

Adaptive techniques for real-time haptic and visual simulation of bone dissection

Marco Agus

Andrea Giachetti

Enrico Gobbetti

Gianluigi Zanetti

Antonio Zorcolo

CRS4

VI Strada Ovest, Z. I. Macchiareddu, I-09010 Uta (CA), Italy
{magus,giach,gobbetti,zag,zarco}@crs4.it – <http://www.crs4.it>

Abstract

Bone dissection is an important component of many surgical procedures. In this paper, we discuss adaptive techniques for providing real-time haptic and visual feedback during a virtual bone dissection simulation. The simulator is being developed as a component of a training system for temporal bone surgery. We harness the difference in complexity and frequency requirements of the visual and haptic simulations by modeling the system as a collection of loosely coupled concurrent components. The haptic component exploits a multi-resolution representation of the first two moments of the bone characteristic function to rapidly compute contact forces and determine bone erosion. The visual component uses a time-critical particle system evolution method to simulate secondary visual effects, such as bone debris accumulation, bleeding, irrigation, and suction.

1 Introduction

Bone dissection plays an important role in a lot of common surgical procedures. It consists in removing part of a patient bone with a cutting burr, in order to clean out bone cells infections, e.g., for cholesteatoma treatment, or to reach otherwise inaccessible surgical sites, e.g., for inner ear and skull base surgery. Successful execution of bone dissection requires a high level of dexterity, experience and knowledge of the patient anatomy. Human cadaver dissections are currently considered the primary teaching tool [18]. However, the physical limitations and decreased availability of the material, its high handling and disposal cost, as well as the risks associated to transmission of diseases, make this training method increasingly problematic. The ability to rehearse such procedures is thus becoming more and more rare. A VR simulator realistically mimicking a patient-specific operating environment would

therefore significantly contribute to the improvement of surgical training.

Accurate and fast burr–bone interaction simulation is a key enabling technology in the development of such a simulator. It has to include burr–bone contact detection, bone erosion, generation of haptic response, and synthesis of secondary visual effects, such as bone debris accumulation, bleeding, irrigation, and suction [1]. The human perceptual requirements of a simulator impose very stringent constraints on performance, making bone dissection simulation a technological challenging task.

In this paper, we discuss adaptive techniques that trade simulation quality with speed in order to meet real-time constraints. The simulator in which those techniques are incorporated is being developed as a component of a training system for temporal bone surgery [2]. We harness the difference in complexity and frequency requirements of the visual and haptic simulations by modeling the system as a collection of loosely coupled concurrent components. The haptic component exploits a multi-resolution representation of the first two moments of the bone density to rapidly compute contact forces and determine bone erosion. The visual component uses a time-critical particle system evolution method to simulate secondary visual effects, such as bone debris accumulation, bleeding, irrigation, and suction (see figure 1).

The rest of the paper is organized as follows. An overview of the related work is presented in section 2. Our techniques for real-time simulation of burr–bone interaction are introduced in section 3, while secondary effects simulation techniques are presented in section 4. Section 5 explains how our methods are integrated in a running prototype training system for mastoidectomy. Section 6 discusses the preliminary results obtained. The paper concludes with a summary of the technique and a view of current and future work.

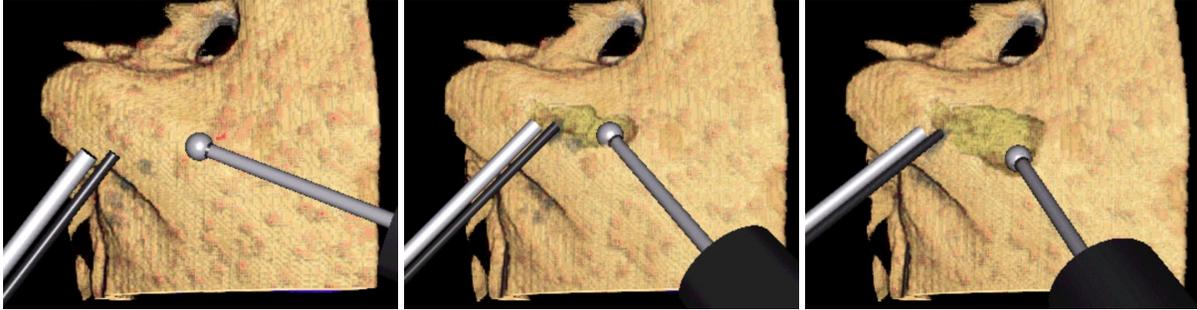


Figure 1. A virtual burring sequence performed in the mastoid region. Using our adaptive techniques, haptics simulation and bone removal run at 1KHz, while visual simulation, that includes bleeding, debris accumulation, and suction, runs at 20Hz.

