

# **Automatic Segmentation and 3D Rendering of Liver Veins from Ultrasonic Images**

by

Armen A. Balian

A thesis

Submitted in Partial Fulfillment of  
the Requirements for the Degree of  
Master of Science in Computer Science

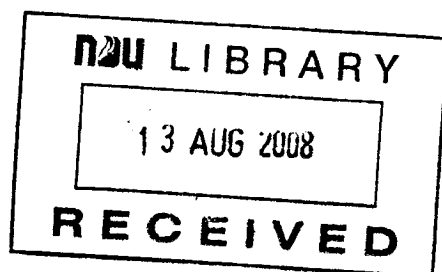
Department of Computer Science

Faculty of Natural and Applied Sciences

Notre Dame University – Louaize

Zouk Mosbeh, Lebanon

January 1999



# **Automatic Segmentation and 3D Rendering of Liver Veins from Ultrasonic Images**

by

Armen A. Balian

Approved:

---

Khaldoun El-Khaldi: Assistant professor of Computer Science.  
Advisor.

---

Fouad Chedid: Associate professor of Computer Science and Chairperson.  
Member of Committee.

---

Mary Khair: Assistant professor of Computer Science.  
Member of Committee.

---

Nasser Saad: Assistant professor of Mathematics.  
Member of Committee.

Date of thesis defense: Wednesday, February 3, 1999

Copyright © 1999  
By  
Armen A. Balian

# TABLE OF CONTENTS

TABLE OF CONTENTS.....	iv
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
ACKNOWLEDGMENT.....	vii
ABSTRACT.....	viii
CHAPTER 1 INTRODUCTION.....	1
1.1 OVERVIEW.....	1
1.2 UNDERSTANDING LIVER ULTRASOUND IMAGES.....	2
CHAPTER 2 AUTOMATIC SEGMENTATION.....	7
2.1 INTRODUCTION.....	7
2.2 FILTERING.....	8
2.3 TEXTURE ANALYSIS.....	12
2.4 CLUSTERING.....	17
2.5 LABELING.....	19
CHAPTER 3 3D VOLUME RENDERING.....	22
3.1 INTRODUCTION.....	22
3.2 RAY CASTING.....	23
CHAPTER 4 EXPERIENCE WORK.....	24
4.1 DATA ACQUISITION.....	24
4.2 DEVELOPMENT ENVIRONMENT.....	24
4.3 AUTOMATIC SEGMENTATION.....	25
4.4 SEGMENTATION PROCEDURE.....	26
4.5 MEDIAN FILTERING TECHNIQUE.....	28
4.6 THE PROPOSED TEXTURE ANALYSIS TECHNIQUE.....	28
4.7 HIERARCHICAL CLUSTERING TECHNIQUE.....	31
4.8 LABELING TECHNIQUE.....	32
4.9 VOLUME RENDERING.....	33
CHAPTER 5 CONCLUSION.....	46
5.1 INTRODUCTION.....	46
5.2 ACCOMPLISHMENTS.....	46
BIBLIOGRAPHY.....	48

## LIST OF TABLES

Table 2.1: Examples of Spatial Filters of 3x3 window.....	11
Table 2.2: Different Classes of Methods .....	12
Table 2.3: Different Texture Features.....	13

## LIST OF FIGURES

Figure 1.1: Anterior and Posterior view of liver.....	3
Figure 1.2: Shaded-Surface 3D reconstruction of the liver .....	3
Figure 1.3: Amplitude and Brightness Ultrasound Display Modes .....	5
Figure 2.1: Original ultrasound image of a liver showing veins.....	9
Figure 2.2: Results of 5 x5 Median Filtering.....	10
Figure 2.3: Results of 3x3 Median Filtering.....	10
Figure 2.4: Labeling Technique.....	19
Figure 4.1: Segmentation Procedure.....	26
Figure 4.2: Original Uncompressed Ultrasound Image showing veins and the gallbladder. ..	27
Figure 4.3: Quantized Histograms .....	32
Figure 4.4: A ray R cast into a scalar function of three spatial variables .....	33
Figure 4.5: Additive reprojection using a lighting model.....	35
Figure 4.6: Combining the shade and the opacity into single value.....	37
Figure 4.7: Constructing the Pyramid.....	39
Figure 4.8: A ray traversing a hierarchical enumeration .....	41
Figure 4.9: Original Cropped Slice.....	44
Figure 4.10: Median Filtering 3x3.....	44
Figure 4.11: Segmented & Clustered.....	45
Figure 4.12: Labeled .....	45
Figure 4.13: 3D Transformation .....	45

## **ACKNOWLEDGMENT**

First, I would like to thank my advisor Dr. Khaldoun El-Khalidy for his consistent support as well as Dr. Fouad Chedid for his advises that kept me on the right path.

Second, I would like to thank Mr. John Mourra from IBM, Toronto for his help in providing me, through the Internet, the needed papers.

Third, I would like to thank Dr. Pierre Haibeh and Dr. Jean Abi Ghosn from Centre Hospitalisé Universitaire de Baabda for providing me with the ultrasound examples.

Finally, I would like to express my deep feelings for the ultimate support of my family, my wife Sylva, My daughter Cyrelle, and my very yet expected child.

## ABSTRACT

In this thesis, automatic segmentation of liver veins from ultrasound slices is implemented and further rendered into 3D image to assist the medical doctor to more accurately apply any study intended on the liver. The main part of this study is the texture analysis since it is the hardest problem in common.

As a start, 4 different techniques in texture analysis have been introduced, the Spatial Gray-level Dependence Matrices (SGLDM), The Fourier Power Spectrum (FPS), The Gray-Level Difference Statistics (GLD), and Law's Texture Energy Measure (TEM). Each of these methods have been discussed in detail in the thesis to show its power and weakness points. From the viewpoint of speed and accuracy of classification, it was found that these features do not perform well enough, either consuming much time or producing low classification rate.

The Multi-resolution Fractal Features is also explained and discussed to show its ability to read important measurement in ultrasound, which are granularity, regularity, and roughness. This method is based upon the concepts of Multiple Resolution Imagery and Fractional Brownian Motion and it was adopted in this research.

Clustering methods (Hierarchical and non-hierarchical) have been introduced and explained, the Vector Quantizer method which is a member of the hierarchical family, is recommended for its ability to simplify the data to be clustered without losing essential information.

Then, labeling techniques which is known for its simplicity and speed, P and 4-connected neighbors, and P and 8-connected neighbors, are also introduced and discussed.

Finally, the segmented veins in each slice are rendered to be displayed in 3D form using Ray Casting technique.



A real time implementation of the proposed algorithms is performed on a Pentium II 400-MHz, 64 Mb RAM, and 4 Mb SVGA Card, with an acceptable processing time which makes the proposed techniques acceptable from all aspects.

**Keywords:** Spatial Gray-level Dependence Matrices, Fourier Power Spectrum, Gray-level Difference Statistics, Law's Texture Energy Measure, Multi-resolution Fractal Features, Multiple Resolution Imagery, Fractional Brownian Motion, Vector Quantizing, Ray Tracing, Automatic Segmentation, 3D Rendering, Ultrasound, Liver Anatomy, Texture Analysis.

# **CHAPTER 1      INTRODUCTION**

## **1.1      OVERVIEW**

The use of ultrasonography as an imaging technique has become widely spread because of its ability to visualize main organs with no side effects (i.e. no radiations). Liver is one of the main human organs that is diagnosed using the ultrasound machine and the sonographer has to do manual segmentation based on the ultrasound slices to detect the unessential part for the doctor to remove. But, the sonographer can undergo many factors that will make him objective when doing the manual segmentation and more the ultrasound slices do not show enough details for the sonographer to be accurate, and needless to say the time for the sonographer will use to do the manual segmentation. So, the surgical challenge is to resect enough liver to allow complete tumor resection while retaining all possible non-tumorous liver to prevent further loss of liver function with a minimum time and high accuracy.

Computerized Segmentation of medical images is a challenging problem in the field of image analysis. Several diagnostics are based on proper segmentation of the digitized image. Segmentation of medical images is needed for applications involving estimation of the boundary of an object, classification of tissue abnormalities, shape analysis, contour detection and texture segmentation. Despite the existence of several techniques, segmentation of specific medical images still remains a crucial problem due to the complex nature of most medical images and especially the ultrasound images.

Because liver vasculature is highly variable, it may be useful to have a 3D model of the liver and vasculature for patients being considered for hepatic surgery. If the liver resection is needed for removal of a tumor(s), it is important to know the tumor(s) location in relation to the hepatic vessels. This thesis will be using the Couinaud classification of liver

anatomy.

In the past, liver resections were made according to what could be told about liver segment anatomy from the surface view. This has several drawbacks. First, anatomic variations may produce errors in identifying the vascular inflow, outflow, and biliary drainage of individual liver segments, thus opening the possibility of leaving non-viable tissues. Second, exact location of tumor relative to the vessels that outline the segments is difficult to determine from surface views and often precise resections cannot be done (i.e. more liver is taken than is necessary).

As a result, automatic segmentation is very important since the computer is subjective and can detect related and important information even from weak ultrasound slices.

## 1.2 UNDERSTANDING LIVER ULTRASOUND IMAGES

### 1.2.1 LIVER SYSTEM

The Couinaud classification divides the liver into 8 independent segments, each of which has its own vascular inflow, outflow, and biliary drainage. Because of this division into self-contained units, each can be resected without damaging those remaining. For the liver to remain viable, resections must proceed along the vessels that define the peripheries of these segments. In general, this means resection lines parallel to the hepatic veins while preserving the portal veins, bile ducts, and hepatic arteries that provide vascular inflow and biliary drainage through the center of the segment. [22]

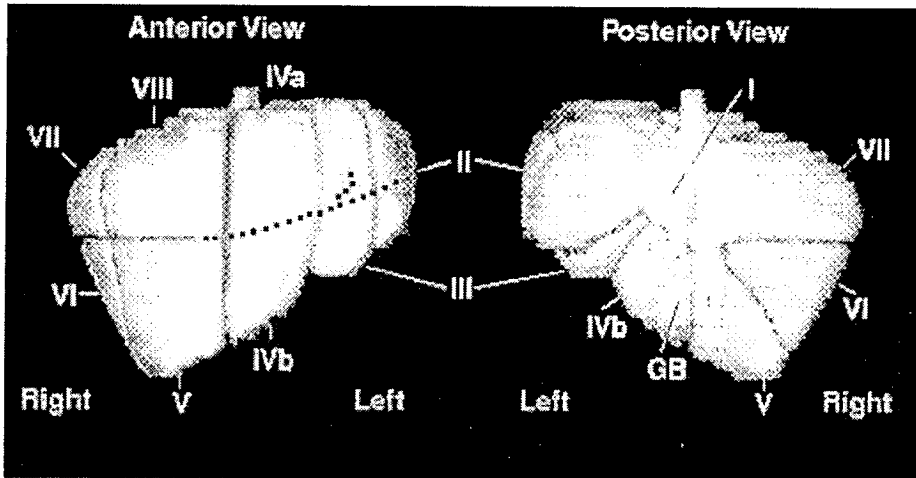


Figure 1.1: Anterior and Posterior view of liver showing 3-dimensional reconstruction of helical CT scan data in shaded surface projections which have been segmented according to the Couinaud classification (dotted line represents the course of the portal vein which is sometimes used to divide segment IV into segment IVa and IVb)

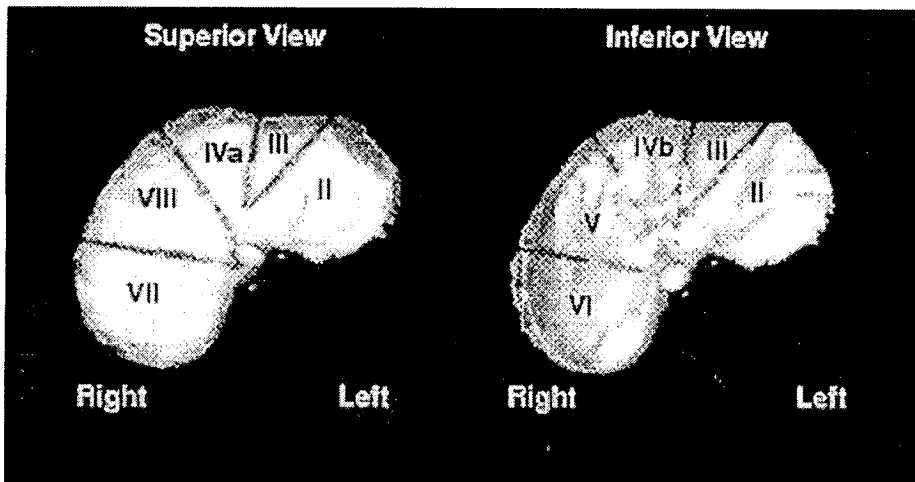


Figure 1.2: Shaded-Surface 3D reconstruction of the liver segments viewed in the transverse plane at the level of the rostral part of the liver and inferiorly from the caudal surface.

