

# A multiprocessor decoupled system for the simulation of temporal bone surgery

Marco Agus, Andrea Giachetti, Enrico Gobetti, Gianluigi Zanetti, Antonio Zorcolo

CRS4

VI strada ovest, Macchiareddu I-09010 Uta (CA), Italy

<http://www.crs4.it>

January 14, 2002

**Abstract.** A training system for simulating temporal bone surgery is presented. The system is based on patient-specific volumetric object models derived from 3D CT and MR imaging data. Real-time feedback is provided to the trainees via real-time volume rendering and haptic feedback. The performance constraints dictated by the human perceptual system are met by exploiting parallelism via a decoupled simulation approach on a multi-processor PC platform. In this paper, system components are detailed and the current state of the integrated system is presented.

**Key words** Surgical simulation, temporal bone dissection, decoupled simulation model, hardware volume rendering, haptic feedback

## 1 Introduction

The increasing importance of minimally invasive operational techniques is preparing the ground for a drastic change in pre-operation planning and basic surgeon training. In fact, the spatial limitation of the operational theater, the way the latter is seen by the surgeon, e.g., by microscope, and the absence of direct physical contact between the surgeon and the patient body, make it possible to replace the patient, for training and pre-operation planning purposes, with a computer system able to reproduce the right visual and tactile feedback.

In this paper we describe our first result in the development of a training system for simulating surgery on the temporal bone, a skull region just behind the ear. The specific target of our simulator is mastoidectomy, a very common operative procedure that consists in the removal by burring of the mastoid portion of the temporal bone. The site anatomy is widely variant. The main risks are related to the detection and avoidance of the facial nerve and of aberrant jugular veins (or branches) and to the resection of adequate amounts of the mastoid air cells. The ability to rehearse the procedure using patient specific data is extremely rare. A VR simulator realistically mimicking a patient-specific operating environment addresses this shortcoming. A number of groups are working toward this goal [WBS<sup>+</sup>00, KJM<sup>+</sup>97, JJT<sup>+</sup>ar].

Our system is designed to work on patient-specific volumetric object models directly derived from 3D CT and MRI images, and to provide realistic visual and haptic feedback, including secondary effects such as the obscuring of the operational site due to the accumulation of bone dust and other burring debris. It is expected that the ability of using directly patient specific data as input will help in the accumulation of a large number of training cases.

Moreover, it will open the road towards the use of the simulator for pre-operation planning and rehearsal, making thus possible to plan surgery directly on a model of the individual patient, rather than by referring to a model surgical procedure on a standard anatomy.

The need to provide real time feedback to users, while simulating burring and related secondary effects, imposes stringent constraints on the system. Our solution is based on a volumetric representation of the scene, and it harnesses the locality of the physical system evolution to model the system as a collection of loosely coupled components running in parallel on a multi-processor PC platform. The prototype described here demonstrates that this option is viable and subjective input from selected end users is encouraging.

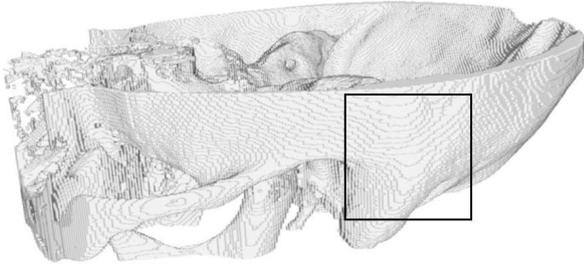
This article focuses on the general system architecture. Reference [AGG<sup>+</sup>02a] provides details on the human factor analysis motivating our design choices, while [AGG<sup>+</sup>02b] provides details on the modeling of bone burring.

The rest of the paper is structured as follows. Section two describes the clinical procedure simulated and lists the features that have been identified as relevant for the clinical simulator. In section three we describe the implied real time constraints and how they can be satisfied with a decoupled simulation model. The following section describes the major system components, while section five reports on the implementation and results. In the final section we comment on our current and future work.

## 2 The clinical procedure

Mastoidectomy is the most superficial and common surgery of the temporal bone. It consists of removal of the air cavities just under the skin behind the ear itself, and it is performed for chronic infection of the mastoid air cells (mastoiditis).

The mastoid air cells are widely variant in their anatomy and the main risks of the procedure are related to the detection and avoidance of the facial nerve, venous sinuses and "dura madre". Figure 1 shows a panoramic view of the mastoid region produced by the surgical simulator viewer.



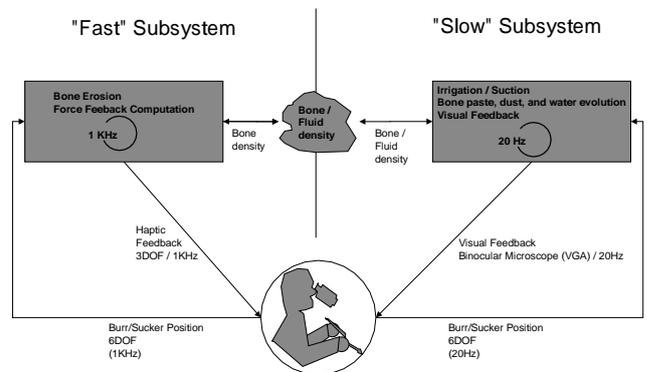
**Fig. 1. Surgical site.** Mastoidectomy is performed in the region indicated by the rectangle. This 800x600 image has been produced in 70 ms by our simulator's volume renderer. The volumetric dataset is 256x256x128 and it is derived from CT images. CT data courtesy of dr. Emanuele Neri, Division of Diagnostic and Interventional Radiology, University of Pisa.

In the typical mastoidectomy surgical setup, see fig. 2, the Ear Nose and Throat surgeon looks at the region interested by the procedure via a stereoscopic microscope and holds in his hands a high speed burr and a sucker, that he uses, respectively, to cut the bone and to remove water (used to cool the burr bit) and bone paste produced by the mixing of bone dust with water, see fig. 4(a). An extensive task analysis has been carried out in order to identify the features relevant to a training simulator.

