Interacting with 3-Dimensional Medical Data
Haptic Feedback for Surgical Simulation

Andrew B. Mor[1,2], Sarah Gibson[2], Joseph T. Samosky[3]
abm@ri.cmu.edu gibson@merl.com jsamosky@mit.edu

ABSTRACT

This paper describes different methods tested to haptically explore voxel-based data acquired through medical scanners. The goal of this project is a multi-modal virtual reality surgical simulator. This simulator will allow surgeons and surgical residents to practice and rehearse surgical procedures using patient-specific data. Several methods for calculating the displayed force were investigated and are presented. Additional medical dataset operations, graphical interfaces, and implementation issues are also presented.

I. INTRODUCTION

Through a collaboration between MERL, Brigham and Women's Hospital, MIT, and CMU, we are developing a volume-based surgical simulation system to accomplish three tasks: assist in the training of surgical residents, help established surgeons prepare for procedures by rehearsing the surgery using actual patient data, and provide an aid to navigation which can be used in the operating theater during a procedure. To achieve this goal, this collaboration is focusing on 3 different areas: deformable object simulation, real-time volume rendering, and voxel- based haptic simulation. The Phantom, a haptic interface developed by Sensable Technologies, Inc., is currently being used for force display.

Three-dimensional, voxel-based data is the natural medium for performing surgical simulation because it is the native format of scanned medical data, including MRI and CT. Also, these methods will allow us to model the complex interior structures present in human anatomy, which is critically important for deformable tissue simulation. In contrast, surface based models only approximate surfaces in the medical data and cannot accurately incorporate internal structure. By interacting directly with the medical data, instead of a polygonal model fit to surfaces in the data, the Phantom can more accurately represent the underlying structure of the patient.

While many researchers have implemented systems utilizing the Phantom, only a small number have attempted to interact with three dimensional, voxel-based data. Avila and Sobierajski at General Electric Corporate Research and Development were the first to utilize voxelized data in a haptic simulation. This voxel-based data presents a novel problem for determining object boundaries and forces to display to the user due to the relative sparseness of the data with respect to the resolution of the Phantom. Additionally, when dealing with segmented voxel data, which consists of a binary classification map, additional filtering methods are required to simulate a smooth surface.

In our current work, three methods for calculating the displayed force were investigated: interpolating between stored normal vectors at each voxel location; trilinear interpolation of smoothed intensities at the vertices with calculation of the gradient to get direction; and factoring the raw binary data with a "Gaussian sphere" around the Phantom position, again using the gradient to calculate the normal direction. In this work, three types of data were utilized: raw binary segmentation data; filtered versions of the above raw data; and subsampled voxel data built from geometric primitives.

II. DATASET ACQUISITION, PREPROCESSING, AND VISUALIZATION

The initial dataset acquired for this project was a T-1 weighted proton density MRI image of a normal male knee. The dataset size was 256x256x124 with a voxel size of 0.625 x 0.625 x 0.9mm. This dataset contains eight million voxels, while a higher resolution dataset currently being worked on contains 24 million voxels. The slices were then hand segmented into the major anatomical structures: bone (femur, tibia, fibula, patella), cartilage (femoral, tibial, and patellar), lateral and medial menisci, and anterior and posterior cruciate ligaments. Hand segmentation was performed due to the difficulties presented to automatic segmentation techniques by MRI data. Unlike CT images, where image intensity is directly related to density, MRI intensity is based on physical parameters which can vary greatly within one structure but sometimes minutely between neighboring bodies.