2 Related work

2.1 Bone dissection training aids

The limitations of human cadaver dissection for teaching has led to the development of alternative training aids. Widely used are plastic models, such as Pettigrew Plastic Temporal Bone series[19]. Using Pettigrew’s models the complete temporal bone, for example, can be fully dissected using standard theater equipment. While such synthetic models are currently a valid alternative to cadaveric exercises, simulators allow increased exposure to pathological variance through patient specific and synthesized models, and support a more quantitative assessment of trainee proficiency [14]. A number of groups are thus developing simulators for bone dissection. Early systems (e.g. [11]) focused on increasing the understanding of the anatomy by providing specialized visualization tools of static models. The VrTool [14] and the VOXEL-MAN system [20] mainly concentrate on accurate visual presentation of free-form volume-sculpting operations. The Ohio Virtual Temporal Bone Dissection simulator [24, 6, 22], similarly to our work, aims instead at realistically mimicking the visual and haptics effects of a real operation. Our work is characterized by the physically based contact model [3], the visual simulation of bone dust, irrigation, and bleeding, as well as the use of multi-resolution and time-critical techniques to meet performance constraints.

2.2 Burr-bone interaction

Burr-bone interaction simulation builds on techniques developed for interaction with volumetric models. Early systems for interactively adding and removing material from a voxelized scalar representation were developed by Galyan and Hughes [9] and Wang and Kaufman [23]. These systems used a voxelized representation of both the editing

tool and the data. Material removal was modeled as a simple Boolean operation, and there was no attempt at replicating the physical effect of tool/material contact. Avila and Sobjaraiki [4] extended the approach by adding repulsion forces, and used a force-feedback system to control a carving tool. Since contact detection and removal requires traversal of all voxels touched by the tool, the performance constraints of haptic feedback limited the system to model only small tools. Our previous work refined these techniques, specializing them to burr/bone interaction [3]. We introduced a physically based contact and erosion model loosely based on Hertz contact theory. The actual bone erosion is implemented by decreasing the density of the voxels that are in contact with the burr in a manner that is consistent with the predicted local mass flows. The method complexity scales, however, with the cube of the burr tip radius, imposing important limitations on the surgical tool size.

A number of authors have presented techniques based on probing a voxelized environment with surface point samples [10, 13, 17, 12]. These techniques easily support free-form tool shapes, and, since they do not sample the interior of the tool tip, they are faster than their volumetric counterparts. They are however prone to severe aliasing artifacts, and are essentially limited to a one-voxel deep penetration[7].

A few multi-resolution approaches have been proposed to speed-up volume sculpting operations, while avoiding the problems of surface point sampling approaches. Baerentzen [5] proposed an octree-based volume sculpting system, that stores at each node local volume density, for quickly skipping empty regions, as in classic volume rendering applications. More recently, Frisken et al. [8] proposed a resolution adaptive approach based on multi-resolution distance fields. The basic idea is to store at each node the Euclidean distance from the closest surface instead of the local density. This representation considerably speeds up collision detection and response, but requires a costly distance propagation step after model modification.

As in [5], we also use an octree-based representation of

the data, which is continuously kept up-to-date during simulation. Instead of storing local density at each node, we store the first two moments of the bone characteristic function, which provide us information on the local mass distribution, without imposing the model update overhead of a distance based approach. This information is exploited for multi-resolution force computation.

2.3 Secondary effects

During bone dissection, the surgeon holds in his hands a high speed burr and a suction device, that she uses, respectively, to dissect the bone and to remove the bone paste produced by the mixing of bone dust with the water used to wash the burring region and to cool the burr bit. The capability of replicating the effects caused by the intertwining of the different physical processes is of primary importance for training. The absence of these effects would reduce, for instance, the importance placed by a trainee on the need for regular irrigation and suction [1].

