CARS 2002 – H.U. Lemke, M.W. Vannier; K. Inamura, A.G. Farman, K. Doi & J.H.C. Reiber (Editors)

Realistic haptic volume interaction for petrous bone surgery simulation

A. Petersik[a], B. Pflesser[a], U. Tiede[a], K.H. Höhne[a], R. Leuwer[b]
[a] Institute of Mathematics and Computer Science in Medicine (IMDM)
[b] ENT-Clinic
University Hospital Hamburg-Eppendorf, Germany
petersik@uke.uni-hamburg.de

Abstract

In this paper, a new approach for haptic volume interaction with high resolution voxel-based anatomic models is presented. The haptic rendering is based on a multi-point collision detection approach which provides realistic tool interaction with the models. Both haptics and graphics are rendered at sub-voxel resolution, which leads to a high level of detail and enables the exploration of the models at any scale. Forces are calculated at an update rate of 6000 Hz and sent to a 3-Degree-of-Freedom (3-DOF) force-feedback device. Compared to single-point based haptic rendering, the unique approach of the multi-point collision detection in combination with sub-voxel rendering provides more realistic and very detailed haptic sensations. As a main application, a simulator for petrous bone surgery was developed. With a simulated drill, bony structure can be removed and the access path to the middle ear can be studied in a realistic manner.

Keywords: haptic, collision detection, volume interaction

1. Introduction

The sense of touch is essential in a wide field of medical applications, like surgery simulators. In contrast to our other senses it allows us to simultaneously explore and interact with our environment. Today most applications concentrate on the simulation of elastic deformations of soft tissue. The simulation of material removal in medical applications is a less developed field and simulation systems either do not include cutting operations at all, or in a simplified manner, which does not provide the 'look and feel' close to a real incision.

Moreover haptic rendering is mostly based on traditional computer graphics methods where objects are represented by polygons only. Creating detailed polygonal models of organs result in a huge number of polygons which increases computation time for collision detection dramatically. However for realistic haptic rendering a collision detection algorithm with a constant computation time is essential. Choosing a volume based model, the computation time for a collision detection is independent of the complexity of the scene. Additionally a volume based model allows the simulation of interactive cutting operations and the display of arbitrary cut planes.

Furthermore we realized that today 3-DOF haptic rendering is mostly point-based, i.e. only one point is used to calculate collisions and forces. This induces several problems:

- Discontinuities (e.g. sharp edges) on the surface can lead to discontinuities in the haptic display.
- The virtual tool can reach points which can not be reached by the simulated real world tool. (A large drill could enter a small hole.)

One goal of the work presented in this paper is to develop a haptic rendering algorithm which is able to display arbitrary complex anatomic models with high realism and accuracy. Another goal was to not only enable the haptic exploration of the anatomic models but also to be able to modify those models interactively with simulated real world tools in a realistic manner. Using the developed algorithms, we implemented a petrous bone surgery simulator, where a simulated drill can be used both to explore and to manipulate the anatomic model.

2. Methods

To develop a simulator which allows drilling into the bone the following points concerning haptics had to be considered:

- Haptic rendering should be based on a multi-point collision detection to allow a realistic tool-object interaction.
- To enable a realistic haptic interaction while modifying the models with drill like tools, an algorithm is needed which calculates realistic drilling forces.

2.1 Data representation

The model of the petrous bone is represented by attributed voxels (volume elements) which have a size of $0.33mm^3$. The attributes describe data like density values and associated organ. The associated organ is determined during the semi-automatic, threshold based segmentation process.
Our voxel-based representation does not contain an explicit representation of the object surfaces. The surfaces are calculated based on the segmentation data. This is done by a ray-casting algorithm [6] which renders isosurfaces at sub-voxel resolution based on the partial volume effect and density value of the voxels. The same ray-casting algorithm is used to determine the surface for both graphic and haptic rendering.

2.2 Haptic surface rendering

In our implementation of the petrous bone surgery simulator we are using a sphere-shaped tool, which simulates a drill. To achieve a realistic haptic rendering, collisions between the tool and the static scene must be computed and a collision-free position must be determined. Then a force which pushes the haptic device to the collision-free position must be applied. For a realistic haptic tool-object interaction, a multi-point collision detection algorithm is needed.
To implement the multi-point collision detection, the tool is represented by a number of sample points which are distributed at preferably equal distances over the tool surface. Each of these points is checked whether it collides with the objects or not. Additionally

every point has an associated normal vector which is pointing to the inside of the tool. All inward pointing surface normals have the same length.
The inward pointing vectors are used to find the static object's surface. An advantage of this approach is that no computation of surface normals for the voxel-based anatomic object is required. To get a good compromise of adequate tool representation and computation time, we use 26 sample points on the sphere's surface.
Whenever a collision between tool and static objects occurs the following two parameters must be computed:

