9 DATA VISUALIZATION AND DISPLAY

Dr. Jinman Kim[1], Dr. Tom Weidong Cai[1], Prof. Michael Fulham[1,2,3], Dr. Stefan Eberl[1,2] and Prof. David Dagan Feng[1,4]
[1]*Biomedical & Multimedia Information Technology (BMIT) Research Group, School of Information Technologies, University of Sydney, Australia*
[2]*Department of PET and Nuclear Medicine, Royal Prince Alfred Hospital, Australia*
[3]*Faculty of Medicine, University of Sydney, Australia*
[4]*Centre for Multimedia Signal Processing, Department of Electronic & Information Engineering, Hong Kong Polytechnic University, Hong Kong*

CHAPTER 9 DATA VISUALIZATION AND DISPLAY

9.1 INTRODUCTION

Advances in digital biomedical imaging are enabling unprecedented visualization of the structure, function, and pathology of the human body. These images can be acquired with multiple modalities and in multiple dimensions, and include modalities such as magnetic resonance (MR) imaging and positron emission tomography and computed tomography (PET/CT). These new techniques have also introduced significant challenges for efficient visualization [1-3]. In line with advances in image acquisition, methods to visualize and display these data have also seen rapid developments and challenges. PET/CT was introduced in 2000 with a PET scanner coupled to an early generation helical CT scanner but in the space of 5 years the leading edge PET/CT has a 64-slice CT coupled to a PET scanner that has much better resolution and sensitivity than the device of 2000. A major challenge now is to put these vast amounts of imaging data in a readily usable and viewable format for interpretation. Fortunately, there has also been tremendous progress in three-dimensional (3D) volume visualization of biomedical data. In general, 3D visualization of biomedical data refers to the ability to interact and navigate the image data in a realistic 3D volumetric display. These volumetric displays are typically constructed from 2D slice images which are acquired in a regular pattern (e.g., one slice every millimeter) and make up a volumetric grid. Rapidly improving capabilities for 3D visualization have made this an attractive method for imaging applications, including those geared towards image guided surgery (IGS), radiotherapy, and computer-aided diagnosis (CAD) [1, 4-11]. With current visualization technologies, it is possible to perform real-time interactive visualization of multi-dimensional volumes using low-cost hardware instead of restricting it to expensive high-end workstations [12, 13]. These visualization advances have been accompanied by developments and improvements in display and input control devices. In current diagnostic workstation, it is common to find multiple liquid crystal display (LCD) screens to visualize the imaging data in 2D and 3D views. Specialized display devices, such as stereoscopic screens, enable new approaches to visualize and interact with biomedical data; virtual reality (VR) systems, for instance, are able to provide a realistic image alongside highly interactive control of the visualization process [2, 14]. To provide convenient and efficient navigational control of 3D visualizations, new input devices designed for 3D controls have also been introduced [15-17]. The appropriate visualization and display of biomedical data, given the vast amount of data, is now mandatory

to ensure accurate diagnosis. Moreover, intelligent and innovative new visualization algorithms are necessary in order to overcome the increase in image dimensions that is making conventional viewing approaches inefficient.

This chapter describes data visualization and display techniques for multi-dimensional biomedical images. Section 2 discusses 2D visualization techniques that include multi-planar reformatting and oblique/curved sectioning. This is followed by a section on volume rendering techniques (surface-based, direct, and texture-based rendering) for 3D volumetric data and for multivariate data (greater than 3D, i.e., time-varying, dynamic sequence). Section 4 introduces methods and devices that are used to navigate through volumetric data. Enhancements in volume visualization are then discussed in Section 5, followed by visualization optimization methods (hardware and software approaches) in Section 6. A case study on dual-modality PET/CT visualization is presented in Section 7, which illustrates how advances in visualization techniques are adapted for use with biomedical data to improve diagnosis and image understanding. Display devices and technologies are presented in Section 8, which is followed by a summary of cutting-edge biomedical data visualization projects in Section 9. Summary of the chapter is than given in Section 10.

9.2 TWO-DIMENSIONAL (2D) VISUALIZATION TECHNIQUES

Conventional methods of visualizing volumetric biomedical images utilize 2D coronal or transaxial views, with multiple images viewed in *montage* or *slice-by-slice* formats. With increases in the resolution of imaging devices and the vast amount of data that they generate, it is now virtually impossible to rely on the previous approaches for image analysis. This has lead to the development of new 2D approaches for biomedical data visualization such as volume slicing, multi-planar reformatting and curved sectioning, which provide views complementary to conventional 2D views.

9.2.1 Multi-Planar Reformatting

Volumetric data allows the voxels in the volume to be reformatted into different orthogonal orientations, namely, *trans-axial*, *sagittal*, and *coronal* views. These orthogonal views are often displayed simultaneously as shown in Figure 9.1, exemplified with a brain MR data set [18]. Multi-planar images provide an effective tool in visualizing volumetric data with interactive control and are the default visualization setup in many diagnostic applications [5, 12, 19]. By simultaneously displaying the three orthogonal views, they enable rapid observation of the volumetric data. These views are often navigated through the ‘click and drag’ method of manipulating a viewpoint within one of the three views of the volume, causing other views to be reformatted according to the new point position. Multi-planar sectioning as shown in Figure 9.1(d) is another popular approach toward viewing multi-planar

images, which visualizes the three orthogonal views stacked perpendicular to each other, thereby creating a 3D visualization.

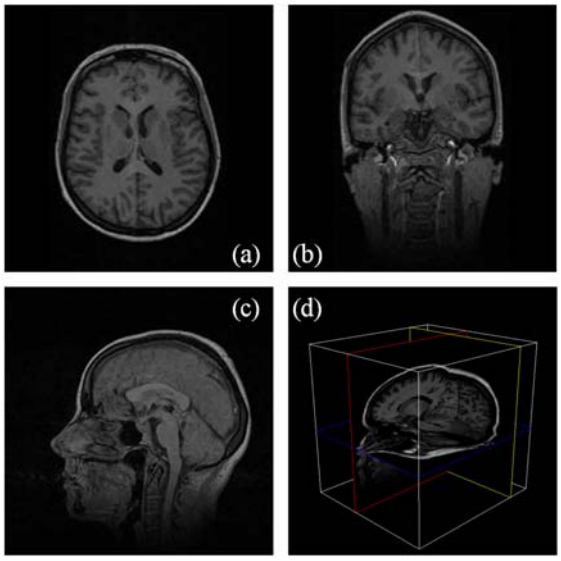

Figure 9.1 Multi-planar reformatted views of a MR brain image: (a) transaxial; (b) coronal; and (c) sagittal views. Multi-planar sectioning shown in (d) is the result of stacking images together.

9.2.2 Oblique and Curved Sectioning

In various clinical applications, the required 2D views may not necessarily lie parallel to the orthogonal orientation of the 3D volume image, as obtained from multi-planar reformatting. It is at times necessary to view cross-sections made at arbitrary angles through the volume. Oblique sectioning, also known as volume slicing, is a technique which cuts the volume in any conceivable orientation by a user-defined cutting plane [20]. This endows users with the ability to depict the inner structure of the volume by removing arbitrary parts of the volume that obscure the primary volumes of interest (VOIs). An example of oblique sectioning is shown in Figure 9.2, which demonstrates the removal of occluding sections from a whole-body CT volume. Instead of utilizing a straight line (cutting plane) selection as in oblique sectioning, curved sectioning traces along an arbitrary path on any orthogonal image to construct an image. Calculation of the voxels on the cutting plane is then essentially performed

via an interpolation or re-sampling operation such as tri-linear or nearest neighbor interpolation [21]. Another approach that has been shown to produce a good selection of viewable data is positioning and resizing a *clipping cube box*, which encapsulates the volume such that only the volumetric data residing inside the box is visible.

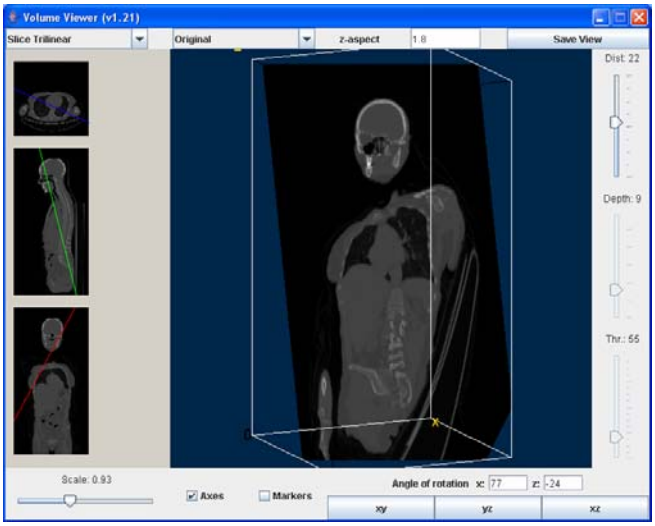

Figure 9.2 Oblique sectioning applied to a whole-body CT dataset using ImageJ [22] software with the Volume Viewer plug-in [23]. Thumbnails at left are the cutting planes applied to different orthogonal views, the resulting 3D volume is in the center. Tri-linear interpolation has been applied to volume rendered images.

9.3 THREE-DIMENSIONAL (3D) VISUALIZATION TECHNIQUES

In 3D visualization of biomedical data, there are four common techniques that may be used to maximize the visual information, which can be obtained from different types of biomedical data. These visualization techniques – surface rendering, direct volume rendering, texture-based volume rendering, and multivariate rendering, can be used independently or in combination. The first three visualization techniques deal with having different ways to render the data, with each technique having its advantages and limitations. Figure 9.3 compares the three techniques in rendering the bone structure from a CT scan. Rendering was performed using marching-cube surface rendering [24], direct volume rendering, and texture-based volume rendering, all available through the visualization tool kit (VTK) [25]. In these rendering techniques, appropriate parameters were applied to select the bone structure. In this figure, all three rendering methods appear to be equally matched in visual qualities. However, marching-cube method only rendered the approximate shapes of the bones (calculating

polygonal surfaces from the voxels), whereas the other methods rendered every voxels belonging to the bones. For multivariate rendering, the rendering techniques above are used alongside others to visualize data that are in multiple states, i.e., time-varying and dynamic data. Differences in the rendering techniques are explained in greater detail in the following sections.