- I) Caudate/Spigel lobe
- II) Left posterolateral segment
- III) Left anterolateral segment
- IVa) Left superomedial segment
- IVb) Left inferomedial segment
- V) Right anteroinferior segment
- VI) Right posteroinferior segment
- VII) Right posterosuperior segment
- VIII) Right anterosuperior segment

## 1.2.2 ULTRASOUND CHARACTERISTICS

Ultrasound is a technique by which sound waves are bounced into the body, and their reflections captured by a machine which transform them into an image we can read. Ultrasound does not involve radiation and doesn't use solutions need to be injected or swallowed. The sound waves are produced and received by a small "transducer" that resembles a handy scanner connected by a cable to the machine produces the sound waves and receives their echoes. Ultrasound is an excellent tool for examining most of the human parts because it is "noninvasive" (not traumatic to the body) and most of the times it can reveal more than conventional X-ray studies.

The information received from an ultrasound transducer can be visualized using different display modes. The simplest display mode is called the amplitude, or A-mode, display. In this mode, the amplitude of each echo is plotted against the time it is received. This type of display can be very difficult to interpret since it only presents a single dimension of information. The brightness, or B-mode, display is more intuitive since it presents a two-dimensional display that effectively represents a "slice" of the liver or the tested membrane. This display is constructed by presenting the echo amplitudes as brightness levels along a line, and then accumulating lines from spatially adjacent ultrasound pulses to form a picture. B-mode displays can be produced in real-time by synchronizing the generation of the image with the automated movement of a transducer. One method for automating this movement is to swivel or rotate the transducer on a pivot. This rotation allows a transducer to read a large area from a fixed point. This is known as sector scanning. Transducers such as this can be easily placed into convenient, hand-held probes.

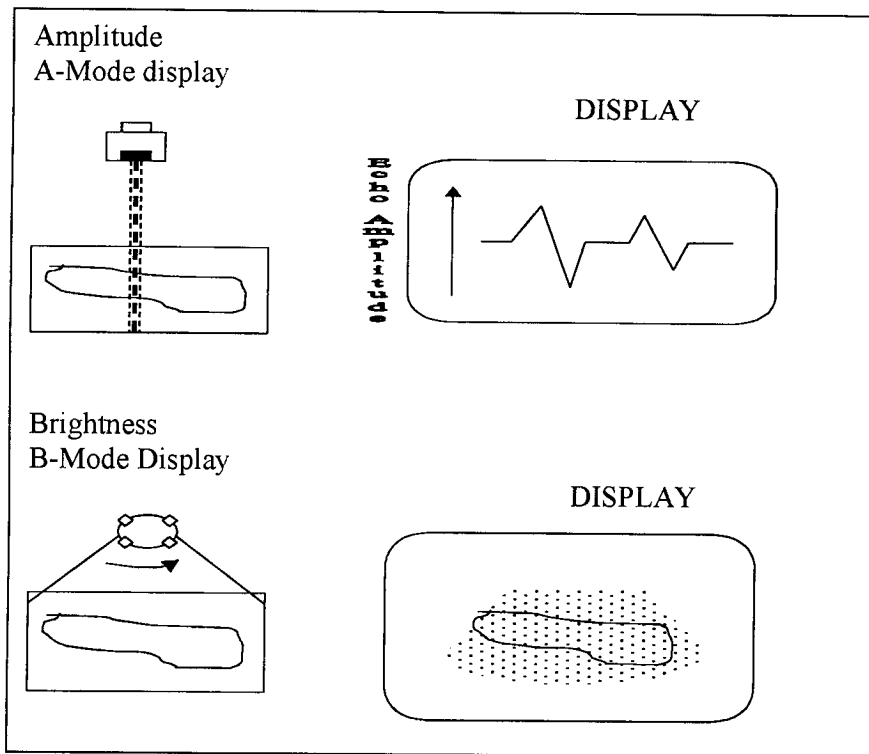


Figure 1.3: Amplitude and Brightness Ultrasound Display Modes

The nature of ultrasound makes these images subject to a large number of image abnormalities. Dropouts which appear as scattered dark spots on the image are one problem. Dropouts occur at interfaces where the ultrasound pulse increases speed upon entering the second medium, resulting in the total internal reflection of the pulse. Dropouts can also occur at material interfaces that are nearly perpendicular to the incident ultrasonic pulse. The amount of the pulse reflected back to the transducer in such cases is very small, resulting in an edge discontinuity in the interface boundary shown on the image.

In addition to dropouts, ultrasound images are also subject to a large amount of background noise. In passing through highly irregular or inhomogeneous materials, the ultrasound pulse may be diffusely scattered. The portions of the scattered pulse that reflect back to the transducer create general background noise consisting of many tiny echoes. More strikingly, the constructive and destructive interference resulting from phase differences in

the various parts of the scattered pulse will produce graininess on the image known as *speckle*. [5,6,22]

### 1.2.3 INTERPRETING A LIVER ULTRASOUND IMAGE

The understanding of an ultrasound image is very important for one to know how to interpret it. The most important thing to remember about the generation of a medical ultrasound image is that the image only indicates interface points between materials of differing acoustical impedance. A region consisting of different materials those are irregularly arranged will produce many echoes and will thus appear brighter. A region consisting uniformly of a single material will produce no echoes, and will thus appear darker. [21,22]



## **CHAPTER 2      AUTOMATIC SEGMENTATION**

### **2.1      INTRODUCTION**

Segmentation refers to the process of decomposing an image into regions of one subject under study. Each of these regions can be described as a region of pixels in which all the pixels share some common properties or characteristics. For this problem, the goal of the segmentation procedure is to identify the regions of the image whose pixels, or best-known texture, can best describe a vein.

Currently, most useful issue differentiation techniques are based on the analysis of a B-scan image of the liver. The radio frequency (RF) signal produced by the transducer when it transforms the ultrasonic echoes into an electrical waveform requires only minimal processing, such as envelope detection, in order to produce an image. More extensive processing and filtering are used to further enhance image quality. This process of manipulating the RF signal results in a loss of phase information and, consequently, a loss of much of the information regarding tissue signature. Thus, some researchers characterize liver disease using the raw RF signal. Although the use of raw RF data is attractive and has many advantages, it also has major drawbacks. At first, RF data must be sampled at a very rapid rate (usually in excess of 20 MHz). The amount of data stored and processed when using RF signals is thus much larger than that required when using image data. Furthermore, nonstandard equipment configurations may be needed. [22]

## 2.2 FILTERING

### 2.2.1 FREQUENCY FILTERING

The high frequency components of an image will typically correspond to noise and edge information in the spatial domain. By removing the high frequency components from an image, it is therefore possible to remove the noise. This removal can be easily accomplished in the frequency domain by multiplying the frequency representation of the image with a low frequency mask.

Unfortunately, since the low-pass filter removes edges information, the image will also be blurred. Friedland *et al.* argue that this blurring effect is not significant for ultrasound images because the amount of edge information left in the lower frequencies after low-pass filtering is still sufficient to define the edge boundaries. The major disadvantage of direct frequency filtering, then, is that it requires a Fourier transform to generate the frequency domain representation of the image, and then an inverse Fourier transform to return the filtered image to the spatial domain. These costly transformations make low-pass frequency filtering an undesired solution for this problem. [8]

### 2.2.2 SPATIAL FILTERING

Spatial filtering appears to be a more desirable solution than frequency filtering because no transformations are required; a filter is instead applied directly to the image in the spatial domain using a convolution operation. In general, this convolution will be faster than transforming the image to the frequency domain, multiplying the image by a filter, and transforming back. If a Fast Fourier Transform (FFT) is used, the frequency-based approach can outperform spatial filtering. However, this will only occur when the spatial filter size is very large. Typical filter sizes of 3x3 and 5x5 are still small enough for spatial filtering to be

faster. Spatial filters often perform an averaging operation. With a *local mean* filter, each pixel is replaced by the average of the values of its neighbors. This method is very effective at smoothing over uniformly distributed noise, although it does show a slightly stronger edge-blurring effect than a direct frequency filter. While a local mean filter performs well, another common spatial filtering method, *median filtering*, has better properties. This method is no more complicated than a local mean filter; each pixel is simply replaced by the median value of its neighbors instead of their average. This filter is excellent because it has a reduced edge blurring effect and also removes the speckle noise, which is a common problem in ultrasound images. Most importantly, a median filter can be implemented to run very quickly. So the median filter was considered the best choice for removing noise from ultrasound images. [4,22]

