In N. Ayache, H. Delingette: Surgery Simulation and Soft Tissue Modeling, Proc. IS4TM 2003.
Springer-Verlag, Berlin, 2003, 194-202

Realistic Haptic Interaction in Volume Sculpting for Surgery Simulation

Andreas Petersik[1], Bernhard Pflesser[1], Ulf Tiede[1], Karl-Heinz Höhne[1], and Rudolf Leuwer[2]

[1] Institute of Mathematics and Computer Science in Medicine (IMDM),
{petersik, pflesser, tiede, hoehne}@uke.uni-hamburg.de
[2] ENT-Clinic,
leuwer@uke.uni-hamburg.de
University Hospital Hamburg-Eppendorf, Germany

Abstract. Realistic haptic interaction in volume sculpting is a decisive prerequisite for successful simulation of bone surgery. We present a haptic rendering algorithm, based on a multi-point collision detection approach which provides realistic tool interactions. 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. With a simulated drill bony structures can be removed interactively. The characteristics of the real drilling procedure like material distribution around the drill are considered to enable a realistic sensation. All forces are calculated at an extra high update rate of 6000 Hz which enables rendering of drilling vibrations and stiff surfaces. As a main application, a simulator for petrous bone surgery was developed. With the simulated drill, access paths to the middle ear can be studied. This allows a realistic training without the need for cadaveric material.

1 Introduction

In surgery simulation it is often not sufficient to deal only with a graphical representation of the anatomy. For most applications the sense of touch is necessary for realistic interaction. In contrast to our other senses it allows us to simultaneously explore and interact with our environment. Most applications of surgery simulation concentrate on the simulation of elastic deformations of soft tissue. In contrast to that, the simulation of material removal (sawing, milling, drilling) as performed in bone surgery is a less developed field and existing systems do not provide the 'look and feel' close to the real procedure.

Realistic look and feel means that on one hand the irregular shaped cut surfaces must have realistic shape and texture in the visual representation, on the other hand the algorithm for haptic rendering must be able to provide detailed sensations of all structures with no delay. Both requirements are fulfilled best by using a volume model. As we have shown earlier, it allows very realistic display of cut surfaces [8]. For realistic haptic rendering it has the decisive advantage over a polygonal representation that computation time for collision detection is independent of the complexity of the scene.

Existing algorithms for haptic rendering are mostly point-based, i.e. only one point of the virtual tool is used to calculate collisions and forces. While this might be sufficient for simulation of deformations by the tip of a tool, it leads to problems in the case of material removal:

- In complex scenes (e.g. with sharp edges) the shape of the tool plays a decisive role for the haptic sensation and cannot be simulated by one single point.
- The virtual tool can reach points which cannot be reached by the simulated real world tool. (A large drill could enter a small hole.)

The purpose of the work presented in this paper is to develop algorithms which can calculate forces for haptic feedback with high resolution and realism which are suitable for material removal from rigid objects. Realistic feeling should be provided while touching the model as well as while actively modifying the model with virtual tools. The algorithms should be able to handle models of arbitrary complexity. The feasibility of the algorithms is to be demonstrated with a petrous bone surgery simulator, where different types of virtual drills can be used both to touch and to surgically modify the anatomic model.

2 Related work

An early approach of haptic interaction with volume visualization was presented in [1]. While this approach allows exploration and modification of the volumetric data, the forces generated are not intended to be a realistic simulation of interacting with materials. Rather, the intent is to convey additional information to the user about the data being explored. Furthermore this approach uses a single-point contact model which does not fulfill our needs for a realistic interaction.

As stated earlier, single-point interaction is not realistic in that it does not prohibit unrealistic situations like entering a small hole with a large drill. Therefore multi-point collision detection approaches were developed [2, 3]. With these approaches, more realistic simulations of tool-object interactions can be achieved.

The voxel-based approach to haptic rendering presented in [2] enables 6-DOF interaction of rigid tool within an arbitrary complex environment of static objects. This approach provides a realistic haptic feedback in regard to the exact shape of the tool. One drawback of this method is that the haptic rendering is at the resolution of the original voxels only which is not sufficient for medical applications. Also modification of the static environment is not included.

In [4] and [5] simulators for petrous bone surgery are presented, but they lack of high resolution rendering. Most structures which are important for a successful petrous bone surgery are not visible.

3 Methods

In order to achieve a realistic haptic surface rendering, collisions between the tool and the static scene must be computed and a collision-free position must be determined. Figure 1 shows a situation where the user moved the virtual tool from position 1 to position 2. Since the tool could not reach position 2 in reality, the rendering algorithm must calculate position 3 which would have been reached in reality. Then a force which pushes the haptic device to position 3 must be applied.

