Improved simulation of virtual bone drilling surgery with a voxel-based model

Zhenhuan Wang, Yanping Lin, Huajiang Chen and Wen Yuan

August 31, 2018

Improved simulation of virtual bone drilling surgery with a voxel-based model

Zhenhuan Wang[1], Yanping Lin[1*], Huajiang Chen[2], Wen Yuan[2*]

[1]Institute of Biomedical Manufacturing and Life Quality Engineering, State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

[2]Department of Orthopedics, Changzheng Hospital, Second Military Medical University, Shanghai 200003, China

Keywords: VIRTUAL SURGERY SIMULATION, VOXELIZATION, HAPTICS FEEDBACK, COLLISION DETECTOIN

Abstract

Virtual surgery simulation provides a safe, cost-effective and a repeatable alternative to traditional surgical training methods. The simulation is expected to not only simulate the graphical changes in the surgery, but also to provide a haptic response via a force feedback device. The objective of this study is to develop a virtual bone drilling simulation system and focus on the key issues that limit surgical simulations like the virtual bone drilling of spine surgery. A voxel-based bone model is developed and the algorithms (based on the model) are presented in this work. To provide operators with a realistic environment, dual thread technology is used for graphic and haptic rendering of the virtual bone drilling simulation. Such a surgical simulation system can be very helpful in surgical training.

1 Introduction

Bone drilling is one of the most frequently used surgical procedures in orthopedics; it is a skill that must be mastered by orthopedic doctors. For example, in spine surgery, spinal pedicle screw fixation is one of the most important surgical procedures (Fig 1). The angle of the implanted bolt has a very limited range, and any slight deviation of the angle or an improperly applied force may result in tissue or neuron damage, causing irreversible consequences for the patient. Thus, this kind of bone drilling operation requires that the surgeon precisely apply adequate pressure and be very proficient in the surgical procedure. The amount of learning and practice to develop this kind of “dexterity” and operational precision is a key step in a surgeon’s training. In recent years, surgical simulation systems based on force feedback (haptics) has provided a new virtual training approach to help improve novice doctors’ surgical skills. The system is also safe, efficient and versatile. The simulation provides haptic feedback to the operator via a force feedback device and an established virtual model to simulate the surgical object. It is a new surgical simulation and training approach [1,2]. Currently, research into simulation methods for virtual spine surgery systems mainly focus on the 3D reconstruction of the spine model, pre-operation planning and surgical navigation. Current surgery simulation systems can only provide doctors with visual feedback, lacking a real-time multi-dimensional interaction with the virtual environment. The limitation negatively affecting current technology is the lack of haptic feedback. Simulations that do not allow a doctor to hold the operational device cannot create a realistic training scenario and completely avoids the operation-based training process.

This study establishes a virtual bone drilling training system for spinal pedicle screw fixation surgery based on haptic feedback technology. The system simulates the drilling force often encountered during the pedicle screw implantation process and provides a virtual training platform for doctors to efficiently master the operation techniques. Virtual bone drilling operations belong to the category of hard tissue simulation in operation simulations. This type of simulation is more focused on the precise simulation of the drilling force, which is different from soft tissue operation simulations where the focus is often on the tissue deformation and visualization of the blood flow in blood vessels. The key to a precise simulation of the drilling force lies in the construction of a precise bone model, real-time haptic interaction and visual feedback. This article focuses on the investigation of two key issues: the design of the voxelization algorithm of the spinal bone and the collision detection algorithm. In this work, we use dual thread programming to ensure precision and real-time representation of haptic and visual rendering.

