CHAPTER 13

IMAGE VISUALIZATION

Visualization is an important aspect of human understanding of objects and structures. Visualization has always been an active field of research in computer image processing and graphics. In medical imaging, the visualization of internal physiological structures is the fundamental objective. However, visualization methods in medical imaging go far beyond the two-dimensional (2-D) and three-dimensional (3-D) imaging techniques. The visual observations in multidimensional medical imaging are often augmented by quantitative measurements and analysis for better understanding. Often the features extracted from analysis of medical images are incorporated in the visualization methods. For example, boundaries of a brain tumor can be superimposed on a 2-D brain image for better visualization of its local position and geometrical structure.

Recent advances in computing hardware and software capabilities have allowed the multidimensional visualization methods to incorporate multilayered representations of surface and volume structures in an interactive manner (1–5). Such methods have been found to be very useful in multimodality image fusion and visualization, allowing for better understanding of physiological structures as applied in diagnostic and intervention protocols including image-guided surgery (3–8). More recently, the multilayered image representation and visualization methods have led to a virtual reality (VR) experience where internal structures can be interactively explored, mapped, and visualized. VR provides an interactive human–computer interface that facilitates the visualization of 3-D computer-generated scenes (3). For example, virtual endoscopy has demonstrated a new role in the surgical planning system, providing adjunctive diagnostic and treatment interventional capabilities (3).

Multidimensional medical image visualization methods can be broadly categorized into four groups:

1. Feature enhanced 2-D image display methods
2. Stereo vision and semi 3-D display methods
3. Surface- and volume-based 3-D display methods
4. Interactive VR visualization methods

Medical Image Analysis, *Second Edition*, by Atam P. Dhawan

13.1. FEATURE-ENHANCED 2-D IMAGE DISPLAY METHODS

In the simplest form, 2-D gray-level images can be visualized using an appropriate medium such as radiological films (that can be put on a light box for better visualization), hard copy prints or photographs, and computer display devices. Using computerized processing, 3-D images can be visualized as a collection or stack of 2-D images that are acquired as cross-sectional images along a selected axis. For example, Figure 12.7 in Chapter 12 shows 2-D serial cross-sectional images of 3-D MR and positron emission tomography (PET) brain image data sets (7). Such a method is essentially a 2-D visualization. Specific features of an image such as edges or segmented regions can be superimposed on the same or a corresponding cross-sectional image from a different modality. Figure 9.7 shows a combined visualization of the registered magnetic resonance (MR) and PET brain images (7). In this visualization, the intensity of the combined MR–PET images is obtained from the MR images, but the color is coded using the gray values (glucose metabolism) of the corresponding PET images. The color representation can be directly mapped from the gray value histogram of the PET images. The histogram can be linearly scaled into blue (lowest partition representing low metabolism), green (middle partition representing medium metabolism), and red (highest partition representing the highest metabolism).

Figure 13.1 shows another example of a 2-D multimodality image visualization of brain images (8). The top left image of Figure 13.1 shows a T_2-weighted MR brain image. A perfusion MR image of the corresponding cross-section of the brain is shown in the top right corner. Using the multiparameter adaptive image segmentation algorithm described in Chapter 9, the segmented MR brain image is shown in the bottom left corner with color-coded segmented regions. The segmented image is superimposed onto the perfusion MR image to visualize the vascular structure in the context of segmented brain regions.

13.2. STEREO VISION AND SEMI-3-D DISPLAY METHODS

Direct 3-D visualization methods such as stereo pairs with color- or polarization-coded depth perception have been investigated in medical imaging. Through the distribution of brightness, the observer can visualize or perceive the depth in the volumetric data. With stereoscopic visualization, two views of an object are created with different viewing angles using a binocular vision approximating the same geometry. The difference in viewing angle depends on the viewing and interocular distances. The depth perception depends on the brain's interpretation of the viewing distance. The digital subtraction angiogram (DSA) data have been integrated with 3-D MR images using stereoscopic techniques of image-guided surgery (9).

Other direct display methods include motion parallax, varifocal mirror system, rotating light-emitting diode system, and holography (9–12). With motion parallax, a selection of rendered images acquired on each side of the viewing angles is

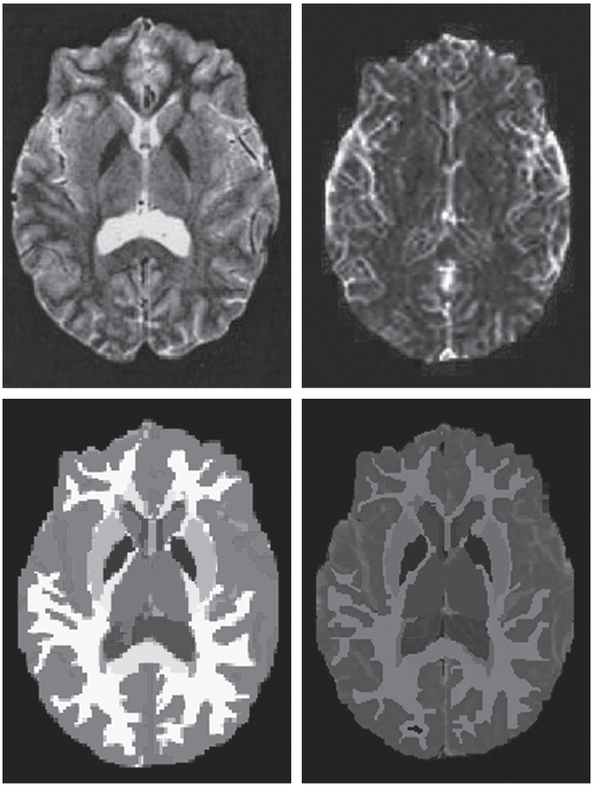

Figure 13.1 Top-left: A 2-D MR brain image; top-right: the registered MR perfusion image corresponding to the same slice position; bottom-left: the segmented MR brain image; bottom-right: superimposed perfusion image on the segmented MR brain image for composite visualization.

displayed in a sequence loop. The dithering about the central viewing axis provides the perception of depth and 3-D position. With specific hardware and numerical projection techniques, direct display methods can be used for visual interpretation to search for a desired region within the volumetric image. A more detailed visualization can then be performed in the regions of interest. A varifocal mirror system for 3-D display and analysis of tomographic volume images has been described in Harris (10). In the varifocal mirror-based display system, each plane of 3-D volumetric data is brought for visualization on a cathode ray tube (CRT) display, one at a time. The user sees the screen through a mirror reflection. The mirror is mounted in such a way that it can be moved through an assembly that is controlled by the user. As each plane of volumetric data is displayed, the mirror is displaced slightly. Thus, due to the displacement in the mirror position, the distance between the viewer's eye and display screen is changed, providing depth perception for 3-D visualization.

