3D Segmentation Techniques for Medical Volumes

Sarang Lakare
Center for Visual Computing
Department of Computer Science
State University of New York at Stony Brook

Research Proficiency Exam
Advisor: Prof. Arie Kaufman

Dec 14, 2000

Abstract

Volume segmentation is an important part of computer based medical applications for diagnosis and analysis of anatomical data. With rapid advances in medical imaging modalities and volume visualization techniques, computer based diagnosis is fast becoming a reality. These computer based tools allow scientists and physicians to understand and diagnose anatomical structures by virtually interacting with them. Volume segmentation plays a critical role by facilitating automatic or semi-automatic extraction of the anatomical organ or region-of-interest. In this review, we provide an introduction to various segmentation algorithms found in the literature. We classify the algorithms into three categories: structural techniques, statistical techniques and hybrid techniques. Under structural techniques we will review algorithms which take into consideration structural information for segmentation. Stochastic techniques are those which perform segmentation based on statistical analysis methods and under hybrid techniques we will review algorithms which make use of structural information in addition to statistical analysis.

Contents

1 Introduction To Volume Segmentation **3**

2 Segmentation Techniques **3**

2.1 Structural Techniques 4

2.1.1 3D Edge-Detection Techniques 4

2.1.2 Morphological Techniques 5

2.1.3 Graph-Searching Algorithms 5

2.1.4 Deformable Models 6

2.1.5 Isosurfaces and Level Sets 9

2.2 Stochastic Techniques 11

2.2.1 Thresholding Approaches 11

2.2.2 Classification Techniques 12

2.2.3 Clustering Algorithms 12

2.2.4 Markov Random Fields 14

2.3 Hybrid Approaches 15

2.3.1 Region Growing 15

2.3.2 Split and Merge 15

2.3.3 Atlas-Guided Approaches 16

2.3.4 Artificial Neural Networks 16

2.3.5 LEGION Based 16

3 Conclusions **17**

1 Introduction To Volume Segmentation

Rapid advances in the field of medical imaging are revolutionizing medicine. Computed tomography (CT), magnetic resonance imaging (MRI), and other imaging modalities provide an effective means of non-invasively mapping the anatomy of a subject. This allows scientists and physicians to virtually interact with anatomical structures and learn potentially life saving information.

Today, the role of medical imaging is not limited to simple visualization and inspection of anatomic structures, but goes beyond that to patient diagnosis, advanced surgical planning and simulation, radiotherapy planning etc. Although modern volume visualization techniques provide extremely accurate and high quality 3D view of anatomical structures, their utilization for accurate and efficient analysis is still limited.

One of the main reasons for this is the highly complex internal structure of animals and humans with vast number of anatomical organs bunched together, hindering the physicians view in more ways than one. Some visualization tricks like making an object transparent do not work in such cases. To tackle this issue, the anatomical structure or the region of interest needs to be delineated and separated out so that it can be viewed individually. This technique is known as *image segmentation* in the world of medical imaging. Since segmentation of organs or region-of-interest from single image is of hardly any significance for volume rendering, we only concentrate on segmentation from 3D volumes (which are basically consecutive images stacked together). We thus refer to this technique as *volume segmentation*.

Segmentation in medical imaging is generally considered a very difficult problem. This difficulty mainly arises due to the sheer size of the datasets coupled with the complexity and variability of the anatomic organs. The situation is worsened by the shortcomings of imaging modalities, such as sampling artifacts, noise, low contrast etc. which may cause the boundaries of anatomical structures to be indistinct and disconnected. Thus the main challenge of segmentation algorithms is to accurately extract the boundary of the organ or region-of-interest and separate it out from the rest of the dataset.

There are many approaches for segmentation proposed in literature. These vary widely depending on the specific application, imaging modality (CT, MRI, etc.), and other factors. For example., the segmentation of lungs has different issues than the segmentation of colon. The same algorithm which gives excellent results for one application, might not even work for another. Besides these, general imaging artifacts like noise, motion and partial volume effect can significantly affect the outcome of a segmentation algorithm. For example., a segmentation algorithm could be robust against noise, but at the same time, it might fail miserably in the presence of partial volume effects. This variability is what makes segmentation a very challenging problem. There is currently no segmentation method that provides acceptable results for every type of medical dataset. There are methods in existence which are generalized and can be applied to a variety of data, but methods specialized for the particular problem always give better results. Often a segmentation approach could consist of more than one segmentation algorithms applied one after the other. Selection of an appropriate algorithm or approach for segmentation can therefore be a difficult dilemma.

We will now review the various segmentation algorithms that have appeared in the literature so far. We will try to understand the algorithms by giving a brief overview of each of them and then we will discuss their merits and down-falls.

2 Segmentation Techniques

The number of segmentation algorithms found in the literature is very high. Due to the nature of the problem of segmentation, most of these algorithms are specific to a particular problem, thus,

having little significance for most other problems. We will try to cover all the algorithms that have a generalized scope and which are the basis of most of the segmentation techniques today. In addition, we will concentrate only on 3D volumes and thus present each algorithm with respect to its application on 3D volumes.

There are many good papers in the literature reviewing the various segmentation algorithms [28][50][54]. Every paper has a different structure of classifying the segmentation algorithms. We try to classify the algorithms in the following way.

We broadly classify the segmentation techniques into three classes.

1. Structural techniques
2. Stochastic techniques
3. Hybrid techniques

The classification is done based on the approach used for segmentation. Under structural techniques, we will review those techniques which utilize some information about the structure of the region in segmenting it. Stochastic techniques are the ones that are applied on discrete voxels without any consideration for the structure of the region. Localized information on a per-voxel basis is used to decide whether or not the voxel belongs to the desired region. These include the traditional low-level segmentation algorithms. The final category is the hybrid methods which include those techniques which posses characteristics of both structural and stochastic techniques.

2.1 Structural Techniques

As already discussed, structural techniques try to find structural properties of the region to be segmented. Structural properties such as intersecting surfaces (edges in 2D) are detected in the volume and then combined to segment the region. In some algorithms, structure information is saved and later retrieved to perform segmentation on a similar dataset (e.g., segmenting lever from many abdominal datasets.)

2.1.1 3D Edge-Detection Techniques

Edge detection techniques are those which aim at detecting edges or surfaces in the volume to perform segmentation. Edges are formed at the intersection of two regions with different intensities. They are one of the main cues for visual distinction of two regions [5]. Edge detection techniques in three dimensions work in two stages:

1. Local edges are detected by using some form of differentiation.
2. These local edges are grouped together to form boundary contours that separate the desired region voxels from other voxels.

A number of edge detecting operators have been proposed for this purpose. Liu [42] proposed a 3D surface detection algorithm that extends the classical Robert's operator into 3D space. Herman and Liu [32] later extended this algorithm to 4D. Zucker and Hummel [90] [91] developed an optimal three dimensional edge detection operator, which was essentially a Sobel operator.

One advantage of edge detection techniques is that they work very well on datasets with good contrast between different regions. The edges are detected perfectly and can be verified visually. On the down side, these algorithms detect all the edges. It is very difficult to find the correlation between the edges and the region-of-interest. In addition, these algorithms do not perform well on datasets with low contrast between regions. These algorithms are also susceptible to noise. In most of the cases, these algorithms are not used on their own for segmentation, but coupled with other segmentation algorithms to solve a particular segmentation problem.

2.1.2 Morphological Techniques

Mathematical morphology uses set transformations for image analysis [65]. It extracts the impact of a particular shape on images via the concept of structuring elements (SE). The SE encodes the primitive shape information. The shape is described as a set of vectors referenced to a particular point, the center. During morphological operations, the center scans the whole image and the matching shape information is used to define the transformation. The transformed image is thus a function of the SE distribution in the whole image. The basic morphological operations can be described on the basis of an arbitrary space E. Let $P(E)$ be the set of all subsets $X \epsilon E$. With each point X of space E, a spatially varying set $B(X)$ called the SE is associated. The set $X \epsilon P(E)$ can be modified based on set transformation of X by E. Let B_x denote the translation of B by the vector x.

