Interactive 3D Medical Visualization: A Parallel Approach to Surface Rendering 3D Medical Data

Terry S. Yoo and David T. Chen
Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599-3175

I. INTRODUCTION

Using Pixel-Planes 5, a parallel multicomputer for computer graphics [Fuchs 90], we have constructed a system for visualizing volumetric medical data based on polygonal approaches to surface rendering. The goal is a medical visualization system that has intuitive navigation and exploration capabilities to present 3D clinical data using three dimensional images. To provide a natural navigation of patient data, segmentation parameters should provide user feedback at minimum rates of one update per second, and viewing direction and lighting control should respond in tenths of seconds. Our approach differs from other methods under investigation at UNC [Yoo 92] as we take advantage of the hardware graphics accelerators for polygon rendering rather than attempt direct volume rendering.

II. BACKGROUND

Fuchs and his colleagues developed an approach for generating surface models from 3D slice-based medical data. Their method uses extracted contours of anatomical elements on individual slices. The contours are subsequently stacked, and a polygonal model is generated by interpolating surfaces between the segmented curves [Fuchs 77]. More recently, Lorensen and Cline circumvented the intermediate contour generation stage using an algorithm called *"Marching Cubes"* that generates polygonal representations of the anatomy directly from the segmented volume data [Lorensen 87][Cline 88].

There is a continuing debate over the relative merits of surface extraction as a presentation method versus direct volume rendering. In particular, *marching cubes* suffers from an algorithmic flaw which leads to mathematical inconsistencies, violating the "Jordan Property" [Kong 92]. That is, the algorithm does not guarantee that the generated polygonal representations are closed orientable surfaces (that they partition space into sets of 'inside' and 'outside'). Other investigations comparing surface and volume rendering have based their findings upon the expected memory requirements and computational load imposed by the various methods [Udupa 91].

We contend that developments in parallel computer architecture enabling fast computation and rendering have removed many of the obstacles to effective volume visualization through surface rendering. In particular, <u>interactive</u> generation of the

Published in Boehme Johannes M., Alan H. Rowberg, and Neil T. Wolfman, *eds.*, *Computer Applications to Assist Radiology,* (Proceedings of S/CAR 94, Symposium for Computer Assisted Radiology, Winston-Salem, NC USA, 12-15 June 1994), Symposia Foundation, 1994, pp. 100-105.

polygonal representation of anatomical surfaces coupled with real time control of the display overcome many of the visualization problems in surface rendering techniques. Visualization flaws arising from the non-oriented surfaces generated by *marching cubes* may be easily overcome by interactively modifying the segmentation parameters and checking the persistence of anomalous features through different segmentations.

Moreover, approaches to volume visualization based upon polygons as primitives take better advantage of today's graphics accelerators. Systems capable of processing in excess of 200 thousand shaded triangles per second (Sun Microsystems Leo) to a half million shaded triangles per second (SGI Extreme Graphics) are now commercially available.

III. DISPLAY ARCHITECTURE

Our development platform, Pixel-Planes 5, is a heterogeneous graphics architecture using both MIMD (multiple instruction multiple data path) and SIMD (single instruction multiple data path) parallelism. This machine has multiple i860-based Graphics Processors (GPs), and multiple SIMD pixel-processor arrays called Renderers. Each Renderer is a 128x128 array of pixel processors capable of executing a general purpose instruction set. GPs send Renderers opcode streams which are executed in SIMD fashion. The GPs, Renderers, several Frame Buffers, and a workstation host communicate over an eight-channel ring-network whose bandwidth is 80 MB per channel (aggregate bandwidth of 5 gigabits per second).

IV. IMPLEMENTATION

We parallelized the *marching cubes* algorithm, optimizing the implementation for Pixel-Planes 5. Questions examined in this research include how to subdivide the data among multiprocessors and how to accelerate the surface extraction given the subdivision. Data must be distributed so that the computational load is balanced the among the processors. Figure 1 shows a schematic of the data and command flow for *marching cubes* on Pixel-Planes 5. The MIMD section is responsible for the construction of the surface models, performing geometric viewing and lighting transformations, and finally invoking the SIMD Renderer units, sending opcode streams to render the polygon primitives of the model. Figure 2 shows a description of the system implemented on the individual graphics processors. The sequence of processing steps include: dataset distribution, voxel gradient estimation, interactive user segmentation and generation of the surface model, geometric viewing transformation, and distributed rendering.