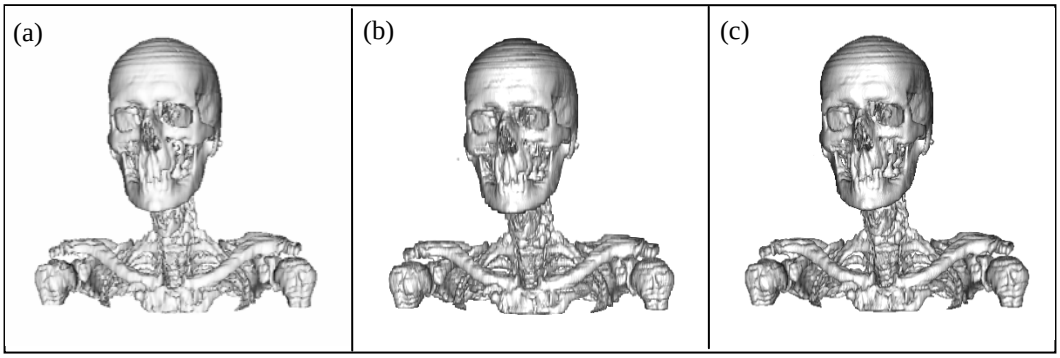

Figure 9.3 Rendering of bone structures from a CT dataset using three different rendering algorithms. (a) surface rendering; (b) direct volume rendering and, (c) texture-based volume rendering.

9.3.1 Surface Rendering

Surface rendering techniques build a geometrical contour representation of the surface defined by the segmentation of the image volume. The contours of the segmented volume(s) are then extracted with surface tiling techniques [21, 24, 26, 27] which creates polygonal surfaces representing the structure. One of the most well known algorithms for surface tiling is the marching cubes algorithm [24], which functions by creating triangulated representations of the surface, and has produced many optimizations and variations for different applications. The main attraction that surface rendering offers is its ability to leverage the advances in modern graphics hardware that is specifically designed to process large numbers of polygons, providing the ability for real-time volume rendering. In addition, lighting and shading models can be applied to the surfaces (e.g. Phong and Gouraud [28, 29]) which aid in depth perception and overall realism of the volume.

Generally, surface rendering techniques require segmentation of the image volume to determine the structures to render. Segmentation as a pre-process is the main disadvantage of surface rendering due to the complexities in accurate structure delineation and computation. There are numerous segmentation techniques available that are optimized for individual imaging modalities. However, accurate recognition and delineation of all individual structures

in an image is likely to be restricted only to controlled environments [2, 30, 31]. Due to the reliance on segmentation results, once the pre-processing has been performed, it is computationally inefficient to modify the parameters which generated the surfaces. A further limitation of surface rendering is that only the surface is rendered and hence potentially important information about structures and pathologies inside the surface rendered organ or structure are lost.

9.3.2 Direct Volume Rendering

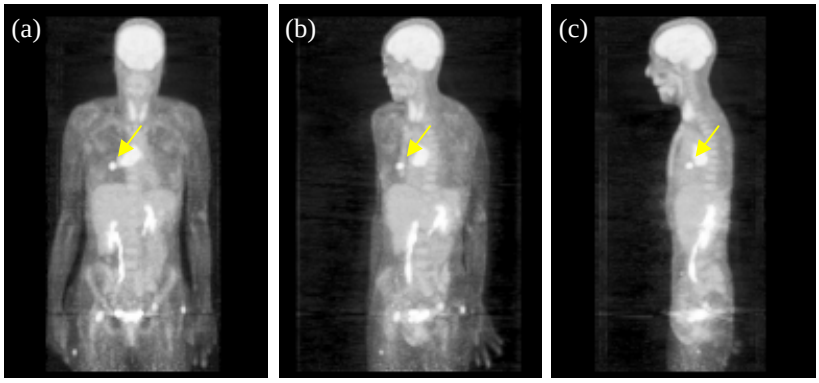

Figure 9.4 Sequence of MIP rendered PET volume data. (a) coronal view; (b) 45 degrees to the left of (a); and (c) 90 degrees left of (a). Animating frames rendered at different angles provides improved perception of the 3D data. This particular data enabled clear identification of the tumor using MIP volume rendering (bright spot, indicated by an arrow).

Direct volume rendering renders every voxel in the image, and thus differs from surface rendering in that it does not require a surface extraction of the image data to be visualized. Rendering of every voxel allows for natural geometrical structures and the representation of the complete volume data. However, due to the calculations required for every voxel, direct volume rendering is computationally expensive. Real-time rendering performance for direct volume rendering is only achievable using commodity graphics hardware through the use of specialized rendering optimization techniques. Section 9.6 covers performance optimization for volume rendering visualization. The most versatile approach to direct volume rendering of biomedical data is through the *ray-casting* (also known as ray-tracing) technique [3, 20, 32]. The basic principle of ray-casting is the casting of rays starting from the viewer's eye through each render-able voxel in the image and involves sampling, filtering, lighting, and accumulation of voxel colors and opacities as the ray passes through the volume. A computationally fast approach to ray-casting is maximum intensity projection (MIP) which has

found several clinical applications [33-35]. MIP works by projecting the voxel with the maximum intensity that falls in the path of the ray. Although computationally efficient, this approach is limited in illustrating depth and orientation, where the MIP of the volume cannot be distinguished between left/right, and front/back. In order to improve the sense of 3D with MIP, animations are often created that consists of MIP rendering with rotated viewing angles, which aids in the viewer's perception of the 3D volume. Example of MIP of whole-body PET data is shown in Figure 9.4, with selection of frame sequences that are used to animate the volume. Other volume rendering techniques that are favorable to biomedical data visualization are *splatting* [36] and *shear warp* [37, 38] methods which are designed to improve the rendering efficiency at the cost of visual quality.

9.3.3 Texture-Based Volume Rendering

Advances in consumer graphic card technologies are resulting in the development of graphic cards that are extremely efficient in their texture mapping capabilities. Texture mapping is the process of applying images as textures to geometric objects. This attribute has been utilized in the visualization of biomedical data with real-time interactive capabilities using low-cost consumer hardware [20, 39-41]. The principal concept of texture-based volume rendering is to create parallel planes through the columns of the volume data, in the direction most perpendicular to the user's line of sight. The number of planes to be created is based on the sampling rate, where higher rates equate to better visual quality. In a typical setting, sampling rate is set proportional to the dimensions of the voxel to be rendered. The parallel planes are converted to polygons that are texture-mapped with appropriate 3D texture co-ordinates [40] that are derived from the volume data, and are then drawn in back-to-front order with blending of the voxel colors to result in a smooth image. Texture-based volume rendering produces renders that are of comparable quality to direct volume rendering, apart from the noticeable differences during volume manipulations, such as rotation [3]. Due to its image quality and computational efficiency, texture-based volume rendering has become a popular choice in many applications of biomedical data visualization [32, 39, 40, 42, 43].

9.3.4 Multivariate Volume Rendering

Many applications in the field of biomedical visualization require visual outputs of data that contain multiple scalar values at each sample (voxel) point [1, 44-46]. Such data is often referred to as multivariate data and can come from numerous applications, for example, multi-volume data (dual-modality PET/CT), and time-varying data (ultrasound, flow visualization). In multi-volume visualization, two or more complementary datasets are combined and volume rendered. Combination of these data may be in the form (1) volume overlay – overlapping of

one volume onto another; or (2) data-intermixing – calculation of a new voxel value according to two or more values from the same sample. An example of data-intermixing of dual-modality biomedical data is discussed in Section 9.7. Time-varying data is frequently used in biomedical applications to visualize the changing scalar values of a selected object over a defined time. Tory et al [47] introduced *glyphs*, animated arrowhead figures that represented scalar properties (uptake and temperature) with regard to time. Glyphs were cleverly used to visualize time-varying 3D MRI data for lesion analysis. Thune and Olstad reported [48], animation of 4D ultrasound images (3D spatial and 1D temporal) by direct volume rendering.

9.4 VOLUME NAVIGATION INTERFACE

In Section 9.2 we introduced 2D approaches to navigate 3D data through the use of a pointer in two dimensions (the x and y axes). In 3D, navigation is based on movement of the volume with an additional z-axis. Data with even greater dimensions, such as multivariate data, demands additional interactive control and input devices for efficient control. This section highlights common approaches for the navigation of volume rendered data.

9.4.1 3D Volume Navigation

Conventional interactive navigational methods for volume rendering consist of scaling, rotation, and translation within the 3D co-ordinate system. To improve rendering performance during interactive navigation, the sampling rate of the volume can be lowered to accelerate the rendering capabilities [7, 39, 41]. Once the interaction is completed, the sampling rate can be reset to its default rates. Automating the toggle in sampling rate by listening to changes in mouse interaction allows for progressive refinement during interactive volume rendering. This attribute is particularly important when rendering large volume data. Comparison of texture-based rendering performance with regard to varying sampling rates is demonstrated in Figure 9.5.

For 4D or greater dimensional data, such as time-varying and multi-modal volume rendering, additional complexity is added to volume navigation. For time-varying rendering, the volumes are often animated with controls similar to those used for video playback (*cine)* [5]. Such video controls can be used together with 3D navigation, a concept which has found usage in biomedical visualization applications such as those enabling virtual endoscopy [49, 50], which requires movement both in 3D and the temporal dimension.