The two most fundamental transforms in mathematical morphology are erosion and dilation. These can be defined on the basis of the above assumptions as

1. Erosion:$\{X : B_x \subset X\}$
   The eroded set of X is the locus of centers x of translated B_x included in the set X. This is denoted as $X\theta B$ and is given by

$$X\theta B = \bigcap_{b \epsilon B} X_b \tag{1}$$

2. Dilation:
   Dilation is dual transform of erosion and can be expressed as

$$X^C \oplus B = (X\theta B)^C \tag{2}$$

   where $\oplus$ denotes dilation and C denotes the compliment operation.

Morphological operations are generally simple to understand and implement. At the same time, these are generally difficult to control. For example., it is difficult to control the dilation operation unless you give the upper limit to the number of times it dilates. Thus, these algorithms generally require some external criteria to control them. These operations also have a risk of changing the morphology of the input datasets. It is well known that a series of dilations followed by erode operations leads to loss of high frequencies (for example, folds in a colon), and fills holes. Similarly a series of erodes followed by dilations can introduce holes and high frequencies. These algorithms should be avoided when accuracy is the primary concern and there is a risk of loss of important data. As with edge detectors, morphological operations are not segmentation algorithms by themselves but they are generally an integral part of a segmentation pipeline.

2.1.3 Graph-Searching Algorithms

In these algorithms, edges and surfaces in a volume are represented as graphs and the algorithm tries to find the lowest-cost path between two nodes of the graph using a search algorithm such as A^* [2][20] or F^* [21]. These algorithms are especially useful when the partitions between regions in the desired segmented volume are not well defined. The F^* algorithm [62] is used extensively in biomedical imaging and hence we will discuss it here.

F^* boundary-forming algorithm

In principle, both the F^* and the A^* are similar. In A^* algorithm, a minimum-cost path from the starting point (s) to the goal point (g) is iteratively constructed by extending the best partial path available at each iteration. This is done by selecting the point v that has the minimum-cost path from s to g via v where the cost is the sum of lowest-cost paths found so far from s to v and the estimate of minimum cost paths from v to g. For a simple implementation the algorithm requires O(N^2) operations. Similarly, the F^* algorithm finds the optimum path from s to g using a cost array C by iteratively updating a path array P. This array (P) is initialized to in-

finity except at s, which is set to $C(s)$. The first step in updating consists of adjusting all the elements of the yth row from left to right using the rule,

$$\begin{aligned} P(x,y) &= \mathbf{min}\{P(x-1,y-1)+C(x,y), \\ &\quad P(c,y-1)+C(x,y), \\ &\quad P(x+1,y-1)+C(x,y), \\ &\quad P(x-1,y)+C(x,y), P(x,y)\} \end{aligned}$$

and then adjusting all the elements in the yth row from right to left, using

$$P(x,y) = \mathbf{min}\{P(x+1,y)+C(x,y), P(x,y)\} \tag{3}$$

Each additional pass involves a bottom-to-top pass followed by a top-to-bottom pass using the above two rules. When all the changes in P are such that the new value is greater than $P(g)$, the algorithm terminates and the optimum path can be found by backtracking from g and moving along the minimum value of P at each neighborhood until s is reached. Since the number of iterations required is the amber of "row" index reversals along the optimal path, this algorithm performs better than A^* in general.

The main advantage of this method is that it can perform well even if the partitions between regions is broken. At the same time, it also requires these surfaces to be represented as graphs, which could be tricky. Another disadvantage (from volume visualization point of view) is that this algorithm deals with surfaces. To get the voxel representation of these surfaces, another pass is needed to convert the surfaces to voxels [52].

2.1.4 Deformable Models

Deformable models are curves, surfaces or solids defined within an image or volume domain and they deform under the influence of external and internal forces. In the physics-based modeling paradigm, the data apply forces (external forces) to the deformable model and as a result the model moves towards the data, while internal forces keep the model smooth during deformation. Deformable models gained popularity after they were proposed to use in computer vision [74] and computer graphics [76] by Terzopoulos and others in 1988.

Mathematically, a deformable model moves according to its dynamic equations and seeks the minimum of a given energy function. The deformation of a typical 2-D deformable model can be characterized by the following dynamic equation:

$$\mu(s)\frac{\partial^2 \mathbf{x}(s,t)}{\partial t^2} + \gamma(s)\frac{\partial \mathbf{x}(s,t)}{\partial t} = \mathbf{F}_{int} + \mathbf{F}_{ext} \tag{4}$$

where $\mathbf{x}(s,t) = (x(s,t), y(s,t))$ is a parametric representation of the position of the model at a given time t, and $\mu(s)$ and $\gamma(s)$ are parameters representing the mass density and damping density of the model, respectively. Eq.(4) causes the model to move according to the direction and magnitude of the forces on the right hand side. The most commonly used internal forces are

$$\mathbf{F}_{int} = \frac{\partial}{\partial s}(\alpha(s)\frac{\partial \mathbf{x}(s,t)}{\partial s}) - \frac{\partial^2}{\partial s^2}(\beta(s)\frac{\partial^2 \mathbf{x}}{\partial s^2}) \tag{5}$$

which represent internal stretching and blending forces. The most commonly used external forces are computed as the gradient of an edge map.

Physically based deformable models can be divided into three categories: energy minimizing snakes, dynamic deformable models, and probabilistic deformable models.

Energy minimizing snakes

Snakes [35] is the most popular form of deformable models. Snakes are planar deformable contours that are useful in several image analysis tasks. Using energy minimization formulation, the goal of this approach is to find a parametric model that minimizes the weighted sum of internal energy and potential energy. The internal energy specifies the tension or the smoothness of the surface of the model. The potential energy is defined over the volume domain and typ-

ically possesses local minima at the edges occurring at object boundaries. Minimizing the total energy yields internal and potential forces. As a result, these are attracted to image features such as lines and edges.

We will now represent this mathematically. A snake embedded in the image plane $(x, y)\epsilon\Re^2$ is represented as $v(s) = (x(s), y(s))^T$, where x and y are the coordinate functions and $s\epsilon[0, 1]$ is the parametric domain. The shape of the contour subject to an image $I(x, y)$ is dictated by the functional,

$$\xi(v) = \psi(v) + \phi(v) \tag{6}$$

The functional can be viewed as a representation of the energy of the contour and the final shape of the contour corresponds to the minimum of this energy. The first term of the functional,

$$\psi(v) = \int_0^1 w_1(s)|\frac{\partial v}{\partial s}|^2 + w_2(s)|\frac{\partial^2 v}{\partial s^2}|^2 ds \tag{7}$$

is the internal deformation energy. It characterizes the deformation of a stretchy, flexible contour. Two physical parameter functions dictate the simulated physical characteristics of the contour: $w_1(s)$ controls the 'tension' of the contour while $w_2(s)$ controls its 'rigidity'. The second term in (6) couples the snake to the image. Traditionally,

$$\phi(v) = \int_0^1 P(v(s))ds \tag{8}$$

where $P(x, y)$ denotes a scalar potential function defined on the image plane. To apply snakes to images, external potentials are designed whose local minima coincide with intensity extrema, edges and other image features of interest.

Dynamic deformable models

Although it is natural to think of energy minimization as a static problem, a potent approach to computing the local minima of functional such as (6) is to construct a dynamical system that is governed by the functional and allow the system to evolve to equilibrium. Equilibrium is achieved when the internal and external forces balance and the contour comes to rest. This leads to dynamic deformable models that unify the description of shape and motion, making it possible to quantify not just static shape, but also shape evolving through time.

Probabilistic deformable models

Deformable models can also be viewed as a model fitting process in a probabilistic framework. This permits the incorporation of prior model and sensor model characteristics in terms of probability distributions. The probabilistic framework also provides a measure of the uncertainty of the estimated shape parameters after the model is fitted to the image data [69].

Let u represent the deformable model shape parameters with a prior probability $p(u)$ on the parameters. Let $p(I/u)$ be the imaging (sensor) model - the probability of producing an image I giving a model u. Bayes's theorem

$$p(u/I) = \frac{p(I/u)p(u)}{p(I)} \tag{9}$$

expresses the posterior probability $p(u/I)$ of a model given the image, in terms of the imaging model and the prior probabilities of model and image.

