Chapter 6   <u>3D Echocardiographic Issues</u>

The techniques of automated analysis developed in previous chapters can be applied to a wide variety of images. The particular application of this dissertation involves the use of RT3D ultrasound images. It is essential to consider the physics and mathematics by which such images are formed as part of the overall process.   This chapter reviews specific challenges encountered in analyzing RT3D ultrasound data and also describes visualization techniques to produce slices for manual segmentation used to judge the accuracy of the automated analysis. As will be seen, RT3D ultrasound is quite a challenging environment, both because of the physics of ultrasound and its unusual coordinate system. The suggestion that the statistical nature of the analysis yields robustness is severely tested in its application to this data.

6A. Coordinate System Issues

This section addresses issues arising from the non-rectilinear mapping between ultrasound's *data space* , i.e., the 3D array where the data samples are stored, and *physical space*. The issues arise when boundariness is computed via the image gradient. Analyzing shape proceeds most naturally in physical space. However, computing intensity gradient using a cubical convolution kernel in data space has many computational advantages. Adjustment of the results via the metric tensor from data space to physical space can produce a correct gradient magnitude and orientation, but the scale of the gradient does not remain isotropic since the projection of the isotropic convolution kernel is no longer isotropic. A case is made as to why this anisotropy is acceptable when the resulting boundaries are used to form core atoms.

RT3D ultrasound, like conventional ultrasound, employs a phased array of transducer elements to steer the ultrasound beam. Instead of a linear array that steers the beam within a single slice, the RT3D scanner uses a matrix array to steer a beam within a 3D volume (see Fig. 2.1). The fact that the beam emanates from (and is received at) a single location defines a geometry for the data unlike other medical imaging modalities. Subsequent analysis and visualization requires special attention to the coordinate system in which the data are collected.

At some point during acquiring data and forming an image, most 3D modalities keep voxel values in an array accessed by 3 independent indices, a *data space* from which each location

maps to a unique location in *physical space*. Shape analysis is conducted most naturally in physical space, which uses the Cartesian coordinate system. The Cartesian system, being both rectilinear and isotropic, permits Euclidean distance to be calculated. Parameterization of shape using Euclidean distance makes it invariant to translation and rotation and easily accommodates changes in scale.

In many imaging modalities, the mapping between data space and physical space is rectilinear, although it is often *anisotropic* with the units of distance unequal along different axes. For example, in magnetic resonance or computed tomography, the data is often collected in slices whose inter-pixel distance is less than the inter-slice distance. Even with this anisotropy, a rectilinear mapping is relatively convenient for analysis because the physical orientation and scale of the coordinate axes do not change with location.

This does not hold for RT3D ultrasound, which has a mapping that is not rectilinear. The three coordinates in RT3D ultrasound data are *azimuth*, *elevation*, and *range* (again, see Fig. 2.1). The *azimuth-elevation* coordinate system is closely related to the conventional *spherical* coordinate system (See Fig. 6.1). In azimuth-elevation coordinates, the axes are range r along the ultrasound beam (distance from the transducer) and azimuth and elevation (degrees between the r-axis and its projection on the y-z and x-z planes, respectively).
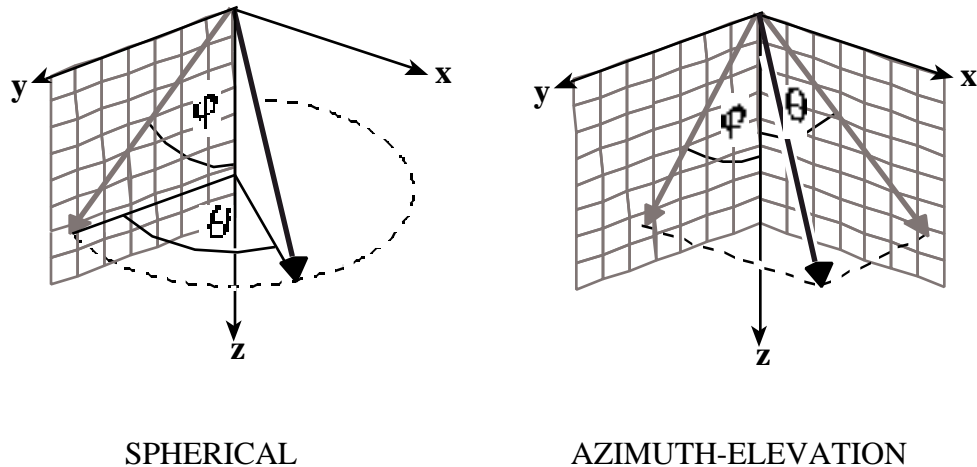

SPHERICAL   AZIMUTH-ELEVATION

Fig. 6.1 Spherical and azimuth-elevation coordinate systems each use two angles (,) and range (length of the large black arrow, not labeled). Cartesian coordinates (x,y,z) also shown.

The azimuth-elevation coordinate system changes orientation for different values of and . Furthermore, scale in the and dimensions changes with r, and only r has a constant

linear dimension. Assuming the same units of linear distance for x, y, z, and r, the transformations between azimuth-elevation coordinates and physical Cartesian coordinates are

$$x = r\sin\ \cos\ \qquad\qquad = \tan^{-1}\ \frac{x}{z}$$

$$y = r\sin\ \cos\ \qquad\qquad = \tan^{-1}\ \frac{y}{z} \qquad\qquad (6.1)$$

$$z = r\cos\ \cos\ \qquad\qquad r = \left(x^2 + y^2 + z^2\right)^{\frac{1}{2}}$$

The , , and r coordinates, with appropriate scaling, offset and discretization, serve as indices into the data array. The offsets include the distance along the r axis between the transducer and the first ultrasound sample as well as an angular offset to make   and   zero along the z-axis.

While the conversion between data space and physical space is straightforward for location, it is less so for intensity gradient. The methods of analysis developed in Chapters 3-5 are built upon an initial measurement of oriented boundariness. However, exactly what is meant by oriented boundariness has so far been unspecified. For the purposes of this dissertation, oriented boundariness takes the form of intensity gradient, which must be measured throughout physical space in a manner that preserves the underlying magnitude and orientation of the gradient. This is accomplished using the *difference-of-Gaussians* operator described in Appendix 1. The difference-of-Gaussians operator uses an efficient convolution kernel of variable aperture that approaches a Gaussian as the aperture increases. The kernel blurs the image and finds the difference within intensity pairs offset along each coordinate axis, thus yielding the three components of the gradient as a function of scale.

The concept of scale in boundariness was introduced in Chapter 3 but bears further examination as it applies to the gradient. For a continuous function, taking the derivative yields another continuous function. The continuous derivative operates at a scale approaching zero in the limit. On discrete data, however, the derivative operates at a finite scale determined by the size of the underlying Gaussian convolution kernel, and the proportional distance at which the results are subtracted to produce the scaled partial derivative approximation. The smallest possible scale for the derivative is the interval at which the data itself is sampled, known as the *inner scale*, but this is by no means the best scale for all applications. The choice of the particular scale for the gradient operator depends on the particular application.