In order to develop a simulator for petrous bone surgery which allows realistic drilling into the mastoid bone the following points concerning haptics had to be considered:

- Haptic rendering should be based on a multi-point collision detection approach to allow realistic tool-object interactions for passive interaction.
- For material removal an algorithm is needed which works with sub-voxel resolution to be able to simulate the effect of small tools as they are used in petrous bone surgery.
- To enable realistic haptic interactions while modifying the models, an algorithm is needed which calculates realistic drilling forces based on parameters like: amount of removed material, distribution of material around the drill etc.

3.1 Data representation

The model of the petrous bone was created from CT-data and is represented by a volume of attributed voxels (volume elements) which have a size of 0.33mm^3. The attributes are density values and membership to an organ. The membership to an organ is determined during the semi-automatic, threshold based segmentation process.

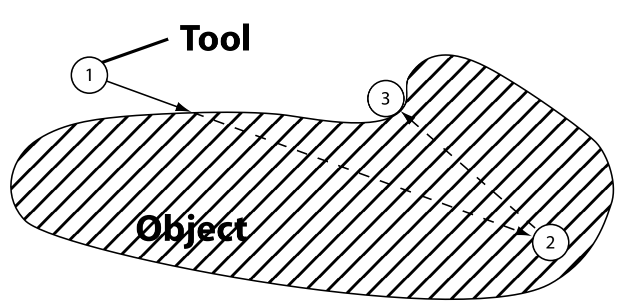

Fig. 1. To get a realistic haptic rendering of surfaces, position 3 must be calculated based on the last device position 1 and the new device position 2.

Since our voxel-based representation does not contain an explicit representation of the object surfaces, the surfaces must be calculated based on the segmentation data. This is done by a ray-casting algorithm [6] which renders iso-surfaces at sub-voxel resolution based on the partial volume effect and density value of the voxels. The sub-voxel approach leads to very detailed surfaces which can be explored at any scale.

3.2 Tool representation

For multi-point collision detection the tool is represented by a number of sample points P_i 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 $\boldsymbol{n_i}$ which is pointing to the inside of the tool.

The inward pointing vectors $\boldsymbol{n_i}$ describe the tool's curvature and can be used by the collision detection algorithm to find the static object's surface.

In our implementation of the petrous bone surgery simulator we are using a sphere-shaped tool, which simulates a drill (Fig. 2). To get an adequate representation of the tool's shape while reducing computation to a minimum, we are using 56 sample points on the tool's surface.

3.3 Haptic surface rendering

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 3.1). This leads to a more precise calculation of force direction and surface location. The algorithm presented in [2] cannot provide the precision which is needed in our cases, since the static objects are voxelized in a binary manner. Since our sub-voxel rendering approach is generating iso-surfaces the problems

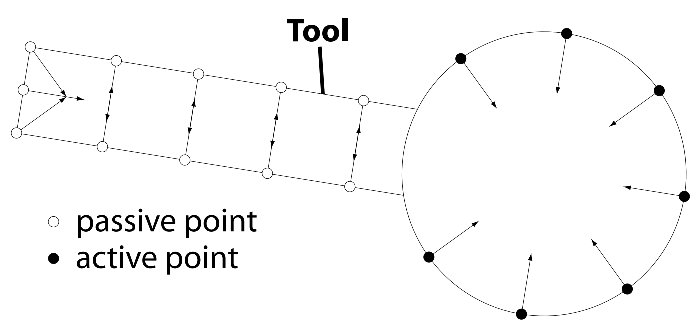

Fig. 2. Tool representation by surface points and inward pointing normal vectors

with discontinuities at voxel boundaries is eliminated. Another improvement we have made is the representation of the dynamic object. 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 dynamic object, which improves the resolution further.

Collision detection algorithm To calculate the collision force direction, all surface points of the tool are checked, whether they are inside or outside the object. All surface points $\boldsymbol{P_i}$ which are inside the volume (Fig. 3, filled dots) are traced along the inward pointing normal until the surface is found. All found vectors $\boldsymbol{f_i}$ are added and the direction of the sum vector $\boldsymbol{f_D}$ is the direction of the force vector which must be applied to the haptic device. As stated in [2], the summation of the found vectors would lead to force instabilities. To avoid this we decided not to use force summing for the calculation of the magnitude of the force vector. Instead we are searching for the longest projection of the vector $\boldsymbol{f_i}$ on the vector $\boldsymbol{f_D}$.

