SELECTING AND ASSESSING QUANTITATIVE EARLY ULTRASONIC TEXTURE MEASURES FOR THEIR ASSOCIATION WITH CEREBRAL PALSY

by

Tyna A. Hope

Submitted

in partial fulfillment of the requirements

for the degree of

DOCTOR of PHILOSOPHY

Major subject: Electrical and Computer Engineering

at

DALHOUSIE UNIVERSITY

Halifax, Nova Scotia June, 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-27641-9
Our file Notre référence
ISBN: 978-0-494-27641-9

Contents

List of Figures

List of Tables

List of Symbols and Abreviations

Δt	pulse duration
$\Delta\theta$	lateral resolution
Δr	axial resolution
Δf	distance to first zero in frequency domain
λ	wavlength
μ	mean
σ	standard deviation
σ^2	variance
a	diameter of a sphere
c	speed of sound
d	distance
f	frequency
f_c	center frequency
k	wavenumber
kNN	k-nearest neighbor
$mtry$	number of randomly selected variables used for node creation
$ntree$	number of trees in the ensemble
pdf	probability density function
pgm	portable grey map
t	time
z	distance
ANN	artificial neural network
B	bandwidth
B_k	bootstrap sample
CAD	computer-aided-diagnosis
CART	classification and regression trees algorithm
CP	cerebral palsy
$D(f,z)$	beam diffraction transfer function
DM	difference in image and mode filter

$E(f,z)$	Fourier transform of $e(t,z)$
EXP0	group of 30 images used for segmentation experiments
EXP1	group of 17 images used for design and selection of texture measures
EXP2	group of 69 images used to design and test the model
F	number of features in a classification problem
FD	fractional differencing
FDS	fully developed speckle
FS	feature selection
FE	feature extraction
FN	test result reported falsly as negative
FP	test result reported falsly as positive
FWHM	full-width-at-half-maximum
GABT19	Gabor filter used in a variable image processing method
GLHW	grey-level histogram width
HLT	Hashimoto's lymphocytic throiditis
IDD30	morphology-based filter
LDA	linear discriminant analysis
MLC	machine learning classifiers
MRI	magnetic resonance imaging
NPV	negative predictive value
NSA	normalized surface area
OOB	out-of-bag error estimate
ORIG	images, noise-reduced and resolution-reduced
P & C	perturbing and combining
$P(f,z)$	transducer transfer function
PCA	principle component analysis
PPV	positive predictive value
PVL	periventricular leukomalacia
QDA	quadratic discriminant analysis
R	R system for statistical computing and graphics
RF	radio frequency

RF-BHC	a modified random forest classifier
RFC	random forest classifier
ROC	receiver operator characteristic
ROI	region of interest
$S(f)$	backscatter transfer function
SNR	signal to noise ratio
SSP	split-spectrum processing
STDV	standard deviation texture measure
T	delay
$T(f, z)$	tissue transfer function,
TGC	time gain compensation
TN	test result correctly reported as negative
TP	test result correctly reported as positive
SVM	support vector machine
US	ultrasound
UTC	ultrasound tissue characterization
WMD	white matter damage

Acknowledgements

I would like to thank my supervisor, Dr. P.H. Gregson, and my co-supervisors, Dr. N.C. Linney, and Dr. M.H. Schmidt. This multi-disciplined team provided the needed knowledge and guidance in this complex research project. Each indivdual provided a unique and valuable perspective for me to learn from.

Thanks to Dr. S. Nugent, Dr. W. Phillips, and Mr. M. Abdolell for agreeing to be my thesis committee members. I'd like to particularly thank Mr. Abdolell for providing guidance in the area of model creation. Thank you to Dr. A. Fenster, of Robart's Research Institute, for your participation as my external examiner.

This research was funded by the Natural Sciences and Engineering Research Council Discovery Grant Program, an IWK category B grant, the Dorothy and Bruce Rosetti Scholarship from the Faulty of Engineering, and a student scholarship from the Nova Scotia Health Research Foundation.

My thanks to the IWK Health Centre for the office space and to all those individuals at the IWK Health Centre who went an extra mile in suppport of this work. Particularly Michael J. Vincer, Director of the Perinatal Follow-up Clinic, IWK Health Centre, for his selection of the case matched data set that formed EXP2.

I'd like to thank the Institute for Computational Astrophysics at Saint Mary's University for providing access to its Linux research cluster "pluto" which was used in support of this work. Thanks to Zhengyan Sun, of Saint Mary's University, for her work on the algorithm to extract the theoretical point of insonation.

Thanks to Jeanette W. Evans for conducting the clinical study of radiologists' assessment of WMD on the patients in EXP2. Also to the pediatric radiologists Marian B. Macken and M. Kathleen O'Brien, of the IWK Health Centre, for their participation in the study.

I would also like to acknowledge the support from my husband, Michael Boudreau, my parents, David and Violet Hope, and my extended network of family and friends. Without them, this would have seemed like an impossible task.