To achieve efficiency, the difference-of-Gaussian convolution kernel needs to be applied not in physical space but in data space, thought of as having its own Euclidean distance on the axes , , and r. Although it is possible to re-sample and interpolate the entire ultrasound

data set into physical coordinates before processing, doing this while preserving resolution near the transducer undesirably increases the number of bytes in a single scan from 2 megabytes to over 100 megabytes. Although it is conceivable to reshape the kernel at each location to keep it physically of constant shape, the necessary interpolation between samples would be very costly in computation time and introduce sampling problems.

Several issues arise from convolving in data space (see Fig. 6.2). The 3x3x3 kernel that is cube in data space takes on a variety of shapes and sizes in physical space.   Adjustments of units along the   ,   , and r axes is necessary for the gradient magnitude to accurately represent the physical space's boundariness and/or the gradient orientation to represent the boundary normal in physical space. This is accomplished by multiplying the gradient in data space $\left(I_r,\ I\ ,\ I\ \right)$ by the metric tensor as follows:

$$
\begin{bmatrix} \tilde{I}_r \\ \tilde{I} \\ \tilde{I} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/r\cos & 0 \\ 0 & 0 & 1/r\cos \end{bmatrix} \begin{bmatrix} I_r \\ I \\ I \end{bmatrix}
\tag{6.2}
$$

The components of the gradient vector $\left(\tilde{I}_r,\ \tilde{I}\ ,\ \tilde{I}\ \right)$ are scaled to physical units, that is, all three components are expressed in terms of linear distance. Furthermore, the components are locally orthogonal. Therefore gradient magnitude $\left| \tilde{I} \right|$ can be calculated in the usual manner:

$$
\left| \tilde{I} \right| = \left(\tilde{I}_r^{\,2} + \tilde{I}^{\,2} + \tilde{I}^{\,2} \right)^{\frac{1}{2}}
\tag{6.3}
$$

For shape analysis to proceed, the orientation of the gradient vector should be expressed in a coordinate system that does not itself rotate from one location to another, namely, the physical Cartesian coordinate system.   The same vector may be expressed in Cartesian coordinates $\left(I_x,\ I_y,\ I_z \right)$ by projecting it onto the x, y, and z axes:

$$
\begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} = \begin{bmatrix} \cos & 0 & x/r \\ 0 & \cos & y/r \\ 0 & 0 & z/r \end{bmatrix} \begin{bmatrix} \tilde{I} \\ \tilde{I} \\ \tilde{I}_r \end{bmatrix}
\tag{6.4}
$$

A key observation is that the azimuth-elevation coordinate system is locally orthogonal, so gradient magnitude can be calculated for the purposes of thresholding without rotating to Cartesian coordinates. Only after strong boundary points have been selected, need they be rotated to physical coordinates as needed avoiding significant unnecessary computation.

Consider now the formation of core atoms from two candidate boundary points. The gradient at each boundary point is determined as above using a difference-of-Gaussian operator in data space corrected by Eq. 6.2 and rotated by 6.4. A potential problem still remains in physical shape of the kernel. As shown in 2D in Fig. 6.2 the 3x3 kernel that would be square in data space is curved, anisotropic, and changes size and shape as a function location in physical space. Several possible core atoms formed using such boundary kernels are shown. At location **c** the kernel has a smaller scale in azimuth and elevation than at **d**, although it has the same scale in range. At **a** and **b** the kernel has identical scales in each dimension but is still curved. Resulting core atoms **a-b** and **c-d** are shown on the right. Core atoms with other orientations would have intermediate qualities.
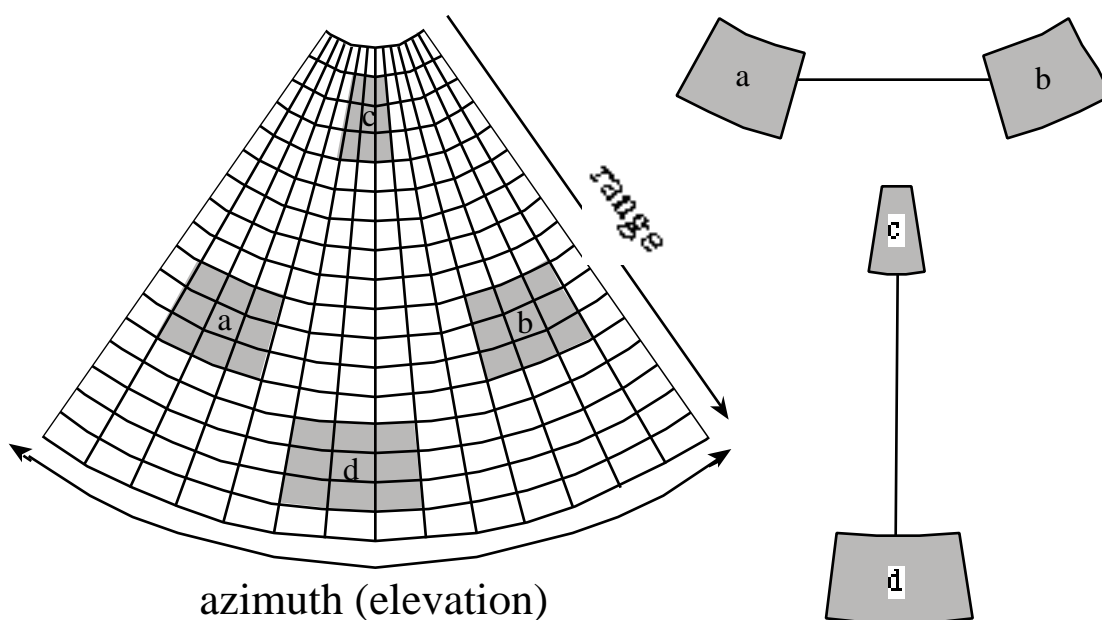

Fig. 6.2 Convolution kernels in physical space used to create core atoms, illustrated in 2D.

The arguments for accepting these shape distortions for the purposes of forming core atoms are as follows:

(1) At least the kernel is symmetrical in azimuth and elevation.

(2) Although the scale of the kernel increases with range in the azimuth and elevation dimensions, this matches a corresponding decrease with range in the lateral resolution of the ultrasound data.

(3) For core atoms oriented *across* the ultrasound beam (Fig. 6.2, **a-b**) the scale of the kernels at the two boundary points is identical. The orientations of the kernels differ, but this is corrected using Eq. 6.4.

(4) For core atoms oriented *along* the ultrasound beam (Fig. 6.2, **c-d**) the scale is constant in the range dimension, the most important component of boundariness expected for these core atoms.

The overall process of forming core atoms is thus as follows: Boundary points whose gradient magnitude is sufficient (Eqs. 6.2 and 6.3) are collected in a lattice of bins rectilinear in physical space. This pre-sorting greatly accelerates the search for matching boundary points (see Fig. 3.1 B). Doing so in physical space eliminates needless redundant coordinate system conversions when computing the distance between boundary points. If that distance is appropriate for the formation of a core atom, only then is Eq. 6.4 applied to the constituent boundary points so that they may be tested for sufficient *face-to-faceness*.