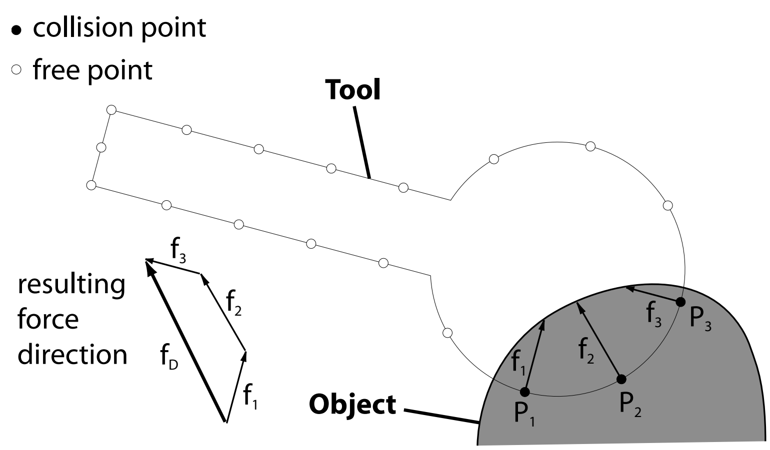

Fig. 3. Calculation of forces during passive interaction. The resulting force direction $\boldsymbol{f_D}$ is calculated by adding the three vectors $\boldsymbol{f_1}$, $\boldsymbol{f_2}$ and $\boldsymbol{f_3}$.

Proxy object algorithm The previously described algorithm works only for small tool object penetrations, when the surface can be found for all collision points. To overcome that limitation a modified proxy object algorithm was implemented. The idea behind a proxy object is to represent the device position by a virtual object which never penetrates objects [7].

Searching for the local minimum to update the proxy position as described in [7] would be computationally too expensive in our model. Thus 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 or one inward pointing vector is completely immersed, the way between proxy and current position is traced until the object surface is found. Starting from this position the way back to the proxy is traced until the number of contacts is below the limit so that the force vector can be calculated as described in chapter 3.3.

3.4 Volume interaction

The sense of touch allows not only the exploration of the anatomy but also the interactive modification of the objects we are touching. In petrous bone surgery the drill allows the surgeon to feel and to drill the bone.

Our freeform volume modification algorithm which is described in detail in [8] is working with sub-voxel resolution. This produces realistic structures even when using very small tools.

Calculation of drilling forces During volume modification, the modified volume must be modelled and the visual renderer must redraw the modified region. Both steps consume too much time to be able to perform the collision detection based on the modified volume. Thus drilling forces must be calculated by a simplified algorithm, which can run asynchronously to the modification process.

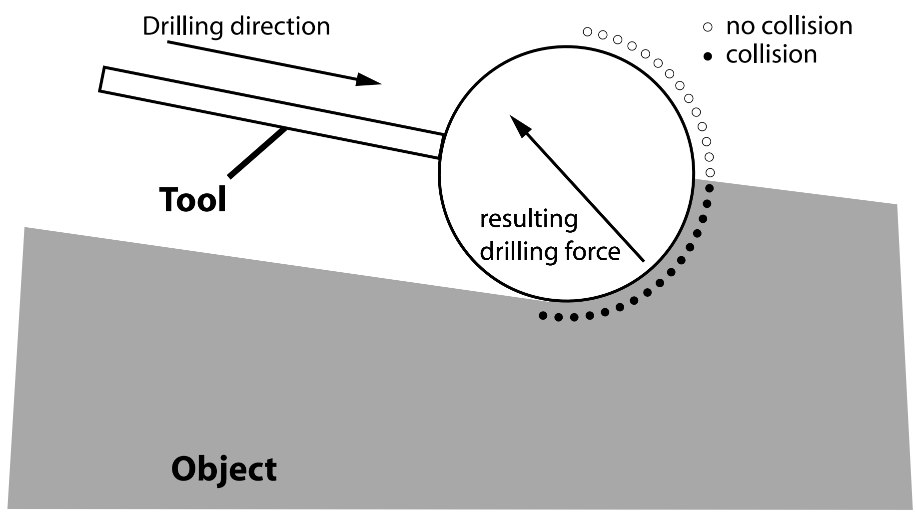

Fig. 4. Calculation of forces during drilling. The resulting force is dependent on collisions with sample points around the drill.