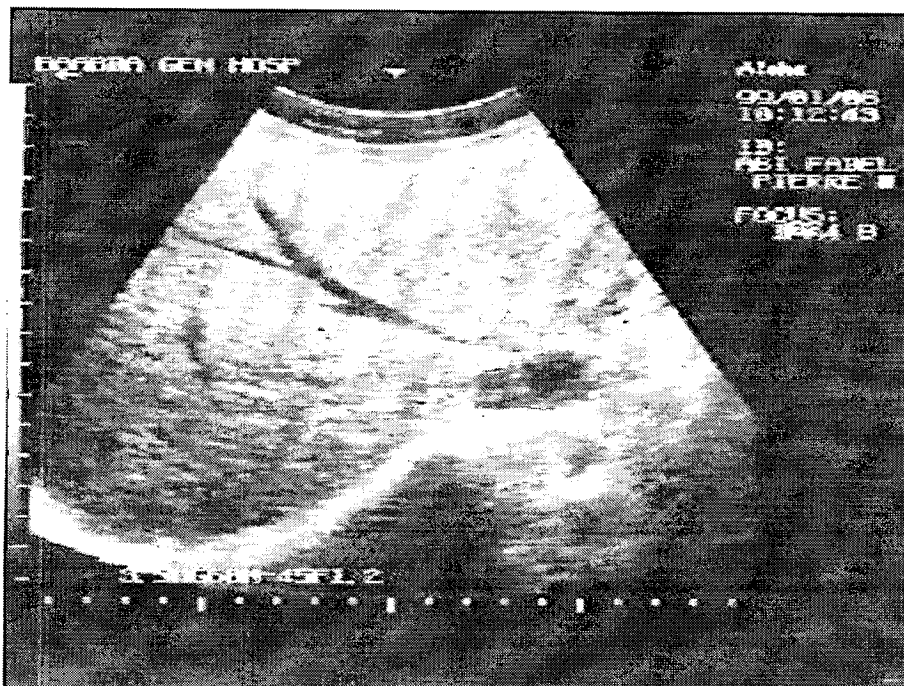
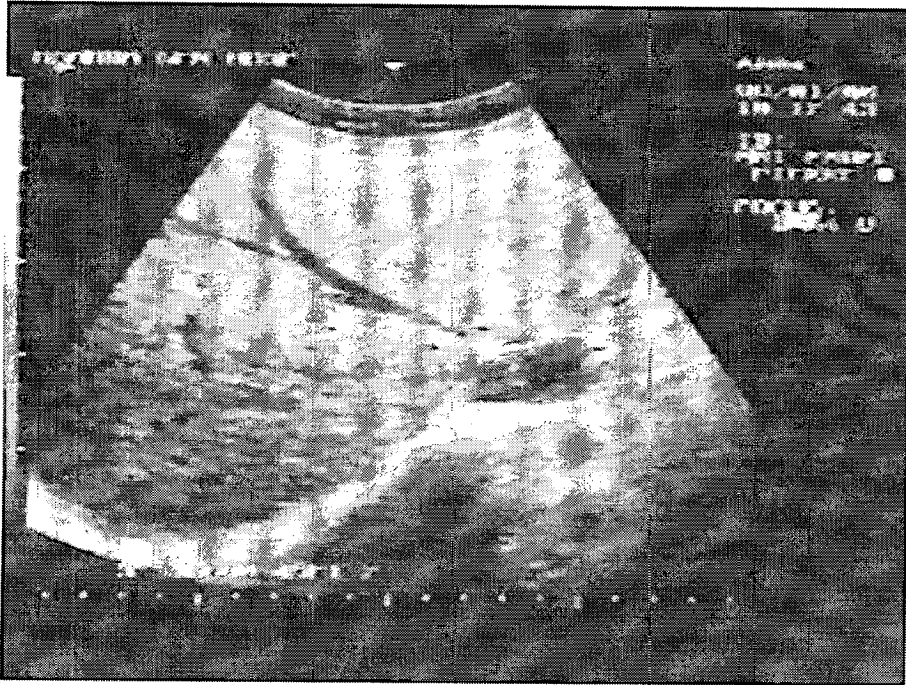
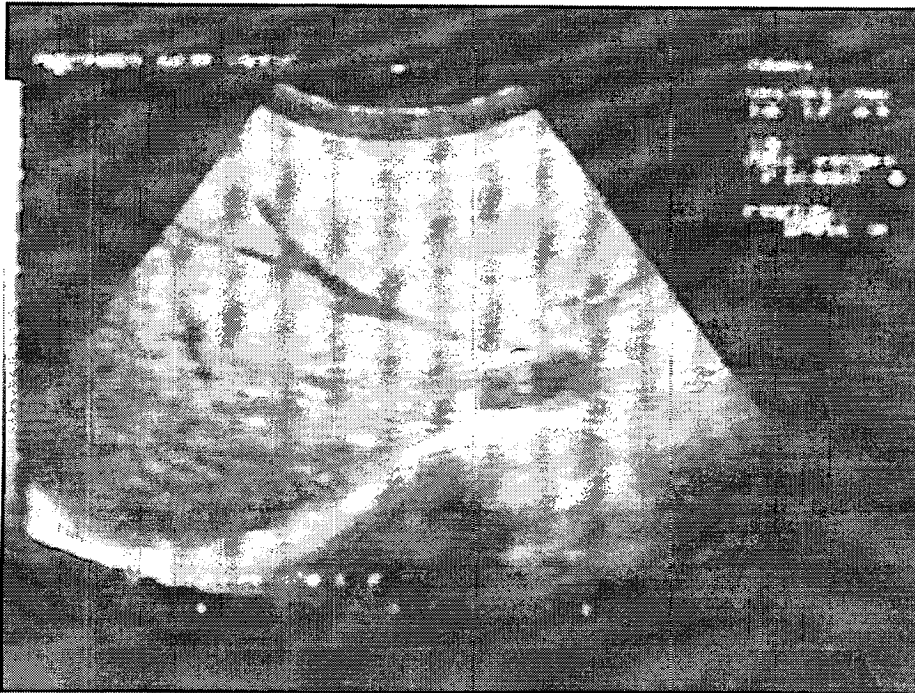


Figure 2.1: Original ultrasound image of a liver showing veins.



*Figure 2.3: Results of 3x3 Median Filtering.*



*Figure 2.2: Results of 5 x5 Median Filtering.*

Other spatial filters of 3x3 window are available and introduced in the following

table:

SPATIAL FILTERS	3 X 3 OPERATORS	EFFECTS
Sobel	$ A + B $ or $\sqrt{A^2+B^2}$ where, $A = \begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ And $B = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	Gradient (finite differences)
Preneit	$ A + B $ or $\sqrt{A^2+B^2}$ where, $A = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ And $B = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	Gradient (finite differences)
Laplacian	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ Or $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	Differential
Smoothing	$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$ Or $\begin{bmatrix} 0 & 1/5 & 0 \\ 1/5 & 1/5 & 1/5 \\ 0 & 1/5 & 0 \end{bmatrix}$	Smoothing
High-pass	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ Or $\begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix}$	Edge enhancement
Sharpening	$\begin{bmatrix} 1/9 & -8/9 & 1/9 \\ -8/9 & 37/9 & -8/9 \\ 1/9 & -8/9 & 1/9 \end{bmatrix}$	Clear Image

Table 2.1: Examples of Spatial Filters of 3x3 window

### 2.3 TEXTURE ANALYSIS

Since the start of analyzing images each researcher gave a different meaning for texture and many were not able to define it but simply we can say that texture is a combination of repeated patterns with a regular frequency. In natural textures, like forest and sea waves, we can define a picture, using the eye, as smooth, coarse, fine etc., but medical imaging, specifically ultrasound, cannot be treated the same. Ultrasonic echoes from soft tissues, when displayed as B-scan image, form a texture pattern that is characteristic of both the imaging system and the tissue being imaged . Many methods where introduced and each embed one or many texture features. Table 2.2 shows different methods [9,14,15,27]

METHOD NAME
- Statistical Methods
- Structural Methods
- Spectral Methods
- Model Based Methods
- Multiscale Methods

Table 2.2: Different Classes of Methods

And as mentioned earlier, a number of approaches to the texture classification problem have been developed over the years some are stated in table 2.3.

TEXTURE FEATURE NAME
- The Spatial Gray-Level Dependence Matrices (SGLDM)
- The Fourier Power Spectrum (FPS)
- The Gray-Level Difference Statistics (GLD)
- Law's Texture Energy Measure (TEM)

- Fractal Dimension (FD)
- Multi-resolution Fractal Features (MF)
- Statistical Geometrical Features (SGF)
- Statistical Features Matrix (SFM)
- Gaussian Markov Random Field (GMRF)
- Gabor Wavelet (GW)
- Run-Length Matrix
- Deformable Models (DM)
- Gabor Convolution Energies (GCE)
- Multi-resolution Wavelet Decomposition (MWD)

Table 2.3: Different Texture Features.

The most commonly used are the Spatial Gray-Level Dependence Matrices (SGLDM), the Fourier Power Spectrum (FPS), the Gray-Level Difference Statistics (GLDS), and the Laws' Texture Energy Measures (LEM). The definitions of these features are summarized in the following sections. The stated features were evaluated using the Bayesian classifier and the Hotelling trace criterion to determine their power and some properties of various feature spaces. From experimental results it can be found that these features can catch only the physical features but not the most important feature which is granularity. So, a more suitable feature was used to describe the property of granularity that is needed for classification of liver tissues in ultrasound slices that is Multi-resolution Fractal Features (MF). [2,10,20]

### 2.3.1 THE SPATIAL GRAY-LEVEL DEPENDENCE MATRICES

Also called Gray-Level Co-occurrence Matrix. As the name indicates, the GLCM is constructed from the image by estimating the pairwise statistics of pixel intensity. Each element  $(i,j)$  of the matrix represents an estimate of the probability that two pixels with a specified separation have gray levels  $i$  and  $j$ . The separation is usually specified by a

displacement  $d$ , and an angle  $\theta$ . That is, GLCM denoted by  $\Phi(d,\theta) = [f(i,j|d,\theta)]$

$\Phi(d,\theta)$ : a square matrix, usually non-symmetric, of side equal to the number of gray levels in the image. Symmetry is often introduced effectively by adding the GLCM to its transpose and dividing every element by 2. This renders  $\Phi(d,\theta)$  and  $\Phi(d,\theta+180^\circ)$  identical and makes the GLCM unable to detect  $180^\circ$  rotations.

In texture classification, individual elements of the GLCM are rarely used. Instead, features are derived from the matrix. A large number of textural features have been proposed starting with the original fourteen features described by Haralick et al., however only some of these are widely used. The features stated above are defined as:

$$Energy = \sum_{i,j} \hat{f}(i,j)^2$$

$$Entropy = \sum_{i,j} \hat{f}(i,j) \log \hat{f}(i,j)$$

$$Correlation = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) \hat{f}(i,j)}{\sigma_i \sigma_j}$$

$$Inverse\ Difference\ Moment = \sum_{i,j} \frac{1}{1 + (i - j)^2} \hat{f}(i,j)$$

$$Inertia = \sum_{i,j} (i - j)^2 \hat{f}(i,j)$$

$$Cluster\ shade = \sum_{i,j} ((i - \mu_i) + (j - \mu_j))^3 \hat{f}(i,j)$$

$$Cluster\ Pro\ min\ ence = \sum_{i,j} ((i - \mu_i) + (j - \mu_j))^4 \hat{f}(i,j)$$

$$Where\ \mu_i = \sum_i i \sum_j \hat{f}(i,j)$$

$$\mu_j = \sum_j j \sum_i \hat{f}(i,j)$$



$$\sigma_i = \sum_i (i - \mu_i)^2 \sum_j \hat{f}(i, j)$$

$$\sigma_j = \sum_j (j - \mu_j)^2 \sum_i \hat{f}(i, j)$$

$$\text{Haralick's Correlation} = \frac{\sum_{i,j} (i, j) \hat{f}(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

Where  $\mu_x$ ,  $\mu_y$ ,  $\sigma_x$  and  $\sigma_y$  are the means and standard deviations of row and column sums, respectively. Note that for any choice of  $d$  and  $\theta$  we obtain a separate GLCM. The results may be combined by averaging the GLCM for each angle before calculating the features or by averaging the features calculated for each GLCM. [1,6]

### 2.3.2 THE FOURIER POWER SPECTRUM

To use the Fourier Power spectrum (FPS), one must first compute the sample power spectrum,

$$\phi(u, v) = F(u, v)F^*(u, v) = |F(u, v)|^2$$

where  $\phi$  is the sample power spectrum,  $*$  denotes the complex conjugate, and  $F$  stands for the Fourier transform of the image. Coarse texture will have high values of  $|F|^2$  concentrated near the origin, while in fine texture the values will be more spread out. Similarly, texture with many edges or lines in a given direction  $\theta$  will have high values of  $|F|^2$  concentrated around the perpendicular direction  $\theta + \frac{\pi}{2}$ , while in homogeneous texture,  $|F|^2$  should be nondirectional.

Annular-ring and wedge sampling geometry are commonly used for FPS features, which are defined as follows:

$$\phi_{r_1, r_2} \equiv \sum_{r_1^2 \leq u^2 + v^2 < r_2^2} |F(u, v)|^2$$

$$\phi_{\theta_1, \theta_2} \equiv \sum_{\theta_1 \leq \tan^{-1}(v, u) < \theta_2} |F(u, v)|^2$$

where  $0 < u, v < M-1$  for a given  $M \times M$  image.