The internal energy measure (Figure 7) of the deformable model is converted into a prior distribution over expected shapes, with lower energy shapes given the highest probability. This is done using a Boltzmann (or Gibbs) distribution of the form

$$p(u) = \frac{1}{Z_s} exp(-S(u)) \tag{10}$$

where S(u) is the discretized version of $\psi(v)$ in 7 and Z_s is a normalizing constant (called the partial function). This prior model is then combined with a simple sensor model based on linear measurements with Gaussian noise

$$p(I/u) = \frac{1}{Z_I} exp(-P(u)) \tag{11}$$

where $P(u)$ is a discrete version of the potential $\phi(v)$ in 8, which is a function of the image $I(x, y)$.

The fitting of the models is done by finding u which locally maximizes $p(u/I)$ in 9. This is known as the maximum *a posteriori* solution.

Deformable models in segmentation Many researches have explored the use of deformable surface models for segmentation. Typically users initialized a deformable model near the region-of-interest and allowed it to deform into place. Users could then manually fine-tune the fitting by using interactive capabilities of the models. The first uses of deformable models in medical images analysis was the application of deformable contour models, such as snakes [35], to segment structures in 2D images [13]. To segment 3D medical datasets, each 2D slice was segmented separately. Once a 2D slice was segmented, the contour of that slice was used as a reference contour for neighboring slices. This reference contour was then deformed into place in those slices. This process was repeated for all the 2D slices. The resulting sequence of 2D contours was then connected to form a continuous 3D surface model [13][14].

The 3D segmentation process described above is both laborious and requires a post-processing step to connect the sequence of 2D contours into a continuous surface. In addition, the reconstructed surface can have various inconsistencies. A true 3D segmentation technique could overcome all these shortcomings giving smooth 3D surfaces. In one of the initial work on segmentation using 3D deformable surfaces, Miller [48] in 1991 constructed a *balloon* by approximating a sphere using polygons. He then geometrically deformed this balloon until its surface conformed to the object surface in 3D CT data. The segmentation process is formulated as the minimization of a cost function, which is a weighted sum of three terms : a deformation potential that *expands* the model vertices towards the object boundary, an image term that identifies features such as edges and opposes the balloon expansion, and a term that maintains the topology of the model by constraining each vertex to remain close to the centroid of its neighbors.

Deformable superquadrics proposed by Terzopoulos et al. [75] and deformable generalized cylinders, proposed by O'Donnell and Gupta [70], incorporated global shape parameters of a super-ellipsoid and generalized cylinder, respectively. Local degree of freedom was based on elastic properties and action of external forces. These models can be used to extract gross shape features from visual data, which can be used for indexing onto a database of stored models to provide shape recognition. Local deformations help in reconstructing the details of complex shapes to provide shape reconstruction.

In related work, Cohen and Cohen [12][14] used finite-element techniques to implement an elastically deformable cylinder. Later, McInerney and Terzopoulos [47] used physics-based techniques to implement an elastically deformable sphere. Whitaker [85], Tek and Kimia [72], Davatzikos and Bryan [18] and others have also done notable work with 3D volumes.

Deformable models have the advantage that they offer a coherent and consistent mathematical description and are robust to noise and boundary gaps due to their incorporation of a smoothness constraint. Another advantage is that the offer sub-voxel accuracy for the boundary representation that may be important to a number of applications. A very important advantage of these models from the point of view of medical imaging is that these models are capable of accommodating the often significant variability of biological structures over time and across different individuals.

A disadvantage is that they require manual interaction to place an initial model in the dataset. Some researchers have attempted to decrease sensitivity to insignificant edges and initial model placement [60][31][9][22][8][45]. These algorithms also require the user to choose appropriate initial parameters. Various methods have been proposed to reduce sensitivity to initialization [14][3][56][26]. Another disadvantage from

the point of view of volume visualization is that these methods work only on surfaces. There is a lot of work being done to extend this idea to volumetric solid models.

2.1.5 Isosurfaces and Level Sets

Isoserfaces are defined by connecting voxels with intensities equal to the *isovalue* in a 3D volume. *level sets*, introduced by Osher and Sethian in 1988 [49] are, in short, moving fronts (curves). The underlying philosophy of this technique is to use isosurfaces as a modeling technology that can serve as an alternative to parameterized models (section 2.1.4).

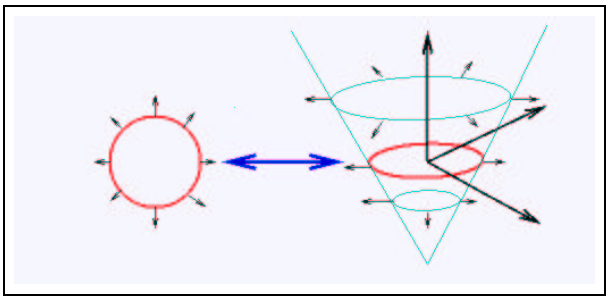

Figure 1: Original front (left) and *level set* function (right).

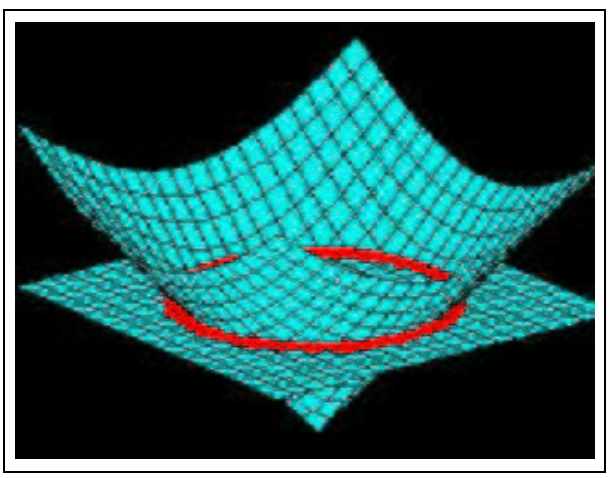

Figure 2: A level set surface.

Level-sets are numerical techniques designed to track the evolution of interfaces, which in our case would be the iso-surface. Other numerical techniques attempt to follow moving boundaries by putting a collection of marker points on the evolving surface and then changing their position to correspond to the moving surface. In contrast, level-set methods exploit a strong link between moving surfaces and equations from computational fluid equations.

Rather than follow the interface itself, the level set approach instead takes the original curve (red one in Figure 1), and build it into a surface. The cone-shaped surface (shown also in Figure 2) has a great property; it intersects the xy plane *exactly* where ther curve sits. This surface is called the level-set function; it takes as input any point on the plane and returns the height of that point. The red curve (Figure 1) is called the *zero level set*, because it is the collection of all points that are at height zero.

Isosurface is an implicit surface model for a 3D volume. It can be defined as a *level set* of scalar function,

$$\phi : U_{x,y,zr} \mapsto \mathbb{R}_{k'} \tag{12}$$

where $U \subset \mathbb{R}^3$ is the domain of the volume (and the *range* of the surface model). Thus, a surface S is

$$S = \{x|\phi(x) = k\} \tag{13}$$

The choice of k is arbitrary, and ϕ is sometimes called the *embedding*. Notice that the isosurface defined in this way divides U into a clear inside and outside - thus, they are always closed whenever they do not intersect the boundary of the domain.

Now the question comes, how to represent ϕ. In this approach, a large number of local *basis* functions are defined. This is the principle behind using a volume as an implicit model. A volume is a discrete sampling of the embedding ϕ. It is also an implicit model with a very large number of basis functions, as show in Figure 3. The total number of basis functions, their positions (grid points), and extents are fixed. The only change allowed to the basis function is its magnitude. Thus, each basis function has only

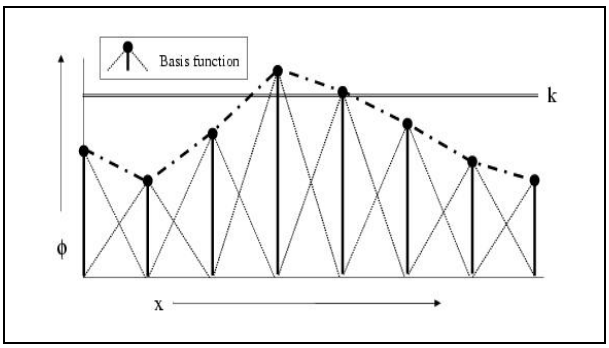

Figure 3: A volume can be considered as an implicit model with a large number of local basis functions.

one degree of freedom. A typical volume of 256x256x256 will have over 16 million basis functions. The shape of each basis function depends on how one interpolates the values between grid points. For example., a trilinear interpolation implies a basis function that is piecewise cubic polynomial with a value of one at the grid point and zero at neighboring grid points.