---

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA
[2] MERL - A Mitsubishi Electric Research Lab, Cambridge, MA
[3] Massachusetts Institute of Technology, Cambridge, MA

When viewing the bony structures acquired from the segmented dataset, the surface of the bones do not appear smoothly segmented, as shown in Figure 1. In fact, when the dataset is viewed in a direction orthogonal to the original slice plane, it is clear that large inter-slice errors, due to segmentation being performed on a per-slice basis, are present. These types of errors will be present in almost any segmented dataset, due to the difficulties inherent in segmentation, whether performed by hand or by automatic and semi-automatic techniques. Therefore, it was necessary to filter the dataset to achieve a surface that can be displayed haptically to the user. Without smoothing, a small change in the Phantom position could cause erratic behavior in the direction of the displayed force. Batch filtering was performed using a Gaussian filter in the frequency domain. A filtered version of the previous image is shown in Figure 2. The bumpiness present in the original image is still present, but is now surrounded by a smooth gradient of values. Note that this filtering is not required for datasets that are built from geometric primitives and are inherently smooth.[1]

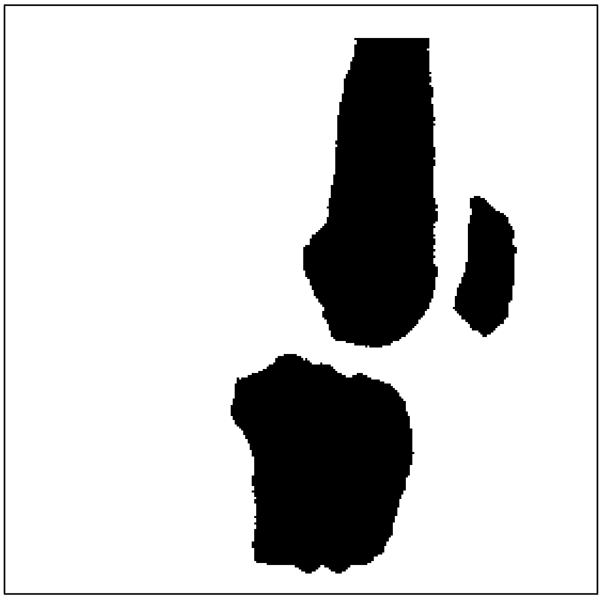

Figure 1 Raw segmentation data

Two methods for visually displaying the dataset were utilized. The first method employs fast volume rendering on a multiprocessor SGI Challenge.[2] This method utilized the 3-D texture memory and implemented the shear-warp algorithm to display depth.[3] The second method employs a flexible 3D sectional visualization system (SectionView) to display the slice of data and where in that slice the user is currently pointing with the Phantom.[4] The two methods are complementary in that the volume rendering presents a global view of the user's position in the dataset, while the section method presents the local view of exactly what the user is interacting with. If the original MRI dataset is loaded and displayed using the sectional method, the system allows the user to see the original available data around what she is feeling, providing a natural interface to explore an area of interest. To maximize the refresh rate on the haptic interface, the graphical interfaces were run on a separate machine with position data broadcast over an Ethernet network.

Figure 2 Filtered segmentation data

III. METHODS FOR FORCE DISPLAY

As mentioned above, three methods of calculating the displayed force vector were investigated. The different methods trade-off speed of execution and storage size of the voxel-based models. The dataset contains eight million voxels, where each voxel can contain multiple datapoints relating to parameters of the voxel.

The first method that was implemented interpolated between stored normal vectors at each voxel center. Utilizing models generated from geometric primitives, each voxel contained both a magnitude and a normal direction. To minimize storage size, the normal direction was stored as three bytes, a byte for the component in each direction. An inverse distance metric was used to interpolate the displayed force from the intensities and normal directions at each vertex location. While this method is fast, it required at least three times the amount of storage compared to using the intensity alone. Additionally, this method tended to flatten out curves that closely paralleled the axis directions, due mainly to the discretization of the normal vector when stored as a triplet of bytes.

The second method involved placing a “Gaussian sphere” around where the user is pointing and performing local smoothing, followed by application of a gradient operator to calculate the displayed force. This “Gaussian sphere” is just a simple Gaussian filter extended to three dimensions, so that the scaling factor decreases with radial distance from its center. By performing the smoothing at run-time instead of by preprocessing, the number of occupied voxels is minimized, an important consideration for a later part of our research involving deformable, voxel-based objects. This method operates by precomputing the Gaussian coefficients for a range of squared distances. For each update cycle, a 5x5x5 area around the Phantom tip is factored and summed to calculate the current intensity, as in the following equation. In this equation:

$$I_{xyz} = \sum_{i=-2}^{2}\sum_{j=-2}^{2}\sum_{k=-2}^{2} \frac{1}{\left(2\pi\sigma^{2}\right)^{3/2}} e^{-\frac{d^{2}}{2\sigma^{2}}} I_{[x]+i,\,[y]+j,\,[z]+k}$$