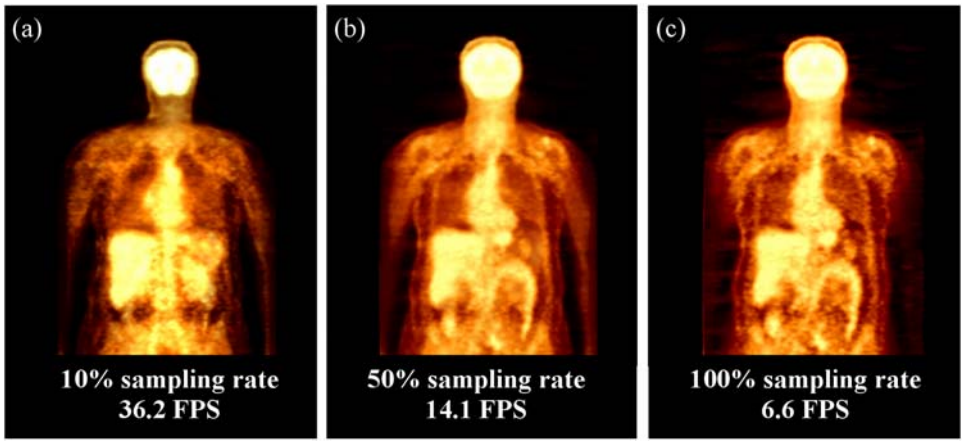

Figure 9.5 Different sampling rates applied to whole-body PET data using texture-based volume rendering. The lowest sampling rate has the poorest visual quality, with obvious loss of detail in rendering the internal organ structures. Increasing the sampling rate improves rendering quality. However, there is a decrease in the frames per second (FPS) when the volume is interacted with (rotated around all axes).

9.4.2 Input Devices

In the design of biomedical data visualization software, due to the complexity in volume navigation, it is often beneficial to customize the input devices. Rosset [12] introduced the use of the *jog-wheel* devices that are widely adopted in the video and movie industry, thereby providing the ability to control the different dimensions more efficiently than is possible by conventional means, such as the computer *mouse* device. Alternative input devices are readily available that can be used to replace or complement the mouse, such as HP's 3D *SpacePilot* [51] consisting of a knob that can be tilted, pushed, pulled, twisted, and turned in any direction to indicate the direction and velocity with which you wish to move your model. Touch screens have been used as alternative input devices for visualization applications. With recent developments in and subsequent adaptations of personal digital assistant (PDA), tablet PC, and ultra mobile PC (UMPC) hardware, ever greater numbers of visualization applications are taking advantage of stylus and touch sensitive screens. Figure 9.6 demonstrates the potential use of a touch-screen enabled device for input control to complement the conventional mouse device. In this example, the touch-screen input interface was separated from other screens, thereby freeing the screens to only display images. Such applications can find usefulness for teaching and presentation purposes [15, 52]. Bornik et al [15] demonstrated a hybrid input device, controlling a stereoscopic projection display (see Section 9.8.1 for stereoscopy) and a desktop display. The input device named *Eye of Ra* is a combination of a wireless mouse, a

stylus pen, and motion tracking sensors. A camera setup was used to track the input device movements which in turn control the stereo projected display for rapid and rough control, whereas a stylus pen with mouse buttons is used to control the desktop display for precise control. Kim and Varshney used an eye-tracking device for saliency-guided visualization [53]. Here, based on the region of interest the eye was focusing, visual emphasis was applied to highlight the interested region. It was shown to be effective in helping users navigate through complex volumetric datasets.

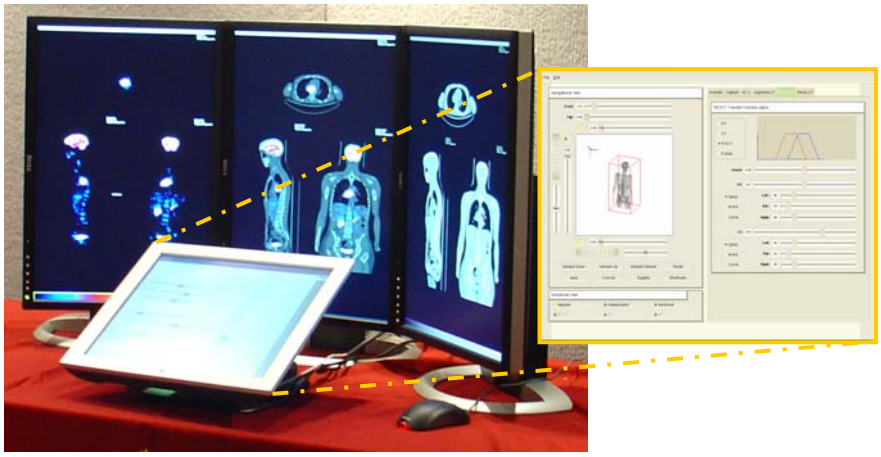

Figure 9.6 The touch-screen is used to display the input interface that controls the display of data in the other three screens. In this example, the displays were used to visualize PET, fused PET/CT, and CT data respectively (left to right). By separating the input controls, additional screen space is available to display the images.

9.5 VOLUME ENHANCEMENT AND MANIPULATION

Enhancement and manipulation techniques that are applied to 2D slice images, such as filtering, segmentation, and geometrical transformation, can also be extended and applied to 3D image volumes [2, 54]. When extended into the 3D domain, additional information is available and this therefore generally allows for more accurate (if complicated) methods. An example is the image interpolation that is applied when images are scaled. In 2D data, bilinear interpolation is applied, which calculates a voxel value based on the voxel's 8-immediate neighbors. In 3D, trilinear interpolation may be applied instead, which uses an additional 18 neighbors (for a total of 26) in the calculation of a new voxel value, resulting in more accurate interpolation while multiplying the complexity. In volume rendering visualization, additional enhancements and manipulations are applicable by manipulating 3D attributes such as illumination, shading, and transparency [3, 28, 29]. This section starts with segmentation

methods that are fundamental to volume visualization, followed by transfer function techniques that enable the control of visualization attributes for volume enhancements.

9.5.1 Segmentation in Visualization

Segmentation is one of the core components in all image processing, and the outcome of the segmentation dictates the quality of the overall processing [31]. Often in volume rendering, the segmentation of image data is employed in order to separate the VOIs from the entire image. By selecting only the VOIs to be rendered, unnecessary rendering of voxels containing little information of interest can potentially be avoided. In surface rendering, segmentation of VOIs is a mandatory requirement for surface generation. The segmentation in volume rendering is also useful for two-level rendering [13], i.e., VOI can be rendered using direct volume rendering for optimal quality, while other structures of less importance can be rendered using surface-based techniques.

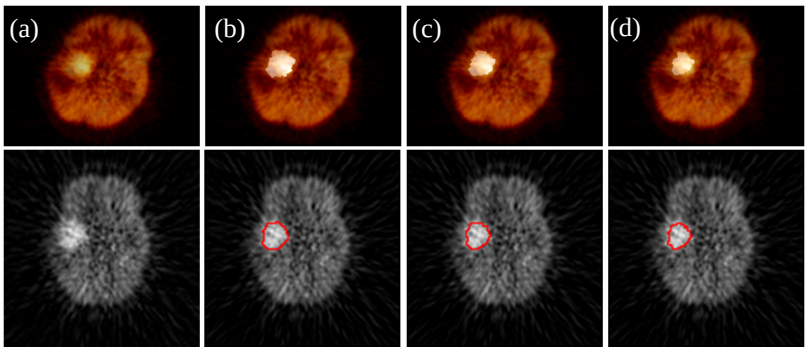

Figure 9.7 Interactive segmentation of dynamic PET images, with changes in the segmentation of the brain tumor's definition. Top row is the volume rendered images of the segmented structures. Bottom row is the 2D representation of the above volumes, with segmentation results highlighted in red outline. (a) is the original image, and (b) to (d) are the results from varying the segmentation parameter in real-time volume rendering.

Recent studies have demonstrated the potential of interactive segmentation in real-time 3D visualization[15, 55-57]. Such methods have the potential to provide a tremendous advantage as the segmentation needs not be restricted to being a pre-process, allowing the ability to see resultant rendering as segmentation changes are applied in real-time. Furthermore, these methods render only the segmented VOIs, without placing them in the context of surrounding structures. Bullitt and Aylward [55], reported a method of correcting segmentation errors from volume rendered VOIs by adjusting the radius of the viewable volume to reveal the

surrounding image, which allowed the physician the ability to correct for segmentation errors in volume rendering. The correction was made by rendering only the surrounding voxels within the radius of the VOIs. An interactive segmentation in volume rendered visualization (texture-based) was introduced in [58] for dynamic PET images. Here, the PET images were pre-segmented using fuzzy c-means cluster analysis, where a fuzzy logic algorithm assigns probabilistic weightings to every voxel representing the likelihood that voxel is a member of a particular segmented cluster (a feature). These weights were then used to interactively adjust the segmentation via computationally efficient thresholding as shown in Figure 9.7.

9.5.2 Transfer Function Specification

A considerable amount of research has been carried out in the application of transfer functions as a tool for feature selection in volume rendering of biomedical images. A typical feature classification, i.e. for a particular anatomical structure, in volume rendering visualization can be achieved by a transfer function specification [44, 59, 60]. The transfer function assigns properties such as color, via lookup table (LUT), and opacity to the voxel data in real-time to alter the visualization of the data. Transfer functions are often applied in 1D, i.e., to the voxels' values from the volume data as shown in Figure 9.8, which demonstrates the selection of lung nodules and bone in a CT volume.

