

This technical report is a more detailed version of the following paper:

Morris D, Sewell C, Barbagli F, Blevins N, Girod S, Salisbury K. Visuohaptic Simulation of Bone Surgery for Training and Evaluation. *IEEE Computer Graphics and Applications*, Vol. 26, No. 4, November 2006, p48-57.

# Visuohaptic Simulation of Bone Surgery

Dan Morris, Christopher Sewell, Federico Barbagli, Nikolas H. Blevins, Sabine Girod,  
and Kenneth Salisbury, *Member, IEEE*

**Abstract**—We present techniques for the visual and haptic simulation of bone surgery, with a specific focus on procedures involving the temporal bone and the mandible. We discuss our approaches to graphic and haptic rendering and interactive modification of volumetric data, specifically focusing on generating force-feedback effects that are relevant to bone drilling. We then discuss how our rendering primitives and simulation architecture can be used to build surgical training techniques that are *not* available in traditional cadaver-based training labs, offering new possibilities for surgical education. In particular, we discuss the automatic computation of performance metrics that can provide real-time feedback about a trainee's performance in our simulator. We also present results from an experimental study evaluating the construct validity of our simulation and the validity of our performance metrics.

**Keywords**—Virtual reality, simulation, input devices, volume visualization, surgery, haptics, simulator evaluation, simulator validity



## 1 INTRODUCTION

**S**URGICAL training has traditionally revolved around an apprenticeship model: residents observe experienced surgeons in the operating room, and eventually are deemed ready to perform their first procedure [1]. In recent years, simulation-based training has emerged as a potential adjunct to this method, and the value of simulation-based learning has been more widely accepted [2]. Simulation can be a safe, cost-effective, customizable, and easily-accessible tool for gaining experience in surgery.

This paper will present methods for simulating surgeries involving bone manipulation, with a specific focus on two categories of procedures: temporal bone surgery and mandibular surgery. Section 1 will provide relevant clinical background on the target procedures. Section 2 will describe the algorithms and data structures used for interactive haptic and graphic rendering, specifically targeted toward providing key sources of intraoperative feedback for surgical interaction with bones. Section 3 will present the results of a study which evaluates the construct validity of our system (its ability to discriminate expert surgeons from novices). Section 4 will describe features of our simulation environment that do not exist in traditional, cadaver-based training labs. Section 5 will discuss our approach to automatically evaluating a trainee's performance in our environment.

We begin with a brief description of the relevant surgical procedures.

### 1.1 Temporal Bone Surgery

Several common otologic surgical procedures – including mastoidectomy, acoustic neuroma resection, and cochlear implantation – involve drilling within the temporal bone to access critical anatomy within the middle ear, inner ear, and skull base. As computer simulation is becoming a more frequently used technique in surgical training and planning, this class of procedures has emerged as a strong candidate for simulation-based learning.

The time spent on a procedure in this area is typically dominated by bone removal, which is performed with a series of burrs (rotary drill heads) of varying sizes and surface properties. Larger burrs are generally used for gross bone removal in the early part of a procedure, while smaller burrs are used for finer work in the vicinity of target anatomy. Surgeons employ a variety of strokes and contact techniques to precisely control bone removal while minimizing the risk of vibration and uncontrolled drill motion that could jeopardize critical structures.

### 1.2 Mandibular Surgery

Incorrect alignment of the jaws – due to congenital malformation, trauma, or disease – can result in cosmetic deformation and problems with chewing and/or breathing. Orthognathic surgeries correct such problems, typically by inducing a fracture in one or both jaws (generally using a bone saw), displacing the fractured components into an anatomically preferable configuration, and installing bone screws and/or metal plates to fix the bone segments in their new positions.

- D. Morris, C. Sewell, F. Barbagli, and K. Salisbury are with the Robotics Laboratory, Department of Computer Science Stanford University, Stanford, CA. E-mail: {dmorris, csewell, barbagli, jks}@robotics.stanford.edu
- N. Blevins is with the Department of Otolaryngology, Stanford University Medical Center, Stanford, CA. E-mail: nblevins@stanford.edu
- S. Girod is with the Division of Plastic Surgery, Stanford University Medical Center, Stanford, CA. E-mail: sgirod@stanford.edu

This approach is often prohibited by the severity of the deformation, the size of the separation that would be required after fracture, or the sensitivity of the surrounding soft tissue. In these cases, distraction osteogenesis is often employed as an alternative. Here a similar procedure is performed, in which only a minor separation is created intraoperatively. Instead of spanning the gap with a rigid plate, an adjustable distractor is fixed to the bone on both sides of the gap. The distractor can be used to gradually widen the fracture over a period of several weeks, allowing accommodation in the surrounding tissue and allowing the bone to heal naturally across the fracture.

These procedures are likely to benefit from surgical simulation for several reasons. The complex, patient-specific planning process and the significant anatomic variation from case to case suggests that an end-to-end simulator will assist physicians in preparing for specific cases. Furthermore, distraction procedures have been introduced to the craniofacial surgical community only within the last ten to fifteen years, and an effective simulator will significantly aid in the training and re-training of this new class of procedures, and with the exploration of alternative techniques for effective surgeries.

### 1.3 Current Training Techniques

Resident training in otologic surgery typically includes dissection of preserved human temporal bones. This allows residents to become acquainted with the mechanical aspects of drilling, but does not incorporate physiological information, continuous feedback for hazard avoidance, or soft tissue work. Temporal bone labs are also costly to maintain, and cadaver specimens can at times be difficult to obtain in sufficient quantity. This approach also limits the precision with which an instructor can monitor a trainee's drilling performance, as the instructor cannot feel the fine details of the trainee's interaction with the bone surface, and cannot easily share the drill and bone surface for demonstration. A further limitation of cadaver-based training is that instructors have little or no mechanism for controlling anatomic variations or the presence of specific pathology that can lead to challenging training scenarios. Interactive atlases such as [3] are available for training regional anatomy. Two-dimensional simulations [4] are available for high-level procedure training.

Surgical training in craniofacial surgery typically does not include cadaver-based procedures. Most residents learn anatomy primarily from textbooks and models; surgical technique is learned through apprenticeship and procedure observation.

### 1.4 Previous Work

Previous work in interactive simulation of temporal bone surgery ([5],[6],[7]) has focused primarily on haptic rendering of volumetric data. Agus et al [5] have developed an analytical model of bone erosion as a function of applied drilling force and rotational velocity, which they have verified with experimental data [20]. Pflesser et al [7] and Petersik et al [14] model a drilling instrument as a point cloud, and use a modified version of the Voxmap-Pointshell algorithm [8] to sample the surface of the drill and generate

appropriate forces at each sampled point. This work has also been incorporated into a commercial simulator [19]. Each of these projects has incorporated haptic feedback into volumetric simulation environments that make use of CT and MR data and use volume-rendering techniques for graphical display.

Agus et al [5] describe several enhancements to their simulation environment that incorporate additional skills, including the use of irrigation and suction; and additional sources of intraoperative feedback, including real-time rendering of bone dust.

Additional work has focused on non-interactive simulation of craniofacial surgery for planning and outcome prediction ([9],[10],[11]). [12] discusses preliminary work on interactive simulation of craniofacial surgery, and [13] presents a simulation architecture for arthroscopic procedures.

## 2 SIMULATION AND RENDERING

The goal of our simulation is high-fidelity presentation of the visual and haptic cues that are present in a surgical environment. This section will discuss our overall rendering scheme, and will focus on how we present the specific cues that are relevant to surgical training.

### 2.1 Data Sources and Preprocessing

Models are loaded from full-head or temporal bone CT data sets, thresholded to isolate bone regions, and resampled to produce isotropic voxels, 0.5mm on a side. Using a standard resampled resolution allows us to calibrate our rendering approaches independently of the image sources used for a particular simulation case.