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. For this reasons, secondary effects were mostly neglected in prior bone burring simulations. The Ohio Virtual Temporal Bone Dissection simulator simply removes voxels by making them transparent [24, 6]. Localized bleeding is simulated by coloring in red the voxels close to the burr bit. The IERAPSI system [3] exploits the difference in frequency requirements of the visual and haptic simulations by running a rule-based particle system simulator in parallel with the bone dissection simulator. The method is able to provide a crude visual approximation of bone debris accumulation, bleeding, irrigation, and suction. In this paper, we improve the technique by introducing a time-critical particle evolution method that trades simulation quality with time.

3 Burr-bone interaction and haptic feedback

A detailed mechanical description of 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 these problems are solved by using experimentally determined curves, but, for the specific case of bone burring, there are no publicly 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, originally described in [3], 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. The basic assumption underlying our model is that the burr bit is moving relatively slowly with respect to the time scale of the haptic feedback loop and that one can estimate the elastic forces exerted by the bone by geometrically characterizing the region of bone intersected by an idealized sphere representing the burr tip.

Specifically, we model the burr bit, B , with a sphere of radius R centered at \mathbf{R}_b , and consider the first two moments of the bone mass density, $\rho(\mathbf{r})$, contained in B .

$$m_0 = \int_{r < R} d\mathbf{r}^3 \rho(\mathbf{r}), \mathbf{m}_1 = \int_{r < R} d\mathbf{r}^3 \rho(\mathbf{r}) \mathbf{r}. \quad (1)$$

The direction of the local normal, $\hat{\mathbf{n}}$, to the bone surface can then be estimated as $\hat{\mathbf{n}} = -\mathbf{m}_1/|m_1|$, and from the amount of mass contained in B , m_0 , we can derive an effective “penetration depth” h as the smallest positive solution of

$$m_0 = \pi \rho_0 R^3 \left(\frac{h}{R}\right)^2 \left(1 - \frac{h}{3R}\right) \quad (2)$$

where ρ_0 is the “solid” bone reference density.

We can now write an expression for an effective force \mathbf{F}_e , that is supposed to model the elastic response of the bone to the impinging burr.

$$\mathbf{F}_e = c_e R^2 (h/R)^{3/2} \hat{\mathbf{n}}, \quad (3)$$

where c_e is a dimensional constant, that, as far as this model is concerned, describes the elastic properties of the material. In the limit of $h/R \ll 1$, eq. (3) is consistent with Hertz’s contact theory [15].

Typical burr radii are between 1 mm and 5 mm, while the typical speed at which the burr bit is moved is < 100 mm/s [1]. Given that the haptic device acquisition period is 1 ms, the burr bit will typically move a distance of the order of a few percents of its radius. Therefore, it is reasonable to compute interaction forces by checking collisions after the fact, rather than trying to predict them in advance.

For a given response force F_e^* one can invert eq. (3) to obtain, assuming that $h/R \ll 1$,

$$m_0^* \approx \pi \rho_0 R^{1/3} (F_e^*/c_e)^{4/3}. \quad (4)$$

Hence, the amount of computational work needed to resolve, say, a zero force threshold increases only slowly with the burr radius. On the other hand, in typical burr usage one

applies a force on the burr so that it will have an instantaneous erosion surface that scales as R^2 . Since the contact surface, S , of the burr with the bone, again for small h , is proportional to hR , this corresponds to a mode of operation where the force applied by the user on the burr is adjusted to maintain h roughly proportional to R , $h \approx \alpha R$. With this assumption,

$$\frac{dF_e}{d m_0} = \frac{3}{4} \frac{c_e}{\pi \rho_0} \frac{1}{R} \left(\frac{h}{R}\right)^{-1/2} \frac{1}{1 - \frac{h}{2R}}, \quad (5)$$

and

$$F_e = c_e R^2 \alpha^3 / 2, \quad (6)$$

therefore

$$\frac{1}{F_e} \frac{dF_e}{d m_0} \approx \frac{3}{4} \frac{1}{\pi \rho_0} \frac{1}{R^3 \alpha^2}. \quad (7)$$

For a given accepted error ratio β in the haptic force, $\beta = \Delta F_e / F_e$, we can estimate the accepted error for m_0 , Δm_0 , to be

$$\Delta m_0 \approx \beta \alpha^2 R^3. \quad (8)$$

Therefore, in this mode of operation, we can maintain the relative error in force estimation constant at a small computational cost even for increasing burr radius R . In fact, we are allowed to increase linearly with R the discretization scale, ℓ , used in computing the integrals in eq 1.