Transfer functions can be extended to multiple dimensions, extending their capabilities to selectively visualize the data, through the control of gradient magnitude and second derivatives of voxels' intensity [44]. Increase in transfer function dimensions also increases complexity. Setting an ideal transfer function specification can be tedious and time-consuming, let alone confusing. Automation of transfer function specifications and improvements in user-interface design has the potential to ease their usage and improve efficiency. Konig and Groller introduced [61], a transfer function interface in which variations in data values, color, and opacity attributes from the input data was arranged in thumbnail views for user selection. Meanwhile Marks et al reported [62] a design gallery, which presents the user with all possible transfer function variations that are automatically generated and organized based on the input render. A novel and intuitive interface for controlling 3D transfer functions (based on data value, gradient magnitude, and second directional derivatives) was presented by Kniss et al [44] through a clever use of probing and classification widgets.

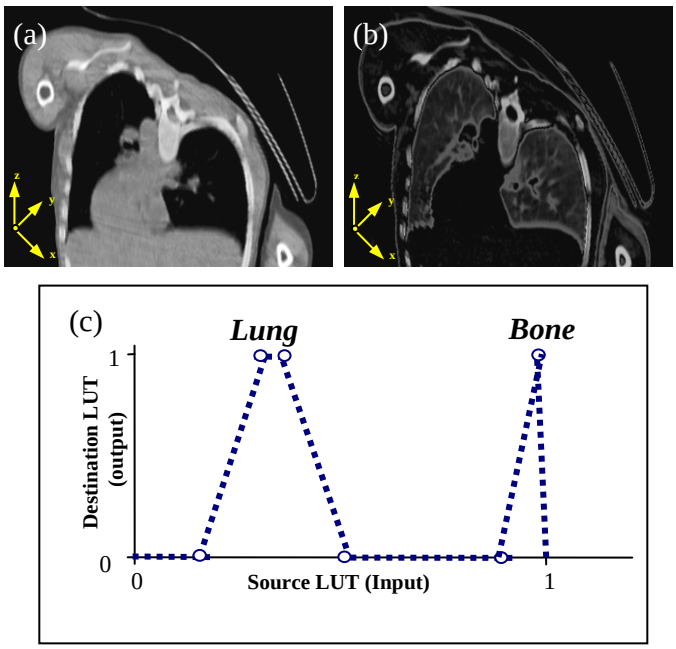

Figure 9.8 (a) CT volume of the lung rendered using texture-based rendering; (b) application of transfer function specified in (c) to volume in (a). A spike-based transfer function appears to produce the most appealing results in revealing structures of interest with gradual fade-out of nearby structures to the selected structures (lung and bone), in CT visualization

9.5.3 Spatial Transfer Function

Section 9.5.1 introduced the importance of incorporating segmentation into the visualization of biomedical data. Instead of rendering the segmentation result, it can also be used to specify the transfer function. This allows for the selection of a transfer function based on the spatial properties of the image volume rather than using LUT, i.e. via spatial transfer function, as described in the section above. An automated approach which removes or suppresses the less important objects of a scene to reveal more important underlying information was reported by investigators [30, 63]. They apply different compositing strategies based on the *importance* of the object calculated via image segmentation, and also provide control of the opacity of the voxels. Tzeng and Ma [64] discussed the use of fuzzy-based cluster analysis to segment and classify the data into individual objects. Their approach enables user-manipulation of rendering properties such as LUT and opacity, as well as fuzzy classification to render spatially-related voxels on an object-by-object basis. Zhou et al [65]reported distance-based volume rendering. Here, the distance of voxels to a focal point was used to control the optical properties of nearby voxels to emphasize the objects of interest and fade out other parts.

9.6 LARGE DATA VISUALIZATION AND OPTIMIZATION

Advances in medical scanning technologies are constantly pushing the limits of visualization techniques. Medical data storage commonly exceeds terabytes; such data require advanced computation and are often restricted to dedicated hardware configurations (i.e., parallel rendering) and require software-based optimization. Texture-based volume rendering in Section 9.3.3 is a software-based optimization that permits visualization of large medical data using the modern graphical processing unit's (GPU) ability to render large amount of textures. Although texture-mapped rendering produces interactive visualization of the data, the capacity of volume rendering is limited to the size of the texture memory on GPUs (currently at 512M for a single consumer GPU), and thus it is difficult to maintain an interactive frame rate when large data have to be rendered [39, 66]. This section highlights some of the solutions for optimizing volume rendering visualization of large biomedical data. Topics covered are multi-resolution, empty-space skipping, visibility culling, and early ray termination. A combination of these methods can be applied and was shown to produce particularly good results in texture-based volume rendering [40, 67, 68]. Finally, parallel rendering is introduced which achieves the computation of a single rendering using several networked computers.

9.6.1 Multi-Resolution

Multi-resolution volume rendering is a common optimization technique that allows applications to interactively render large volume data by assigning multiple levels of detail (LOD) in volume rendering [42, 66, 69]. LOD is used as a controller for the trade-off between quality of rendering and interactive performance, and is often applied to data such that when the entire dataset is viewed, a low LOD is applied, which progressively increases in detail as smaller sections of the data are selected for visualization. Multi-resolution rendering works by computing data structures that are constructed by decomposition of the data into *bricks* [66] as shown in Figure 9.9 which illustrates variation of rendering resolution according to different LODs. A typical data structure utilized in multi-resolution rendering is an *octree*, which is a simple yet efficient algorithm that breaks up the data into uniformly shaped rectangles. The bricks of different LOD levels are combined using interpolation to minimize the visual artefacts that may arise with different LODs.

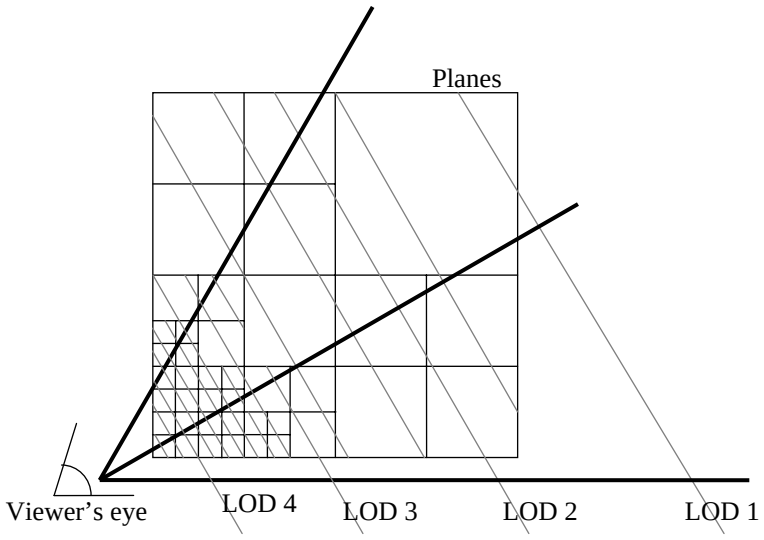

Figure 9.9 Multi-resolution for single volume data with view-aligned planes. Each brick (square box) in the figure has the same data dimension but they have different geometry sizes. i.e. Smaller bricks have the same amount of information used in rendering as the bigger bricks, thereby increasing the resolution and quality of data nearest to the viewing plane.

9.6.2 Empty Space Skipping

In volume rendering, at any particular rendering scene, only a portion of the entire data is visualized. Empty space skipping is a technique of avoiding the rendering of portions of the volume not being visualized, through the use of pre-computed data structures [67, 68]. The principle of empty space skipping is to divide the data into subsections as in multi-resolution rendering. In the rendering phase, visibility testing is performed to determine if the subsection is visible and should be rendered. There are many data structures that can be used to partition the data which has optimal performance under varying visualization requirements [67, 70].

9.6.3 Early Ray Termination

Early ray termination is probably the best known optimization technique in ray-casting volume rendering [68, 70]. As its name implies, this technique is based on the termination of a ray

when the contribution from that ray is minimal in the computation of the voxel to display. The early ray termination works only in ray-casting that is traversed front-to-back.

9.6.4 Parallel Rendering

Whereas the methods described above (Section 9.6.1 to 9.6.3) work on optimizing the computation of data via software-based approaches by trading between quality and performance, alternate approaches to render large data are to harness the computation from multiple graphic cards (hardware) for parallel rendering [40, 71-73]. One way to obtain parallel rendering is to split the rendering processes into several distinct functions which can be applied in series to individual data items. Such a technique is often favoured for surface rendering applications [71] which require several separate functional stages to complete rendering, i.e., segmentation, surface generation, and then rendering. Instead of computing functions in series, a more common method is to split the data into multiple streams which can then be operated independently and simultaneously, with each individual stream responsible for rendering a subsection of the data that are combined together. Parallel data streaming has been successfully applied on a variety of platforms, ranging from networks of PCs to massively parallel supercomputers for the visualization of large biomedical data via texture-based or direct volume rendering [40, 71].

9.7 DUAL-MODALITY PET/CT VISUALIZATION

In the visualization of dual-modality PET/CT, the PET images provide high sensitivity in tumor detection and tissue characterization, while the co-registered CT data provides the localizations of the anatomic landmarks and boundary definitions of tumors and organs [1, 74]. PET/CT images have led to a new paradigm in biomedical diagnosis and interpretation by enabling visualization of fused anatomical and functional structures. These dual-modality images, which consist of large image dimensions (full-body PET/CT data is typically in the range of 512×512×262 for CT and 128×128×262 for PET), have introduced interesting challenges for efficient 3D visualization [74] and stand to benefit significantly from advances in visualization technologies. The following section discusses the application of visualization techniques with PET/CT data.

9.7.1 Dual-Transfer Function

In PET/CT visualization, PET and CT data with independent LUTs are often required to be fused in real-time. These volumes may be fused together into a single volume prior to volume rendering [5], or these volumes may be individually rendered using different rendering

techniques, such as surface and direct volume rendering, which are then fused together as a pre-process to form a resultant volume [13]. These approaches all rely on the fusion of volumes to be a pre-process prior to volume rendering and are subsequently limited to the application of volume rendering manipulations, such as window/level and transfer function, to the fused volume and cannot be applied individually to its component scans. Alternatively, fusion of multiple volumes by data intermixing in the rendering phase in direct volume rendering was proposed by Cai and Sakas [75], thus allowing interactive and real-time fusion of the volumes.