The analysis involved a review of existing documentation, training aids, and video recordings, interviews with experienced operators, as well as direct observation of the procedure being performed in theater. The analysis indicates that the main processes to simulate are the following: a burr reducing bone in fine dust, an irrigator introducing water that mixes with bone dust, and a sucker removing the mixture, [JTP<sup>+</sup>01, JJT<sup>+</sup>ar, AGG<sup>+</sup>02a]. The capability of replicating the effects caused by the intertwining of the different physical processes is of primary importance for training, see fig. 4. The absence of these effects would reduce the importance placed by a trainee on the need for regular irrigation and suction. Subjective analysis of video records, together with in-situ observations highlighted a correlation between drilling behaviours and type and depth of bone. High-quality force feedback is thus needed for the dominant hand (controlling the burr/irrigator), while only collision detection is required for the non-dominant one (controlling the sucker). As for the nature of the technology required for displaying drill, drill site, bone, and so on, binocular viewing systems are deployed in the operating theatre and used by surgeons, and so binocular imaging should be available to the simulator. However, the wearing of any form of stereoscopic display, such as a head-mounted display or liquid crystal shutter glasses should be avoided. We refer the reader to [AGG<sup>+</sup>02a] for more details on the human factors analysis.

### 3 The decoupled simulation model

The results of the human factors analysis indicate that, to be able to feed the appropriate sensorial inputs to the human perceptual system, the system needs to produce data at two very different time-scales: about 15-20 Hz for the visual rendering, and around 1 KHz for the haptic response [AGG<sup>+</sup>02a]. The computations needed to obtain the haptic force response can be drastically simplified, since response forces can be computed by just considering a small neighborhood around the contact surfaces between surgical instruments and bones. The simulation of secondary effects and the visualization of the evolving operating theater requires, however, a larger computational effort.



**Fig. 3. Logical system decomposition.** The system is divided in a "fast" subsystem, responsible for the high frequency tasks (surgical instrument tracking, force feedback computation, bone erosion), and a "slow" subsystem, essentially dedicated to the production of data for visual feedback.

We have exploited this difference in complexity and frequency requirements by modeling the simulator as a collection of loosely coupled concurrent components. Logically, the system is divided in a "fast" subsystem, responsible for the high frequency tasks (surgical instrument tracking, force feedback computation, bone erosion), and a "slow" one, essentially dedicated to the production of data for visual feedback (see figure 3). The "slow" subsystem is responsible for the global evolution of the water, bone dust and bone paste. These secondary effects can be considered purely visual, since they just contribute to visual clutter without producing important forces to be returned to the user. The algorithms used to control the simulations are local in character and they are structured so that they communicate only via changes in the relevant, local, substance densities. This arrangement leads naturally to a further break-up of the slow subsystem in components, each dedicated to the generation of a specific visual effect, and thus to a parallel implementation on a multiprocessor architecture. Figure 5 outlines the main components of the system, as implemented in our current prototype. The system runs on two multiprocessor machines connected with a 100 Mbit Ethernet link. The data is initially replicated on the two machines. The first machine is dedicated to the high-frequency tasks: haptic device handling and bone removal



**Fig. 2. Virtual vs real mastoidectomy surgical setup.** The surgeon looks at the region interested by the procedure via a stereoscopic microscope and holds in his hands a high speed burr and a sucker. The main photo shows the virtual simulator setup, while the inset photo represents a typical mastoidectomy surgical theatre. Inset photo courtesy of Prof. Bob Stone, Virtual Presence Ltd.

simulation, which run at 1 KHz. The second machine concurrently runs, at about 15-20 Hz, the low-frequency tasks: bone removal, fluid evolution and visual feedback. Since the low-frequency tasks do not influence high-frequency ones, the two machines are synchronized using one-way message passing, with a dead reckoning protocol to reduce communication bandwidth.

A major design decision is the definition of the actual representation of the data. We have chosen to consistently use a voxel-based volumetric approach, where the model is represented by a regular array of material labels with associated density. This representation has a number of advantages: first, since data organization is the same as the one of the acquired data, errors introduced by reformatting and/or surface extraction are avoided; second, local editing and point location operations can be implemented at low cost; finally, an array-based data structure can be shared very efficiently between concurrent processes. This representation, however, brings important challenges: the number of contacts between voxel-based volumetric objects poses a problem for calculating collisions response [GSMF97]; fluid-dynamic computations scale with the cube of volume dimensions; rendering a dynamic volume under real-time constraints is an inherently complex task, since a large number of volume elements may contribute to the final image.

The technical solutions implemented in our prototype are presented in the following section.

## 4 System components

### 4.1 Bone dissection haptic feedback

The simulation of bone burring involves first the detection of collisions of the burr with bone surface, then, depending on the type and location of the contact, a prediction on the amount of bone to be removed and of the forces that should be returned to the hand of the user via the haptic feed-back device. Bone is hard and has a stress-strain relationship similar to many engineering materials.

Hence, as discussed in Fung [Fun93], stress analysis in bone can be made in a way similar to what it is usually done in structural engineering. Given the particular nature of the process simulated, the natural way to model the temporal bone anatomy would be to use a finite element volumetric approach. However, due to temporal constraints imposed by the haptic feed-back loop, this is impractical and we are forced to use a simplified model.