- Direction and
- Magnitude of collision force

Both variables must be computed with high precision to allow a realistic interaction for e.g. petrous bone surgery. Our multi-point collision detection algorithm was inspired by the work described in [2]. However our approach differs from this work in several points. While our model is also using a voxel representation for the static objects, the exact location of the surfaces is calculated by a ray-casting algorithm at sub-voxel resolution (see 2.1). This leads to a more precise calculation of both force direction and magnitude. The algorithms presented in [2, 4] can not provide the precision which is needed in our case, since the static objects are voxelized in a binary manner. Additionally our sub-voxel approach leads to a stable contact situation, even under sliding motion without having to reduce the device's bandwidth by methods like e.g. "virtual-coupling" or force-averaging.

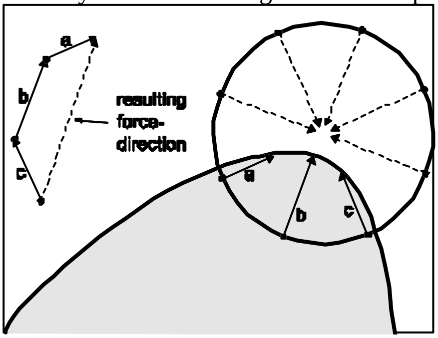

Fig. 1: Calculation of collision forces on the object surface

Another improvement we have made is the representation of the tool. While the dynamic object in [2] is voxelized and the center points of the voxels are used for the collision detection, we are using sample points which are located exactly on the surface of the tool.
To calculate the collision force direction, the surface points of the tool are checked, whether they are colliding with the object. All colliding surface points are traced along the corresponding inward pointing normal until the surface of the object is found or the end of the normal is reached. All vectors found are added and the direction of the sum vector is the direction of the force vector which must be applied to the haptic device (fig. 1). As stated in [2], the summation of the vectors found would lead to force instabilities. As more of the tool's contact points collide with the static object, more stiffness is the result which creates a less stable contact situation. To avoid instabilities, we use an improved algorithm which averages the vectors depending on the number of found vectors.

2.3 Proxy object algorithm

Any haptic device has a limited maximum force which can be applied to it. When the user pushes harder than this limit, it is possible that the physical position of the tool immerses completely into the virtual model. In this case calculation of the force direction as described above is not possible anymore.

To overcome that limitation a modified proxy object algorithm [7, 5] was implemented. The idea behind the proxy is to store a position where no collision between tool and scene occurs. When the tool moves in freespace, tool and device position are identical. But when the device immerses into an object, the proxy remains on the object's surface.

In order to update the proxy position on the static object's surface while the device is moving, the distance between the device position and the proxy must be locally minimized by regarding the surface constraints. Since searching for the local minimum would be computationally too expensive in our model, a simplified algorithm was implemented. Whenever more than a certain number of surface sample points of the dynamic object are in contact with an object, the path between proxy and new tool position is traced until the object surface is found and the resulting force can be calculated as described above. Since the difference between two successive positions is very small, this approximation gives a realistic feeling of the virtual objects.

2.4 Volume modification

Our freeform volume modification algorithm which is descibed in detail in [3] is working with sub-voxel precision. The parts removed are modeled by simulating the partial volume effect. This produces realistic structures even when using very small tools.

2.5 Calculation of drilling forces

Since the volume modification algorithm updates the volume at a rather low frequency, we can not use the same haptic rendering algorithm as described above to calculate a realistic force while drilling.

An approach to overcome that problem is to use a force which is opposite to the drilling direction and takes several parameters into account to improve the sensation of the drilled structures.

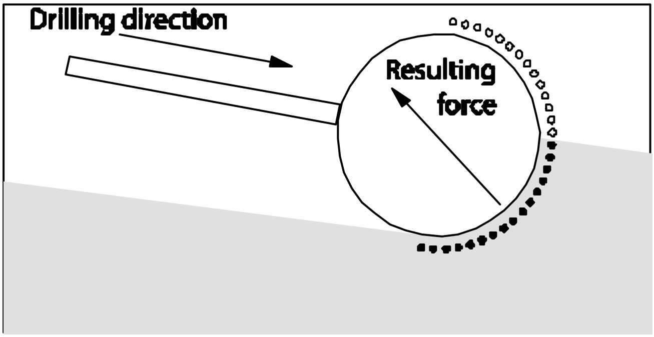

Fig. 2: Adaption of magnitude and direction of the drilling force

- The higher the drilling speed, the higher must be the drilling force.
- The more volume is removed, the higher must be the drilling force.
- Depending on the position of the removed structures in relation to the drill, the direction of the drilling force must be adapted (fig. 2).

