



Faculty Publications

1990-05-01

Interactive Morphometrics from Three-Dimensional Surface Images

William A. Barrett
william_barrett@byu.edu

Shawn C. Becker

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Computer Sciences Commons](#)

BYU ScholarsArchive Citation

Barrett, William A. and Becker, Shawn C., "Interactive Morphometrics from Three-Dimensional Surface Images" (1990). *Faculty Publications*. 733.

<https://scholarsarchive.byu.edu/facpub/733>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Interactive Morphometrics from Three-Dimensional Surface Images

Shawn C. Becker and William A. Barrett
Department of Computer Science
Brigham Young University

Abstract

Techniques have been developed for extracting three-dimensional (3D) measurements directly and interactively from 3D representations of surface anatomy. In order to obtain 3D measurements the anatomical surface is interrogated directly through the use of morphometric tools. Morphometric tools include points, 3D graphical "probes", and logical sectioning functions which interact with the surface through depth buffer compositing operations. Tools are interactively positioned on or through the surface to measure surface dimensions and partial volumes, or to define intersecting functions which allow surface anatomy to be visualized in the context of surrounding anatomy. Visualization is also enhanced because tools reveal much pertaining to surface morphology while providing valuable depth cuing for visual feedback and probe placement. Since tools are implemented using a general depth buffer, the techniques are applicable to a variety of 3D imaging algorithms. Moreover, it is believed that the morphometric tools described in this paper will greatly enhance the diagnostic utility of 3D visualization techniques.

Introduction

Scientific visualization of multidimensional data has recently received much attention. In particular, a variety of techniques have been advanced for visualization of three-dimensional (3D) volumes containing data from such diverse sources as medical imaging, astronomy, seismic recordings, and X-ray crystallography. The ability to understand and learn from the data contained within these image volumes is essential to those that work with them. One method of preparing the data for human understanding is to visualize it by generating 2-dimensional views (projection images) of the volume using surface or volume rendering techniques. Image volumes are frequently large and complex ($\geq 256^3$ bytes) making the rendering operation a compute-intensive task. Consequently, much research has been devoted to improvements in the quality and speed of rendering algorithms.

Visualization alone, however, does not provide all of the information needed to understand the data contained within the image volume. Equally important is the ability to interactively study the morphology (i.e. shape, size and spatial interrelationships) of objects in the sampled volume. Interactive morphometric tools (computer tools which allow interactive measurement of 3D morphology) are needed to extend the observer's ability beyond mere "seeing" of 3D structure to virtually "touching and feeling" or quantifying the object. The purpose of this paper is to describe enhancements to scientific visualization of 3D image volumes through the development of tools for interactive morphological measurement.

The Surface Image

In the context of this paper, a surface image is the depiction of a surface within an image volume using any rendering technique. Most volume visualization methodologies make use of either surface rendering or volume rendering techniques. The techniques described in this paper can make use of either technique as long as an apparent surface or surfaces are generated. Depiction of surfaces is a realistic objective in medical imaging since skeletal anatomy and organ boundaries consist of well-defined tissue interfaces (within the limits of the detector technology). In other words, any "local mixing" of tissues at organ boundaries is not caused by different tissue types actually intermixing or overlapping, but rather, partial volume effects resulting from a finite detector geometry.

Unlike many recent techniques which defer the segmentation/surface detection problem to the visual apparatus of the user, our methodology [1-2] produces an explicit surface description of the imaged anatomy, resulting in rendered anatomical surfaces which are highly realistic and quantifiable. We believe that the latter will become increasingly more important, not only as an adjunct to visualization, but particularly to enable more precise clinical diagnosis and treatment, more accurate presurgical planning and postsurgical assessment, and a solid quantitative basis for following the time course of a disease process. Beyond

the basic need for a surface representation, realism and image quality are the primary factors which influence the selection of a particular rendering technique.

Surface Extraction and Rendering

The process of surface extraction and rendering is performed in four basic steps: (1) formatting the 3D image volume into a cubic voxel array (2) segmenting the array into object and nonobject (3) tracking the object surface and outputting a 3D surface description (x,y,z position and dx,dy,dz orientation of each voxel face on the surface) and (4) rendering of the surface at one or more views based on the surface description.