Abstract

Preterm infants are susceptible to white matter damage (WMD), which is associated with cerebral palsy (CP) and cognitive impairment. Ultrasound (US) is the preferred imaging modality to detect WMD but suffers from poor sensitivity and specificity in the early postnatal period. To improve on existing diagnostic rates, quantitative measures incorporating new information are needed. Ultrasound texture measures have been shown to reveal diagnostic information about human tissue. In this research, unique texture measures are extracted using adaptive preprocessing and high-resolution feature enhancement. The clinical diagnosis of CP presently is made at 12 to 18 months. As it is desirable to detect the disease in its early stage, clinical B-mode images taken within days of birth are used in this research.

In this study, the images are not standardized but use the patient as his or her own control. Speckle is not removed as speckle may contain information. To test the hypothesis that ultrasonic texture in these early images are associated with patient outcome, a model using only texture measures is created and evaluated. The "Random Forest" algorithm is used to form the model. The design of the texture measures and the selection of the variables are performed with a data set distinct from the set used for design and evaluation of the model. The resulting model has an accuracy of 72.5%. Random noise would provide a model with 50% accuracy, and designating all patients as having CP would result in 54 % accuracy. This result suggests that early quantitative texture measures contain diagnostic information relevant to patient white matter health.

Chapter 1

Introduction

1.1 Motivation

Very preterm infants are prone to brain damage, in particular white matter damage (WMD). WMD is associated with cerebral palsy, cognitive impairment [1] and sudden infant death syndrome [2].

Ultrasound (US) is commonly used to screen for WMD. It is the preferred imaging modality, as it is the safest for very preterm infants. Magnetic resonance imaging (MRI) detects WMD with greater sensitivity than US [3], but obtaining MRI images is non-trivial. It requires that the infants to be moved from their incubators and to be sedated. MRI images are not only difficult to obtain but can pose a health risk to the infant due to these manipulations. US machines, on the other hand, are readily available and can be taken to the infant in the neonatal intensive care unit.

Currently, diagnosis with US is qualitative. Radiologists depend on echolucencies and echodensities as cues to the presence of WMD [4]. The diagnostic process is subject to inter- and intra-observer variability. Current diagnosis of WMD using US catches only the tip of the iceberg [5]. While the focal component of WMD can be diagnosed with US, the diffuse component is considered invisible [1]. US intensity information alone is not satisfactory for diagnosing WMD [3] [5]. Using US images, it is desirable to improve the consistency of diagnosis of WMD and to increase the amount of diagnostic information that is extracted.

1.2 Scope of the Research

The hypothesis explored in this work is that ultrasound texture measures, taken from the first cranial ultrasound images, contain information that is associated with patient outcome. To prove the hypothesis, it must be shown that there is signal in the texture measures obtained. This will be accomplished by building a model and evaluating its error rate. If it shows greater than 50 % accuracy, for the given available sample, this will be evidence that the texture measures contain signal associated with patient outcome.

There are previous experiments that correlate acoustic properties with B-mode texture properties [6]. A new combination of processing methods is proposed to extract these properties. The measures were obtained from a single image from each patient and limited to the white matter and choroid plexus regions of the brain. A large number of measures, two-hundred and fifty-six, were initially investigated as candidate measures. Prior to the construction of the model, variables were assessed based on engineering principles and observed relationships between the variables and the patient outcome. Standard statistical practice was also used to reduce the number of correlated variables. Using this prior knowledge, variable selection was performed. This is a necessary step because excessive noise is known to degrade model performance.

The model used to assess the association between the quantitative measures and patient outcome was a multiple classification system known as random forests. The model is created and its error evaluated with a data set separate from the one used for the design and selection of the texture measures to determine if the measures contain diagnostic information.

1.3 Organization of the Dissertation

The dissertation is organized in the following manner:

- Chapter 2 : A description of the physics of B-mode imaging and the disease process of white matter damage are provided. Also discussed is previous work on the measurement of image texture in medicine and other fields. This discussion is limited to ultrasound and other coherent imaging modalities. Finally, the use of the random forest algorithm in imaging problems is reviewed.

- Chapter 3 : The texture-measure extraction algorithm is presented. A complete discussion of the texture measures extracted in this research, and how they were derived and selected, is provided. A review of the various modeling techniques is presented, as well as the rationale behind the choice of the random forest algorithm for model creation. The issues around the selection of variables are provided.
- Chapter 4 : A description of how the experiments were performed is presented.
- Chapter 5 : The results of the experiments are presented. A discussion of the significance of the results in terms of future signal analysis of B-mode images is presented as well as the potential implications for clinical practice.
- Chapter 6 : Conclusions are drawn and suggestions for future research are provided.

Chapter 2

Background

Ultrasound tissue characterization is a field that requires knowledge of many topics. One must have a familiarity with the imaging technology to understand what it measures and what it has the potential to measure. One must also have an understanding of the disease process to determine if the changes that it causes have properties that can be detected by the imaging system. As with any topic, one must know the history to learn from successful and less-than-successful research. And finally, it can be of enormous value to review the work in related fields. These topics will be covered in the following sections.