Holography has the potential to provide realistic 3-D displays that can be viewed from different directions (12). The best holograms are displayed using coherent light from a laser and high-resolution film. Holographic displays can also be displayed on a computer-controlled LCD monitor, but the resolution of such displays is not adequate for medical image visualization.

Although direct display methods provide interesting visualization for depth perception, they have little significance in diagnostic radiology dealing with tomographic images.

13.3. SURFACE- AND VOLUME-BASED 3-D DISPLAY METHODS

Three-dimensional volumetric data can be obtained from a 3-D medical imaging modality such as computed tomography (CT), magnetic resonance imaging (MRI), or PET. In many applications the 3-D volumetric may simply be organized as a stack of serial sections that are 2-D images of a 3-D object along a specific axis. For example, the CT, MRI, and PET scanners can acquire serial 2-D images along a specific axis to cover the 3-D object. However, recent developments in medical imaging allow direct 3-D imaging to produce 3-D volumetric data. In the representation of 3-D volumetric data in the rectangular coordinate system, a voxel is defined as a 3-D element or data point at the location (x, y, z). The visualization methods map the voxel-based 3-D representation of the volumetric data onto a pixel-based 2-D image representation that is displayed on a computer monitor. The mapped intensity values of the pixels in the image provide a visualization of a specific view of the 3-D object present in the volumetric data. The methods to visualize the volumetric data can be broadly classified into two categories: surface rendering or volume rendering methods.

In the surface rendering methods, the gray-level-based 3-D data is first converted into a binary volume representation. The 3-D binary volume is then searched for the boundary voxels that are interpolated to render a specific geometrical primitive-based 3-D surface. A surface is defined by the region that contains the boundary voxels or where the binary volume changes from the value "0" (background) to the value "1" (object). The rendered surface is shown in the image for visualization. Thus, the surface rendering-based visualization shows the geometrical shape of the surface of the object but with a loss of structural details within the volume. Usually, such visualization methods are useful in understanding the geometrical shape of the object and its variation within a group or with respect to an anatomical atlas. The binary volume-based surface visualization methods are also useful in image registration as described in Chapter 12.

Volume-rendering methods allow visualization of the structural variations within the 3-D volumetric data. A common method used in volume-rendering methods is ray tracing. In the simplest form of volumetric ray tracing, a uniform light source is placed behind the voxels of the volumetric data. For each straight-line-based ray from the light source to the corresponding point (pixel) in the image plane of the display screen, the density values and location of the voxels in the path of the ray are used to calculate the intensity value of the pixel in the corresponding image. Such a volume visualization method is based on the computed projection of the object onto the image plane with weighted intensity values based on the density and locations of the voxels along the respective rays. Using the location information of the voxels of the registered volumetric data, depth-dependent shading methods can be used to modify the intensity or color values of the image for better visualization (13–21).

13.3.1 Surface Visualization

The primary objective of surface display is to present 3-D shape information efficiently. The surface is first identified in terms of regions either manually or semiautomatically using a computer algorithm. The extracted regions are then represented as a set of contours or as a surface that reflects light from an imaginary light source (12–15). The output of the latter process is displayed through a 2-D frame buffer using the monocular depth cues of shading and motion parallax (rotation) to provide a visual interpretation of depth and shape. The surface displays provide a tool for analyzing shapes and spatial relationships through the aid of visual cues such as perspective, shading, and shadowing.

Let $f(x, y, z)$ represent a point or voxel location (x, y, z) in the 3-D volume data. A binary representation $f_b(x, y, z)$ of the 3-D volume data $f(x, y, z)$ can be obtained through a binary segmentation transform T such that

$$\begin{aligned} f_b(x,y,z) = T[f(x,y,z) &= 1 \qquad \text{if } f(x,y,z) \in R \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{13.1}$$

where R represents the range of gray values such that voxels with gray values in R belong to the object to be visualized.

Once a voxel model is obtained through the segmentation, boundary voxels are determined for surface rendering. Boundary detection algorithms search for the voxels that are connected and at the edge of the binary volume. The connectivity can be defined by setting up an adjacency criterion. For example, an 8-connected adjacency criterion can be used to identify the voxels that are adjacent to each other within a given plane. Two voxels v_1 and v_2 at locations, respectively, (x_1, y_1, z_1) and (x_2, y_2, z_2) are connected through the 8-connected adjacency criterion if

$$\text{either} \quad |x_1 - x_2| = 1 \quad \text{or} \quad |y_1 - y_2| \quad \text{and} \quad z_1 = z_2 \tag{13.2}$$

For example, Figure 13.2a shows four voxels: r, s, t, and u. The voxels r and t are adjacent to the voxel s but none is adjacent to the voxel u. Figure 13.2b shows a contour of the adjacent voxels belonging to the same z-plane. It can be noted that an 8-connected contour has at least three connected voxels that are closed, and none of the voxels are identical.

