

HPMVS: A High Performance Visualization Tool Suite that Assists in Kidney Assessment

Timothy S. Newman and Ning Tang

Department of Computer Science, University of Alabama in Huntsville, Huntsville, USA

This paper introduces an interactive volume visualization tool suite, the **High Performance Medical Visualization tool Suite (HPMVS)**. The suite of tools is designed to aid medical staff in the assessment of renal disorders such as those caused by the von Hippel Lindau (VHL) Syndrome. The tools are useful for image-based evaluation of the severity and progression of disease and for planning and monitoring treatment. The configuration of HPMVS can provide near real-time visualization by allowing highly intensive computations to be computed on a supercomputer and less intensive computations and final display to be realized on a desktop workstation. Focus in this paper is on the exploration and extraction tools and the tool set configuration.

Keywords: volume visualization, interactive exploration, high performance computing, imaging, segmentation, anatomic localization

1. Introduction

Traditionally, volumetric medical data (especially, computer-assisted tomography (CAT) and magnetic resonance imaging (MRI) data) has been studied by medical staff by examining a fixed series of printed slice images. That mode of display does not allow direct 3D exploration of the data; the data cannot be fully exploited for diagnostic or treatment planning and monitoring uses. Medical staff using these images must construct mental spatial models of a patient's anatomy to diagnose disease or to plan intervention. The High Performance Medical Visualization tool Suite (HPMVS) is aimed at providing volume visualization tools that allow medical staff to more effectively utilize CAT data for assessing cystic renal disorders (particularly the complications of von Hippel Lindau

(VHL) Syndrome) and for planning and monitoring treatment of the disorder. The HPMVS tools can provide near-real-time 3D rendering and volumetric exploration.

A number of medical visualization and imaging investigations have been discussed in the literature. For example, efforts for the modeling, extraction (segmentation), or display of anatomic structures, such as the heart [Park et al., 1995, Solaiyappan et al., 1996, Weng et al., 1997], aorta [Rueckert et al., 1997], upper torso structures [Brown et al., 1997], and abdominal organs [Kobashi & Shapiro, 1995, Newman et al., 1996], have been described. A method for muscle fiber orientation extraction, interpolation, and display has also been described recently [Sachse et al., 1998]. In addition, radiological therapy planning techniques that utilize computing and information technology have been presented (e.g., [Kessler et al., 1994]). A visualization system with virtual reality capability that could eventually enable virtual rehearsal of the operative process has also been described [Robb et al., 1996].

Improved computer hardware performance and software maturity have increased the utility of visualization in medical environments. However, key portions of processing are quite computationally intensive, making near-real-time performance on the desktop a challenge. Advanced computational devices (e.g., supercomputers) may offer sufficient computational power for near-real-time performance in principle, but these machines often have less flexible graphic

environments than the desktop computers. Previously, Robertson et al. [Robertson et al., 1991] have suggested that a networked visualization system could be employed to combine the computational advantage of high-performance computing with the flexibility of workstation or PC desktop environments. The HPMVS framework is aimed at coupling high performance computing capability with desktop interactive environments in the context of a high-speed network, such as the configuration proposed by Robertson et al.

1.1. Target Application

Patients with VHL Syndrome tend to develop cysts and tumors in and on the kidneys. These diseased tissues can be removed surgically from the kidneys. However, due to renal system stress caused by such interventions, it is important for the health of the kidney that any surgery be completed within rigid time constraints. In addition, a patient should be subjected to very few interventions. Therefore, careful planning for surgery by preoperative determination of the spatial extent and relative position of diseased tissue can aid medical staff in planning a surgical intervention that can be completed in a minimum amount of time with minimal impact on kidney health. Traditionally, medical staff have planned for surgery by manual study of individual cross-sectional images of CAT data.

HPMVS enables improved diagnosis and intervention planning by more thorough exploitation of image data. For example, HPMVS includes special-purpose tools that aid in determining tumor size and location. These tools enable medical staff to assess renal health and determine when surgery is necessary. (One test of the necessity of treatment is if any tumors exceed a critical volume or extent [Neumann, 1998], and the tumor segmentation and quantification tools of HPMVS are two important components in aiding diagnosis.) HPMVS also includes data exploration tools that enable interactive exploration of the data to determine severity of individual tumors. In addition, HPMVS has computationally-efficient integrated rendering tools that support 3D display of tumors and nearby organs and other structures of interest. Such tools enable surgical staff to plan

surgery and then carry it out with greater efficiency, resulting in less organ stress.

In this paper, the architecture of HPMVS and its key new exploration tools are presented.

2. System Architecture

In this section of the paper, the functional units and network configuration of the suite of tools are described.

A programming environment that is oriented toward visualization applications is often a key enabling factor in the development of effective complex visualization tools and systems. The current generation of modular visualization environments provides visual programming techniques that are layered on top of graphics application programmer's interfaces (APIs), such as OpenGL. These environments satisfy the needs in developing complex visualization tool sets, including support of fast prototyping, modular coding, and easy construction of graphical user interfaces. Cameron's paper [Cameron, 1995] outlines these characteristics in more detail. AVS, DataExplorer, IrisExplorer, and Khoros are some of the well-known modular visualization programming environments. Visualization function libraries such as vtk [Schroeder et al., 1998] have also been developed and can aid in visualization tool implementation. However, vtk does not offer a visual programming environment. The HPMVS is built as a layer atop AVS.

AVS

AVS, like other modular visualization environments, provides a high-level visual-programming framework for development of interactive scientific visualization applications. This framework includes image processing, visualization, viewing, importing/exporting, and other functionality. Applications can be built using the built-in functions as building blocks. However, AVS also supports an extension mechanism which allows development of new modules that support added or different functionality.