The ring  $\phi_{r_1, r_2}$  and wedge  $\phi_{\theta_1, \theta_2}$  applied to the classification of ultrasonic liver images are  $\phi_{2,4}$  and  $\phi_{0^\circ, 45^\circ}$ , since better results can be obtained than any other combination of rings and wedges. [6,22]

### 2.3.3 THE GRAY-LEVEL DIFFERENCE STATISTICS

To describe the gray-level difference statistics (GLD), let  $I(x, y)$  be the image intensity function. For any given displacement  $\delta \equiv (\Delta x, \Delta y)$ , let  $I_\delta(x, y) \equiv |I(x, y) - I(x + \Delta x, y + \Delta y)|$  and  $p_\delta$  be the probability density of  $I_\delta(x, y)$ . If there are  $m$  gray levels, this has the form of an  $m$ -dimensional vector whose  $i$ th component is the probability that  $I_\delta(x, y)$  will have value  $i$ . It is easy to compute  $p_\delta$  by counting the number of times each value of  $I_\delta(x, y)$  occurs where  $\Delta x$  and  $\Delta y$  are integers.

If texture is coarse, and  $\delta$  is small compared to the texture element size, the pairs of points at separation  $\delta$  should usually have similar gray levels, so that  $I_\delta(x, y)$  should usually be small, i.e. the values in  $p_\delta$  should be concentrated near  $i = 0$ . On the other side, for fine texture, the values in  $p_\delta$  should be more spread out. Thus, a good way to analyze texture coarseness would be to compute, for various magnitudes of  $\delta$ , some measure of the spread of values in  $p_\delta$  away from the origin. The feature used to classify the tree sets of ultrasonic liver

images is the entropy that is defined by [1,4]

$$ENT \equiv - \sum_i p_\delta(i) \log p_\delta(i)$$

### 2.3.4 LAWS' TEXTURE ENERGY MEASURES

Law's Texture Energy Measures (TEM), are derived from three simple vectors of size 3,  $L3 \equiv (1,2,1)$ ,  $E3 \equiv (-1,0,1)$ , and  $S3 \equiv (-1,2,-1)$ , which represent the one dimensional operations of center-weighted local averaging symmetric first differencing for edge detection, and second differencing for spot detection. If these vectors are convolved with themselves or with each other, we obtain five vectors of size 5,  $L5 \equiv (1,4,6,4,1)$ ,  $S5 \equiv (-1,0,2,0,-1)$ ,  $R5 \equiv (1,-4,6,-4,1)$ ,  $E5 \equiv (-1,-2,0,2,1)$ , and  $W5 \equiv (-1,2,0,-2,1)$  where L5 again performs local averaging, S5 and E5 are, spot and edge detectors respectively, and finally R5 and W5 can be regarded as "ripple" and "wave" detectors. If we multiply the column vectors of length 5 by row vectors of the same length, we obtain Laws' 5x5 masks. To use these masks to describe texture in an image, we convolve them with the image and use statistics (e.g. energy) of the results as texture properties. The best 2 masks to fit this study are the 2 listed below and they are both zero-sum mask [6,27].

L5'E5

L5'S5

-1	-2	0	2	1	-1	0	2	0	-1
-4	-8	0	8	4	-4	0	8	0	-4
-6	-12	0	12	6	-6	0	12	0	-6
-4	-8	0	8	4	-4	0	8	0	-4
-1	-2	0	2	1	-1	0	2	0	-1

## 2.4 CLUSTERING

Clustering is a grouping of data with similar characteristics. It is divided into

hierarchical clustering and non-hierarchical clustering as mentioned as follows.

#### 2.4.1 HIERARCHICAL CLUSTERING

The similarity of a cluster is evaluated using a distance measure. The minimum distance between clusters will give a merged cluster after repeated procedures from a starting point of pixel-wise clusters to a final limited number of clusters.

The distances to evaluate the similarity are selected from the following methods.

- Nearest neighbor method: Nearest neighbor with minimum distance will form a new merged cluster.
- Furthest neighbor method: Furthest neighbor with maximum distance will form a new merged cluster.
- Centroid method: Distance between the gravity centers of two clusters is evaluated for merging a new merged cluster.
- Group Average method: Root mean square distance between all pairs of data within two different clusters is used for clustering.
- Ward method: Root mean square distance between all gravity centers and each member is minimized.

#### 2.4.2 NON-HIERARCHICAL CLUSTERING

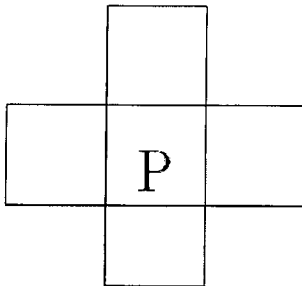
At the initial stage, an arbitrary number of clusters should be temporally chosen. The members belonging to each cluster will be checked by selected parameters or distance and relocated into the more appropriate clusters higher separability. The ISODATA method and K-mean method are examples of non-hierarchical clustering.

The ISODATA method is composed of the following procedures

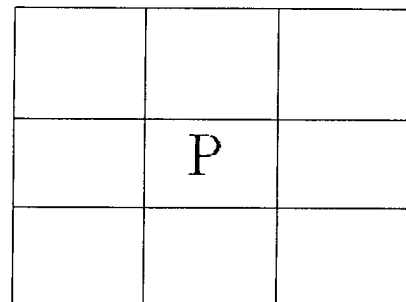
- All members are relocated into the closest clusters by computing the distance between the member and the clusters.
- The center of gravity of all clusters is recalculated and the above procedure is repeated until convergence.
- If the number of clusters is within a certain specified number, and the distances between the clusters meet a prescribed threshold, the clustering is considered complete. [8,19,26]

## 2.5 LABELING

Labeling of a binary image is the operation of assigning a unique value to pixels belonging to the same connected region. Depending on the definition of a “connected region”, different results can be obtained. Two widely used square grid topologies are shown below:



P and 4-connected neighbors



P and 8-connected neighbors

*Figure 2.4: Labeling Technique*

### 2.5.1 P AND 4-CONNECTED NEIGHBORS

Let's assume that we have an image  $N \times N$ , the background pixel is 0, the pixel belongs to an object is 1, and  $p$  is the pixel we're reading from the image,  $r$  and  $t$  are the upper and left-hand neighbors of  $p$ , respectively.

- If the value of  $p$  is 0 so move to the next pixel
- If the value of  $p$  is 1 then, check  $r$  and  $t$
- If they both are 0 then assign a new label to  $p$
- If only one of the two neighbors is 1 then assign its label to  $p$ .
- If they are both 1 and have the same label, then assign the label to  $p$ .
- If they are both 1 and each has a different label then assign one of the labels to  $p$  and make a note that the 2 labels are equivalent (that is point  $r$  and  $t$  are connected through  $p$ ).

At the end of the scan, all points with value 1 have been labeled, but some of these labels may be equivalent. All we need to do now is sort all pairs of equivalent labels into equivalence classes, assign a different label to each class, and then do a second pass through the image, replacing each label by the label assigned to its equivalence class.

### 2.5.2 P AND 8-CONNECTED NEIGHBORS

To label 8-connected components we proceed in the same way, but the upper diagonal neighbors of  $p$ , denoted by  $q$  and  $s$ , also have to be examined. The nature of the scanning sequence ensures that these neighbors have already been processed by the time the procedure gets to  $p$ .

- If  $p$  is 0, move on to the next scanning position.
- If  $p$  is 1 and all four neighbors are 0 then assign a new label to  $p$ .
- If only one of the neighbors is 1 then assign its label to  $p$ .
- If two or more neighbors are 1 then assign one of the labels to  $p$  and make a note of the appropriate equivalencies.

After completing the scan of the image, sort the equivalent label pairs into

equivalence classes, assign a unique label to each class, and do a second scan through the image, replacing each label by the label assigned to its equivalence class. [25]

## **CHAPTER 3      3D VOLUME RENDERING**

### **3.1      INTRODUCTION**

The term volume rendering is used to describe techniques that allow the visualization of three-dimensional data. Currently, the major application area of volume rendering is medical imaging, where volume data is available from X-ray Computer Tomography (CT) scanners and Positron Emission Tomography (PET) scanners. But, all the scanners mentioned earlier hold the risk of radiation which makes creating 3D shape from ultrasound images is much preferred. And more, it only contains information from each slice. A conventional X-ray image, on the other hand, contains information from all the planes, and the result is an accumulation of shadows that are a function of the density of the tissue, bone, organs, etc., anything that absorbs the X-rays. Nevertheless, the availability of the stacks of parallel data produced by ultrasound machine urged the development of techniques for viewing the data as a three-dimensional field rather than as individual planes. This gives the immediate advantage that the information could be viewed from any viewpoint.

There are a number of different methods used:

- Rendering voxels in binary partitioned space
- Marching cubes
- Ray casting

There are some problems with the first two of these methods. In the first method, choices are made for the entire voxel. This can produce a "blocky" image. It also has a lack of dynamic range in the computed surface normals, which produces images with relatively poor shading.

The marching cubes approach solves this problem, but causes some others. Its biggest



disadvantage is that it requires that a binary decision be made on the position of the intermediate surface that is extracted and rendered. Also, extracting an intermediate structure can cause the display of artifacts that do not exist and as well discarding small or poorly defined features. [17]

### 3.2 RAY CASTING

The basic goal of ray casting is to allow the best use of the three-dimensional data and not attempt to impose any geometric structure on it. It solves one of the most important limitations of surface extraction techniques, namely the way in which they display a projection of a thin shell in the acquisition space. Surface extraction techniques fail to take into account that, particularly in medical imaging, data may originate from fluid and other materials which may be partially transparent and should be modeled as such. Ray casting doesn't suffer from this limitation and that is why it was selected and used in this thesis.

[12,13]

## **CHAPTER 4      EXPERIENCE WORK**

### **4.1      DATA ACQUISITION**

The ultrasound machine used in this study was Aloka SSD-500. The machine uses a 3.5 MHz probe to produce real-time B-mode scans. The scans produced have a precision of 6-bits and are displayed using linear gray-scale monomap, with 0 corresponding to black and 64 corresponding to white. The machine is connected to a Sony UP-850 Video Graphic Printer for producing hardcopies of the current ultrasound scan. The machine also has two RF video-out lines with BNC connectors.

For this thesis, A Hitachi DA4 Head VT-M838E (SW) ME S-VHS video cassette recorder was used to record the scanning sessions, an AV cable was used to connect the video recorder to the ultrasound machine. The recordings are made in the standard VHS mode at SP speed using Fuji SHG VHS-C tape and a Panasonic VHS-C NV-VX55EN Movie Camera was used to playback the recordings to check if the recording was complete.

### **4.2      DEVELOPMENT ENVIRONMENT**

A Pentium II running at 400 MHz with 64 Mb. RAM was used for both the development of the automatic segmentation procedure as well as the end-user application. The operating system used on the PC was windows 3.11 Arabic/English.

A Pixel View 5465P+ W/FM card frame grabber was to use to record the scanning session as movie directly from a video-input signal onto the hard disk. Input can be provided to this board via a RF video-in line with a RCA connector. Images captured with this board have a 480x640-pixel resolution with a precision of seven bits. During development, the input video signal was obtained from videotapes of previously performed ultrasound scanning sessions. Playback was done using the same Hitachi DA4 Head video cassette

recorder. A RCA to RCA cable was used to connect the board to the video-out line on the VCR. Once development was completed, the input video signal was obtained directly from the video cassette recorder. As a result, a clip of the scanning session was saved under type AVI on the hard disk.

After transferring the ultrasound session from VCR to PC, a graphic software, Animator Studio v1.1, was used to cut the movie into frames, i.e. slices.

The software developed for this thesis was written using Delphi language v1.0 for the Graphic User Interface and Visual C++ for the task procedures.

### 4.3 AUTOMATIC SEGMENTATION

The development of an automatic procedure for segmenting liver veins from ultrasound images was the hardest part in this thesis. Usually, in order to segment the veins of a liver from ultrasound slices a clinician needed to trace the regions by hand. This manual tracing used to have several drawbacks that the automatic segmentation needed to solve.

First, an experienced clinician was required to perform the manual tracing. To expect reasonable results from the manual tracing, it was important that the tracing be performed by an individual with a strong knowledge of liver structure and an understanding of what features to look for on the ultrasound image. While an automated procedure would still need to be monitored by a trained clinician, the clinician needs not to be that high expert.