The method of level-sets, proposed by Osher and Sethian [44] [49][66], provides a set of numerical methods that describe how to manipulate the grey-scale values in a volume, so that the isosurfaces of ϕ move in a prescribed manner (Figure 4). To understand this, we denote the movement of a point on a surface as it deforms as dx/dt, and we assume that this motion can be expressed in terms of the position of $x \epsilon U$ and the geometry of the surface at that point. In this case, there are generally two options for representing such surface movements implicitly:

Static: A single, static $\phi(x)$ contains a family of level sets corresponding to surfaces at different times t. That is,

$$\phi(x(t)) = k(t) \Rightarrow \nabla\phi(x).\frac{\partial x}{t} = \frac{dk(t)}{dt}. \quad (14)$$

To solve this static method requires constructing a ϕ that satisfies equation 14. This representation has some significant limitations, because by construction a surface cannot pass back over itself over time, i.e., motions must be strictly monotonic – inward or outward.

Dynamic: This is a one-parameter family of embeddings, i.e., $\phi(x,t)$ changes over time, x remains on the k level set of ϕ as it moves, and k remains constant. The behavior of ϕ is obtained by setting the total derivative of $\phi(x(t),t) = k$ to zero. Thus,

$$\phi(x(t),t) = k \Rightarrow \frac{\partial\phi}{\partial t} = -\nabla\phi.\frac{dx}{dt}. \quad (15)$$

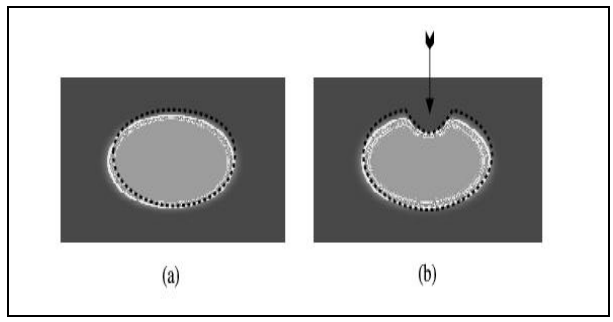

Figure 4: Level-set models represent curves and surfaces implicitly using 3D volume (a slice is shown): a) an ellipse is represented as the level set of an image, b) to change the shape, grey-scale values at the voxels of the volume are modified.

This approach can accommodate models that move forward and backwards and cross back over their own paths (over time). However, to solve this requires solving the initial value problem (using finite forward differences) on $\phi(x,t)$ – a potentially large computational burden. We will now only look at the dynamic case, because of its superior flexibility.

All surface movements depend on position and geometry, and the level-set geometry is expressed in terms of the differential structure of ϕ:

$$\frac{\partial\phi}{\partial t} = -\nabla\phi.\frac{dx}{dt} = -\nabla\phi.\mathbf{F}(x, D\phi, D^2\phi, ...) \quad (16)$$

where $D^n\phi$ is the set of order-n derivatives of ϕ evaluated at x. Because this relationship applies to every level-set of ϕ, i.e. all values of k, this equation can be applied to all of U, and therefore

the movements of *all* the level-set surfaces embedded in ϕ can be calculated from equation 16.

The level-set method has been shown to be effective for segmentation in medical datasets. Whitekar et al. have shown [87][86] that level sets can be used to simulate conventional deformable surface models, and demonstrated this by extracting skin and tumors from *thick-sliced* (eg. clinical) MR data, and by reconstructing a fetal face from 3D ultrasound. Recently, Sethian [67] presented several examples of level-set curves and surface for segmenting CT and MR data.

The level-set representation has a number of practical and theoretical advantages over conventional surface models, especially in the context of deformation and segmentation. First, level-set models are topologically flexible, they can easily represent complicated surface shapes that can, in turn, form holes, split to form multiple objects, or merge with other objects to form a single structure. These models can incorporate many (millions) of degrees of freedom, and therefore can accommodate complex shapes. Thus, there is no need to re-parameterize the model as it undergoes significant changes in shape.

2.2 Stochastic Techniques

We will now look at algorithms which perform segmentation by statistical analysis only. These algorithms do not take into account any structural information.

2.2.1 Thresholding Approaches

Thresholding is probably the simplest of the segmentation techniques for scalar volumes [84]. In this technique a single value called *threshold* is used to create a binary partition of voxel intensities. All voxels with intensities greater than the threshold are grouped together into one class and those with intensities below the threshold are grouped together into another class. Use of a single threshold thus results in a binary segmented volume.

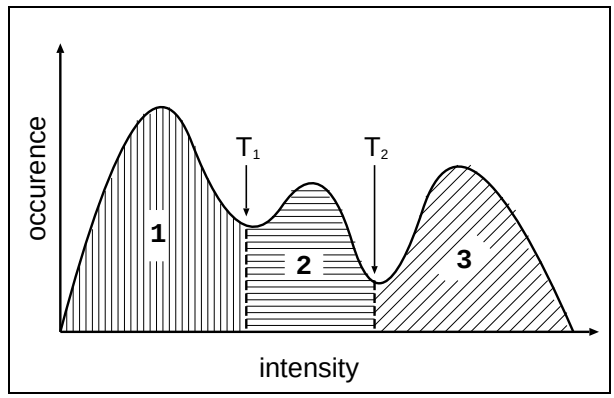

Figure 5: Histogram of a volume with two thresholds T_1 and T_2 dividing the histogram in three regions.

This technique can be extended to using multiple thresholds, where a region is defined by two thresholds, a *lower* threshold and an *upper* threshold. Each voxel of the input volume then belongs to one of the regions based on its intensity. This technique is known as multi-thresholding [61]. In Figure 5 we showed histogram of a volume. To apply thresholding, we take two thresholds T_1 and T_2 as shown. We then get three distinct regions as seen from the histogram.

Although simple, this technique is very effective in getting segmentation done in volumes with a very good contrast between regions. This is generally used as the first step towards segmentation of a volume.

The main drawback of this technique is that the results are too tightly coupled with the thresholds used. Any change in the threshold values can give a different segmented region. The thresholds are usually generated interactively by using visual feedback. Some automatic methods do exist with varying degree of success to automate the process of finding correct thresholds [34]. Another drawback which is a direct consequence of the previous one is that the technique is very sensitive to noise and intensity inhomogeneities. Thus it cannot be easily applied to MRI and ultrasound volumes.

2.2.2 Classification Techniques

Classification techniques are pattern recognition techniques that seek to partition a feature space derived from the volume using data with known labels [55][64][7]. A *feature space* is the range of an N-dimensional feature vector made from features at each voxel. The features could include the voxel intensity, the gradient at the voxel, the distance of the voxel from the volume boundary and so on. Mathematically, a *feature space* can be the range space of any function of the volume.

Classifiers belong to the *supervised* category as they require training data that are pre-segmented (either manually or by other method). The pre-segmented data is then used as reference to carry out automatic segmentation on new data.

The simplest form of a classifier is the nearest-neighbor classifier, where each pixel or voxel is classified in the same class as the training datum with the closest intensity. The k-nearest-neighbor (kNN) classifier is the generalization of this approach, where the pixel is classified according to the majority of the k closest training data. Another example of a similar classifier is the Parzen window, where the classification is made according to the majority vote within a predefined window of the feature space centered at the unlabeled voxel (mapped to feature space). Both these classifiers are non-parametric since they don't make any assumption about the statistical structure of the data.