Dataset-distribution: Typically, X-ray CT and MRI data have significantly higher resolution in two of the major axis dimensions (x and y), and are fairly sparse in the third (z) dimension. After some consideration, we elected to preserve the orientation of this innate coordinate system of the medical data, and distribute the data as sets of X-Y slices. Overlapping slice sets, four contiguous slices each, are distributed in a round-robin fashion among the available graphics processors.

Gradient estimation: The data in its initial distribution is replicated four times throughout the processors. A method of central differences is used to estimate the gradient vectors at each voxel location. After this calculation is complete, two of the four data slices may be discarded. This step results in estimated normals for rendering smooth surfaces and reduces the data replication by fifty percent.

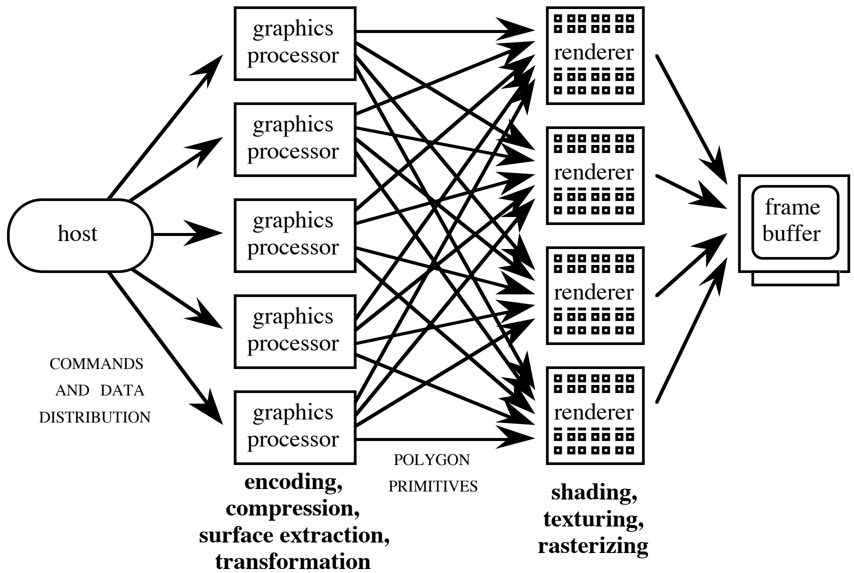

Figure 1. - Schematic of the Pixel-Planes 5 *marching cubes* algorithm

Interactive segmentation (isosurface selection): User controlled inputs supplied via the host interface are broadcast among the graphics processors. Upon receipt of new threshold information the processors use the *marching cubes* edge intersection calculations and polygon lookup tables. A table based approach, though prone to interpolation error proves to be one of the fastest mechanisms for computing polygonal surfaces.

Interactive viewing control and distributed rendering: Parallel rendering is accomplished using the existing graphics software infrastructure. Polygon primitives are transformed to screen space coordinates, and the resulting polygon primitives are distributed among the Renderers for rasterization to one or more frame buffers. Pixel-Planes 5 is capable of sustained frame rates of 20 frames per second and polygon rates exceeding 2 million smooth (Phong) shaded triangles per second.

V. OPTIMIZATIONS

The slice based data distribution suggests raster based encoding for data compression. Run length encoding provides both a mechanism for memory optimization and a means of traversing intervals between isosurface boundaries. Run-length encoding accelerates the intersection calculations for edges along the compressed row. The isosurfaces will not intersect edges between voxels of the same value. So the edges that are between voxels of similar value may be skipped. No such acceleration is attempted for edges across rows or between slices.

