxy to the HIP, creating a list of voxels that are crossed by the ray. The closest voxel to the current proxy is defined as the voxel in which the collision between the object and the tool occurred. Then, the path of the ray is travelled with small steps, smaller than the size of a voxel, until it reaches the voxel of the collision. After that, the last position visited, before reaching it, is defined as the new proxy.

After defining the proxy, it is necessary to calculate the force vector and send it to the haptic device. This vector is normally calculated with Hooke's Law $\mathbf{F} = k\mathbf{x}$, where k is a stiffness constant of the material and \mathbf{x} is the vector between the position of proxy and the HIP [6]. In order to illustrate this calculation,

Figure 1 presents the vector (d_t) as the vector between the HIP and the proxy, which must be multiplied by the stiffness constant of the material being simulated. In $t+1$ a new force vector is calculated by using the new positions of the HIP and the proxy. This process is repeated after each loop of the haptic rendering.

Content removal techniques

The haptic systems may be used to remove parts of the material that form the objects based on the action of the tool. Depending on the density properties of the material and the force with which the operator uses the haptic device, the removal can be done faster or slower [12]. In the most common approach, the voxel density is gradually reduced to zero, when the voxel is eliminated. In general, rotatory tools with fixed [22] or variable [19] speed are used.

3. PROPOSED METHOD

Our method for haptic rendering proposed has been developed with the aim of using volumetric objects with multiple materials to represent rigid structures of the human body, such as teeth and bones, in procedures that involve drilling and scraping with tools that have a drill bit.

The next sections describe the techniques applied in this method to represent the objects to be sculpted and the tools manipulated by the user, the collision detection process, the algorithm to choose the proxy position, and the strategies to generate force feedback and to perform the volume removal.

Representing the objects to be sculpted

The objects are stored in a volumetric way using a three-dimensional matrix, in which each entry represents a voxel. The voxels contain an attribute that indicates its density, determining the type of material it is representing. The density aims at informing the graphic renderer if the voxel is visible or not, and the haptic renderer about the amount of material remaining to be removed. When the voxel has a positive density, it is used in the graphic rendering and haptic rendering algorithms; when it is zero, the voxel is ignored.

Another characteristic in voxels concerns their position in relation to their neighbors. The possible classes are: **external** voxel, when it presents null density; **internal** voxel, when it has density and has no external neighbors; and **boundary** voxel, when it has a positive density and at least one external neighbor. The classification of external and internal voxels is simple, by simply checking its density. For the classification of boundary voxels, it is necessary to analyze their 26 neighboring voxels, which is called 26-neighbor (Figure 2a) of each voxel. If the voxels in this neighboring are not all of the same type (internal or external), the voxel is classified as a boundary one.

With this information, the occupancy map of the object is obtained, which is illustrated two-dimensionally in Figure 2b, indicating the external (0), internal (2) and boundary (1) voxels.

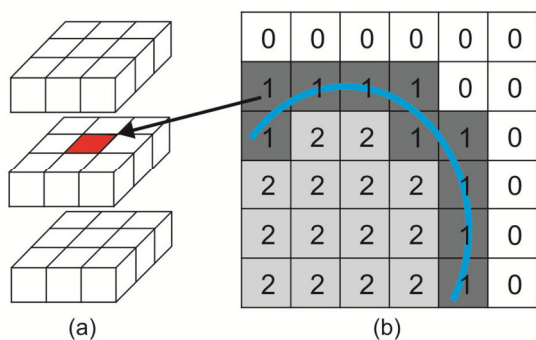


Figure 2- 26 Neighboring (a) and occupancy map (b)
(Adapted from [12])

Other relevant information about voxels refers to its coordinates. Two coordinate systems are used, a local and a global one. In the local system, the coordinates represent the position of the voxel inside the 3D matrix, which is used to facilitate the localization of the voxel's neighbors. In the global one, the voxel coordinates are stored inside the coordinate system of the haptic device, which are used in the haptic rendering algorithm for collision detection and force feedback calculation. In order to complete the object modeling, a bounding box that delimitates the volume occupied by the object's voxels is stored.

Representing the tools

The tools are modeled in a hybrid way, with a geometric model for the body of the tool and a volumetric model for the part used for drilling the volume (drill bit). Figure 3 shows some possible drill bit models, but other models are possible.