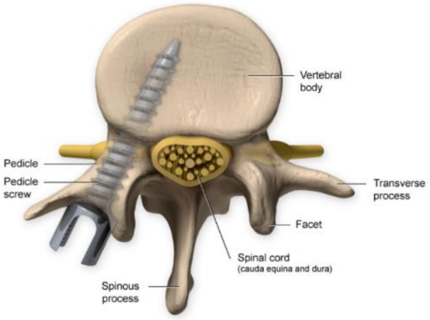

Fig.1 Spinal pedicle screw fixation

2. Related work

Since the 1980’s, surgical simulation technology has slowly emerged into the mainstream educational training market. In 1990, Del et al. [3] created the first worldwide virtual surgical training system; it was used to observe the process and results of joint transplant operations. In 1993, the first virtual abdominal operation training system was established by Satava et al. [4]. Limited by the immaturity of the virtual reality technology, and the computer hardware of the time, the interaction between the user and the simulation was not satisfactory. However, this negative experience paved the way for later technology development in this field. In early virtual surgical training systems, haptic rendering was mostly based on a surface model. This usually involved a triangulated-surface model that is commonly used to model the outer profile of an object. The stiffness information was added to the surface model separately. Massie et al. [5] used a surface mesh model that added a single layer with stiffness information to model the virtual objects. This is the most common physical model used in surgical simulations. This model is mainly used for simple force feedback rather than precise force feedback because of the lack of anisotropic material information of the bone. Well-known anatomical knowledge suggests that bone material can be divided into cortical bone and cancellous bone, and that the compact and hard cortical bone is wrapped around the light porous cancellous bone. Because of the different properties of the two materials, the force applied to drill the bone must be nonhomogeneous. To render different feedback forces, Cho et al. [6] proposed a multi-layer surface model that had gradually changing material. This model simulated the material characteristics of cortical bone and cancellous bone, but the material removal step, which occurred during bone drilling, was not simulated.

In recent years, voxel models have been widely used in hard tissue modeling. In 2006, Girod [7] constructed a voxel model made up of 0.5mm voxels that were based on the CT images. Two sets of models–a surface and body model–were used to model the spine. A surface model was used for graphical rendering, while a body model was used for haptic rendering. Additionally, many researchers worked on establishing precise (exact) bone voxel models for virtual surgery systems. Li et al [8] showed a strong correlation between the gray values of CT image and the bone density, which is one of the most important parameters for calculating the feedback force of the bone drilling operation. Xing et al [9] came up with a bone modeling method based on a CT image by mapping the grayscale information, of the CT image, to the corresponding bone density to establish a precise physical bone model. This method can match the characteristics of the material to the internal data information of the bone model so that an anisotropic physical bone model can be established.

Over the past years, many virtual simulation systems with visual and haptic feedback have been developed [10–13]. Based on the previous work of other researchers and to obtain more accurate virtual bone drilling force in the simulation of spinal drilling surgery simulation, this study mainly studies the virtual drilling simulation technology based on voxel models. An octree data structure was used to store these voxels, and then a collision detection algorithm based on the octree data set was designed with dual thread programming to provide a continuous and stable visual and haptic interaction for virtual bone drilling surgery.

3 Methods

3.1 Establishment of voxel-based bone model

The key to an accurate simulation of drilling force is to establish a precise physical bone model. The material in bone tissue is anisotropic, the surface is cortical bone with high density and hardness, and the interior is soft cancellous bone with low density. Compared to the surface model, the voxel model has internal material information, which can better represent the actual bone. The voxelization of the 3D object has been extensively studied since the last century [14,15]. In this study, a voxelization method was designed and the voxels were stored via the octree data structure.

3.1.1 Data preprocessing

This study is based on the CT scan image data of a patient. After image registration and segmentation, via the image segmentation software MIMICS, the bone model is reconstructed with MAGICS software. The size of the CT image is 512*512, and the corresponding layer spacing is 0.625 mm. After the 3D reconstruction, the triangular facets and the vertex information of the bone surface model are obtained.

3.1.2 Spatial subdivision based on octree

Octree structure has a wide range of applications in computer graphics, image processing and other fields [16]. Its principle is discussed in earlier literature [17].

In this study, the bounding cube box of the bone is subdivided, based on octree, and the voxel is generated via recursive subdivision.