The boundary voxels are sometimes referred to points for rendering the surface for visualization. A contour is defined as the ordered sequence of boundary points. For surface visualization, the points or vertices on the boundary of the object can be sampled at random and therefore do not carry an ordered sequence. The surface is the region comprised of boundary voxels where the binary volume changes from the value “0” to “1”. The simplest method of surface visualization is to connect the sampled points using the zero-order interpolation to define the faces, which are shared by voxels with differing values of the binary volume. Using a higher-order interpolation function between the sampled points, a smoother surface can be

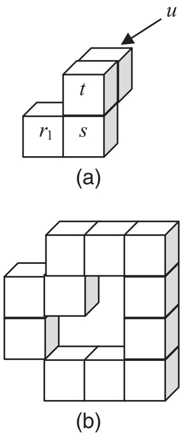

Figure 13.2 (a) Adjacent voxels *r*, *s*, and *t* with the 8-conneceted adjacency criterion. (b) A contour of adjacent voxels with the 8-conneceted adjacency criterion.

constructed. For example, B-splines can be used to define curved surfaces more efficiently (22, 23). Faces can also be defined through a mesh that is constructed around sampled points using a geometrical primitive to render surfaces. A mesh can be generated using a geometrical primitive such as a square or triangle (1, 2, 21). Thus, a surface visualization method connects a set of sampled points or vertices on the boundary of the object using an interpolation function or a geometrical primitive. To improve visual effects, a predefined scalar parameter such as color or brightness can be assigned to the face regions of the surface. When a continuous interpolation function is used, an iso-valued surface (or simply an iso-surface) can be defined by using a single value of the predefined scalar parameter (24–27).

If the sample points are centered at the voxels of the rendering surface, the surface passes through the sampled points and thus the zero-order interpolation method is required. However, if the voxels of the rendering surface are not centered at the sampled points of the volumetric data set (as is the case for rendering a smoother surface with higher resolution), a higher-order interpolation method is required to determine the values of a predefined scalar parameter of the voxels. Although there are a number of interpolation methods used in medical image visualization, a trilinear interpolation method, one of the most commonly used in visualization, is described here. It is the 3-D version of the popular 2-D bilinear interpolation method.

Let *A*, *B*, *C*, *D*, *E*, *F*, *G*, and *H* represent the sampled points at vertices of a hexahedron cell *ABCDEFGH* with known values, V_a, V_b, V_c, V_d, V_e, V_f, V_g, and V_h (Fig. 13.3). Assuming the unit dimensions of the hexahedron along each direction, a value V_p of a point *P* at the location (x_p, y_p, z_p) inside the hexahedron can be computed linearly using the trilinear interpolation method as

$$\begin{aligned} V_p = {} & V_a(1-x_p)(1-y_p)(1-z_p) + V_b x_p(1-y_p)(1-z_p) + \\ & V_c(1-x_p)y_p(1-z_p) + V_d x_p y_p(1-z_p) + V_e(1-x_p)(1-y_p)z_p + \\ & V_f x_p(1-y_p)z_p + V_g(1-x_p)y_p z_p + V_h x_p y_p z_p \end{aligned} \tag{13.3}$$

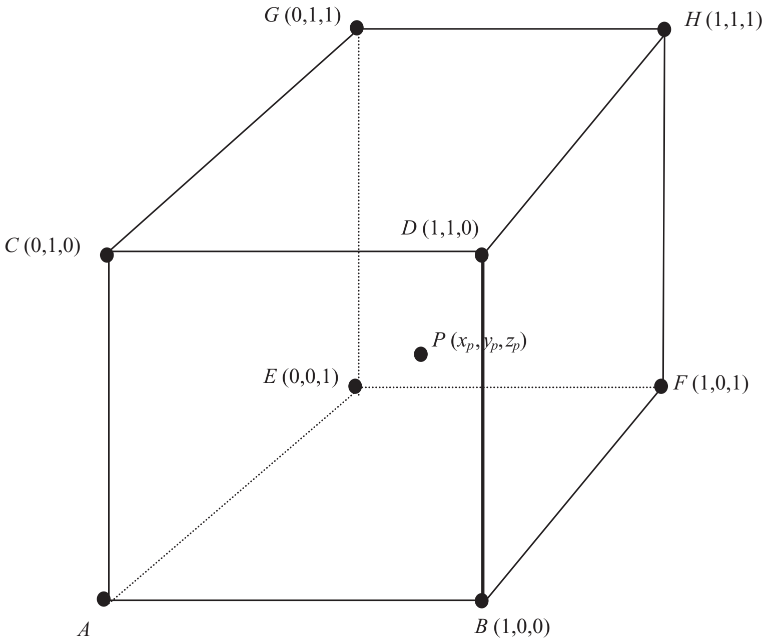

Figure 13.3 A schematic diagram of hexahedron-based sample points for trilinear interpolation method.

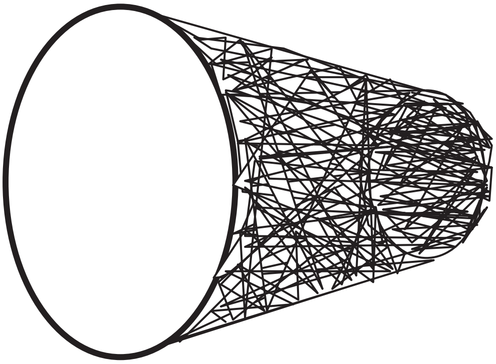

Figure 13.4 An example of rendering a mesh with triangles for surface visualization.

An example of rendering a mesh with triangles for surface visualization is shown in Figure 13.4.

Meshes generated using the sampled points in the volumetric data can be used to render iso-surfaces for visualization. A contour is defined in the 2-D space by the line path on which the value of a scalar parameter is constant. An iso-surface is the 3-D surface representing the locations of a constant scalar parameter within the volumetric data. The scalar parameter can be a raw property of the data (e.g., brightness) or a computed feature such as gradient or texture.