Second, manual tracing is subjective. Since the human person follows his knowledge and experience, manual tracing will vary from one to another even with the presence of high equal knowledge and experience among the clinicians. While the automatic procedure follows a sequence of analysis and calculations that are applied systematically on any set of ultrasound slices, which would reduce this variability and improve the results.

Finally, and most importantly, manual tracing is very time consuming. For every measurement, the outline of every needed region on each of about thirteen ultrasound images has to be manually traced and filled. Even with a paint program to assist, it still takes an average of 15 minutes to manually trace an entire sequence. This time, which is one minute per slice as an average, is considered too long.

Due to the drawbacks, and although different approaches were tried and considered, each approach was evaluated against its processing time and its accuracy to detect the needed regions in each slice.

#### 4.4 SEGMENTATION PROCEDURE

The sequential stages of the procedure are illustrated in Figure 4.1. The image preprocessing stage uses two median filters to remove noise from the input ultrasound image.

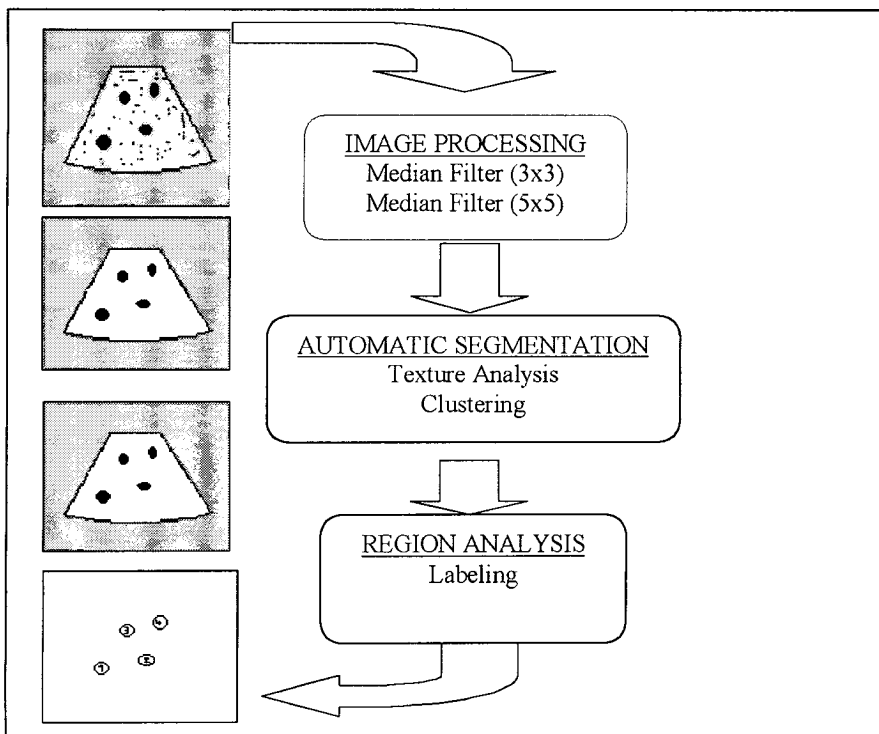
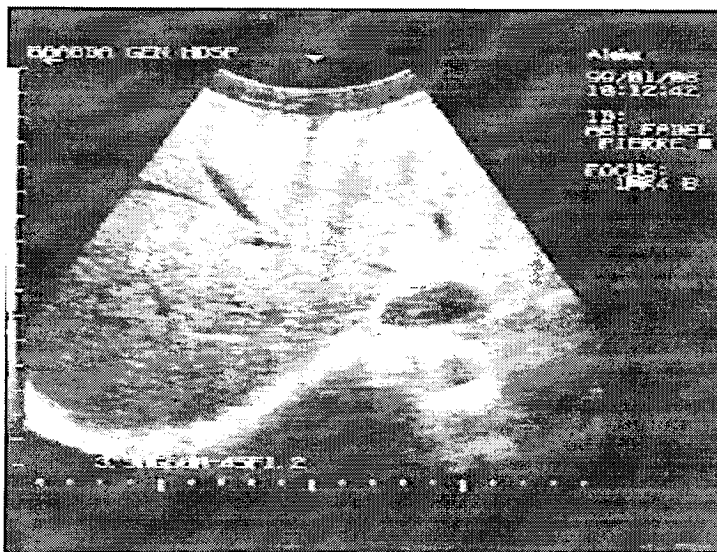


Figure 4.1: Segmentation Procedure

A Texture analysis is applied to partition the data into distinct areas in the slice. A clustering operation then segments the image into several relevant data. Then, a labeling technique is used to label the regions.

The remainder of this section explains the development and functionality of each stage of this procedure in greater detail. An example will also be followed to illustrate each processing step. The ultrasound image used is taken from the example set. It shows the veins and the gallbladder of the liver. The unprocessed image is shown in Figure 4.2.



*Figure 4.2: Original Uncompressed Ultrasound Image showing veins and the gallbladder.*

Since the veins are always located near the middle of the image, the entire image needs not to be processed. A region of interest is therefore extracted so that only the relevant parts of the image are considered. The primary reason for reducing the area to be processed is to minimize the procedure's processing time. A sequence of figures showing results of the processing techniques used can be found on page 44.

## 4.5 MEDIAN FILTERING TECHNIQUE

The purpose of the image preprocessing stage is to remove noise and distortions from the image so that the segmentation can be performed more effectively. As discussed in the first section, dropouts and background noise are the most common problems with ultrasound images. Of these problems, the background noise was seen as the most serious. The primary goal of the preprocessing is therefore to remove, or at least reduce, this noise without destroying any essential information in the image or adding irrelevant information. As a result a median filtering technique was adopted because it belongs to the class of edge preserving smoothing filters which are non-linear filters. It smoothes the data while keeping the small and sharp details, and more, it is a simple and very effective noise removal filtering process.

## 4.6 THE PROPOSED TEXTURE ANALYSIS TECHNIQUE

The information measured by the four conventional features stated earlier are coarseness, regularity, directionality, and energy. One of the most important information, granularity, is not considered by these features. So, to obtain useful information from ultrasonic liver images, a feature extraction technique will be introduced which is based on the concepts of multiple resolution imagery and fractional Brownian motion model.

### 4.6.1 MULTIPLE RESOLUTION FEATURE EXTRACTION

Textural and contextural features are two of the fundamental pattern elements used in human interpretation of visual information. Textural features contain information about the spatial distribution of tonal variations within a band of the visible data, while contextural features contain information derived from blocks of visual data surrounding the area being analyzed. When a machine independently processes small image areas from black and white

photographs, then texture and tone are most important.

Context, texture and tone are always present in an image although at times one property can dominate the other. One way of measuring the three data is to visualize the image at different resolutions. The pyramidal approach is one of the popular ways for multi-resolution image analysis. Based upon the concept of pyramidal data analysis, we can define the image at resolution level  $i$  as

$$I^{(i)}(x, y) = \left[ I^{(i+1)}(2x, 2y) + I^{(i+1)}(2x+1, 2y) + I^{(i+1)}(2x, 2y+1) + I^{(i+1)}(2x+1, 2y+1) \right] / 4$$

$$0 \leq i \leq m, \quad 0 \leq x, y \leq 2^i$$

while  $I^{(m)}$  is the original image. The reason for the simplified definition is to lower computational cost. [2,10]

#### 4.6.2 FRACTIONAL BROWNIAN MOTION MODEL

The fractional Brownian motion (FBM) model was developed by Mandelbrot used to describe the roughness of nature surfaces. It regards naturally occurring surfaces as the end result of random walks. An intensity surface of an ultrasonic image can also be viewed as the end result of a random walk, so the FBM model can be suitable for the analysis of ultrasonic liver images.

One of the most important parameters to represent a fractal surface is the fractal dimension. Theoretically, the fractal dimension  $D_f$  is estimated from the equation

$$E(\Delta I^2) = c(\Delta r)^{(6-2D_f)}$$

where  $E(\cdot)$  denotes the expectation operator,  $\Delta I \equiv I(x_2, y_2) - I(x_1, y_1)$  is the intensity variations,  $c$  is some constant, and  $\Delta r \equiv \|(x_2, y_2) - (x_1, y_1)\|$  is the spatial distance. A simpler

method is to estimate the  $H$  parameter from the relationship

$$E(|\Delta I|) = k(\Delta r)^H$$

where  $k = E(|\Delta I|)_{\Delta r=1}$ . If we apply the log function to both sides, we obtain

$$\log E(|\Delta I|) = \log k + H \log(\Delta r)$$

from it, we can deduce a procedure to estimate the  $H$  parameter and the fractal dimension can be easily computed from the relationship

$$D_f = 3 - H$$

A small value of fractal dimension  $D_f$  (large value of parameter  $H$ ) means a smooth surface, and large  $D$  (small  $H$ ) means a rough surface.

Given an  $M \times M$  image  $I$ , the intensity difference vector is defined as  $IDV \equiv [id(1), id(2), \dots, id(s)]$  where  $s$  is the maximum possible scale and  $id(k)$  is the average of the absolute intensity difference of all pixel pairs with horizontal or vertical distance  $k$ . Value of  $H$  can be obtained by using least-squares linear regression to estimate the slope of the curve of  $id(k)$  versus  $k$  in log-log scales. [3,9,16,20,24]

#### 4.6.3 MULTIREOLUTION FRACTAL FEATURE VECTOR

Based on the concepts described as above, the multiresolution fractal (MF) feature vector is defined as

$$MF \equiv \left( H^{(m)}, H^{(m-1)}, \dots, H^{(m-n+1)} \right)$$

where  $M = 2^m$  is the size of the original image,  $H^{(k)}$  is the  $H$  parameter estimated from image  $I^{(k)}$ , and  $n$  is the number of resolution levels.

The feature vector describes not only the roughness but also the lacunarity of the



image as well as the regularity. [10,11,16]

#### 4.7 HIERARCHICAL CLUSTERING TECHNIQUE

Hierarchical algorithms work by iteratively merging smaller clusters or iteratively splitting larger clusters until the desired number of groups have found. Since clusters are merged or divided according to a simple heuristic method, this algorithm can be performed very quickly. It is for this reason that a hierarchical clustering algorithm was used.

The specific hierarchical clustering algorithm used here is a vector quantizer developed by Scott Wilson. It is approximately based on Aul Heckberts Median Cut algorithm. It begins by representing the feature space as an N-dimensional hyperbox. The box is first shrunk to match the spatial extent of the feature points that it contains and is next split into two boxes. These two boxes are then shrunk to fit the points that they contain and then one of the boxes is selected for splitting. This process continues until a desired number of hyperboxes have been obtained. The decision on which box to split is made by finding the box representing the largest subspace in the 2-norm sense. The box is then bisected by a hyperplane aligned with the axis that is perpendicular to the largest span across the feature space. The division is made at the median point of that span.