First, the reconstructed bone surface model is parsed. By comparing the vertex coordinates of triangular facets, the smallest axially-aligned cube (containing the bone as the bounding box) is created. Considering many collision detection methods and voxel deletion in the simulation of bone drilling, the non-coding octree data structure was adopted to store and manage voxels. Octree nodes are voxels that are defined as follows:

```
typedef struct OctreeNode
{
   double Di;
   double xMin, xMax;
   double yMin, yMax;
   double zMin, zMax;
   int status;
   struct OctreeNode *top_left_front, *top_left_back;
   struct OctreeNode *top_right_front, *top_right_back;
   struct OctreeNode* bottom_left_front, bottom_left_back;
   struct OctreeNode *bottom_right_front,bottom_right_back;
}OctreeNode;
```

Struct OctreeNode * defines the node and its eight sub-nodes, Di represents the bone density values and xMin, xMax, yMin, yMax, zMin and zMax represent the node's vertex coordinates. A status represents the state of the node: internal voxels, external voxels or surface voxels, which correspond to the voxels inside, outside or on the surface of the bone model, respectively. All the operations (in the simulation) are performed at the leaf node. This step takes place so that the state value of the intermediate node is ignored. Only the state value of the leaf node is meaningful.

$$status = f(x, y, z) = \begin{cases} 0 & \text{(internal voxel)} \\ 1 & \text{(surface voxel)} \\ 2 & \text{(external voxel)} \end{cases}$$

The specific process of spatial subdivision based on octree is as follows:

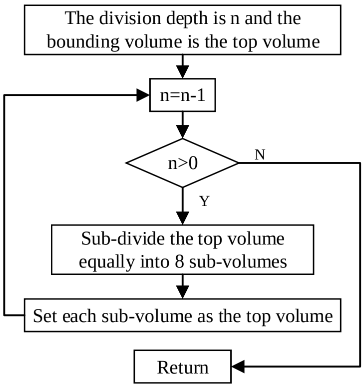

Fig.2 Spatial subdivision based on octree

In this study, the octree depth is n=9 and 16777216 voxels are generated after the division. The side length of the bounding cube box is 93.5779 mm, so the voxel spatial resolution (or the size of the voxel) is 0.36 mm.

3.1.3 Voxelization of the bone

After the voxels have been generated via the spatial subdivision, the next step is to map the bone and voxels. The step is to voxelize the surface of the bone. This is accomplished by traversing the voxels in the voxel space to search for voxels that intersect at the triangular facets of the bone surface. Thus, a spatial intersection algorithm is designed. If one of the following conditions is satisfied, the voxel is determined to be a surface voxel:

1) At least one of the 12 edges of a voxel intersects with one or more triangular facets;

2) There is no intersection point between any edge of voxel and any triangular facet, but there are one or more vertices of triangular facets in the voxel.

After surface voxelization of the bone, voxel-filling is performed. We then select the appropriate topological relation of voxels [18] to fill the internal bone: 6-adjacency constraints are the strongest. These are surface constraints, followed by 18-adjacent surface or line constraints, with the weakest constraint being the 26-adjacent surface, line or point constraints. Thus, we show that boundary overflow may occur when the filling step is performed (by using weakly-constrained adjacent topological relations) so the flood-filling algorithm is based on a 6-adjacency topology that is used to fill the bone, which does not produce boundary overflow: First, a voxel in the bone model is found in the voxel space and then a search for the six adjacent voxels around the voxel is initiated. If the voxel is not labeled as a surface voxel, it will be labeled as an internal voxel and the program will search continuously for its six adjacent voxels until the internal voxel of the bone is fully labeled.

```
void FloodFill(int x, int y, int z)
{
  OctreeNode* temp = GetLeafPtr(x,y,z);
  if (temp->status ！ = 1)
     temp->status = 0;
  FloodFill(x-1,y,z);
  FloodFill(x+1,y,z);
  FloodFill(x,y-1,z);
  FloodFill(x,y+1,z);
  FloodFill(x,y,z-1);
  FloodFill(x,y,z+1);
}
```

GetLeafPtr(x,y,z) is used to find the voxels corresponding to the coordinates (x, y, z) in the voxel space. The status, as previously defined, is 0, 1 or 2, and correspond to the internal surface and external voxels, respectively.

