

Display Methods for Grey-Scale, Voxel-Based Data Sets

The dramatic increase in the use of 3D image acquisition devices over the past decade has inspired major new developments in the display of volume data sets. In this chapter we present an overview of these diverse display methods and discuss the relative advantages and disadvantages of each of the different approaches. In addition, we touch upon some of the major issues involved in creating high-quality images from volume data, including the problems of surface definition and object segmentation.

Due in part to the rapid, almost frantic, pace of recent developments in methods for rendering images from volume data, there has not yet emerged any widely accepted taxonomy for these methods. Because the human visual system is adapted for environments in which images of surfaces predominate, most algorithms emphasize in one way or another the display of surface-like information, either implicitly or explicitly. For clarity, we will avoid using the terms "surface rendering" and "volume rendering" to describe the various methods, since although prevalent in the literature they have no precise, commonly accepted definitions. Instead, we will differentiate the various rendering methods using the following three characteristics, which are somewhat more precise and, we hope, less misleading: 1) whether the explicit creation of an intermediate surface representation is required (if so, then we will refer to the method as "indirect"), 2) whether the method is designed to operate on binary or on grey-scale data, and 3) for methods that operate on grey-scale data, whether a binary decision must be made about the existence of a surface at any given location in the volume.

11.1 Indirect methods

11.1.1 Tiling methods

One of the earliest approaches used to create images from volume data sets involved the construction of three-dimensional polygonally-tiled surfaces from planar contour curves defined on successive two-dimensional slices through a volume {KEPPEL75}. In the simplest cases, a single closed contour (either hand-drawn or computed using an automatic boundary-detection algorithm) is used to outline the object of interest separately on each slice and contour points from adjacent slices are then joined to form triangular tiles. Although a number of different methods have been proposed for determining the connections between contour points, it has been shown that optimal results- in the form of a minimum area surface- can be obtained when this process is recast in terms of a graph-traversal

problem {FUCHS77}. If speed is more important than quality, heuristic methods can be used to approximate these optimal results {GANEPATHY82}.

Because these tiling methods allow a user to explicitly define surface contours using his own judgement based on the slice data, these methods are still popular in many applications where surface localization is a highly sensitive and subjective task. The extraordinary amount of user input required by tiling methods are probably also their greatest drawback, although other problems with this approach certainly exist. Many surfaces are not well-represented by the relatively large polygons produced from tiling, and inaccuracies in the surface approximation are exacerbated when the distance between successive contours is particularly large, the number of points in each contour particularly small, or the shapes of adjacent contours especially dissimilar. In addition, surfaces can appear to be skewed or warped if the points between two contours are not connected appropriately. Spline interpolation methods have been used to correct some of these problems by more closely approximating the underlying data both within and between slices, and by allowing a more accurate estimation of surface normals {SUNGUROFF78}. Difficulties also arise in the use of tiling methods to represent surfaces with multiple, complex bifurcations. In these cases, a single slice may contain a number of closed contours which each need to be joined to one or more contours on a neighboring slice and it is often difficult to determine, without a priori information about the three-dimensional structure of the object being imaged, how the various contours should be connected to represent the surface most accurately. Although some heuristic methods have been developed to handle bifurcations {CHRISTIANSEN78}, user intervention is often required to ensure that the branches have been constructed correctly {PIZER86}.

11.1.2 Variations on tiling methods

The problems associated with the conventional tiling approach stimulated research into alternate methods of displaying polygonal surfaces from volume data sets. One variation on these methods, which is particularly well-suited for modelling thin, highly complex twisting and branching structures, uses connected segments made of truncated cones as structural primitives {BARILLOT85}. Data points are manually selected from the volume in sequence along each structure to be modelled, and a radius value associated with each point is used to generate a circle in the plane defined by the sum of the vectors between the previous and next sample locations along the structure. Truncated cones are then constructed between adjacent samples by tiling between points defined on the circles around each sample using quadrilateral polygons. Sub-branches are created individually by selecting a data point along the axis of an existing structure as the start of a new segment.

A "volume" tiling method was also developed, which for the first time correctly handled objects containing bifurcations, and objects defined by multiple contours in a single slice plane, in an automatic way {BOISSONNAT88}. In this method, solid objects are represented by selected tetrahedra derived from the three-dimensional Delaunay triangulation of the contour points on adjacent slices through the data. Beginning with a set of contours defined on individual slices, the first step in this method is to ensure that each contour is completely contained in the two-dimensional Delaunay triangulation of its vertices, by splitting contour segments and adding extra points where necessary. After each of the 2D Delaunay triangulations has been defined, a set of three-dimensional Delaunay triangulations of the combined vertices of two adjacent contours is computed. External and unsolidly-connected tetrahedra are then deleted from the representation, and a correction step is applied where necessary to resolve inconsistencies.

11.1.3 Indirect, voxel-based methods

In addition to the tiling methods described above, a number of voxel-based methods have been introduced which compute and display surfaces from volume data. Most of these methods assume, for algorithmic simplicity, that all data points in the volume are equidistant. If the data has been sampled at unequal rates in the different dimensions (for example, if the distance between data points within a slice is less than the distance between data points in adjacent slices), then some kind of interpolation is generally used to obtain the necessary intermediate values. In most grey-scale data sets, the value at a voxel represents a point sample of a continuous three-dimensional distribution, which we will refer to as "intensity" (not to be confused with pixel intensity in a 2D rendered image). The quality of the rendered images produced by the various voxel-based algorithms will depend to a great extent on the methods they use to reconstruct this underlying distribution.

In the cuberille method {HERMAN79}, the volume occupied by each voxel is modelled as a cube of constant intensity consisting of six polygonal faces, and object surfaces are constructed from appropriately selected connected subsets of these cube faces. Despite the large number of primitives, images can be displayed fairly quickly once the set of cube faces comprising the surface has been found because the shading and hidden surface computations can be optimized to take advantage of the regularity of the polygonal representation. The quality of an image produced by this method will be highly dependent on the accuracy of both the surface localization and the estimation of surface normals, and progress has been made in both of these areas since the method was first introduced.

A variety of algorithms have been developed for choosing the desired set of cube faces. In one approach, closed surfaces are found using a sequential search from an initial boundary element; subsequent boundary elements are selected from the neighbors of the current element that satisfy certain

criteria based on voxel density or gradient magnitude (encoded in a continuously-updated global decision variable), and backtracking is employed where necessary to reduce the likelihood of error propagation [LIU77]. An alternate approach, which requires that the data first be transformed by thresholding or some other segmentation process into a binary-valued volume, uses a sequential search of a different kind to extract closed surfaces. After the set of cube faces that separates the two voxel types has been isolated, a directed graph can be constructed from this set by taking each cube face as a node and each edge as an arc, with at most two incoming and two outgoing arcs being attached to each node. Beginning from an initial face element, a connected surface can then be defined as the subset of cube faces visited during a traversal of this graph [ARTZY81].

Because the cuberille method relies on an intermediate representation in which polygons may only appear in one of three possible orientations, the images produced by this method will exhibit sharp discontinuities in surface brightness if shading is based solely on the polygonal approximation. Attempts to mitigate the effects of these blocky artifacts by lowpass filtering the final image, anti-aliasing using subpixel sampling, or averaging surface normals from adjacent polygons [CHEN85] have met with little success.

Better results have recently been achieved with a variation of the cuberille method [UDUPA90] in which the cube faces comprising the surface are not restricted to lie exactly on voxel boundaries and the cube normals used in shading the surface are derived from the underlying voxel data alone, ignoring the cuberille approximation.

While the cuberille methods typically use linear interpolation to produce cubic voxels in the grey-scale data before segmentation, an alternate method performs the segmentation first and then interpolates. Beginning with contours defined on slices through the volume, this approach uses "dynamic elastic interpolation" to generate contour curves on the required number of intermediate slices, taking higher-order continuity into consideration and incorporating global information into the interpolation process [LIN88, CHEN90]. A solid model of the object is created by first filling in each contour and then stacking all of the contours together; the result is a binary voxel representation that can be displayed using any number of rendering methods. An advantage of this approach is its ability to create smoothly-varying surfaces between widely separated, disparate or branching contours, once the correspondences between the contours have been established.

11.1.4 Indirect, subvoxel-based methods

One of the most popular voxel-based methods is the "marching cubes" algorithm [LORENSEN87], which uses subvoxel-sized triangular elements to represent isovalue surfaces in a volume. Unlike other methods, in which each data point, or voxel, is represented as a small cube, the basic unit of operation in this algorithm is the cubic volume defined between eight neighboring data

samples. Each such cube in the data set is examined, either sequentially or in parallel, and if the desired isovalue surface is found to be contained within that volume, a triangular tessellation of the enclosed portion of the surface is derived. As a first step, the voxel intensity (or some precomputed object inclusion likelihood value) at each of the eight cube vertices is compared with a threshold value to determine whether that voxel lies inside or outside of the desired surface. The cube is then assigned one of 256 possible index values based on the configuration of its vertices. These 256 cube configurations can be mapped by rotation into 15 topologically distinct arrangements, and a triangular tessellation is defined for each of these canonical cases. Vertex coordinates are computed for each triangle in the tessellation by linearly interpolating between the endpoints of the enclosing cube edge, according to the difference between the threshold value and the values at each of the endpoints. To achieve the best possible results in shading, normals at the triangle vertices are computed not from the surface's polygonal approximation but rather from the intensity gradients of the underlying voxel data; gradient values found at each of the eight cube vertices, using central differences, are linearly interpolated along cube edges to the triangle vertex locations.

To maximize rendering efficiency for data sets in which the number of triangles is large and the screen area covered by each triangle is small, a variation of the marching-cubes algorithm has been developed which uses point primitives instead of triangles in the surface representation [CLINE88]. In this "dividing cubes" approach, the cube formed between eight neighboring voxels is subdivided until the area projected by any subcube face fits within a single pixel on the screen. Voxel intensity values are interpolated to the subcube vertex locations, and subcubes are then classified as interior, exterior or intersecting the isovalue surface depending on the intensity values at their vertices. If a subcube is identified as intersecting the surface, a point primitive is defined at its center and a gradient value is interpolated to that point from the gradients at the vertices of the enclosing undivided cube.