A commonly held misconception is that segmentation of the image volume into a binary array consisting of object and background will necessarily result in a rendered surface image which is "blocky" and models the discrete nature of the segmentation and detector geometry. However, consider the surface description simply as the best "nearest voxel" surface representation which can be obtained with a discrete sampling grid. The surface rendering, then, merely uses the surface description as a foundation on which to build the final image. By using this foundation as a first order approximation of the location and orientation of the surface, and then fine tuning the rendered surface intensity through careful estimation of the surface normal (using image space, contextual, or actual density gradient shading), highly realistic images can be obtained which often reveal subtle details beyond the resolution of the surface description (Figure 1).

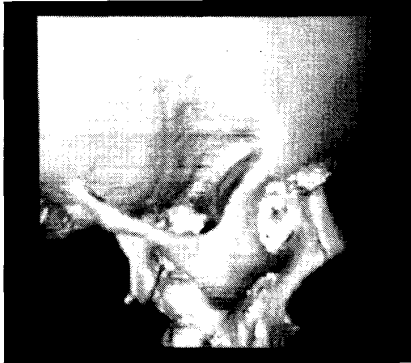


Figure 1. Surface image using density gradient shading.

The important point to be made in the context of this paper is that an explicit surface description provides a depth or z-buffer component to the rendered surface image for any given view, which allows direct quantitative interaction with the surface anatomy. This is described under Interactive Morphometric Tools, below.

Because of current hardware limitations, rendering calculations are not possible at desired real-time rates. Therefore, in order to provide interactive viewing capability, a set of views is precalculated in a matrix of tilts and spins (based on a set of view transformations) for later interactive view selection. Treatment of the rendering process as a background task yields some advantages: limitations of rendering speeds do not limit the use of these images and the user may interactively select a view from the provided set of views. Since many tilts and spins are provided, the user has a great deal of control over view orientation. The rendering process can be as involved as desired to result in the best possible visualization environment without adversely affecting its interactive use.

Interactive Morphometric Tools

Interactive morphometric tools refer to a set of geometric primitives which can be used to extract measurements from and enhance visualization of surfaces and objects contained within an image volume. Tools include points, lines, planes, and volumes (defined by swept curves), which interact with a depth buffer to facilitate visualization and measurement techniques.

Depth Buffer Composition

If certain steps are taken during the rendering stage to include surface depth with the surface image, more than mere interactive visualization from the precalculated surface images can be achieved. Depth or z-buffer values at each pixel are useful for **composition** of added objects during interaction. Composition allows new parts to be added to a surface image. The visual result is as if the scene has been completely re-rendered. In actuality, only the added parts have been sent through a viewing pipeline. Simple z-buffer composition allows 2 or more images with depth information to be mixed so that those objects with z-values that put them in front (closer to the viewer) will obscure objects behind them (further from the viewer). A z-buffered surface image can have simple geometric primitives composited into it with the advantage that z-buffered simple primitives can be generated at real-time rates and are excellent tools for interacting with a complex surface.

$$x_v[v] = T[v] \cdot x_w \quad [1]$$

$$x_w = T^{-1}[v] \cdot x_v[v] \quad [2]$$

The **view transformation**, T , is used to transform each x,y,z world coordinate, x_w , into a corresponding x',y',z' view space coordinate, x_v , for a given view, v (Equation 1). Each x',y' pixel in a given view consists

of a z-buffer value, z' , of the visible surface coordinate that mapped to that location. If the **inverse view transformation**, T^{-1} , is stored with each rendered view, v , then a view space coordinate, x_v , in that image may be used to determine original world space coordinates, x_w , of the corresponding visible point on the surface (Equation 2). This allows the user to not only to see a representation of the object from a given orientation but also to recover original information pertaining to the object's surface.

A hierarchy of primitives that operate on these principals have been developed. The most basic of these is the surface point. Higher order primitives, linear probes, curves, planes, and volumes can be interactively defined by using cooperating sets of surface points. The result is a simple but effective interactive environment that allows the user to see as well as "touch and feel" objects contained within the image volume.

Surface Points

The user can attach a point to the surface of an object with a conventional 2D mouse driven cursor. The cursor is moved and by clicking the button selects a visible location on the surface. A **surface point** is deposited and visualized at that point on the surface (Figure 2 below). The surface point's location may be manually changed by selecting an existing point and dragging it along the surface to a new location. Once deposited on the surface, a point is statically attached to the object. This means that as the user interactively "tilt/spins" the scene the surface point appears to be statically attached to the surface and rotates with it.