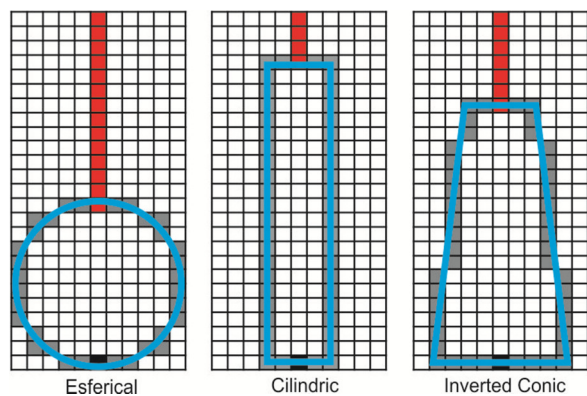


Figure 3-Drill bit models

For the volumetric definition of the drill bits, two types of voxels are used, one in the cutting part of the drill bit, which is responsible for removing the material of the object to be sculpted (represented in gray in Figure 3), and another one (in red) that

represents the drill bit handle and is used for collision detection only.

For the haptic rendering algorithms to work correctly, the size of the voxels used to represent the drill bit of the tool must be the same used for the object to be sculpted. The displacement of the tool in the virtual environment happens through the manipulation of the haptic device that controls the HIP, which works as a pivot of the tool. This pivot is represented in Figure 3 and Figure 4 by a black voxel. Based on the pivot, the tool's voxels will be superposed in the occupancy map of the object in order to check the existence of collision and perform the removal of the object's voxels. In the example illustrated in Figure 4a, the drill bit is inside the object's occupancy map, but without collision, while in Figure 4b it is colliding with the boundary. More details on the process of collision detection will be presented next.

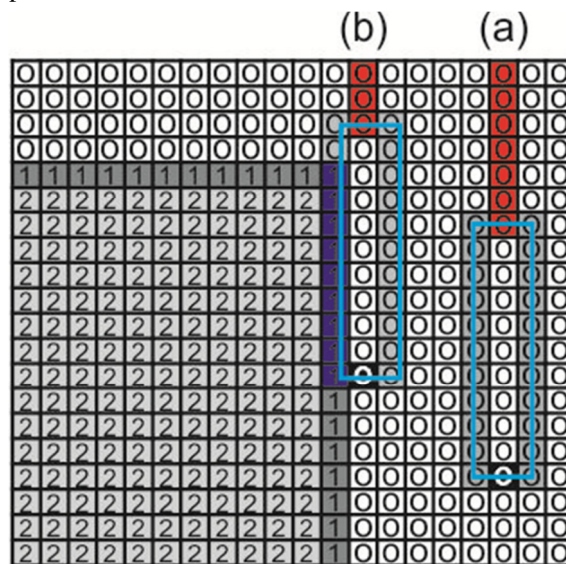


Figure 4-Occupancy map with superposed drill bit

Collision detection

Collision detection is carried out by checking if the drill bit of the tool is touching the border of the object to be sculpted. The algorithm used for this is an adaptation of the algorithm proposed by McNeely et al. [12], which uses an occupancy map to check the existence of collision.

In the approach proposed in this paper, collision detection is performed in two stages. In the first one, we check if the tool's pivot point is inside the bounding box of the object. If it is, the tool's matrix of voxels is superposed to the object's voxels, so that the collision verification is carried out with each voxel of the tool. The algorithm considers that there is a collision when a tool's voxel occupies the same space of an object's voxel, with non-null density, that is, with internal or boundary voxels.

Figure 5 shows the process of collision detection during the tool's movement. In picture (a) the tool's pivot is outside the object's bounding box and the other voxels of the tool are not considered. In pictures (b) and (c) the pivot penetrates the object's bounding box and the tool's voxels are superposed to the object's voxels without any collision. In picture (d) there is a collision between the tool's voxels and the object's boundary.