Another commonly used classifier is the maximum likelihood (ML) or Bayes classifier. The basic assumption is that the voxel intensities are independent samples from a mixture of probability distributions, usually Gaussian. This mixture, called a finite mixture model, is given by the probability density function

$$f(y_j;\theta,\pi) = \sum_{k=1}^{K} \pi_k f_k(y_j;\theta_k) \qquad (17)$$

where y_j is the intensity of pixel j, f_k is a component probability density function parameterized by θ_k, and $\theta = [\theta_1, ..., \theta_K]$. The variables π_k are mixing coefficients that weight the contribution of each density function and $\pi = [\pi_1, ..., \pi_K]$. To collect the training data, representative samples from each component of the mixture model are obtained. Each θ_k is estimated from them. For Gaussian mixtures, this means estimating K means, covariances, and mixing coefficients. Classification of new data is obtained by assigning each voxel to the class with the highest posterior probability. When the data truly follows a finite Gaussian mixture distribution, the ML classifier can perform well and is capable of providing a soft segmentation composed of the posterior probabilities.

Standard classifiers require that the structure to be segmented possess distinct quantifiable features. Because training data can be labelled, classifiers can transfer these labels to new data as long as the feature space sufficiently distinguishes each label as well. Being non-iterative, they are relatively computationally efficient and unlike thresholding, they can be applied to multi-channel volumes. A disadvantage of classifiers is that they generally do not perform any spatial modeling. This weakness has been addressed by a recent work which incorporated neighborhood and geometric information. Another disadvantage is the manual interaction for obtaining training data. Training sets can be acquired from each volume that requires segmentation, but this can be time consuming and laborious. On the other hand, use of the same training set for a large number of scans can lead to biased results which do not take into account anatomical and physiological variability between different subjects.

2.2.3 Clustering Algorithms

These are clustering-based techniques which use characteristics of the voxel and its immediate neighborhood to do clustering. Clustering can be loosely defined as the process of grouping objects into groups, whose members show similar

properties. In our case these "objects" are the data voxels and the "groups" are the segmented regions. "Similar properties" could be any property the data voxel posses, like the density, gradient, color (for a color dataset) etc.

Clustering-based segmentation is similar to the classifier methods (2.2.1) with the exception that these do not use any training data. These techniques thus come under the *unsupervised* class of algorithms for segmentation. These algorithms overcome the need for a training data by iterating between segmenting the volume and characterizing the properties of each class. We could say that clustering-based algorithms train themselves using the available data.

The various clustering algorithms available today can be grouped into two broad categories :

1. Hierarchical methods
   These methods include those techniques where the input data is not partitioned into clusters in a single step. A series of successive fusions of data are performed until each cluster of size greater than one is composed of smaller clusters.

2. Non-Hierarchical methods
   In these methods, the desired number of clusters is known or assumed at the beginning of the clustering process. The end result is such that each data voxel gets assigned to exactly one cluster in this algorithm.

As in the case of classification, voxel properties such as intensity, gradient, neighborhood information etc. are used to form an N-dimensional feature vector for each voxel. Each class of the region is assumed to form a distinct cluster in the N-dimensional feature space. A suitable clustering algorithm, (K-means clustering, leader clustering, spatial clustering, etc.) is then applied to each voxel in the feature space. The resultant clusters in the feature space are then mapped to spatial domains to give the desired regions.

The most commonly used clustering algorithms for segmentation are K-means clustering [15],

K-means clustering

This algorithm takes as input a set of N dimensional vectors without any prior knowledge about the set. After processing, the algorithm forms K disjoint nonempty subsets such that each subset minimizes some measure of dissimilarity. By minimizing dissimilarity of each subset locally, the algorithm will globally yield an optimal dissimilarity of all subsets. The dissimilarity for a voxel is its distance from the mean of each of the classes in the feature space. The mean for each class is computed iteratively. The voxel is added to the cluster whose mean is the nearest to the voxel (meaning least dissimilarity between the voxel and the cluster's mean).

The algorithm has a time complexity $O(RKN)$, where K is the number of desired clusters, and R is the number of iterations until it converges.

fuzzy clustering

The input to the algorithm is a finite data set $X = x_1, x_2, ..., x_n$, each $x_i \epsilon X$ is a feature vector; $x_i = (x_{i1}, x_{i2}, ..., x_{is})$ where x_{ij} is the jth feature of subset x_i, and s is the dimensionality of x_i. A function $u : X \rightarrow [0, 1]$ is defined, which assigns to each x_i in X its grade of membership in the fuzzy set u. The function u is called a fuzzy subset of X. The goal is to partition X by means of fuzzy sets. A fuzzy c-partition is defined as c x n matrix U such that

1. Each row U_i represents the ith fuzzy subset of X.

2. Each column U^j exhibits the membership grades of datum j in every fuzzy subset.

3. The membership grades of each datum in all fuzzy subsets adds up to 1.

4. No fuzzy subset is empty.

5. No fuzzy subset is all of X.

Let M_{fc} denote the fuzzy c-partitions of X, then $U \epsilon M_{fc}$. The fuzzy c-means algorithm uses iterative optimization to approximate minima of an objective function J_m [1].

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (U_{ik})^m (d_{ik})^2 \qquad (18)$$

where $v = (v_1, v_2, ..., v_c)$ with v_i being the cluster center of class $i; 1 \leq i \leq c$ and $d_{ik}^2 = \| x_k - v_i \|^2$.

clustering using graph theory
Various graph-theoretic approaches have been proposed for data clustering [33][46][89][78][37][88] We review an algorithm presented by Wu and Leahy [88] in 1993. In this algorithm, the data to be clustered are represented by an undirected adjacency graph G. Each vertex of G corresponds to a data point, and an arc links two vertices in G if the corresponding data points are neighbors according to a given neighborhood system. A flow capacity is then assigned to each arc in G. This is chosen to reflect the feature similarity between the pair of linked vertices. The clustering is achieved by removing arcs of G to form mutually exclusive subgraphs. For the case of an unconstrained optimal K-subgraph partition of G, the arcs selected for removal are those in a set of K - 1 minimum cuts with the smallest K - 1 values among all possible minimum cuts separating all pairs of vertices. The method minimizes the largest inter-subgraph maximum flow among all possible K partitions of G, hence minimizing the similarity between subgraphs, which in this case are clusters. The reason for this method of minimization can be explained as follows.

The purpose of the clustering algorithm is to group together components into a minimal number of clusters. This can be formulated in terms of the adjacency graph G formed from the components. G can be divided into a number of unconnected subgraphs by removal of the arcs connecting the subgraphs. The set of vertices in each subgraph then represents a single cluster. Each of the remaining subgraphs contains a set of connected vertices or components whose union represents a spatially connected region of the volume. Since arc capacities are a measure of the similarity between connected neighbors, partitioning a graph G into two subgraphs with as dissimilar features as possible would involve minimizing the maximum flow between the two subgraphs [88].

Although clustering algorithms do not require training data, they do require an initial segmentation (or equivalently, initial parameters). The expectation-minimization (EM) [41] algorithm has demonstrated greater sensitivity to initialization than the K-means or fuzzy c-means algorithms [19]. Like classifier methods, clustering algorithms do not directly incorporate spatial modeling and can therefore be sensitive to noise and intensity inhomogeneities. This lack of spatial modeling, however, can provide significant advantages for fast computation. Work on improving the robustness of clustering algorithms to intensity inhomogeneities in MR images has demonstrated excellent success [25][53]. Robustness to noise can be incorporated using Markov random field modeling as we will see in the next section.

2.2.4 Markov Random Fields

Markov random field (MRF) modeling itself is not a segmentation method but a statistical model which can be used within segmentation methods. MRFs model spatial interaction between neighboring or nearby voxels. These local correlations provide a mechanism for modeling a variety of image properties [40]. In medical imaging, they are typically used to take into account the fact that most pixels belong to the same class as their neighboring pixels. In physical terms, this implies that any anatomical structure that consists of only one pixel has a very low probability of occurring under a MRF assumption.

MRFs are often incorporated into clustering segmentation algorithms such as the K-means algorithm under a Bayesian prior model [51][57] [30][25]. The segmentation is done by maximizing the *a posteriori* probability of the segmentation given the volume data using iterative methods such as iterated conditional modes [6] or simulated annealing [24].

A difficulty associated with MRF models is proper selection of the parameters controlling the strength of spatial interactions[40]. Too high a setting can result in an excessively smooth segmentation and a loss of important structure details. In addition, MRF methods usually require computationally intensive algorithms. Despite these disadvantages, MRFs are widely used not only to model segmentation classes, but also to model intensity inhomogeneities that can occur in MR images [30] and texture properties [59].