I_{xyz} is the intensity being calculated; the brackets around the indices i, j, and k imply rounding to the nearest integer, and therefore the nearest voxel location; and d^2 is the distance from the $([x]+i, [y]+j, [z]+k)$ voxel position to the location of the Phantom tip. The smoothing is then repeated at points around the current position to calculate the normal direction. While this method holds great promise for future work and research, it currently does not operate smoothly and displays a very choppy force to the user.

The last method currently being investigated and researched utilizes trilinear interpolation to calculate intensities between voxel centers and then central differences to determine the local surface normal. Trilinear interpolation is an extension of simple interpolation in 1D to three dimensions and is guaranteed to be continuous. As in Figure 3, to calculate the intensity at a point within a cube comprised of the eight surrounding vertices, interpolate first between the vertices along common edges in the x-direction. Then, using the four values returned from that interpolation, interpolate in the y-direction. The two values calculated from these last interpolations are then used to interpolate to the value at the point of interest. Trilinear interpolation is also used to calculate the intensity at six surrounding points to determine the normal direction using central differences. This method does require preprocessing of the data to create a smooth region containing a ramp of values between free space and the object. If the smoothed region is not sufficiently wide, the effective gain in the local region can be great enough to cause unstable behavior. This method convincingly displays the forces associated with the voxel-based data, and provides the cornerstone for a surgical simulator.

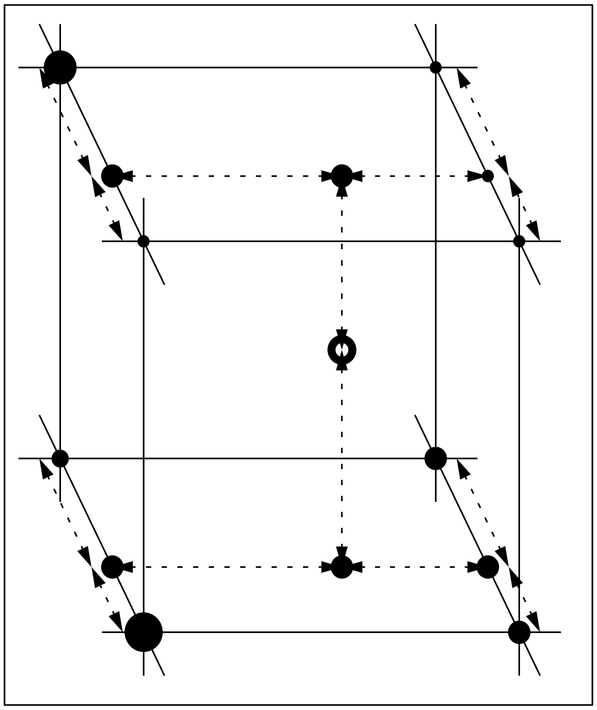

Figure 3 Trilinear interpolation

IV. IMPLEMENTATION ISSUES

As with many digital hardware systems, difficulties arise due to problems with the discretization of time. For instance, in a continuous time system, velocity is just the instantaneous derivative of position. But, in the Phantom system, the change in position from one update cycle to the next does not look anything like the actual velocity due to two factors, the discretization of time and the discretization of space due to the encoders. Therefore, velocity of the Phantom tip was calculated as a moving average over the previous 0.01 seconds, using a ring buffer to store the change in position and the elapsed time for each update cycle. Viscosity was implemented in the usual manner, as a force acting opposite to the velocity vector. Viscosity was applied whenever the user penetrated the surface of an object. This could occur because the direction of the displayed force was generated from the local gradient. When the user penetrates past the smoothly varying intensities on the surface of the object, the gradient goes to zero because all the local voxels possess the maximum intensity. A pseudo-friction model was used, where a viscosity force was applied when moving along the surface of the object. The magnitude of the viscosity was less than or equal to a preset maximum value, to simulate Coulomb sliding friction.