The scenario of detecting tumor structures in a lung cancer patient illustrates the requirement to define transfer functions independently in PET/CT visualization. Selection of VOI from CT (such as, for example, by transfer function) and adjusting window/level via transfer functions of PET, necessitates manipulation to be applied to individual volumes, with the ability to visualize the changes in the fused volume in real-time. Physicians may benefit from visualizing the functional tumor apparent in the PET scan alongside corresponding anatomical structures for localization from the CT scan. Kim et al [76] suggested using dual-transfer functions for PET/CT images, thereby providing separate 1D LUT transfer functions independently to the PET and CT volumes to be controlled, with the resultant volumes being fused in real-time. Results from dual-transfer function use are shown in Figure 9.10. An axial view of a fused PET/CT volume is shown in Figure 9.10(a) with its dual-LUT transfer function in Figure 9.10(b). Equal fusion ratio was applied which was able to visualize the functional tumors (indicated by arrows) with surrounding anatomical structures. Figure 9.10(c) shows the result using a dual-LUT transfer function in Figure 9.10(d). The transfer function was applied to the PET component for the selection of the LUT range belonging to tumors (high LUT values). The slope of the transfer function enables gradual increase in the visualization of the voxel's LUT values as these voxel's LUT value reaches that of the tumor. Therefore, no voxels belonging to the tumors were erroneously excluded and avoided abrupt discontinuation of the selection of structures. For the CT, a trapezoidal transfer function was applied to visualize the boundaries of anatomical structures while minimizing spatial overlap with the functional tumor structures. These results demonstrated that the dual-LUT transfer function can be utilized to control the fusion between the PET and CT volumes to avoid overlap and highlight the structure of interest, in this case the tumors, while still retaining the anatomical context provided by the CT data. A different combination of dual-transfer functions is shown in Figure 9.10(e)-(f). The real-time dual-LUT transfer function applied to the PET/CT volume rendering provides immediate feedback on the accuracy of the feature selection and provides selective rendering of anatomical and functional structures. This permits improved visualization of the anatomical frame of reference and localization of the tumors when compared to the fusion of PET/CT in Figure 9.10(a).

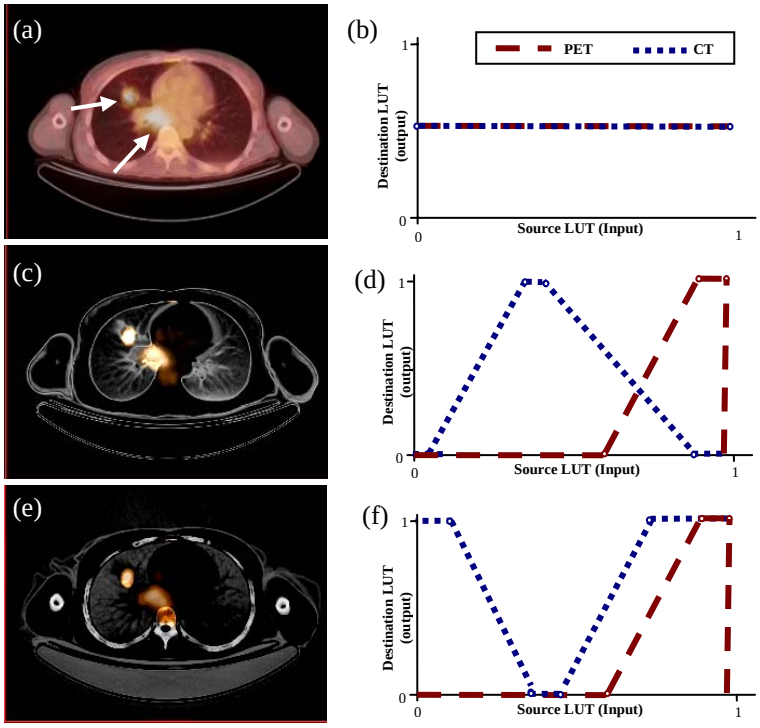

Figure 9.10 Application of dual-LUT transfer functions. (a) Axial view of a PET/CT with its dual-LUT transfer function in (b) set to equal fusion ratio. Tumors inside the lungs are highlighted by arrows. (c-d) Selection of tumors from PET and the surrounding anatomical lung boundary from the CT. (e-f) Identical PET transfer function with inverted and modified CT transfer function.

9.7.2 Spatial Transfer Function in PET/CT

Application of dual-transfer function in PET/CT data enables visualization of, for example, tumor structure from PET with surrounding anatomical structures from CT. However, due to the tumor having similar color values as functional organs (e.g. the liver), identification of the tumor using only a transfer function based on LUT is limited. For such cases, if the location of the tumor is known, a spatial transfer function can be applied which allows control of the viewable range of the segmented structure according to the spatial distance of voxels to the segmented structure [30, 64, 65]. Kim et al [77] presented an application of spatial transfer to PET/CT data, where tumors identified on PET were segmented as a pre-process (Figure 9.11(a)). The segmentation result was then used to construct a *distance map* according to spatial distance of voxels to the segmented structure based on the algorithm introduced by

Fichtinger et al [4]. These distances were used to calculate the weights assigned to each individual voxel in the PET, as illustrated in Figure 9.11(b). These weights were then multiplied with the PET, during voxel-by-voxel data-intermixing with CT data [75]. Several parameters in this algorithm can be fine-tuned. Firstly, the weight applied to the PET can be controlled such that greater emphasis can be placed on voxels near the tumors. Secondly, cutoffs can be added, such as the red circles in Figure 9.11(b), which can be used to control the appliance of the weights to only the voxels that reside within these cutoffs. The result of the spatial transfer function is shown in Figure 9.11(c) which shows tumors to be clearly depicted with their surrounding anatomical CT structures. Here, a cutoff of 30 voxels was applied. The spatial transfer function has the advantage of visualizing spatially-related voxels to the segmented tumor and enables gradual increase in the transition of the voxel's fusion ratio as the voxels approach the tumor.

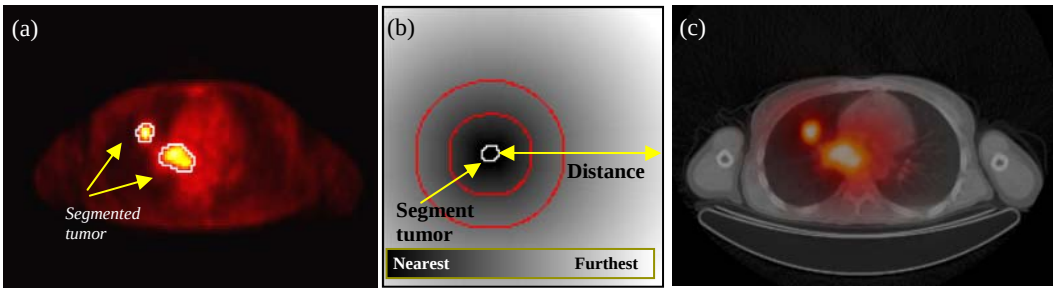

Figure 9.11 Overview of the segmentation-based spatial transfer function applied to PET/CT visualization. (a) Original PET image (single axial-view slice) with segmented structures highlighted (arrow); (b) Distance map calculated from the segmented result; (c) PET/CT fused result from spatial transfer function using (b).

9.7.3 Interactive Segmentation and Volume Interchange in PET/CT

Instead of using segmentation results for the spatial transfer function, these segmentation results can also be used directly in volume rendering. Rendering the segmented VOI as an independent volume that is transparent, together with the original PET and CT volumes, is potentially useful in highlighting the VOIs. In PET/CT, such ability can be further extended to allow interchange of the volumes as shown in Figure 9.12 [41]. Here, the segmented VOIs from the PET data were shown rendered together with either the PET (Figure 9.12(b)) or the CT (Figure 9.12(c)). This was possible through the utilization of the high memory bandwidth capacity of modern graphic hardware to rapidly transfer 3D textures from the system memory

into the graphic memory, and was used to replace an old volume in the graphic memory with a new volume in real-time volume rendering.

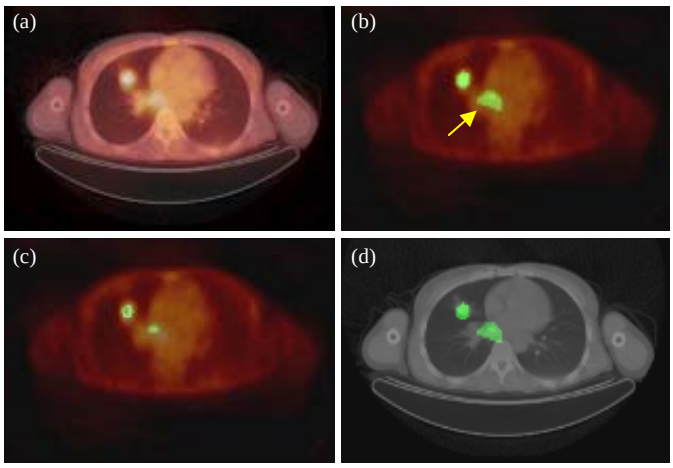

Figure 9.12 (a) Volume rendering of PET/CT image using texture-based volume rendering. (b) Automated FCM segmentation result with segmented tumor structures fused with PET. (c) Result from varying the PET segmentation parameter to select voxels that more closely resemble the tumor. (d) Segmentation result of (b) fused with CT. All volumes have been fused with equal fusion ratios. The transparency level and the LUT of the segmented volumes can be adjusted to reduce obscuration of underlying structures relevant for the interpretation of the images and segmentation results.