An extension of this marching-cubes approach has also been developed to represent isovalue surfaces in arbitrarily-shaped elements using bi-cubic patches [GALLAGHER89].

The marching cubes algorithm has a number of features that recommend it. Because subvoxel elements of arbitrary orientation are used in the polygonal representation, surfaces produced by this method generally fit the underlying data more closely than surfaces obtained through either a cuberille or tiling approach, and virtual memory requirements are kept to a minimum since the algorithm operates on only two slices of data at a time. A marching-cubes surface will usually take longer to render than a cuberille or tiled surface, however, due to the larger number of polygons and also to the loss of the regularity in polygonal orientation that allowed simplified shading and hidden surface computations to be used in the rendering of cuberille data.

The greatest drawback of the marching cubes approach is that surfaces in a volume are generally located on the basis of thresholding alone, which in many cases is not a sufficient criterion for

boundary detection and does not guarantee connectivity. Because surfaces are generated in response to all threshold crossings, regardless of context, isolated voxels are free to independently spawn spurious surfaces in an image when the threshold value is set too low and desired surfaces may become fragmented when the threshold value is set too high. These problems can be alleviated somewhat by applying a more sophisticated surface detection algorithm to the data, and using these surface likelihood values rather than the original voxel values in the marching cubes algorithm. An additional problem is that ambiguities exist in the marching cubes tessellation of the surface within cubes in which the threshold value is exceeded only at diagonally opposing vertices across a face. If not handled correctly, these ambiguities can cause tiny holes to appear in the surface where none should exist {WILHELMS90}.

It has recently been shown that the efficiency of a marching cubes method can be substantially increased if an octree-traversal of the data is used instead of a sequential approach, allowing large areas of the volume that lie inside or outside of the isovalue surface to be bypassed in one step rather than on a cube-by-cube basis {WILHELMS90}.

Semitransparent polygons have also been used to model volume data, in some of the direct projection and splatting algorithms discussed in section 11.5. It should be noted, however, that due to the need for compositing, the time required to render a set of semitransparent polygons will always be greater than that required to render an opaque polygonal data set of similar size.

11.2 Direct methods for the display of surfaces in binary data

In the mid-1980's a number of new methods for visualizing volume data sets were proposed, which did not require the extraction of surfaces as a first step.

11.2.1 Octree methods

One of the earliest direct methods uses octree encoding {MEAGHER82} to model objects defined by a binary-valued volume obtained from the grey-scale data through thresholding or some other segmentation procedure. Octree methods achieve data compression by storing voxel information in a hierarchical tree structure, which is built in a top-down fashion by recursively subdividing inhomogeneous regions of the volume into eight subregions until each terminal node of the tree corresponds to a region of the volume in which all voxels share the same value. The original goal of the octree approach was to allow more rapid manipulation and display of complex three-dimensional objects than was currently possible using polygonal models. Efficient integer algorithms were developed to implement Boolean operations for cut-away views, and to perform geometric

transformations such as rotation, translation and scaling on octree-encoded data; hidden surface views of octree-encoded objects can also be generated quickly for any observer position, due to the spatially pre-sorted format in which the data is stored. In addition, the hierarchical octree structure is useful in successive refinement for generating images at varying resolutions using data from intermediate levels in the tree. The primary disadvantages of octree encoding binary data are the time it takes to build the tree and the complexity of the tree-traversal operations. In addition, the octree representation of a data set is highly dependent on the initial position and orientation of the data; two data sets that differ only by a small translation or rotation may have substantially different octree representations.

Images can be generated from an octree data set by establishing a correspondence between pixels in the image and face-connected strings of voxels in the volume, then recursively traversing the octree in a back-to-front order and whenever a non-empty voxel is encountered overwriting the corresponding pixel value with the color of the current voxel {DOCTOR81}. Pseudo-shading can be implemented by defining top, bottom, left and right illumination factors which are added to a voxel's color when its neighbor in the corresponding direction is empty. Front-to-back octree display algorithms have also been developed for octree data sets {MEAGHER80}.

11.2.2 Object-space methods for the display of surfaces in binary data

Algorithms which create images by projecting volume information onto the image plane are commonly referred to as "object-space" methods. An important feature of these algorithms is that they do not require that the complete data set be resident in memory at all times. One of the earliest object-space approaches renders images from binary-valued voxel data sets by traversing slices of the arbitrarily-oriented volume from back to front, beginning within each slice at the most distant corner and progressing along either rows or columns, projecting a value derived from the depth of each voxel perpendicularly onto an individual pixel of the final image {FRIEDER85}. This method, although straightforward, is not optimal for a variety of reasons. The resulting images also suffer from the same shading problems that were encountered in the cuberille approach, and because a given pixel may be colored and re-colored many times before a final value is assigned, efficiency is sacrificed. In addition, this method is limited to performing parallel planar projections; a perspective view can be achieved only by pre-warping the volume data before rendering is begun. Moreover, due to the poor quality of the reconstruction algorithm, small hole artifacts can appear in the rendered image when the volume is viewed at certain angles or magnifications.

If voxel data is projected onto the image plane in order of increasing distance from the viewer, only the first value received by each pixel needs to be retained and all subsequent attempts to write to the same pixel must be disallowed. A front-to-back version of the above method which produces equivalent images in less time avoids pixel re-writes through the use of a dynamic data structure that

keeps track of unwritten scanline segments {REYNOLDS87}. The algorithm first ensures that face-connected voxels in a single row of the volume will project onto adjacent pixels in a single scanline of the two-dimensional image by requiring that all rotations be done in a specific sequence. As each row of voxels is projected onto the image plane, the endpoints of the newly generated spans can then be compared with the endpoints of the unwritten portions of the scanline, which eliminates the need to test for overwrites at each individual pixel.

11.2.3 Image-space methods for the display of surfaces in binary data

In addition to the object-space methods described above, image-space methods - which render the data on a pixel-by-pixel basis - have also been developed to display solid objects defined by binary-valued voxel data sets. The simplest of these methods {TUY82} steps through the volume along rays cast perpendicularly from the center of each pixel until the first non-empty voxel is encountered, shading pixels of the final image with a value based primarily on their distance from this surface voxel. When the viewing direction is not aligned with an axis of the volume, nearest-neighbor interpolation is used to compute the values encountered at discrete intervals along the ray.

One advantage of the ray-casting approach is that it does not require a complete traversal of the volume; processing can be stopped as soon as a surface voxel is encountered along the ray from each pixel. A disadvantage of this approach is that in order to generate images at arbitrary orientations the entire data set must generally be accessible in memory at all times. Techniques have been developed which reduce the time required to render a series of images by exploiting the coherence between consecutive frames of a rotation sequence {RHODES87}.

11.3 Direct, binary methods for the display of surfaces in grey-scale data

While most early methods were designed to operate on binary-valued voxel data sets, in which surfaces have been pre-defined using some type of segmentation procedure, more recent methods have moved towards integrating segmentation into the rendering process. The motivation for doing this is twofold. First of all, retaining the grey-scale information at each voxel allows the use of shading models which more accurately reflect the orientation of underlying object surfaces. Because the values found at each voxel typically represent point samples of a smoothly-varying function in three dimensions, shading methods that base their estimate of surface orientation on a 0th-order reconstruction in which the volume is represented by a set of cubes of uniform intensity are clearly not optimal. A much better approximation to the surface orientation at a boundary voxel is given by the gradient of grey-scale values across the voxels in a 3x3 neighborhood {HOEHNE86}. In addition,

surface position can be estimated with greater precision when the possibilities are not limited to specific locations on a regular grid.

An investigation of shading algorithms for grey-scale data has shown that of the methods currently in use, a technique called "adaptive grey-level gradient shading" {POMMERT89} produces the smoothest images with the least amount of error in surface normal estimation. This method operates by adaptively selecting the neighborhood over which gradients are computed to more accurately represent surface normals on very thin objects. For example, the partial derivative in the x-direction at a voxel (i,j,k) is normally computed as $\rho(i+1,j,k) - \rho(i-1,j,k)$. In this method, if $\rho(i,j,k) > \rho(i+1,j,k)$ and $\rho(i,j,k) > \rho(i-1,j,k)$ then the voxel is interpreted as being part of a very thin surface and the partial derivative is computed as $\rho(i,j,k) - \max\{\rho(i+1,j,k), \rho(i-1,j,k)\}$.

One method which produces high-quality images of surfaces detected by thresholding in a grey-scale volume uses grey-scale gradients to approximate surface normals and linear interpolation to resample the volume data {HOEHNE87}. Rays are projected perpendicularly from the image plane, which is kept aligned with the data volume, until a voxel is encountered whose value exceeds the specified threshold. Rotated or perspective views are generated by a preprocess in which the entire volume is transformed and then resampled before any rays are cast. Because grey-scale information is retained at each voxel, it is possible to generate hybrid images in which raw voxel values are displayed on selected cutplanes in combination with a surface representation of the data. In this case, voxel values are sampled along each ray until the cutplane is encountered, and if the grey-scale value at the voxel intersected by the cutplane exceeds the surface threshold, this grey-scale value is assigned to the pixel of the final image. Otherwise, sampling continues along the ray until a surface voxel is hit and the image pixel is assigned a color derived from the distance and orientation of the surface at that point.

An extension of this method renders surfaces detected by arbitrarily complex segmentation procedures through the use of a data structure called the "generalized voxel model" {HOEHNE88}, in which an array of values is stored at each data location in the volume. This technique can also be used to generate combined images using information from multiple, registered data volumes.