The surface point is actually an artificial part of the scene. The cursor position and z-value at that location in the surface image are used with the view to world transformation (Equation 2) to determine the world coordinates of that surface point. View to world coordinate conversion occurs whenever a surface point is deposited or dragged over the surface. The surface point is an object in a geometry database that defines a location in world coordinates. As the view changes, the surface point is sent through that views' rendering pipeline (Equation 1) to get new view coordinates for proper visualization. The surface point is composited into those view coordinates of the image through the use of the z-buffer.

The z-buffer is used to approximate the location of a voxel face visible from a given pixel in the image plane. If greater accuracy is needed, a sub-voxel position can be obtained by resampling original grey scale data of n voxel neighbors in the column projected to the pixel. Since a surface specified by the segmentation is known to exist somewhere in this column, a convolutional filter which matches the grey values in the neighborhood of the discrete

boundary can be applied to determine the precise surface boundary. The position that maximizes the convolution determines the sub-voxel location of the surface. This technique has been used to provide sub-pixel precision in other medical imaging applications [3].

World coordinate values in terms of voxel units or some standard unit of measurement (cm) are a useful parameter and can be echoed during point specification. Voxel to cm conversion requires scaling voxel coordinates by the acquisition resolution factor of the image volume.

Visualization of a point on a 3D surface is more realistic when it has visible 3D characteristics. For example, the surface endpoints in Figure 2 have been made to look like a spherical metal beads. Visibility is improved by enlarging the radius of the bead. Visibility is also improved by giving the bead surface characteristics (by modifying its specular coefficient) which differ from those of the underlying anatomy. Since the bead has its own image and z-buffer, rendering a bead on a surface image with a z-buffer is a simple composite operation. Since all beads of a given radius look the same, a bead icon can be precalculated. During interactive placement the bead z-values are increased by the z-value of the center surface point and then composited into the original surface image.

Surface orientation is another property of a surface point. The surface normal can be easily approximated from the gradient of its immediate neighborhood in the z-buffer. This is the method used to determine surface normals by image space gradient shading. A more precise surface normal can be extracted from the surface description or recalculated by resampling from its neighborhood of voxels in the image volume. Visualization of all 6 dimensions (position and orientation) of a surface point may be useful during interactive specification or editing. This can be accomplished by using a cylindrical extension of the bead in the direction of the surface normal.

The techniques required to quantify and visualize 3D surface characteristics at a given location can be implemented very effectively using the z-buffer. The z-buffer allows direct access to original world coordinates when the view coordinates and view transformation is available. The z-buffer provides sufficient information to approximate the surface normal without accessing the original surface description or resampling the image volume. The z-buffer also makes interactive visualization of surface primitives possible through the use of compositing techniques. Compositing allows geometric primitives such as beads to be integrated into the scene without requiring costly re-rendering of the surface image.

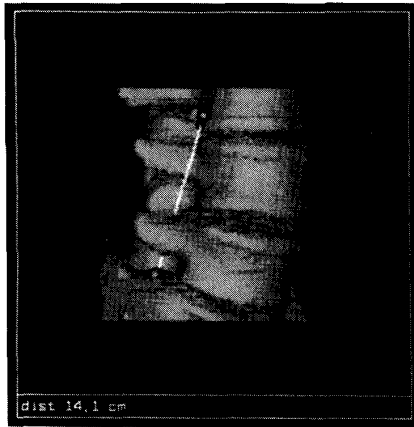


Figure 2. Two Surface Points and a Surface Probe stretched between them.

Surface Probes

A wide range of 3D measurements may be obtained by interrogating the surface of an object directly through the use of a 3D graphical "probe" (line). By using one surface point as an anchor and another as a moving end, a probe can be interactively positioned and stretched between them like a 3D rubber band line (Figure 2). However, by making use of the z-buffer, the 3D probe becomes more than a simple rubber band line. Mouse-defined probe endpoints (A and B) in view plane coordinates (Figure 3) have corresponding depth values (A_z, B_z) obtainable directly from the z-buffer. A modified Digital Differential Analyzer algorithm [4] is used to interpolate (x_v, y_v) pixels and associated (z_v) depth values between the endpoints.