After determining the surface voxel and the internal voxel, and to save computer storage space and improve the efficiency of subsequent collision detection, the external voxels of the bone were removed from the octree structure. The deleted external voxels accounted for about 91% of the total octree leaf nodes, and 1454976 voxels were left to represent the bone model after the external voxels were deleted. Next, bone density was calculated from the gray values of a CT image. After judging whether each voxel belongs to cortical or cancellous bone, the Di of the voxel was set to 0.95 g/cm^3 and 0.25 g/cm^3, respectively. Then, the voxelization of the bone model was completed and the computer spent about 7 seconds. The complete process of establishing bone physical model is as shown in Fig.3.

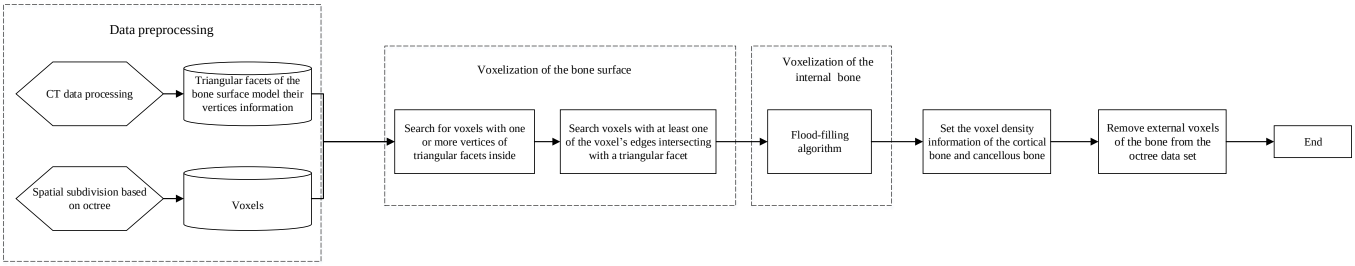

Fig.3 The complete process of establishing voxel-based bone model

3.2 Haptic feedback simulation

Good haptic feedback is an important requirement in virtual bone drilling simulations. In this study, we use haptic rendering, based on a force feedback device called Omega 6, produced by Force Dimension company in Switzerland. Omega 6 has the function of full gravity compensation and drift correction. The precision of the translational motion is 0.01 mm and the precision of the rotational motion is 0.09 degrees. Thus, we can provide a high precision force feedback system. In this study, a collision detection algorithm and voxel removal algorithm are designed to ensure real-time force interaction.

3.2.1 Collision detection and force feedback

In recent years, collision detection algorithms have been widely and increasingly studied [19-22]. The object of this study is to develop an octree that stores voxels of the bone model so that collision detection is carried out between the force feedback device and the octree. A recursive method can be used to detect whether octree nodes contain collision points to determine whether a collision event has occurred. First, we begin at the root node of the octree. It detects whether the node contains a collision point via a virtual drill bit. If so, it further detects whether the eight subnodes of the root node contain a collision point within the virtual drill bit until the leaf nodes located within the bone model are detected. If a collision event is detected, the voxel at the collision point can be found quickly and the voxel information can be read as subsequent feedback for the force calculation. Next, a set of collision detection points is set up at the cutting edges of the drill bit during the force computation (fig. 4 (a)). Every point participates in the collision detection with the voxels of the bone model. The force magnitude and direction of each point is the calculated. The resultant force is the feedback force. Then, the prediction model of the drilling force (constructed by Lin [23] et al.) is used to calculate the drilling force at each collision point. This is done by calculating the density information of the voxels at the collision point and by determining the feed speed of the drill via the expression : $lnu = a_0 + a_t lnt_c + a_v lnv + a_\alpha \ln(1 - \sin\alpha_n)$, where u is the specific normal pressure during drilling and is dependent on the uncut chip thickness t_c, normal rake angle α_n, and cutting velocity v. The other parameters, $a_{0,} a_t, a_v, a_\alpha$, are density dependent and are obtained from Table 2 of the Lin article [23] (each is related to a density value).