One approach in which more elaborate surface detection criteria are used is called "active-ray tracing" {TROUSSET87}, an ad hoc technique that uses local information about voxel intensity and gradient magnitude to locate boundaries in a grey-scale volume. Rays are cast through the data in a direction orthogonal to the slice axis (thus ensuring that each ray will be completely contained within a single slice), and as each ray penetrates the volume, the intensity values of the voxels it passes through are compared with predetermined upper and lower threshold bounds. A connected subset of voxels along the ray whose intensity values fall within the specified range is classified as candidate boundary segment. If the potential boundary indicated by this segment is determined not to be part of a surface already detected by previous rays, it is judged to represent either the start of a new surface or a

piece of noise, depending on an estimate of the area of the potential surface. Once a candidate segment has been validated, the location of the surface within that segment is chosen. If the gradient magnitude is not sufficiently large, either at an individual voxel or across the connected set of voxels, gradient information in the candidate segment is regarded as unreliable and the boundary is simply placed at the first voxel whose intensity falls within the threshold range. When the gradient information is judged to be reliable, however, then for a new surface the boundary is placed at the voxel of largest gradient magnitude in the candidate segment, or for an existing surface at the voxel of largest gradient magnitude whose gradient direction is compatible with the direction of the surface.

11.4 Direct, binary methods for the display of intensity distributions in grey-scale data

Most of the methods presented so far are designed to display surfaces in grey-scale data sets. Although the majority of algorithms that attempt to portray volume intensity distribution information are nonbinary and will be discussed in the next section, a direct binary method has been developed which uses color to represent different intensity ranges within a volume. Before three-dimensional surface or volume rendering methods came into widespread use, volume data was typically viewed on a slice-by-slice basis. A display method that builds upon this slice-based approach forms images using stacks of two-dimensional slices [FARREL84]. Data is extracted from the volume along a user-defined axis to produce the set of slices, and on each slice data points are assigned one of several possible colors based on the density range into which they fall. The slices are then projected onto the screen, one at a time from back to front, with each slice slightly offset from its predecessor to produce an oblique view. Only those points falling within the desired density ranges are displayed on each slice, to prevent the unwanted obscuration of data from preceding slices.

11.5 Direct, nonbinary methods for the display of grey-scale data

11.5.1 Multi-planar reconstruction

Multiplanar reconstruction is a two-dimensional method that has been used to visualize grey-scale volume data. This method allows structural details that originally spanned multiple two-dimensional slices along a given axis to be portrayed in a single slice image, through a process in which the volume is reconstructed from the original slices and a second set of two-dimensional slices is then extracted along a different, orthogonal axis [GLENN75].

11.5.2 Reprojection methods

One of the earliest direct, nonbinary methods used to visualize grey-scale volume data is a technique called reprojection [HARRIS79]. In this method, planar orthogonal views are generated by projecting the voxel data along parallel paths through the volume into an image buffer, summing the values of the voxels encountered along each path into individual pixels of the image. A stereo-pair can be produced when two such images are generated using appropriate parameters for the position and angle of view, but there is no perspective distortion and the image will be equally in focus at all distances into the volume. Obscuring structures can be partially removed in a reprojected image by selectively decreasing the relative contributions of certain voxels, distinguished either by their spatial location or by their intensity value, during the summation process; obscuration can be also avoided to a certain extent through computing the reprojection from a variety of different angles. While reprojection allows more of the volume data to be seen than methods which only display surfaces, a significant drawback to this approach is that all depth information is lost during the summation process making the images look uniformly flat. Figure 1 shows a stereo-pair of reprojected images of metaphase chromosomes in a human lymphocyte.

A similar method assigns each pixel of the image a grey-scale value equal to the maximum intensity encountered along the ray emanating from that pixel [VANDERVOORT85]. Because of its simplicity, this method is frequently used with success for many applications. Images produced with this approach lack depth information, however, and discontinuities may arise when a voxel which contains the maximum value along a ray from one direction is no longer found to be a maximum when the rays are cast from a slightly different direction. Figure 2 shows a pair of images generated using the maximum intensity method.

11.5.3 Image-space methods

Methods have also been developed which display surfaces in grey-scale volumes without requiring a binary decision to be made about the exact surface location. These methods are based on a recognition of the "partial volume effect", in which a mixture of materials may be contained within the volume covered by a single voxel whose value represents a weighted average of the intensities of these combined materials.

One of the earliest such approaches traced perspective rays from an eye point through the pixels of the image plane into the volume data, computing a color, transparency and filter value for each voxel encountered using a look-up table indexed by voxel intensity [SCHLUSSELBERG86]. The filter value was used to define shading characteristics and to implement masking, and transparency

calculations were accumulated along a ray until the first completely opaque voxel was encountered. Although the approach was promising, image quality was compromised by the use of a non-optimal nearest-neighbor interpolation technique and by undersampling in the rear portions of the volume due to ray divergence.

Very high-quality images were produced several years later by two similar, independently-developed rendering algorithms. The first of these methods [LEVOY88a] uses a multi-step approach to display smooth-looking surfaces from grey-scale, voxel-based data. If the data in the original volume is not equally spaced in all three dimensions, the volume is first resampled, using B-spline interpolation to obtain the necessary data resolution. A "surface normal" is then computed for each voxel in the volume, based on the grey-scale gradients at that point. Classification ramps are set up which define opacity as a piecewise linear function of both voxel intensity and gradient magnitude (the latter is used as a surface likelihood indicator), and the results are stored in look-up tables which are used in later steps to assign an opacity value to each voxel in the volume. A color value is also computed for each voxel using a Phong shading model based on the estimated surface normal, surface distance, and light position. Finally, parallel rays are cast from each pixel in the image through the computed color and opacity volumes, and trilinear interpolation is used to sample these values at equal intervals along each ray. The color value is pre-multiplied by the opacity at each sample, and both color and opacity are composited from front to back along each ray until either the opacity has accumulated beyond a specified cutoff value or the data has been exhausted. Finally, the color accumulated along each ray is assigned to the corresponding pixel of the image. Although a back-to-front compositing order was first proposed, this front-to-back approach achieves greater efficiency by allowing the adaptive termination of rays [LEVOY90b]. The stereo images in Figure 3 were produced using the Levoy algorithm.

Isovalue surfaces in grey-scale volume data can also be rendered using a variation of this direct, nonbinary approach [LEVOY88b]. While a binary approach would merely assign an opacity α_v to voxels having the selected intensity value ρ_v , this method creates a smoother surface by also assigning voxels whose values are close to ρ_v opacities that are close to α_v . In order to avoid holes while ensuring that the surface does not become too wide, opacity values are defined to fall off with the difference from the target intensity at a rate inversely proportional to the magnitude of the gradient at the surface.

Geometric primitives such as polygons can be integrated into the final images produced by either of these methods through the use of a hybrid ray-casting algorithm [LEVOY90a].

An extension of this ray-casting method has recently been developed which allows the production of high-quality perspective images [NOVINS90]. Because perspective is most useful as a depth cue only when used on data containing recognizable geometry, it was originally not implemented in many algorithms designed for the visualization of medical or biological data. With the

introduction of geometric cut-aways as a technique for viewing raw data in the interiors of objects and with the increasing speed of many displays which encourages interactive fly-throughs of complex, three-dimensional objects, the need for perspective rendering has increased. In order for a ray-casting method to correctly handle perspective, care must be taken to ensure adequate sampling in all parts of the volume. The approach taken by this method is to split each ray into four divergent sub-rays whenever the sampling rate falls below a specified threshold and then to combine the accumulated results from each sub-ray to yield a composite value for the final pixel.

The high quality of the images produced by the methods of Levoy and Drebin et al. (discussed below) is primarily due to the effort they take to avoid any unnecessary introduction of aliasing or quantization artifacts. One example is the preservation of continuity in the underlying volume data through the use of cubic or linear interpolation for resampling rather than a nearest-neighbor method. In addition, by avoiding binary classification these algorithms allow a representation of the uncertainty of data segmentation results. In the nonbinary approach, voxels in the vicinity of a surface boundary will have an opacity proportional to the likelihood of their actually being a part of this surface. One drawback with these algorithms, which applies slightly more to the first method than to the second, is the difficulty of defining appropriate classification and shading functions. This can be a non-intuitive process, requiring several iterations to "get right", and no one has yet been able to objectively measure the adequacy of the parameters selected.

A different nonbinary ray-casting approach is used in a technique referred to as "simulated fluorescence" {VANDERVOORT89}, a two-step algorithm which models the excitation and emission phases of a simplified fluorescence process. The first step in this method is to compute the amount of "exciting radiation" that is absorbed by each voxel in the data set. A set of parallel rays is cast through the data from the excitation direction and the intensity value at each voxel encountered along a ray is multiplied by the radiation intensity at that point. Radiation is attenuated at each step along the ray by an amount derived from the intensity of the voxel through which it has passed. The second step in the method is to compute the amount of "emitted radiation" reaching each pixel of the final image. A second set of parallel rays is cast through the data from the viewing direction and the excitation values are composited from back to front, using voxel intensity to regulate the attenuation. This method produces three-dimensional-looking images without making any attempt to extract surface information, and the incorporation of shadows into the shading process heightens the perception of depth.

A similar approach, but one that uses gradients of voxel intensity for shading, is described by the Heidelberg Ray Tracing Model {MEINZER91}.

11.5.4 Object-space methods

At the same time that Levoy's image-space method was being developed, a high-quality object-space approach was also proposed {DREBIN88} that produces very similar results. In this approach, which also requires that the voxel data be evenly spaced in all three dimensions, the image plane is kept aligned with the data grid and rotated or perspective views are generated by resampling the volume. Rotating by resampling the entire data set will in general be more expensive than interpolating values as needed along each ray, since most rays will in general be traced only partway through the volume. However, maintaining alignment between the image plane and the volume allows processing to be confined to a well-defined subset of the data, substantially reducing virtual memory requirements. The first step in generating images with this method is to classify voxels into material types based on their intensity values. Separate volumes are created for each material type, in which the value stored at each voxel represents the percentage of material present in the voxel. This classification is non-binary, and reflects the fact that a single voxel may contain a mixture of materials. Color, opacity and "density characteristic" values are associated with each material, with the gradient magnitude of the density characteristic used as a measure of surface strength. A shaded volume is then computed using these parameters, and the resulting voxel values are composited from back to front to produce the final image.