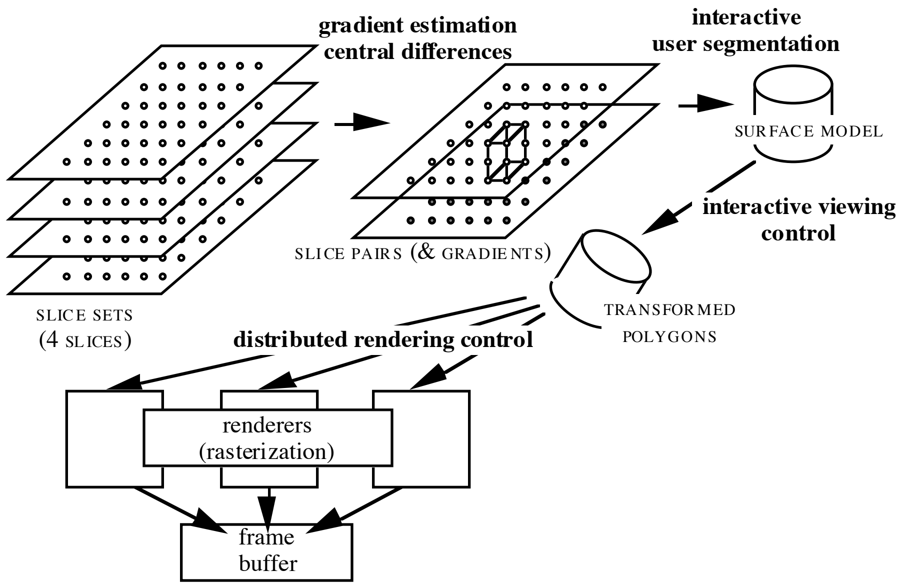

Figure 2. - The parallel *marching cubes* pipeline

VI. RESULTS

Using the data distribution and control flow described in the previous sections, we constructed the system and measured its performance on several different data sets. The Pixel-Planes 5 configuration that we used included 36 graphics processors and 16 Renderers which represents approximately 60% of the maximum configuration.

Figures 3 through 5 show a CT pelvic study (128x128x56 voxels) displaying between 40,000 and 110,000 triangles at frame rates of between 8 and 20 frames per second (depending on the complexity of the representation). The system can perform 2 to 3 isosurface calculations per second of this data. Figure 6 is a rendering of volume ultrasound data (128x64x96 voxels) of the face of a human fetus in utero. The surface model was generated in 357 milliseconds. The surface is rendered at 871 thousand polygons per second (12 frames per second). Figure 7 is an MR study of a human female (resampled to 96x96x109 voxels). The surface model was generated in under 6-tenths of a second and contains 269720 triangles. Because of the complexity, the view update rate is limited to 3 frames per second.

VII. RELATED WORK

Alternate approaches to parallelizing volume visualization through surface rendering have been developed in conjunction with different architectures. Notably, significant effort has been extended in the development of SIMD algorithms for fine-grain massively parallel surface extraction [Hansen 92][Song 93]. Alternate image analysis encoding may be employed if the volume is not subdivided. Wilhelms and van Gelder explored octtree based sorting and searching algorithms to accelerate the generation of surface models [Wilhelms 90].

Finally, simplifying the surface representation is a straightforward approach to increasing the speed of rendering complex surfaces. Graphics and visualization literature contain several algorithms for simplifying polygonal models [Schroeder 92][Turk 92][Hinker 93]. Each of these methods are hindered by the distributed nature of volume data in our implementation and subsequently were not incorporated.

VIII. INTERACTIVE HEAD-MOUNTED DISPLAYS

Fast polygonal rendering enables the technology of virtual reality along with its many applications in medicine [Bajura 92]. The emphasis of this work has been to provide fast, natural, interactive navigation of 3D medical data. We have implemented a prototype VR interactive medical display (Figure 8). The future may lie in more personal presentation of volume data through VR. The applications and opportunities to the field of computer aided medicine are manifold, and are yet to be explored.

ACKNOWLEDGMENTS

This research is partially supported by NIH MIP grant PO1CA47982, NSF cooperative agreement ASC-8920219, and ARPA ISTO contract DABT63-92-C-0048. Data is courtesy of UNC Department of Radiation Oncology and the Tomtec Corporation.

REFERENCES

Bajura, M., H. Fuchs and R. Ohbuchi, "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient," *Computer Graphics*, (Proc. SIGGRAPH 92), 26(2), (July), 1992, 203-210.

Cline, H. E., et. al. "Two Algorithms for the Three-Dimensional Reconstruction of Tomograms," *Medical Physics*, 15(3), (May/June), 1988, 320-327.

Fuchs, H., Z. M. Kedem and S. P. Uselton, "Optimal Surface Reconstruction from Planar Contours," *Communications of the ACM,* 20(10), (Oct.), 1977, 693-702.

Fuchs, H., et. al., "Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories," *Computer Graphics*, (Proc. SIGGRAPH 89), 23(3), (July), 1989, 79-88.

Hansen, C. D., and P. Hinker, "Massively Parallel Isosurface Extraction," *Proc. Viz. 92*, IEEE CS Press, Los Alamitos, CA., 1992, 77-83

Hinker, P. and C. Hansen, "Geometric Optimization," *Proc. Viz. 93*, IEEE CS Press, Los Alamitos, CA., 1993, pp. 189-195

Kong, T. Y. and J. K. Udupa, "A Justification of a Fast Surface Tracking Algorithm," *CVGIP: Graphical Models and Image Processing*, 54(2), (Mar.), 1992, 162-170.