(a)

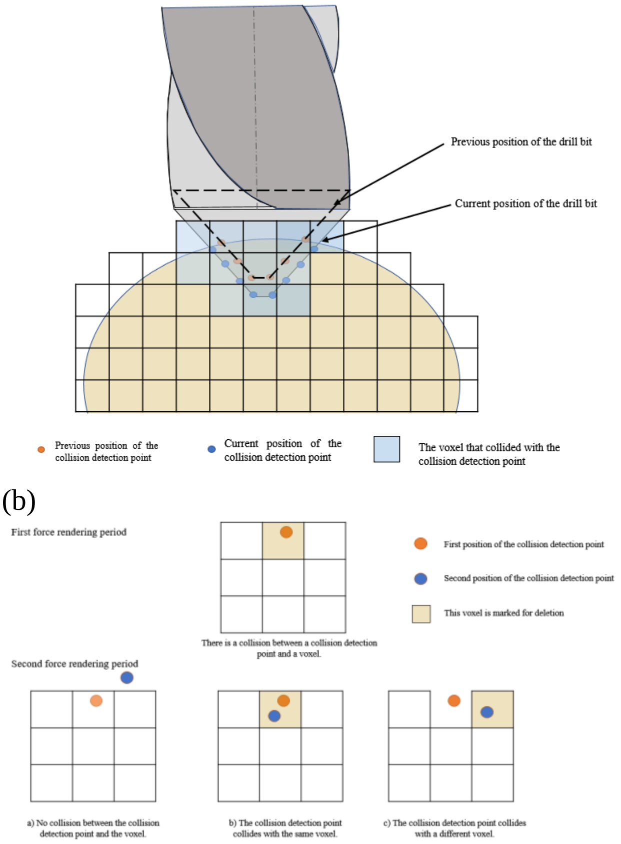

Fig.4 A simulated drilling procedure:(a) Collision detection point; (b) Three scenarios of collision detection

3.2.2 Voxel removal algorithm

After the virtual drill force is transmitted to the force feedback device, the voxel that collided with the drill bit needs to be removed from the octree data set. However, the method for directly removing the voxel will lead to a sudden disappearance of the drilling force. The discontinuity of the feedback force may even cause an unstable flutter of force feedback devices. This research improves the voxel removal algorithm (Fig. 5) to solve this problem.

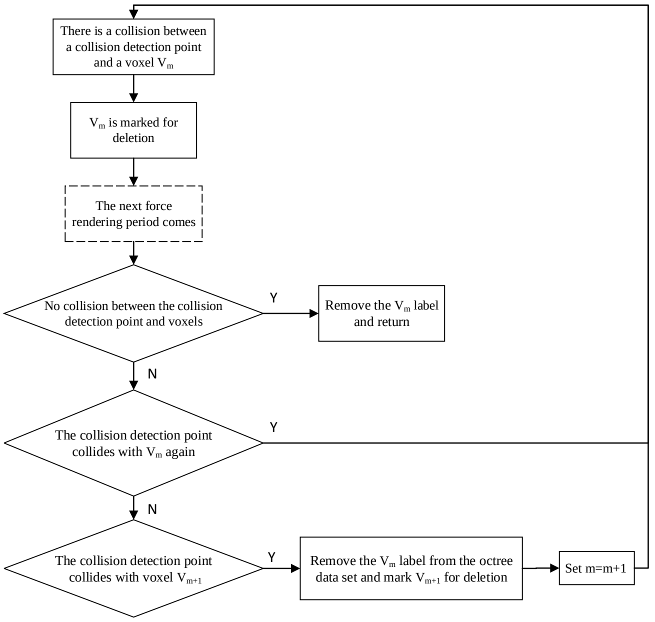

Fig.5 The voxel removal algorithm

When the collision between the voxel and the virtual drill is detected, the voxel is marked for deletion. The voxel can still detect collisions and participates in the force feedback calculation. When the next force-rendering period arrives, there are three possible scenarios (fig. 4 (b)): a) no collision between the collision detection point and the voxel is detected; b) the collision detection point collides with the same voxel again; c) the collision detection point collides with a different voxel. For case a), the voxel mark is removed; for case b), nothing is done; for case c), the initial voxel is deleted from the octree data set and the second voxel (that was collided with) is marked for deletion. Thus, a continuous and stable drilling bone force in the virtual bone drilling simulation is achieved.

For the incomplete octree, with a depth of 9, the detection times of a collision event increase by up to 65 times. The computer can complete collision detection in 1ms; thus, the simulation can find the voxel of the collision point and calculate the resultant force of all the collision detection points in real-time. The feedback force is passed to the force feedback device via the communication protocol and the collision voxel is removed from the octree. In this way, the operator achieves a real-time haptic interaction with the virtual surgery environment.