2.3 Hybrid Approaches

In this section we will look at the segmentation algorithms which cannot be classified into the previous two categories. These algorithms use something from both the previous two types of segmentation algorithms.

2.3.1 Region Growing

This is probably the simplest among the hybrid techniques. Region growing is a technique to extract a connected region from a 3D volume based on some pre-defined connecting criterion. This criteria can be as simple as the voxel intensity or could be the output of any other segmentation algorithm [28]. In the simplest form, region growing requires a *seed* point to start with. From the seed point, the algorithm grows till the connecting criteria is satisfied.

As with thresholding, region growing is simple, but not often used for segmentation by itself. More often than not, region growing forms a part of a segmentation pipeline for a particular approach. It is often used as the primary method to understand a 3D data before more complex segmentation is applied to it.

The primary disadvantage of this algorithm is that it requires seed points which generally means manual interaction. Thus for each region to be segmented, a seed point is needed. Region growing can also be sensitive to noise and partial volume effect causing the extracted region to have holes or disconnections. Some recent work has been reported which tries to alleviate these problems. In another recent work, fuzzy analogies to region growing have also been developed.

2.3.2 Split and Merge

This algorithm is similar to region growing we saw earlier. This algorithm requires the input data to be organized into a pyramidal grid structure of regions, with each region organized in groups of eight (for 3D) [5]. Any region can be split into eight subregions and the appropriate eight can be merged into a single larger region. As in region growing, the criteria for merging (growing for region-growing) could be anything. It could be as simple as voxel intensity or some condition checking based on the output of some previous segmentation stage. Let us assume that the criterion is C. The algorithm can be written down in two steps as follows:

1. Pick a region R in the grid structure. If C(R) is false, split the region into eight subregions. If for eight regions $R_1, R_2, ..., R_8$, $C(R_1 \cup R_2 \cup ... \cup R_8) = true$, merge into single region. When no regions can be merged, stop.

2. If there are neighboring regions R_i and R_j such that $C(R_i \cup R_j) = true$, merge these regions.

The big advantage of this method over region-growing is that no seed points are needed and hence no manual interaction is needed. On the down side, it requires the input to be organized into a pyramidal grid structure which could be undesirable for the huge datasets in use today.

2.3.3 Atlas-Guided Approaches

Atlas-guided approaches use a standard atlas or template to perform segmentation. The atlas is generated by compiling information on the anatomy that requires segmentation. This atlas is then used to segment other images. The standard atlas-guided approach treats segmentation as a registration [43] problem. It first finds a one-to-one transformation that maps a pre-segmented atlas image to the target image that requires segmentation. This process is often referred to as *atlas-warping*. The warping can be performed using linear transformations [71][39][4].

Atlas guided approaches have been mainly applied in MR brain imaging. An advantage of atlas-guided approaches is that labels are transferred as well as the segmentation. The main shortcoming of this method is due to anatomical variability. To overcome this problem, many researchers have tried to apply a sequence of linear and non-linear transformations [10][16][17][63]. Even with this, accurate segmentation of complex structures is very difficult. Thompson and Toga in 1997 [77] introduced *probabilistic atlases* to model anatomical variability, but their method required additional time and interaction to accumulate data. Hence, these approaches are best suited for segmenting structures which are stable over the population of study.

2.3.4 Artificial Neural Networks

Conventional segmentation algorithms based on structural knowledge often require considerable user expertise. The Artificial neural networks (ANN) based approaches tried to partially overcome these drawbacks. ANNs are massively parallel networks of processing elements or nodes that simulate biological learning. Each node in an ANN is capable of performing elementary computation. Learning is achieved through the adaptation of weights assigned to the connections between nodes. A more detailed description of ANNs can be found in [11] [29].

The main features of ANNs which the segmentation algorithms try to use are :

1. Learning from examples and generalizing that knowledge
2. Noise rejection
3. Fault tolerance
4. Optimum seeking behavior

Valli [79] presented three architectures for medical image segmentation based on ANNs. These architectures showed that ANNs can successfully exploit and integrate different kinds of a priori information contained in medical images. His experiments demonstrated robustness and sensitivity of the approach, but at the expense of generality.

ANNs are widely used in segmentation as a classifier [27][23], where the weights are determined using training data, and the ANN is then used to segment new data. ANNs can also be used in an unsupervised fashion as a clustering method [7] [58], as well as for deformable models [80].

Since the ANNs are tightly interconnected, spatial information can be easily incorporated into its classification procedures. Although ANNs are inherently parallel, their processing is usually simulated on a standard serial computer, thus reducing this potential computational advantage.

2.3.5 LEGION Based

These segmentation methods are based on a biologically inspired oscillator network, called the locally excitatory globally inhibitory oscillator network (LEGION) [82][73][83]. LEGION was proposed by Terman and Wang [82] [73] as a biologically plausible computational framework for image analysis. The network was proposed based on theoretical and experimental considerations that point to *oscillatory correlation* as a representational scheme for the working of the brain.

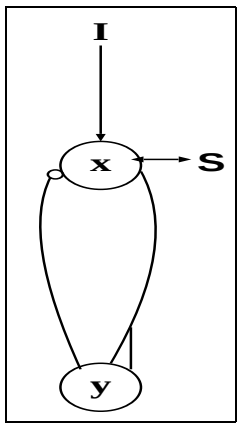

Figure 6: Diagram of a single oscillator in a LEGION.

The oscillatory correlation theory assumes that the brain groups and segregates visual features on the basis of correlation between neural oscillations [82] [81]. It has been shown theoretically and later, by experiments that neural oscillations in the visual cortex are a possible mechanism by which the brain detects and binds features in a visual scene.

LEGION is a network of relaxation oscillators, each constructed from an excitatory unit x and an inhibitory unit y as shown in Figure 6. Unit x sends excitation to unit y which responds by sending inhibition back. When external input stimulus I is continuously applied to x, this feedback loop produces oscillations. Neighboring oscillators are connected via mutual excitatory coupling, as well as the global inhibitor.

The formal definition of LEGION can be found in [83]and [73]. The behavior of each oscillator, indexed by i in a network, is defined by the following equations:

$$\dot{x}_i = 3x_i - x_i^3 + 2 - y_i + \rho + I_i H(p_i + exp(-\alpha t) - \theta) + S_i \quad (19)$$

$$\dot{y}_i = \epsilon(\gamma(1 + \tanh(x_i/\beta)) - y_i) \quad (20)$$

$$\dot{p}_i = \lambda(1 - p_i) H[\sum_{k \epsilon N_2(i)} T_{ik} H(x_k - \theta_x) - \theta_p] - \mu p_i \quad (21)$$

$$S_i = \sum_{k \epsilon N_2(i)} W_{ik} H(x_k - \theta_x) - W_z H(z - \theta_z) \quad (22)$$

$$\dot{z} = \phi(\sigma_o - z) \quad (23)$$

The explanation of these equations can be found in [68]. To summarize , after a number of oscillation cycles a block of oscillators corresponding to a major image region will oscillate in synchrony, while any two oscillator blocks corresponding to two different major regions will desynchronize from each other. For other voxels which do not belong to any major region, the corresponding oscillator will stop oscillating shortly after the system starts.

Simulation of a LEGION network is computationally expensive since it requires numerically integrating a huge number of differential equations. This makes it almost impossible to segment large volume datasets using this technique. To make it feasible, Wang and Terman [83] proposed a simplified algorithm. Recently Shareef et al. [68] further simplified the Wang and Terman algorithm for efficiency purposes, which are particularly important from the point of view of volume segmentation.

A few of the advantages of this method are: requires less intervention compared to most of the structural techniques, initial parameter setting can be fully automated, and can achieve good noise tolerance. On the other hand, due to generality, domain-specific knowledge is not utilized to the full compared to the structural techniques.

3 Conclusions

We have reviewed and discussed most of the segmentation algorithms widely used for segmenting 3D medical datasets. We classified the approaches into three categories : structural techniques, stochastic techniques and hybrid techniques. Under structural techniques, we reviewed algorithms which try to find structural properties like edges and then segment the organ or region-of-interest. On the other hand, stochastic techniques do not give any consideration to structural information. They aim at performing segmentation based on mathemat-

ical analysis of data. Hybrid techniques are those which contain components from the previous two.