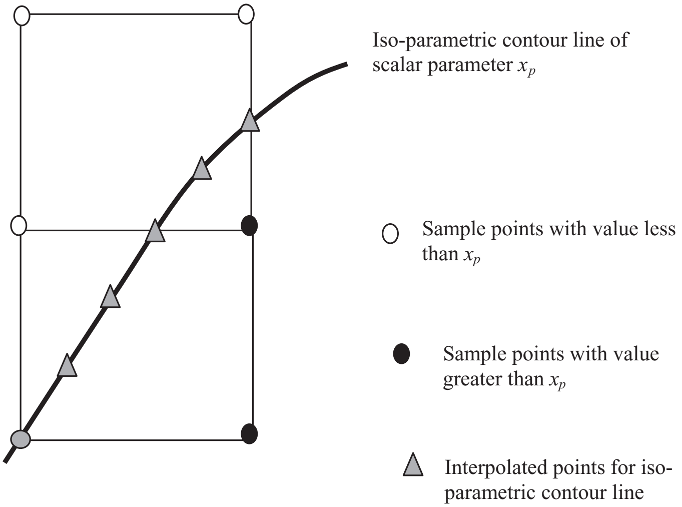
Iso-parametric contour line of scalar parameter x_p
○ Sample points with value less than x_p
● Sample points with value greater than x_p
△ Interpolated points for iso-parametric contour line

Figure 13.5 An iso-parametric contour line passing through square cells of sample points.

To compute the iso-parameteric contour line, a contour or line segment is first identified with the value of the associated scalar parameter. This can be done by setting up a look-up table of the neighboring sample points with similar values of the associated scalar parameter. Each entry in the look-up table is then followed up to define the iso-parametric contour lines. When following along the iso-parametric contour, if the value of the next sample point is the same as that of the previous point, the iso-parametric contour line is extended to include the new sample point. To improve the accuracy of the location of the iso-parametric contour lines, an interpolation function is usually required within a predefined cell of sampled points. Generally, the iso-parametric contour lines in the 2-D space are computed using a cell such as a square of vertices or sampled points. If all vertices of the cell have the values of the scalar parameter below or above the predefined value of the iso-parametric contour line, none of the vertices can be included to extend the iso-parametric contour line. If one vertex has a lower value and another has a value higher than the scalar value of the contour line, an interpolation function such as bilinear interpolation is required to determine the path of the iso-parametric contour line. As shown in Figure 13.5, additional vertices can be computed using bilinear interpolation methods to determine the path of the iso-parametric contour line between the sampled points.

Similar to the concept of the iso-parametric contour lines in the 2-D space, iso-surfaces can be computed within the 3-D volumetric data. As stated above, iso-surfaces represent the 3-D surfaces of constant values of the scalar parameter. The interpolation function required for determining iso-surfaces must interpolate in the 3-D space using a 3-D cell that may be shaped as a hexahedron or any other polygon depending on the mesh representation. If a hexahedron is used as a cell element, the eight vertices or sample points of each cell in the entire volume are searched to generate the iso-surfaces. A trilinear method (as described in Eq. 13.3) can be used as needed for interpolation to determine the path of the iso-surface. A general algorithm of generating iso-surfaces using the cubical cells in the volume is known as the "marching cube" algorithm (18). Other variants of the marching cube algorithm, such as the dividing cube algorithm, have been developed to speed up the iso-surface generation in large volumetric data sets (2). In the dividing cube algorithm, the search for the cells contributing to the iso-surface is accelerated by using a hierarchical structure of volume elements. The data is first organized using large-sized volume elements (voxels). Thus the entire volume can be searched quickly to focus the algorithm in the specific subvolumes containing the iso-surface. The subvolumes are then divided into subvoxels to improve the resolution and localization of the iso-surface.

The derivates of the volumetric data can be used to determine principal curvature direction and values to determine iso-surface directly from the voxel data (25, 26). Let C represent a curve with tangent vector $\boldsymbol{t}$, along a section of the iso-surface S. The tangent relationship to the volume V can be expressed as

$$\nabla V \cdot \boldsymbol{t} = 0 \tag{13.4}$$

It should be noted that C is contained in a plane that contains the surface normal $\boldsymbol{n}$. It can be shown that (25)

$$\boldsymbol{t}^t H\boldsymbol{t} + \nabla V \cdot k\boldsymbol{n} = 0 \tag{13.5}$$

where k represents the normal curvature of C, which is the surface curvature in the direction $\boldsymbol{t}$, and $H(V)$ is the Hessian of V as

$$H(V) = \begin{bmatrix} v_{xx} & v_{xy} & v_{xz} \\ v_{xy} & v_{yy} & v_{yz} \\ v_{xz} & v_{yz} & v_{zz} \end{bmatrix}. \tag{13.6}$$

The principal curvatures can be obtained by optimizing k over all possible directions of $\boldsymbol{t}$ using Equation 13.5 as

$$k = -\frac{\mathbf{t}^t H\mathbf{t}}{\|\nabla V\|}. \tag{13.7}$$

In 3-D surface display methods, the intensity at a point to be displayed is determined using a predefined criterion such as a function of the distance between the point and the viewer or the gradient of the surface at that point. The intensity and shading of the pixels associated with the display of the rendered surface help to provide visual cues for easy 3-D perception. Color mapping can be used to help feature visualization associated with the rendered surfaces.

Figure 13.6 shows an iso-surface visualization of the brain cortical convolutions computed from MR volumetric data. Figure 13.7 shows a 3-D composite visualization of ventricles extracted and registered from brain MRI volumetric data sets of 16 subjects. Figure 13.7b shows four rotated views of the color-rendered visualization of the spatial probability distribution of the composite 3-D ventricle model. The red areas represent the minimum probability value (1/16), while the white areas indicate the maximum probability value of 1.0 (6).

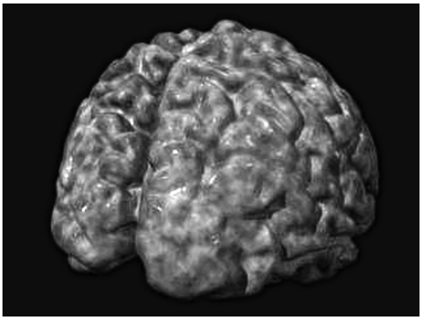

Figure 13.6 The iso-surface visualization of cortical convolutions of the human brain computed from the MRI brain volumetric data.