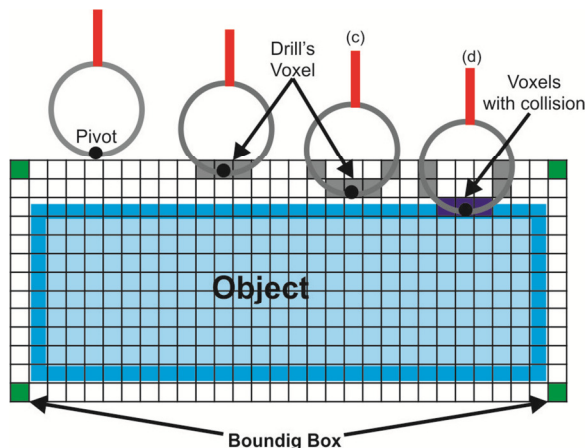


Figure 5–Collision Detection

Choosing the position of the proxy

The method presented in this paper is based on a technique presented by Vlasov et al. [19], which uses a Ray Casting algorithm to determine the proxy's position. The calculation of the proxy's position begins when a collision between the tool and the bounding box that covers the object being sculpted is detected. In this moment, the occupancy map is used to create a list with all the voxels of the object classified as external and that belong to the 26-neighbor of the voxel that contains the proxy. Among these voxels, the closest voxel to the current position of the HIP is found, which will be the voxel containing the new proxy. Figure 6 illustrates the process of choosing the voxel that will contain the proxy. In $t-1$ the current voxel containing the proxy (in gray) and the external neighbors of this voxel (in yellow) are presented. In $time t$ the HIP is moved and we check which of the neighbor voxels is closer to the HIP, defining it as the voxel that will contain the proxy.

At first, the position chosen for the proxy is the center of the closest voxel to the HIP, which can be observed in Figure 6. However, depending on the size of this voxel, the displacement to move the HIP becomes too large and may cause instability in the system. In order to avoid this, a ray from the center of this voxel is casted up to the current position of the HIP. Then this ray is sampled in small points (not exceeding 0,1mm [21]) and these points (Figure 7) are visited from the center of the chosen voxel to the position of the HIP, until a bordering voxel is

reached. When it happens, the last visited point is chosen as the new proxy's adjusted position.

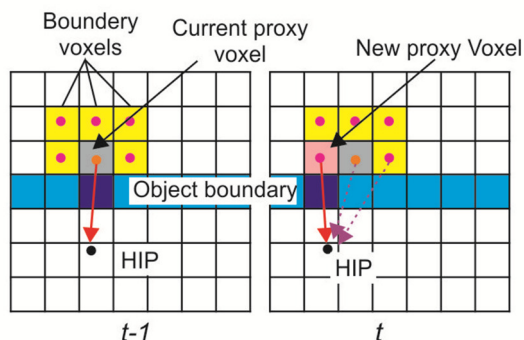


Figure 6 – Determining the voxel that will contain the proxy

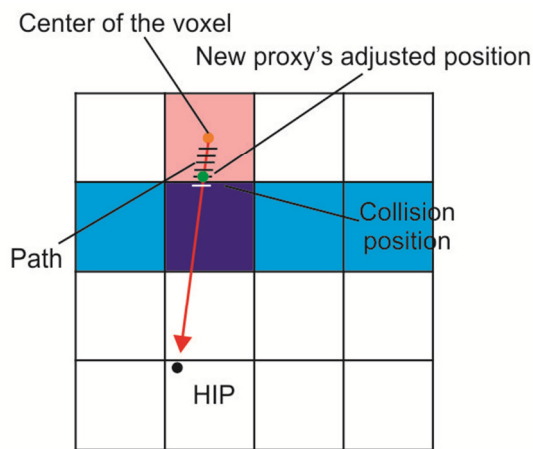


Figure 7–Adjustment of the proxy's position

Force Feedback

As it has already been described above, the virtual tool always follows the proxy and should not penetrate the bordering or internal voxels of the object being sculpted. Therefore, it will navigate only where the occupancy map indicates the existence of external voxels. The HIP, on the other hand, can penetrate the object, and its distance in relation to the proxy is used to indicate the direction and magnitude of the force the user is exerting on the haptic device. Depending on the displacement of the HIP, the tool can slip through the object's boundary, as long as there are no obstacles preventing this movement. Figure 8 illustrates the movement of the HIP and the proxy, where (a) shows the HIP and the proxy together, and (b) shows the proxy in the boundary and the HIP inside the object. When the HIP penetrates the object, it is possible to calculate the force vector that will be sent to the haptic device, which will go in the same direction but in an opposite way to the vector that connects the proxy and the HIP. The magnitude of the vector is given by the distance between the proxy and the HIP.