The visual programming interface allows the AVS developer to build the application on-screen

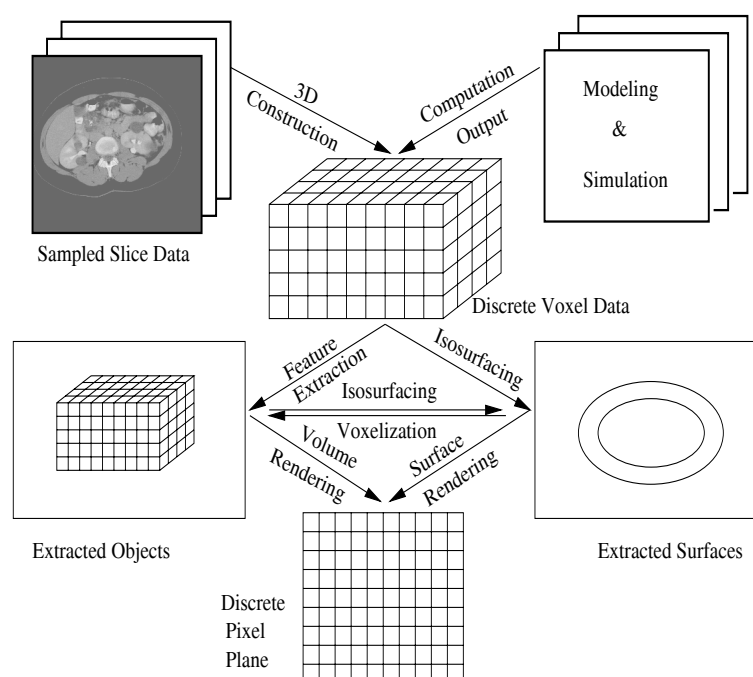


Fig. 1. The general data flow in volume visualization (adapted from Kaufman [Kaufman et al., 1991])

using a mouse to position and wire-together the basic (or extended) building blocks. The data, once imported, flows from module to module in the network. Typically, the flow terminates at a display-class of module which displays the final rendering produced by the series of steps. The AVS supports rapid prototyping and modularity through its visual programming interface with the developer.

End users typically use the developer's AVS application through a customized end-user interface designed by the developer. AVS applications have the advantage of ease of use for end users and ease of development (of powerful applications) for the visualization developer.

2.1. Functional Units

The various visualization functions of AVS (or any other visualization environment) exhibit quite diverse computational requirements. For example, some functions involve time-consuming computation. Effective usage of other methods might require user-friendly interactivity. The approach in HPMVS was to identify and distinguish the requirements, thereby allowing functions to be implemented and optimized on the most appropriate computing platform.

Visualization processing can be decomposed using the data flow illustrated in Figure 1. This diagram is in the spirit of Kaufman's diagram in [Kaufman et al., 1991]. Visualization techniques can be classified according to their primary processing role, such as feature extraction, rendering, and interactive data exploration. Our tool suite provides interactive tools within each of these functional areas. The components for feature extraction and rendering—which include Marching Cubes, surface ray tracing, and volume ray casting tools in HPMVS—typically are computationally intensive. HPMVS has optimized these components for high performance on a supercomputer (slower serial versions of these components that execute directly on a front-end workstation have also been realized). Other tools (for example, the data interaction components) require extensive user interaction, a very flexible graphical capability, and lack high computational requirements, and therefore have been realized on the workstation platform. A diagram of the decomposition of the tool suite is shown in Figure 2.

The tools in the data exploration and manipulation functional unit allow clinical staff to explore tomographic data and to collect statistical information about objects of interest. These tools supply meaningful information and assist the user in guiding the visualization pro-

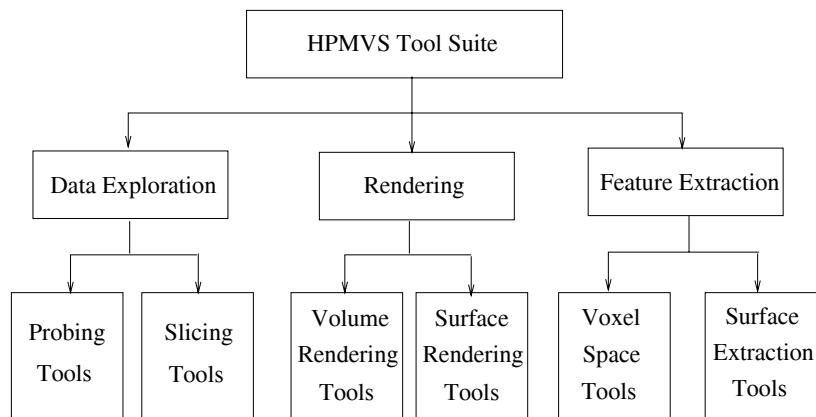


Fig. 2. Toolset Functional Model

cess. The feature extraction tools are especially important because they allow the location and characteristics of structures of interest to be discovered in the tomographic data, thereby aiding in the understanding of disease and the planning and monitoring of treatment. The feature extraction tools output 3D surface-, voxel-, or image-based primitives. HPMVS also includes surface, voxel, and image rendering tools to create displayable images from these primitives. Details about individual tools are presented in Section 3 of this paper.

2.2. Network Configuration

As shown in Figure 3, the tool suite is implemented on a workstation that is connected by a high-speed (10 Mbps) network to a supercomputer. The supercomputer to which we had access was a Cray C94A and the workstation was an SGI Indigo 2 Extreme. The workstation is equipped with a 150 MHz R4400 CPU and 128 MB of main memory. It is used for end display and most computations. However, certain feature extraction and rendering tools require access to a more powerful computational engine

to support the goal of interactive processing on large volume datasets and these tools have been realized on the C94.

The C94 has CPUs that feature efficient vector-parallel computation. Vector-parallel processing can be categorized as a subclass of SIMD (i.e., Single Instruction Multiple Data) computation. The available C94 has two CPUs, each with a peak performance of approximately 1 GFLOP and each with two vector processing pipelines. The clock period is 4.2 nsec on the C94, although the CPU has 8 vector registers that can each hold up to 64 64-bit elements, with up to 64 results produced by a single vector operation. The machine available to us has 64 MWords (192 MB) of main memory with memory access bandwidth of 1200 MB/sec. It also has a 256 MB solid state storage device with access bandwidth of 1000 MB/sec and a 50 GB disk subsystem.

We did not have exclusive access to the C94; it is a shared resource among a number of organizations. Therefore, while we have been able to benchmark the performance capability of the Cray realizations of the tools, physical near-real-time performance cannot usually be

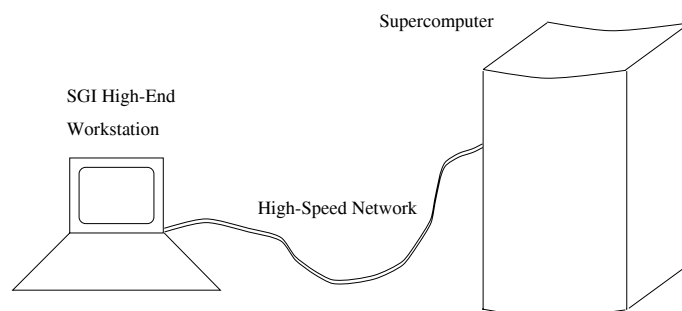


Fig. 3. Networked visualization environment

realized due to contention with other users. Our studies indicate that the configuration can support near-real-time computation when exclusive access to the Cray can be guaranteed, however. Details of how several of the rendering and feature extraction methods can be efficiently vector-parallelized are presented in [Newman et al., 2000].