The main source of difficulties is intrinsic in the physics of the burring process. A detailed mechanical description of



(a) Mud (bone past) formation



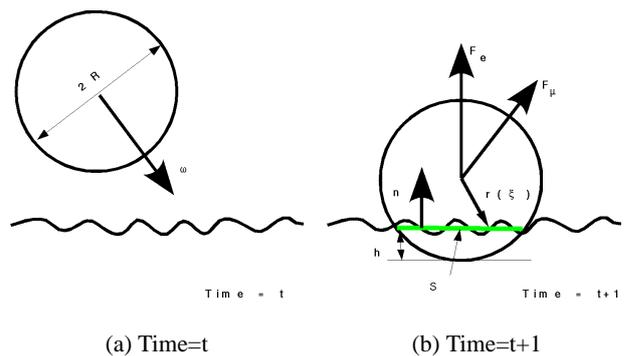
(b) Obscuring effects

**Fig. 4. Operation scene.** These two images are typical examples of what is seen by the surgeon while performing mastoidectomy. In (a) it is clearly visible the paste created by the mixing of bone dust with water. If the paste and the water are not removed, they can obscure the field of view (b). Photos courtesy of Prof. Stefano Sellari Franceschini, Ear Nose and Throat Surgery, Dept. of Neuroscience, University of Pisa.

the cutting of material by a rotating burr is complicated because it involves: the tracking of the continuously changing free surface of the material being cut; the impact of the burr blades on the surface; the resulting stress distribution in the material; and the consequent plastic deformation and break-up. In the general engineering context this problem is solved by using experimentally determined curves, but, for the specific case of bone burring, there are no publically available data. Furthermore, in the specific context of haptic feedback, one cannot apply the standard methods found in the mechanical engineering literature for the simulation of milling. In fact, an haptic feedback system is driven by an open-loop controller that needs to rapidly evaluate a reasonable response force for arbitrary tool penetrations.

To circumvent these complications, we have developed a simplified model specifically designed to describe burr-bone interactions in our haptic feedback system. The model, described below and with more details in [AGG<sup>+</sup>02b], is based on a limited number of parameters that are, at the moment, tuned by trial and error following the opinion of expert surgeons as feedback. It is clear, however, that it would be of extreme interest to have an objective validation of the haptic system by comparing its results with direct forces measurements obtained by drilling actual samples. We are currently starting an activity aimed at defining an experimental setup and measurement procedures.

In our model, the simulation of the burring process is organized so that each time step is divided in two sub-steps. The first estimates the bone material deformation and the resulting elastic forces, given the relative position of the burr with respect to the bone. The second estimates the local rate of cutting of the bone by using a – postulated – energy balance between the mechanical work performed by the burr motor and the energy needed to cut the bone, that we assume to be proportional to the bone mass removed. In figure 6 we represent two successive instants, at time  $t$  and  $t+1$ , of an idealized



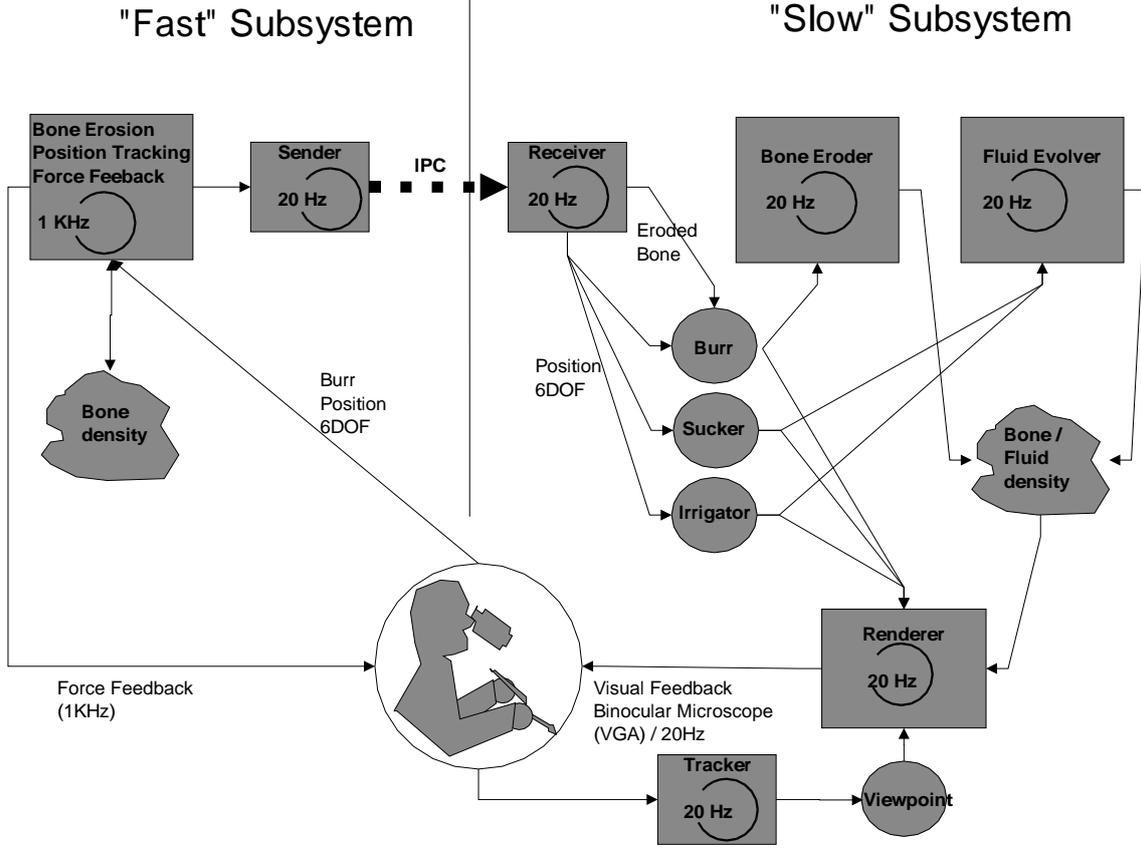
**Fig. 6. The impact of burr on bone.** Here we represent two successive instants, at time  $t$  and  $t+1$ , of an idealized version of a surgeon burr. The burr has a spherical bit, of radius  $R$ , that is rotating with angular velocity  $\omega$ . The surface  $S$  is the effective “contact surface” between the burr and the bone.

version of a surgeon burr. The burr has a spherical bit, of radius  $R$ , that is rotating with angular velocity  $\omega$ . At time step  $t$  the burr is just outside the bone material, while in the next time step it is intersecting the bone surface. In the following we will refer to the surface  $S$  as the “contact surface” between the burr and the bone.