In the following, we will describe a computational method that exploits these observations to compute \mathbf{F}_e with a computational cost that grows slowly with R and is well within the time constraints, 1 msec total for force estimate and bone erosion, imposed by the haptic feedback device.

The new method completely overcomes the limitations to small burr sizes of the technique used in [3].

3.1 Multi-scale spatial description

The integrals requested by eq. 1 can be easily computed using a multi-resolution volumetric description of the region of interest.

We partition the volume of interest using an octree, with the leaves of the octree that directly refer to the scene voxels, and the coarsest level to the whole scene. In an initialization phase, starting from the leaves, we precompute, for each octree block, I , the local values of m_0^I and \mathbf{m}_1^I . The zeroth moment of the mass contained in block I is simply the sum of its values at the block children $\{I, k\}$, $m_0^{\{I, k\}}$. To compute \vec{m}_1^I we use the center of mass decomposition rule

$$\mathbf{m}_1^I = \sum_k [\mathbf{r}_I^k m_0^{\{I, k\}} + \mathbf{m}_1^{\{I, k\}}], \quad (9)$$

where \mathbf{r}_I^k is the vector that goes from the center of block I to the center of its child k .

The algorithm used to estimate m_0 and \mathbf{m}_1 is then the following. At each haptic cycle, we descend the octree until we find blocks that are either fully contained or partially intersecting the burr sphere. If they are fully contained, we add their contribution to m_0 and \mathbf{m}_1 ; if they are partially intersecting, we compare the block size with ℓ and if it is larger we refine; otherwise, we add the partial volume contributions

$$\Delta m_0 = \frac{\Delta V}{V_I} m_0^I, \quad (10)$$

$$\Delta \mathbf{m}_1 = \frac{\Delta V}{V_I} (\mathbf{R}_b - \mathbf{R}_c)(m_0^I + \mathbf{m}_1^I \cdot (\mathbf{R}_c - \mathbf{R}^I)) + O((\ell/R)^2), \quad (11)$$

where ΔV is the volume of the region of intersection between block I and the sphere, \mathbf{R}_c is the position of the center of mass of the latter intersection, and \mathbf{R}^I is the position of the center of block I . In the current implementation of the algorithm, both ΔV and \mathbf{R}_c are approximated by replacing the block with a sphere of equal volume (see [3] for details).

Therefore, at the cost of a minor computational overhead in the precomputing and update (see below the discussion on erosion), we are able to estimate m_0 and \mathbf{m}_1 with a computational cost that grows as most as $h^2 R / \ell^3$. Moreover, the availability of the precomputed \mathbf{m}_1^I moments allow us to estimate the contribution of partially overlapping blocks at a higher order to what would have been possible using only m_0^I . We are thus allowed to use larger values for ℓ .

3.2 Multi-scale erosion

Erosion, i.e. material removal in response to burring, is modeled as a position dependent erosion rate described by f , an erosion shape function,

$$\frac{d\rho(\mathbf{r})}{dt} = \alpha f(r/R) \rho(\mathbf{r}); \quad (12)$$

where, again, \mathbf{r} is measured from the center of B , and α is an appropriate dimensional constant. f is constrained to have a maximum at $r/R = 0$ and to be null for $r/R > 1$. In a previous work,[3], erosion was modeled by assuming that all the power spent by working against the frictional forces on a ‘‘contact surface’’ element of the bone would have gone toward the erosion of the bone material on the surface. The resulting expression for the local mass derivative was, however, rather complex and computationally expensive. Eq. (12) provides essentially comparable results at a much lower computational cost.

From the point of view of the implementation, in our model the bone is described as a collection of voxels, each one containing up to 255 values of bone occupation. To accommodate for a wide range of erosion rates using only

8 bits, we convert the rate of erosion given in Eq. (12) to a probability that the value of the voxel at position \mathbf{r} will be reduced by one at next time step. A Russian roulette scheme is then used for deciding whether to fully erode a bit (i.e. remove 1/255th of the mass of a full voxel) or not.