To greatly reduce the rendering time required by this method, the surface detection and shading steps can be omitted to produce reprojected ("unshaded") images {NEY90}. In both approaches, the rotation/resampling step is usually performed just prior to compositing, so that multiple views of the data can be generated from a single intermediate volume.

A different kind of projection algorithm was designed to produce high-quality images from low-resolution data sets, exploiting coherence in the data to increase the speed of rendering {UPSON88}. In this algorithm, the basic unit of operation is the cell defined between eight corner voxels, and cells are processed in a front-to-back order with the results composited at each pixel to form the final image. Color and opacity values are first assigned to each voxel by a shading procedure, and these values are assumed to vary as a cubic function within each cell. A bounding box is computed for each cell that defines the area of the screen onto which the cell contents will project, and each scanline segment in this bounding box is subdivided into spans within which the depth of the projected cell will vary linearly. For each pixel in a span, the color and opacity functions are numerically integrated in depth through the projecting cell, along the lines defined by the four corners of the pixel. These four values are then averaged to yield the cell's color and opacity contributions to that pixel.

A related algorithm {SHIRLEY90} projects tetrahedral cells in a volume, producing a set of semitransparent triangles that can be displayed relatively quickly on a general-purpose graphics workstation. The projection of each tetrahedron is decomposed into between 1-4 triangles, depending on the orientation of the cell relative to the image plane, and color and opacity values at each triangle vertex are determined based on the thickness of the projecting tetrahedron along the line of sight

through the vertex and on the data values at the entry and exit points of this ray. Processing is fairly quick, because for each set of projected triangles there will be exactly one (possibly shared) vertex corresponding to a nonzero tetrahedral thickness. An additional feature of this algorithm is that it gracefully handles volumes containing data samples that are not equally spaced on a rectangular grid.

A very similar algorithm {WILHELMS91} operates on volume data that is regularly sampled in each dimension, and achieves greater speed by exploiting the fact that each rectangular cell projects onto the image plane in exactly the same way. This work also includes a discussion of the trade-offs between image generation time and image quality when different methods are used for interpolation between projected vertices and for integration in depth through a cell. These faster projection methods do not incorporate directional shading, and they make no attempt to display surface information, resulting in images with a rather fuzzy, cloud-like appearance.

A different object-space technique, called "splatting" {WESTOVER90}, reconstructs the intensity distribution in the volume occupied by each voxel and projects this reconstructed data onto an area defined as the image plane "footprint" of the voxel. A Gaussian is usually used for the reconstruction, although other functions might also be appropriate. If a parallel projection is used, the footprint function will be identical for all voxels in the data set and can be precomputed and stored in a look-up table. Each entry in the footprint table will represent the weight of a voxel's contribution to the sub-pixel area covered by that part of the footprint, and the size of the footprint table can be varied to trade off image quality for rendering speed. If the data is equally distributed in all three dimensions, the footprint will be circular, but unequal spacing can be easily handled by stretching the footprint into an ellipse. Once each voxel is assigned a color and opacity according to some shading algorithm, these values are projected using the footprint function into an accumulation buffer for each slice. After all voxels in a slice have been projected, the values in the accumulation buffer are composited into the final image. Processing can proceed in either front-to-back or back-to-front order, and the intermediate results can be displayed interactively, making it possible to view the image as it is being constructed.

A very similar method, which operates on a hierarchical representation of the volume, uses semitransparent polygons to approximate the footprint functions {LAUR91}. The resulting images are not of as high quality, but they can be displayed rapidly on a fast polygon rendering workstation.

The choice of whether to use an object-space or image-space method will depend on a number of factors. Object-space methods are typically concerned with how a single data sample maps onto multiple pixels in the image, while image-space methods look at how a single pixel of the image is affected by multiple data samples in the volume. Thus an image-space method such as ray-casting could process each pixel in parallel, although it would be difficult to partition the volume between multiple processors since the rays typically require random access to the data. With object-space

algorithms it is easy to partition the data, but bottlenecks can occur in accessing the image and care must be taken to maintain correct ordering when compositing.

The desire to enhance the interactivity of these high-quality methods has also led to the increasing use of a technique called "progressive refinement", in which coarser images are displayed at more rapid update rates while the data is being oriented and higher resolution images are generated in a longer amount of time once the motion has stopped. Ray-casting methods implement progressive refinement by sending fewer rays through the data and interpolating from these results to fill in the remaining pixels. Splatting methods implement progressive refinement by using increasingly smaller reconstruction kernels, which map into smaller footprint functions.

11.5.5 Radiation transport methods

A physically-based approach to rendering images from volume data sets applies concepts from transport theory to the problem of modelling how light interacts with clouds of particles.

The first algorithm to use such an approach {BLINN82} worked with data that is modelled as a distribution of spherical reflecting particles, too small to be individually distinguished, which are randomly positioned in a layer. Images are generated by computing the amount of light that is reflected from the layer in a given direction E after entering from a direction L and hitting one or more particles inside the layer, and also the amount of light that is transmitted through the layer in the direction $-E$. A number of simplifying assumptions, such as uniform particle density, low particle reflectivity and single scattering, are used in this algorithm to make the problem computationally tractable.

This method was later extended to volumes of particles and implemented as a ray-tracing algorithm, with the restriction to media of low albedo removed and higher-order scattering effects included in the computations, to which approximate solutions must be found {KAJIYA84}.

A subsequent approach achieves greater computational efficiency by assuming a model of varying density emitters, eliminating the shadowing and secondary scattering effects {SABELLA88}. This method was applied to the visualization of density distributions in three-dimensional solids, with color used to highlight density extrema in the data set by defining hue according to the maximum value encountered along a ray, and with saturation tied to distance for use as a depth cue.

A different formulation of this method operates by probing a data set with a beam of particles, tracing the piecewise linear transfer of these particles through the volume and generating images from the resulting patterns of absorption and scattering derived using equations from linear transport theory {KRUEGER90}. This approach has the advantage of providing a mathematically rigorous and physically meaningful framework for the definition of visualization parameters, and it has been

shown that many earlier volume rendering techniques can be understood as specific mappings or approximations of this method.

11.6 Object definition and segmentation

In nearly all of the rendering methods discussed so far, some kind of object segmentation is used to emphasize certain parts of the data set- most often in order to display object surfaces. Thus, in many cases, what is shown in a rendering represents a specific model of the data rather than simply the data itself. Conclusions drawn from the rendered image must carry with them a justification for the specific model used. Recognizing this, we can develop a variety of complex models to demonstrate relationships that could not be shown using only a direct or simple mapping between voxel intensity and pixel color.

For example, if we wish to evaluate the connectivity of the vasculature of a renal glomerulus, we find that the capillary lumens cannot be seen clearly through the surrounding mesangium and connective tissue. One solution, shown in Figure 4, is to remove a portion of the volume to expose local features in the interior. In another data set, we find that it is impossible to use any simple combination of parameters for surface classification to clearly illustrate the relationship of nucleoli and chromocenters to the nuclear membrane in a human lymphocyte. Because the voxels containing nuclear membrane are of equal or higher intensity than the voxels containing nucleoli and chromocenters, any straightforward rendering would leave the nucleoli obscured by the nuclear rim. One way to visualize both of these structures in a single image, as shown in Figure 5, is to remove a portion of the volume with cutplanes and display the voxel intensity information on the cut surfaces. Alternately, as shown in Figures 6 and 7, each structure could be extracted individually from the data volume and then rendered using a different set of parameters to produce a composite image. Such renderings cannot be considered "correct" in the sense of some simple relationship to the original signal, but they can at times be useful in demonstrating certain biologic points. The usefulness of this type of rendering is limited, of course, by the accuracy of the segmentation used to extract the portions of the data to be displayed.

In order to provide the three-dimensional information shown in images like these we must identify, locate, and extract regions of interest (ROI) from the volume data. Once the ROI is isolated, the results can be used for boundary determination and rendering. Alternatively the ROI mask itself can be rendered, as in Figure 8, or manual determinations of surfaces from the spatial groupings can be done.

The problem of ROI segmentation is related to that of surface segmentation but is nonetheless different. In surface segmentation we are looking for some local measure of "boundariness" that can be used to locate surfaces in the volume. In contrast, ROI segmentation depends more on the object than on its surface, so connectivity and other global aspects need to be considered. ROI segmentation is

important because it is not always the case that all objects are bounded by a well-defined surface; often the boundariness of any given point in space is best described probabilistically. Surface determination is a very different thing than object classification, as a full specification of boundaries may not give a full specification of objects, and vice versa.

11.6.1 Primitives

All segmentation strategies rely first on the measure and organization of local geometric primitives. The choice of which local features to exploit depends upon the properties of the particular data set being segmented. For instance, if we are working with a volume in which the ROI is clearly delineated by sharp edges, methods that measure edge strength might be most appropriate. In cases where edges are weak, noisy, or in a complex surround, some other feature, such as a measure of local homogeneity of object intensity or texture, might form the basis of the segmentation scheme. In the data set used to generate Figures 5-7, simple edge-finding techniques might do a quick and easy job of finding nucleoli but fail at isolating chromatin clumps, while variable conductance diffusion (discussed later) might be able to isolate the chromatin clumps nicely.

Ideally, primitives should be invariant under image rotation, translation, scaling and illumination. For example, surface curvature measures do not change if one changes the viewing angle or coordinate system representation of a surface. Thus, one might segment nuclei from an image covering numerous cells by detecting their ovoidal shape.

The most basic primitive is simple intensity. While it can be used directly, as with simple thresholding or template matching, it is not invariant to illumination and for template matching not invariant to rotation. Therefore, most often other measures are derived from it.