### 2.2 Hybrid Data Structure Generation

In order to leverage previous work on haptic rendering of volumetric data [14] while still maintaining the benefits of surface rendering in terms of hardware acceleration and visual effects, we maintain a hybrid data structure in which volumetric data are used for haptic rendering and traditional triangle arrays are used for graphic rendering. In order to simplify and accelerate the process of updating our polygonal data when the bone is modified, we build a new surface mesh – in which vertices correspond directly to bone voxels – rather than using the original isosurface mesh.

The voxel array representing the bone model is loaded into our simulation environment, and a polygonal surface mesh is generated to enclose the voxel grid. This is accomplished by exhaustively triangulating the voxels on the surface of the bone region, i.e.:

```
for each voxel v1
  if v1 is on the bone surface
    for each of v1's neighbors v2
      if v2 is on the bone surface
        for each of v2's neighbors v3
          if v3 is on the bone surface
            generate vertices representing v1,v2,v3
            generate a triangle t(v1,v2,v3)
            orient t away from the bone surface
```

Here being 'on the bone surface' is defined as having non-

zero bone density and having at least one neighbor that has no bone density. Although this generates a significant number of triangles (on the order of 200,000 for a typical full-head CT data set), we use several techniques to minimize the number of triangles that are generated and/or rendered. To avoid generating duplicate triangles, each voxel is assigned an index before tessellation, and triangles are rejected if they do not appear in sorted order. A second pass over the mesh uses the observations presented in [15] to eliminate subsurface triangles that will not be visible from outside the mesh.

Voxels are stored in a compact, in-memory hash table, which is indexed by three-dimensional grid coordinates. This allows very rapid point/volume collision-detection without excessive memory requirements.

Secondary data structures map each voxel to its corresponding vertex index, and each vertex index to the set of triangles that contain it. This allows rapid access to graphic rendering elements (vertices and triangles) given a modified bone voxel, which is critical for shading vertices based on voxel density and for re-triangulation when voxels are removed (see Section 2.4). Figure 1 summarizes the relevant data structures.

### 2.3 Haptic Rendering

Virtual instruments are controlled using a SensAble Phantom [16] haptic feedback device, which provides three-degree-of-freedom force-feedback and six-degree-of-freedom positional input. Users can select from a variety of drills, including diamond and cutting burrs ranging from one to six millimeters in diameter. We will first discuss our approach to gross force-feedback, then we will present our methods for providing specific haptic cues that are relevant to surgical training.

#### *Gross Feedback: Volume Sampling*

We initially adopted a haptic feedback approach similar to [14], in which the drill is represented as a cloud of sample points, distributed approximately uniformly around the surface of a spherical burr. At each time step, each sample point is tested for contact with bone tissue. By tracing a ray from each immersed sample point toward the center of the tool, the system can generate a contact force that acts to

move that sample point out of the bone volume (Figure 2a).

We found that this approach worked well overall, as reported by [14], but had several undesirable artifacts. Due to sampling effects (Figure 2a), this approach produced uneven voxel removal at high resolutions, creating unrealistic bone removal patterns that depended on surface sampling. Furthermore, floating-point computations are required to find the intersection points at which rays enter and leave voxels. Since sampling density is limited by the number of samples that can be processed in a haptic timestep (approximately one millisecond), extensive floating-point computation limits the potential sampling density. This sparse sampling limits the effective stiffness of the simulation (which depends on rapid and accurate computation of penetration volume), which disrupts the illusion of contact with a highly rigid object. Furthermore, this sparse sampling limits the implementation of certain higher-level effects, such as bone modification that is dependent on the precise sub-parts of the drill that are used to contact the bone. These drawbacks motivate an approach that uses a higher ratio of integer to floating-point computation and allows a higher sampling density.

We thus take a more exhaustive approach to sampling the tool for haptic feedback and bone density reduction. The tool itself is discretized into a voxel grid (generally at a finer resolution than the bone grid), and a preprocessing step computes an occupancy map for the tool's voxel array. At each interactive timestep, each of the volume samples in the tool is checked for intersection with the bone volume (a constant-time, integer-based operation, using the hash table described in Section 2.2). A sample point that is found to lie inside a bone voxel generates a unit-length contribution to the overall haptic force vector that tends to push this sample point toward the tool center, which – with adequate stiffness – is always outside the bone volume (Figure 2b). Thus overall penetration depth is computed based on the *number* of immersed sample points, rather than on the results of a per-sample ray-trace.

The overall force generated by our approach is thus oriented along a vector that is the sum of the “contributions” from individual volume sample points. The magnitude of this force increases with the *number* of sample points found to be immersed in the bone volume.

#### *Nonlinear magnitude computation*

Because the drill is densely sampled, a large number of sample points often become immersed immediately after the drill surface penetrates the bone volume, which leads to instability during low-force contact. Reducing the overall stiffness leads to “softer” haptic feedback that does not accurately represent the stiffness of bone. We thus employ a multi-gain approach, in which the magnitude of haptic feedback is a nonlinear function of the number of immersed sample points.

More specifically, we define two gains, one of which is used when fewer than a threshold number of sample points are immersed; the other is used for deeper penetrations. This threshold is set such that the discontinuity in the force function occurs shortly after contact is initiated, so no dis-

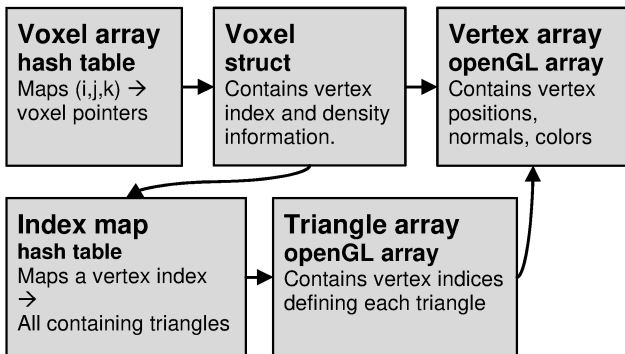


Fig 1. A summary of the structures binding our volumetric (haptic) and surface (graphic) rendering data. When voxels are removed or modified, the corresponding vertices and triangles can be accessed from the (i,j,k) voxel index in approximately constant time.

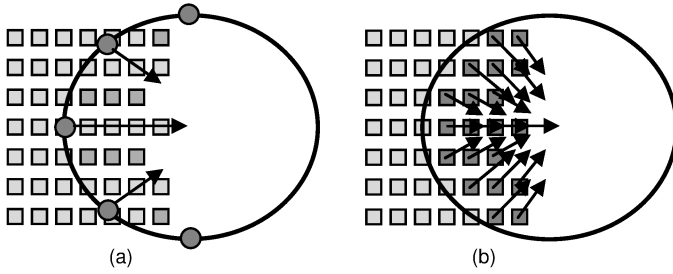


Fig 2. A summary of contrasting approaches to haptic rendering. (a) The ray-tracing approach. Red points are surface samples on the surface of a spherical drill. Each sample contributes a vector to the overall force that points toward the tool center and is proportional to the penetration of the sample. Voxels labeled in purple would be missed by the raytracing algorithm, thus creating uneven bone removal. (b) Our volume-sampling approach. Here, the full volume of the drill is sampled, and each point that is found to be immersed in the bone volume contributes a vector to the overall force that points toward the center of the tool but is of unit length.

continuity is perceived by the user. This relationship is summarized in Figure 3. We find that this approach allows large stiffnesses during haptic interaction, while avoiding instability during the “high-risk” period immediately following initial penetration.

Our volume-sampling approach requires sampling a significantly higher number of points than the ray-tracing approach, since the complete volume of the burr is sampled, instead of just the surface. However, the operation performed when a tool sample is found to lie within the bone volume is a constant-time computation, rather than a complex ray-tracing operation. Overall, we are able to achieve a significantly higher stiffness than they ray-tracing approach allows. We do build on the ray-tracing approach for less time-critical tasks, including bone thickness estimation (Section 2.9) and haptic feedback for non-physically-based tools (Section 2.5).