The first two moments of the bone density,  $\rho(\mathbf{r})$ , in the region contained in the burr bit are, respectively,

$$M = \int_{r < R} d^3 \rho(\mathbf{r}), \quad (1)$$

$$\mathbf{M}_1 = \int_{r < R} d^3 \rho(\mathbf{r}) \mathbf{r}. \quad (2)$$



**Fig. 5. Decoupled simulation architecture.** The system uses a volumetric approach, with the initial configuration of the model directly derived from patient CT data. The data is initially replicated on the two machines. The first machine is dedicated to the high-frequency tasks: haptic device handling and bone removal simulation. The second machine concurrently runs at 10-20 Hz the low-frequency tasks: bone removal, fluid evolution and visual feedback. The two machines are synchronized using one-way message passing with a dead reckoning protocol.

The origin for  $\mathbf{r}$  is the center of the sphere. We can now estimate the normal direction,  $\hat{\mathbf{n}}$ , to  $S$ , as  $-\mathbf{M}_1/|\mathbf{M}_1|$  and the thickness  $h$  by solving

$$M = \rho_0 \pi h^2 \left(R - \frac{h}{3}\right), \quad (3)$$

where  $\rho_0$  is a bone density scale. Using Hertz's contact theory, [LL86], we can derive an expression for the total force,  $\mathbf{F}_e$ , exerted on the burr by the elastic deformation of the bone:

$$\mathbf{F}_e = C_1 R^{\frac{1}{2}} h^{\frac{3}{2}} \hat{\mathbf{n}}, \quad (4)$$

where  $C_1$  is a dimensional constant that describes the elastic properties of the material. Moreover, we can give an expression for the pressure,  $\mathbf{P}(\boldsymbol{\xi})$ , exerted by the drill on the point  $\boldsymbol{\xi}$ , where  $\boldsymbol{\xi}$  is measured from the center of  $S$ , see fig. 6(b):

$$\mathbf{P}(\boldsymbol{\xi}) = -\frac{3}{2\pi a^2} \sqrt{1 - \frac{\boldsymbol{\xi}^2}{a^2}} \mathbf{F}_e, \quad (5)$$

where  $a$  is the radius of the contact region estimated as

$$a = (C_1 R)^{\frac{1}{3}} F_e^{\frac{1}{3}}. \quad (6)$$

Strictly speaking, Hertz's theory is only valid for small elastic deformations, a condition that, in the case of the simulator, will be often violated, e.g., in the case of large time

steps (and/or a too energetic user). On the other hand, the objective here is to derive a computationally tractable, robust, expression for the response forces that, at least in some limit, is physically reasonable.

From equation 5, we can estimate the frictional forces,  $\mathbf{F}_\mu$ , that the bone will oppose to the burr rotation:

$$\mathbf{F}_\mu = \mu \int_{\xi < a} d\sigma P(\boldsymbol{\xi}) \frac{\mathbf{r}(\boldsymbol{\xi}) \times \boldsymbol{\omega}}{|\mathbf{r}(\boldsymbol{\xi})| |\boldsymbol{\omega}|}, \quad (7)$$

where  $\mu$  is a dimensional friction coefficient. Therefore, the total force that should be returned by the haptic feedback device is  $\mathbf{F}_T = \mathbf{F}_e + \mathbf{F}_\mu$ .

We model the cutting of the burr by assuming that all the power spent by working against the frictional forces on a "contact surface" element  $d\sigma$  goes toward the erosion of the bone material on the surface. In other words, we equate

$$\mu P(\boldsymbol{\xi}) \omega r(\boldsymbol{\xi}) \left(1 - \left(\frac{\mathbf{r}(\boldsymbol{\xi}) \cdot \boldsymbol{\omega}}{|\mathbf{r}(\boldsymbol{\xi})| |\boldsymbol{\omega}|}\right)^2\right) d\sigma = \alpha \phi(\boldsymbol{\xi}) d\sigma, \quad (8)$$

where  $\alpha$  is a dimensional constant and  $\phi(\boldsymbol{\xi})$  is the mass flux at the contact surface point  $\boldsymbol{\xi}$ . Using the mass flux  $\phi$  one can update the position of the bone surface.

In principle, in the balance above one should consider also the contribution of the work done by the user by moving the

burr against the elastic response force. On the other hand, due to the high angular velocity of the burr – more than 50,000 rpm – this can be neglected.

In our system we use a discretized version of the model described above, see [AGG<sup>+</sup>02b] for details.

#### 4.2 Bone dust, water and mud

Figures 4(a,b) are typical examples of what is seen by the surgeon while performing mastoidectomy. As was mentioned before, the burring of the bone produces dust that mixes with water in a paste, “mud”, clearly visible in figure 4(a). The paste material has a quite complex behavior, from sand-like to gel-like. The water paste mixture needs to be continuously removed, otherwise it can obscure the field of view as it is seen in figure 4(b). Although the presence of the water/paste mixture is essentially irrelevant with respect to the interaction between the burr and the bone, its presence cannot be neglected in the creation of the visual feed-back, because its “obscuring” effects constitute the principal cue to the user for the use of the sucker device.

A direct, “physically correct”, simulation of the dust-water system would require, to be able to capture all the dynamically relevant length scales, a very fine spatial resolution and it would be computationally incompatible with the real-time requirements of the simulation. Therefore, we are modeling the dust/fluid dynamics using what essentially amounts to an hybrid particles/sand pile model [RS99,LM93].

In our volumetric description of the scene, voxel labelled as bone must give reaction to the manipulators through the haptic feedback devices, but they do not evolve unless they are removed by burring. Dust and water are both modeled by assigning density labels to voxel. Their interaction is controlled by local cellular automata rules, that update the voxel values by taking into account local densities of bone, dust and water in a small neighborhood of the voxel. The water coming from the irrigator or being splashed away from the bone surface is modeled through particle objects with well defined masses and velocities. The interaction between particles and voxels is a critical part of this approach. In the current implementation, it depends on the particle velocity: “Slow” particles have a high probability of either mixing with dust or to stay close to the bone surface because of adhesive forces; while “fast” particles could be scattered away. To be able to accommodate the needs of real-time rendering we give higher priority to the update of voxels and particles that are closer to the burr bit. Figure 7 shows the beginning of a typical bone cutting sequence performed in the mastoid region. The tool on the left is the sucker. It interacts with the scene by simply removing all the dust and water particles within a certain radius from its tip.