A number of other SIMD-parallel approaches to visualization and rendering have been reported in the literature. For example, a data-parallel SIMD approach for ray-casting has been reported by Hsu [Hsu, 1993]. An isosurface extraction approach tuned for good SIMD performance has been demonstrated by Hansen and Hinker [Hansen & Hinker, 1992]. A SIMD-efficient polygon clipping method has been recently proposed by Schneider and van Welzen [Schneider et al., 1998]. A systolic array architecture for efficient object-order volume ray-casting has been described by Ogata et al. [Ogata et al., 1998]. Renaud and Rouselle [Renaud & Rouselle, 1997] have developed a method for rendering indoor scenes using a massively parallel approach to progressive radiosity calculation. However, in general SIMD has received much less attention for visualization and rendering than has MIMD-style parallelism. Lacroute [Lacroute, 1995] and Drebin et al. [Drebin et al., 1988] have noted some of the reasons that many have tended to overlook SIMD for visualization and/or rendering. Yet, with the recent introduction of SIMD instructions in the current generation of widely-available microprocessors, there is an opportunity to exploit SIMD to deliver better visualization and rendering performance. The HPMVS's use of SIMD offers insight into how efficient performance could be achieved using current and forthcoming commodity microprocessors.

The survey paper by Crockett [Crockett, 1997] contains extensive information on parallel rendering techniques, and the reader is directed to that paper for additional information on parallel rendering.

The design and implementation of a networked visualization tool set is more complex than implementation on a single computing platform. Design considerations involve functional decomposition of tasks and careful evaluation of the types and complexities of the computations.

Coupling these considerations with the characteristics of the computer platforms guides the selection of the appropriate platform for each task.

3. Visualization Tools

In this section, details of the functional units of HPMVS are described. Sample renderings and other outputs of the tool suite are also exhibited. Focus is on the tools that extend the basic functionality of the environment in which HPMVS is realized. Basic functionality, such as capabilities to view individual orthogonal slice images, perform contrast enhancement and color map manipulations, etc., is provided through adaptation of modules supplied directly by AVS and is not described in detail in this paper.

3.1. Data Exploration

HPMVS includes two major data exploration tools in addition to standard functionality supplied by AVS (e.g., orthogonal slice image display). The first of these tools is the slicing explorer.

Slicing Explorer

As described earlier in the paper, the conventional approach in using tomographic datasets is to view the data without aid of a computer as a series of individual images. This basic viewing approach is extended by modules of AVS and other modular visualization environments that allow viewing of data slices that are perpendicular to the coordinate system axes. The HPMVS slicing explorer tool further extends slice viewing by allowing the user to freely select the orientation of the slicing plane through the data (i.e., a slicing plane of any orientation can be selected). The tool resamples the volume dataset and displays the data intersected by the selected plane as a slice image, \hat{S} . The slicing explorer tool also provides the capability to simultaneously display images of two orthogonal cutting planes, \hat{S}_a and \hat{S}_b , that are perpendicular to one another and to the selected slicing plane ($\hat{S} \perp \hat{S}_a \perp \hat{S}_b \perp \hat{S}$). The cutting plane \hat{S} 's position and orientation can be selected and adjusted using mouse clicks and drags. The tool

allows the user to select the position of \hat{S}_a and \hat{S}_b with the mouse by choosing the desired point of intersection of \hat{S}_a and \hat{S}_b with \hat{S} . Clinical staff benefit greatly from being able to select and view an arbitrary plane and the planes orthogonal to it. The slicing explorer improves user understanding of the spatial relationships of organs and other structures of interest; approach planes to cysts, tumors, and other abnormalities can be explored in an interactive manner with the tool. Simultaneous display of the orthogonal planes helps medical staff gain a quicker understanding of the extent (severity) of disease and also helps staff determine more quickly if an abnormality exceeds a critical size, thus requiring surgical removal.

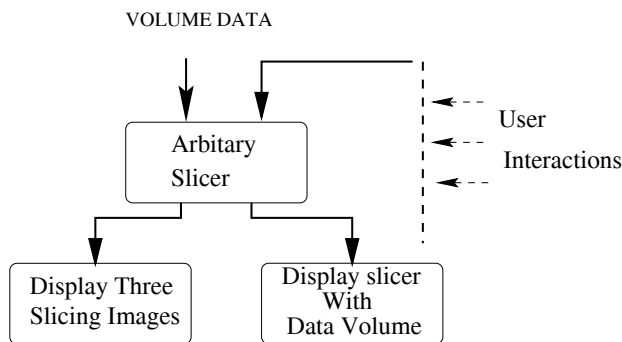


Fig. 4. Slicing explorer functional diagram

Figure 4 diagrams the functional principle of the slicing explorer tool. Based on the feedback information of a user's interactions, the tool displays the location and extents of the slicing plane and the two planes perpendicular to the slicing plane in a window on screen. The slice images are computed and displayed

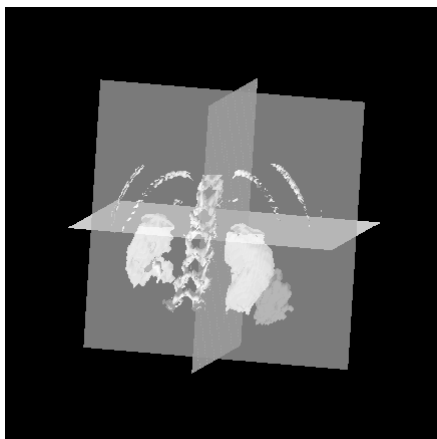


Fig. 5. Slicing planes overlaid on anatomic structures extracted from dataset

in another window. The tool can be used in a low level mode to display images from the raw dataset or in a higher level mode to display slice planes through structures extracted by HPMVS feature extraction tools.

Figure 5 demonstrates one of the windows created by the slicing explorer. The figure shows application of the slicing exploration tool on structures extracted from an abdominal CAT dataset. (Anatomic structure segmentation is provided by an HPMVS feature extraction tool (discussed in Section 3.3).) The three orthogonal slice planes selected by the user are displayed as translucent spatial overlays on the structures of interest to the user. In Figure 6, the cross-sectional images corresponding to the selected orthogonal slice planes are displayed. The images are labeled I, II, and III in the figure. Image I corresponds to the vertical plane parallel to the page in Figure 5, Image II corresponds to the other vertical plane, and Image III corresponds to the nearly horizontal plane orthogonal to the page.