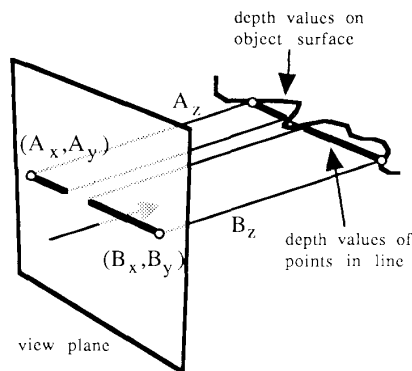


Figure 3. Probe visualization using depth buffer described surface.

By compositing the line into the surface image during interaction it dynamically "penetrates" any intervening surfaces. This provides valuable depth cuing for visual feedback and probe placement. In addition, 3D

euclidean distance measurements can be obtained directly in world coordinates while the probe is displayed from any view orientation, as described below.

Like the surface point, a 3D probe is entered into the geometry database and is rendered correctly for any view. The effect is that during rotation the probe remains statically attached to the surface and appears to be an integral part of the scene. Then when the surface is orthographically projected, the pixel coordinates and associated depth value are simply the result of the surface point's world coordinates being sent through the view transformation (Equation 1). Linear measurements using this approach have been shown to be both accurate (limited by scene and image resolution) and reproducible when taken from different orientations. While the dynamic nature of the probe cannot be adequately demonstrated with a static image, note the dimensional measurements and spatial interrelationships of surface anatomy indicated by the probe in Figure 2.

A 3D surface curve may be specified as a series of head-to-tail connected probes. Curvilinear measurements can then be obtained by summing the linear distances between multiple user-defined surface points. Another curve measurement technique amounts to projecting a probe onto the surface of the object. Each pixel in the probe has corresponding surface point coordinate information available in the depth buffer. Accurate curve distance can be thus calculated by summing linear distance between pairs of visible surface points in the probe's path. The piecewise linear method may be appropriate for taking point to point cumulative approximations of object geometry dimensions. The projected curve method, which may be likened to laying a measuring tape directly on the object's surface, provides more precise circumferential measurements.

Probes may also be given characteristics that help make them more distinguishable against the surface image, (i.e. cylindrical radius and contrasting reflectance). If the probe endpoints are attached to a convex surface the probe itself may be hidden completely by the surface. In this case, interactive extension of the probe beyond its endpoints promotes visibility.

Display of quantitative vector and magnitude measurements are useful during interactive probe specification. The visual feedback provided by the 3D probes and curves conveys much to the user pertaining to 3D surface geometry. In fact, it may provide more information about subtle variations in surface shape than 3D stereo viewing.

Surface points were introduced above as being constrained to lie on the surface of an object. Similarly, point motion can be constrained to lie along a given probe. This type of interaction technique enables one to "slide a bead along a wire". Such a combination of a line and point make what is essentially a three-dimensional scrollbar. The visualization rules for points may need to be a little different if they are allowed to ignore the surface constraints normally placed upon them and pass through a surface. When a point needs to be visible beneath the object's surface an **alpha blending** method is used to simulate transparency of the surface.

The degree to which a point is constrained to a probe is another useful property. For example, in some cases the point should not be allowed to exceed the probe's endpoints. A probe constrained point has several applications, one of which is to specify a given plane.

Planar Specification

One simple way to describe a plane of interest within an image volume is to choose an orthogonal plane orientation (i.e. the yz plane) and a specify its distance from the origin (along the x axis). Interactive orthogonal plane specification can be implemented using point and probe tools that have already been introduced. It requires a probe aligned to a major axis (x) and a point constrained to it. The probe defines the normal axis and the sliding point represents the point on the plane nearest to the origin (Figure 4.5a).

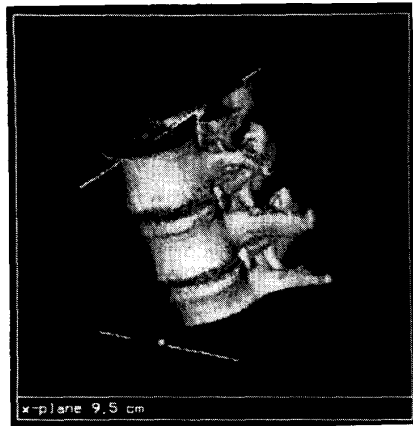


Figure 4. Sliding orthogonal plane specification with "back plane resampling".