The centers of the quantized clusters give a condensed representation of the distribution of the data. The essential spatial distribution of the points in feature space is preserved, but the actual number of point is reduced. Considering that a histogram is simply 1-D feature space of pixel intensity values, this quantization routine can be used to reduce the number of pixel values needed to represent the image while preserving the histogram's relative spread. This has a positive effect in terms of thresholding because a modality is effectively being forced on the histogram without changing the data distribution. This

modality allows a fixed threshold to behave more predictably since the threshold value will act more on the distribution of the pixels rather than on their absolute values. A histogram of an image example before and after quantization is shown in Figure 4.3 [8,16]

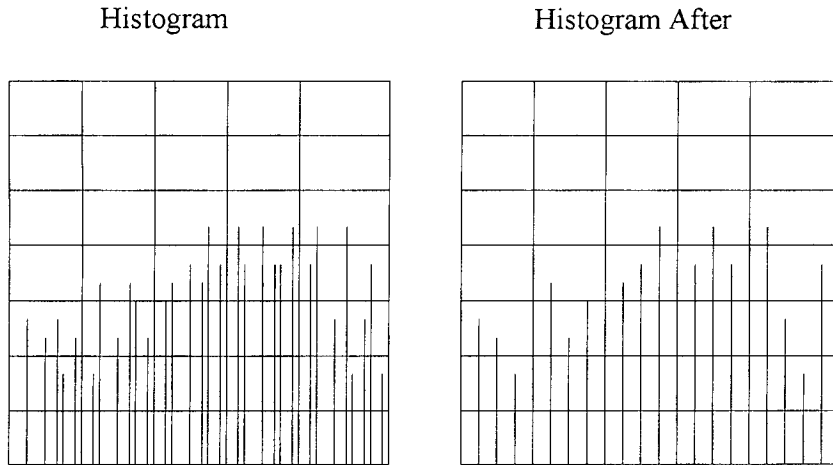


Figure 4.3: Quantized Histograms

#### 4.8 LABELING TECHNIQUE

In order to distinguish between the different regions, it is necessary to distinctly label them. A simple approach is taken for doing this. Each pixel is initially considered not to belong to any region. Starting from the top left pixel and scanning from left to right and top to bottom, each pixel is compared to its adjacent neighbors. If the pixel has the same value as its neighbor, then it is added to its neighbor's region. If it doesn't, then a new region is started. Note that since the image produced by the thresholding is binary, it is sufficient to look at only the values of adjacent pixels in order to determine if they belong in the same region. After each pixel has been considered, the regions are checked and any spatially common regions which still remain disconnected are merged together.

This labeling stage also has the capability of merging smaller regions with spatially adjacent larger regions according to a minimum size threshold. This capability is used to eliminate any very small region that doesn't belong to the criteria. [25]

## 4.9 VOLUME RENDERING

### 4.9.1 RAY TRACING THEORY

Currently, most volume rendering that uses ray casting is based on the Blinn/Kajiya model. In this model we have a volume which has a density  $D(x,y,z)$ , penetrated by a ray  $R$ .

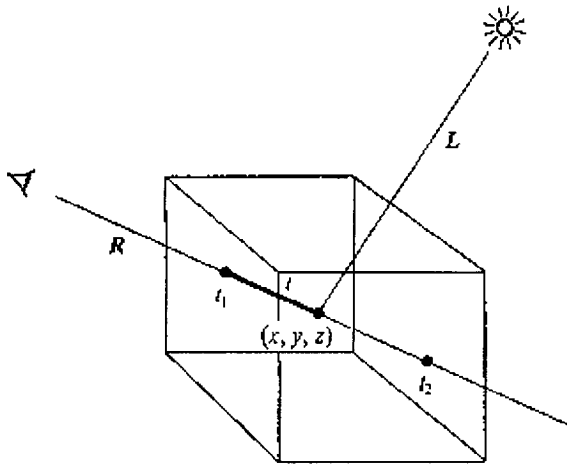


Figure 4.4: A ray  $R$  cast into a scalar function of three spatial variables

At each point along the ray there is an illumination  $I(x,y,z)$  reaching the point  $(x,y,z)$  from the light source(s). The intensity scattered along the ray to the eye depends on this value, a reflection function or phase function  $P$ , and the local density  $D(x,y,z)$ . The dependence on density expresses the fact that a few bright particles will scatter less light in the eye direction than a number of dimmer particles. The density function is parameterized along the ray as:

$$D(x(t), y(t), z(t)) = D(t)$$

and the illumination from the source as:

$$I(x(t), y(t), z(t)) = I(t)$$

and the illumination scattered along  $R$  from a point distance  $t$  along the ray is:

$$I(t)D(t)P(\cos\theta)$$

where  $\theta$  is the angle between R and L, the light vector, from the point of interest.

Determining  $I(t)$  is not trivial, it involves computing how the radiation from the light sources is attenuated and/or shadowed due to its passing through the volume to the point of interest. This calculation is the same as the computation of how the light scattered at point  $(x,y,z)$  is affected in its way along R to the eye. In most algorithms, however, this calculation is ignored and  $I(x,y,z)$  is set to be uniform throughout the volume. For most practical applications, we're interested in visualization and including the line integral from a point  $(x,y,z)$  to the light source may actually be undesirable. In medical imaging, for example, it would be impossible to see into areas surrounded by bone if the bone were considered dense enough to shadow light. On the other hand, in applications where internal shadows are desired, this integral has to be computed.

The attenuation due to the density function along a ray can be calculated as:

$$\exp\left(-r \int_{f_1}^{f_2} D(S) ds\right)$$

where  $r$  is a constant that converts density to attenuation, and the intensity of the light arriving at the eye along direction R due to all the elements along the ray is given by:

$$B = \int_{f_1}^{f_2} \left( \exp\left(-r \int_{f_1}^{f_2} D(S) ds\right) \right) (I(t)D(t)P(\cos\Theta)) dt$$

#### 4.9.2 IMPLEMENTATION

When it pertains to volume visualization, ray casting is often referred to as ray tracing. Strictly speaking this is inaccurate, since the method of ray tracing with which most people are familiar is more complex than ray casting. However, the basic ideas of ray casting are identical to those of ray tracing, and the results are very similar.

Algorithms, which implement the general ray casting technique described above, involve a simplification of the integral which computes the intensity of the light arriving at the eye. The method by which this is done is called "additive reprojection." It essentially projects voxels along a certain viewing direction. Intensities of voxels along parallel viewing rays are projected to provide an intensity in the viewing plane. Voxels of a specified depth can be assigned a maximum opacity, so that the depth that the volume is visualized to can be controlled. This provides several useful features:

- The volume can be visualized from any direction.
- Hidden surface removal can be implemented so that, for example, front ribs can obscure back ribs.
- Color can be used to enhance interpretation.

Additive reprojection uses a lighting model that is a combination of reflected and transmitted light from the voxel. All of the approaches are a subset of the model in the following figure.

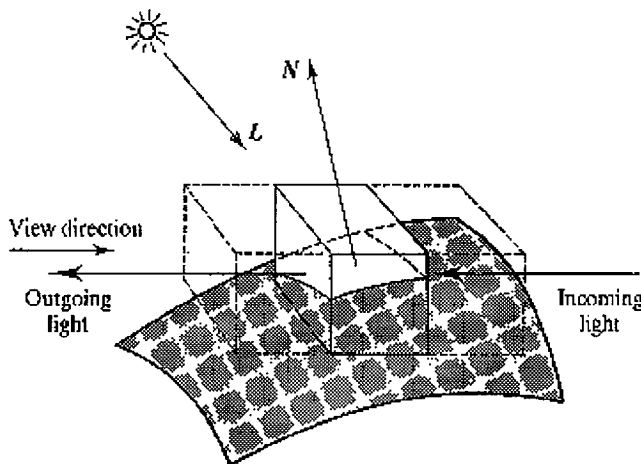


Figure 4.5: Additive reprojection using a lighting model.

In the above figure, the outgoing light is made up of:

- Light reflected in the view direction from the light source
- Incoming light filtered by the voxel
- Any light emitted by the voxel

For every pixel in the output image, a ray is shot into the data volume. At a predetermined number of evenly spaced locations along the ray the intensity and opacity values are obtained by interpolation. The interpolated intensities and opacities are then merged with each other and with the background by compositing in either front-to-back or back-to-front order to yield the color of the pixel. (The order of compositing makes no difference in the final output image.)

This technique consists of two pipelines - a visualization pipeline and a classification pipeline. The output produced by these two pipelines is then combined using volumetric compositing to produce the final image.

In the visualization pipeline the volume data is shaded. Every voxel in the data is given a shade using a local gradient approximation to obtain a "voxel normal." This normal is then used as input into a standard Phong shading model to obtain intensity. The output from this pipeline is a three color component intensity for every voxel in the data set.

Note that in completely homogeneous regions the gradient is zero. The shade returned from such regions is unreliable. To solve this problem, this technique bases the opacity of each voxel on the gradient. The gradient is multiplied by the opacity. Homogeneous regions have an opacity of zero, and thus do not affect the final image.

The classification pipeline associates an opacity with each voxel. The intensity parameters, which are used in the shading model, are also used in this pipeline.

$$\alpha(X) = \begin{cases} \alpha_{V_{n+1}} \left\{ \frac{V(X) - f_{V_n}}{f_{V_{n+1}} - f_{V_n}} \right\} \\ + \alpha_{V_n} \left\{ \frac{f_{V_{n+1}} - V(X)}{f_{V_{n+1}} - f_{V_n}} \right\} \\ \text{if } f_{V_n} \leq V(X) \leq f_{V_{n+1}} \\ 0 \quad \text{Otherwise} \end{cases}$$

We now have two values associates with each voxel:

$C(X)$  - a shade calculated from a reflection model using the local gradient

$\alpha(X)$  - an opacity derived from tissue type of known CT values

The next stage, volumetric compositing, produces a two-dimensional projection of these values in the view plane. Rays are cast from the eye to the voxel, and the values of  $C(X)$  and  $\alpha(X)$  are "combined" into single values to provide a final pixel intensity.

For a single voxel along a ray, the standard transparency formula is:

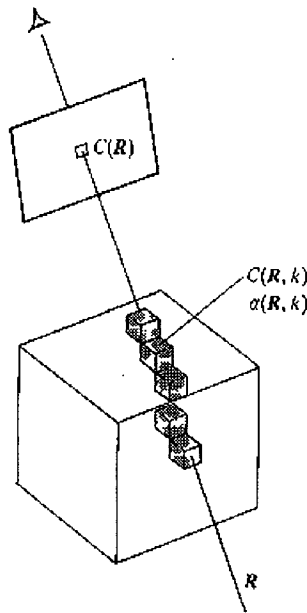


Figure 4.6: Combining the shade and the opacity into single value.

$$C_{out} = C_{in} (1 - \alpha(x_i)) + c(x_i)\alpha(x_i)$$

where:

- $C_{out}$  is the outgoing intensity/color for voxel X along the ray
- $C_{in}$  is the incoming intensity for the voxel

It is possible to interpolate from the vertex values of the voxel that the ray passes through, but it's better to consider neighboring voxels (8 or 26) and trilinearly interpolate.

This yields values that lie exactly along the ray.

As a result, this yields the following pseudocode:

```

Procedure TraceRay1(u)
Begin
   $C(u) = 0;$ 
   $A(u) = 0;$ 
   $X1 = First(u);$ 
   $X2 = last(u);$ 
   $U1 = Image(x1);$ 
   $U2 = Image(x2);$ 
  *** Loop through all samples falling within data ***
  For  $U := U1$  to  $U2$  do
    Begin
       $X := Object(U);$ 
      *** If sample opacity > 0, then resample and composite into ray***
       $A(U) := Sample(A, X);$ 
      If  $A(U) > 0$  then
        Begin
           $C(U) := Sample(C, X);$ 
           $C(u) := C(u) + C(U)(1-A(u));$ 
           $A(u) := A(u) + A(U)(1-A(u));$ 
        End
      End
    End
  End *** TraceRay ***

```



### 4.9.3 OPTIMIZING RAY CASTING

Ray casting is an extremely computation intensive process, much like ray tracing is. And like ray tracing, there are a number of optimization techniques that can be used to increase the speed with which an image can be rendered.

#### 4.9.3.1 HEIRARCHICAL SPATIAL ENUMERATION

This is the first optimization technique for ray casting algorithms. If we have a data set measuring  $N$  voxels on a side, and where  $N = 2^M + 1$  for some integer  $M$ , we can represent this enumeration by a pyramid of  $M+1$  binary volumes. Volumes in this pyramid can be indexed by a level number  $m$  where  $m = 0, \dots, M$ , and the volume at level  $m$  is denoted by  $V_m$ . Volume  $V_0$  measures  $N-1$  cells on a side, volume  $V_1$  measures  $(N-1)/2$  cells, and so on, up to volume  $V_m$  which is a single cell.