#### Modeling Drill Surface Non-uniformity

Our system also associates a “drilling power” with each sample point based on its location within the drill head; each tool voxel that intersects a bone voxel removes an amount of bone density that depends on the drilling power of the sample point. This approach allows us to simulate key aspects of drill/bone contact, particularly the fact that the equatorial surface of the burr carries a larger linear velocity than the polar surface and thus removes more bone

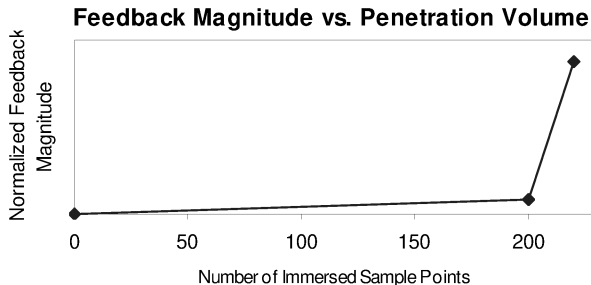


Fig 3. Multi-gain mapping from penetration volume (number of immersed sample points) to feedback magnitude.

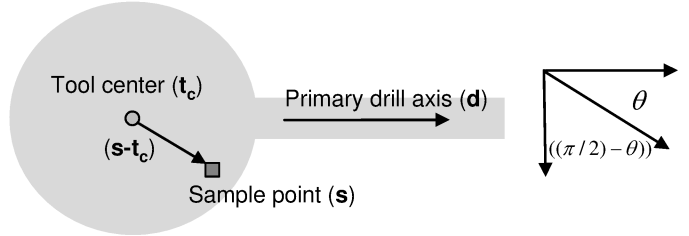


Fig 4. The computation of the “latitude” of a volume sample point for bone removal rate computation.

per unit of applied force. Simulating this effect is critical for encouraging trainees to use proper drilling technique.

More precisely, the amount of bone removed per unit time by a given sample point is computed as  $R_{br}$  in the following expression:

$$\theta = \text{abs}(\cos^{-1}(\mathbf{d} \bullet (\mathbf{s} - \mathbf{t}_c)))$$

$$R_{br} = f \bullet \max(0, R_{\max} - \text{falloff} \bullet \text{abs}((\pi/2) - \theta))$$

...where  $\mathbf{s}$  is the location of this sample point,  $\mathbf{t}_c$  is the location of the tool center,  $\mathbf{d}$  is the axis of the tool handle, and  $\theta$  is thus the angle between the drill handle and  $(\mathbf{s} - \mathbf{t}_c)$ . The expression  $\text{abs}(\pi/2 - \theta)$  is thus the “latitude” of the current sample point. *falloff* is a constant parameterizing the non-uniformity of the drill surface. If *falloff* is zero, the pole and the equator of the drill remove bone with equal efficiency.  $R_{\max}$  is the maximum rate of bone removal per unit force, and  $f$  is the magnitude of force currently being applied by the user. The computation of latitude is summarized in Figure 4. Note that *falloff* parameters are precomputed for drill samples to avoid performing expensive arc-cosine operations hundreds of times per haptic timestep.

This approach allows us to encourage proper drilling technique and to model critical differences among burr types. For example, our model captures the fact that cutting burrs typically show more dependence on drilling angle than diamond burrs do, but have higher overall bone removal rates. A cutting burr would thus be associated with both a higher  $R_{\max}$  and a higher *falloff* in the above expression.

#### Modeling Tangential Forces

Another property of surgical drills that should be accurately represented in a simulation environment is their tendency to drag the user along the surface of the bone, due to the contact forces between the teeth of the drilling burr and the bone (Figure 5). Stroking the drill on the bone surface in a direction that allows these forces to *oppose* a surgeon’s hand motion permits the surgeon to control the velocity of the drill. Stroking the drill such that these forces complement the surgeon’s hand motion causes the drill to catch its teeth on the bone and rapidly “run” in the direction of movement, which can be extremely dangerous. Simulating this effect is thus critical to training correct drilling technique.

Modeling the contact forces between the individual teeth in the drill's geometry and the bone surface would be computationally expensive, so we again employ our dense sampling approach to approximate tangential drill forces during the computation of penalty forces.

Each sample that is found to be immersed in the bone (i.e. the red samples in Figure 2b) computes its own tangential force vector, according to:

$$\mathbf{f}_{\text{tan}} = (\mathbf{p} - \mathbf{sc}) \times \mathbf{d}$$

...where  $\mathbf{f}_{\text{tan}}$  is the tangential force created by this sample,  $\mathbf{p}$  is the position of this sample,  $\mathbf{sc}$  is the center of the "slice" of the drill in which this sample lies (the sample position projected onto the drill axis), and  $\mathbf{d}$  is the primary axis of the drill (and thus the axis of rotation), as shown in Figure 4.

The vector  $(\mathbf{p} - \mathbf{sc})$  is a vector from the tool axis to this sample point, an approximation of the local surface normal (the true surface normal is generally unknown, since most samples are not on the surface of the model and thus don't have defined normals). The drill axis vector is normalized to unit length, and the magnitude of the vector  $(\mathbf{p} - \mathbf{sc})$  indicates its distance from the tool axis and thus its linear velocity (since the drill spins at constant rotational velocity, samples farther from the axis of rotation carry larger linear velocity than close near the axis of rotation). The cross-product  $(\mathbf{p} - \mathbf{sc}) \times \mathbf{d}$  is thus scaled according to sample velocity, and is perpendicular to both the drill's axis and the approximate surface normal.

Summing these vectors over all samples that are found to lie on the bone creates a net force that simulates the interaction between the teeth of the drill and the bone surface. Scaling this vector by -1 is equivalent to reversing the "handedness" of the drill.

#### Modeling Drill Vibration using Recorded Data

Another key aspect of the haptic sensation associated with drilling is the vibration of the instrument, which varies with applied force and with burr type. In order to generate realistic drill vibration frequencies, we outfitted a physical drill with an accelerometer and collected vibration data at a variety of applied drilling forces. These data are summarized in Figure 6. The key spectral peaks were identified for each burr type and used to synthesize vibrations during the simulation. Since we are driving our haptic feedback

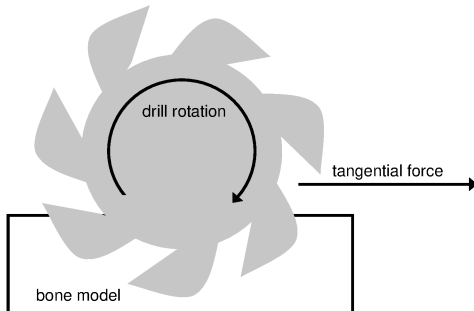


Fig 5. A spinning, burred drill creates a tangential force that propels the drill along the bone surface.

device at approximately 1.5 kHz, we are unable to preserve the highest-frequency vibrations identified in these experimental recordings. However, we are able to preserve the lower-frequency harmonics and the variations in vibration associated with changes in burr type and/or changes in applied drilling force.