6B. Ultrasound Physics Issues

This section considers concerns (summarized in Table 6.1) arising from the physics of ultrasound and explains the manner in which each concern is addressed during the application of core atoms to images of the heart. Each of these concerns is considered in detail.

Concern	Method of Addressing the Problem
(1) Noise	Keep the aperture of the boundariness kernel appropriately large.
(2) Spatial resolution	Choose targets which are larger than the spatial resolution, such as the left vetnricle and the mitral valve.
(3) Temporal resolution	The speed of RT3D ultrasound is considered high engough to sample the motion of the heart.
(4) Anisotropy and attenuation	Compensate by anisotropic weighting of gradient components as a function of range.
(5) Spatial distortion	Not considered sufficiently large to be a problem.

Table 6.1 Concerns arising from ultrasound physics, and methods of addressing them.

(1) Noise

Noise, a significant problem in general with ultrasound, is especially high in matrix array ultrasound because of the relatively small size of the matrix-array transducer elements as compared to those in a conventional linear array. The small size of the individual elements results in a high electrical impedance and a low signal-to-noise ratio.

Another type of noise, *speckle,* is inherent to ultrasound. Speckle is an artifact of coherent radiation, the term originating from its occurrence in laser illumination. The *resolution cell* for ultrasound is defined by the smallest discernible resolution between targets at a certain location relative to the transducer. Within a single resolution cell, multiple sub-resolution reflectors such as red blood cells each reflects energy with a different amplitude and phase. These reflections add up to a total intensity from a particular resolution cell that can provide a signature differentiating one resolution cell from another within a fundamentally homogenous target. Since the speckle can survive from one time frame to the next, it can be used to track the

motion of blood and tissue, as discussed in Chapter 2. However, in terms of finding boundaries of objects in a single time frame, speckle is a hindrance.

Another artifact which causes variation between pixels is especially prevalent in slices parallel to the face of the transducer. Since pixel intensity is directly related to echo strength and since the ultrasound energy has to pass (twice) through intervening tissue, the pixel carries information about the intervening tissue. The paths differ between pixels in a C-mode image (parallel to the transducer) adding a source of variation between pixels that does not derive from the slice itself.

The techniques developed in this dissertation are based on the concept of the *core* , which strives to be relatively immune to noise by linking the size of a boundariness aperture to the scale of the corresponding medialness. The effectiveness of this approach has been demonstrated in other core-based applications (Morse, Pizer et al. 1998). At the scale of major cardiac structures such as the RV and LV, the boundariness aperture is significantly larger than the size of the speckle and many other sources of noise. Robustness to noise is also sought through the statistical nature of the analysis of core atom populations.

(2) Spatial resolution

The spatial resolution of RT3D ultrasound is another of its fundamental limitations. The size of the resolution cell is determined by wavelength, transducer aperture, location relative to the transducer, and the ability to form a beam in spite of inhomogeneity in local sound velocity. The shape of the RT3D resolution cell differs from that of a conventional ultrasound scanner, as shown in Fig. 6.3.

A linear array (Fig. 6.3A) actively focuses in only one dimension, producing a voxel elongated in elevation through the plane of the slice. Thus, although its cross-sectional area within the slice is small, its total volume is not. The matrix-array scanner (Fig. 6.3B) actively focuses in two dimensions, producing a symmetric voxel. The symmetry between azimuth and altitude in RT3D ultrasound is lacking in 3D ultrasound based on mechanically-moved linear arrays. The spatial resolution is not a serious problem in achieving the goals of this dissertation because the targets, i.e., the left ventricle and the mitral valve, are large enough to be clearly seen in the RT3D ultrasound images.
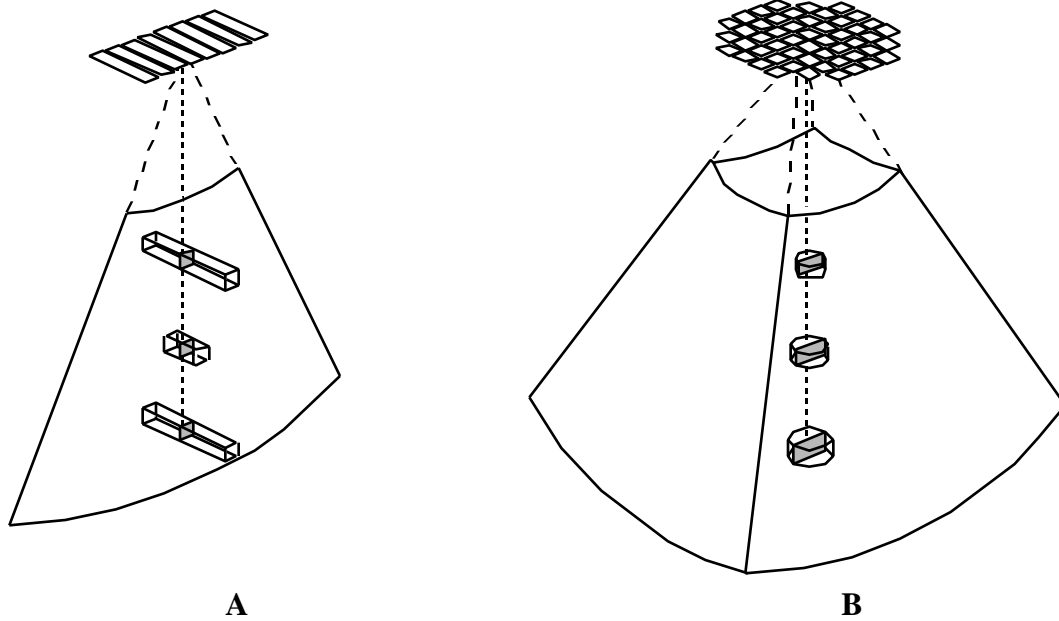
Fig. 6.3 Resolution cells in conventional ultrasound (A) and matrix-array ultrasound (B).

(3) <u>Temporal resolution</u>

The temporal resolution of matrix-array ultrasound is perhaps its strongest advantage compared to other 3D imaging modalities. The temporal resolution depends on the number of parallel receive channels operating simultaneously (Shattuck, Weinshenker et al. 1984). In the current generation of machine, 16-to-1 parallel processing can accomplish a typical cardiac scan of a 13 cm deep, 64° x 64° pyramid-shaped volume containing 4096 receive paths at a rate of 22 volumes per second. Much faster scan rates, on the order of 1000 per second, are possible if the scan is limited to a small volume. Greater parallel processing is expected to increase the scan rate in the future. Experimental techniques using multiple simultaneous interrogation also hold the promise of faster scan rates. The analysis techniques in this dissertation involve only a single time frame, and RT3D ultrasound produces such scans without gating so that no averaging across different cardiac cycles is required.