Cells are indexed by a level number  $m$  and a vector  $i = (i, j, k)$  where  $i, j, k = 1, \dots, N-1$  and the value contained in cell  $i$  on level  $m$  is denoted  $V_m(i)$ . The size of the cells on level  $M$  is defined to be  $2^m$  times the spacing between voxels. Voxels are treated as points, and cells fill the space between voxels. Thus, each volume is one cell larger in each direction than the underlying dataset. Voxel  $(1,1,1)$  is placed at the front-lower-right corner of cell  $(1,1,1)$ . Thus, cell  $(1,1,1)$  on level zero encloses the space between voxels  $(1,1,1)$  and  $(2,2,2)$ .

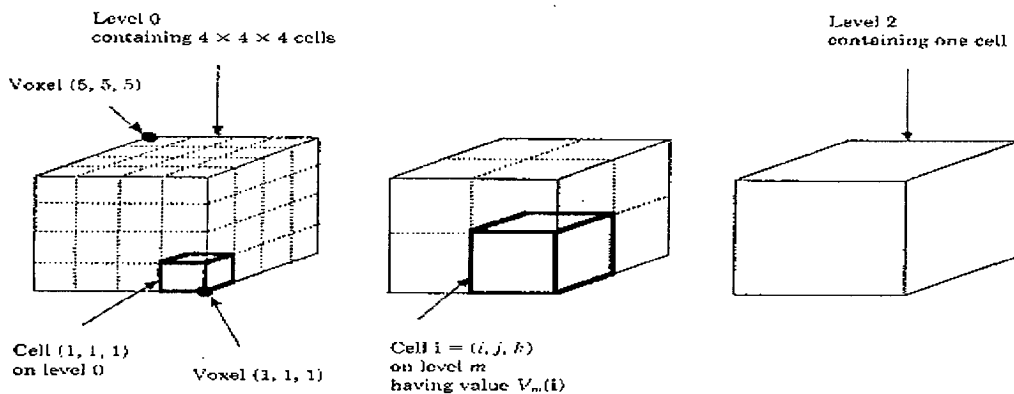


Figure 4.7: Constructing the Pyramid

To construct the pyramid, we do the following: Cell  $i$  in the base volume  $V_0$  contains a zero if all eight voxels lying at its vertices have opacity equal to zero. Cell  $i$  in any volume  $V_m$ ,  $m > 0$ , contains a zero if all eight cells on level  $m-1$  that form its octants contain zeros.

To state this differently, let  $\{1,2,\dots,k\}^n$  be the set of all  $n$ -vectors with entries  $\{1,2,\dots,k\}$ . Since we're concerned with three-dimensional data, we only care about the condition where  $n=3$ . This gives us  $\{1,2,\dots,k\}^3$ , which is the set of all vectors in 3-space with integer entries between 1 and  $k$ . Then, we can define

$$V_0(i) = \begin{cases} 1 & \text{if } \alpha(i + \Delta i) = 1 \\ & \text{for } i \in \{1,2,\dots,N-1\}^3 \text{ and any } \Delta i \in \{0,1\}^3 \\ 0 & \text{Otherwise} \end{cases}$$

and

$$V_m(i) = \begin{cases} 1 & \text{if } V_{m-1}(2i - \Delta i) = 1 \\ & \text{for } i \in \left\{1,2,\dots,\frac{N-1}{m+1}\right\}^3 \text{ and any } \Delta i \in \{0,1\}^3 \\ 0 & \text{Otherwise} \end{cases}$$

where  $m=1,\dots,M$ .

We have to reformulate the ray casting, resampling, and compositing steps of the rendering algorithm to use this pyramidal data structure. For each point, we first compute the point where the ray enters the single cell at the top level. We then traverse the pyramid in the following manner: When we enter a cell, we test its value. If it contains a zero, we advance along the ray to the next cell on the same level. If the parent of the new cell differs from the parent of the old cell, we move up to the parent of the new cell. We do this because if the parent of the new cell is unoccupied we can advance the ray farther on our next iteration than if we had remained on a lower level. This ability to quickly advance across empty regions of

space is where the algorithm saves its time. On the other hand, if the cell being tested contains a one, we move down one level, entering whichever cell encloses our current location. If we are already at the lowest level, we know that one or more of the eight voxels at the vertices of the cell have opacity greater than zero. We then draw samples at evenly spaced locations along that part of the ray falling within the cell, resample the data at these sample locations, and composite the resulting color and opacity into the color and opacity of the ray.

The following image shows in two dimensions how a typical ray may traverse a hierarchical enumeration.

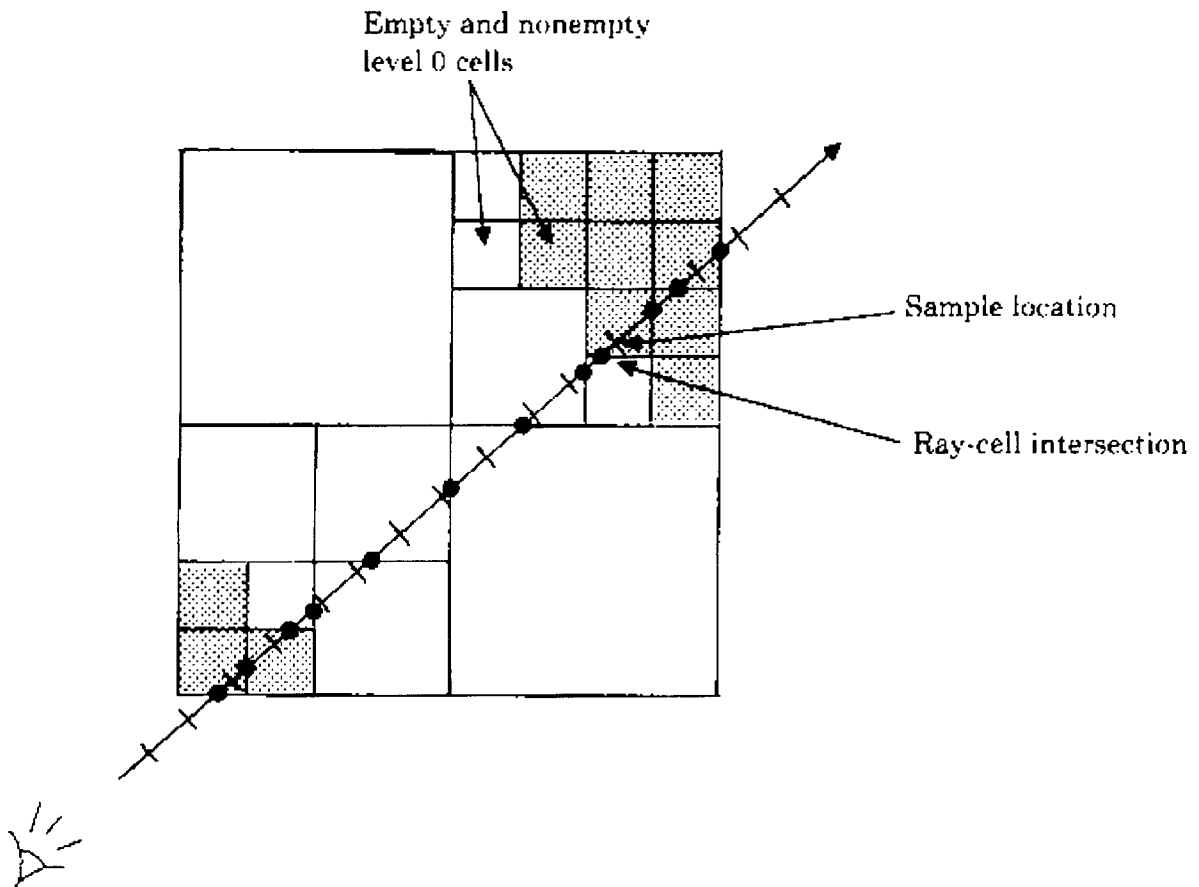


Figure 4.8: A ray traversing a hierarchical enumeration

#### 4.9.3.2 ADAPTIVE TERMINATION

The goal of adaptive termination is to quickly identify the last sample along a ray that significantly changes the color of the ray. We define a significant color change as one in which  $C_{out} - C_{in} > \text{Epsilon}$  for some small  $\text{Epsilon} > 0$ . Thus, Once  $\alpha$  first exceeds  $1 - \text{Epsilon}$ , the color will not change significantly. This is the termination criteria.

Higher values of Epsilon reduce rendering time, while lower values of Epsilon reduce image artifacts producing a higher quality image.

#### 4.9.3.3 COMBINING THE METHODS

By combining the two optimization methods, we can further increase the performance again. The two methods are combined in the following algorithm:

```
Procedure TraceRay2(u)
Begin
  C(u) := 0;
  A(u) := 0;
  M := mmax;
  *** Loop until beyond data or opacity > threshold ***
  While InBounds(x) and A(u) <= 1 - e do
  Begin
    I := Index(m,x);
    *** If high-level cell contains a one, drop a level ***
    if Vm(i) and m > mmin then m := m - 1;
    else
    begin
      *** If level-zero cell contains a one, render it ***
      if Vm(i) then RenderCell(u,x,Next(m,x,u));
      *** Advance to next cell and maybe jump to higher level. ***
      While Parent(m,Index(m,Next(m,x,u))) <> Parent(m,i) and m < M do
      Begin
        i := Parent(m,i);
        m := m + 1;
      end
      x := Next(m,x,u);
    end
  end
end TraceRay2
```