13.3.2 Volume Visualization

Volume rendering provides visualization of a 3-D volume through a 2-D image. Volume rendering can be implemented using an object order, image order, or a hybrid combination of these two methods (2). In object order-based volume rendering methods, an image is produced by projecting voxels of the volumetric data onto the image plane. This method is also called the forward mapping method to map volume data onto the image plane for visualization. The difficulty with this method is the likelihood of many-to-one mapping to cause ambiguous visualization. If voxels are projected onto the image plane using an arbitrary selection criterion, it is possible that two or more voxels can be mapped to the same pixel in the image. To avoid this, the voxels should be projected using one plane at a time in a sequential manner. Thus, the projections are computed slice-by-slice within the volume and row-by-row within a slice. With the mapping, distance information corresponding to the pixels in the image plane can be used to create depth information for visualization. The depth values thus obtained are used for computing 2-D discrete shading effects in the visualized image. The shading in the image can also be computed by using a combination of the 2-D depth and surface gradient information. In such methods, the 2-D depth information is passed through a gradient operator to account for the surface orientation and the distance between voxels and the light source placed in the image plane. A gradient vector at each pixel location is then estimated and used as a normal vector for computing shading and intensity values in the image for visualization.

The gradient value at a pixel location (x, y) in the image can be expressed as (2)

$$\nabla z = \left(\frac{\delta z}{\delta x}, \frac{\delta z}{\delta y}, 1 \right) \tag{13.8}$$

where $z = D(x, y)$ is the depth associated with the pixel (x, y).

In the image-order methods, a backward mapping is implemented through tracing the rays from pixels in the image to the volume data for visualization. For each pixel in the image plane, a ray is defined originating from that pixel to intersect with the objects and surfaces within the volumetric data. Two types of ray

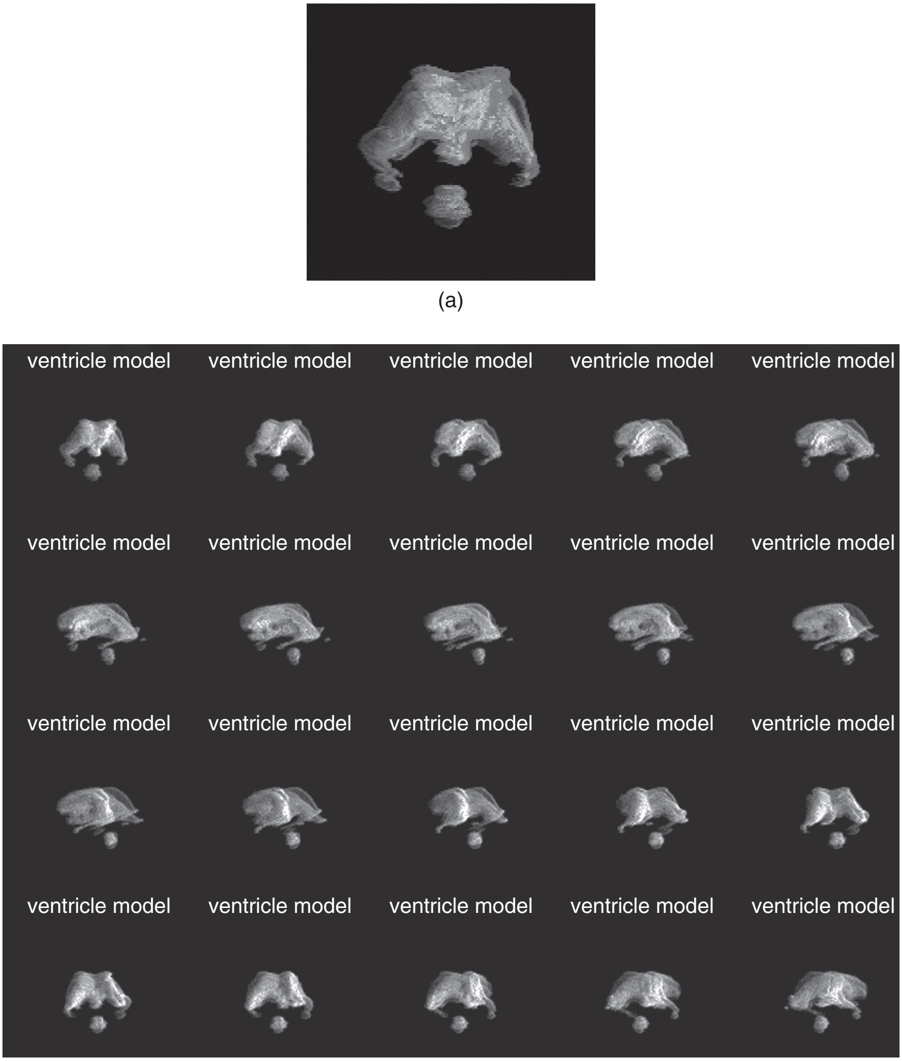

Figure 13.7 (a) Three dimensional visualization of a composite ventricle model computed from the MRI brain volumetric data of 16 subjects. (b) Four rotated views of the 3-D ventricle model.

geometries can be used: parallel beam and divergent beam. In parallel geometry, all rays are parallel to the viewing direction providing parallel projections. In divergent beam geometry, perspective projections are obtained using the rays that are cast from the eye point according to the viewing direction and the field of view (1, 2). Shading is performed at the intersections of the rays and surfaces within the volume. For each intersection, the intensity of the ray is computed using attenuation and reflection parameters for structural and shading representations. In general, the intensity of the ray from the eye point, through pixels in the image plane, and onto the volume can be expressed by

$$I = \int_{t_1}^{t_2} e^{-\tau \int_{t_1}^{t} \rho^{\gamma}(\lambda) d\lambda} \rho^{\lambda}(t)\, dt \tag{13.9}$$

where t is the location of the ray that is transversed from t_1 to t_2, $\rho^{\gamma}(t)$ is the accumulated density function at the location t, τ is the attenuation parameter, and γ is the parameter that controls the diffusion.