To find the voxels that should be eroded, we integrate the following modifications to the octree descent algorithm introduced above. When we identify a block as contained in the burr, we descend down to all the leaves and erode the voxels using the probabilistic version of Eq. (12). When the block is instead only partially contained in the burr, we continue recursion until we find completely contained sub blocks and then proceed as above. If we reach a leaf which is only partially contained, the erosion probability is scaled by the overlap fraction before testing for erosion. In descending the octree we keep track of the number of voxels touched while visiting a node children. If it changes, we perform an update of the node value from its children values using the same scheme used for octree construction (i.e. pulling moment updates from octree leaves up to the root).

3.3 Other contributions to the haptic response

Together with the elastic force \mathbf{F}_e defined in eq. (3), we also compute a frictional force, \mathbf{F}_μ , that is supposed to model the friction forces that oppose burr rotation when the latter is in contact with the bone material; and an impact force, \mathbf{F}_i that can be thought as what would be the response of the bone material if it were modeled as a collection of unconnected point masses swept by the moving burr sphere.

$$\mathbf{F}_\mu = c_\mu R^2 (h/R) (\mathbf{m}_1/m_0) \times \boldsymbol{\omega} \quad (13)$$

$$\mathbf{F}_i = -(c_i R^2) (h/R) \mathbf{V} \quad (14)$$

where we have introduced $\boldsymbol{\omega}$, representing the burr angular velocity vector, and \mathbf{V} the velocity of the burr center.

4 Secondary visual effects

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 suction device[1].

For the computational reasons discussed in section 2, we are modeling the dust/fluid dynamics using what essentially amounts to an hybrid particles-volumetric model, inspired by previous work on particle systems and sandpiles [21, 16]. In this scheme, particles are created by the irrigator, which injects water particles, by blood spots and vessels, that inject blood particles, and by the burr during erosion, that converts bone to bone dust particles. All particles move ballisti-

cally when in empty space, and interact with the other materials according to a set of rules that ensure that only a single particle may occupy a given voxel at given time (see [2]). Particles are deleted when they exit from the operation site or when they are sucked by the suction device.

The computational cost of update in this scheme is essentially constant per particle and, thus, the total computation cost would naively grow linearly in the number of particles and quickly degrade the real-time performance of the system. To avoid this problem, we are using a time-critical evolution algorithm designed to trade simulation quality with speed. The idea behind the algorithm is to concentrate resources on the visually most important parts of the simulation, by controlling both individual particles update rates and total number of particles.

The update rate control methods associates to each particle an update rate proportional to the particle speed. To avoid the costs associated to sorting the particles, the particles are divided in groups, $\{G_i\}$, so that all the particles in group G_i have speed v , measured in units of a predefined reference maximal velocity scale, in the range $2^{-i} \leq v < 2^{-(i+1)}$. Particle velocities are clamped so that they cannot be larger than the maximal velocity scale. At each evolution time step we randomly select $\{a_i\}$ particles from each group and, for each selected particle, integrate the motion from its last recorded time of update to the current time.

The effective time step for particles in group G_i is then $(dt)_i = n_i/a_i(dt)_\mu$ where n_i is the number of particles in group G_i and $(dt)_\mu$ is the actual simulation time step. The selection counters $\{a_i\}$ are chosen so that, on average, particles in channel i will move with a time step $(dt)_i = 2(dt)_{i-1}$, and thus

$$\frac{a_{i+1}}{n_{i+1}} = \frac{1}{2} \frac{a_i}{n_i}. \quad (15)$$

The total computational cost for one time step will then be $W = w \sum_i a_i$ where w is the average cost per particle update, which is measured at run time by the simulator. Using the equation above we find that, when all the $n_i > 0$,

$$A = \sum_i a_i = \frac{a_0}{n_0} \sum_i \frac{n_i}{2^i}. \quad (16)$$

Therefore, for given W , w , and n_i , we can reconstruct the required a_i . The case $n_j = 0$ for some j is a trivial generalization of the above.

Given a reasonable approximation of w , the update rate control algorithm is guaranteed to meet timing constraints and to probabilistically move the particles with the largest visual error. If the update rate of the particle system falls below a specified threshold (currently, if we move less than 10% of the particles per step), we reduce the particle count by removing the “less important” ones. The importance of

a particle is currently inversely proportional to the distance from the current lookat point of the microscope and to the particle velocity.