Object rotations were implemented utilizing a quaternion to rotation matrix transform. The quaternion was calculated approximately ten times a second, and the object was "rotated" by rotating the user position in the opposite direction. In this way, the user position is transformed, and then just indexed into the voxel space in the usual manner. These rotation matrices are also sent over the network to the graphical interface to provide feedback as to how the object is rotating.

As was mentioned above, the binary segmentation data was filtered to provide smooth gradient values to the trilinear interpolation. Unfortunately, this filtering did cause some "buzzing" in a couple of locations in the dataset. When two bones were sufficiently close together in the segmentation map, the filtered intensity values between them would be artificially high. For instance, when a byte is used to store the intensity, values as high as 40 were seen between the fibula and the tibia and also between the femur and the tibia. These high intensities in the free space between the bones were large enough to cause unstable behavior of the Phantom. One view of this behavior is to think of the Phantom tip as bouncing between the two sides of a valley, where the sides are formed by the bones in the model, with the bouncing caused by overshoot of the tip. Because of this "buzzing," new filtering techniques, such as morphological filters which will not increase the size of the object, are being investigated.

These methods were all implemented on a Silicon Graphics Indigo2 Extreme equipped with a R4400 processor running at 250MHz. The optimized version of the trilinear interpolation method updated at approximately 4800Hz, well above the accepted minimum of 1000Hz.[5] This method was also ported to a 200MHz Pentium Pro PC running Windows95. On this machine, using the Microsoft Visual C++ compiler, optimized code ran at 6500Hz, a 35% improvement.

V. CONCLUSION

This research provides a solid groundwork toward voxel-based modeling and interaction for surgical simulation. A fast haptic display method, trilinear interpolation with a local gradient operator, for interacting with static voxel data was implemented and tested. This knee model was shown to a local orthopedic surgeon, who provided helpful feedback. More user input will be garnered during the next stage of development of the overall surgical simulator. Three dimensional voxel-based simulators like this one, because they utilize raw segmentation data instead of surfaces that approximate the data, minimize the approximations needed to accurately display medical data and are more appropriate for surgical simulation and training.

VI. ACKNOWLEDGMENTS

The authors would like to thank the MELCO Wellness Business Group for providing the financial support for this collaboration and the Mitsubishi Electric Research Laboratory for providing equipment and resources necessary for this research.

We would also like to thank Dr. Scott Martin of Brigham and Women's Hospital for his input on procedures and his feedback on the feel of the simulator, Gordon Douglass for his support with the computing facilities, and Dr. Shin Nakajima of Brigham and Women's Hospital and Akira Sawada of MELCO for their work in hand-segmenting the knee dataset.

REFERENCES

[1] Avila, Ricardo and Sobierajski, Lisa, personal communication, Summer 1996.

[2] Fraser, Robert, "Interactive Volume Rendering using Advanced Graphics Architectures," Whitepaper, http://www.sgi.com/Technology/volume/VolumeRendering.html

[3] Lacroute, Philippe G., "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," Stanford Technical Report CSL-TR-95-678, September 1995.

[4] Samosky, Joseph T., "SectionView--A System for Interactively Specifying and Visualizing Sections Through Three-Dimensional Medical Image Data," M.S. Thesis, Massachusetts Institute of Technology, 1993.

[5] Srinivasan, Mandayam A., "Haptic Interfaces," in Virtual Reality: Scientific and Technical Challenges, eds. Durlach, N.I. and Mavor, A.S., National Research Council, National Academy Press, 1994.

[6] Gibson, S., Samosky, J., Mor, A., Fyock, C., Grimson, E., Kanade, T., Kikinis, R., Lauer, H., McKenzie, N., Nakajima, S., Ohkami, H., Osborne, R., Sawada, A., "Simulating Arthroscopic Knee Surgery using Volumetric Object Representations, Real-Time Volume Rendering and Haptic Feedback," MERL Technical Report #TR96-19, 1996.