#### 2.4 Data Manipulation

When bone voxels are removed from our environment, our hybrid data structure requires that the area around the removed bone be retessellated. Consequently, bone voxels are queued by our haptic rendering thread as they are removed, and the graphic rendering thread retessellates the region around each voxel pulled from this queue. That is, for each removed voxel, we see which of its neighbors have been "revealed" and create triangles that contain the centers of these new voxels as vertices. Specifically, for each removed voxel  $v$ , we perform the following steps:

```
for each voxel  $v'$  that is adjacent to  $v$ 
  if  $v'$  is on the bone surface
    if a vertex has not already been created
      to represent  $v'$ 
        create a vertex representing  $v'$ 
        compute the surface gradient at  $v'$ 
        queue  $v'$  for triangle creation

for each queued voxel  $v'$ 
  generate triangles adjacent to  $v'$  (see below)
```

Once again, a voxel that is "on the bone surface" has a non-zero bone density and has at least one neighboring voxel that contains no bone density. When all local voxels have been tested for visibility (i.e. when the first loop is complete in the above pseudocode), all new vertices are fed to a triangle generation routine. This routine finds new triangles that can be constructed from new vertices and their neighbors, orients those triangles to match the vertices' surface normals, and copies visible triangles to the "visible triangle array" (see Section 2.7). The reason for "queuing triangles for triangle creation" is that the genera-

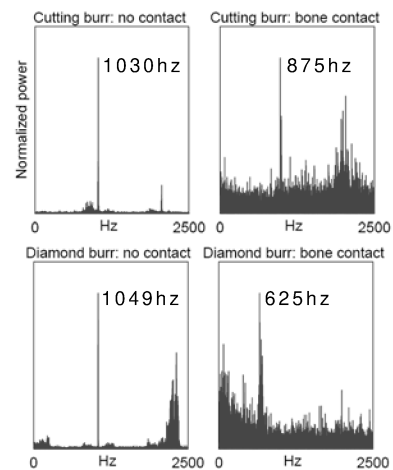


Fig 6. A spectral representation of drill vibration, collected from cutting (top row) and diamond (bottom row) drilling burrs, when in contact with bone and when powered but held away from the bone surface. The frequencies of the largest peaks are highlighted. The sharp spectral peaks make this data suitable for real-time vibration synthesis.

tion of triangles – performed in the second loop above – depends on knowing which local voxels are visible, which is only known after the completion of the first loop.

## 2.5 Additional Tools

An additional bone modification tool allows the introduction of large bone cuts via a planar cut tool (see Figure 7). This tool generates no haptic feedback and is not intended to replicate a physical tool. Rather, it addresses the need of advanced users to make rapid cuts for demonstration or for the creation of training scenarios. Bone removal with this tool is implemented by discretizing the planar area – controlled in six degrees of freedom – into voxel-sized sample areas, and tracing a ray a small distance from each sample along the normal to the plane. This is similar to the approach used in [14] for haptic rendering, but no haptic feedback is generated, and each ray is given infinite “drilling power”, i.e. all density is removed from any voxels through which each ray passes. The distance traced along each ray is controlled by the user. This allows the user to remove a planar or box-shaped region of bone density, demonstrated in Figure 7b. This approach will often generate isolated fragments of bone that the user wishes to move or delete. This operation is discussed in Section 2.6.

A final set of tools allows the user to manipulate rigid models that can be bound to bone objects. This is particularly relevant for the target craniofacial procedures, which center around rigidly affixing metal plates to the patient’s anatomy. We thus provide models of several distractors and/or industry-standard bone plates (it is straightforward to add additional models). The inclusion of these plate models allows users to plan and practice plate-insertion operations interactively, using industry-standard plates. Collision detection for haptic feedback is performed using a set of sample points, as was the case with drilling tools. In this case, the sample points are generated by sampling 100 vertices of each model and extruding them slightly along their normals (because these models tend to be very thin relative to our voxel dimensions) (Figure 8a). For this tool/bone contact, which generally involves objects with much larger volumes than the drill tools, we elected to use the ray-tracing approach presented in [14]. This approach allows reasonable haptic feedback with lower numbers of samples than the volumetric approach we use for our drilling tools (Section 2.3). Since there is no well-defined tool center toward which we can trace rays for penetration cal-

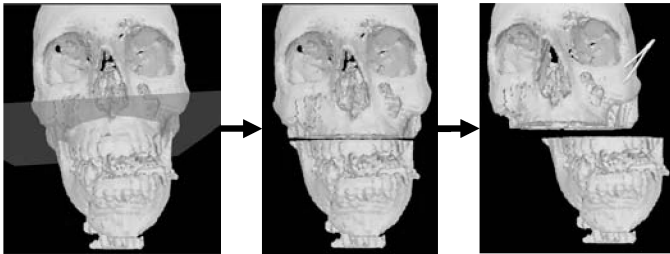


Fig 7. The use of the cut-plane tool and the independent manipulation of discontinuous bone regions. (a) The cut-plane tool is used to geometrically specify a set of voxels to remove. (b) The volume after voxel removal. (c) The flood-filling thread has recognized the discontinuity, and the bone segments can now be manipulated independently.

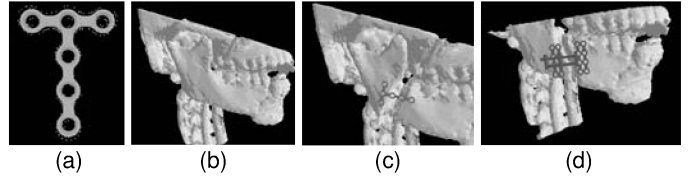


Fig 8. The modeling and attachment of rigid bone plates. (a) The surface of a bone plate after sampling and extrusion. (b) A bone surface before modification. (c) The same bone surface after drilling, distraction, and plate attachment. (d) The same bone surface after drilling, distraction, and distractor insertion.

culational, rays are traced along the model’s surface normal at each sample point. At any time, the user can rigidly affix a plate tool to a bone object with which it is in contact using a button on the haptic device (Figures 8b,8c,8d).

## 2.6 Discontinuity Detection

A critical step in simulating craniofacial procedures is the detection of cuts in the bone volume that separate one region of bone from another, thus allowing independent rigid transformations to be applied to the isolated bone segments.

In our environment, a background thread performs a repeated flood-filling operation on each bone structure. A random voxel is selected as a seed point for each bone object, and flood-filling proceeds through all voxel neighbors that currently contain bone density. Each voxel maintains a flag indicating whether or not it has been reached by the flood-filling operation; at the end of a filling pass, all unmarked voxels (which must have become separated from the seed point) are collected and moved into a new bone object, along with their corresponding data in the vertex and triangle arrays. Figure 9 summarizes this operation and provides an example.

Figures 7a and 7c display a bone object that has been cut and the subsequent independent movement of the two resulting structures. Here – for demonstration – the cut-plane tool is used to create the fracture; during simulated procedures, fractures are generally created by the drilling/sawing tools.

## 2.7 Graphic Rendering

To take advantage of the fact that the user does not frequently change the simulation’s viewing perspective, we maintain two triangle arrays, one containing the complete

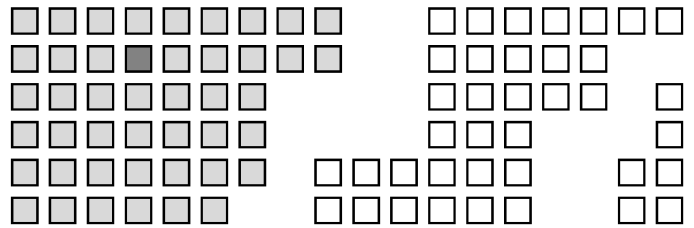


Fig 9. Discontinuity detection by flood-filling. The seed voxel is highlighted in red, and the shaded (blue) voxels were reached by a flood-filling operation beginning at the seed voxel. These voxels thus continue to be part of the same bone object as the seed voxel, while the unshaded voxels on the right have become disconnected and thus are used to create a new bone object. In a subsequent pass through the flood-filling algorithm, a third bone object would be created, because the unfilled voxels are further fragmented.

tessellation of the current bone volume (the “complete array”), and one containing only those that are visible from positions close to the current camera position (the “visible array”). The latter array is initialized at startup and is re-initialized any time the camera comes to rest after a period of movement. Visible triangles are those with at least one vertex whose normal points towards (less than 90 degrees away from) the camera. Because this visibility-testing pass is time-consuming, it is performed in the background; the complete array is used for rendering the scene during periods of camera movement (when the visible array is considered ‘dirty’) and during the reinitialization of the ‘visible’ array.