Additionally a vibration is modulated onto the drilling force to improve the sensation of the drilling procedure. By changing parameters, different types of drills, like diamant and metal drills can be simulated.

3. Results

With our approach we achieve a high quality haptic rendering of non-deformable arbitrary complex anatomic models [1]. The perception of spatial relationships is greatly enhanced with haptic feedback. Even very small surface details or small objects like nerves and vessels can be sensed realistically due to our sub-voxel based approach. The main focus of our work was the interaction with non-deformable material (e.g. bone). With the algorithms presented in this paper we developed a system for the simulation of petrous bone surgery. In this simulator, the haptic device is used to simulate drilling into the mastoid bone to get access to the middle-ear. The monitor of the simulator is mounted above and the haptics device below a mirror which allows the surgeon to work at the location where the scene is displayed (fig. 3). The spatial perception is enhanced by using stereoscopic viewing. The configuration of the simulator provides a position of the surgeon's hands, patient orientation and viewing direction which is similar to the real procedure. The simulator is of great assistance for learning the complex anatomy of the petrous bone.

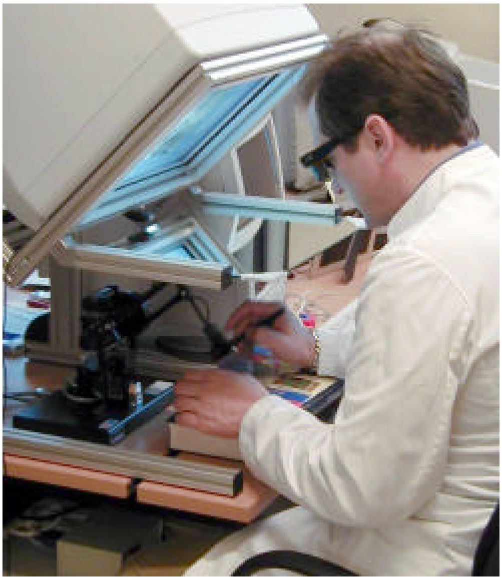

Fig. 3: Training with the simulator

4. Conclusions and future work

We presented an approach for realistic haptic rendering and volume interaction with anatomic models. It overcomes several problems of single-point based haptic rendering. The multi-point based collision detection approach makes tool-object interactions more realistic, especially for the use in surgery simulations. Another important advantage is that surface details can be sensed as expected from the graphical representation. Sub-voxel rendering leads to both a graphical and haptic realistic, high detailed and congruent display. The volume interaction uses a sub-voxel based algorithm which allows the use of small tools, like drills used in petrous bone surgery.

While only sphere-shaped tools are realized in the petrous bone surgery simulator, future implementations will extend this to more general shapes. The calculation of the magnitude of the collision force will be investigated further to improve haptic rendering at locations with many tool object intersections as they appear especially in deep clefts.

Other improvements will be done to speed up graphical rendering of the modified structures, to allow for faster movements while drilling.

References

[1] K. H. Höhne, B. Pflesser, A. Pommert, M. Riemer, R. Schubert, T. Schiemann, U. Tiede, and U. Schumacher. A realistic model of human structure from the Visible Human data. Meth. Inform. Med., 40(2):83-89, 2001.

[2] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. Computer Graphics (SIGGRAPH99 Proceedings), pages 401-408, 1999.

[3] B. Pflesser, U. Tiede, K. H. Höhne, and R. Leuwer. Volume based planning and rehearsal of surgical interventions. In H. U. Lemke, M. W. Vannier, K. Inamura, A. G. Farman, and K. Doi, editors, Computer Assisted Radiology and Surgery, Proc. CARS 2000, volume 1214 of Excerpta Medica International Congress Series, pages 607-612. Elsevier, Amsterdam, 2000.

[4] M. Renz, C. Preusche, M. Potke, H.-P. Kriegel, and G. Hirzinger. Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm. Proc. of the Eurohaptics Conference, pages 149.154, 2001.

[5] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. Proc. of ACM SIGGRAPH, pages 345-352, 1997.

[6] U. Tiede, T. Schiemann, and K. H. Höhne. High quality rendering of attributed volume data. In D. Ebert, H. Hagen, and H. Rushmeier, editors, Proc. IEEE Visualization '98, pages 255-262. IEEE Computer Society Press, Los Alamitos, CA, 1998.

[7] C. Zilles and K. Salisbury. A constraint-based god object method for haptics display. Proceedings of IEEE/RSJ, 1995.