*Procedure RenderCell(u,x1,x2)*

*Begin*

*U1 := Image(x1);*

*U2 := Image(x2);*

*\*\*\* Loop through all samples falling within cell \*\*\**

*For U := U1 to U2 do*

*Begin*

*X := Object(U);*

*\*\*\* If any of eight surrounding voxels have opacity > 0, \*\*\**

*\*\*\* Then resample color and opacity and composite into ray \*\*\**

*if Va(Index(0,x)) then*

*begin*

*C(U) := Sample(C,x);*

*A(U) := Sample(A,x);*

*C(u) := C(u) + C(U)(1-A(u));*

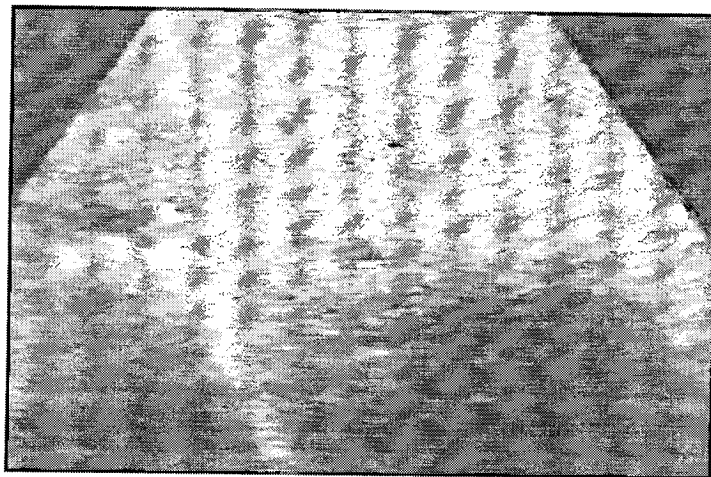
*A(u) := A(u) + A(U)(1-A(u));*

*end*

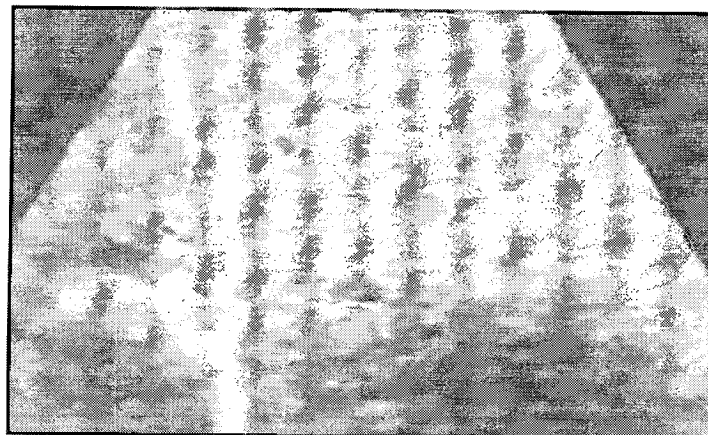
*end*

*End RenderCell*

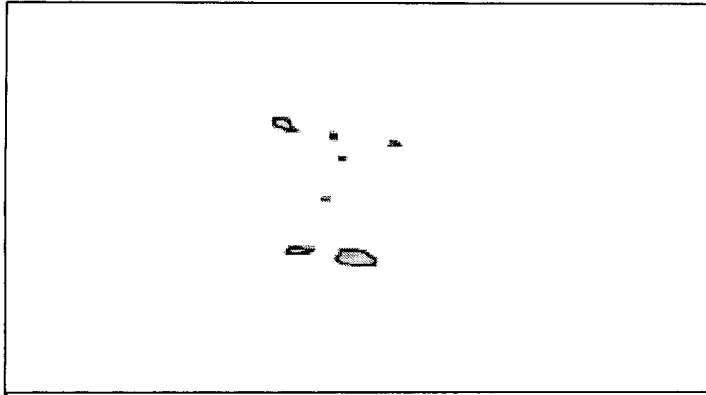
In the above algorithm, the Index function accepts a level number and the object-space coordinates of a point, and returns the index of the cell that contains it. The Parent function accepts a level number and cell index, and returns the index of the parent cell. The Next function accepts a level number and a point on a ray, and computes the coordinates of the point where the ray enters the next cell on the same level. The RenderCell function composites the contribution made to a ray by the specified interval of volume data. The algorithm terminates when the ray exits the data as detected by the InBounds procedure. [12,13].



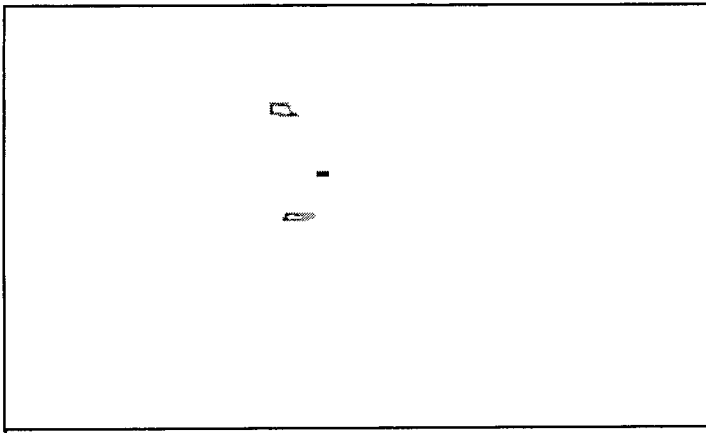
*Figure 4.9: Original Cropped Slice*



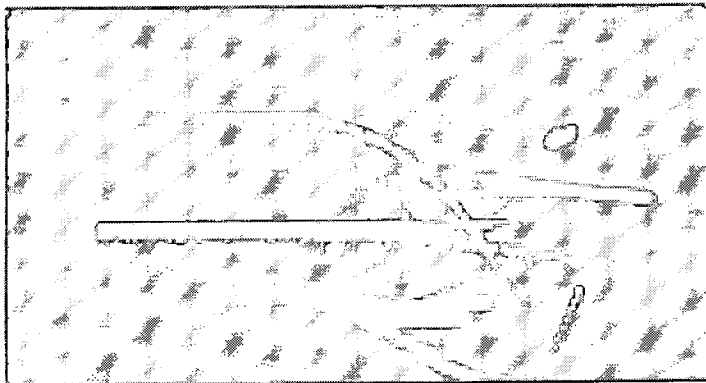
*Figure 4.10: Median Filtering 3x3*



*Figure 4.11: Segmented & Clustered*



*Figure 4.12: Labeled*



*Figure 4.13: 3D Transformation*

## **CHAPTER 5      CONCLUSION**

### **5.1      INTRODUCTION**

In planning the treatment of primary and metastatic liver tumors, understanding of the location of all hepatic tumor regions relative to major liver vascular anatomy is critical. This research's interest is the creation of three-dimensional liver images useful for pre-surgical planning. To be effective, these images must convey the three-dimensional relationships between the liver vasculature, tumor, and liver parenchyma. Diagnostic ultrasound, which is the most common form of medical imaging for monitoring liver problems, only provides a qualitative measure of the liver's veins. A manual method to segment veins from ultrasound liver slices exist, which requires a clinician to print a copy of the ultrasound slices sequence, digitize the sequence into a computer, and then manually trace the veins on each image of the sequence.

This method worked well, but it consisted of time-consuming and disjoint steps. Of these steps, the most critical was the manual tracing of the veins. An automatic segmentation for identifying the veins areas will not only decrease the amount of time for identifying the needed regions, but also to improve the accuracy of selections. The creation of a complete automated volumetric measurement system, which utilized this automatic segmentation procedure, would then further integrate the disjoint steps of the measurement process. An automated system will not only let clinicians make volumetric measurements more quickly, but also easily, and frequently.

### **5.2      ACCOMPLISHMENTS**

The first accomplishment of this thesis was the development and implementation of an automatic segmentation procedure for identifying the veins in a liver ultrasound image.



The first stage of this procedure is to preprocess the image using two median histogram filters. This removes noise and smoothes the image. A texture analysis technique is then used to automatically identify the veins regions from the ultrasound sequence slices and here Multiresolution Fractal feature is used which is made out of Multiple Resolution feature technique and Fractional Brownian Motion model, using both of them has an advantage over other texture analysis techniques since it reads what other techniques can read from an ultrasound image and more the granularity, roughness and regularity parameters which are vital information to identify texture types from ultrasound images. A quantizing algorithm is then used to partition the data into distinct clusters according to the intensity level of the pixels. This serves to normalize the histogram distribution so that a fixed thresholding operation can be used to cluster the veins from other tissues or scatters in the image. After this thresholding operation, these veins are labeled to connect them in a later stage to be displayed in 3D image.

Finally, this research should be considered as the beginning into a more sophisticated automated system that should identify the existing of tumors and their place with relation to the veins and be displayed in 3D. As well, to identify other liver metastases and problems and to also display them in 3D with respect to the veins to give as much help to the clinicians for their pre-surgical planning.

## BIBLIOGRAPHY

- 1 R. M. Haralick, K. Shanmugan, and I. H. Dinstein. Texture features for image classification. *IEEE Trans. Syst., Man, Cyber.* SMC-3: 610-621, 1973.
- 2 S. Peleg, J. Naor, R. Hartley, and D. Avnir. Multiple resolution analysis and classification. *IEEE Trans. Pattern Anal. And Machine Intell.* PAMI-6: 518-523, 1984.
- 3 T. Lundahl, W. J. Ohley, S. M. Kay, and R. Siffert. Fractional Brownian Motion: A maximum likelihood estimator and its application to image texture. *IEEE Trans. Med. Imaging.* MI-5: 152-161, 1986.
- 4 F. L. Lizzi, M. Ostromogilsky, E. J. Feleppa, M. C. Rorke, and M. M. Yaremko, Relationship of ultrasonic spectral parameters to features of tissue microstructure. *IEEE Trans. Ultra. Ferro. Freq. Control*, UFFC-34: 319-329, 1986.
- 5 K. Shung. General engineering principles in diagnostic ultrasound. *IEEE Engineering in Medicine and Biology*, 6(4):7-13, December 1987.
- 6 E. J. Feleppa and M. M. Yaremko. Ultrasound tissue characterization for diagnosis and monitoring. *IEEE Engineering in Medicine and Biology*, 6(4): 18-26, December 1987.
- 7 J. W. Klinger, C. L. Vaughan, T. D. Fraker, and L. T. Andrews. Segmentation of echocardiographic images using mathematical morphology. *IEEE Transactions on Biomedical Engineering*, 35:925, 1988.
- 8 N. Friedland and D. Adam. Automatic ventricular cavity boundary detection from sequential ultrasound images using simulated annealing. *IEEE Transactions on Medical Imaging*, 8(4): 344, 1989.
- 9 C. C. Chen, J. S. Daponte, and M. D. Fox. Fractal feature analysis and classification in medical imaging. *IEEE Trans. Med. Imaging.* 8(2): 133-142, 1989.
- 10 M. Unser and M. Eden. Multiresolution feature extraction and selection for texture segmentation. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-11: 717-728, 1989.
- 11 S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-11: 674-693, 1989.
- 12 J. Levoy. Ray Casting Techniques. *IEEE Transactions on Medical Imaging Graphics.* 9(2): 344-367, 1990
- 13 J. Levoy. Optimization of Ray Casting. *IEEE Transactions on Medical Imaging Graphics.* 9(3): 75-92, 1990

- 14 J. F. Thomas, R. A. II Peters, and P. Jeanty. Automatic segmentation of ultrasound images using morphological operators. *IEEE Transactions on Medical Imaging*, 10(2):180, 1991.
- 15 R. Gonzalez and R. Woods. Digital Image Processing. Addison-Wesley publishing company. 1992.
- 16 C. Fortin, R. Kumaresan, W. Ohley, and S. Hofer. Fractal Dimension in the analysis of medical images. *IEEE Eng. In Med. And Bio.* 11(2): 65-71, 1992.
- 17 M. Joliet and B. M. Mazoyer. Three-dimensional segmentation and interpolation of magnetic resonance brain images. *IEEE Transactions on Medical Imaging Graphics*, 12(2): 269, 1993.
- 18 J. Buhmann and H. Kuhnel. Vector quantization with complexity costs. *IEEE Trans. On Information theory.* 39: 1133-1145, 1993.
- 19 Hoover, D. Goldgof and K. W. Bowyer. Extracting a valid boundary representation from a segmented range image. *IEEE Trans. On Pattern Anal. And Mach. Intell.* 17(9): 880-893, 1995
- 20 B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Trans. On Pattern Anal. And Mach. Intell.* 17(1): 72-77, 1995.
- 21 E. K. Fishman, B. S. Kuszyk, D. G. Heath, and L. Gao. Surgical Planning for liver Resection. *Computer.* 29(1): 64-72, 1996.
- 22 Y. M. Kadah, A. A. Farag, J. M. Zurada, A. Youssef, and A. Badawi. Efficient algorithms for computer-assisted diagnosis of diffused liver diseases from ultrasound images. *IEEE Trans. On Medical Imaging.* 15(4): 466-478, 1996.
- 23 Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experiment comparison of range image segmentation algorithms. *IEEE Trans. On Pattern Anal. And Machine Intell.* 18(7): 673-688, 1996.
- 24 R. P. Grzeszczuk and D. N. Levin. "Brownian Strings": Segmenting images with stochastically deformable contours. *IEEE Trans. On Pattern Anal. And Machine Intell.* 19(10): 1100-1114, 1997.

- 25 S. R. Gunn and M. S. Nixon. A robust snake implementation; A dual active contour. *IEEE Trans. On Pattern Anal. And Machine Intell.* 19(1): 63-68, 1997.
- 26 T. Hoffman and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. On Pattern Anal. And Machine Intell.* 19(1): 85-99 1997.
- 27 K. L. Vincken, A. S. E. Koster, and M. A. Viergever. Probalistic multiscale image segmentation. *IEEE Trans. On Pattern Anal. And Machine Intell.* 19(2): 109-120, 1997.