This is an example of probe and point-constrained plane specification. A more general approach to plane specification would be to use an arbitrary probe as the plane normal vector (Figure 5b). By sliding a point along the probe we define a sliding plane. By rubber-

banding an endpoint of the plane's normal vector probe the plane will pivot about the probe's anchor point (Figure 5c).

Another point/probe-based plane specification technique takes advantage of the fact that any 3 noncolinear points define a plane. By using 3 points (2 static and 1 moving) a tilting plane specification technique is possible (Figure 5d).

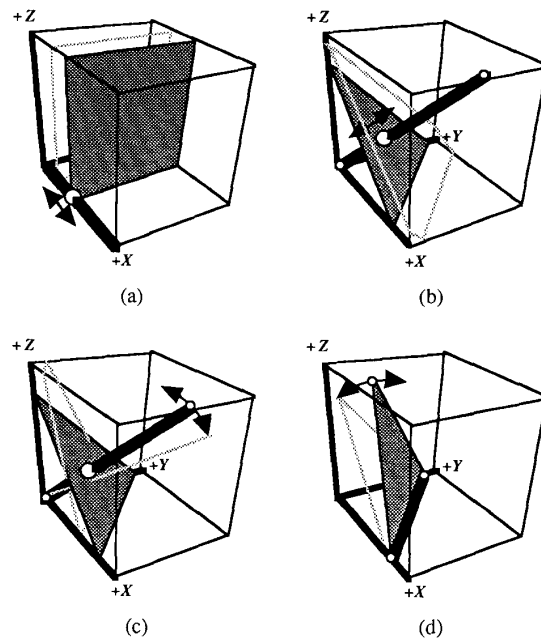


Figure 5. (a) sliding orthogonal, (b) general sliding, (c) pivoting and (d) tilting plane specification techniques.

In order to make interactive plane specification truly believable, fast visualization techniques are needed. Most importantly, re-rendering of the surface image needs to be avoided at all costs, since complex time-consuming rendering techniques are allowed as a preprocessing stage.

The z-buffer is well suited to interact with a plane of z-values. By using forward difference techniques the z-values of a plane can quickly be calculated at every pixel in a specified view. By using image composition techniques z-values of the geometric plane and the surface image can be quickly compared on a pixel by pixel basis. Visualization decisions can be made that depend upon whether the plane lies in front of, behind, or on the surface of the object for each pixel in the view.

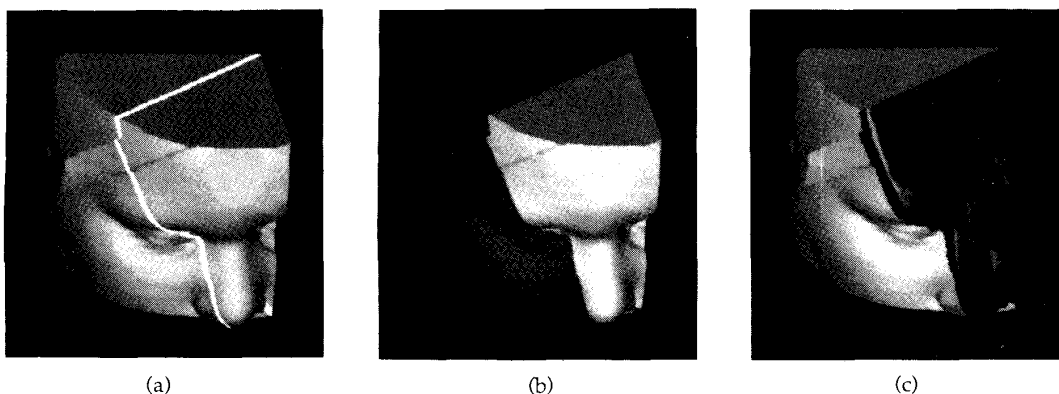


Figure 6. Plane visualization by (a) Wire frame intersection, (b) Plexiglass shading and (c) Front clipping plane resampling.