As an additional optimization, we use the nvtristrip library [17] to reorder our triangle and vertex arrays for optimal rendering performance. We could have further reduced rendering time by generating triangle strips from our triangle lists, but this would add significant computational complexity to the time-critical process of updating the surface mesh to reflect changes to the underlying voxel grid.

## 2.8 Bone Dust Simulation

We also build on the work presented in [5] to provide a simulation of bone dust accumulation, which is particularly critical in otologic procedures. Bone dust tends to accumulate in the drilling area, and must be suctioned off to enhance visibility of the bone surface.

Agus et al [5] simulate the behavior of individual particles of bone dust, sampling a subset of the particles in each rendering pass to minimize the computational load demanded by the simulation. Since individual particles of bone dust are not generally visible, it is unnecessary to simulate particulate motion. Therefore we take an Eulerian approach similar to [18], in which we discretize the working region into a three-dimensional hashed grid. Rather than tracking individual particles, we track the density of particles contained in each grid cell. This allows us to simulate the piling of dust particles, particle flow due to gravity, and particle movement due to tool contact for all accumulated bone dust, without simulating individual particles. Gravity and tool forces transfer density between neighboring grid cells, rather than modifying the velocity of individual particles.

Each grid cell containing bone dust is rendered as partially-transparent OpenGL quad, whose dimensions scale

with the density of dust contained in that cell. This provides a convincing representation of accumulated particle volume and density, and does not require that we render each particle (that is, each quantum of density) individually.

This grid-based approach significantly reduces computation and rendering time relative to a particle-based (Lagrangian) approach. Coupled with the hash table we use to minimize memory consumption for the grid, we are able to render large quantities of accumulated bone dust without impacting the interactive performance of the application. Figure 11 shows a volume of accumulated bone dust and the suction device used by the trainee to remove it. The suction device is controlled with an additional Phantom haptic interface.

## 2.9 Data-Driven Sound Synthesis

Sound is a key source of intraoperative feedback, as it provides information about drill contact and about the nature of the underlying bone. We simulate the sound of the virtual burr as a series of noisy harmonics, whose frequency modulates with applied drilling force. Building upon the harmonic-based synthesis approach presented in [6], we have recorded audio data from cutting and diamond drill burrs applied to cadaver temporal bone under a series of drilling forces, in order to determine the appropriate frequencies for synthesized sound, as well as the dependence of this data on drill type and applied drilling force. Figure 10 summarizes the spectral information collected from diamond and cutting burrs.

Sound can also be a key indicator of bone thickness intraoperatively; sound quality and frequency change significantly as the drill contacts a thin layer of bone, providing a warning that the surgeon is approaching sensitive tissue. In our simulator, the pitch of the synthesized sound increases when the drilled area becomes thin. In order to estimate the thickness of bone regions, we used a raytracing algorithm similar to that used for haptic rendering in [14]. At each voxel that is determined to be on the surface of the bone, the surface gradient is used to approximate the surface normal, and a ray is cast into the bone along this normal. The ray is traced until it emerges from the bone volume, and the thickness is estimated as the distance from the

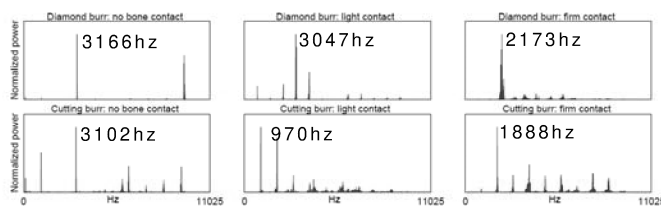


Fig 10. A spectral representation of audio data collected from diamond (top row) and cutting (bottom row) drilling burrs. Columns represent no bone contact, bone contact without significant pressure, and bone contact with a typical drilling pressure (applied by an experienced surgeon). The sharp spectral peaks and distinct variation among drill types and contact forces make this data suitable for real-time synthesis.

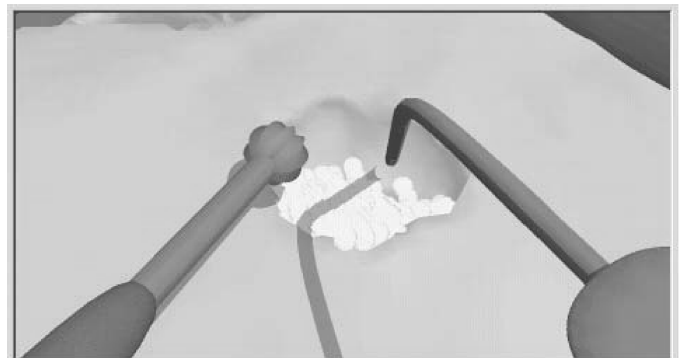


Fig 11. Bone dust simulation. The user has removed a volume of bone, which has now accumulated as bone dust. The physical simulation has allowed the bone dust to fall to the bottom of the drilled area. The user is preparing to remove the bone dust with the suction device.

ray's entry point to its exit point. For sound synthesis, this thickness is averaged over all surface voxels with which the drill is in contact. Below an empirically selected thickness threshold, sound frequency increases linearly with decreasing bone thickness. The slope of this relationship is selected so that the key harmonics span the same range of frequencies in simulation that they do in our measured data.

### 3 RESULTS: CONSTRUCT VALIDITY

#### 3.1 Experimental Procedure

The surgical simulation community defines several levels of "validity" – the ability for a simulator to mimic the real-world properties of the environment it aims to represent. The present study assesses the "construct validity" of our simulation environment: the ability to explain subject behavior in simulation with appropriate parameters describing subject experience level. In other words, expert surgeons should perform objectively better on a simulated surgical task than novices.

For the present study, fifteen right-handed participants were asked to perform a mastoidectomy (removal of a portion of the temporal bone and exposure of relevant anatomy) in our simulator. Participants included four experienced surgeons, four residents in head and neck surgery with surgical experience, and seven novices with no surgical experience.

Participants were presented with a tutorial of the simulator and were given fifteen minutes to practice using the haptic devices and the simulator's user interface. Participants were then presented with an instructional video describing the target procedure, and were given access – before and during the procedure – to still images indicating the desired appearance of the bone model at various stages in the procedure (Figure 12). Participants were asked to perform the same procedure twice.

Each participant's hand movements, haptic forces, and surgical interactions were logged to disk, then later rendered to video. Videos were scored on a scale of 1 to 5 by an experienced head and neck surgery instructor; the instructor was not aware of which videos came from which subjects and viewed them in randomized order. This global scoring approach is similar to the approach used to evaluate resident progress in a cadaver training lab. Our hypothesis is that participants with surgical experience

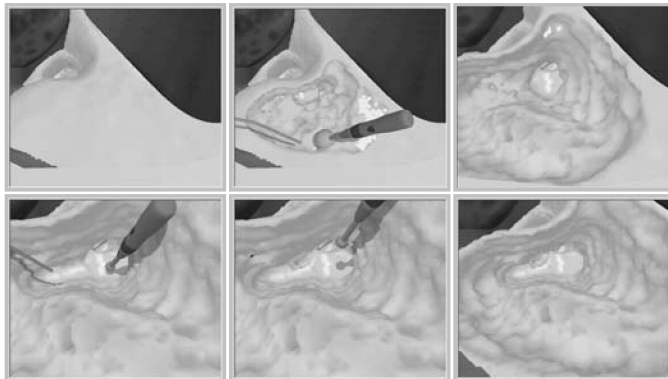


Fig 12. Still images presented to experimental participants, indicating the stages of the mastoidectomy procedure.

should receive consistently higher scores than those with no surgical experience.

Figure 13 shows a summary of the experimental results. Participants with surgical experience received a mean score of 4.06, and novices received a mean score of 2.31, a statistically significant difference according to a one-tailed t-test ( $p < 0.0001$ ). This clear difference in performance when operating in our simulator demonstrates the construct validity of the system.

