1 Large-scale volume rendering and manipulation

This is hard to address individually; as the issue cross-cuts all of the other issues we deal with, *all* of our efforts consider the problems that large data pose, and how we can solve those in an effective, packaged system. In this section and the ones that follow in the visualization TRD chapter, we will expound on the issues of combining multiple potentially-large volumes, rendering transparent meshes in concert with direct volume rendering techniques, collaborative visualization tools and their application, distributed rendering techniques, and the exploitation of a novel parallel architecture. All of the solutions mentioned herein are in some way related to the architecture powering ImageVis3D, which is now formally published [5].

1.1 Volume Combinations

A common problem experienced in the medical imaging community is combining data from multiple imaging modalities. For example, MRI data can provide relevant information on the tissue in question, but denser structures of interest, such as bone, might provide useful context information and only be available from CT imaging techniques. Researchers want to combine these data sets into a single volume for further visualization and analysis.

In this cycle, we have added preliminary support for combining volumes in ImageVis3D via user-configurable ways. This allows users to combine multiple imaging modalities, as in Figure 1, or apply novel transformations on their data. One use case this enables is applying a segmentation to a data set. Figure 1 depicts this use case: a data set of a human head is combined with a segmentation to produce just the brain component of the scan.

1.2 Meshing

A second combinatorial issue which arises in medical research is inserting meshing data into other visualizations, such as the volume renderings produced by ImageVis3D. From a technical standpoint, this becomes very difficult when both visualizations are partially transparent. We have produced a novel method for solving this problem and deployed it in the renderer powering ImageVis3D, Tuvok.

2 Collaboration

Even while parallel supercomputing resources grow ever-larger, mobile compute devices simultaneously decrease in size and increase in capabilities. With the recent explosion of netbooks and tablet computing devices, as well as the cell phones researchers already carry daily, the mostly-untapped potential of mobile visualization has become one of the fastest growing opportunities for enabling visualization in the hands of those who can utilize it best.

In collaboration with our colleagues at the University of Saarbrücken, we have deployed 'ImageVis3D Mobile'. This mobile visualization application is designed to run on iOS devices, including the Apple iPad and iPhone, and perform functions similar to the 'desktop' version of ImageVis3D already deployed by the SCI institute. Data can be pushed to mobile devices in a manner guided by an imaging specialist, without requiring that practicing physicians understand or realize the

Figure 1: Volume combinations applied to a head data set. The original imaging modality is combined with a generated segmentation to produce the imaging data of the brain alone.

complicated data acquisition and preparation phases which might be necessary for the data.

As a free, publicly available tool, ImageVis3D Mobile has already been utilized in domains as far-reaching as nuclear engineering education [6]. However, our interests lie in the medical application of this mobile tool. In collaboration with Chris Butson of the Department of Neurology, Medical College of Wisconsin, we present recent research results – for one of our driving biological problems – with a deep brain stimulation visualization application used in neurology[3].

2.1 Medical Visualization on Mobile Computing Platforms

In recent years, there has been significant growth in the use of patient-specific models to predict the effects of neuromodulation therapies such as deep brain stimulation (DBS) [2]. However, translating these models from a research environment to the everyday clinical workflow has been a challenge, primarily due to the complexity of the models and specialized software required to provide the visualization. Here we describe the use of ImageVis3D Mobile in an evaluation environment. It was employed to visualize models of four Parkinson's patients who received DBS therapy. Selection of DBS settings is a significant clinical challenge that often requires repeated revisions to achieve optimal therapeutic response, and it is often performed without the advantage of a visual representation of the stimulation system in the patient. We used ImageVis3D Mobile to provide models to movement disorders clinicians and asked them to use the software to determine 1) which of the four electrode contacts they would select for therapy and 2) what stimulation settings they would choose.

We used ImageVis3D Mobile to provide models to movement disorders clinicians (Figure 2) and asked them to use the software to determine 1) which of the four electrode contacts they would select for therapy and 2) what stimulation settings they would choose. We compared the stimulation protocol chosen from the software versus the stimulation protocol that was chosen via clinical practice (independently of the study). Lastly, we compared the amount of time required to reach

Figure 2: ImageVis3D Mobile displaying the results from multiple patients in a deep brain stimulation study.

these settings using the software versus the time required through standard practice. We found that the stimulation settings chosen using ImageVis3D Mobile were similar to those used in standard of care, but were selected in considerably less time. On average, the standard of care required 4±1.4 hours for programming simulation settings, whereas clinicians could perform the same task in 1.7±0.8 *minutes* using the ImageVis3D Mobile-based system. ImageVis3D Mobile is an example of how a visualization system, available directly at the point of care on a device familiar to the neurologist, can be used to improve critical clinical decision making [3].