Edge strength as a feature has been intensively studied, both because of the importance of edges for delineating regions and because of the biologic importance of edge strength in human vision {LEVINE85}. The measure can be as simple as the magnitude of the intensity gradient or it may depend on intensity gradient directions or on properties of contours of iso-intensity or iso-edge strength {FISCHLER86, ASADA86, PIZER88}. The evaluation of curvature has also been extensively studied, and Koenderink has demonstrated a biologic basis for such measures {KOENDERINK87}. Techniques based on differences of Gaussians or on maxima of gradient magnitude along gradient directions (the Canny method) also fall into this class {CANNY86}. For both edge strength and curvature, the incorporation of scale (resolution) by Gaussian blurring allows the measurement to match the information in the data by computing only at supportable locational tolerances.

Other geometric features, such as intensity extrema and local shape features of intensity surfaces, have also been investigated and integrated into segmentation algorithms. An example of this is watershed region segmentation. The concept is most obvious in 2D, although extensions to 3D have

also been implemented. In this approach, intensity is used as a topographic third dimension in a 2D image, where bright spots are peaks and dark spots are depressions. This topographic analogy can be extended to define watershed areas, analogous to drainage regions in geographic topography, to produce an image description that connects sets of intensity maxima (hilltops), minima (pits), ridges and valleys. This connectivity can then be used to define image regions and relationships between image regions {VINCENT91}.

Another common primitive is the medial axis, which provides a "skeleton" of the object {NACKMAN82, BLUM78}. Originally defined for presegmented objects, medial axes have recently been computed in multiscale, for greyscale images {FRITCH91}. Medial axis methods segment an object based upon the branching pattern of its skeleton, which can be organized in a manner invariant over translation, rotation and size. These methods are presently most useful when applied to data sets in which simple region definition has already been accomplished and one wants to identify the subset of regions that forms an object ROI.

11.6.2 Strategies and Representations

Given a set of primitives and a focus on either surface or object geometry, one must then develop a strategy for organizing these primitives. The most common strategies roughly fall into one of two categories: those which perform some sort of spatial characterization and those which create a model of the object or image and perform an optimization based on the parameters of that model.

In parameter optimization methods, the image intensities or object properties within an object class form a set of parameters to be determined. In the Gibbs/Markov Random Field (MRF) methods, the parameter value is intensity and the model (*a priori*) properties are given by the probability of the intensity of any given voxel, given the intensities of voxels in some local neighborhood. The *a posteriori* probability also reflects the way the computed segmentation fits the image data. These methods attempt to find the segmentation of the data that gives the maximum *a posteriori* estimate of the unknown field {GEMAN84, HASLETT85, QIAN89, COHEN91}. MRF methods work best in areas where the regions of interest are compact and the data is of limited complexity. They suffer from often having to resort to complicated optimization methods with a slow convergence.

In the active surfaces method, also based on optimization, an expression is defined which quantifies the "energy" of a surface relative to its geometric properties and to the way it fits image features, and the segmentation algorithm tries to find a minimum energy configuration {TERZOPOULOS91, PENTLAND91, TERZOPOULOS88}. Because they are often sensitive to initial parameter settings, active surfaces methods work best when the shape characteristics of the region of interest are known and when the region is fairly uncomplicated. There have been attempts to combine

MRF and active surface methods, using the probability estimates as one of the features in the energy expression {LIN91}.

A classic parameterized model is template matching via the Hough transform, in which feature matching is done in the parameter space itself rather than in image space {XU90, RISSE89}. The Hough transform is useful for finding images within a very limited range of shape, orientation, and size, since the parameterizations are usually not invariant over these transformations. The image is matched against a family of templates by letting each voxel vote for those templates with which it is consistent. Brakenhoff et al. have used template matching methods to locate chromosomes in confocal images {BRAKENHOFF88}.

Instead of parameter optimization, one can perform grouping based on spatial characteristics. A common strategy is to progressively blur the image and follow features through the resulting scale space {WITKIN83, CROWLEY84, WITKIN87}. Such multiscale methods work well with noisy images, since noise blurs away quickly, and with images containing compact regions without thin connectors. Image features tend to blend together and details tend to disappear when an image is blurred, and one can follow the pattern of how this occurs. A large number of features have been examined in scale space, including edges, medial axes, and watershed regions. Pizer and others have investigated the loss, or annihilation, of features while Coggins and others have followed the paths of features in scale space {LIFSHITZ90, LU89, DILL87, PIZER87, COGGINS90, LOWE88, KROPATSCHE87}. Pyramid methods are similar, but instead of just blurring they reduce spatial sampling at each blurring step {SAMET84}. A particularly interesting pyramid method is based on the so-called "wavelet" representation {MALLAT89, LEITNER91}. Pyramid methods have the advantage of speed, data compression, and support of parallel algorithms, but their rigid resolution reduction causes them to suffer from variance with translation and rotation. Pyramidal approaches that use image-sensitive irregular tessellations have been proposed to reduce the effect of image orientation {MONTANVERT91}.

If the region of interest is well demarcated from the surrounding image by some feature (such as an edge), one can modify the amount of blurring that occurs locally in order to take advantage of this feature. An example of such adaptive blurring is presented by Saint-Marc et al., who repeatedly blur very small regions by an amount weighted by image feature strength {SAINT-MARC91}. Similarly, variable conductance diffusion methods "diffuse" features into each other using the heat equation, with heat conductance modulated by the strength of change in some image feature {PERONA87, NORDSTROM89}. These methods are iterative and thus can be slow, and they may suffer from a dependence on initial parameters, but they can also be extremely effective in separating objects that are fully surrounded by a locus of adequate change in the feature in question.

In some cases, the difference between regions is based on texture rather than on some easily defined geometric primitive. Here, features may more easily be examined in the frequency domain than in the spatial domain. Examination of the frequency domain alone, however, carries with it the problem

of loss of spatial localization. Whitaker has recently attacked this problem by performing variable conductance blurring using locally derived frequency space features {WHITAKER91}. Wecshler argues convincingly that one should look to both the spatial and frequency domains when examining images, and he describes a variety of cojoint representations in the space-frequency plane that can be convolved with a kernel in much the same way as the spatial blurring described above, providing a texturally-based scale-space representation. Wecshler shows how operators such as the periodogram, Gabor, Laplacian zero-crossings and differences of Gaussians can be expressed as cojoint representations in space and frequency {WECSHLER90}.

Moment representations have been of great interest because they are invariant over size, orientation and position, and a moment tensor-based template matching method described by Cyganski and Orr has been used to match objects over a wide range of affine transformations in the data {CYGANSKI85, DIRILTEN77}. Hu derived a set of moment invariants for use in image segmentation, and these have been extended to 3D by Sadjadi and Hall, and recently modified by Reiss {HU62, SADJADI80, REISS91}. Moment representations are best used to recognize and choose among objects that have already been separated from the background image.

Finally, methods which are a hybrid of parameterized optimization and spatial characterization have also been tried. For instance, Bouman and Liu have recently proposed a multiresolution Gibbs approach in an attempt to avoid the overemphasis on local features common in classic MRF methods {BOUMAN91}.

11.6.3 Region Delineation

Given a collection of primitive regions, perhaps organized into a graph or hierarchy, or a set of primitive feature values, the tactics of delineating the ROI must be considered.

Thresholding is a method which simply selects as the ROI all voxels between two levels of feature strength. Simple thresholding is sensitive to noise and even to smooth spatial variation in feature strength {PARKER91, BRINK89}. However, it is very useful when used with features that are measured to avoid just this kind of variation, such as the output of variable conductance diffusion. Thresholding is also useful for quickly "coloring" a large number of voxels which can be then edited by hand. Montag et al. used this method with success in combination with morphologic operations (discussed below) for the segmentation of intranuclear chromatin {MONTAG90}. To produce surface descriptions for surface-based rendering, Brakenhoff et al. used threshold operators on edge-based intermediate images to fit shape models to look for surfaces {BRAKENHOFF88}. Schormann et al. proposed using local blurring followed by thresholding for nuclear segmentation in confocal images {SCHORMANN90, SCHORMANN88}.

Clustering algorithms attempt to segment a feature space into statistically significant groups. For instance, Baxter and Coggins cluster voxels according to their intensity paths in scale space, and Manjunath et al. evaluate the MRF parameters over non-overlapping small regions of an image and cluster regions based on those parameters {BAXTER90, MANJUNATH91}. Most clustering methods are sensitive to initial parameterization and can be iterative and slow {JOLION91}. They are most useful when there are a large number of disparate, relatively uncorrelated parameters being evaluated and when measures of similarity are well defined {XIE91, JAIN88, DUDA73}.

Merging algorithms take small regions and accrete them into larger regions based on measures of similarity {HONG84}. Splitting algorithms take large regions and break them into smaller regions based on measures of dissimilarity. These are often combined in an iterative manner which, the user hopes, will converge to some meaningful representation of regions. They are often used in conjunction with multiscale representations and graph-based methods {GELFAND91}. Split-and-merge algorithms work well only with fairly simple images and are most often used as a preprocessing step for other methods.

Graph-building methods try to organize primitive regions in a manner recognizing inter-region geometry, such as adjacency, containment, directional relations, and watershed relations {MORET82, CHOU91}. Regions can then be found by graph matching algorithms matching subgraphs to model graphs. These methods or those in which a decision tree is formed by clustering feature vectors form the basis for most artificial intelligence (rule-based or grammar-based) approaches. Graph-building methods can also be used to connect edges into closed contours which can be used to delineate regions. They are very rarely ends in themselves but most often form a scaffolding upon which further processing or interactive modification can be done.