4 Results and discussion

In this study, a voxel-based bone model is established and based on CT images of spinal bone. A collision detection algorithm and voxel removal algorithm are designed. With the help of force feedback equipment and OpenHaptics API, a bone drilling surgery simulation system is developed. The graphic and haptic achieved by using dual threads. The framework of the virtual bone drilling simulation system is as follows:

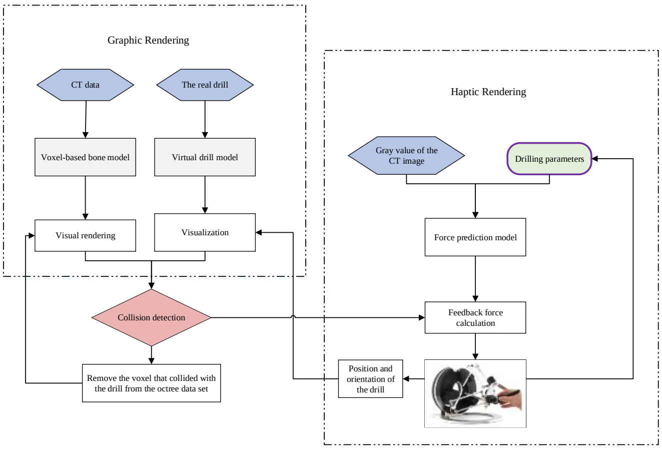

Fig.6 The system framework

An octree data structure is used to generate the voxels of the bone model and manage the voxels. After voxelization, the bone model is composed of 1454976 cubic voxels with a 0.36-mm side length. Using OpenGL graphical interface, we set the graphic frame rate to 300 Hz. The computer environment is:

- Windows10 operating system
- 64-bit Intel○R Core○TM i7-44720HQ CPU @ 2.6 GHz
- 16GB memory
- NVIDIA GEFORCE GTX960 graphics card

Real-time haptic interaction with the help of a force feedback device (Omega 6), a collision detection algorithm and a voxel removal algorithm were used to generate force feedback with an h haptic frame rate of 1000 Hz. Fig 7 shows a picture of the operator using the system to simulate spinal bone drilling surgery. section.

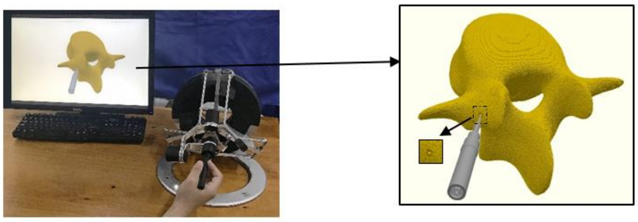

Fig.7 The operator operates the simulation system.

5 Conclusion

By considering a spinal bone drilling operation as an example, this study focuses on the key issues of virtual bone drilling surgery simulation. The core outcome of our work is the establishment of a voxel-based bone model and the subsequent algorithm suite based on the model. With the help of force feedback equipment and OpenHaptics API, a continuous force feedback system simulated the bone drilling operation. Dual thread technology ensured real-time rendering of visual-haptics. The graphic frame rate reached 300 Hz and the haptic frame rate reached 1000 Hz. Thus, this simulation can be used to build a training platform to help novice doctors to improve their surgical skills.

Future research will include experiments to measure the drilling force during real-world bone drilling and build an accurate force prediction model. Additional work will also be conducted to classify the bone tissue and establish more accurate density levels. These modifications may help to render a more realistic haptic simulation.

6 Acknowledgements

This study was funded by the National Natural Science

Foundation of China (Grant Nos. 51575343 and 81372017).