#### 4.3 Real-time visual rendering

The surgical simulator must achieve the visual illusion of animation and responsiveness by rapid successive presentation of a sequence of static images of the evolving operating theater as seen from the surgical microscope. Since humans are

very sensitive to synchronization problems between synthesized and real-world sensory input, it is of primary importance for the visual rendering subsystem to operate within the timing constraints imposed by the human perceptual system (i.e. latency of less than 300 ms, and frequency above 10-15 Hz [MZ92,HD91,YJN<sup>+</sup>95]).

We reach this goal using a parallel processing approach, which exploits the capabilities of current graphics PC architectures. In our system, the renderer is totally decoupled from the simulator and the tracking system, and runs at his own frequency. At each rendered frame, the following actions are taken:

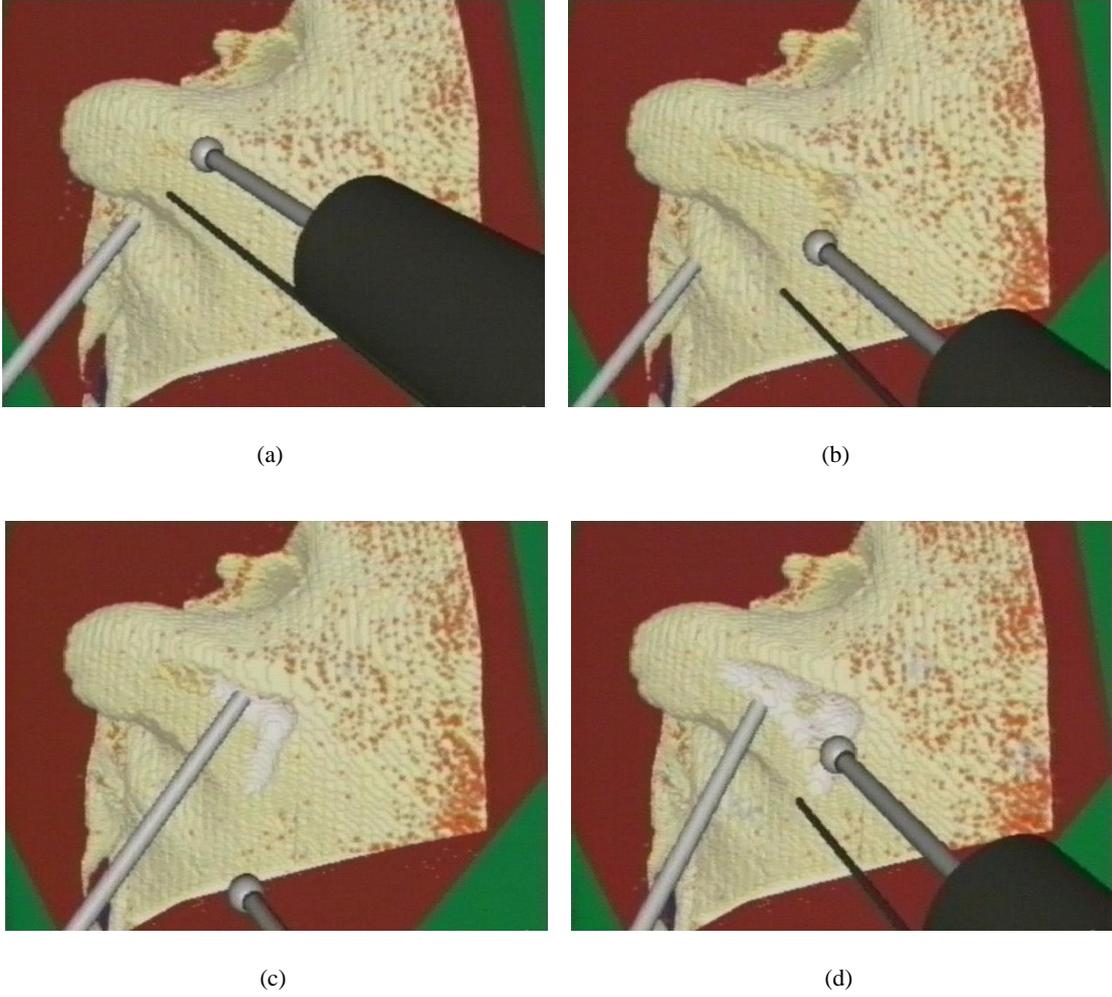
1. the time of presentation of the frame is predicted;
2. the Z and color buffer are cleared;
3. the position/orientation of the surgical microscope at the end of the frame is extrapolated from the latest sensor data; the camera view/projection matrices are set accordingly;
4. the position/orientation of the surgical instruments at the end of the frame are extrapolated from the latest sensor data; a polygonal representation of the surgical instruments is rendered to the Z and color buffer;
5. the simulation state is presented by projecting and compositing onto the image the elements of the volumetric data representation, which is shared with the simulator;
6. the image is presented;

This technique relies on the ability to rapidly render a good quality view of a continuously changing scalar volume. Our algorithm, based on texture mapping and back-to-front composition of volume slices, maximizes parallel efficiency by asynchronously performing volume rendering while the simulator is updating the volume.

##### 4.3.1 Shaded direct volume rendering of dynamic volumes

In direct volume rendering, images are produced by integrating along selected projectors the value of a continuous emission/reflection/absorption volume function reconstructed from discrete sampling points [Max95]. By manipulating the mapping from values of the original volume data to emission, reflection, and absorption coefficients, various effects can be achieved, including isosurfaces and opaque objects. In our case, the volume is a regular 3D grid containing at each voxel a material identifier (e.g. air, bone, dust, water, blood). The latter is continuously reassigned by the simulation, that is running in parallel to the rendering process. Rendering such a dynamic volume under real-time constraints is particularly challenging.