To determine the resulting force (\vec{F}) that will be sent to the haptic device, two vectors are used, as shown in Equation (1). The \vec{F}_R vector represents the resistance of the material that composes the boundary of the object in collision, and the \vec{F}_F , with lower intensity, indicates the frictional force occurring between the voxels of the tool and the boundary of the object.

$$\vec{F} = \vec{F}_R + \vec{F}_F \quad (1)$$

The resistance vector (\vec{F}_R) is calculated with Equation (2), where, P_{Proxy} is the position of the proxy in the coordinates of the haptic device, P_{HIP} is the current position of the haptic device's pointer, and K is the stiffness constant of the material of the voxel where the proxy is located.

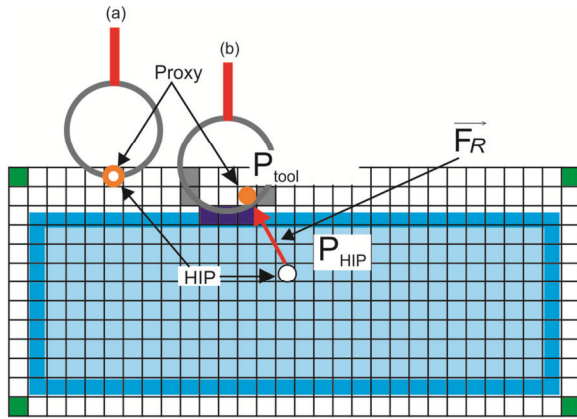


Figure 8 - Components of the Force Feedback Calculation

$$\vec{F}_R = (P_{Proxy} - P_{HIP}) * K \quad (2)$$

Besides the distance between the proxy and the HIP, it is necessary to take into consideration the number of voxels in collision between the tool and the object. This generates a frictional force, \vec{F}_F , which is calculated from the average of the distances between the voxels in collision and the center of the tool. Equation (3) presents the formula of \vec{F}_F , where n is the amount of the tool's voxels in collision with the object, P_{Center} is the position of the center of the tool in the device's coordinates, $P_{Collision(i)}$ is the position of each collision point in the device's coordinates, and K_T is the frictional constant of the tool's material.

$$\vec{F}_F = \frac{\sum_{i=0}^n P_{Center} - P_{Collision(i)}}{n} * K_T \quad (3)$$

Figure 9 shows two examples of frictional force vector calculation. On the left, the contact occurs with three voxels from the bottom, which makes the force vector point up wards, while on the right the collision occurs both at the bottom and on the side of

the tool, which makes the force vector point on a diagonal line.

Volume Removal

The collisions between the tool's cutting parts and the object's surface cause the removal of voxels and the appearance of holes in the object. Consequently, the shape of the object changes with the elimination of some boundary voxels, which turns internal voxels into boundary ones.

The method applied in this paper was inspired by the material removal model proposed by Wang et al. [20], which uses rotary tools for dental drilling. In their method, the amount of material to be removed is calculated based on the tangential velocity of the tool and its displacement. Besides, the authors point out that the amount of material removed from the object will not be greater even if excessive force is exerted during the drilling process, due to the physical limit of the haptic device being used.

Vector of distance between the center of the drill bit and the voxels with collision

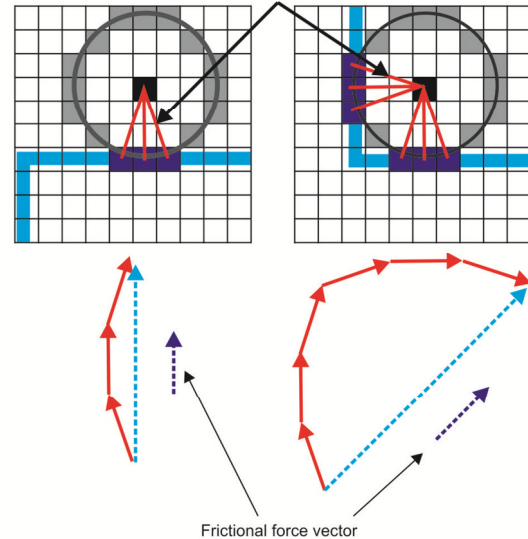


Figure 9 – Voxels with collision for Force Feedback Calculation through the friction between the drill bit and the object's boundary

In the method presented in this paper, a voxel is removed when its density reaches zero. The reduction in voxel density occurs gradually, taking the force exerted by the user on the haptic device into consideration and respecting the device's limit of force feedback, without using the tangential velocity of the tool.