The slicing explorer is implemented directly on the workstation as its computations can be performed on the SGI in a fraction of a second.

Statistical Probing

A second HPMVS tool for data exploration is the statistical probe tool. The statistical probe allows interactive determination of statistical information about structures of interest. This tool outputs quantitative measurements, such as the data value at selected voxels and the density,

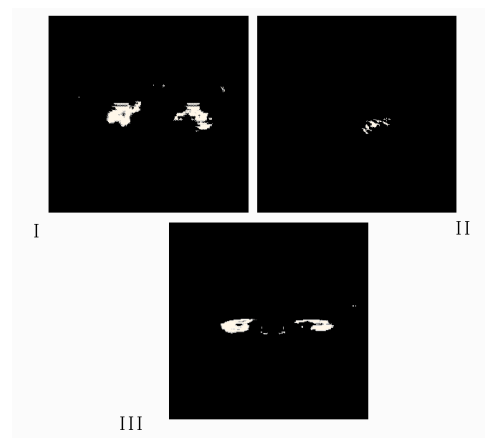


Fig. 6. Three orthogonal slicing images for kidney. Planar images are labeled I, II, III.

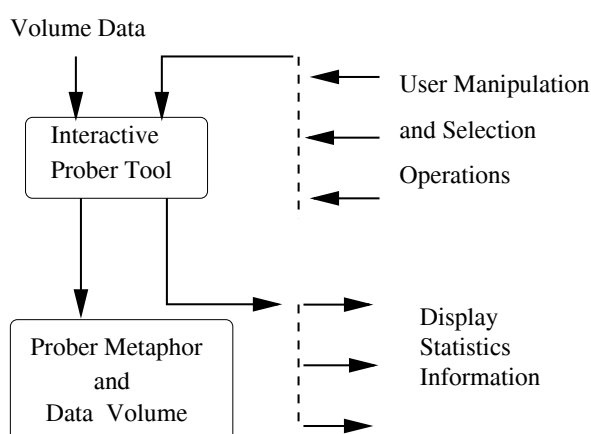


Fig. 7. Statistical probe functional diagram

mean, and variation of an organ or other structure of interest. The statistical probe enables medical staff to assess severity of tumor abnormalities as well as assess functioning of organs. The assessment can be in a static mode (i.e., in comparison against reference standards) or in a dynamic mode (i.e., in comparison with prior assessment of the patient).

As illustrated in Figures 7 and 8, the probe tool supplies a 3D spherical icon that shows the focus of measurement in 3D space. The user is able to interactively manipulate the probe icon to any position in the 3D space of the data. Mouse manipulation information is fed back to the probe tool and the new position of the probe is calculated based on this feed-back information. 3D positioning is supported with key/button selection in combination with mouse movement; holding down the right mouse button while moving the mouse indicates left-right and up-down movement of the probe within the current depth plane, whereas holding down the shift key while moving the mouse indicates forward-backward movement of the probe. The probe then updates the image and the corresponding statistical information. The most useful statistics for the kidney application are location, volume, mean, and variance of an anatomical structure, and the tool computes these statistics for any structure selected by the user.

The interactive statistical probe supplies a fast visual means to extract desired information from the dataset; it provides a simplified and enhanced method for measurement, eliminating the need for laborious manual counting. Figure 9 illustrates use of the probe on structures extracted using the HPMVS segmentation tool

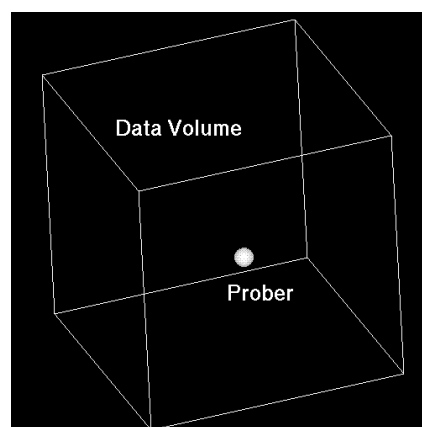


Fig. 8. Data volume and the probe metaphor

and an HPMVS extraction tool (these tools are discussed in Section 3.3). All of the structures visible in this figure (spine, ribs, kidneys, etc.) can be selected by the probe. A companion text window displays the computed statistical measures for the selected structures. Information such as precise position, sizes, and composition of cysts and organs are especially useful in planning and assessing treatment.

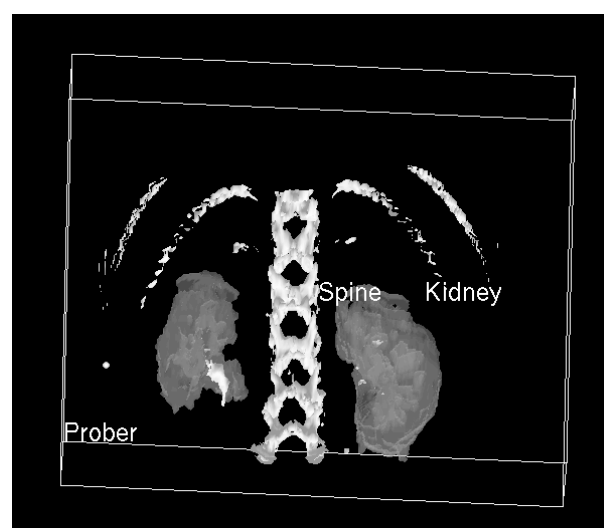


Fig. 9. Probing in abdominal CT data volumes

The statistical probing tool's computational requirements are moderate, and real-time interactive performance is provided by the implementation of the probing tool on the SGI workstation.

3.2. Rendering Tools

The HPMVS tools include a number of rendering functions. Components of these tools were realized by assembling functionality directly available in the AVS environment. For example, the tool suite allows the bounding surfaces of extracted structures (e.g., the kidneys or kidney cysts) to be displayed using many shading schemes, including wireframe rendering, flat shading, Gouraud shading [Gouraud, 1971], etc. The 3D renderings can also be rotated by the user to allow study from multiple view points.

The HPMVS rendering tools are especially useful in planning interventions which need to be completed in a minimal amount of time. Surgical staff can pre-plan the location of cuts and cutting directions using the rendering tools. Use of the supercomputer for the most computationally-intensive rendering enables better interactivity, which is of practical importance for users (who may lack the patience to use information technology tools which cannot deliver reasonable performance). The rendering tools are also useful for training purposes; students and young staff benefit from being able to view highly realistic images which closely match the actual anatomical shape.