By varying rules and actions slightly, several different visualization methods can be achieved. A curved line can be made to appear where the plane and surface intersect. This is done by highlighting the intensity of pixels that have equal (or near equal) plane and surface image depth values (Figure 6a). The surface can appear to be shaded by the plane by re-mapping the color of surface image pixels when the surface image lies behind the plane (Figure 6b).

More sophisticated techniques are possible if resampling of original density data is allowed. The plane is treated as a cutting plane and the following conditions and actions are used to visualize the newly exposed density data:

```

int x,y      view pixel coords
surf_image[] view data image+zbuf
planez      plane view depth value
int i,j,k    world coords from view coord
             x,y,planez
world[][][] world data

/* front plane resampling */
if surf_image[x][y].z in_front_of planez then
  view_to_world(x,y,planez, i,j,k);
  surf_image[x][y].img = world[i][j][k];
  surf_image[x][y].z = depthz;
endif

```

Superimposition of the interior image data on the cutting plane improves understanding of the surface/plane geometry by allowing the original image (density) data to be visualized and analyzed in the context of the surrounding surface anatomy (Figure 6c). This is more than satisfactory compensation for the temporary loss of grey scale data during the initial segmentation and rendering stage.

Another technique (Figure 7) uses the plane to reveal what the surface image has been "hidden in" (ie. the visual equivalent of digging dinosaur bones out of the earth). It uses resampled density data with an inverted condition of the above technique:

```

/* back plane resampling */
if surf_image[x][y].z in_back_of planez then
  view_to_world(x,y,planez, i,j,k);
  surf_image[x][y].img = world[i][j][k];
  surf_image[x][y].z = depthz;
endif

```

If an extension of the previous analogy may be permitted, the "excavation" can even be visualized dynamically, allowing enhanced visualization of surrounding background or soft tissue structures in the context contained surface anatomy.

Just as points and lines can be used to constrain planes, a plane can form a constraint surface for point and line specification. Planes may also be used to modify the z-buffer description of the surface image. In this way planes can be used to specify a new volume of interest.



Figure 7. Rear clipping plane resampling.

Volume Specification

The volume of the entire rendered surface can be obtained by integrating over the surface description along one of the major axes, and if total volume is the only parameter of interest, it need not be computed "interactively" or in real time. However, frequently it is desirable to be able to subdivide the anatomy and compute the partial volume of each component part. For example, in Figure 8, [5], it may be of interest to obtain the left ventricular volume exclusively, but the surface consists of the left ventricle and a part of the left atrium and aorta.



Figure 8. Volume Specification using multiple (2) "half-space" decision functions for left ventricle, atrium, and aorta

The left ventricular volume could be obtained exclusively by redefining a segmentation plane and regenerating the new (clipped) surface. However, calculation of partial volumes can be performed without retracing or rerendering the surface anatomy through use of the original surface description and interactively defined segmentation planes. Through definition of a new (cutting) plane, $p_A(x)$, (Figure 9), the anatomy enclosed by the original surface can be logically segmented into two parts, A and A' . The respective volumes, V_A and $V_{A'}$, can be obtained by computing the "decision function", $p_A(x)$, for each surface face, x_s , and then adding the function value to the respective volume. The algorithm is given below.

Input: Cutting plane, p_A
 Surface faces, x_s (in $+x$ and $-x$ direction)
 (face positions can also be with respect to y or z axes)
Output: Volumes, V_A and $V_{A'}$

```

 $V_A <- V_{A'} <- 0$ 
for all  $x_s$ 
  if  $p_A(x_s) \leq 0$ 
    then  $V_A <- V_A + p_A(x_s)$ 
    else  $V_{A'} <- V_{A'} + p_A(x_s)$ 
    
```

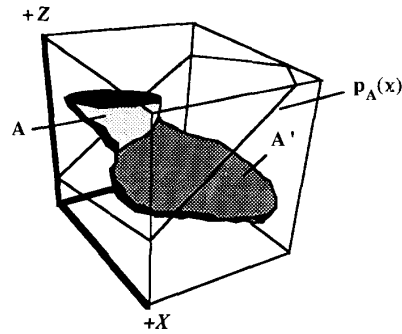


Figure 9. Interactive segmentation plane for calculation of partial volumes.