In order to quantify the amount of material to be removed, a wear coefficient (α) is used, which is calculated with Equation 4. For this calculation it is necessary to determine the maximum force feedback value supported by the haptic device (d_{max}), and the distance between the proxy and the HIP (Δd), indicating when the maximum wear of the material will be achieved. Thus, if the user applies a little

force, a small amount of material will be removed, and if the user applies more force than the limit supported by the haptic device, only the amount of material allowed by the system will be removed.

$$\alpha = \begin{cases} 1 & \text{se } \Delta d \geq d_{\max} \\ \frac{\Delta d}{d_{\max}} & \text{se } \Delta d < d_{\max} \end{cases} \quad (4)$$

Figure 10 illustrates how the wear coefficient calculation is done. In the left picture, the HIP is inside the force limit accepted by the haptic device (d_{\max}), thus α is calculated. In the picture on the right, the HIP exceeds the limit, so α assumes the maximum value allowed and the exceeding force applied by the user is ignored.

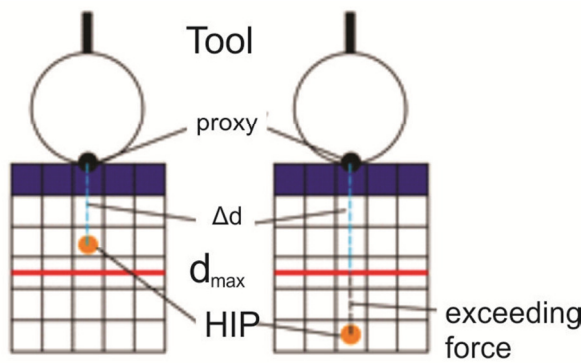


Figure 10 – Demonstration of the calculation of the multiplication factor for voxel removal.

Besides the wear coefficient, a constant representing the material that constitutes the drill bit is applied, which may scrape the object with more or less intensity. The final calculation that will define the new density of the object's voxels is performed on haptics data every 1ms by using Equation (5), where D is the current density of the voxel, K_{Tool} is the constant that represents the material of the tool's drill bit, and α is the wear coefficient. As the graphics rendering is slower than the device update, the new data are sent to the graphic renderer at every 25ms.

$$D' = D - k_{Tool} * \alpha \quad (5)$$

When the density of a voxel gets null, the voxel is eliminated from the haptic and graphic rendering and the object's boundary must be redefined. In order to do this, it is necessary to reclassify the neighbors of the removed voxel. In this process, the internal voxels of the 26-neighbor of the removed voxel are selected and each of them is reclassified. Figure 11, on the left, shows the collision between the voxels of the tool and of the object, and on the right, it demonstrates the boundary after the removal of the voxels that were in collision.

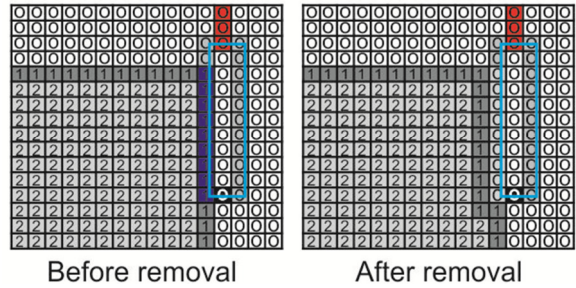


Figure 11–Update boundary

4. Experiment

The method proposed in this paper has been implemented on the C++ programming language using Visual Studio 2010, the OpenGL graphic API and the Chai3d [7] library to control the haptic device. In order to move the tool, the system uses a Novint Falcon Controller [8] device, with 3DOF for tracking and force feedback and 8,9N of force feedback in each axis. We used an Intel Core i5 3.1Ghz processor desktop computer, with 4GB of memory and an AMD Radeon HD 7570 video controller. The system was developed with the use of two *threads*, one of them responsible for the haptic rendering with 1 KHz minimum update, and the other one responsible for the graphic rendering only, which works on about 30 Hz. The tasks of tracking and updating are accomplished inside the haptic renderer, in addition to tasks of collision detection, force feedback calculation and voxel removal.

As to the experiments, we used an object composed by three materials created from polygonal models and converted to voxels with a 128x128x128 voxel resolution, with the software BinVox [3]. Figure 12 shows the stacked block-shaped materials forming a cube. Each block is 10 mm long and wide and 3.33 mm high, so that the cube formed by the stacking of these materials measures 10 mm in each edge. Therefore, each voxel measures 0,078 mm (10 mm / 128 voxels) in each dimension. The density of the voxels is calculated based on the time required to move the tool during the process of drilling, as it will be presented next.