9.8 DATA DISPLAY DEVICES

Previous sections of this chapter have discussed algorithms and methods that are used to generate volume rendered visualizations. The hardware devices which display these visual outputs are equally important. The utilization of digital displays in biomedical data visualization in recent times has been accelerated by the acceptance of PACS [78] as a digital image management system, as well as due to the additional abilities that digital images provide, such as image processing and volume visualization. The effectiveness of all diagnostic imaging visualization is affected by the quality of the display devices [79]. In current diagnostic workstation, it is common to find multiple LCD screens for use in visualizing biomedical data in 2D and 3D views. These screens typically have resolutions of 1600x1200 and greater, with pixel depth of 24bit colour, and thus providing sufficient display capacity for modern medical imaging modalities. Apart from LCDs, more advanced display devices are

being designed and developed for diagnostic rooms and operating theatres. These include stereoscopic displays for virtual reality and depth display devices.

9.8.1 Stereoscopic Visualization

On a 2D screen, true 3D visualization can be obtained only by displaying the data stereoscopically, thereby creating the illusion of depth. This technique has been widely adopted in biomedical visualization for creating virtual reality environments and has found large uses in surgical simulations [14, 50, 80]. The fundamentals of stereoscopy are to present slightly different images to the left and right human eyes, such that each eye can only see a single image. The most routinely utilized stereoscopy equipment in biomedical visualization are the infrared synchronized shutter glasses that allow the left/right eye images to be quickly alternated (generally at 60Hz). Stereoscopic visualization has also been applied to immersive environments, with examples such as Immersadesk [81] and the CAVE environment [82, 83]. An alternative to the shutter glasses, which restricts the viewing capability to a single user per computer screen, the study in [80] proposed the use of polarized filters to separate the stereo information, which than allows multiple users to visualize from the same screen with relatively cheap polarized glasses.

As the requirement of wearing glasses was an impediment for adoption of stereoscopic visualization, there has been a significant research by display manufacturers (in particular, Sharp [84] and Philips [85]) aiming to allow realistic 3D viewing without the need for 3D glasses. Glasses-free 3D displays, also known as auto-stereoscopic displays, work by the device being able to display two separate images concurrently. These images have the same disparity and the display device uses switchable parallax barrier technology [84] to ensure that each eye only sees one of these images, and in doing so, creating the illusion of depth without the need for glasses to separate the different signals.

9.8.2 Depth Projection

Stereoscopy is a visualization technique that creates the illusion of depth, and therefore the display has no physical depth definition [86]. This has major limitations in creating true depth perception. A device produced by LightSpace Technologies, the DepthCube [87], is a solid-state multi-planar volumetric display that uses a high-speed video projector to project a sequence of slices of the 3D image into multi-planar optical elements. The individual slices in the multi-planar views are than anti-aliased to result in smooth images.

9.9 APPLICATIONS OF BIOMEDICAL VISUALIZATION

Modern visualization and display technologies are enabling the development of clinical applications which provides new approaches to interact and interpret biomedical data, such as in virtual biopsy [88], motion activity visualization [89], and in radiotherapy planning [8]. These applications are capable of providing complete 3D views of the human body that are rendered in real-time to volume navigations and manipulations. Furthermore, advances in visualization and display technologies have also accelerated the developments of computer-integrated surgery (CIS), IGS, and CAD systems which are detailed in Chapters 16 and 18. This section summarizes some of the researches that have been made possible through the utilization of the cutting-edge biomedical data visualization technologies.

Research efforts in biomedical data visualization have been primarily focused on reconstructing the data into 3D volumetric views. However, many other information and uses can be obtained from the same data, for example, by tracer kinetic modeling, as well as the modeling of different elements (such as bones, cartilage, ligaments, muscles and tendons) and their inter-relationships [89-94]. Study by Feng et al [90, 91] discussed the medical parametric imaging which requires modeling and parameter estimation for certain metabolic, pharmacokinetic, endocrine or other biochemical systems at voxel by voxel level. It is an important technique which provides image-wide quantifications of physiological and biochemical functions and allows the distributions of these functions corresponding to anatomic structures to be visualized. For more information about parametric imaging, please refer to Chapter 6. In another study, the visualization and understanding of functionalities of human articulation of the shoulders was presented by Magnenat-Thalmann et al [93]. Here, a generic 3D shoulder was constructed as a physically-based model along with other elements, which was used to produce a simulation of movements and deformations. This project used the visible human dataset (VHD) [95] and pioneered the innovative use of 3D visual simulation in the medical computing field. This project is currently expanding to the study of the hips articulation using dynamic MRI images. Study by Benjamin et al [89] presented the ability to visualize an accurate estimate of joint kinetics for the understanding of joint motion activity. The modeling and simulations of the hip articulation were validated using medical data and demonstrated the potential to provide physicians with the benefit of having the technology that enables visualization and observation of motion activity (in 3D) for use in diagnosis of problems of the human body articulations. Further work of this project will investigate the biomechanical model of the hip to simulate and visualize the motion of the hip and to understand the possible malfunctioning of the articulation of individual patients. The biomechanical model will rely on the inter-relations of the various anatomic elements involved in the joint articulation and their influence on the range of motion of the hip [89, 94].

The Visible Human Project (VHP) [95] is an effort to create a detailed data set of cross-sectional photographs of the human body, in order to facilitate anatomy visualization applications for education, research, and for clinical diagnosis. The importance of this project has lead to the formation of many research projects, such as the motion modeling study above,

and also to the development of Korean [96] and Chinese visible human projects [97]. In VHD related researches, the Biomedical Imaging Resource (BIR) at the Mayo Clinic was one of the first laboratories to be involved, and during the last decade, they has produced many novel 3D visualization and image processing methods (i.e., rendering, registration, segmentation, modeling, classification), and evaluated the effectiveness of these methods for eventual applications in clinical diagnosis and therapy [20, 98, 99]. In the group's recent study, Robb et al developed an anesthesiology training simulation system, in close collaboration with clinicians, using the VHD as the patient. This system enabled the simulation of examinations using 3D visualization of the relationships among the anatomical structures, in addition to its use in needle insertion practice. Training system was built through an immersive environment (virtual operating theater) created through the use of a head tracking system, head mounted display, needle tracking system, and haptic input/feedback. The user may interact with the virtual patient using a wand or haptic feedback devices that provides a sense of touching the patient [100, 101]. This system has been extended to medical simulation system used for popliteal nerve (knee) block and prostate brachytherapy training [98] which can also be used with individual patient scans as well as the VHD.

An innovative project which deals with full body virtual autopsies was recently reported by Ljung et al [88]. This study reported a procedure based on interactive 3D visualization of large scale, high resolution CT data of human cadavers for virtual autopsies, which demonstrated potential to provide key information in guiding criminal investigations. Unique challenge in this project arises from the large volume of data acquired from multi detector CT (MDCT) which captures 8000 trans-axial images for a full body imaging (data size of several gigabytes). In order to interactively visualize these MDCT images in real-time, a state-of-the-art volume rendering pipeline was developed which applied and refined several visualization optimization algorithms including transfer function based adaptive LOD [102], interblock interpolation [103], and a single-pass ray-casting of multi-resolution volumes [104], as well as memory management techniques. Furthermore, this study introduces an extension to the GPU-based ray-casting algorithm that enabled efficient dual transfer function rendering for fast localization of, for example, metal fragments. The described autopsy procedure was evaluated using examples from the routine forensic examinations and demonstrated great potential as an imaging application that complement to standard autopsies by enabling broad and systematic examination of the full body in forensic investigations.

Diffusion tensor imaging (DTI) is a MRI technique (also known as DT-MRI) that enables the measurement of the diffusion of water in tissues such as bone, muscle, and white matter of the brain. Here, diffusion refers to the ability of the water molecules to transport from one part of a biological system to another, in a random molecular motion (moving faster in some direction than others) [105, 106]. This geometrical feature of DT-MRI enables quantitative characterization and visualization of the local structures in tissues. Due to the large information available in DT-MRI, visualization approaches for DT-MRI is a challenging field with many different approaches to represent the DT-MRI image. One tool for interpreting this image is the use of glyph – a parameterized icon that represents the data with its shape, color, texture, location etc., and can be used to represent the diffusion properties [107]. In a recent

study, Kindlemann and Westin [108] proposed a *glyph packing* method which used a particle system with anisotropic potential energy profiles to arrange glyphs, rendered using texture-based technique, into a dense pattern that displays some of the visual continuity of texture-based visualizations, while maintaining the ability to discern the full tensor information at each glyph. This study demonstrated that the use of visualization techniques can complement and also reveal additional information that can aid in the diagnosis and interpretation of DT-MRI images.

9.10 SUMMARY

With continuous development in image acquisition technologies that are resulting in ever increasing data sizes, the need for visualization of these data in an efficient and intelligent manner are becoming crucial for clinical diagnosis and image understanding. In line with advances in imaging modalities, development of new theories and refinement of techniques in biomedical data visualization are improving the way these data are utilized and interpreted. Furthermore, new discoveries in visualization technologies are enabling alternate approaches to conventional methods. This chapter introduced the visualization and display technologies that are currently being employed in clinical applications, as well as techniques that have the potential to enhance and provide improved diagnostic capabilities. Starting from 2D and 3D visual methods, core visualization technologies were discussed, including navigation interfaces, volume enhancement and manipulations, and large data visualization. An example PET/CT visualization was then presented that took advantage of many of the techniques discussed in this chapter. Finally, next generation in display devices and technologies were briefly covered with regard to their potential clinical applications. Future developments in visualization and displays for biomedical data will carry on the development of new technologies and produce new ways to improve the display and interpretation of biomedical data.

Acknowledgement

The authors are grateful to the support from ARC and PolyU/UGC grants.

9.11 EXERCISES

1. In Figure 9.3, renders of CT volume data using three different techniques are shown. List the main differences between the rendering techniques and their advantages/disadvantages in biomedical data visualization.