### 4 NOVEL TRAINING TECHNIQUES

The previous sections of this paper discussed our simulator's approach to replicating interaction with bones, i.e. replicating the features available in a traditional cadaver-based training lab. The following section discusses our incorporation of training features that are not possible in a traditional training lab, and thus demonstrate the potential for simulation to not only replicate but also extend existing training techniques.

#### 4.1 Haptic Tutoring

Surgical training is typically focused on visual observation of experienced surgeons and verbal descriptions of proper technique; it is impossible for a surgeon to physically demonstrate the correct 'feel' of bone manipulation with physical tools. With that in mind, we have incorporated a 'haptic mentoring' module into our environment, allowing a trainee to experience forces that are the result of a remote user's interaction with the bone model.

Ideally, the trainee would experience both the movements of the instructor's tool and the force applied to/by the instructor, but it is difficult to control both the position and the force at a haptic end-effector without any control of the compliance of the user's hand. To address this issue, we bind the position of the trainee's tool to that of an instructor's tool (running on a remote machine) via a low-gain spring, and add the resulting forces to a 'playback' of the forces generated at the instructor's tool, according to:

$$F_{\text{trainee}} = K_p(P_{\text{trainee}} - P_{\text{instructor}}) + F_{\text{instructor}}$$

...where  $F_{\text{instructor}}$  and  $F_{\text{trainee}}$  are the forces applied to the

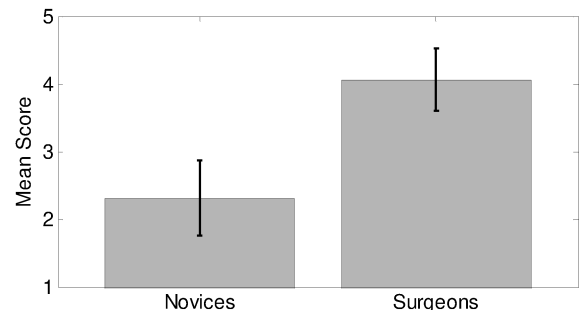


Fig 13. Mean scores for simulated mastoidectomies performed by novice participants (left) and participants with surgical experience (right). Error bars indicate 95% confidence intervals.



instructor's and trainee's tools, and  $P_{\text{instructor}}$  and  $P_{\text{trainee}}$  are the position of the instructor's and trainee's tools.  $K_p$  is small enough that it does not interfere significantly with the perception of the high-frequency components transferred from the instructor's tool to the trainee's tool, but large enough that the trainee's tool stays in the vicinity of the instructor's tool. In practice, the error in this low-gain position controller is still within reasonable visual bounds, and the trainee perceives that he is experiencing the same force and position trajectory as the instructor.

We use the same approach and the same force constants for "haptic playback", allowing a user to play back force data collected from a previous user's run through our system. This has potential value both for allowing trainees to experience the precise forces applied during a canonically correct procedure, and for allowing an instructor to experience and evaluate the precise forces generate during a trial run by a trainee.

## 4.2 Neurophysiology Console Simulation

Another goal of our simulation environment is to train the surgical skills required to avoid critical and/or sensitive structures when using potentially dangerous tools. The inferior alveolar nerve, for example, is at particular risk during most of the craniofacial procedures this environment is targeting. We thus incorporate a virtual nerve monitor that presents the user with a representation of the activity of nerve bundles in the vicinity of the procedure (Figure 14a). Nerves are currently placed explicitly for training scenarios; future work will include automatic segmentation of large nerves from image data.

This approach will also potentially contribute to the simulation-based training of a complete surgical team,

which often involves several technicians focused on neurophysiology monitoring. Simulated neural data is streamed out via Ethernet for remote monitoring, and can be visualized on a console that is similar to what would be available intraoperatively to a technician. Our system uses the visualization and analysis software distributed with the Cerebus neural recording system (CyberKinetics, Inc.) (Figure 14b).

## 5 AUTOMATED EVALUATION AND FEEDBACK

Another exciting possibility for virtual surgery is the use of simulation environments to automatically evaluate a trainee's progress and provide targeted feedback to help improve a user's surgical technique.

A straightforward approach to evaluating a trainee's performance on the simulator is determining whether a given objective has been achieved while avoiding injury to vulnerable structures (such as nerves, ossicles, or veins). However, many of the finer points of technique that surgeons learn are taught not because failure to adhere to them will *necessarily* result in injury, but because it increases the *likelihood* of injury. Therefore, it is useful to be able to quantify the risk inherent in the trainee's performance.

This section describes several metrics for evaluating a user's bone-drilling technique, and our approach to visualizing these metrics. We also present approaches to validating these metrics (confirming that they are medically meaningful) and initial validation results.

### 5.1 Visibility Testing

One of the most important ways in which risk is minimized in temporal bone surgery is by taking care to only remove bone that is within the line of sight, using a "saucerizing" drilling technique (removing bone so as to create a saucer-shaped cavity on the bone surface). This enables the surgeon to avoid vulnerable structures just below the bone surface, using subtle visual cues that indicate their locations. If instead some bone is removed by "undercutting" (drilling beneath a shelf of bone that obscures visibility), there is increased risk of structure damage.

In our environment, as each voxel of bone is removed, the simulator determines whether this voxel was visible to the user at the time of removal. Making use of the same ray-tracing techniques that are used for haptic rendering (Section 2.5), a line is traced from the removed voxel to the virtual eye point. If any voxels (other than those currently in contact with the drill) are intersected by this ray, the removed voxel is determined to be invisible.

During or after a virtual procedure, a user can visualize the visibility/invisibility of every voxel he removed, to explore the overall safety of his technique and find specific problem areas. Voxels that were visible when removed are shown in one color while those that were obscured are rendered in another color (Figure 15). The scene may also be rotated and rendered with only selected structures visible, allowing unobstructed visualization of the locations of the removed voxels and their proximities to crucial structures.

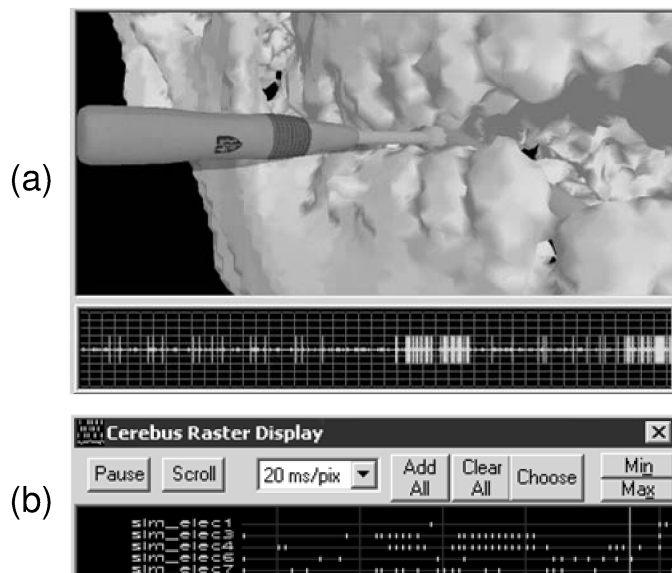


Fig 14. Virtual neurophysiology monitoring. (a) The user drills near a simulated nerve (in blue) and views a real-time simulated neural monitor, which also provides auditory feedback. (b) A remote user visualizes the activity of several simulated nerves, observing activity bursts when the user approaches the nerve structures.

Although it makes intuitive sense that voxel visibility should be an appropriate metric for evaluating a user's performance, it is important to validate this metric – and all automatic metrics – against a clinically-standard assessment of user performance. In this case, we use the data collected from the user study presented in Section 3, which includes complete simulated procedures by experts and novices, along with scores assigned to each simulated procedure by an experienced surgical instructor. A metric that correlates well to an instructor's manually-assigned scores is likely to be an effective metric for automatic user evaluation.