The tool used in the experiment has a polygonal model, used by the graphing renderer only, and a volumetric model, which represents a 0.5mm spherical drill bit.

We developed two kinds of tests. The first one aimed to check the stability of the method by measuring the refresh rate of the device. It also checked whether or not the force feedback is consistent with the movement of the tool. In the second test, the purpose was to verify if the method is able to simulate materials with different degrees of resistance to perforation.

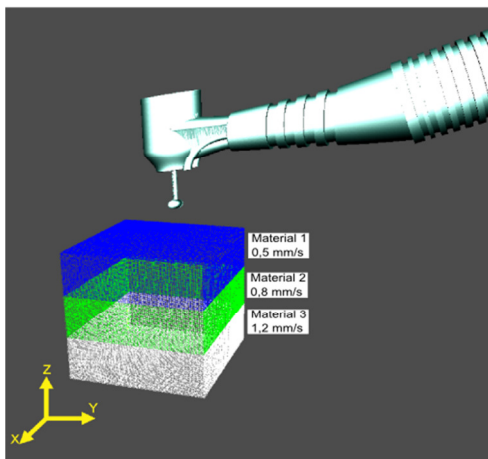


Figure 12 – Screenshot of the models used in the experiment

To evaluate the refresh rate of the device, data related to the use of the system were collected every 25ms, while the user performed the **displacement**, **exploration** and **perforation** procedures. Displacement occurs when there is no contact between the tool and the object being sculpted. Exploration happens when the tool slips through the object without altering the density of the voxels. Perforation occurs when the tool reduces the density of the voxels and eliminates it. In order to activate the perforation mode it is necessary to keep pushing a button on the haptic device.

The graph in Figure 13 shows the data collected for the refresh rate of the haptic device. It can be noticed that in the displacement task (A), the refresh rate is high, as only the collision of the object’s bounding box with the HIP is detected, and the device is free to move. In the exploration task, the refresh rate reduces to about 62 KHz, as the force feedback calculations and the collision detection with the voxels of the tool are performed. In the perforation task, the refresh rate is about 50 KHz, due to the number of voxels of the tool involved in the force feedback calculation as well as the data update inside the object’ matrix for reclassifying the voxels to form the new boundary.

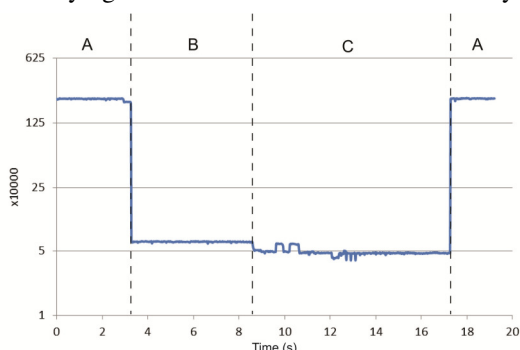


Figure 13 – Refresh rate of the haptic device, in 10 KHz

As to the displacement of the tool and the force feedback, it is possible to see in the graph in Figure 14 that during the navigation task (A) the magnitude of the displacement is great in all the axes and the force feedback is null. When the exploration task is performed (B), the axis with higher force feedback indicates the existence of collision and, as a result, less magnitude of the displacement. In this example, the movement is in the upper part of the object, with higher force feedback and less movement in the Z-axis, which indicates that the user is trying to penetrate the object downwards and sliding on the surface. Finally, in the perforation task (C), the greater the magnitude of the movement, the higher is the force feedback in the axis. In other words, the more the user moves towards a direction, the more the haptic device works against the movement, with the goal of keeping the tool in external voxels.