2.2 Distributed Rendering

As part of our efforts to expand the ability to generate high-quality visualizations at the location in which they are most useful, we are also investigating distributed and hybrid rendering systems. With the initial launch of ImageVis3D Mobile, one of the problems that quickly became apparent is the lack of enough compute power to display extremely large data at native resolutions: such operations would take minutes, hours, or even days on the compute power available to mobile devices, clearly too slow to be usable.

We have implemented remote rendering for ImageVis3D Mobile, in which a more powerful compute system renders at the request of the mobile device, and sends images over the network to the device. As we have successfully scaled our volume rendering system out to virtually unlimited resolutions – as detailed in Section 4 – this allows us to render extremely high resolution data on a mobile device.

The unfortunate consequence is that the system becomes unusable with unreliable network connections. In the next cycle, we will explore a hybrid rendering system, which combines the rendering capabilities of mobile devices with the higher-powered resources of workstation- and cluster-based systems.

2.3 Web-based Rendering

One recent development relevant to the field of *ad hoc* visualization has been the standardization and proliferation of WebGL-enabled browsers. These allow rendering of complex three-dimensional geometry and volumes in a standard web browser. We have already received requests for volume renderers which can be embedded in a web browser.

Though this was not foreseen and thus not proposed initially, the CIBC is responsive to

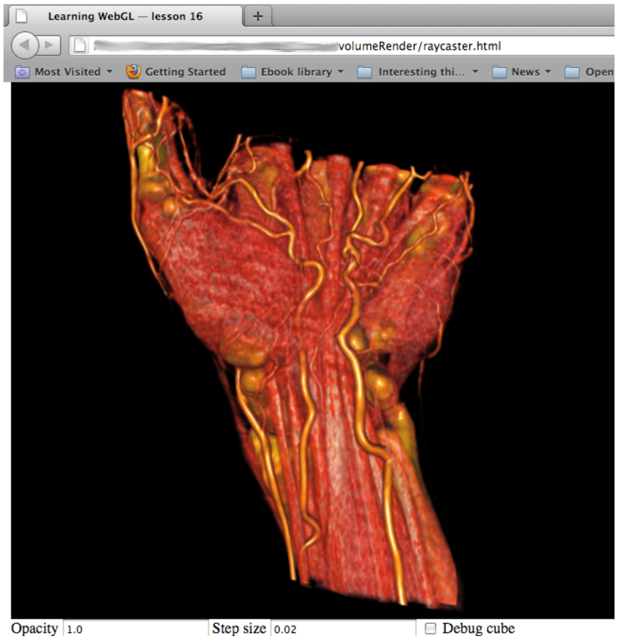

Figure 3: Volume rendering system running in a browser.

the needs of the medical community. We are exploring browser-based rendering based on these technologies. Figure 3 shows an example of a volume rendering system running within a web browser.

Due to technical limitations, current systems have limited volume rendering applicability due to size restrictions on the rendered volumes. However we hope that the next round of standardization will solve these issues, and in the meantime we are exploring alternative methods for working around these limitations.

3 Visualization on Unstructured/Curvilinear Grids

We have not made strides in this area during this cycle.

4 Parallel Architectures

The size of data is growing rapidly, driven largely by the acquisition of new scanning and simulation technologies which can deliver larger data more rapidly. This presents difficulties for many visualization systems, which should remain interactive to avoid decelerating the process of medical and scientific insight.

Hardware systems are continuously improving to help us keep pace with these larger data sizes, but the application of new hardware systems to existing problems is not always straightforward. In particular, modern hardware solutions are delivering increased performance via parallelism instead of clock speeds, presenting the challenging problem of exploiting that parallelism to application-level software [1].