7 References

[1] Hamza-Lup, F., Bogdan, C., Popovici, D., Costea, O.: 'A Survey of Visuo-Haptic Simulation in Surgical Training'. eLmL 2011, Gosier, Guadeloupe, France, February 2011, pp. 57–62

[2] Coles, T.R., Meglan, D., John, NW.: ' The role of haptics in medical training simulators: a survey of the state of the art ', IEEE T Haptics, 2011, 4, (1), pp. 51–66

[3] Delp, SL., Loan, JP., Hoy, MG., et al.: ' An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures ', IEEE Trans Biomed Eng, 1990, 37, (8), pp. 757–767

[4] Satava, R.: ' Virtual reality surgical simulator ', Surg Endosc, 1993, 7, (3), pp. 203–205

[5] Massie, T., Salisbury, J.: ' The phantom haptic interface: A device for probing virtual objects '. Proc. ASME Winter Annual Meeting, Chicago, US, 1994, pp. 295–301

[6] Cho, J., Jung, H., Lee, K., et al.: ' Haptic rendering of drilling into femur bone with graded stiffness '. FBIT 2007, Jeju Island, Korea, Octorber 2007, pp. 525–530

[7] Morris, D., Sewell, C., Barbagli, F., et al.:' Visuohaptic simulation of bone surgery for training and evaluation', IEEE Comput Graph, 2006, 26, (6), pp. 48-57

[8] Li, D., Mao, S., Khazai, B., et al.:' Non contrast cardiac computed tomography image-based vertebral bone mineral density: The Multi-Ethnic Study of Atherosclerosis (MESA)', Acad Radiol, 2013, 20, (5), pp 621-627

[9] Xing, Q., Chen, J., Li, J., et al.: ' A real time haptic simulator of spine surgeries '. VRST 2015, Beijing, China, November 2015, pp. 121–124

[10] Tsai, MD., Hsieh, MS., Tsai, CH.: ' Bone drilling haptic interaction for orthopedic surgical simulator', Comput Biol Med, 2007,37(12), pp. 1709–1718

[11] Konukseven, EI., Önder, ME., Mumcuoglu, E., et al.: ' Development of a visio-haptic integrated dental training simulation system', J Dent Educ, 2010, 74(8), pp 880–891

[12] Wang, DX., Zhang, YR., Wang, Y., et al.: 'Haptic rendering for dental training system', Sci China Ser F, 2009, 52(3), pp. 529–546

[13] Arbabtafti, M., Moghaddam, M., Nahvi, A., et al.: 'Physics-based haptic simulation of bone machining ', IEEE Trans Haptics, 2011, 4(1), pp. 39–50

[14] Kaufman, A.: ' Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes '. ACM SIGGRAPH 1987, Anaheim, USA, 1987, pp. 171–179

[15] Huang, J., Yagel, R., Kurzion, Y.: ' An accurate method to voxelize polygonal meshes '. IEEE Volume Visualization'98 , Chapel Hill, North Carolina, USA, October 1998, pp. 119–126

[16] Cui, J., Chow, YW., Zhang, M.: 'A voxel-based octree construction approach for procedural cave generation', IJCSNS, 2011, 11, pp.160-168

[17] Meagher, D.: ' Geometric modeling using octree encoding ', Comput Graph Image Process., 1982, 19, (2), pp. 129–147

[18] Kong, TY., Rosenfeld, A.: ' Digital topology: introduction and survey ', Computer Vision Graphics & Image Processing, 1989,48(3), pp.357-393

[19] Govindaraju, N., Redon, S., Lin, M.: ' CULLIDE: interactive collision detection between complex models in large environments using graphics hardware '. Graphics Hardware 2003, San Diego, California, USA , July 2003, pp. 25–32

[20] Haddadin, S., Albu-Schaffer, A., Luca, AD., et al.: ' Collision detection and reaction: A contribution to safe physical Human-Robot Interaction '. IEEE/RSJ 2008, Nice, France, September 2008, pp. 3356–3363

[21] Baciu, G., Wong, SK., Sun, H.: 'RECODE: An Image-based Collision Detection Algorithm', J Visual Comp Animat., 1999, 10, (4), pp. 181–192

[22] Jr, GV.: ' Back-face culling applied to collision detection of polyhedra ', J Visual Comp Animat., 1994, 5, (1), pp. 55–63

[23] Lin, Y., Chen, H., Yu, D., et al.: 'A predictive bone drilling force model for haptic rendering with experimental validation using fresh cadaveric bone '. Int J CARS, 2017, 12, pp.91–98