One HPMVS rendering tool involves Cray C94-based computation with end display on the SGI. This tool is the custom shading tool that provides surface ray-traced renderings. Execution on the supercomputer enables greatly improved performance. The surface ray-tracing [Whitted, 1980] is well-known to allow the synthesis of photo-realistic images by simulating opti-

cal effects including shadowing, reflection, and transparency [Foley, 1996]. The surface ray-tracing can enable synthesis of more realistic images than is possible using well-known interpolative surface shading techniques such as Phong [Phong, 1975] or Gouraud [Gouraud, 1971] shading, although at a higher computational cost. This tool is especially useful for training applications where realism has a high utility.

Figure 10 shows the ray-tracing model with view, objects, and light sources. The view includes a view point and a view plane (i.e., image plane) composed of pixels at regular grid points. Each ray is traced from the view point, through a pixel in the view plane, and into the scene where it may intersect an object surface and then be reflected from the object to other objects. The Phong optical model (the Phong model is described by Foley et al. [Foley, 1996]) is applied to determine intensity.

Additional details about how the surface ray-tracing was efficiently parallelized are contained in the paper [Newman et al., 2000]. Surface ray-tracing is invariably time-consuming. However, the tool is able to compute renderings approximately 7 times faster on the C94 than on the front-end workstation. Near-real-time performance is still not possible for complex geometries, though (for example, scenes with approximately 9100 surfaces can be rendered in 175 seconds using the C94). A surface ray-tracing of the spine and ribs isosurfaces (originally extracted from a $256 \times 256 \times 37$ CAT dataset) is shown in Fig. 11.

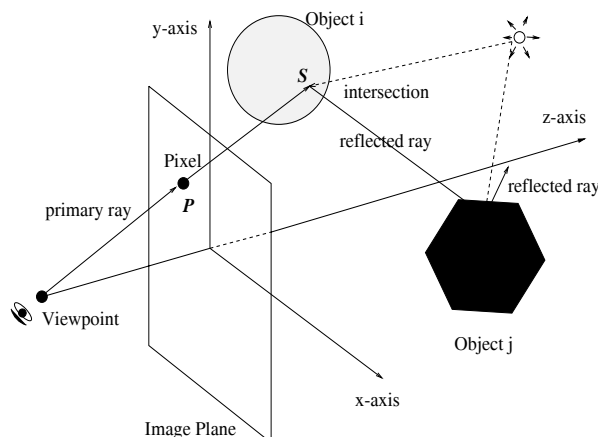


Fig. 10. Surface ray-tracing principle



Fig. 11 Surface ray-traced spine and ribs

3.3. Feature Extraction Tools

In this section of the paper, two feature extraction tools and an anatomical structure segmentation tool are discussed. The computational requirements of the two feature extraction tools generally prevents the front-end workstation from computing the extractions in near-real-time. However, near-real-time performance can be achieved for most datasets using the HPMVS feature extraction tools' realization on the C94 supercomputer. For information technology tools to have an impact in the clinical setting, responsiveness of the tools is critical. The information technology can enable more complete extraction of image data, hence improving diagnosis and treatment accuracy, but without a reasonable responsiveness of the tools, many users will fail to utilize the tools. Thus, the two feature extraction tools have been implemented on the supercomputer; the extractions are computed on the supercomputer and displayed on the SGI workstation.

Isosurface Extraction

The first feature extraction tool is a tool for Marching Cubes isosurface extraction [Lorensen & Cline, 1987]. Isosurface extraction (i.e., the extraction of the set of voxels whose values are above some threshold T) is an important functionality because it allows dense structures of interest, such as the spine, ribs, and kidney pelvis, to be extracted from the CAT datasets. These structures provide spatial reference for relative locations of kidney structures and kidney cysts and thus are useful for treatment planning.

The Marching Cubes technique can be understood by considering a set of volume data with data values (voxels) distributed on a regularly-spaced 3D grid. The grid lines can be said to define the bounding edges of cubes. An isosurface could potentially intersect any bounding edge of any cube. The intersection points between the isosurface and the cube's bounding edges can be estimated by linear interpolation using the values at the vertices of each edge. These intersections are used as the vertices of triangular facets that approximate the isosurface. The Marching Cubes method is applied cube-by-cube to extract this isosurface.

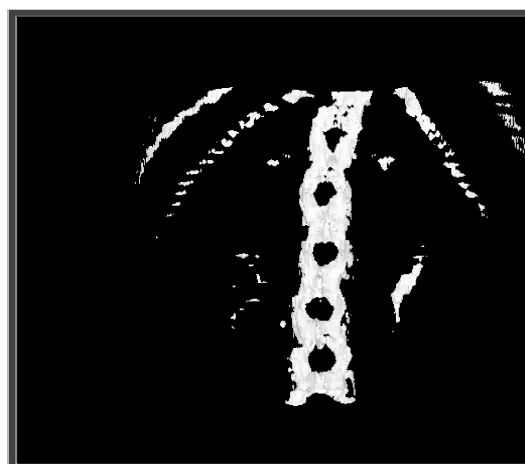


Fig. 12. Isosurface showing spine, ribs, and kidney pelvis in one abdominal CAT dataset.

The Marching Cubes algorithm is fairly computationally intensive for large datasets making real-time operation of the Marching Cubes tool a difficult goal on desktop PCs or workstations. Careful organization of Marching Cubes computations can allow near-real-time performance, however, on a vector-parallel machine [Newman et al., 2000]. The HPMVS Marching Cubes tool is able to compute the Marching Cubes approximately ten times as fast using the C94 than on the front-end workstation. For most CAT datasets, structures of interest can be extracted in approximately 1 second on the C94, which is quite acceptable for interactive use. An example isosurface extraction of the spine, ribs, and kidney pelvis from a $256 \times 256 \times 37$ CAT dataset is shown in Figure 12. This isosurface, which contains over 25,000 facets, was extracted in 0.83 seconds of CPU time on the C94. Transmission of the isosurface via ethernet takes just under 1.0 seconds, thus enabling acceptable interactivity for users. For comparison, the same isosurface can be extracted in 10.3 seconds of CPU time on the C94 using the vtk's [Schroeder et al., 1998] Marching Cubes extraction functionality; our realization of isosurface extraction is considerably more efficient than the vtk.