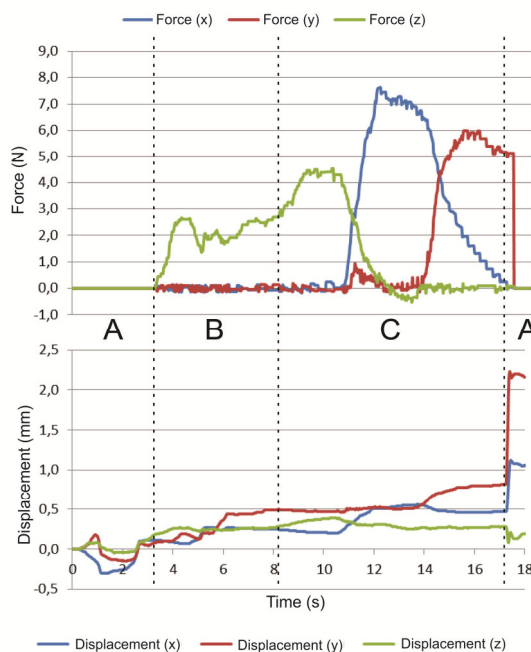


Figure 14 – Displacement of the tool and force feedback

The second test was developed with the goal of checking the materials’ resistance to perforation. For that purpose, it is necessary to simulate objects with different densities. To determine density, the tool’s displacement velocity is measured in a perforation task during the displacement of the tool in the Z-axis, passing it through the three materials, as in the research by Wang et al. [20] and Arbabtafti et al [2].

For this example, we simulated materials whose perforation velocities are 1.2mm/s, 0.8mm/s and 0.5mm/s. These values respectively represent the perforation times of dentin [20], bone [11] and enamel [20], common materials in dentistry procedures. Equation (6) was used to calculate density, where D is the density of the voxel, APS is

the amount of volume removal updates performed in 1 second, $VPMM$ is the amount of voxels that fit into 1.0 mm, and RV is the removal velocity in mm/s. As the size of the voxel is 0.078mm in this example and the update occurs every 25ms, $VPMM = 12.8$ voxels/mm and $APS=40$ updates/s. By applying Equation (6), the density values associated with each material were 2.6, 3.9 and 6.24.

$$D = \left(\frac{APS}{VPMM} \right) \div RV \quad (6)$$

The graph in Figure 15 presents the results obtained in this experiment. It is possible to notice that by using the density values calculated and applying the maximum allowed force of density reduction, the displacement of the tool during the perforation process is coherent with the drilling velocity calculated.

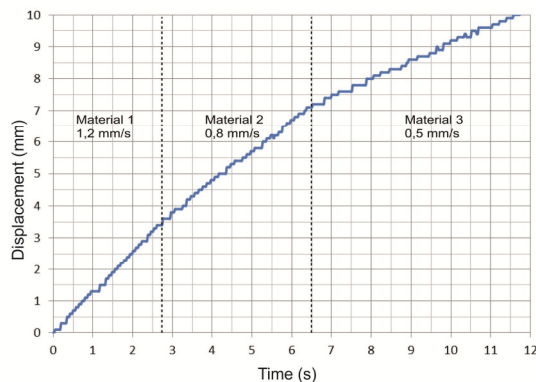


Figure 15 – Displacement of the haptic device during perforation

5. Conclusions and Future Research

The use of the method proposed in the paper makes it possible to model both simple rigid objects with homogeneous density, such as metal or wood blocks, and complex objects formed by materials with different degrees of hardness, such as teeth and bones. The method is independent on the way the data are acquired, as long as they can be stored in a three-dimensional matrix. Thus, it is possible to use data obtained from computer tomography, magnetic resonance or polygonal models converted into voxels.

Considering that the system allows the user to perform displacement, exploration and perforation, the results of the tests demonstrate that the force feedback for the haptic device was kept stable during all the tasks, and it was always updated above 1 KHz, which caused neither undesired displacement in the tool nor vibration in the user's hand. This allowed the user to feel the collisions between the tool and the object and to remove voxels properly.

In addition, the process of drilling and material removal has been effective in correctly removing the voxels and reconstructing the object's boundary in every removal, adhering to the estimated time for the voxel removal based on the density of each material.

Future research can make use of devices with 6 DOF, which allow the rotation of the tool, besides including the tangential velocity control of the tool. Finally, the method presented in this paper can be applied to implement a simulator for training professionals in the area of dentistry with a setup composed by specific hardware that allows the rotation of the tool and controls the tangential velocity of the drill bits by customizing the Novint Falcon Controller haptic device.

6. Acknowledgements

Our research is funded by the National Institute of Science and Technology in Medicine Assisted by Scientific Computing (Grant CNPq 181813/2010-6 and FAPERJ E-26/170.030/2008).