Set operations form the basis for methods such as mathematical morphology, which can be used to select regions satisfying certain shape requirements or to edit regions according to their shape {JANG90, HARALICK87}. Mathematical morphology is applicable both to binary masks and to regions with intensity values. In these methods a kernel or structuring element (often a sphere or cube) is moved through an image, much like a convolution, but here the portion of the image covered by the structuring element is included in or deleted from a region. Mathematical morphologic methods are often used as a postprocessing step following application of some other method in order to clean up the data -- to impose connectivity, fill in gaps, etc. A number of common operations have been described, including opening (which tends to delete small objects, break apart thinly connected regions, smooth contours, and such), closing (opening the object complement), skeletonization, thinning, thickening, pruning, and operations to find the convex hull of an object. Variations in the structuring element, including making it a function which is applied to the "earth" under the topographic region intensity surface, allow sensitivity to particular shapes or orientations. The above-mentioned threshold method of Montag et al. applied one level of thinning in a way maintaining connectivity. Schorman et al. and

Forsgren also use thresholding in combination with mathematical morphologic operations for segmentation of confocal images {SCHORMANN90, SCHORMANN88, FORSGREN90}.

The recent interest in neural network approaches has been directed towards segmentation, for the most part because it represents a highly parallel implementation of the algorithms already described. For instance, Yamaguchi and Kropatsch have implemented a neural network in a pyramid, Hsieh has used neural network implementations for the extractions of subjective contours, and Pedrycz has proposed a neural implementation of relational clustering {YAMAGUCHI89, HSIEH89, PEDRYCZ91}.

11.6.4 Interactive Modification

We have pointed out that most methods do not stand on their own, but must be used in conjunction with other methods to solve a specific problem. For instance, a possible sequence of methods to segment nuclear features in a confocal image might consist of:

- thresholding or active contouring to isolate nuclei,
- edge-finding, active contouring, or watershed analysis to isolate nucleoli,
- watershed analysis or variable conductance blurring to isolate chromocenters,
- mathematical morphologic methods to clean up the data, and
- interactive editing.

Furthermore, no single approach will likely work for all problems.

Currently, most methods are best used as tools to make the process of interactive segmentation less difficult and time consuming. For example, when defining boundaries by contouring it is generally easier to edit automatically contoured boundaries than to draw all boundaries from scratch. The same principle exists for object segmentation. To be most useful, then, a segmentation method should be computationally efficient, so that it can be run interactively, with any lengthy preprocessing done before the user becomes involved. Also, the user should be allowed to direct the segmentation process or to modify the results in a manner consistent with the concepts driving the segmentation method and with the needs of the biologic problem.

Most biologists and physicians have neither the time nor the expertise to develop new methods of segmentation and will most likely never have the time or resources to put together their own front ends. Instead, they must use previously or easily implemented segmentation methods and commercially available or public domain front ends. One example of such a semi-automated system is the Interactive Hierarchy Viewing system developed at the University of North Carolina at Chapel Hill, in which regions are found using an automatically precomputed region containment hierarchy. Masters and Paddock have used the interactive capabilities of Voxel View in the rendering of confocal images of

the rabbit cornea {MASTERS90}, and Jones et al. describe a 3D interactive segmenter in their processing of confocal images of chromosomes {JONES90}.

In the final analysis, there is as yet no philosopher's stone of segmentation. But the techniques available are useful, and when integrated into a reasonably interactive front end they can help overcome the time-consuming and dreary work of manual segmentation.

11.7 Conclusions

Each of the methods we have described has its own advantages and disadvantages, and the choice of which to use for a particular application will depend on a number of different factors. Some of the issues that need to be considered are: the visualization objectives (what kind of information we would like to display), image quality, speed, ease of use, and ease of specification (how difficult it is to achieve desired results using the method). Because indirect methods often require a costly preprocessing step to create the intermediate polygons, it can be difficult to interactively modify the surface representation of data portrayed with these techniques. Nevertheless, indirect methods are still commonly used in applications designed to run on graphics workstations that specialize in fast polygon rendering. Direct methods that use segmentation to produce a binary volume before rendering sacrifice surface orientation information, making it difficult to obtain smoothly-shaded images and impossible to represent detail at subvoxel resolution. In addition, any method that makes a binary decision about the location of a surface is more likely to give unsatisfactory results when applied to data sets in which surface boundaries are not well-defined; nonbinary methods handle this problem by reflecting the uncertainty of an edge decision through the use of partial transparency, at the cost of a certain amount of fuzziness in the final image. However, methods that render opaque objects will generally run much faster than methods in which the final pixel values are affected by multiple voxels or primitives.

Acknowledgements

The code used to render the images in Figures 1-4 and 8 was written by Marc Levoy, and extended by the authors to generate Figures 5-7. The nuclear data in Figures 1-3 and 5-7 was supplied by Dr. Gary Smith, obtained with partial funding from NIH-CA24144. Dr. Robert Becker, Armed Forces Institute of Pathology and Dr. Carlo Pesce, National Institutes of Health provided the glomerulus data shown in Figure 8. Dr. Charles Jennette provided important aid in the acquisition and preparation of the tissue rendered in Figure 4. Ulrich Neumann offered valuable comments on a draft version of this chapter.

References

- Ehud Artzy, Gideon Frieder and Gabor T. Herman. "The Theory, Design, Implementation and Evaluation of a Three-Dimensional Surface Detection Algorithm", *Computer Graphics and Image Processing*, vol. 15, no. 1, January 1981, pp. 1-24.
- Haruo Asada and Michael Brady. "The Curvature Primal Sketch", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, January 1986, pp. 2-14.
- Christian Barillot, Bernard Gibaud, Jean-Marie Scarabin and Jean-Louis Coatrieux. "3D Reconstruction of Cerebral Blood Vessels", *IEEE Computer Graphics and Applications*, vol. 5, no. 12, December 1985, pp. 13-19.
- Lisa C. Baxter and James M. Coggins. "Supervised Pixel Classification Using a Feature Space Derived from an Artificial Visual System", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision IX: Algorithms and Techniques*, Boston, November 1990, *SPIE Proceedings Series*, vol. 1381, pp. 459-469.
- James F. Blinn. "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces", *Computer Graphics*, vol. 16, no. 3, July 1982, pp. 21-29.
- Harry Blum and Roger N. Nagel. "Shape Description Using Weighted Symmetric Axis Features", *Pattern Recognition*, vol. 10, no. 3, 1978, pp. 167-180.
- Jean-Daniel Boissonnat. "Shape Reconstruction from Planar Cross Sections", *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, October 1988, pp. 1-29.
- Charles Bouman and Bede Liu. "Multiple Resolution Segmentation of Textured Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, February 1991, pp. 99-113.
- G. J. Brakenhoff, H. T. M. van der Voort, M. W. Baarslag, B. Mans, J. L. Oud, R. Zwart and R. van Driel. "Visualization and analysis techniques for three dimensional information acquired by confocal microscopy", *Scanning Microscopy*, vol. 2, no. 4, December 1988, pp. 1831-1838.
- A. D. Brink. "Grey-level Thresholding of Images Using a Correlation Criterion", *Pattern Recognition Letters*, vol. 9, no. 5, June 1989, pp. 335-341.
- John Canny. "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, November 1986, pp. 679-698.

- Lih-Shyang Chen, Gabor T. Herman, R. Anthony Reynolds and Jayaram K. Udupa. "Surface Shading in the Cuberille Environment", *IEEE Computer Graphics and Applications*, vol. 5, no. 12, December 1985, pp. 33-43.
- Shiuh-Yung Chen, Wei-Chung Lin, Cheng-Chung Liang and Chin-Tu Chen. "Improvement on Dynamic Elastic Interpolation Technique for Reconstructing 3-D Objects from Serial Cross Sections", *IEEE Transactions on Medical Imaging*, vol. 9, no. 1, March 1990, pp. 71-83.
- Philip A. Chou. "Optimal Partitioning for Classification and Regression Trees", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, April 1991, pp. 340-354.
- H. N. Christiansen and T. W. Sederberg. "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics", *Computer Graphics*, vol. 12, no. 3, August 1978, pp. 187-192.
- H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford and B. C. Teeter. "Two algorithms for the three-dimensional reconstruction of tomograms", *Medical Physics*, vol. 15, no. 3, May/June 1988, pp. 320-327.
- Fernand S. Cohen, Zhigang Fan and Maqbool A. Patel. "Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, February 1991, pp. 192-202.
- James M. Coggins. "A multiscale description of image structure for segmentation of biomedical images", *Proceedings of the First Conference on Visualization in Biomedical Computing*, Atlanta, Georgia, 22-25 May, 1990, pp. 123-130.
- James L. Crowley and Alice C. Parker. "A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 2, March 1984, pp.156-170.
- David Cyganski and John A. Orr. "Applications of Tensor Theory to Object Recognition and Orientation Determination", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 6, November 1985, pp. 662-673.
- Andreas R. Dill, Martin D. Levine and Peter B. Noble. "Multiple Resolution Skeletons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, July 1987, pp. 495-504.
- Hudai Diriltan and Thomas G. Newman. "Pattern Matching Under Affine Transformations", *IEEE Transactions on Computers*, vol. 26, no. 3, March 1977, pp. 314-317.