To get a realistic force during drilling, we apply as a first approximation a force which is opposite to the drilling direction and the speed of the drill movement. The vector of this force is pointing from the current position of the tool P_C to the last stored position P_L.

Additionally we consider the material distribution around the drill and the amount of removed material. The material distribution and the amount of removed material is calculated by looking for collisions at sample points in front of the drill (Fig. 4).

The balance point $\boldsymbol{b}$ of the mass removal in tool coordinates can be calculated by adding all colliding active points of the tool $\boldsymbol{A_i}$ and dividing the sum by the number c of colliding vectors found:

$$\boldsymbol{b} = \frac{\sum_{i=1}^{c} \boldsymbol{A_i}}{c} \tag{1}$$

The direction of the drilling force vector $\boldsymbol{f}$ can now be calculated as follows:

$$\boldsymbol{f} = \boldsymbol{P_C} - \boldsymbol{b} \cdot \frac{\|\boldsymbol{P_C} - \boldsymbol{P_L}\|}{\|\boldsymbol{b}\|} \tag{2}$$

In figure 4 you can see one example for such a calculation: Material will be removed and the vector of the drilling force will be rotated about 45 degrees because only one quarter of the spherical drill is in contact with the material.

4 Implementation

The petrous bone surgery simulator was integrated in the VOXEL-MAN [9] system, which provides the anatomic model, high-quality visual rendering and also free-form volume modification. The system was implemented on a dual processor AMD PC with two AthlonMP 1900+ processors. The PC is equipped with 1GB DDRAM. As haptic device we are using a 3-DOF Phantom Premium 1.0A (Sensable Technologies Inc.). Our system is running SuSE Linux 8.0. For connecting the device directly to the system, we are using the open-source Phantom Linux-driver (http://decibel.fi.muni.cz/phantom), which enables haptic update rates between 1 and 10kHz. To enable stereoscopic viewing we are using an Nvidia Quadro2MXR graphics board in combination with ELSA Revelator shutter glasses.

5 Results and Conclusions

With the described algorithms for haptic material removal interactions, haptic realism can be achieved through:

- congruent sub-voxel resolution both for visualization and haptic rendering
- multi-point collision detection
- high haptic update rate of 6000Hz
- consideration of material distribution around the drill
- simulation of drilling vibrations

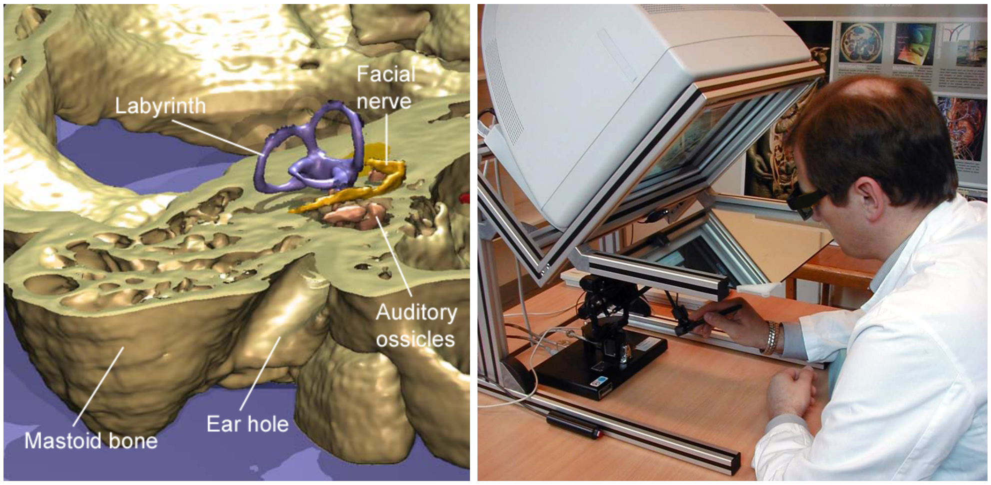

Fig. 5. The petrous bone model was built from CT data (left). The setup of the simulator together with stereoscopic viewing allows a realistic handling close to the real procedure (right).

The implemented simulator for petrous bone surgery has shown that the developed algorithms allow a realistic simulation of material removal procedures which could be used in other medical areas like craniofacial surgery or dentistry.

Of course a quantitative validation of the algorithms is difficult. So far we rely on the judgement of our ENT-surgeons that the haptic feeling of the simulation is nearly indistinguishable from that of the real procedure.

The next step of our research will be the extension of the algorithms to be able to deal with more complex shapes and functions of the tool. An example for this is a saw blade with an active face and a passive backside.