2. Volume rendering visualization introduces several rendering attributes such as shadowing and transparency that create more realistic display of biomedical data. Together with conventional enhancement tools such as window/level, these attributes provide powerful tools for volume enhancement. List and give examples of how these attributes are used for visualization enhancement.
3. Transfer function specification is applicable in 1D, 2D, or in greater dimensions. This chapter discusses multi-dimensional transfer functions that are used for LUT manipulations. Discuss how multi-dimensional transfer functions can be applied in the spatial domain.
4. Segmentation is a key component of visualization. However, interactive segmentation is often too computationally intensive for real-time performance in volume rendering. State how this problem can be minimized, in terms of software and hardware solutions.
5. State the basic principles of parallel direct volume rendering. What is the *bottleneck* in parallel rendering of large biomedical data?
6. Section 9.4.2 presents different input devices that may be used for navigating volumetric biomedical data. Design and analyze a setup of input devices for use in controlling the multi-dimensional transfer function of multivariate biomedical data.

9.12 BIBLIOGRAPHY AND REFERENCES

1. Ratib, O., *PET/CT image navigation and communication.* J. Nucl. Med, 2004. **45**(1): p. 46S-55S.
2. Bankman, I.N., *Handbook of Medical Imaging*. 2000, San Diago: Academic Press.
3. Meißner, M., et al., *Volume Visualization and Volume Rendering Techniques* In: Tutorials 6, Eurographics, 2000.
4. Fichtinger, G., et al., *System for Robotically Assisted Prostate Biopsy and Therapy with Intraoperative CT Guidance.* Academic Radiology, 2002. **9**(1): p. 60-74.
5. Rosset, A., L. Spadola, and O. Ratib, *OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images.* J. Digital Imaging, 2004. **17**(3): p. 205-216.
6. Shahidi, R., R. Tombropoulos, and R.P. Grzeszczuk, *Clinical applications of three-dimensional rendering of medical data sets.* Proceedings of IEEE, 1998. **86**: p. 555-565.
7. Beltrame, F., et al. *Three-dimensional visualization and navigation tool for diagnostic and surgical planning applications*. in *Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures*. 2001. San Diego, CA, USA: SPIE.
8. Ciernik, I.F., et al., *Radiation treatment planning with an integrated positron emission and computer tomography (PET/CT): a feasibility study.* Int. J. Radiation Oncology Biol. Phys., 2003. **57**: p. 853-863.
9. Jani, A.B., J.-S. Irick, and C. Pelizzari, *Opacity transfer function optimization for volume-rendered computed tomography images of the prostate1*. Academic Radiol., 2005. **12**(6): p. 761-770.
10. Gering, D.T., et al., *An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging*, in *Proc. Medical Image Computing and Computer Assisted Intervention, Springer Verlag*. 1999. p. 809–819.
11. Loh, Y.C., et al., *Surgical planning system with real-time volume rendering*, in *Proc. IEEE Int. Workshop Med. Imag. Augmented Reality*. 2001. p. 259-261.
12. Rosset, A., et al., *Informatics in Radiology (infoRAD): Navigating the Fifth Dimension: Innovative Interface for Multidimensional Multimodality Image Navigation.* Radiographics, 2006. **26**(1): p. 299-308.

13. Hauser, H., et al., *Two-level volume rendering.* IEEE Trans. Visualization and Computer Graphics, 2001. **7**(3): p. 242-252.
14. Dai, P.-D., et al., *A virtual laboratory for temporal bone microanatomy*. IEEE Computing in Science & Engineering, 2005. **7**(2): p. 75-79.
15. Bornik, A., et al. *A Hybrid User Interface for Manipulation of Volumetric Medical Data*. 2006.
16. Frohlich, B., et al., *On 3D input devices.* Computer Graphics and Applications, IEEE, 2006. **26**(2): p. 15-19.
17. Sherbondy, A.J., et al., *Alternative Input Devices for Efficient Navigation of Large CT Angiography Data Sets.* Radiology, 2005. **234**(2): p. 391-398.
18. *Chapel Hill Volume Rendering Test Data Set*.
19. *MedView*. [PET/CT viewer] [cited; 11:[Available from: http://www.medimage.com/pet-ct-software.html.
20. Robb, R., A., *Visualization in biomedical computing.* 1999. **25**(13-14): p. 2067-2110.
21. Lohmann, G., *Volumetric Image Analysis*. 1 ed. 1998, New York: John Wiley & Sons Ltd.
22. *ImageJ - Image Processing and Analysis in Java*. [cited; Available from: http://rsb.info.nih.gov/ij/.
23. Kai, U.B., *Volume Viewer*. 2005, Internationale Medieninformatik: Berlin, Germany
24. Lorensen, W., E. and H. Cline, E. , *Marching cubes: A high resolution 3D surface construction algorithm*, in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 1987, ACM Press.
25. Schroeder, W., K. Martin, and B. Lorensen, *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*. 3rd Edition ed. 2003: Kitware, Inc. publishers
26. Udupa, J., K. , *Multidimensional digital boundaries*. 1994, Academic Press, Inc. p. 311-323.
27. Tiede, U., et al., *Investigation of medical 3D-rendering algorithms.* IEEE Computer Graphics and Applications, 1990. **10**(2): p. 41-53.
28. Watt, A.H., *3D Computer Graphics* 3ed. 1999: Addison Wesley.
29. Shreiner, D., et al., *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL*. 5th ed. 2005: Addison-Wesley Professional.
30. Viola, I., A. Kanitsar, and M.E. Groller, *Importance-driven feature enhancement in volume visualization.* IEEE Trans. Visualization and Computer Graphics, 2005. **11**(4): p. 408-418.
31. Harders, M. and G. Szekely, *Enhancing human-computer interaction in medical segmentation.* Proceedings of the IEEE, 2003. **91**(9): p. 1430-1442.
32. Westermann, R. and B. Sevenich. *Accelerated volume ray-casting using texture mapping*. in *IEEE Visualization*. 2001.
33. Kiefer, G., H. Lehmann, and J. Weese, *Fast maximum intensity projections of large medical data sets by exploiting hierarchical memory architectures.* IEEE Trans. Information Technology in Biomedicine, 2006. **10**(2): p. 385-394.
34. Venema, H.W., F.J.H. Hulsmans, and G.J. den Heeten, *CT Angiography of the Circle of Willis and Intracranial Internal Carotid Arteries: Maximum Intensity Projection with Matched Mask Bone Elimination--Feasibility Study*. 2001. p. 893-898.
35. Marks, M.P., et al., *Diagnosis of carotid artery disease: preliminary experience with maximum-intensity-projection spiral CT angiography.* American Journal of Roentgenology, 1993. **160**(6): p. 1267-1271.
36. Laur, D. and P. Hanrahan, *Hierarchical splatting: a progressive refinement algorithm for volume rendering*, in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 1991, ACM Press.
37. Lacroute, P. and M. Levoy. *Fast volume rendering using a shear-warp factorization of the viewing transformation.* in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 1994
38. Sweeney, J. and K. Mueller, *Shear-Warp deluxe: the Shear-Warp algorithm revisited*, in *Proceedings of the symposium on Data Visualisation 2002*. 2002, Eurographics Association: Barcelona, Spain.
39. Bhanirantka, P. and Y. Demange. *OpenGL volumizer: a toolkit for high quality volume rendering of large data sets*. in *IEEE symposium on Volume visualization and graphics*. 2002.
40. Kniss, J., et al., *Interactive texture-based volume rendering for large data sets.* IEEE Computer Graphics and Applications, 2001. **21**(4): p. 52-61.
41. Kim, J., et al., *Real-Time Volume Rendering Visualization of Dual-Modality PET/CT Images with Interactive Fuzzy Thresholding Segmentation* IEEE Trans. Information Technology in Biomedicine, 2006.