(4) <u>Anisotropy and attenuation</u>

The anisotropy of ultrasound voxels presents a serious impediment to shape identification and measurement. (Ultrasound researchers use the term *sample,* where most image analysis researchers would say *voxel* . The latter term will be used here.) In addition to the anisotropy

already mentioned due to the azimuth-elevation coordinate system, additional anisotropy arises from the physics of ultrasound. Both forms of anisotropy effect the calculated orientation and magnitude of the intensity gradient. Attenuation of the ultrasound energy introduces yet another variable which must be considered.

Ultrasound energy is transmitted and received along radial beams, resulting in a fundamental difference between components of the gradient *along* the ultrasound beam and *across* the beam. In general, the gradient component along the beam is stronger, since echoes are generated and detected in this dimension. Generally weaker gradient components across the beam derive from differences in echo strength at a given range between one beam and the next.

To compensate for the inherent discrepancy in component strength, my computation introduces a *lateral boost* parameter during the calculation of gradient components. Unlike compensation for kernel scale, which may be calculated directly from the coordinate transformation, lateral boost is dependent on the physics of ultrasound and must be determined empirically.

An optimal lateral boost would equalize the gradient components of an isotropic target. Optimal lateral boost could be determined using an isotropic target such as a sponge. At each range r and scale (refers to the boundariness aperture size), the mean value of each component of the gradient

$$\bar{I}_r(r,\ ),\ \bar{I}(r,\ ),\ \bar{I}(r,\ ) \tag{6.5}$$

could be calculated. Since and are symmetrical, lateral boost could be defined as

$$l(r,\ ) = \frac{2\,\bar{I}_r(r,\ )}{\bar{I}(r,\ ) + \bar{I}(r,\ )} \tag{6.6}$$

and applied to the components of the gradient as follows:

$$\tilde{I}(r,\ ) = l(r,\ )\ I(r,\ )$$

$$\tilde{I}(r,\ ) = l(r,\ )\ I(r,\ ) \tag{6.7}$$

$$\tilde{I}_r(r,\ ) =\ I_r(r,\ )$$

where "~" indicates that the gradient component has been adjusted for beam anisotropy. These compensated gradient components could be combined to produce a measure of gradient magnitude

$$\left| \tilde{\nabla} I \right| = \left(\tilde{I}_r^{\ 2} + \tilde{I}_\theta^{\ 2} + \tilde{I}_\phi^{\ 2} \right)^{\frac{1}{2}},$$ (6.8)

which would theoretically be invariant to rotation in physical space.

The universality of lateral compensation between targets raises interesting questions. For example, the scale   in an isotropic target such as a sponge would be determined by the structure of the target. Other methods for establishing $l(r, \ )$ could be devised, such as matching the known orientation of planar targets to the relative strength of their gradient components.

Attenuation of the ultrasound energy on its way to the target and back to the transducer greatly effects the strength of the received signal as a function of range. The receiver gain in most ultrasound scanners (including the RT3D scanner) is manually adjustable as a function of range to compensate for this signal attenuation. Receiver gain should theoretically affect all components of the gradient equally and therefore not affect lateral boost, but this should be verified empirically. It is likely that lateral boost will show at least some dependence on   and   , as well as the particulars of beam-forming such as the angular spacing between interrogation lines, etc. A thorough exploration of lateral boost lies outside the scope of this dissertation. Actual values for lateral boost have been determined in an *ad hoc* manner for the purposes of this dissertation, including a term that makes gradient component strength a function of range to compensate for attenuation, with satisfactory results.

(5) Spatial distortion

One last potential source of problems in analysis of ultrasound images is spatial distortion. This may result from several factors relating to inhomogeneity in acoustic impedance of biological tissue: (1) errors in computed distance due to differences in the speed of sound, and (2) errors in computed angular location due to refraction of the ultrasound beam. These errors are assumed to be minor compared to other sources of error.

6C. Testing Core Atoms on Ultrasound Data

As a proof-of-concept of the methods applying core atoms to RT3D ultrasound data, scans were performed on a fluid-filled balloon in a water tank and on a left ventricle (LV) of an *in vivo* human heart. These same objects will be used in Chapters 7 and 8 to validate object identification and volume measurement. The goal here is simply to demonstrate the process of determining whether a local section of a structure is roughly spherical, cylindrical, or slab-like, and thereby show that core atoms do indeed work as expected. The balloon and the LV are both considered roughly ellipsoidal.

All core atom clusters containing more than 1% of the total number of core atoms are displayed on the lambda triangle in Fig. 6.4, in the same manner as Fig. 4.4. The balloon and LV in Fig 6.4 each show a more dispersed cluster of densities than the parametric test objects used in Chapter 4. Their dimensionality seems to be roughly between that of a sphere and a cylinder, with the LV being somewhat more cylindrical than the balloon, which is consistent with their observed shapes.
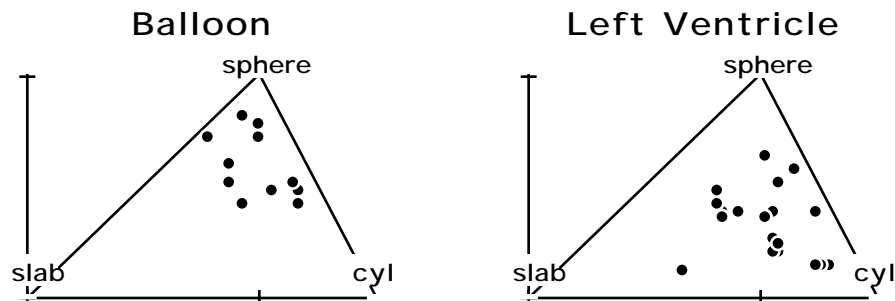

Fig. 6.4. Coronal densities containing more than 1% of the total number of core atoms.