- Louis J. Doctor and John G. Torborg. "Display Techniques for Octree-Encoded Objects", *IEEE Computer Graphics and Applications*, vol. 1, no. 3, July 1981, pp. 29-38.
- Robert A. Drebin, Loren Carpenter and Pat Hanrahan. "Volume Rendering", *Computer Graphics*, vol. 22, no. 4, August 1988, pp. 65-74.
- Richard O. Duda and Peter E. Hart. Pattern Classification and Scene Analysis, Wiley, New York, 1973.
- Edward J. Farrell, Rosario Zappulla and Wen C. Yang. "Color 3-D Imaging of Normal and Pathologic Intracranial Structures", *IEEE Computer Graphics and Applications*, vol. 4, no. 9, September 1984, pp. 5-17.
- Martin A. Fischler and Robert C. Bolles. "Perceptual Organization and Curve Partitioning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, January 1986, pp. 100-105.
- Per-Ola Forsgren. "Visualization and coding in three-dimensional image processing", *Journal of Microscopy*, vol. 159, Pt. 2, August 1990, pp. 195-202.
- Gideon Frieder, Dan Gordon and R. Anthony Reynolds. "Back-to-Front Display of Voxel-Based Objects", *IEEE Computer Graphics and Applications*, vol. 5, no. 1, January 1985, pp. 52-60.
- Dan E. Fritch, James M. Coggins and Stephen M. Pizer. "A Multiscale Medial Description of Greyscale Image Structure", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, November 1991.
- H. Fuchs, Z. M. Kedem and S. P. Uselton. "Optimal Surface Reconstruction from Planar Contours", *Communications of the ACM*, vol. 20, no. 10, October 1988, pp. 693-702.
- Richard S. Gallagher and Joop C. Nagtegaal. "An Efficient 3-D Visualization Technique for Finite Element Models and Other Coarse Volumes", *Computer Graphics*, vol. 23, no. 3, July 1989, pp. 185-194.
- S. Ganepathy and T. G. Dennehy. "A New General Triangulation Method for Planar Contours", *Computer Graphics*, vol. 16, no. 3, July 1982, pp. 69-75.
- Saul B. Gelfand, C. S. Ravishankar and Edward J. Delp. "An Iterative Growing and Pruning Algorithm for Classification Tree Design", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, February 1991, pp. 163-174.

- Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, November 1984, pp. 721-741.
- W. V. Glenn, R. J. Johnston, P. E. Morton and S. J. Dwyer. "Image generation and display techniques for CT scan data", *Investigative Radiology*, vol. 10, no. 5, September/October 1975, pp. 403-416.
- Samuel M. Goldwasser. "Rapid Techniques for the Display and Manipulation of 3-D Biomedical Data", *Proceedings of the Seventh Annual Conference and Exposition of the National Computer Graphics Association*, vol. 2, 1986, pp. 115-149.
- Robert M. Haralick, Stanley R. Sternberg and Xinhua Zhuang. "Image Analysis Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, July 1987, pp. 532-550.
- L. D. Harris, R. A. Robb, T. S. Yuen and E. L. Ritman. "Display and Visualization of Three-Dimensional Reconstructed Anatomic Morphology: Experience with the Thorax, Heart, and Coronary Vasculature of Dogs", *Journal of Computer Assisted Tomography*, vol. 3, no. 4, August 1979, pp. 439-446.
- John Haslett. "Maximum Likelihood Discriminant Analysis on the Plane Using a Markovian Model of Spatial Context", *Pattern Recognition*, vol. 18, nos. 3/4, 1985, pp. 287-296.
- Gabor T. Herman and H. K. Liu. "Display of Three-Dimensional Information in Computed Tomography", *Journal of Computer Assisted Tomography*, vol. 1, no. 1, 1977, pp. 155-160.
- Karl Heinz Hoehne and Ralph Bernstein. "Shading 3d-Images from CT Using Gray-Level Gradients", *IEEE Transactions on Medical Imaging*, vol. 5, no. 1, March 1986, pp. 45-47.
- Karl Heinz Hoehne, Robert L. Delapaz, Ralph Bernstein and Robert C. Taylor. "Combined Surface Display and Reformatting for the Three-Dimensional Analysis of Tomographic Data", *Investigative Radiology*, vol. 22, no. 8, August 1987, pp. 658-664.
- Karl Heinz Hoehne, Michael Bomans, Ulf Tiede and Martin Riemer. "Display of Multiple 3D-Objects Using the Generalized Voxel-Model", *Proceedings of the SPIE Conference on Medical Imaging II*, Newport Beach, California, 31 Jan - 5 Feb 1988, SPIE Proceedings Series, vol. 914, pp. 850-854.
- T. H. Hong and Azriel Rosenfeld. "Compact Region Extraction Using Weighted Pixel Linking in a Pyramid", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 2, March 1984, pp. 222-229.

- Cheng-Hong Hsieh. "A Connectionist Algorithm for Image Segmentation", PhD dissertation, University of North Carolina at Chapel Hill, 1989.
- Ming-Kuei Hu. "Visual Pattern Recognition by Moment Invariants", *IRE Transactions on Information Theory*, vol. 8, no. 2, February 1962, pp. 179-187.
- Anil K. Jain and Richard C. Dubes. Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- Ben-Kwei Jang and Roland T. Chin. "Analysis of Thinning Algorithms Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, June 1990, pp. 541-551.
- Jean-Michel Jolion, Peter Meer and Samira Bataouche. "Robust Clustering with Applications in Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, August 1991, pp. 791-802.
- S. J. Jones, M. L. Taylor, W. Baarslag, J. J. Krol, B. Mosterd, A. Mans, J. Brakenhoff and N. Nanninga. "Image processing techniques for 3-D chromosome analysis", *Journal of Microscopy*, vol. 158, Pt. 2, May 1990, pp. 235-248.
- James T. Kajiya and Brian P. Von Herzen. "Ray Tracing Volume Densities", *Computer Graphics*, vol. 18, no. 3, July 1984, pp. 165-173.
- E. Keppel. "Approximating Complex Surfaces by Triangulation of Contour Lines", *IBM Journal of Research and Development*, vol. 19, January 1975, pp. 2-11.
- Jan J. Koenderink and A. J. van Doorn. "Representation of Local Geometry in the Visual System", *Biological Cybernetics*, vol. 55, no. 6, March 1987, pp. 367-375.
- Walter G. Kropatsch. "Curve Representations in Multiple Resolutions", *Pattern Recognition Letters*, vol. 6, no. 3, August 1987, pp. 179-184.
- Wolfgang Krueger. "Volume Rendering and Data Feature Enhancement", *Computer Graphics*, vol. 24, no. 5, November 1990, pp.21-26.
- David Laur and Pat Hanrahan. "Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering", *Computer Graphics*, vol. 25, no. 4, July 1991, pp. 285-288.

- F. Leitner, I. Marque, S. Lavallee and P. Cinquin. "Dynamic Segmentation: Finding the Edge With Snake Splines", in *Curves and Surfaces*, Pierre-Jean Laurent, Alain Le Mehaute and Larry L. Schumaker, eds., Academic Press, Boston, 1991, pp. 279-284.
- Martin D. Levine. *Vision in Man and Machine*, McGraw-Hill, New York, 1985, pp 151-209.
- Marc Levoy. "Direct Visualization of Surfaces from Computed Tomography Data", *Proceedings of the SPIE Conference on Medical Imaging II*, Newport Beach, California, 31 Jan - 5 Feb 1988, SPIE Proceedings Series, vol. 914, pp. 828-839.
- Marc Levoy. "Display of Surfaces from Volume Data", *IEEE Computer Graphics and Applications*, vol. 8, no. 3, May 1988, pp. 29-37.
- Marc Levoy. "A Hybrid Ray Tracer for Rendering Polygon and Volume Data", *IEEE Computer Graphics and Applications*, vol. 10, no. 2, March 1990, pp.33-40.
- Marc Levoy. "Efficient Ray Tracing of Volume Data", *ACM Transactions on Graphics*, vol. 9, no. 3, July 1990, pp. 245-261.
- Lawrence M. Lifshitz and Stephen M. Pizer. "A Multiresolution Hierarchical Approach to Image Segmentation Based on Intensity Extrema", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, June 1990, pp. 529-540.
- Wei-Chung Lin, Cheng-Chung Liang and Chin-Tu Chen. "Dynamic Elastic Interpolation for 3-D Medical Image Reconstruction from Serial Cross Sections", *IEEE Transactions on Medical Imaging*, vol. 7, no. 3, September 1988, pp. 225-232.
- Wei-Jih Lin, Stephen M. Pizer and V. Eugene Johnson. "Boundary Estimation in Ultrasound Images", *Proceedings of the 12th International Conference on Information Processing in Medical Imaging*, Wye, U.K., July 1991, pp. 285-299.
- Hsun K. Liu "Two- and Three-Dimensional Boundary Detection", *Computer Graphics and Image Processing*, vol. 6, no. 2, April 1977, pp. 123-134.
- William E. Lorensen and Harvey E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, vol. 21, no. 4, July 1987, pp. 163-169.
- David G. Lowe. "Organization of Smooth Image Curves at Multiple Scales", *Proceedings of the 2nd International Conference on Computer Vision*, 1988, pp. 558-567.
- Yi Lu and Ramesh C. Jain. "Behavior of Edges in Scale Space", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 4, April 1989, pp. 337-356.

- Stephane G. Mallat. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, July 1989, pp. 674-693.
- B. S. Manjunath and R. Chellapa. "Unsupervised Texture Segmentation Using Markov Random Field Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, May 1991, pp. 478-482.
- Barry R. Masters and Stephen W. Paddock. "Three-dimensional reconstruction of the rabbit cornea by confocal scanning optical microscopy and volume rendering", *Applied Optics*, vol. 29, no. 26, 10 September 1990, pp. 3816-3822.
- Donald Meagher. "Geometric Modeling Using Octree Encoding", *Computer Graphics and Image Processing*, vol. 19, no. 2, June 1982, pp. 129-147.
- Hans-Peter Meinzer, Kirsten Meetz, Dinu Scheppelmann, Uwe Engelmann and Hans Jurgen Baur. "The Heidelberg Ray Tracing Model", *IEEE Computer Graphics and Applications*, vol. 11, no. 6, November 1991, pp. 34-43.
- Markus Montag, Herbert Spring, Michael F. Trendelenburg and Andres Kriete. "Methodical aspects of 3-D reconstruction of chromatin architecture in mouse trophoblast giant nuclei", *Journal of Microscopy*, vol. 158, Pt. 2, May 1990, pp. 225-233.
- Annick Montanvert, Peter Meer and Azriel Rosenfeld. "Hierarchical Image Analysis Using Irregular Tessellations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, April 1991, pp. 307-316.
- Bernard M. E. Moret. "Decision Trees and Diagrams", *ACM Computing Surveys*, vol. 14, no. 4, December 1982, pp. 593-623.
- Lee Richard Nackman. "Three-dimensional Shape Description Using the Symmetric Axis Transform", PhD dissertation, University of North Carolina at Chapel Hill, 1982.
- Derek R. Ney, Elliot K. Fishman, Donna Magid and Robert A. Drebin. "Volumetric Rendering of Computed Tomography Data: Principles and Techniques", *IEEE Computer Graphics and Applications*, vol. 10, no. 2, March 1990, pp. 24-32.
- K. Niklas Nordstrom. "Biased Anisotropic Diffusion -- A Unified Regularization and Diffusion Approach to Edge Detection", Technical Report UCB/CSD 89/514, May 1989, Computer Science Division, University of California, Berkeley, California.
- Kevin L. Novins, Francois X. Sillion and Donald P. Greenberg. "An Efficient Method for Volume Rendering using Perspective Projection", *Computer Graphics*, vol. 24, no. 5, November 1990, pp. 95-102.