A number of authors have proposed to exploit texture mapping and rasterization hardware to render scalar volumes at interactive speeds [CN94,CCF94,GL94,VK96,Kul96]. These techniques are based on uploading the scalar volume to texture memory prior to rendering object-aligned or view-direction-aligned textured volume slices. One of the major limitations of these methods is their inability to efficiently implement surface illumination models, since texture lookup is based only on data values and not on gradient information. Various authors have proposed alternative techniques for supporting hardware-accelerated direct volume rendering with shading [VK96,WE98,RSEB<sup>+</sup>00,EKE01]. However, this comes



**Fig. 7. A burring sequence.** Here we show the beginning of a typical bone cutting sequence performed in the mastoid region, see figure 1. The burr is initially above the bone (a); the user starts cutting (b,c), note the accumulation of bone dust (marked in yellow); the user irrigates the cut region and removes the bone paste using the sucker (d).

at the expense of performance and texture memory overheads, since the proposed techniques require multiple passes through the rasterization hardware and/or precomputation of gradient volumes. This is unacceptable in our case, since the volume is continuously varying, and thus we cannot compute and reload gradient maps.

In our approach, a fast approximation of the shading equation is computed on the fly by the graphics pipe-line directly from the scalar data. We do this by exploiting the possibilities offered by multi-texturing with the register combiner OpenGL extension, that provides a configurable mean to determine per-pixel fragment coloring [Kil00]. The extension is available on commodity graphics boards (e.g., NVIDIA GeForce series).

To simulate shading effects from contour surfaces at sharp changes in a scalar volume function, a common approach [Max95] is to use the opacity gradient to measure surface “strength”, and to shade the volume using a simple Lambert

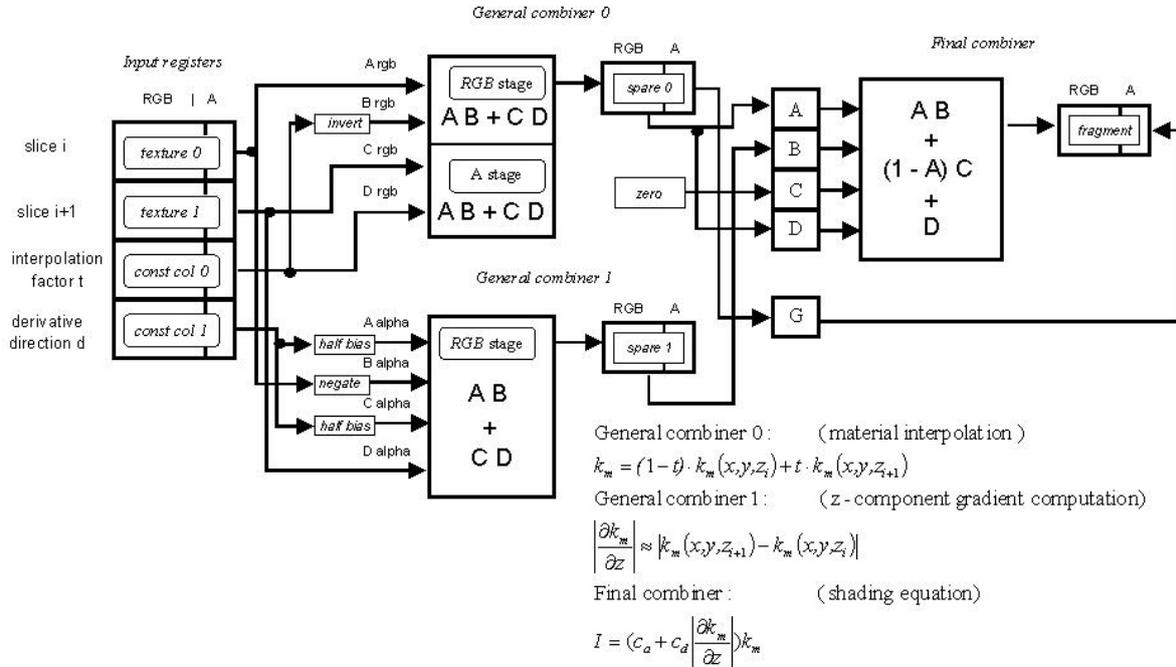
diffuse shading formula multiplied by the strength, giving, for a single directional light:

$$I(x, y, z) = (c_a + c_d |\nabla k_m^\alpha(x, y, z) \cdot \mathbf{l}|) \cdot k_m(x, y, z) \quad (9)$$

where  $c_a$  and  $c_d$  are the ambient and diffuse RGBA intensities of the light,  $k_m$  is the material RGBA color, and  $\mathbf{l}$  is the direction of the light. If we assume that the light direction is coincident with the volume coordinate axis which is pointing towards the viewer (e.g., the local Z axis), we need to compute only a single component of the gradient (in the example, the Z component). This approximation is acceptable in our case, because of the particular microscope setup which limits the viewer to almost frontal views [JTP<sup>+</sup>01, AGG<sup>+</sup>02a]. The shading formula becomes, using a forward difference approximation of the gradient:

$$\frac{\Delta k_m^\alpha}{\Delta z} = \left| \frac{k_m^\alpha(x, y, z + \Delta z) - k_m^\alpha(x, y, z)}{\Delta z} \right| \quad (10)$$

$$I(x, y, z) = (c_a + c_d \frac{\Delta k_m^\alpha}{\Delta z}) \cdot k_m(x, y, z) \quad (11)$$



**Fig. 8. OpenGL combiner setup.** The final combiner blends the interpolated slice value from the first combiner with the shading value from the second combiner.

This equation can be implemented in the graphics hardware by programming the register combiners (see figure 8), leading to an efficient shaded volume rendering algorithm in which all computation is performed by the graphics hardware starting simply from the scalar volume.