To show that the core atoms were indeed located within the desired targets, the balloon and LV were assigned a central axis by manual placement of end-points within the data. For the LV the apex of the ventricle and the mitral valve served as the end-points of the axis. Fig. 6.5 shows densities containing more than 1% of the core atoms as a function of distance from the central axis, in the same manner as Fig. 4.5. These figures show populations of core atoms collected near the centers of the balloon and the left ventricle do indeed give the expected results when clustered and analyzed. The balloon is somewhat more spherical than the left ventricle.
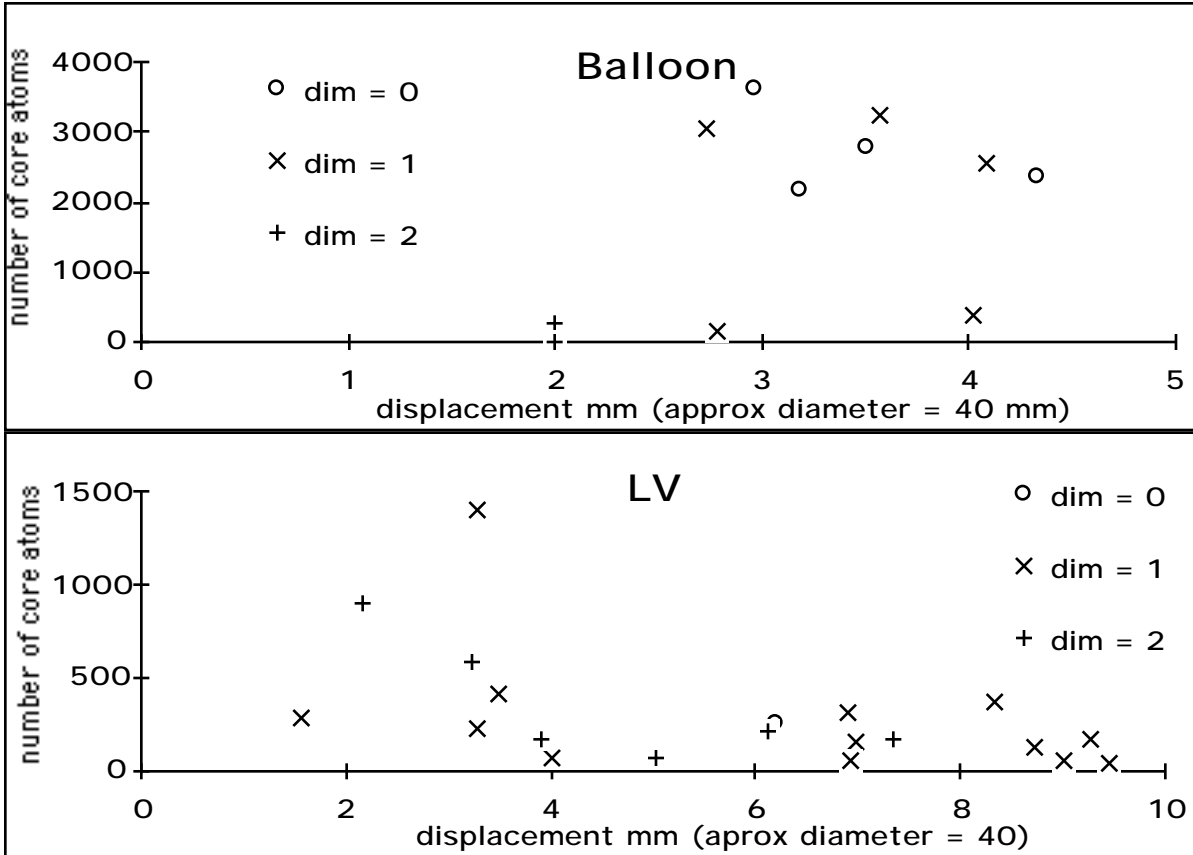
Fig. 6.5 Number of core atoms per sample vs. displacement from the manually placed axis of the balloon and the LV.

Fig. 6.6 shows the balloon and the heart, with a B-mode slice through the 3D ultrasound data, rotated so that the boundary points of core atoms appear in 3D off the slice plane. Only the boundary point in each core atom nearest to the viewer is shown lit with an intensity proportional to the dot product between its core atom vector and the viewing direction. Thus the near surfaces of the balloon and the LV are shown with pseudo lighting, while the far surfaces are automatically hidden. The use of core atoms for such visualization tasks as automated surface removal and shading is a possible avenue for future research.
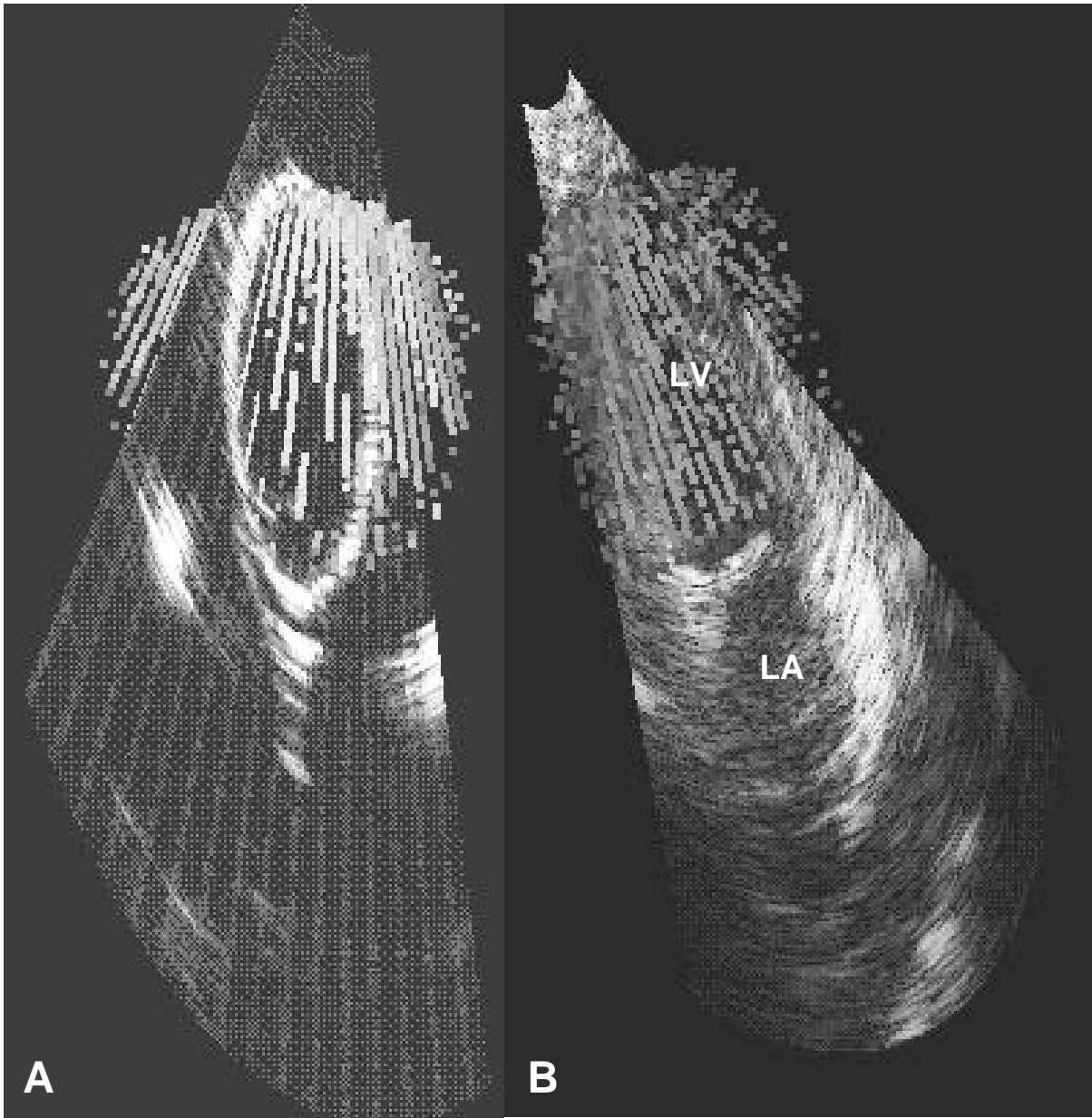
Fig. 6.6 B-mode slices through balloon (A) and *in vivo* human heart, showing the nearer boundary points to the viewer for core atoms formed near the manually-determined axes of the objects (**LV** = left ventricle, **LA** = left atrium).