7. References

- [1] Acosta, E.; Liu, A. "Real-time Volumetric Haptic and Visual Burrhole Simulation." IEEE Virtual Reality Conference, 2007. VR '07. p.247-250, 2007. IEEE.
- [2] Arbabafti, M.; Moghaddam, M.; Nahvi, A.; Mahvash, M.; Rahimi, A. "Haptic and visual rendering of virtual bone surgery: A physically realistic voxel-based approach." IEEE International Workshop on Haptic Audio visual Environments and Games, 2008. HAVE 2008. p.30-35, 2008. IEEE.
- [3] binvox 3D mesh voxelizer. www.cs.princeton.edu/~min/binvox
- [4] Cavusoglu, M. C.; David, F.; Frank, T. "A critical study of the mechanical and electrical properties of the phantom haptic interface and improvements for highperformance control." Presence: Teleoperators & Virtual Environments 11.6 (2002): 555-568.
- [5] Eriksson, M.; Flemmer, H.; Wikander, J. "A haptic and virtual reality skull bone surgery simulator." Proceedings of World Haptics, 2005.
- [6] Ho, C-H, Basdogan, C.; Srinivasan, M.A. "Efficient point-based rendering techniques for haptic display of virtual objects." Presence 8.5 (1999): 477-491.
- [7] <http://www.chai3d.org>
- [8] <http://www.novint.com>
- [9] Kimin Kim; Ye-Seul Park; Jinah Park. "Volume-based haptic model for bone-drilling." International Conference on Control, Automation and Systems, 2008. ICCAS 2008. p.255-259, 2008. IEEE.
- [10] Ki-Uk Kyung; Dong-Soo Kwon; Sung-Min Kwon; Heung Sik Kang; Jong Beom Ra. "Force feedback for a spine biopsy simulator with volume graphic model." 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings. v. 3, p.1732-1737 vol.3, 2001. IEEE.
- [11] Kusumoto, N.; Sohmura, T.; Yamada, S.; Wakabayashi, K.; Nakamura, T.; Yatani, H. "Application of virtual reality force feedback haptic device for oral implant surgery." Clinical oral implants research, 17(6), 708-713, 2006.
- [12] Mcneely, W. A.; Puterbaugh, K. D.; Troy, J. J. "Six degree-of-freedom haptic rendering using voxel sampling." ACM SIGGRAPH 2005 Courses. p.42, 2005.

- [13] Morris, D.; Sewell, C.; Barbagli, F.; Salisbury, K.; Blevins, N. H.; Girod, S. "Visuohaptic Simulation of Bone Surgery for Training and Evaluation." IEEE Computer Graphics and Applications, v. 26, n. 6, p. 48-57, 2006.
- [14] Petersik, A.; Pflesser, B.; Tiede, U.; Hoehne, K. H.; Leuwer, R. "Haptic volume interaction with anatomic models at sub-voxel resolution." 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. p.66-72, 2002. IEEE.
- [15] Rhiemora, P.; Gajananan, K.; Haddawy, P.; Dailey, M. N.; Suebnukarn, S. "Augmented reality haptics system for dental surgical skills training." Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology, p.97-98, 2010.
- [16] Ruspini D.;Kolarov K.;Khatib O."The haptic display of complex graphical environments." In Proc. of ACM Siggraph 97, Los Angeles, CA, pp. 345-352, 1997.
- [17] Salisbury, K.; Tarr. C. "Haptic rendering of surfaces defined by implicit functions." ASME Dynamic Systems and Control Division. Vol. 61. 1997.
- [18] Tsai, Ming-Dar; Ming-Shium Hsieh. "Accurate visual and haptic burring surgery simulation based on a volumetric model." Journal of X-ray Science and Technology 18.1 (2010): 69-85.
- [19] Vlasov, R.; Karl-Ingo F.; Franz-Erich W."Haptic Rendering of Volume Data with Collision Determination Guarantee Using Ray Casting and Implicit Surface Representation." Cyberworlds (CW), 2012 International Conference on. IEEE, 2012.
- [20] Wang, Changling Charlie; Charlie CL Wang. "Toward stable and realistic haptic interaction for tooth preparation simulation." Journal of Computing and Information Science in Engineering 10.2 (2010): 9.
- [21] Wang, Dangxiao; Yuru Zhang. "Effect of haptic device's position resolution on stability." Proceedings of EuroHaptics. 2004.
- [22] Wu, J.; Yu, G.; Wang, D.; Zhang, Y.; Wang, C. "Voxel-based interactive haptic simulation of dental drilling." International Design Engineering Technical Conferences & Computer and Information in Engineering Conference. ASME Press, California. 2009.
- [23] Zilles, C. B.; Salisbury, J. K. "A constraint-based god-object method for haptic display." Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on. v. 3, p.146-151, 1995.