Direct Volume Rendering

The second extraction tool is the direct volume rendering (DVR) tool. Direct volume rendering combines feature extraction and image rendering. The HPMVS direct volume rendering tool renders the data directly from voxel data without extracting an explicit geometric surface model.



Fig. 13. Direct volume rendering by volume ray casting for an abdominal CAT dataset.

It is very appropriate for display of materials such as soft tissue, which are difficult to locate using surface-based extraction techniques such as the Marching Cubes.

The direct volume rendering tool uses the volumetric ray casting [Sabella, 1988] with gradient shading. In volume ray casting, a ray is cast through each pixel in the image plane and traced through the data volume. The image plane pixel intensity is determined based on the accumulation of optical effects encountered by the ray as it passes through the volume. In the DVR tool, rays are cast according to a parallel projection and attenuated as they pass through the volume. The intensity of image plane pixel P can be formulated as:

$$I = \sum_{t=t_1}^{t_2} I_{shade}(t) V_{transp}(t), \quad (1)$$

where $I_{shade}(t)$ is the illumination of the voxel V that the ray encounters at a time t , $V_{transp}(t)$ is the transparency of V with respect to the ray direction, and t is an intrinsic time parameter expressing the movement of a particle along the ray, with t_1 the starting time (i.e., the time that the first sample point is encountered) and t_2 the ending time. The transparency of a voxel is affected by the attenuation of the transmitted light from preceding voxels, thus the transparency of the voxel encountered at a time \hat{t} is a function of the opacity of the voxels in front of it along the ray:

$$V_{transp}(t = \hat{t}) = \prod_{t=\hat{t}}^{t_2} [1 - V_{opacity}(t)], \quad (2)$$

where $V_{opacity}(t)$ is voxel opacity.

Realizations of the direct volume rendering tool have been achieved on both the C94 and the front-end workstation. Algorithmic details of

efficient volume ray-casting in a vector-parallel environment can be found in [Newman et al., 2000]. The DVR tool enables volume ray-casting to be computed approximately 50 times faster on the C94 than on the front-end workstation; using the C94, an initial rendering can be computed in less than 0.5 seconds and a sustained animation rate of approximately 12 Hz is possible for $256 \times 256 \times 37$ CAT datasets of the abdomen. Thus, the DVR tool can achieve borderline real-time frame rates for ray-casting through a rotating volume. The output of one HPMVS direct volume rendering is shown in Figure 13. In this figure, the rays are cast from the back of the data (i.e., from the patient's back towards the patient's front) with the rays commencing from a location about one third of the way through the dataset. (This origination plane allows more clear discernment of the kidneys in the rendering.) One of the kidneys is especially clear in this image and is visible as a dark region on the left side of the displayed figure. A number of gastro-intestinal structures can be perceived in the rendering.

Anatomic Structure Segmentation

The tool suite also includes a segmentation tool for the extraction of specific features of interest that cannot be easily visualized using either iso-surface extraction or direct volume rendering. Specifically, we have developed and included functionality that allows the kidneys, nearby organs of interest, the kidney's collecting system, and kidney cysts to be determined and then displayed in an integrated visualization. (These structures of interest and other useful anatomic landmarks (e.g., the spine) can be displayed and interactively explored using the system.) The tool is especially well-suited for segmentation of the kidney tumors, and is thus helpful in assessment of tumor location and size and, through incorporation of segmented tumors in



Fig. 14. One cross-sectional slice of a CAT dataset.

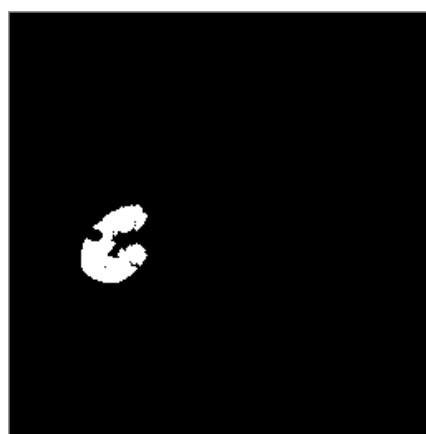


Fig. 15. Slice image of kidney extracted using segmentation tool (same slice as image at left).

integrated renderings of tumors and organ structures of interest, in planning for surgery.

The segmentation method is a 3D approach that produces a segmented volume. It consists of several stages. The first stage is a volume-growing step that forms an initial grouping of volume data elements (voxels) into an appropriate class (e.g., normal kidney tissue, abnormal kidney tissue, or some other structure of interest). To initialize the volume-growing, the user must select one point on the structure of interest and the classification of that point. The tissue identity is used to determine thresholds for the volume growing. Subsequent stages use morphological operations and the slice-adaptive histogramming [Newman et al., 1999] to refine the initial segmentation.

An example of one cross-sectional slice of an abdominal CAT dataset is shown in Figure 14. One cross-sectional image of a kidney volume segmented using the tool is shown in Figure 15. The cross-sectional plane of the image is the same as that shown in Figure 14. HPMVS allows the segmented structures to be displayed in individual slice images or as 3D geometries integrated into a 3D rendering of multiple extractions and segmentations. A rendering showing, for one dataset, the segmentation tool's extraction of the kidneys overlaid on the spine and ribs (which were extracted using the HPMVS Marching Cubes tool) is shown in Figure 16. An image from a 3D rendering of segmented kidneys and extracted cysts is shown in Figure 17. This image is displayed using an inverted gray map to clarify the structures of interest extracted by the tool. In the image, the kidneys, cysts, and

lungs are semi-transparent to allow the viewing of all internal structures of interest. The cysts are displayed using somewhat darker gray levels than the normal kidney organ tissue. The cysts appear as blotchy regions on and in the kidneys in the image. Bones, spine, ribs, and kidney collecting system are displayed in darkest grays. A portion of one of the lungs has also been segmented with the tool and is visible above the rightmost kidney in the figure. (Only the lowermost portion of that lung is present in the abdominal CAT dataset that this figure was created from.)