6D. Rendering Slices for Manual Tracing

   To verify the goals of this dissertation it is necessary to be able to perform manual segmentation on RT3D images of *in vivo* hearts. This is undertaken by manually tracing contours on a stack of 2D slices, and visualizing the resulting stack of contours in 3D superimposed on slices that traverse the stack (see Fig. 8.6D). Although several methods already exist for such manual segmentation of RT3D data, none are completely satisfactory. Therefore a new method was devised and implemented specifically for the purpose. This section reviews the need for 2D slices and explains the limitations of the two existing methods. It then describes the new method, which provide the manual traces used to validate the segmentation of the left ventricle in Section 8E.

   Flat slices are the primary method used by clinicians and researchers for display and manual analysis RT3D ultrasound. The reason that attempts at 3D visualization have not met with great success in RT3D ultrasound bears examination. The retina is a 2D device, as are most graphical display systems. The human visual system has been described as "2½ D" rather than 3D, because it generates the third dimension from parallax, motion, relative size, shading, opacity and other cues(Marr 1982). These cues rely on the physical properties of real surfaces, which are inherently lacking in voxels, especially RT3D ultrasound voxels. Automatically determining surfaces would permit artificial 3D cues such as shading and opacity to be added, but such determination is the very goal of the dissertation. Therefore, generating 2D slices remains essential.
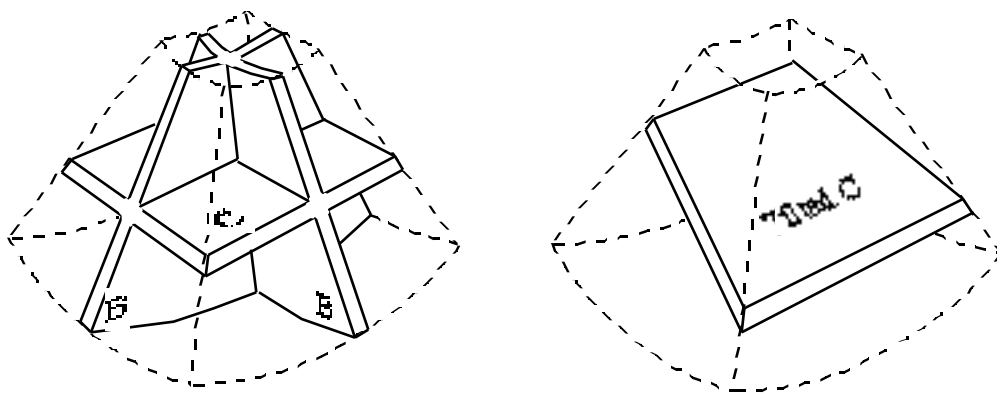

Fig. 6.7 B-mode slices are the conventional "sector" scans of ultrasound, C-mode slices are parallel to the face of the transducer and Tilted C-mode (or I-mode) scans are arbitrarily oriented.

The RT3D ultrasound scanner used in this research provides the ability to generate each of the slices shown in Fig. 6.7 in real time. As shown, slices are labeled by their orientation as *B-mode*, *C-mode*, and *tilted C-mode* (*I-mode* for "inclined"). The operator can manually trace contours on a stack of parallel I-mode slices, from which a volume is automatically calculated. However, the scanner does not provide a means of storing the resulting contours for detailed comparison with automated segmentation. Therefore it was necessary to develop an independent ability to produce slices and to understand rendering issues specific to the azimuth-elevation coordinate system.

The simplest type of RT3D slice to render is the *R-mode* slice, which is curved in physical space and contains voxels at a given range (see Figure 6.8). Unlike C-mode or B-mode slices, R-mode slices require no interpolation or coordinate transformations. R-mode slices of uniform thickness may be produced from the 3D data simply by averaging voxels over a fixed set of ranges. Within a given R-mode slice, the   and   dimensions are orthogonal and linear, making 2D gradients straightforward to calculate. However, since an R-mode slice is actually a spherical shell, significant errors can arise when tracing stacks of contours, since the human operator naturally intends the contours to be flat. In the top slices (nearest the transducer) a concave piece of the intended volume will be missing. In the bottom slices a convex piece of *un*intended volume will be present. These error may be negligible for an object such as a balloon, whose top and bottom slices have very small contours. However, an apical scan of the left ventricle produces a stack of contours whose bottom (at the mitral valve) is quite large. That contour, traced as if it were flat, really bulges down into the left atrium contributing a significant unintended volume to the measurement. As a result, a method for generating I-mode slices is necessary and is described later in this section.
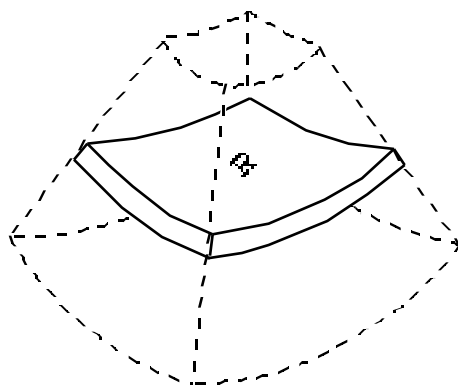

Fig. 6.8 R-mode slice contains voxels at a given range from the transducer.

71

The next-simplest type of slice to render is the B-mode slice, which in data space is a rectangle and projects in physical space to the truncated sector of a circle (see Fig. 6.9). The rectangle in data space contains all voxels with a particular value of   (or   ) and represents a simple 2D sub-array of the 3D data. The projection can therefore be accomplished using 2D texture mapping, a standard graphics technique that interpolates a rectangular array of pixel values onto an arbitrary four-sided polygon. By breaking data space into long radial rectangles and mapping each of these onto a long radial trapezoid, the sector can be rapidly projected using standard inexpensive graphics hardware (Ohazama 1998).
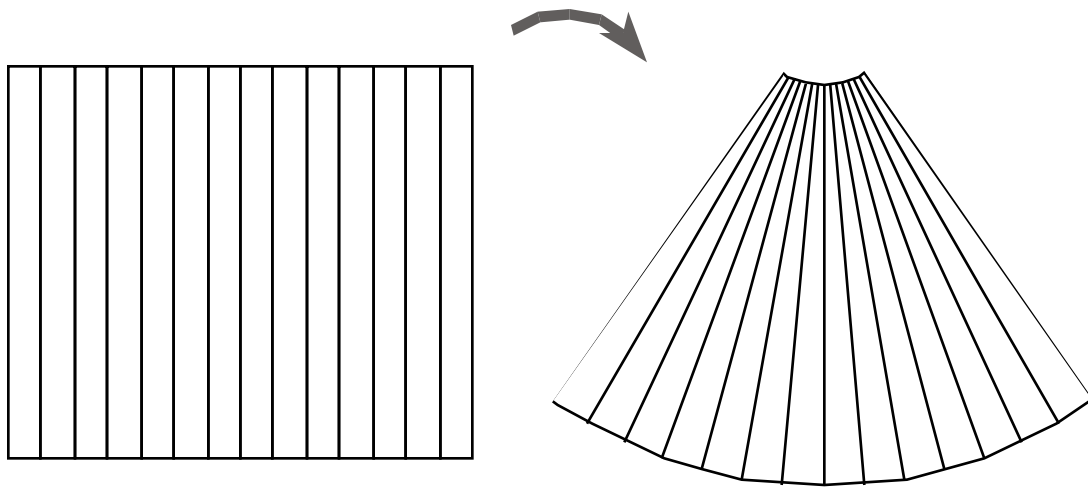
Fig. 6.9 B-mode slice rendered using 2D texture mapping of radial ribbons onto trapezoids.