The volume visualization can also be implemented using a cell-by-cell processing method (21). In this method, each cell in the volume is traced using a ray in front-to-back order. The ray tracing and possessing starts with the plane closest to the viewpoint and continues toward the back in a plane-by-plane manner. Within each plane, the cell closed to the viewpoint is processed first. Thus the cells with the longest distance from the viewpoint are processed last. A scan line is established in the image plane to determine the pixels affected by the cell. The shading equation providing the perceived intensity $I(\lambda)$ as a function of wavelength λ can be expressed as (2)

$$I(\lambda) = \int_x \int_y \int_z \left[\tau(d) O(s) \left[K_a(\lambda) I_a + K_d(\lambda, M) \sum [(N \cdot L_j) I_j] \right] + (1 - \tau(d) b(\lambda)) \right] dx\, dy\, dz \tag{13.10}$$

where I_a is the ambient intensity; $\tau(d)$ is the attenuation as a function of distance d; $O(s)$ is the opacity transfer parameter as a function of the scalar parameter s; K_a and K_d are, respectively, the ambient and diffusion coefficients; M is another parameter, such as texture, modeled to affect the diffusion; N is the normal computed from the local gradient; L_j is the position vector to the jth light source; I_j is the intensity of the jth light source; and $b(\lambda)$ is the background.

In the ray tracing method, the intensity of each ray is computed along with other parameters such as maximum value, the distance for the maximum value, and the center of gravity of the density emitters. These parameters can be used for specific color coordinates such as saturation, hue, and intensity to produce dramatic effects for color-based volume visualization. Figure 13.8 shows a volume

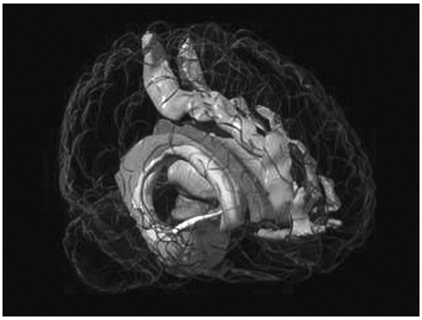

Figure 13.8 A volume visualization of MR brain volumetric data (http://www.bic.mni.mcgill.ca/).

visualization of brain structures. The density values of the cortical convolutions are assumed to be low so that the internal structures can be visualized through transparent layers.

Surface and volumetric visualization methods have been used to assist surgical planning, requiring the visualization of the location of tumors within the brain, radiation treatment planning, surgery, and other medical applications (27–37).

13.4. VR-BASED INTERACTIVE VISUALIZATION

Virtual reality is an advanced computer visualization technology that provides users with the ability to control the visualization volume. In the ideal software environment, VR allows a user to experience a synthetic world with visual scenes that mimic real-world situations. The VR environment may use extensive computer graphics, sounds, and images to reproduce electronic versions of real-world situations. In medical imaging applications, the current VR systems provide only visual experiences created by computer-assisted visualization and animation to walk through and feel the anatomical structures of an organ within a virtual human body.

From a historical perspective, a breakthrough in VR came with the development of a head-mounted display with two tiny stereoscopic screens positioned just a few inches in front of a viewer's eyes. The VR system designed by Jaron Lanier in 1989 featured a head-mounted display called the EyePhone (30, 31). Users also wear a DataGlove that generates movement and interaction in the virtual environment. The interaction and movements in the computerized virtual world or cyberspace are provided by moving the head, fingers, or entire hand through the appropriate input devices such as haptic gloves. The movements of gloves or other input devices directly control the optics in the field of vision to provide interaction and sensation in the virtual world.

The EyePhone uses a set of wide-angle optics that cover approximately 140 degrees, almost the entire horizontal field of view (31). As the user moves his or her head to look around, the images shift to create an illusion of movement. The user moves while the virtual world is standing still. The glasses also sense the user's facial expressions through embedded sensors, and that information can control the virtual version of the user's body. For example, the DataGlove, a key interface device, uses position-tracking sensors and fiber optic strands running down each finger, allowing the user to manipulate objects that exist only within the computer-simulated environment (30, 31). When the computer "senses" that the user's hand is touching a virtual object, the user "feels" the virtual object (30, 31). In VR-based systems, the user can pick up an object and do things with it just as he or she would do with a real object. VR systems have been successfully used in many simulator applications such as flying aircraft, controlling spacecrafts, and navigating and repairing instruments in space.

Recent developments in computer technology and sensory input devices have enabled VR technology to have potential use in diagnostic radiology for patient-specific dynamic interactive visualization. However, the computational issues to deal with large data sets and the complexity of physiological structures and responses

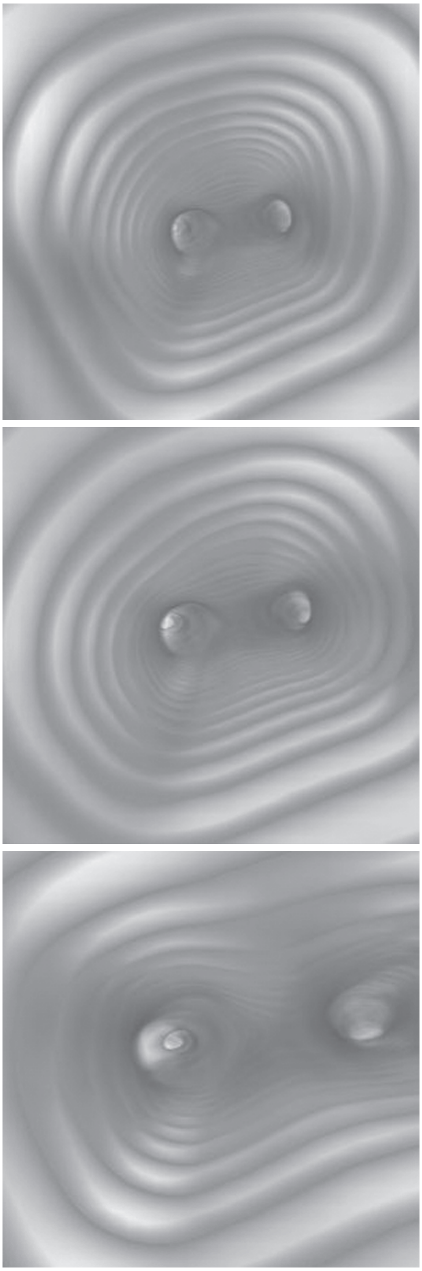

Figure 13.9 A sequence of volume-rendered images along the path through the central axis of the bronchial tube is used to produce a fly-through animation.

stjll remains as a major challenge. Recently, VR systems have been explored in medicine to simulate surgical procedures and facilitate image-guided surgery (32–34).