Lorensen, W. E. and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, (Proc. SIGGRAPH 87) 21(4), (July), 1987, 163-169.

Schroeder, W. J., J. A. Zarge, and W. E. Lorensen, "Decimation of Triangle Meshes," *Computer Graphics*, (Proc. SIGGRAPH 92), 26(2), (July), 1992, 65-70.

Song, D. and E. Golin, "Fine-Grain Visualization Algorithms in Dataflow Environments," *Proc. Viz. 93*, IEEE CS Press, Los Alamitos, CA, 1993, 126-133

Udupa, J. K., H. M. Hung, and K. S. Chuang, "Surface and Volume Rendering in Three-Dimensional Imaging: A Comparison," *J. of Digital Imaging*, 4(3), (Aug.), 1991, 159-168.

Wilhelms, J. and A. V. Gelder, "Octrees for Faster Isosurface Generation," *Proc. San Diego Workshop on Volume Viz.*, (Nov.), 1990, 57-62.
Yoo, T., *et.al.,* "Direct Visualization of Volume Data," *IEEE Computer Graphics & Applications,* 12(2), (July), 1992, 63-71.

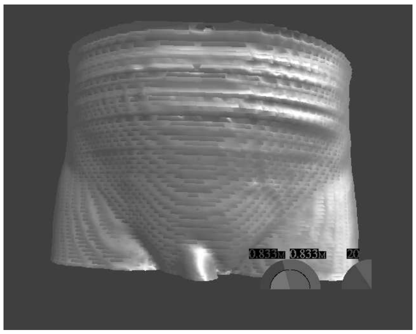

Figure 3. - X-ray CT Pelvic Study (128x128x56 voxels). Skin surface (41708 triangles) generated in 409 msec. View update rate 20 frames/sec (fps)

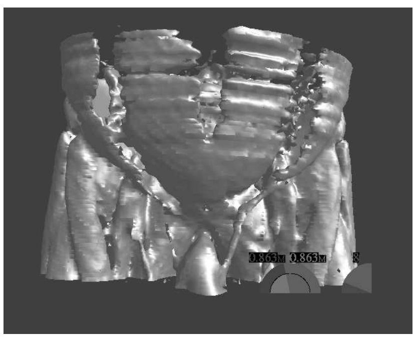

Figure 4. same study as Figure 3. Muscle surface (110450 triangles) generated in 676 msec. View update rate 8 fps

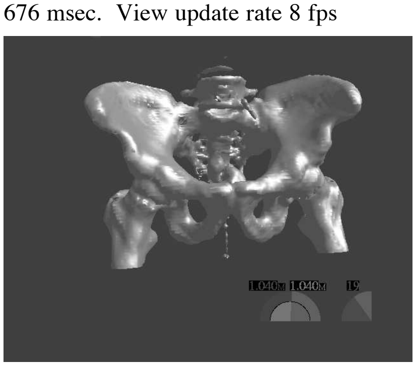

Figure 5. - same study as Figure 3. Bone surface (49552 triangles) generated in 409 msec. View update rate 19 fps

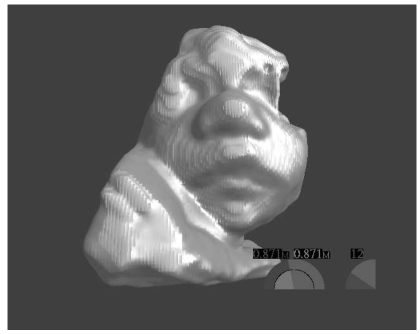

Figure 6. - Volume ultrasound of human fetus in utero (128x64x96 voxels). Face (70874 triangles) generated in 357 msec. View update rate 12 fps

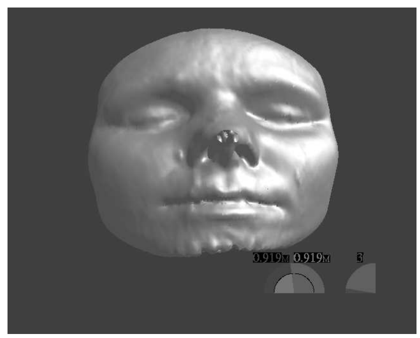

Figure 7. - MR head study (96x96x109 voxels). Skin surface (269720 triangles) generated in 591 msec. Update rate 3 fps

Figure 8. - David Chen pictured using a head mounted display. Stereo surface renderings are presented in a virtual environment for medical visualization.