The empirical validation of the core atom method in Chapter 8 is accomplished for *in vivo* hearts by tracing short-axis slices through apical scans. A stack of these short axis slices roughly parallel to the transducer is established by selecting end points manually and then slicing orthogonal to the axis at regular intervals. Inclined C-mode (or I-mode) slices are produced using a new method of forward projection developed during the dissertation research.

Rendering slices through a 3D data set may be accomplished with two basic approaches: forward projection and backward projection . In backward projection, or *raycasting*, each pixel is projected back along a ray through the volume to collect interpolated effects from voxels near the ray. In forward projection each voxel is mapped onto the image, where its effect is combined with that of other voxels. The above methods of rendering R-mode and B-modes slices are both examples of forward projection.

I-mode slices cannot be rendered so simply. Backward projection using specialized *3D texture mapping* hardware is possible by interpolating between voxels in 3D and projecting

onto meshes that are curved in data space but flat in physical space (see Fig. 6.10). Such a system exists, but is very expensive (Ohazama 1998).
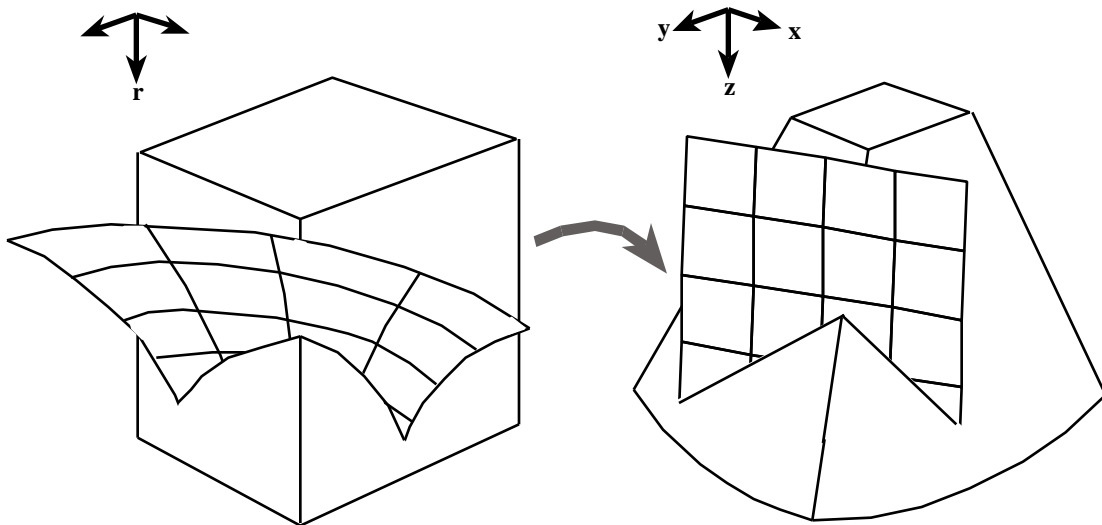

Fig. 6.10 B-mode slice rendered using 2D texture mapping of radial ribbons onto trapezoids.

A new practical alternative for creating I-mode images using forward projection has been developed as part of the research for this dissertation. Forward projection was chosen due to its advantages over backward projection: (1) It permits any arbitrary subset of voxels to be projected, (2) it guarantees that each voxel contributes equally to the image, and (3) it permits a straightforward pipeline architecture to display a data stream without first storing it.

Two basic types of forward projection, *splatting* and *cell projection,* have been widely explored. In splatting, the footprint varies along its cross-section and the voxel contributes more to pixels at the center of the footprint than at its edge. The footprint thus resembles a point spread function as described in optics or in many imaging systems. In cell projection, on the other hand, the footprint is binary and all pixels are treated equally within the footprint. The only decision in cell projection is whether a pixel lies inside or outside the voxel's silhouette.

Cell projection of RT3D ultrasound voxels must take into account the variable shape of the voxel due to the azimuth-elevation coordinate system. Since the r , , and axes are locally orthogonal, the voxel will always approximate a right parallelepiped. Such a description produces a space-filling voxel as shown in Fig. 6.11. The variable has been substituted for r in the diagram and in the following discussion to emphasize that any locally-orthogonal coordinate system can be projected using this method. The relative lengths of a voxel's sides may change with location, and it may be projected onto the image using any affine transform.

The affine transform permits rotation, translation, and anisotropic scaling. The resulting projection will always be a hexagon whose opposite sides have equal lengths, as shown on the right-hand side of Fig. 6.11. The symbols $\vec{}$, $\vec{}$, and $\bar{}$ denote the 2D projections onto an image of the corresponding unit vectors at the voxel location.
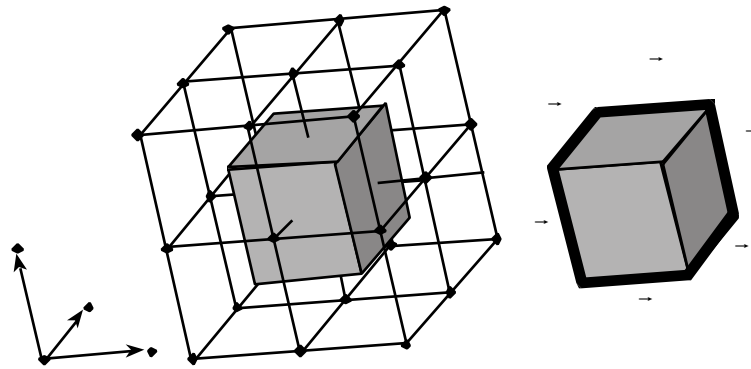

Fig. 6.11 Space-filling RT3D voxel at the center of a grid of 27 neighboring voxel locations, and its projection on the image as a symmetrical hexagon.
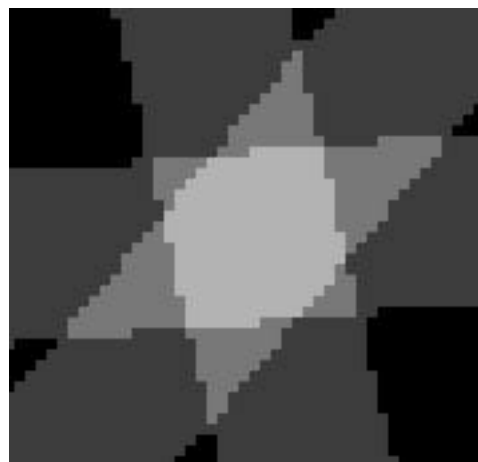