The major steps in developing VR-based interactive visualization are as follows:

1. Specific volumetric data sets from medical imaging scanners are segmented for regions and objects of interest.
2. Surfaces and volumes of interest are rendered in 3-D space to create sequences of 3-D scenes.
3. Specific geometrical and response models for interaction with 3-D scenes are developed and integrated with visualized data in the virtual reality modeling language (VRML) environment.
4. Interfaces to control the navigation and responses from the visual scenes are developed in the VRML environment using the appropriate input devices.
5. Navigation and interaction tutorials are developed to train the user to use the VR system.

13.4.1 Virtual Endoscopy

Recently, virtual endoscopy (VE) has been used for the visualization of bronchial airways (32, 33). The VR-based visualization technique provides a noninvasive way to examine the interior of the bronchial tubes. X-ray CT images are first segmented for airways using an adaptive thresholding method. The segmented images are used for volume rendering along the medial axis of the airways that is used as the path for animation.

The bronchial tube usually has two major skeleton branches and many smaller ones. The 3-D segmentation obtained by thresholding can provide several regional branches, making it difficult to determine a global axis through the medial axis transformation or skeletonization. A user interaction can be used to select the desired axis for fly-through animation using a VRML-based model (30). A ray-tracing method can be used with rays traveling from the user eye point into the structure for volume visualization. To visualize tube-like structures, the animation is directed toward the end of the tube. Figure 13.9 shows volume-rendered images of the bronchial tubes from a VR visualization (33).

13.5. EXERCISES

13.1. Why is 3-D visualization important in medical imaging? Explain your answer with the help of an example.

13.2. What are the different methods used in 3-D visualization in medical imaging?

13.3. What is the difference between the surface and volume visualization approaches? For brain imaging applications, which approach would you prefer and why?

13.4. What is the purpose of rendering a mesh in the volumetric data?

13.5. What type of polygon would you prefer for mesh rendering in MR brain data? Give reasons to support your answer.

13.6. What do you understand by the term iso-surface?

13.7. Explain an algorithm to generate iso-surfaces in the volumetric data. Can this algorithm be improved? If so, how would you improve the performance of this algorithm?

13.8. Why is shading important in visualization? How would you provide shading in iso-surface-based visualization of cortical structures from the MR brain volumetric data?

13.9. Segment the MR brain volumetric data for ventricles and cortical convolutions in the MATLAB environment. Display 2-D slice images in the MATLAB environment for the following

a. Display the ventricles with boundaries outlined and superimposed on the respective 2-D slice images.

b. Display the cortical convolutions with boundaries outlined and superimposed on the respective 2-D slice images.

13.10. Generate iso-surfaces from the MR brain volumetric data set used in Exercise 13.9. Display the extracted iso-surfaces as 2-D slice images.

13.11. Generate the 3-D iso-surface-based visualization of cortical structure segmented in Exercise 13.9. Explain the shading method used in the surface-rendered volume.

13.12. Change the opacity factor for the voxels corresponding to the cortical convolutions. Render a volume to visualize ventricles through the "see-through" cortical structure using the ray-tracing method. Comment on the quality of your visualization. How can you improve it?

13.6. REFERENCES

1. C. Bajaj (Ed.), *Data Visualization Techniques*, John Wiley and Sons, New York, 1999.
2. R.S. Gallagher (Ed.), *Computer Visualization*, CRC Press, Boca Raton, FL, 1995.
3. I.N. Bankman, *Handbook of Medical Imaging*, Academic Press, San Diego, CA, 2000.
4. A. Gueziec, "Extracting surface models of the anatomy from medical images," in M. Sonka and J.M. Fitzpatric (Eds), *Handbook of Medical Imaging*, Volume 2, SPIE Press, Bellingham, WA, 2000, pp. 343–398.
5. L.D. Harris, "Display of multidimensional biomedical image formation," in R.A. Robb (Ed.), *Three-Dimensional Biomedical Imaging*, Volume II, CRC Press, Boca Raton, FL, 1985, pp. 74–107.
6. L.K. Arata, A.P. Dhawan, J.P. Broderick, M.F. Gaskil-Shipley, A.V. Levy, and N.D. Volkow, "Three-dimensional anatomical model-based segmentation of MR brain images through principal axes registration," *IEEE Trans. Biomed. Eng.*, Vol. 42, No. 11, pp. 1069–1078, 1995.
7. A.P. Dhawan, L.K. Arata, A.V. Levy, and J. Mentil, "Iterative principal axes registration method for analysis of MR-PET brain images," *IEEE Trans. Biomed. Eng.*, Vol. 42, No. 11, pp. 1079–1087, 1995.
8. A. Zavaljevski, A.P. Dhawan, S. Holland, W. Ball, M. Giskill-Shipley, J. Johnson, and S. Dunn, "Multispectral MR brain image classification," *Comput. Med. Imaging Graph. Image Process.*, Vol. 24, pp. 87–98, 2000.