The tool produces segmentations that are equivalent to the manual segmentations of unaided

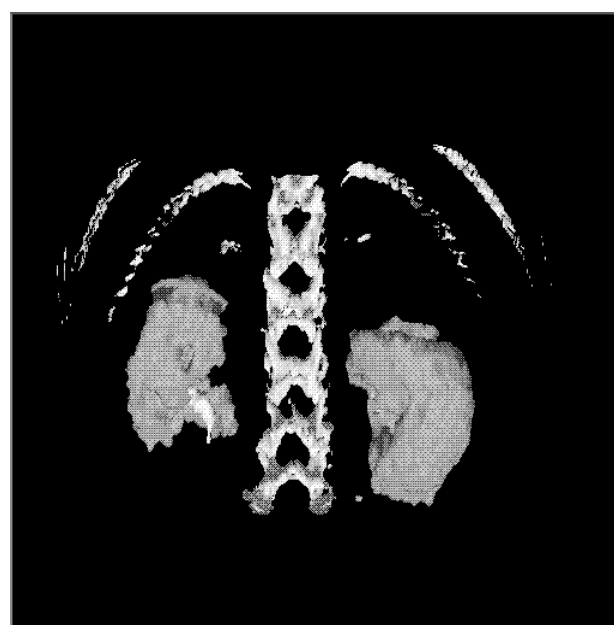


Fig. 16. Rendering showing extracted kidneys overlaid on ribs and spine from an abdominal CAT dataset.

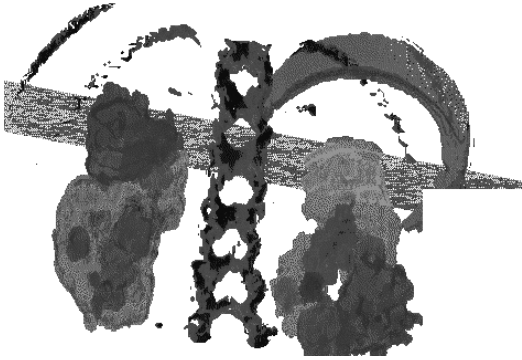


Fig. 17 Rendering showing extracted kidneys, cysts, and a lung overlaid on ribs and spine from a CAT dataset. The horizontal plane intersecting the spine is of an arbitrary cross-sectional slice that can be interactively positioned by the user.

human experts. The segmentation tool also executes quickly on the workstation. For example, for a $256 \times 256 \times 37$ CAT dataset, the kidney can usually be segmented in several seconds of time and small- or moderate-sized kidney cysts can generally be segmented in a few seconds of CPU time. The segmentation tool is one of the key functionalities that allow generation of useful visualizations for the renal application.

4. Final Discussion and Future Directions

The HPMVS tools are quite useful for assessing the progression of renal disease and for planning surgical interventions. The framework of the tools allows integrated renderings of extracted structures to be explored interactively by a user in near-real-time. This aspect of the tool suite enables multiple view point study of the spatial relationships of diseased tissue and anatomical landmarks. Understanding of these relative locations is particularly useful for considering paths of approach for tissue excise.

From a high level view, the tools for feature and structure extraction, data exploration, and rendering can be said to aid in the more full exploitation of tomographic data. The applicability of the tool suite is enhanced through the near-real-time processing allowed by the networked framework.

To date, the HPMVS has been used as an experimental prototype in order to validate the feasibility of information technology to improve

renal assessment and treatment planning for the von Hippel Lindau syndrome. Clinical staff report that the interactive tools included in the HPMVS enable faster discovery of renal tumor location and size and better understanding of relative positions of tumors. Based on experiences using the HPMVS, clinical staff desire incorporation of information technology tools into daily kidney assessment activities.

Although the system as developed requires access to a C94 computer (which could be too costly for some clinical centers), due to recent architectural enhancements to commodity microprocessors, the HPMVS technology can likely be realized using new low-cost technology. Specifically, the HPMVS is designed to deliver optimal performance by exploiting vector-processing capabilities of the C94, and vector-processing capabilities have recently become available on current generation desktop machines through the AMD Athlon's enhanced 3DNow! capability, the Intel Pentium III's Streaming SIMD Extension (SSE) capability, and Sun UltraSPARC VIS capability. These desktop vector-processing capabilities present an opportunity to re-deploy HPMVS on PCs or workstations. The opportunity seems particularly hopeful because the vector-parallel-efficient approaches developed for the Cray seem well-suited for efficient performance using desktop SIMD capability such as that offered by the 3DNow! and SSE instruction sets.

Acknowledgments

The work described in this paper was partially supported by a grant from Cray Research. We are thankful for their support. We are also thankful for fruitful discussions related to this work with Drs. Peter Choyke and Stephen Bacharach of the U.S. National Institutes of Health and for the helpful comments of the reviewers.

References

- [Brown et al., 1997] M. S. BROWN, M. F. McNITT-GRAY, N. J. MANKOVICH, J. G. GOLDIN, J. HILLER, L. S. WILSON, D. R. ABERLE, "Method for Segmenting Chest CT Image Data Using an Anatomical Model: Preliminary Results," *IEEE Trans. on Medical Imaging*, **16** (1997), 828–839.