5 System integration

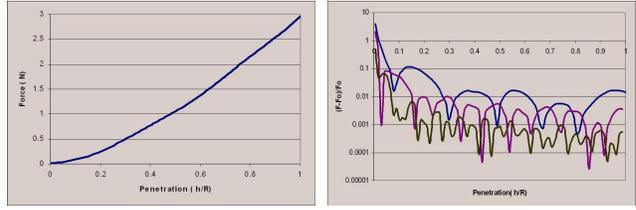
Our technique for bone dissection simulation has been integrated in a prototype training system for mastoidectomy. We summarize here the general system architecture, which is described in more details in reference [2].

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. The "slow" subsystem is responsible for the global evolution of the water, bone dust and bone paste. 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. The system runs on two interconnected multiprocessor machines. The data is initially replicated on the two machines. The first is dedicated to the high-frequency tasks: haptic device handling and bone removal simulation, which run at 1 KHz. The second concurrently runs, at about 15–20 Hz, the low-frequency tasks: bone dust generation, 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 specialized volumetric rendering component, exploiting multi-texturing and register combiner OpenGL extensions, provides the required high frequency visual feedback.

6 Implementation and results

Our current configuration has a single-processor PIV/1400 MHz with 256 MB PC133 RAM for the high-frequency tasks and a dual-processor PIII/800 MHz with 512 MB PC800 RAM and a NVIDIA GeForce 4 Ti4600, for the low frequency tasks. Haptic feedback is provided by a Phantom Desktop haptic device for the dominant hand and a Phantom 1.0 haptic device for the non-dominant hand, while visual feedback is provided by a n-vision VB30 binocular display.

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.



(a) Elastic force

(b) Relative error

Figure 2. Virtual bone reaction against burr penetration and relative error in force evaluation introduced by the multi-scale algorithm.

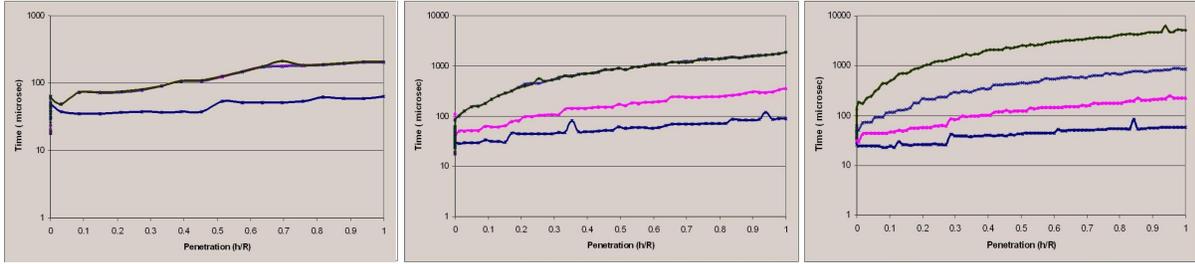
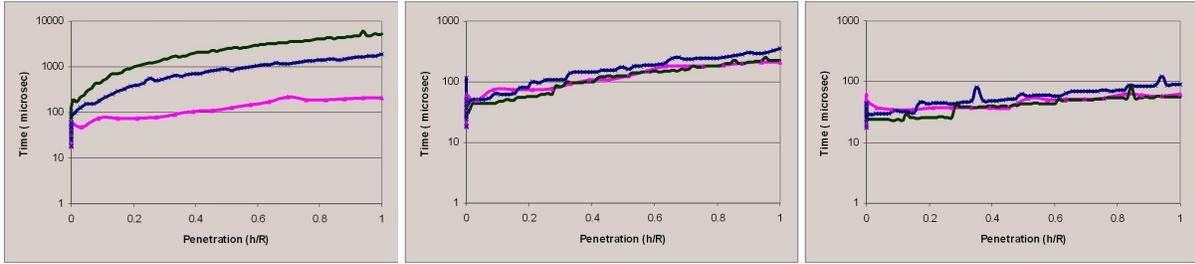
We are currently using a volume of 256x128x128 cubical voxels (0.3 mm side) to represent the region where the operation takes place. The resolution of the volume is the same as the original CT data.

In the following we will report on a series of experiments done using the prototype described above.