6 Acknowledgements

We are grateful for the help of Andreas Pommert, Martin Riemer, Rainer Schubert and all members of the VOXEL-MAN team.

References

1. Avila, R.S., Sobierajski, L.M.: A haptic interaction method for volume visualization. Proceedings of Visualization 96 (1996) 197–204
2. McNeely, W.A., Puterbaugh, K.D., Troy, J.J.: Six degree-of-freedom haptic rendering using voxel sampling. Computer Graphics (SIGGRAPH99 Proceedings) (1999) 401–408
3. Kim, Y.J., Otaduy, M.A., Lin, M.C., Dinesh, M.: Six-degree-of-freedom haptic display using localized contact computations. Proceedings of 10th Symposium on Haptic Interfaces (2002) 209–216

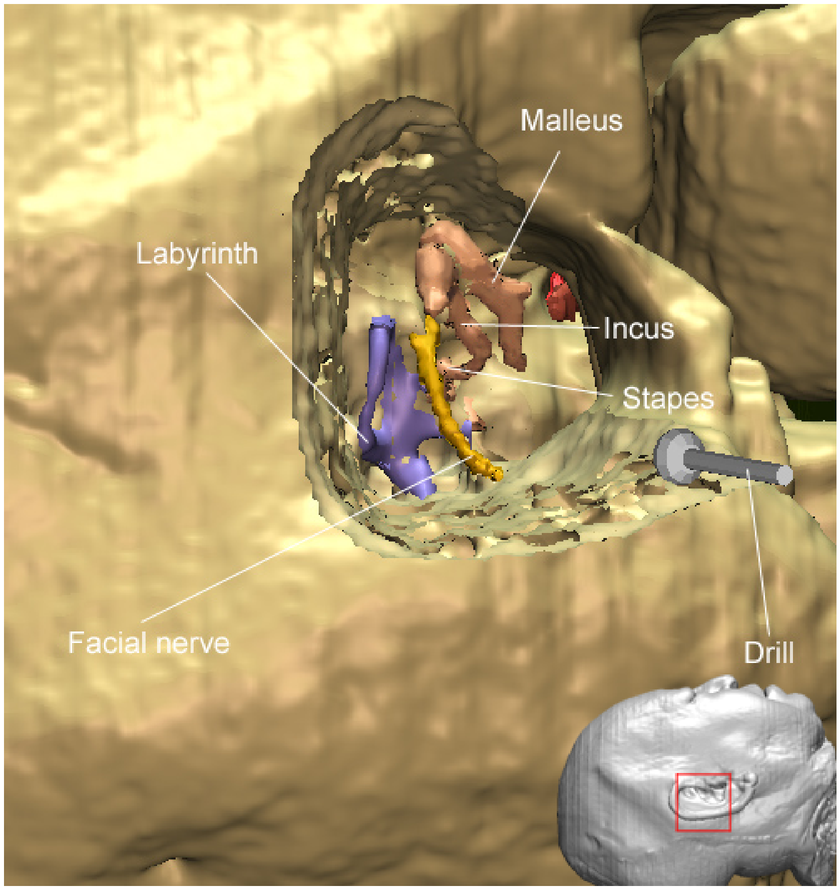

Fig. 6. Simulation of petrous bone surgery. The haptic sensation is congruent to the detailed visualization of the irregular structures.

4. Wiet, G.J., Bryan, J.: Virtual temporal bone dissection. Proceedings of MMVR2000 (2000) 378–384
5. Agus, M., Giachetti, A., Gobbetti, E., Zanetti, G., Zorcolo, A.: Real-time haptic and visual simulation of bone dissection. Proceedings of IEEE VR 2002 (2002) 209–216
6. Tiede, U., Schiemann, T., Höhne, K.H.: High quality rendering of attributed volume data. In Ebert, D., Hagen, H., Rushmeier, H., eds.: Proc. IEEE Visualization '98, Research Triangle Park, NC (1998) 255–262 (ISBN 0-8186-9176-X).
7. Zilles, C., Salisbury, K.: A constraint-based god object method for haptics display. Proceedings of IEEE/RSJ (1995)
8. Pflesser, B., Petersik, A., Tiede, U., Höhne, K.H., Leuwer, R.: Volume cutting for virtual petrous bone surgery. Comput. Aided Surg. **7** (2002) 74–83
9. Höhne, K.H., Pflesser, B., Pommert, A., Riemer, M., Schiemann, T., Schubert, R., Tiede, U.: A new representation of knowledge concerning human anatomy and function. Nat. Med. **1** (1995) 506–511