- [Cameron, 1995] G. CAMERON, "Modular Visualization Environments: Past, Present, and Future," *Computer Graphics*, **29**, No. 2 (1995), 3–4.
- [Crockett, 1997] T. W. CROCKETT, "An Introduction to Parallel Rendering," *Parallel Computing*, **23** (1997), 819–843.
- [Drebin et al., 1988] R. DREBIN, L. CARPENTER, AND P. HANRAHAN, "Volume Rendering," *Computer Graphics*, **22**, No. 4 (1988), pp. 65–74.
- [Foley, 1996] J. D. FOLEY, A. VAN DAM, S. K. FEINER, J. F. HUGHES, *Computer Graphics: Principles and Practice, Second Edition in C*, Addison-Wesley, New York, 1996.
- [Gouraud, 1971] H. GOURAUD, "Continuous Shading of Curved Surfaces," *IEEE Trans. on Computers*, **C-20**, No. 6 (1971), 623–629.
- [Hansen & Hinker, 1992] C. HANSEN, P. HINKER, "Massively Parallel Isosurface Extraction," *Proc., Visualization '92*, (1992) Boston, Mass., pp. 77–83.
- [Hsu, 1993] W. HSU, "Segmented Ray Casting for Data Parallel Volume Rendering," *Proc., 1993 Parallel Rendering Symposium*, (1993) San Jose, Calif., pp. 7–14.
- [Kaufman et al., 1991] A. KAUFMAN, R. BAKALASH, D. COHER, R. YAGEL, "Architectures for Volume Rendering," *IEEE Engineering in Medicine and Biology*, **9**, No. 4 (1991), pp. 18–23.
- [Kessler et al., 1994] M. KESSLER, L. CATALLO, D. MC-SHAN, "Design and Simulation of Conformal Radiation Therapy Using AVS," *Proc., Int'l AVS Users Group Conf. '94*, (1994) Boston, Mass., pp. 397–407.
- [Kobashi & Shapiro, 1995] M. KOBASHI, L. SHAPIRO, "Knowledge-Based Organ Identification From Images," *Pattern Recognition*, **28**, No. 4 (1995), 475–491.
- [Lacroute, 1995] P. LACROUTE, "Real-Time Volume Rendering on Shared Memory Multiprocessors Using the Shear-Warp Factorization," *Proc., 1995 Parallel Rendering Symposium*, (1995) Atlanta, pp. 15–22.
- [Lorensen & Cline, 1987] W. E. LORENSEN, H. E. CLINE, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *SIGGRAPH 87 Conf. Proceedings, Computer Graphics*, **21**, No. 4 (1987), pp. 163–169.
- [Neumann, 1998] H. P. H. NEUMANN, *Von Hippel-Lindau Krankheit* [in German], Medical University Clinic/Hospital, (1998) Freiburg, Germany.
- [Newman et al., 2000] T. NEWMAN, N. TANG, "Approaches that Exploit Vector-Parallelism for Three Rendering and Volume Visualization Techniques," accepted, *Computers & Graphics* (2000).
- [Newman et al., 1996] T. S. NEWMAN, N. TANG, S. BACHARACH, P. CHOYKE, "A Volumetric Segmentation Technique for Diagnosis and Surgical Planning in Lower Torso CT Images," *Proc., 13th International Conf. on Pattern Recognition*, (1996) Vienna, pp. C–553 to C–557.
- [Newman et al., 1999] T. NEWMAN, N. TANG, C. DONG, P. CHOYKE, "Slice-Adaptive Histogram for Improvement of Anatomical Structure Extraction in Volume Data," submitted, *Pattern Recognition Letters* (1999).
- [Ogata et al., 1998] M. OGATA, T. OHKAMI, H. LAUER, H. PFISTER, "A Real-Time Volume Rendering Architecture Using an Adaptive Resampling Scheme for Parallel and Perspective Projections," *Proc., 1998 Symposium on Volume Visualization*, (1998) Research Triangle Park, N. Carolina, pp. 31–38.
- [Park et al., 1995] J. PARK, D. METAXAS, L. AXEL, "Volumetric Deformable Models with Parameter Functions: A New Approach to the 3D Motion Analysis of the LV from MRI-SPAMM," *Proc., Fifth International Conf. on Computer Vision*, (1995) Cambridge, Mass., pp. 700–705.
- [Phong, 1975] B. T. PHONG, "Illumination for Computer Generated Pictures," *Comm. of the ACM*, **18**, No. 6 (1975), 311–317.
- [Renaud & Rousselle, 1997] C. RENAUD, F. ROUSSELLE, "Fast Massively Parallel Progressive Radiosity on the MP-1," *Parallel Computing*, **23** (1997), pp. 899–913.
- [Robb et al., 1996] R. A. ROBB, D. P. HANSON, J. J. CAMP, "Computer-Aided Surgery Planning and Rehearsal at the Mayo Clinic," *IEEE Computer*, **29**, No. 1 (1996), 39–47.
- [Robertson et al., 1991] D. ROBERTSON, V. JACOBSON, W. JOHNSTON, S. LOKEN, E. THEIL, B. TIERNEY, "Distributed Visualization Using Workstations, Supercomputers, and High Speed Networks," *Proc., Visualization '91*, (1991) San Diego, Calif., pp. 379–382.
- [Rueckert et al., 1997] D. RUECKERT, P. BURGER, S. M. FORBAT, R. D. MOHIADDIN, G. Z. YANG, "Automatic Tracking of the Aorta in Cardiovascular MR Images Using Deformable Models," *IEEE Trans. on Medical Imaging*, **16**, No. 5 (1997), 581–590.
- [Sabella, 1988] P. SABELLA, "A Rendering Algorithm for Visualizing 3D Scalar Fields," *Computer Graphics*, **22**, No. 4 (1988), 51–58.
- [Sachse et al., 1998] F. B. SACHSE, M. WOLF, C. WERNER, K. MEYER-WAARDEN, "Extension of Anatomical Models of the Human Body: Three-Dimensional Interpolation of Muscle Fiber Orientation Based on Restrictions," *Journal of Computing and Information Technology*, **6**, No. 1 (1998), 95–101.
- [Schneider et al., 1998] B.-O. SCHNEIDER, J. VAN WELZEN, "Efficient Polygon Clipping for an SIMD Graphics Pipeline," *IEEE Transactions on Visualization and Computer Graphics*, **4**, No. 3 (1998), 272–285.
- [Schroeder et al., 1998] W. SCHROEDER, K. MARTIN, B. LORENSEN, *The Visualization Toolkit, Second Edition*, Prentice Hall: Upper Saddle River, New Jersey, 1998.

- [Solaiyappan et al., 1996] M. SOLAIYAPPAN, T. POSTON, P. A. HENG, E. R. MCVEIGH, M. A. GUTTMAN, E. A. ZERHOUNI, "Interactive Visualization for Rapid Noninvasive Cardiac Assessment," *IEEE Computer*, **29**, No. 1 (1996), 55–61.
- [Weng et al., 1997] J. WENG, A. SINGH, M. Y. CHIU, "Learning-Based Ventricle Detection from Cardiac MR and CT Images," *IEEE Transactions on Medical Imaging*, **16**, No. 4 (1997), 378–391.
- [Whitted, 1980] T. WHITTED, "An Improved Illumination Model for Shaded Display," *Comm. of the ACM*, **23**, No. 6 (1980), 343–349.

Received: January, 1999

Revised: January, 2000

Accepted: May, 2000

Contact address:

Timothy S. Newman
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL 35899 USA
e-mail: tnewman@cs.uah.edu
phone: (256) 890-6619
Fax: (256) 890-6239

TIMOTHY S. NEWMAN is an Assistant Professor of Computer Science at the University of Alabama in Huntsville. He received the Ph.D. in Computer Science from Michigan State University and then was awarded a National Research Council postdoctoral fellowship with the U.S. National Institutes of Health. His interests include visualization, medical imaging, computer graphics, computer vision, and applications of high performance computing to problems in these areas. He is a senior member of the IEEE and a member of the ACM.

NING TANG recently completed his Masters of Science in Computer Science at the University of Alabama in Huntsville. Mr. Tang also holds a Masters in Electrical Engineering from Delft University and a Bachelors in Electrical Engineering from Tsinghua University. His interests include computer graphics, visualization, and object-oriented design and development.