6.1 Multi-resolution Force Evaluation

Figure 2(a) shows the reaction of the virtual bone against burr penetration, using different burr-sizes and different accuracy parameters. The computations are done in absence of erosion, $\alpha = 0$ in equation 12, and using the actual force evaluation kernel of the force-feedback loop with a volume composed of cubical voxels with 0.3 mm side. The figure shows the "elastic" response of the material when using two different burr sizes ($R = 1.0mm, R = 5.0mm$), which correspond to a standard polishing burr tip and a large initial burring tip. The force has been computed using the mono-resolution algorithm, as well as three different accuracy settings of the multi-resolution algorithm, corresponding to ($\ell = 0.1R, \ell = 0.3R, \ell = 0.5R$). The graphs clearly show that the mono-resolution and the multi-resolution version of the algorithm are in agreement. In figure 2(b), we report how the relative error with respect to the reference mono-resolution solution changes with penetration. As it can be seen from the figure, it is typically of the order of few percents or below. The oscillations in the curves are due to resonances between the burr position and the octree grid used to compute the forces.

In figure 3, we report the wall clock time required by the force computation kernel to compute the forces of figure 2(a). Each subfigure shows the wall clock time required for different values of the burr radius and for different resolution scales. For $R = 1mm$, as expected, there is no appreciable between the mono-resolution results and the multi-resolution ones for $\ell = 0.1R, \ell = 0.3R$, while the $\ell = 0.5R$

(a) $R = 1\text{mm}$ (b) $R = 3\text{mm}$ (c) $R = 5\text{mm}$ **Figure 3. Time required to compute the forces of figure 2(a) compared by radius.**(a) $\ell = 0$ (b) $\ell = 0.3R$ (c) $\ell = 0.5R$ **Figure 4. Time required to compute the forces of figure 2(a) compared by resolution scale.**

is faster. For $R = 3\text{mm}$ $\ell = 0.1R$ is still of the same order of the voxel size, 0.25mm , while the $\ell = 0.3R$ and $\ell = 0.5R$ are now clearly faster than the monoresolution case.

In figure 4, we show the growth of the computational cost for a given resolution scale and different radius values, $R = 1, 3, 5\text{mm}$. The figures show the growth of the computational cost for a given resolution scale and different radius values, $R = 1, 3, 5\text{mm}$. It is clear from the figures that the mono-resolution algorithm is limited to $R < 2\text{mm}$, the computational cost of the multi-resolution algorithm grows very slowly with R , and always easily meets the 1ms haptic feedback time constraint.

6.2 Visual feedback

Figure 1 shows a few frames taken from the live recording of a typical virtual bone dissection sequence performed in the mastoid region. The tool on the left is the suction device. It interacts with the scene by simply removing all the particles within a certain radius from its tip. The tool on the right is the burr, connected with the irrigator. All the visual feedback in the images is provided by the particle simulator, that transforms removed bone into bone dust particles, injects water and blood particles, and removes the

resulting paste. The selected images correspond to what is seen by the user in the microscope. On our PIII/800 simulator machine, with standard simulation settings, we use ten channels for velocity sorting and impose a simulator time step of $(dt)_\mu = 10\text{ms}$ and a visual feedback rate of ten to twenty frames per second. The total number of particles in the scene is in the order of the tens of thousands, with a per-particle update time w of a few microseconds.

7 Conclusions and Future Work

We have presented adaptive techniques for providing real-time haptic and visual feedback during simulation of a bone cutting burr. The simulator is being developed as a component of a training system for temporal bone surgery. The current implementation, directly operating on a voxel discretization of patient-specific 3D CT and MR imaging data, is efficient enough to provide real-time feedback on a low-end multi-processing PC platform. Thanks to our adaptive techniques, we are able to simulate in real-time a wide range of operating conditions, such as initial cortex burring with large burr tips (up to 5mm of radius), and deeper burring with accumulation of debris.

Our future work will concentrate on testing the simulator with end users and on comparing the results obtained

with our simplified model with experimental data. Subjective input from selected end users is encouraging. One major limitation identified by end users is the resolution of the dataset, which is considered enough for early training but does not have enough anatomical detail to allow recognition of fine features. To this end, IERAPSI project partners are working towards providing higher resolution datasets derived from combined MRI and MicroCT data, similarly to the work presented in reference [22]. Another area of improvement will be in the realism of the dust/fluid simulation. We plan to introduce in the next version of the simulator a more sophisticated treatment of particle collisions and thus to remove limitations such as the single particle per voxel constraint of the current implementation.