Figure 16 shows the results of correlating computed voxel visibilities to an instructor's score (on a scale of 1 to 5) for each simulated procedure performed by each of our study participants. Linear regression shows a correlation coefficient of 0.68, which is particularly high considering that the manual evaluation was based on a wide array of factors, only one of which was voxel visibility. This approach is suitable for assessing the effectiveness of individual metrics, which can be combined to form an overall score for a simulated procedure.

## 5.2 Learning Safe Forces

Another component of safe drilling is applying appropriate forces and operating the drill at appropriate speeds. The acceptable range of forces and speeds is closely related to the drill's distance from vulnerable structures. However, this function is difficult for a human, even an expert surgeon, to precisely quantify. Therefore, we learn maximal safe forces and speeds via statistical analysis of forces, velocities, and distances recorded during a run of the simulation by experienced surgeons. Trainees' performance can then be compared to the experts' values, and areas in which

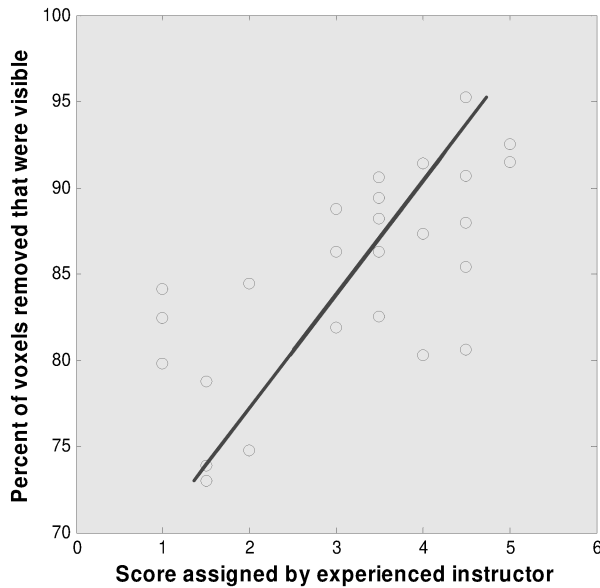


Fig 16. Relationship between expert-assigned scores (x axis) and computed voxel visibility (y-axis), along with a linear fit ( $R=0.68$ ,  $p<0.001$ ). Each dot represents one pass through the simulated procedure by one subject. The strong correlation supports the value of computed visibility as an automatic performance metric.

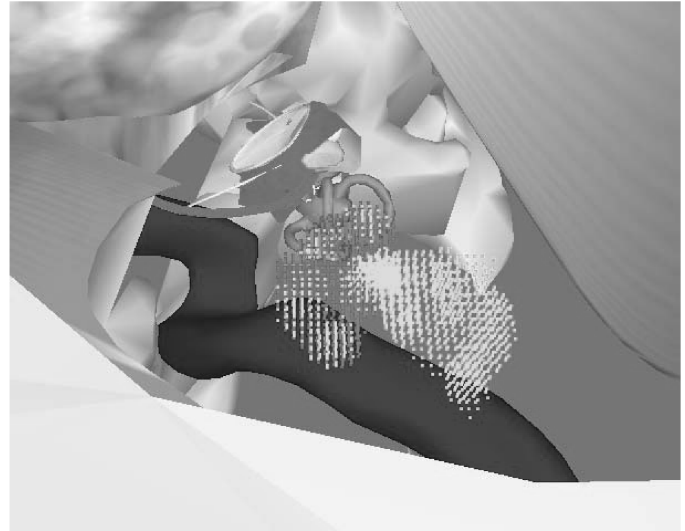


Fig 15. Visualization of removed voxel visibility. In this simulation, the trainee has correctly "saucerized" on the right side, removing only visible bone, while he "undercut" on the left side, removing bone that was hidden by other bone. This interactive visualization – in which the bone itself is not rendered – displays the regions in which he exercised proper technique (visible voxels in green) and regions in which he did not (obscured voxels in red). Undercutting in close proximity to the sigmoid sinus (in blue) was dangerous as he could not see the visual cues indicating the vein's location below the bone surface.

excessive speeds or forces were applied can be visualized and presented to the user.

For example, Figure 17 shows the force profiles of all expert and novice study participants as they approached a critical and sensitive structure, the chorda tympani, a branch of the facial nerve. At the instant that any voxel within 3cm of this structure was removed, the user's applied force was recorded. These samples were then sorted by distance from the nerve and binned into 0.2cm intervals;

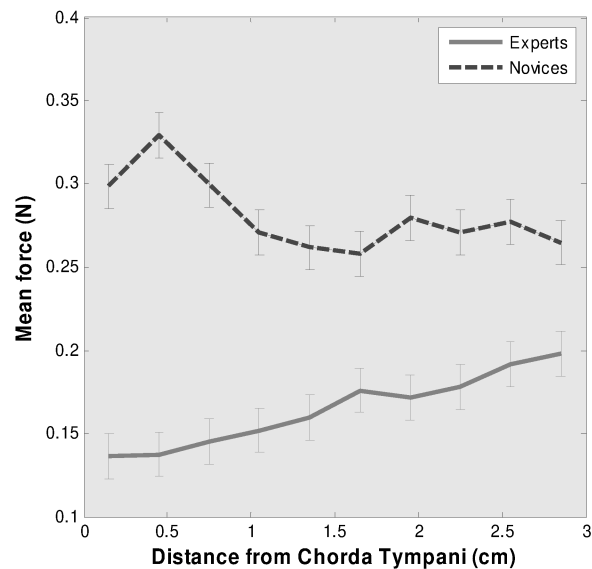


Fig 17. Forces applied by experts and novices in the vicinity of the chorda tympani (a sensitive branch of the facial nerve). Error bars indicate 95% confidence intervals. Experts display a significantly different force profile in this region than novices, as experts tend to reduce their applied forces when approaching the nerve.

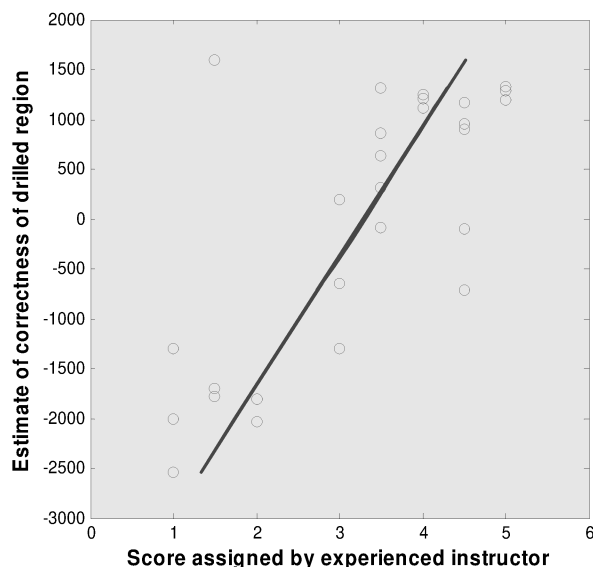


Fig 18. Relationship between expert-assigned scores (x axis) and estimate of drilled region correctness (y-axis), along with a linear fit ( $R=0.76$ ,  $p<0.001$ ). Each dot represents one pass through the simulated procedure by one subject. The strong correlation supports the validity of our drilled-region-correctness estimate as an automatic performance metric.

the mean value of each bin was computed and plotted in Figure 17. The profiles for experts and novices are significantly different, as indicated by the plotted confidence intervals. Experts clearly tend to use lower forces overall in the vicinity of this critical structure, and reduce their forces as they approach, a trend not seen in the novice plots.

### 5.3 Learning Correct Bone Regions for Removal

In addition to instantaneous metrics like force and visibility, an instructor evaluating a surgical trainee would also evaluate the overall shape of the drilled region after a complete procedure, i.e. the set of voxels removed by the trainee.