- J. R. Parker. "Gray Level Thresholding in Badly Illuminated Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, August 1991, pp. 813-819.
- W. Pedrycz. "Neurocomputations in Relational Systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, March 1991, pp. 289-296.
- Alex Pentland and Stan Sclaroff. "Closed-Form Solutions For Physically Based Shape Modeling and Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, February 1991, pp. 715-729.
- Pietro Perona and Jitendra Malik. "Scale Space and Edge Detection using Anisotropic Diffusion", *IEEE Workshop on Computer Vision*, Miami, June 1987, pp. 16-22.
- Stephen M. Pizer, Henry Fuchs, Charles Mosher, Lawrence Lifshitz, Gregory D. Abram, Sukumar Ramanathan, Brian T. Whitney, Julian G. Rosenman, Edward V. Staab, Edward L. Chaney and George Sherouse. "3-D Shaded Graphics in Radiotherapy and Diagnostic Imaging", *Proceedings of the Seventh Annual Conference and Exposition of the National Computer Graphics Association*, vol. 3, 1986, pp. 107-113.
- Stephen M. Pizer, William R. Oliver and Sandra H. Bloomberg. "Hierarchical Shape Description Via the Multiresolution Symmetric Axis Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, July 1987, pp. 505-511.
- Stephen M. Pizer, John M. Gauch and Lawrence M. Lifshitz. "Interactive 2D and 3D object definition in medical images based on multiresolution image descriptions", *Proceedings of the SPIE Conference on Medical Imaging II*, Newport Beach, California, 31 Jan - 5 Feb 1988, SPIE Proceedings Series, vol. 914, pt. A, pp. 438-444.
- A. Pommert, U. Tiede, G. Wiebecke and K. H. Hoehne. "Image Quality in Voxel-Based Surface Shading", *Proceedings of the International Symposium on Computer Assisted Radiology*, 1989, pp. 737-741.
- W. Qian and D. M. Titterington. "On the use of Gibbs Markov chain models in the analysis of images based on second-order pairwise interactive distributions", *Journal of Applied Statistics*, vol. 16, no. 2, 1989, pp. 267-281.
- T. H. Reiss. "The Revised Fundamental Theorem of Moment Invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, August 1991, pp. 830-834.

- R. Anthony Reynolds, Dan Gordon and Lih-Shyang Chen. "A Dynamic Screen Technique for Shaded Graphics Display of Slice-Represented Objects", *Computer Vision, Graphics and Image Processing*, vol. 38, 1987, pp. 275-298.
- Michael L. Rhodes, Yu-Ming Kuo, Alex T. Pang and Stephen L. G. Rothman. "An Image Coherence Technique for Ray Traced Three-Dimensional Presentations in Clinical Radiology", *Proceedings of the International Symposium on Computer Assisted Radiology*, 1987, pp. 615-623.
- Thomas Risse. "Hough Transform for Line Recognition: Complexity of Evidence Accumulation and Cluster Detection", *Computer Vision, Graphics and Image Processing*, vol. 46, no. 3, June 1989, pp. 327-345.
- Paolo Sabella. "A Rendering Algorithm for Visualizing 3D Scalar Fields", *Computer Graphics*, vol. 22, no. 4, August 1988, pp. 51-55.
- Firooz A. Sadjadi and Ernest L. Hall. "Three-Dimensional Moment Invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 2, March 1980, pp. 127-136.
- Philippe Saint-Marc, Jer-Sen Chen and Gerard Medioni. "Adaptive Smoothing: A General Tool for Early Vision", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 13, no. 6, June 1991, pp. 514-529.
- Hanan Samet. "The Quadtree and Related Hierarchical Data Structures", *Computing Surveys*, vol. 16, no. 2, June 1984, pp. 187-260.
- Daniel S. Schlüsselberg and Wade K. Smith. "Three-Dimensional Display of Medical Image Volumes", *Proceedings of the Seventh Annual Conference and Exposition of the National Computer Graphics Association*, vol. 3, 1986, pp. 114-123.
- Thorsten Schormann, Michel Robert-Nicoud and Thomas M. Jovin. "Improved Stereovisualization Method for Confocal Laser Scanning Microscopy", *European Journal of Cell Biology*, Supplement 25 (vol. 48), Proceedings of the First International Conference on 3-D Image Processing in Microscopy, Giessen, West Germany, March 1988, pp. 53-54.
- Thorsten Schormann and Thomas M. Jovin. "Optical sectioning with a fluorescence confocal SLM: procedures for determination of the 2-D digital modulation transfer function and for 3-D reconstruction by tessellation", *Journal of Microscopy*, vol. 158, Pt. 2, May 1990, pp. 153-164.
- Peter Shirley and Allan Tuchman. "A Polygonal Approximation to Direct Scalar Volume Rendering", *Computer Graphics*, vol. 24, no. 5, November 1990, pp.63-70.

- Alexander Sunguroff and Donald Greenberg. "Computer Generated Images for Medical Applications", *Computer Graphics*, vol. 12, no. 3, August 1978, pp. 196-202.
- Demetri Terzopoulos, Andrew Witkin and Michael Kass. "Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion", *Artificial Intelligence*, vol. 36, no. 1, August 1988, pp. 91-123.
- Demetri Terzopoulos and Dimitri Metaxas. "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, February 1991, pp. 703-714.
- Ulf Tiede, Karl Heinz Hoehne, Michael Bomans, Andreas Pommert, Martin Riemer and Gunnar Wiebecke. "Investigation of Medical 3D-Rendering Algorithms", *IEEE Computer Graphics and Applications*, vol. 10, no. 2, March 1990, pp.41-53.
- Yves Troussset and Francis Schmitt. "Active-Ray Tracing for 3D Medical Imaging", *Proceedings of the European Computer Graphics Conference and Exhibition, EUROGRAPHICS '87*, pp. 139-150.
- Heang K. Tuy and Lee Tan Tuy. "Direct 2-D Display of 3-D Objects", *IEEE Computer Graphics and Applications*, vol. 4, no. 10, October 1984, pp. 29-33.
- Jayaram K. Udupa and Hsiu-Mei Hung. "Surface Versus Volume Rendering: A Comparative Assessment", *Proceedings of the First Conference on Visualization in Biomedical Computing*, Atlanta, Georgia, 22-25 May 1990, pp. 83-91.
- Craig Upson and Michael Keeler. "V-BUFFER: Visible Volume Rendering", *Computer Graphics*, vol. 22, no. 4, August 1988, pp. 59-64.
- H. T. M. van der Voort, G. J. Brakenhoff, J. A. C. Valkenburg and N. Naninga. "Design and use of a computer-controlled confocal microscope", *Scanning*, vol. 7, pp. 66-78.
- H. T. M. van der Voort, G. J. Brakenhoff and M. W. Baarslag. "Three-dimensional visualization methods for confocal microscopy", *Journal of Microscopy*, Vol. 153, Pt 2, February 1989, pp. 123-132.
- Luc Vincent and Pierre Soille. "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, June 1991, pp. 583-598.
- Harry Wechsler. Computational Vision, Academic Press, Boston, 1990.

- Lee Westover. "Footprint Evaluation for Volume Rendering", *Computer Graphics*, vol. 24, no. 4, August 1990, pp. 367-376.
- Ross Whitaker. "A Multi-Scale Approach to Nonuniform Diffusion", Technical Report 91-040, University of North Carolina at Chapel Hill Department of Computer Science, 1991, to appear in *Graphical Models and Image Processing*.
- Jane Wilhelms and Allan Van Gelder. "Topological Considerations in Isosurface Generation", Technical Report 90-14, Computer Research Laboratory, University of California at Santa Cruz, April 1990.
- Jane Wilhelms and Allan Van Gelder. "Octrees for Faster Isosurface Generation", Technical Report 90-28, Computer Research Laboratory, University of California at Santa Cruz, June 1990.
- Jane Wilhelms and Allan Van Gelder. "A Coherent Projection Approach for Direct Volume Rendering", *Computer Graphics*, vol. 25, no. 4, July 1991, pp. 275-284.
- Andrew Witkin. "Scale-Space Filtering", *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983, vol. 2, pp. 1019-1022.
- Andrew Witkin, Demetri Terzopoulos and Michael Kass. "Signal Matching Through Scale Space", *International Journal of Computer Vision*, vol. 1, no. 2, 1987, pp. 133-144.
- Xuanli Lisa Xie and Gerardo Beni. "A Validity Measure for Fuzzy Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, August 1991, pp. 841-847.
- Lei Xu, Erkki Oja and Pekka Kultanen. "A New Curve Detection Method: Randomized Hough Transform (RHT)", *Pattern Recognition Letters*, vol. 11, no. 5, May 1990, pp. 331-338.
- T. Yamaguchi and W. G. Kropatsch. "A Vision by Neural Network or by Pyramid?", *Proceedings of the 6th Scandinavian Conference on Image Analysis*, June 1989, Oulu, Finland.