Structural techniques reviewed in here have the advantage that they can segment out a single organ or region-of-interest. The accuracy of segmentation depends on the quality of input data. Since these algorithms locate structural information, any noise in the input data could affect the outcome. A lot of research has been done to make these techniques robust against noise with varying degree of success. This is especially useful when we consider MRI datasets, which are known to have a lot of noise. Structural techniques also need good contrast between the regions to be segmented in the input dataset. A low contrast means difficulty in finding "edges", which might affect these algorithms. Another disadvantage for structural techniques is that they are difficult to automate. Manual intervention is mostly needed to select seed points inside the desired region-of-interest.

Stochastical techniques presented here perform segmentation on the entire dataset. This results in the segmentation of entire datasets into different regions. Although this process of segmentation is fully automated (baring the selection of initial parameters), selecting the region among the segmented regions could be tedious and may need manual intervention. For eg., if we classify the visible human dataset, we will get all bones classified as one region. If we are just interested in one of them, say the skull, we will have to manually locate the skull and remove all other bones. This is generally done by running the dataset through another pass of an algorithm like region-grow to mark the selected region and to remove everything else. The accuracy and quality of segmentation in these algorithms depends on the selection of initial parameters. This could be good and bad. The good part being that with proper selection of parameters, the algorithm could be made robust against noise for a particular problem. It could also be tuned to perform segmentation on low-contrast datasets. On the other hand, the selection of parameters is too critical. Any incorrect selection could lead to undesirable segmentation. These properties make some of these techniques more suitable to MRI datasets than the structural techniques.

Where are we heading to? If we glance at the history of "image segmentation", we observe that the initial work was done with 2D images. As computers became powerful and surface rendering techniques started to evolve, the focus shifted to segmenting 3D volumes. But this generally meant constructing a surface for the segmented region, either during segmentation (eg., deformable models, graph-searching algorithms etc.) or after segmentation (eg., stochastical techniques). Then started the era of volume visualization. Volume visualization is considered to be a superior technique to surface rendering due to the ability to look *through* the volume. Until recently, interactive volume visualization was a difficult task, but with the availability of special-purpose hardware[36], it is now possible to interactively visualize volumes even on low-end PCs. This has resulted in a considerable interest in volume rendering based tools.

Due to these advances in volume rendering techniques, the focus of segmentation is now shifting slowly to volume segmentation. Techniques such as level-sets (section 2.1.5) convert the surface-based deformation techniques to volume-based deformation. They directly alter the voxel intensities in the volume to get the desired effect of deformation. Due to this voxel-based approach, these techniques are gaining a lot of interest. All the stochastic techniques mentioned before (section 2.2) are already used for voxel based volume segmentation. Other techniques [38] are also being developed which are of special interest to volume rendering. To describe this in one line, the day is not far when *image* segmentation is known as *volume* segmentation.

Another important desired feature of segmentation is "automation". Performing automated segmentation still remains one of the most diffi-

cult problems in the world of segmentation. Although researchers have shown success with automation in some cases, there is no generic algorithm which can perform automatic segmentation on any given dataset.

References

[1] R. Acharya and R. P. Menon. A review of biomedical image segmentation techniques. pages 140–161, 1998.

[2] A.Martelli. Edge Detection Using Heuristic Search Methods. *Computer Graphics and Image Processing*, 25:169–182, Aug. 1978.

[3] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(9):855–867, Sept. 1990.

[4] N. C. Andreasen, R. Rajarethinam, and T. C. et al. Automatic atlas-based volume estimation of human brain regions from mr images. *J. Comp. Assist. Tom.*, 20:98–106, 1996.

[5] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, N.J., 1982.

[6] J. Besag. On the statistical analysis of dirty pictures (with discussion). *J. Roy. Statist. Soc. Ser. B*, 48:259–302, 1986.

[7] J. C. Bezdek, L. O. Hall, and L. P. Clark. Review of MR image segmentation techniques using pattern recognition. *Med. Phys.*, 20:1033–1048, 1993.

[8] A. Chakraborty and J. S. Duncan. Integration of boundary finding and region-based segmentation using game theory. In *Information Processing in Medical Imaging*, pages 189–200, France, June 1995.

[9] A. Chakraborty, L. H. Staib, and J. S. Duncan. Deformable boundary finding influenced by region homogeneity. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 624–627, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.

[10] G. E. Christensen, S. C. Joshi, and M. I. Miller. Volumetric transformation of brain anatomy. *IEEE Trans. Medical Imaging*, 16:864–877, 1997.

[11] J. W. Clark. Neural network modelling. *Phys. Med. Biol.*, 36:1259–1371, 1991.

[12] I. Cohen, L. D. Cohen, and N. Ayache. Using deformable surfaces to segment 3d images and infer differential structures. *CVGIP: Image Understanding*, 56(2):242–263, 1992.

[13] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.

[14] L. D. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2d and 3d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[15] G. B. Coleman and H. C. Andrews. Image segmentation by clustering. In *Proc. IEEE*, pages 773–785, 1979.

[16] D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans. Automatic 3-d model-based neuroanatomical segmentation. *Human Brain Mapping*, 3:190–208, 1995.

[17] C. Davatzikos. Spatial normalization of 3d images using deformable models. 20:656–665, 1996.

[18] C. Davatzikos and R. N. Bryan. Using a deformable surface model to obtain a mathematical representation of the cortex. In *International Symp. on Computer Vision*, pages 212–217. IEEE Computer Society Press, Nov. 1995.

[19] J. W. Davenport, J. C. Bezdek, and R. J. Hathaway. Parameter-estimation for finite mixture distributions. *Computers & Mathematics with Applications*, 15(10):819–828, 1988.

[20] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, N. Y., 1973.

[21] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–203, Mar 1981.

[22] J. M. Gauch, H. H. Pien, and J. Shah. Hybrid boundary-based and region-based deformable models for biomedical image segmentation. In *Mathematical Methods in Medical Imaging III*, volume 2299, pages 72–83, San Diego, CA, USA, 1994. SPIE.

[23] E. Gelenbe, Y. Feng, and K. R. R. Krishnan. Neural Network methods for volumetric magnetic resonance iamging of the human brain. pages 1488–1496. IEEE, 1996.

[24] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions of Pattern Recognition and Machine Intelligence*, 6:721–741, Nov. 1984.

[25] A. F. Goldszal, C. Davatzikos, D. L. Pham, M. X. H. Yan, R. N. Bryan, and S. M. Resnick. An image processing system for qualitative and quantitative volumetric analysis of brain images. 22:827–837, 1998.

[26] R. P. Grezeszczuk and D. N. Levin. Brownian strings: Segmenting images with stochastically deformable contours. In *Proc. Third Conf. on Visualization in Biomedical Computing*, pages 72–89. SPIE, 1994.

[27] L. O. Hall, A. M. Bensaid, L. P. Clarke, R. P. Velthuizen, M. S. Silbiger, and J. C. Bezdek. A comparison os neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE T. Neural Networks*, 3:672–682, 1992.

[28] R. M. Haralick and L. G. Shapiro. Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, Jan. 1985.

[29] S. Haykin. *Neural Networks; A comprehensive Foundation*. Macmillan, New York, 1 edition, 1994.

[30] K. Held, E. R. Kops, B. J. Krause, W. M. Wells, and R. K. et al. Markov random field segmentation of brain mr images. *IEEE T. Med. Imag.*, 16(6), 1997.

[31] I. L. Herlin, C. Nguyen, and C. Graffigne. A deformable region model using stochastic processes applied to echocardiographic images. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 534–539. IEEE Computer Society Press, June 1992.

[32] G. T. Herman and H. K. Liu. Dynamic Boundary Surface Detection . *Computer Graphics and Image Processing*, 7:130–138, 1978.

[33] L. J. Hubert. Some applications of graph theory to clustering. 38:435–475, 1974.

[34] T. Jiang, M. B. Merickel, and E. A. Parrish. Automated Threshold Detection Using a Pyramid Data Structure. In *9th International Conference on Pattern Recognition*, pages 689–692, 1988.

[35] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Conputer Vision*, 1(4):321–331, 1988.