To capture this important criterion in a quantitative metric, we use a Naïve Bayes approach to categorize “correct” and “incorrect” drilling regions. We assume that voxels from the full voxel mesh are chosen for removal (drilling) according to separate distributions for experts and novices. For each voxel, we compute the probability that an expert would remove this voxel and the probability that a novice would remove this voxel. Then for each subject’s run through a simulated procedure, we look at the set of removed voxels and ask “what was the probability that an expert (or novice) performed this procedure?”, by multiplying together the probabilities of each removed voxel. We then compute the ratio of these cumulative probabilities ( $p_{\text{expert}}$  and  $p_{\text{novice}}$ ) and take the log of that ratio, to compute a scalar value that estimates the correctness of the drilled region ( $\log(p_{\text{expert}}/p_{\text{novice}})$ ).

We would like to show that this is a valid performance metric by correlating it with scores assigned by an experienced instructor, as we did in Section 5.1. Figure 18 shows the result of this analysis, along with a linear regression onto the scores assigned by an instructor ( $R=0.76$ ). Again,

the high correlation suggests that this is a valuable component in a suite of individual metrics that can produce an accurate estimate of trainee performance.

## 6 CONCLUSION AND FUTURE WORK

We have described a system for visuohaptic simulation of bone surgery, including a volume-sampling algorithm for haptic rendering and a hybrid data structure for linking visual and haptic representations of volume data. We presented empirical results evaluating the construct validity of our system, and we presented our approach to building task-level scenarios and evaluation mechanisms on top of our physical simulation.

Subsequent work on the simulation environment will focus on incorporating a representation of soft tissue simulation into our environment, to enable the representation of more complete procedures, including, for example, skin incision and tumor resection.

Subsequent work on our automated evaluation techniques will focus on the development of additional automated metrics and the visualization of automated metrics.

Supplemental material for this paper, including movies and images of the simulation environment, is available at:

<http://cs.stanford.edu/~dmorris/projects/bonesim/>

## ACKNOWLEDGMENTS

Support was provided by NIH LM07295, BioX 2DMA178, the AO Foundation, and the NDSEG and Stanford Graduate Fellowships.

## REFERENCES

- [1] P.J. Gorman, A.H. Meier, C. Rawn, and T.M. Krummel, “The future of medical education is no longer blood and guts, it is bits and bytes.” *American J Surgery*. 2000 Nov. 180(5):353-356.
- [2] R.S. Haluck, R.L. Marshall, T.M. Krummel, and M.G. Melkonian, “Are surgery training programs ready for virtual reality?” *J of the American College of Surgery*. 2001 Dec. 193(6):660-665.
- [3] K.H. Hohne, M. Bomans, M. Riemer, R. Schubert, U. Tiede, and W. Lierse, “A 3D anatomical atlas based on a volume model,” *IEEE Visualization 1992*, 12 (1992) 72-78
- [4] N.H. Blevins, R.K. Jackler, and C. Gralapp, *Temporal Bone Dissector*, Mosby, January 1998.
- [5] 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*. 2002 5(1):35-43.
- [6] J. Bryan, D. Stredney, G. Wiet, and D. Sessanna, “Virtual Temporal Bone Dissection: A Case Study,” *Proc. of IEEE Visualization 2001*, Ertl et. AL, (Eds): 497-500, October 2001.
- [7] B. Pflesser, A. Petersik, U. Tiede, K.H. Hohne, and R. Leuwer, “Volume cutting for virtual petrous bone surgery,” *Computer Aided Surgery* 2002;7(2):74-83.
- [8] M. Renz, C. Preusche, M. Potke, H.P. Kriegel, and G. Hirzinger, “Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm,” *Proc Eurohaptics*, p149-154, 2001.
- [9] E. Keeve, S. Girod, R. Kikinis, and B. Girod, “Deformable modeling of facial tissue for craniofacial surgery simulation,” *Computer Aided Surgery*, 1998, 3:228-38.
- [10] R.M. Koch, S.H.M. Roth, M.H. Gross, A.P. Zimmermann, and H.F. Sailer, “A Framework for Facial Surgery Simulation,” *Proc of the 18th Spring Conference on Computer Graphics*, p33-42, 2002.

- [11] J.G. Schmidt, G. Berti, J. Fingberg, J. Cao, and G. Wollny, "A Finite Element Based Tool Chain for the Planning and Simulation of Maxillo-Facial Surgery," Proceedings of the fourth ECCOMAS, Jyväskylä, Finland, 2004.
- [12] D. Morris, S. Girod, F. Barbagli, and K. Salisbury, "An Interactive Simulation Environment for Craniofacial Surgical Procedures," Proceedings of MMVR (Medicine Meets Virtual Reality) XIII, Long Beach, CA, January 2005.
- [13] S.F. Gibson, J. Samosky, A. Mor, C. Fyock, W. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, S. Nakajima, T. Ohkami, R. Osborne, and A. Sawada, "Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback," Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery, p.369-378, March 19-22, 1997.
- [14] A. Petersik, B. Pflesser, U. Tiede, K.H. Hohn, and R. Leuwer, "Haptic Volume Interaction with Anatomic Models at Sub-Voxel Resolution," Proc IEEE VR, Orlando, FL, Mar 2002.
- [15] D.J. Bouvier, "Double-Time Cubes: A Fast 3D Surface Construction Algorithm for Volume Visualization," Int'l Conf on Imaging Science, Systems, and Technology, June 1997.
- [16] T.H. Massie and J.K. Salisbury, "The PHANTOM Haptic Interface: A Device for Probing Virtual Objects," Symp. on Haptic Interfaces for Virtual Environments. Chicago, IL, Nov. 1994.
- [17] NVIDIA Corporation. nvtristrip library. February 2004. <http://developer.nvidia.com/>.
- [18] J. Stam, "Real-Time Fluid Dynamics for Games". Proceedings of the Game Developer Conference, March 2003.
- [19] Voxel-Man TempoSurg, Spiggle and Theis GmbH. <http://www.voxel-man.de/simulator/temposurg/>
- [20] M. Agus, G.J. Brelstaff, A. Giachetti, E. Gobbetti, G. Zanetti, A. Zorcolo, B. Picasso, and S.S. Franceschini. "Physics-based burr haptic simulation: tuning and evaluation". Proceedings of the 12<sup>th</sup> IEEE Haptics Symposium, pp. 128-135, March 2004.



**Dan Morris** received his MS from Stanford in Computer Science in 2003. He is currently pursuing his PhD in Computer Science at Stanford. His research interests include interactive physical simulation, haptics, surgical simulation, and neural prosthetics.



**Christopher Sewell** received his MS from Stanford in Computer Science in 2004. He is currently pursuing his PhD in Computer Science at Stanford. His research interests include surgical simulation, haptics, and machine learning.



**Federico Barbagli** received his PhD from Scuola Superiore S. Anna in robotics and control. He is currently a research fellow in the Robotics Laboratory at Stanford. His research interests include haptic rendering algorithms, haptic control, and haptic device design.



**Nikolas Blevins** received his MD from Harvard and completed his residency in Otolaryngology at the University of California at San Francisco. He is currently an Assistant Professor in the Department of Otolaryngology at Stanford. His research interests include surgical simulation, cochlear microendoscopy, microrobotics, and innovations in surgical education and preoperative planning.



**Sabine Girod** received her MD from Hanover Medical School in Hanover, Germany, and completed her residency in Oral Surgery at Hanover as well. She received her PhD from the University of Cologne in Cologne, Germany, and is currently an assistant professor in the Division of Plastic Surgery at Stanford. Her major clinical and research interests are functional oral and craniofacial rehabilitation of craniofacial trauma and craniofacial deformities, bone grafting, implantology and oral pathology in children and adults.



**Kenneth Salisbury** received his PhD from Stanford in Mechanical Engineering in 1982, and is currently a professor in the Computer Science and Surgery departments at Stanford. His research interests include human-machine interaction, collaborative computer-mediated haptics, and surgical simulation. He has served on the US National Science Foundation's Advisory Council for Robotics and Human Augmentation and as scientific advisor to Intuitive Surgical.