Fig. 6.12 The intersection of 3 stripes forms a hexagon with opposite sides of equal length.
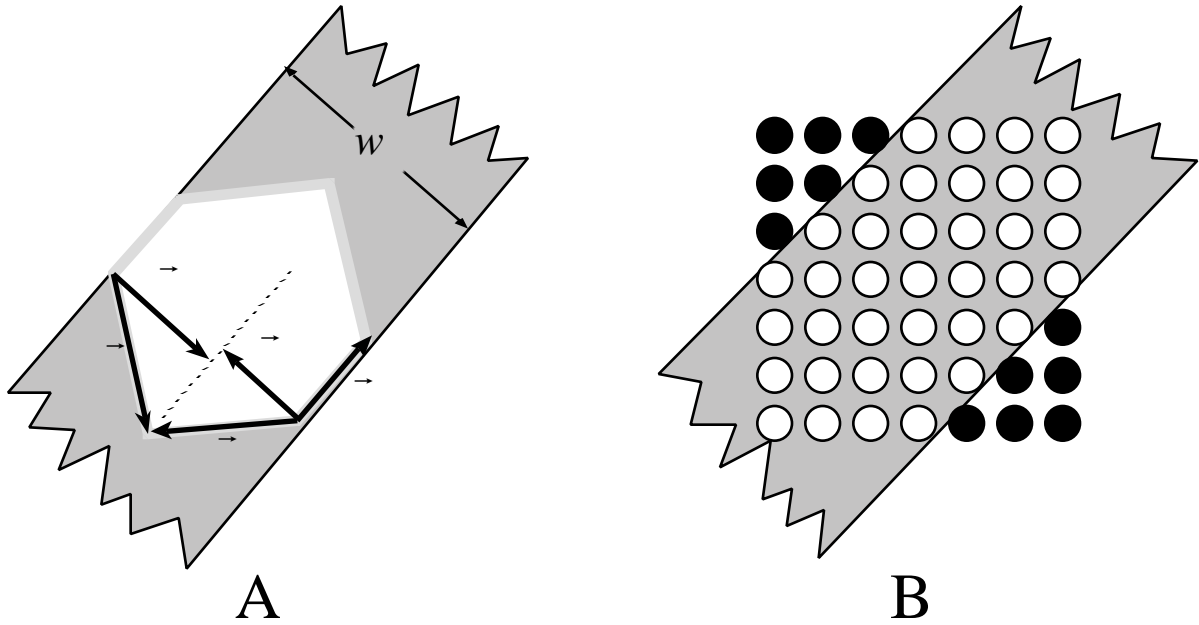
74

Fig. 6.13 The intersection of 3 stripes forms a hexagon with opposite sides of equal length.

The silhouette of the voxel shown as a thick line in Fig. 6.11 can be created by the intersection of 3 stripes, as shown in Fig. 6.12, in the $\vec{}$, $\vec{}$, and $\vec{}$ directions respectively. The hexagon appears as the lightest area in the center. Determination of which pixels lie inside each stripe proceeds as shown in Fig. 6.13, based on the projected local unit vectors of the voxel $\vec{}$, $\vec{}$, and $\vec{}$. The width of a particular stripe, in this case the stripe in the $\vec{}$ direction, is designated w , and can be computed as

$$w = \left\| \vec{} \right\| + \left\| \vec{} \right\| \tag{6.9}$$

where

$$\vec{} = \vec{} - \left(\vec{} \bullet \hat{} \right) \qquad \vec{} = \vec{} - \left(\vec{} \bullet \hat{} \right) \tag{6.10}$$

and, as usual , $\hat{} = \vec{}/\left\| \vec{} \right\|$. By precalculating a set of stripes of varying width and orientation, a reasonably fast rendering algorithm has been implemented based on this approach.

To render an I-mode slice with the above algorithm, it must be determined which voxels are within a given slice. This is done by pre-sorting them throughout the data space into bins numbered by distance along an arbitrary axis. Slices orthogonal to the axis can then be quickly

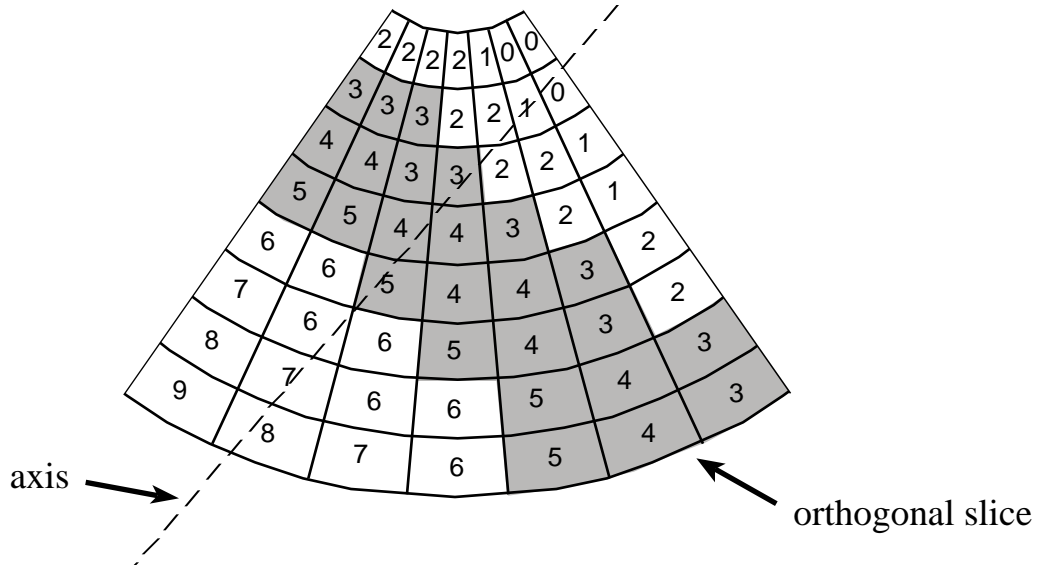assembled from contiguously numbered sets of voxels. This concept is illustrated in Fig. 6.13.

Fig. 6.13 Pre-sorting of voxels by distance along an arbitrary axis for rapid selection of orthogonal slices.

This procedure for projecting I-mode slices of arbitrary orientation and thickness was used to produce 2D images for manual tracing, yielding endocardial boundaries for comparison with the results from the automated method. These experiments are discussed in Chapter 8.

Before leaving the topic, a fundamental limitation should be noted in the analysis of 3D structures using 2D slices. Any stack of slices implies a choice of orientation, which may not be optimal for that particular shape. In some cases, no suitable orientation may exist at all. The human right ventricle, for example, defies straightforward slicing. The method based on 3D core atoms does not require such a choice to be made. This is one of its primary advantages.