[36] A. Kaufman, F. Dachille, B. Chen, I. Bitter, K. Kreeger, N. Zhang, Q. Tang, and H. Hua. Real Time Volume Rendering. *To appear in the Special Issue on 3D Imaging of the International Journal of Imaging Systems and Technology*, 2000.

[37] W. L. Koontz, P. M. Narendra, and K. Fukunaga. A graph theoretic approach to nonparameter cluster analysis. 24:936–944, Sept. 1976.

[38] S. Lakare, M. Wan, M. Sato, and A. Kaufman. 3D Digital Cleansing Using Segmentation Rays. In *IEEE Visualization*, pages 37–44, 2000.

[39] J. L. Lancaster, L. H. Rainey, J. L. Summerlin, C. S. Freitas, P. T. Fox, A. C. Evans, and A. W. T. an J. C. Mazziotta. Automated labeling of the human brain: A preliminary report on the development and evaluation of a forward-transform method. *Human Brain Mapping.*, 5:238–242, 1997.

[40] S. Z. Li. Markov random field modeling in computer vision. *Springer*, 1995.

[41] Z. Liang, J. R. MacFall, and D. P. Harrington. Parameter estimation and tissue segmentation from multispectral mr images. 13:441–449, 1994.

[42] H. K. Liu. Two and Three Dimensional Boundary Detection . *Computer Graphics and Image Processing*, 6:123–134, Apr. 1977.

[43] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2:1–36, 1998.

[44] R. Malladi, J. A. Sethian, and B. C. Vemuri. A Fast Level Set based Algorithm for Topology-Independent Shape Modeling. *J. Math. Imaging and Vision*, 6(2/3):269–290, 1996.

[45] J. F. Mangin, F. Tupin, V. Frouin, I. Bloch, R. Rougetet, J. Regis, and J. Lopez-Krahe. Deformable topological models for segmentation of 3d medical images. In *Information Processing in Medical Imaging*, pages 153–164, France, June 1995.

[46] D. W. Matula. Graph theoretic techniques for cluster analysis algorithms. pages 95–129, 1977.

[47] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, 1995.

[48] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara, and M. J. Wozny. Geometrically Deformed Models: A Method for Extracting Closed Geometric Models from Volume Data. *Computer Graphics (Proc. SIGGRAPH '91 Conf.)*, 25(4):217–226, 1991.

[49] S. Osher and J. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[50] N. R. Pal and S. K. Pal. A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[51] T. N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Trans. Signal Processing*, 40(4):901–914, Apr. 1992.

[52] T. Pavlidis. Segmentation of pictures and maps through functional approximation. 1:360–372, 1972.

[53] D. L. Pham and J. L. Prince. An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity homogeneities. *Pattern Recognition Let.*, pages 57–68, 1999.

[54] D. L. Pham, C. Xu, and J. L. Prince. A Survey of Current Methods In Medical Image Segmentation. Technical report, Johns Hopkins University, Baltimore, Jan. 1998.

[55] D. L. Pham, C. Xu, and J. L. Prince. Current Methods in Medical Image Segmentation. *Annual Review of Biomedical Engineering*, 2, 2000.

[56] C. S. Poon, M. Braun, R. Fahrig, A. Ginige, and A. Dorrell. Segmentation of medical images using an active contour model incorporating region-based image features. In *Proc. Third Conf. on Visualization in Biomedical Computing*, pages 90–97. SPIE, 1994.

[57] J. C. Rajapakse, J. N. Giedd, and J. L. Rapoport. Statistical approach to segmentation of single-channel cerebral mr images. *IEEE T. Med. Imag.*, 16:176–186, 1997.

[58] W. E. Reddick, J. O. Glass, E. N. Cook, T. D. Elkin, and R. J. Deaton. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. *IEEE T. Med. Imag.*, 16:911–918, 1997.

[59] T. R. Reed and J. M. H. D. Buf. A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 57(3):359–372, May 1993.

[60] N. Rougon and F. Prêteux. Deformable markers: Mathematical morphology for active contour models control. In *Image Algebra and Morphological Image Procssing II*. SPIE, 1991.

[61] P. K. Sahoo, S. Soltani, and A. K. C. Wong. A Survey of Thresholding Techniques. *Computer Vision, Graphics, and Image Processing*, 41:233–260, 1988.

[62] J. K. Samarabandu, R. S. Acharya, and P. C. Cheng. *Analysis and Presentation of Three Dimensioanl Data Sets*. Spring Verlag, Berlin, 1993.

[63] S. Sandor and R. Leahy. Surface-based labelling of cortical anatomy using a deformable atlas. 16:41–54, 1997.

[64] R. J. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches*. Wiley, New York, 1992.

[65] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[66] J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.

[67] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[68] N. Shareef, D. L. Wang, and R. Yagel. Segmentation of medical images using legion. *IEE Transactions on Medical Imaging*, 18(1):74–91, Jan 1999.

[69] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal on Computer Vision*, 5:271–301, 1990.

[70] e. a. T. O'Donnell. Extruded generalized Cylinder: A Deformable Model for Object Recovery. In *Proc. CVPR*. IEEE Computer Society Press, 1994.

[71] J. Talairach and P. Tournoux. *Co-Planar Stereotaxic Atlas of the Human Brain. 3-Dimensional Proportional System: An Approach to cerebral Imaging.* Thieme Medical Publisher, Inc., Stuttgart, NY, 1988.

[72] H. Tek and B. B. Kimia. Shock-based reaction-diffusion bubbles for image segmentation. In N. Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medicine*, Lecture Notes in Computer Science, pages 434–438. Springer-Verlag, Apr. 1995. ISBN 3-540-59120-6.

[73] D. Terman and D. L. Wang. Global competition and local cooperation in a network of neural oscillators. *Physics D*, 81:148–176, 1995.

[74] D. Terzopoulos and K. Fleischer. Deformable Models. *The Visual Computer*, 4(6):306–331, Dec. 1988.

[75] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(7):703–714, July 1991.

[76] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–124, 1988.

[77] P. Thompson and A. W. Toga. Detection, visualization and animation of abnormal anatomic structure with a probabilistic brain atlas based on random vector field transformations. 1:271–294, 1997.

[78] R. Urquhart. Graph theoretic clustering based on limited neighborhood sets. 15:173–187, 1982.

[79] G. Valli, R. Poli, C. Bigozzi, S. Cagnoni, and G. Coppini. Artificial Newral Networks for the Segmentation of Medical Images. Technical report, McGill University, Toronto, Ontario, Canada, July 1994.

[80] D. L. Vilarino, V. M. Brea, D. Cabello, and J. M. Pardo. Discrete-time cnn for image segmentation by active contours. *Patt. Rec. Let.*, 19:721–734, 1998.

[81] C. von der Malsburg. The correlation theory of brain function. Technical report, Max-Planck-Institute Biophysical Chemestry, 1981.

[82] D. Wang and D. Terman. Locally excitatory globally inhibitory oscillator networks. *IEEE Transactions on Neural Networks*, 6(1):283–286, Jan. 1995.

[83] D. L. Wang and D. Terman. Image segmentation based on oscillatory correlation. *Neural Computing*, 9:805–836, 1997.

[84] J. S. Weszka. A Survey of Thresholding Techniques. *Computer Graphics, and Image Processing*, 7:259–265, 1978.

[85] R. Whitaker. Volumetric deformable models. In *Proc. Third Conf. on Visualization in Biomedical Computing*. SPIE, 1994.

[86] R. T. Whitaker. Algorithms for implicit deformable models. In *Fifth International Conference on Computer Vision*. IEEE Computer Society Press, 1995.

[87] R. T. Whitaker and D. T. Chen. Embedded active surfaces for volume visualization. *SPIE Medical Imaging*, 1994.

[88] Z. Wu and R. Leahy. An optimal graph theoretic appraoch to data clustering: Theory and its application to image segmentation. 15:1101–1113, Nov. 1993.

[89] C. T. Zahn. Graph theoretic methods for detecting and describing gestalt clusters. 20:68–86, 1971.

[90] S. W. Zucker and R. A. Hummel. An Optimal Three-Dimensional Edge Operator. Technical report, McGill University, Toronto, Ontario, Canada, Apr. 1979.

[91] S. W. Zucker and R. A. Hummel. A Three-Dimensional Edge Operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 324–331, 1981.