42. Manfred, W., et al., *Level-of-detail volume rendering via 3D textures*, in *Proc. IEEE symposium on Volume visualization*. 2000, ACM Press: Salt Lake City, Utah, United States.
43. Tomandl, B.F., et al., *Local and Remote Visualization Techniques for Interactive Direct Volume Rendering in Neuroradiology*. Radiographics, 2001. **21**(6): p. 1561-1572.
44. Kniss, J., G. Kindlmann, and C. Hansen, *Multidimensional transfer functions for interactive volume rendering.* IEEE Trans. Visualization and Computer Graphics, 2002. **8**(3): p. 270-285.
45. Post, F.H., et al., *The State of the Art in Flow Visualisation: Feature Extraction and Tracking.* Computer Graphics Forum, 2003. **22**(4): p. 775-792.
46. Laramee, R.S., et al., *The State of the Art in Flow Visualization: Dense and Texture-Based Techniques.* Computer Graphics Forum, 2004. **23**(2): p. 203-221.
47. Tory, M.K., T. Moeller, and M.S. Atkins. *Visualization of time-varying MRI data for MS lesion analysis*. in *Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures*. 2001. San Diego, CA, USA: SPIE.
48. Thune, N. and B. Olstad. *Visualizing 4-D medical ultrasound data*. in *Proc. IEEE Visualization*. 1991.
49. Neubauer, A., et al., *Advanced Virtual Endoscopic Pituitary Surgery.* IEEE Trans. Visualization and Computer Graphics, 2005. **11**(5): p. 497-507.
50. Gronningsaeter, A., et al., *Initial experience with stereoscopic visualization of three-dimensional ultrasound data in surgery.* Surgical Endoscopy, 2000. **14**(11): p. 1074-1078.
51. *Space Pilot*. 2006 [cited; Available from: www.hp.com.au.
52. Myers, B.A., *Using multiple devices simultaneously for display and control.* IEEE Personal Communications, 2000. **7**(5): p. 62-65.
53. Kim, Y. and A. Varshney, *Saliency-guided Enhancement for Volume Visualization.* IEEE Trans. Visualization and Computer Graphics, 2006. **12**(5): p. 925-932.
54. Nikolaidis, N., *3-D Image Processing Algorithms*. 1 ed. 2001, New York: John Wiley & Sons.
55. Bullitt, E. and S.R. Aylward, *Volume rendering of segmented image objects.* IEEE Trans. Medical Imaging, 2002. **21**(8): p. 998-1002.
56. Yoo, S.-S., et al., *Interactive 3-dimensional segmentation of MRI data in personal computer environment.* Journal of Neuroscience Methods, 2001. **112**(1): p. 75-82.
57. Vosilla, L., et al. *An interactive tool for the segmentation of multimodal medical images*. 2000.
58. Kim, J., et al., *Interactive fuzzy temporal thresholding for the segmentation of dynamic brain PET images.* Journal of Cerebral Blood Flow & Metabolism, 2005. **25**: p. S620.
59. Vereda, P., et al., *Visualization of Boundaries in Volumetric Data Sets Using LH Histograms.* IEEE Trans. Visualization and Computer Graphics, 2006. **12**(2): p. 208-218.
60. Levoy, M., *Display of surfaces from volume data.* IEEE Computer Graphics and Applications, 1988. **8**(3): p. 29-37.
61. König, A. and E. Gröller, *Mastering Transfer Function Specification by Using VolumePro Technology*, in *Spring Conference on Computer Graphics*. 2001. p. 279-286.
62. Marks, J., et al., *Design galleries: a general approach to setting parameters for computer graphics and animation*, in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 1997, ACM Press/Addison-Wesley Publishing Co.
63. Viola, I., et al., *Importance-Driven Focus of Attention.* IEEE Trans. Visualization and Computer Graphics, 2006. **12**(5): p. 933-940.
64. Tzeng, F.-Y. and K.-L. Ma. *A Cluster-space Visual Interface for Arbitrary Dimensional Classification of Volume Data*. in *IEEE TVCG Symposium on Visualization*. 2004.
65. Zhou, J., A. Doring, and K.D. Tonnies. *Distance based enhancement for focal region based volume rendering*. in *Proceedings of Bildverarbeitung fur die Medizin'04*. 2004.
66. Guthe, S., et al. *Interactive rendering of large volume data sets*. in *Proc. IEEE Visualization*. 2002.
67. Li, W., K. Mueller, and A. Kaufman. *Empty space skipping and occlusion clipping for texture-based volume rendering*. in *Proc. IEEE Visualization*. 2003.
68. Kruger, J. and R. Westermann. *Acceleration techniques for GPU-based volume rendering*. in *Proc. IEEE Visualization*. 2003.

69. Ruediger, W., *A multiresolution framework for volume rendering*, in *Proceedings of the 1994 symposium on Volume visualization*. 1994, ACM Press: Tysons Corner, Virginia, United States.
70. Levoy, M., *Efficient ray tracing of volume data*. 1990, ACM Press. p. 245-261.
71. Crockett, T.W., *An introduction to parallel rendering.* Parallel Computing, 1997. **23**(7): p. 819-843.
72. Molnar, S., et al., *A sorting classification of parallel rendering.* IEEE Computer Graphics and Applications, 1994. **14**(4): p. 23-32.
73. Bhaniramka, P., P.C.D. Robert, and S. Eilemann. *OpenGL multipipe SDK: a toolkit for scalable parallel rendering*. in *Proc. IEEE Visualization*. 2005.
74. Wernick, M.N. and J.N. Aarsvold, *Emission tomography - the fundamentals of PET and SPECT*. 1 ed. 2004, London: Elsevier Academic Press.
75. Cai, W. and G. Sakas, *Data intermixing and multi-volume rendering.* Computer Graphics Forum, 1999. **18**(3): p. 359-368.
76. Kim, J., S. Eberl, and D. Feng, *Visualization of Dual-Modality Anatomical and Functional Rendered Volumes with Image Fusion using a Dual-Lookup Table Transfer Function.* IEEE Computing in Science and Engineering, 2006 (in press).
77. Kim, J., S. Eberl, and D. Feng. *Multi-Modal Medical Visualization based on Spatial Transfer Function*. in *Proc. IEEE Visualization 2006*. 2006 (accepted).
78. Huang, H.K., *PACS, image management and imaging informatics*, in *Multimedia Information Retrieval and Management*. 2003, Springer-Verlag: New York. p. 476.
79. Badano, A., *AAPM/RSNA Tutorial on Equipment Selection: PACS Equipment Overview: Display Systems*. 2004. p. 879-889.
80. Jones, S.T., S.E. Parker, and C.C. Kim, *Low-cost high-performance scientific visualization.* IEEE Computing in Science & Engineering 2001. **3**(4): p. 12-17.
81. Marek, C., et al., *The ImmersaDesk and Infinity Wall projection-based virtual reality displays*. 1997, ACM Press. p. 46-49.
82. Demiralp, C., et al., *CAVE and fishtank virtual-reality displays: a qualitative and quantitative comparison.* IEEE Trans. Visualization and Computer Graphics, 2006. **12**(3): p. 323-330.
83. Cruz-Neira, C., D. Sandin, J. , and T. DeFanti, A. , *Surround-screen projection-based virtual reality: the design and implementation of the CAVE*, in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 1993, ACM Press.
84. *Sharp3D*. [cited; Available from: http://www.sharp3d.com/.
85. *Philips*. [cited; Available from: http://www.research.philips.com/.
86. Balasubramanian, K., *On the realization of constraint-free stereo television.* IEEE Trans. Consumer Electronics, 2004. **50**(3): p. 895-902.
87. Sullivan, A., *3-Deep: new displays render images you can almost reach out and touch.* IEEE Spectrum, 2005. **42**(4): p. 30-35.
88. Ljung, P., et al., *Full body virtual autopsies using a state-of-the-art volume rendering pipeline.* IEEE Trans. Vis. Comput. Graph., 2006. **12**(5): p. 869-76.
89. Gilles, B., et al., *Bone Motion Analysis From Dynamic MRI: Acquisition and Tracking.* Academic Radiol., 2005. **12**: p. 1285-1292.
90. Feng, D., Z. Wang, and S.-C. Huang, *A study on statistically reliable and computationally efficient algorithms for generating local cerebral blood flow parametric images with positron emission tomography.* IEEE Trans. Medical Imaging, 1993. **12**(2): p. 182-188.
91. Feng, D., *Information Technology Applications in Biomedical Functional Imaging.* IEEE Trans. Information Technology in Biomedicine, 1999. **3**(3): p. 221-230.
92. Wen, L., et al., *Fast and Reliable Estimation of Multiple Parametric Images Using an Integrated Method for Dynamic SPECT.* IEEE Trans. Medical Imaging, 2007. **26**(2): p. 179-189.
93. Magnenat-Thalmann, N. and F. Cordier, *Construction of a human topological model from medical data.* IEEE Trans. Information Technology in Biomedicine, 2000. **4**(2): p. 137-143.

94. *MIRALAB*. 2007 [cited; Available from: http://www.miralab.unige.ch/.
95. Ackerman, M.J., *The Visible Human Project.* Proceedings of IEEE, 1998. **86**(3): p. 504-511.
96. Park, J.S., et al., *Visible Korean Human: Improved serially sectioned images of the entire body*. IEEE Trans. Medical Imaging, 2005. **24**(3): p. 352 - 360
97. Zhang, S.-X., P.-A. Heng, and Z.-J. Liu. *Chinese Visible Human Project: Dataset Acquisition and Its Primary Applications*. in *Proc. IEEE EMBC*. 2005.
98. Robb, R.A. *Biomedical Imaging: Past, Present and Prediction*. in *Proc. Symp. Intelligent Assistance in Diagnosis of Multi-Dimensional Medical Images*. 2005.
99. Robb, R.A. and D.P. Hanson, *Biomedical image visualization research using the Visible Human Datasets.* Clinical Anatomy, 2006. **19**(3): p. 240 - 253.
100. Martin, D.P., D.J. Blezek, and R.A. Robb, *Simulating lower extremity nerve blocks with virtual reality*. Techn. Reg. Anesth. Pain Manage., 1999. **3**: p. 58-61.
101. Pommert, A., et al., *Creating a high-resolution spatial/symbolic model of the inner organs based on the Visible Human.* Medical Image Analysis, 2001. **5**(3): p. 221-228.
102. Ljung, P., et al. *Transfer Function Based Adaptive Decompression for Volume Rendering of Large Medical Data Sets* in *IEEE Symp. Volume Visualization and Graphics* 2004.
103. Ljung, P., C. Lundström, and A. Ynnerman. *Multiresolution Interblock Interpolation in Direct Volume Rendering*. in *Proc. Eurographics/IEEE Symp. Visualization*. 2006.
104. Ljung, P. *Adaptive sampling in single pass, GPU-based raycasting of multiresolution volumes*. in *Proceedings Eurographics/IEEE Workshop on Volume Graphics*. 2006.
105. Kindlmann, G., D. Weinstein, and D. Hart, *Strategies for direct volume rendering of diffusion tensor fields.* IEEE Trans. Vis. Comput. Graph., 2000. **6**(2): p. 124-138.
106. Westin, C.-F., et al., *Processing and visualization for diffusion tensor MRI.* Medical Image Analysis, 2002. **6**: p. 93-108.
107. Pierpaoli, C. and P.J. Basser, *Toward a quantitative assessment of diffusion anisotropy.* Magn. Reson. Med. , 1996. **37**(6): p. 893-906.
108. Kindlmann, G. and C.-F. Westin, *Diffusion Tensor Visualization with Glyph Packing* IEEE Trans. Vis. Comput. Graph., 2006. **12**(5): p. 1329-1336.

9.13 INDEX

CHAPTER 9 DATA VISUALIZATION AND DISPLAY