At the beginning of the procedure, the material table, which maps the material identifiers in the volume to the RGBA colors  $k_m$ , is loaded in the shared texture palette. The register combiners are then configured as in figure 8 to implement slice interpolation and fragment shading. The volume is then traversed back-to-front, and the 2D slices are sequentially loaded into texture memory, alternating between texture 0 and texture 1. For each pair of slices, a number of intermediate slices are synthesized by rendering planar polygons and storing the interpolation factor in one of the constant color registers. For each fragment, general combiner 0 generates the color of the intermediate slice by interpolating between the front and back slice using the given interpolation factor, general combiner 2 computes the opacity gradient, and the final combiner computes the fragment's final RGBA color as in equation 10.

This procedure is extremely efficient, since all the computation is performed in parallel in the graphics hardware and no particular synchronization is needed between the renderer and the process that is modifying the dataset. Only a single sweep through the volume is needed, and volume slices are

sequentially loaded into texture memory on current standard PC graphics platform using AGP 4X transfers, which provide a peak bandwidth of 1054 MB/s. A 256x256x256 dynamic volume using 8 bit material identifiers may thus potentially be transferred to texture memory at over 60 fps. Only two slices need to be present in texture memory at the same time.

#### 4.3.2 Reducing fill-rate bottleneck

Pixel fill-rate is the major limiting factor when using a texturing approach to volume rendering. In zoom rendering, an appropriately down-scaled image is rendered in the back buffer and then enlarged and copied to the front buffer [MSG95]. This way, delays associated with buffer swap synchronization are avoided, and the number of pixels filled during volume rendering is reduced. In our implementation, the copy and zoom operations are implemented by copying the reduced size image in texture memory and then rendering a textured polygon in the front buffer. This way, sophisticated texture interpolation algorithms can be used to reduce the artifacts caused by magnification. Zoom rendering is particularly useful in our application, because the pixel resolution is much larger than the resolution of the data that is displayed in the window.

## 5 Implementation and results

A prototype system, based on the techniques discussed above, is running on a dual PC platform. Our current configuration is the following:

- a single-processor PIII/600 MHz with 256 MB PC100 RAM for the high-frequency tasks; two threads run in parallel: one for the haptic loop (1KHz), and one for sending volume and instrument position updates to the other machine;
- a dual-processor PIII/600 MHz with 512 MB PC800 RAM and a NVIDIA GeForce 2 GTS running a Linux 2.4 kernel for the low frequency tasks; three threads are continuously running on this machine: one to receive volume and position updates, one to simulate bone removal and fluid evolution, and one for visual rendering;
- a Phantom 1.0 haptic device for the dominant hand; the device is connected to the single processor PC. It provides 6DOF tracking and 3DOF force feedback for the burr/irrigator;
- a Phantom Desktop haptic device for the non-dominant hand; the device is connected to the single processor PC. It provides 6DOF tracking and 3DOF force feedback for the sucker;
- an n-vision VB30 binocular display for presenting images to the user; the binoculars are connected to the S-VGA output of the dual processor PC.

The performance of the prototype is sufficient to meet timing constraints for display and force-feedback, even though the computational and visualization platform is made only of affordable and widely accessible components. The volumetric datasets used to represent the region where the operation takes place incorporate information on bone and on the noble structures that should be avoided while performing the simulated operation. To obtain these datasets, we need to combine information from several modalities that contain complementary data, specifically, computed tomography (CT) provides high spatial resolution bone images whilst magnetic resonance imaging (MRI) provides images of soft tissues. In the simulations reported in this paper we are using a volume of  $256 \times 256 \times 128$  cubical voxels (0.3 mm side) obtained by manually adding soft-tissue information to an high resolution CT scan. We are in the process of adapting the system to the direct use of dataset obtained by applying probability maps methods [PTSJ01] for automatic multi-dimensional medical image segmentation.

The force-feedback loop is running at 1 KHz using a  $5 \times 5 \times 5$  grid around the tip of the instruments for force computations. Shaded volume rendering of dynamic volumes currently takes 70 ms per frame (i.e. over 14 frames per second) using 256 depth slices on an  $800 \times 600$  window with 16 bit color and 2X zoom rendering.

We have gathered initial feedback on the prototype system from Ear Nose and Throat surgeons that are collaborating to this project. The overall realism of the simulation is considered sufficient for training purposes. Infact, as required by the task analysis [AGG<sup>+</sup>02a], the haptic subsystem is able to provide a reasonable force feedback effect, bone removal and noise simulation. The visual system is able to provide bone dust and water bleeding effects, blood simulation, and

manipulators displayed at the required frame rate. Figure 7 shows selected frames of the initial phase of a virtual mastoidectomy. Debris formation and suction effects are clearly visible. See [ABG<sup>+</sup>01] for a video demonstration. The system is currently tested at the CRS4 site. Within next year, the prototype will be installed at the Medical School of the University of Pisa. There the virtual training environment will be tested as a tool for training, evaluation and human performance assessment. An effort is currently on-going to specify the scope of tasks expected of trainees and the relative measures. Current plans include overall time, reaction time to contingencies, multiple errors whilst marking out cut area, contact with key anatomical features (e.g., facial nerve, jugular bulb, sigmoid sinus, brain tissue), inadequate drill/burr selection, over-pressure whilst using the PHANToM and so on.

## 6 Conclusion and Future Work

We have presented the design and implementation of a training system for simulating temporal bone surgery. The system is based on patient-specific volumetric object models derived from 3D CT and MR imaging data. Real-time feedback is provided to the trainees via real-time volume rendering and haptic feedback.

The performance constraints dictated by the human perceptual system are met by exploiting parallelism with a decoupled simulation approach on a multi-processor PC platform.

The simulation, currently, does not consider the possibility of large bone blocks that become completely separated from the skull and fall, or fly, away. This is a non-local effect that require a global model for the bone. As a first approximation, we plan simply to check periodically if there are isolated bone regions (i.e. not connected with bone voxel at the extrema of the dataset) and to impose some dynamics to them. This will have to be reflected, however, in an update of the bone model handled by the bone erosion subsystem. We are also working on the speed-up of the water flow simulation.