CIBC personnel are always searching for novel ways to exploit current hardware to perform existing tasks more quickly, as well as to increase the size of problems the medical community can tackle. One of the somewhat recent additions to modern architectures is the so-called "GPU cluster", a distributed memory system in which each node has one or more graphics processing units (GPUs) attached. The SCI Institute, as an NVIDIA CUDA center of excellence, has one such GPU cluster. Other such systems exist throughout the U.S., such as the Texas Advanced Computing Center's 'Longhorn', a teragrid resource used for a wide variety of computationally challenging problems. Software which can take advantage of these GPUs can accelerate rendering or even general purpose computations by orders of magnitude.

We have conducted field-leading research evaluating the effectiveness of multi-GPU cluster resources (clusters in which each node contains multiple GPUs) on a popular med-

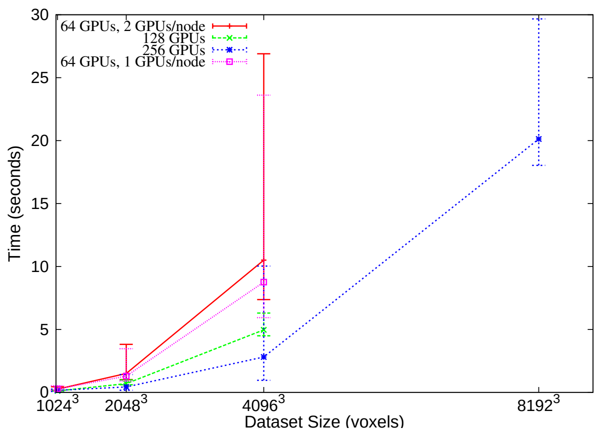

Figure 4: Overall rendering time when rendering to a 1024x768 viewport on a modern GPU cluster. Scalability is shown for up to 256 GPUs on data sets up to 8192^3 voxels.

ical visualization technique, volume rendering [4]. Effective use of these clusters requires applying multi-scale parallelization strategies, on the GPU for fine scale parallelism and across nodes for macro-scale parallelism. Figure 4 shows the primary result of this work: scalability out to large numbers of GPUs. Current systems allow for interactive performance for data sizes of 2048^3 voxels and smaller. We have also ascribed numbers to the known issue of non-power-of-two texture use on GPUs. The community previously expected lower performance from such textures, but it turns out that average performance is equivalent and the performance *variance* is what differs in this case. Results such as these help to inform us as we design systems which can scale out to the largest of data sets, as well as perform adequately on the future commodity workstations most personnel in the medical research community will possess.

5 Provenance Enabled Tools

After an initial integration between VisTrails and ImageVis3D, as well as external work integrating Tuvok (ImageVis3D's rendering core) in other contexts, it became apparent to us that both provenance and self-contained use would be better served by a command-style interface to the library. We are presently engaged in a small restructuring effort to support both of these use cases more naturally.

References

[1] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick. The landscape of parallel computing research: A view from berkeley. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, Dec 2006.

[2] C. R. Butson, S. E. Cooper, J. M. Henderson, and C. C. McIntyre. Patient-specific analysis of the volume of tissue activated during deep brian stimulation. *NeuroImage*, 34(2):661–70, 2007. `http://linkinghub.elsevier.com/retrieve/pii/S1053811906009669`.

[3] C. R. Butson, G. Tamm, S. G. Jain, T. Fogal, and J. Krüger. Evaluation of medical visualization on mobile computing platforms for clinical decision making. In *Proceedings of IEEE Visualization*, page (submitted), 2011.

[4] T. Fogal, H. Childs, S. Shankar, J. Krüger, R. D. Bergeron, and

P. Hatcher. Large Data Visualization on Distributed Memory Multi-GPU Clusters. In *Proceedings of High Performance Graphics 2010*, pages 57–66, June 2010.

[5] T. Fogal and J. Krüger. Tuvok, an Architecture for Large Scale Volume Rendering. In *Proceedings of the 15th International Workshop on Vision, Modeling, and Visualization*, pages 57–66, June 2010.

[6] T. Jevremovic, T. Fogal, D.-O. Choe, H. Yang, and J. Krüger. The role of virtual engineering and emerging visualization tools in nuclear engineering education and training at the university of utah. In *Proceedings of the Nuclear Engineering Science and Technology Conference on Education and Training*, 2011.