8 Acknowledgments

We would like to thank Pietro Ghironi and Fabio Bettio, CRS4, for technical support, and Prof. Stefano Sellari Franceschini, University of Pisa, for his collaboration in the design and testing of the system. These results were obtained within the framework of the European Union IERAPSI project (EU-IST-1999-12175).

References

- [1] M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, N. W. John, and R. 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*, pages 17–23. IOS Press, Jan. 2002.
- [2] M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, and A. Zorcolo. A multiprocessor decoupled system for the simulation of temporal bone surgery. *Computing and Visualization in Science*, 5(1), 2002.
- [3] M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, and A. Zorcolo. Real-time haptic and visual simulation of bone dissection. In *IEEE Virtual Reality Conference*, pages 209–216. IEEE Computer Society Press, Feb. 2002.
- [4] R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of the conference on Visualization '96*, pages 197–ff. IEEE Computer Society Press, 1996.
- [5] A. Baerentzen. Octree-based volume sculpting. In *IEEE Visualization*, 1998.
- [6] J. Bryan, D. Stredney, G. Wiet, and D. Sessanna. Virtual temporal bone dissection: A case study. In *IEEE Visualization*, pages 497–500, 2001.
- [7] E. Ferley, M. P. Cani, and J.-D. Gascuel. Practical volume sculpting. *The Visual Computer*, 16(8):469–480, 2000.
- [8] S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 249–254. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [9] T. A. Galyean and J. F. Hughes. Sculpting: an interactive volumetric modeling technique. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 267–274. ACM Press, 1991.
- [10] A. García-Alonso, N. Serrano, and J. Flaquer. Solving the collision detection problem. *IEEE Computer Graphics and Applications*, 14(3):36–42, 1994.
- [11] T. Harada, S. Ishii, and N. Tayama. Three-dimensional reconstruction of the temporal bone from histological sections. *Arch Otolaryngol Head Neck Surg*, 114:1139–1142, 1988.
- [12] T. He and A. Kaufman. Collision detection for volumetric objects. In *Proceedings of the conference on Visualization '97*, pages 27–ff. ACM Press, 1997.
- [13] A. Kaufman, D. Cohen, and R. Yagel. Volume graphics. *Computer*, 26(7):51–64, July 1993.
- [14] R. Koppersmith, R. Johnston, D. Moreau, R. 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. IOS Press, January 1997.
- [15] L. Landau and E. Lifshitz. *Theory of elasticity*. Pergamon Press, 1986.
- [16] X. Li and J. Moshell. Modeling soil: Realtime dynamic models for soil slippage and manipulation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 361–368, 1993.
- [17] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degrees-of-freedom haptic rendering using voxel sampling. In A. Rockwood, editor, *Siggraph 1999, Annual Conference Series*, pages 401–408. ACM Siggraph, Addison Wesley Longman, 1999.
- [18] R. A. Nelson. *Temporal Bone Surgical Dissection Manual*. House Ear Institute, Los Angeles, second edition, 1991.
- [19] A. Pettigrew. Pettigrew plastic temporal bones. <http://www.temporal-bone.com/>.
- [20] B. Pflessner, A. Petersik, U. Tiede, K. H. Hohne, and R. Leuwer. Haptic volume interaction with anatomic models at sub-voxel resolution. In *10th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Proc. Haptics 2002*, pages 66–72, 2002.
- [21] J. H. R.W. Sumner, J.F. O'Brien. Animating sand, mud and snow. *Computer Graphics Forum*, 18:1, 1999.
- [22] D. Stredney, G. Wiet, J. Bryan, D. Sessanna, J. Murakami, O. Schamllbrock, K. Powell, and B. Welling. Temporal bone dissection simulation – an update. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, and D. Stredney, editors, *Medicine Meets Virtual Reality 2002*, pages 507–513. IOS Press, Jan. 2002.
- [23] S. W. Wang and A. E. Kaufman. Volume sculpting. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–ff. ACM Press, 1995.
- [24] G. Wiet, J. Bryan, D. Sessanna, D. Stredney, P. Schmalbrock, and B. Welling. Virtual temporal bone dissection simulation. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 2000*, pages 378–384. IOS Press, January 2000.