Finally we have started an activity aimed at defining an experimental setup and measurement procedures for a complete validation of the haptic simulation.

## 7 Acknowledgments

These results were obtained within the framework of the European Union IERAPSI project (EU-IST-1999-12175).

## References

- [ABG<sup>+</sup>01] Marco Agus, Fabio Bettio, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Real-time haptic and visual simulation of bone dissection. Video demonstration, <http://www.crs4.it/ierapsi>, 2001.
- [AGG<sup>+</sup>02a] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, Nigel W. John, and Robert J. Stone. Mastoidectomy simulation with combined visual and haptic feedback. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, and D. Stredney, editors, *Medicine Meets Virtual Reality 2002*, January 2002.

- [AGG<sup>+</sup>02b] Marco Agus, Andrea Giachetti, Enrico Gobetti, Gianluigi Zanetti, and Antonio Zorcolo. Real-time haptic and visual simulation of bone dissection. In *IEEE Virtual Reality Conference, Conference*, Orlando, FL, USA, March 2002. IEEE Computer Society Press.
- [CCF94] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In Arie Kaufman and Wolfgang Krueger, editors, *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, October 1994. ISBN 0-89791-741-3.
- [CN94] Timothy J. Cullip and Ulrich Neumann. Accelerating volume reconstruction with 3D texture hardware. Technical Report TR93-027, Department of Computer Science, University of North Carolina - Chapel Hill, May 1 1994.
- [EKE01] K. Engel, M. Kraus, and T. Ertl. High quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *EuroGraphics/SIGGRAPH Workshop on Graphics Hardware*, 2001.
- [Fun93] Y.C. Fung. *Biomechanics: Mechanical Properties of Living Tissues*. Springer, 1993.
- [GL94] S. Guan and R. G. Lipes. Innovative volume rendering using 3D texture mapping. In *Image Capture, Formatting and Display*, volume 2164 of *SPIE*. 1994.
- [GSMF97] S. Gibson, J. Samosky, A. Mor, and C. Fyock. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. *Lecture Notes in Computer Science*, 1205:369–374, 1997.
- [HD91] R. Held and N. Durlach. Telepresence, time delay, and adaptation. In Stephen R. Ellis, editor, *Pictorial Communication in Real and Virtual Environments*. Taylor and Francis, 1991.
- [JJT<sup>+</sup>ar] A. Jackson, N. W. John, N. A. Thacker, E. Gobetti, G. Zanetti, R. J. Stone, A. D. Linney, G. H. Alusi, and A. Schwerdtner. Developing a virtual reality environment for petrous bone surgery: a “state-of-the-art” review. *Journal of Otology & Neurology*, To appear.
- [JTP<sup>+</sup>01] N. W. John, N. Thacker, M. Pokric, A. Jackson, G. Zanetti, E. Gobetti, A. Giachetti, R. J. Stone, J. Campos, A. Emmen, A. Schwerdtner, E. Neri, S. Sellari Franceschini, and F. Rubio. An integrated simulator for surgery of the petrous bone. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 2001*, pages 218–224, Amsterdam, The Netherlands, January 2001. IOS Press.
- [Kil00] Mark J. Kilgard. A practical and robust bump-mapping technique for today's gpus. Technical report, NVIDIA Corporation, July 2000.
- [KJM<sup>+</sup>97] RB Kuppersmith, R Johnston, D Moreau, RB Loftin, and H Jenkins. Building a virtual reality temporal bone dissection simulator. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 5*, pages 180–186, Amsterdam, The Netherlands, January 1997. IOS Press.
- [Kul96] Todd Kulick. Building an opengl volume renderer. SGI Dev. News, 1996.
- [LL86] L. Landau and E. Lifshitz. *Theory of elasticity*. Pergamon Press, 1986.
- [LM93] X. Li and J.M. Moshell. Modeling soil: Realtime dynamic models for soil slippage and manipulation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 361–368, 1993.
- [Max95] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.
- [MSG95] Tomasz Mazuryk, Dieter Schmalstieg, and Michael Gervautz. Zoom rendering: Improving 3-D rendering performance with 2-D operations. Technical Report CG, Institute of Computer Graphics, Vienna University of Technology, 1995.
- [MZ92] M. McKenna and D. Zeltzer. Three dimensional visual display systems for virtual environments. *Presence*, 1(4):421–458, 1992.
- [PTSJ01] Maja Pokric, Neil Thacker, Marietta L.J. Scott, and Alan Jackson. The importance of partial voluming in multi-dimensional medical image segmentation. *Lecture Notes in Computer Science*, 2208:1293–1294, 2001.
- [RS99] J.K. Hodgins R.W. Sumner, J.F. O'Brien. Animating sand, mud and snow. *Computer Graphics Forum*, 18:1, 1999.
- [RSEB<sup>+</sup>00] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume rendering on standard PC graphics hardware using multi-textures and multi-stage rasterization. In Stephan N. Spencer, editor, *Proceedings of the 2000 SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, pages 109–118, N. Y., August 21–22 2000. ACM Press.
- [VK96] Allen Van Gelder and Kwansik Kim. Direct volume rendering with shading via three-dimensional textures. In *1996 Volume Visualization Symposium*, pages 23–30. IEEE, October 1996. ISBN 0-89791-741-3.
- [WBS<sup>+</sup>00] G. Wiet, J. Bryan, D. Sessanna, D. Streadney, P. Schmalbrock, and B. Welling. Virtual temporal bone dissection simulation. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 2000*, pages 378–384, Amsterdam, The Netherlands, January 2000. IOS Press.
- [WE98] Rüdiger Westermann and Thomas Ertl. Efficiently using graphics hardware in volume rendering applications. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 169–178. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.
- [YJN<sup>+</sup>95] Christine Youngblut, Rob E. Johnson, Sarah H. Nash, Ruth A. Wienclaw, and Craig A. Will. Review of virtual environment interface technology. IDA Paper P-3186, Institute for Defense Analyses, March 1995.