9. T.M. Peters, C.J. Henri, P. Munger, A.M. Takahashi, A.C. Evans, B. Davey, and A. Olivier, "Integration of stereoscopic DSA and 3-D MRI for image-guided surgery," *Comput. Med. Imaging Graph.*, Vol. 18, pp. 289–299, 1994.
10. L.D. Harris, J.J. Camp, E.L. Ritman, and R.A. Robb, "Three-dimensional display and analysis of tomographic volume images utilizing a varifocal mirror," *IEEE Trans. Med. Imaging*, Vol. 5, pp. 67–73, 1986.
11. W. Simon, "A spinning mirror auto-stereoscopic display," *Proc. SPIE*, Vol. 120, pp. 180–185, 1977.
12. P. Greguss, "Holographic displays for computer assisted tomography," *J. Comput. Assist. Tomogr.*, Vol. 1, pp. 184–191, 1977.
13. G.T. Herman and H.K. Liu, "Three-dimensional display of human organs from computerized tomograms," *Comput. Graph. Image Proc.*, Vol. 9, pp. 1–9, 1979.
14. J.K. Udupa, "Display of 3D information in discrete 3D scenes produced by computerized tomography," *Proc. IEEE*, Vol. 71, pp. 420–431, 1983.
15. J.J. Udupa and R.J. Goncalves, "Imaging transform for visualizaing surfaces and volumes," *J. Digit. Imaging*, Vol. 6, pp. 213–236, 1993.
16. K.H. Hohne and R. Bomstein, "Shading 3D images from CT using gray-level gradients," *IEEE Trans. Med. Imaging*, Vol. MI-5, No. 1, p. 45, March 1986.
17. C.A. Pelizzari, G.T.Y. Chen, D.R. Spelbring, R.R. Weichselbaum, and C.T. Chen, "Accurate three-dimensional registration of CT, PET and/or MR images of the brain," *J. Comput. Assist. Tomogr.*, Vol. 13, No. 1, pp. 20–26, 1989.
18. W. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3-D surface construction algorithm," *Proc. SIGGRAPH Comput. Graph.*, Vol. 21, pp. 163–169, 1987.
19. L. Axel, G.T. Herman, J.K. Udupa, P.A. Bottomley, and W.A. Edelstein, "Three-dimensional display of nuclear magnetic resonance (NMR) cardiovascular images," *J. Comput. Assist. Tomogr.*, Vol. 7, pp. 172–174, 1983.
20. P. Thomson and A.W. Toga, "A surface based technique for warping three-dimensional images of the brain," *IEEE Trans. Med. Imaging*, Vol. 15, pp. 402–417, 1996.
21. C. Upson and M. Keeler, "V-Buffer: Visible volume rendering," *Comput. Graph. Proc. SIGGRAPH*, pp. 59–64, 1988.
22. P. Dierckx, *Curves and Surface Fitting with Splines*, Oxford Science Publications, London, 1993.
23. T.M. Lehmann, C. Gonner, and K. Spitzer, "Addendum: B-spline interpolation in medical image processing," *IEEE Trans. Med. Imaging*, Vol. 20, pp. 660–665, 2001.
24. T.M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Trans. Med. Imaging*, Vol. 18, pp. 1049–1075, 1999.
25. O. Monga, S. Benayoun, and O. Faugeras, "Using third order derivatives to extract ridge lines in 3D images," Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Champaign, IL, IEEE Press, June 15–18, 1992.
26. J.P. Thirion and A. Gourdon, "Computing the differential characteristics of isointensity surfaces," *Comput. Vis. Image Underst.*, Vol. 61, pp. 190–202, 1995.
27. S.L. Hartmann and R.L. Galloway, "Depth buffer targeting for spatial accurate 3-D visualization of medical image," *IEEE Trans. Med. Imaging*, Vol. 19, pp. 1024–1031, 2000.
28. Y. Sato, M. Nakamoto, Y. Tamaki, T. Sasama, I. Sakita, Y. Nakajima, M. Monden, and S. Tamura, "Image guidance of breast cancer surgery using 3-D ultrasound images and augmented reality visualization," *IEEE Trans. Med. Imaging*, Vol. 17, pp. 681–693, 1998.
29. J.L. Herring, B.M. Dawant, C.R. Maurer, D.M. Muratore, R.L. Galloway, and J.M. Fitzpatrick, "Surface-based registration of CT images to physical space for image guided surgery of the spine," *IEEE Trans. Med. Imaging*, Vol. 17, pp. 743–752, 1998.
30. H. McLellan, "Virtual field trips: The Jason project," *VR World*, Vol. 3, pp. 49–50, 1995.
31. M. Fritz, "Eyephones, datasuits, and cyberspace," *CBT Dir.*, Vol. 28, pp. 11–17, (EJ 420 905), 1990.
32. I. Bricault, G. Ferretti, and P. Cinquin, "Registration of real and CT derived virtual bronchoscopic images to assist transbronchial biopsy," *IEEE Trans. Med. Imaging*, Vol. 17, pp. 703–714, 1998.
33. S. Loncaric and T. Markovinovic, "Web based virtual endoscopy," Proceedings, Medical Informatics Europe, Hannover, Germany, 2000.

34. J.Z. Turlington and W.E. Higgins, “New techniques for efficient sliding thin-slab volume visualization,” *IEEE Trans. Med. Imaging*, Vol. 20, pp. 823–835, 2001.
35. P. St-Jean, A.F. Sadikot, L. Collins, D. Clonda, R. Kasrai, A.C. Evans, and T.M. Peters, “Automated atlas integration and interactive three-dimensional visualization tools for planning and guidance in functional neurosurgery,” *IEEE Trans. Med. Imaging*, Vol. 17, pp. 672–680, 1998.
36. I. Bricault, G. Ferretti, and P. Cinquin, “Registration of real and CT-derived virtual bronchoscopic images to assist transbronchial biopsy,” *IEEE Trans. Med. Imaging*, Vol. 17, pp. 703–714, 1998.
37. S. Cotin, H. Delingette, and N. Ayche, “Real-time elastic deformation of soft tissue for surgical simulations,” *IEEE Trans. Vis. Comput. Graph.*, Vol. 5, pp. 62–73, 1999.