In practice, the coordinate used for face position would be the one with the minimum incremental difference (slope) to avoid summing against a plane with infinite slope.

Through generalization of the techniques described above, objects can be spatially segmented for volume measurements by using multiple intersecting "half-spaces" and swept curves as volumetric decision functions. For example, smaller partial volumes could be obtained through testing of faces against combinations of cutting planes (ie. if $(p_A(x_s) \leq 0)$ and $(p_B(x_s) \leq 0)$... then). In this way, half-spaces logically remove all object points above some plane or set of planes which are defined by triplets of surface points (beads) in Figure 8. Swept curves, on the other hand, can be defined using simple 1D primitives (lines and curves) which are swept about and/or along a probe-defined line to logically remove all object points outside of the closed volume. In general, any function(s), $f_A(x)$, (spheres, bicubic surfaces, etc.) could be used to isolate and calculate partial volumes.

The important factor for interactive determination of partial volumes is real time graphical update and positioning of logical segmentation plane(s) or functions. Logical volumes can be visualized using the z-buffer based plane visualization techniques described above.

Conclusions

Depth buffers are commonly used to solve hidden surface problems in the rendering of objects contained within image volumes. However the depth buffer provides additional utility during interactive visualization and measurement when it is used to define the visible surface of the objects. Without ever accessing or changing the data within the image volume, probes, curves, planes and volumes may be interactively specified on the object's surface. These tools provide valuable visual cues and shape measures that extend the observer's ability beyond mere "seeing" of 3D structure to virtually "touching and feeling" or quantifying the object.

The need for tools that allow quantification of real world data in applications such as biomedical visualization has been expressed by other investigators [7,10]. The simple morphometric tools described here enhance scientific visualization of 3D image volumes by supporting direct interactive measurement of anatomical shape, size, and surface dimensions. Furthermore, implementation of the tools using a general depth buffer makes them applicable to a variety of surface or volume rendering techniques. However, even more important than this is the hope that interactive morphometric tools will also greatly increase the diagnostic utility of 3D visualization techniques.

References

- [1] S.P. Raya, J.K. Udupa, and W.A. Barrett: "A PC-based 3D imaging system: algorithms, software, and hardware considerations", Medical Image Processing Group, *Technical Report MIPG147*, Department of Radiology, University of Pennsylvania, Philadelphia, 1989.
- [2] W.A. Barrett and J.K. Udupa, "A Low-Cost PC-Based Image Workstation for Dynamic Interactive Display of Three-Dimensional Anatomy", SPIE Medical Imaging, Newport Beach, (1989).
- [3] W.A. Barrett, A. Siebert, H. Hines, and P. Schiebe, "Automated Detection of Coronary Arteries and Quantitation of Percent Stenosis from DSA Images: A Comparison of Geometric and Densitometric Techniques," in IEEE Proceedings of Computers in Cardiology, 1984, pp. 123-126.
- [4] D. Field, "Incremental Linear Interpolation", ACM Transactions on Graphics, 4(1):1-11 (1985).
- [5] Barrett, W.A. and Udupa, J.K.: Dynamic Display and Quantitative Analysis of Three-Dimensional Left Ventricular Pathology, Proceeding of the IEEE, Computers in Cardiology, Bethesda, MD, 1988.
- [6] Eric Allan Bier, "Skitters and Jacks' Interactive 3D Positioning Tools", Proceedings 1986 Workshop on Interactive 3D Graphics, Chapel Hill, 183-196 (1986).
- [7] Fred L. Bookstein "Computer Graphics for the Biometrics of Shape", NCGA Computer Graphics '85, 3:161-170 (1985).
- [8] Lih-Shyang Chen, Gabor T. Herman, Anthony Reynolds, and Jayaram K. Udupa, "Surface Shading in the Cuberille Environment", IEEE Computer Graphics and Applications, December 33-43 (1984).
- [9] James D. Foley, Victor L. Wallace and Peggy Chan, "The Human Factors of Computer Graphics Interaction Techniques", IEEE CG&A, November 13-48 (1984).
- [10] Sushma S. Trivedi, Gabor T. Herman, Jayaram K. Udupa, Prasad Margasahayam, Lih-Shyang Chen, "Measurements on 3D Displays in the Clinical Environment", Proceedings of NCGA '86, 93-100 (1986).