2.1 B-mode Ultrasound Image Formation

An US image is constructed from ultrasonic pulses reflected by biological tissue. Most references describe the speed of sound in biological tissues as 1540 m/s. However, there are local variations in the speed of sound, the absorption, and the reflective properties of tissue. These variations in acoustic properties provide the information within US images. Some diseases alter the normal tissue structure and components, causing changes in the tissues' acoustic properties and thus the US image. In order to understand the limitations of US images, the basics of ultrasound image formation are presented.

The ultrasound image is formed by emitting an ultra-high frequency sound-wave pulse, obtaining the reflected signal and building the image based on the reflected-signal properties. A pulse specification determines the properties of the transmitted acoustic pulses. The specification is defined by user input and system parameters. After transmission and

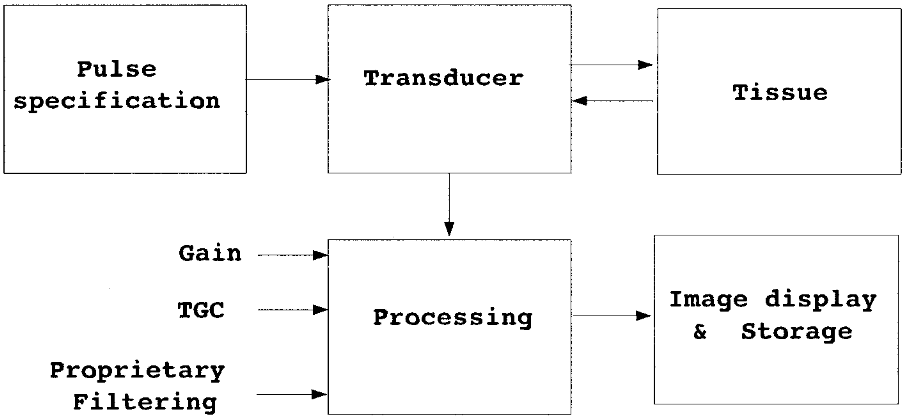

Figure 2.1: Block diagram of a medical ultrasound system.

reflection through the body, the transducer array receives the returned pulses. The returned signal is then processed and an image is formed. In B-mode images, the return radio-frequency (RF) signal is envelope-detected so all phase information is lost. The image-pixel intensity is based on the amplitude of the envelope. The pixel coordinates are defined by the time that the signal takes to return and the direction in which the beam was sent. After the image is formed, further processing takes place to enhance the image. See figure 2.1 for a block diagram of a medical ultrasound system. The following details of the system are provided with respect to the signal transmission, changes to the signal, and processing.

2.1.1 Signal Transmitted

Pulse formation

The ultrasonic pulse is generated by piezoelectric transducers. These transducers convert electrical pulses to mechanical vibrations and *visc versa* on their return [7]. The center frequency of the pulse is in the range of 7.0 MHz to 8.5 MHz for neonatal cranial images. Typical ultrasound imaging frequencies range from 2 MHz to 10 MHz for medical applications, with some higher-frequency exceptions in ophthalmology and skin imaging [8].

The depth of penetration of the sound is inversely proportional to the center frequency (f_c). Conventional wisdom holds that image resolution is proportional to wavelength, λ, which is $\lambda = c/f_c$, where c is the speed of sound in the body [9]. Image resolution is also determined by the time between the pulses. The time delay between pulses must be long enough to detect the slowest returning pulse before transmitting an additional pulse. The system cannot tell the difference between a pulse sent back from a very close object and a pulse that has taken longer than the time delay between two pulses. It should be noted however, that sub-wavelength sized particles (0.09λ) have been identified using ultrasound with the appropriate processing of the RF signal [10]. Clearly, there is some question about the resolution limitations of US.

The ultrasound pulse is shaped to improve its propagation characteristics. Typically a Gaussian shape is used, however, new shapes are being investigated to improve the pulse time-bandwidth product. The time-bandwidth product is a measure of the degree to which the generated pulse approximates the theoretically ideal pulse, and improving this value increases the signal-to-noise ratio (SNR). Shaping concentrates the energy of the pulse at the focal point and removes the wasted energy in the side lobes generated by square pulses. Pulse shaping is also used to improve imaging when contrast media are used [11].

Beam Steering

To control the direction of the ultrasound signal and its focal point, the ultrasound transducer consists of a steerable array of piezoelectric transducers rather than a single transducer. In the same manner as radar transmission, the direction of the beam is electronically controlled by varying the relative phase of oscillation of the elements [12]. Figure 2.2 illustrates beam steering [13].

2.1.2 Signal Received

General

The return signal $e(t, z)$ where t = time and z = distance, can be described as the transmitted signal modified by the transducer point-spread function and tissue properties such as attenuation, beam diffraction and backscatter. Thijssen [6] presents this model in the

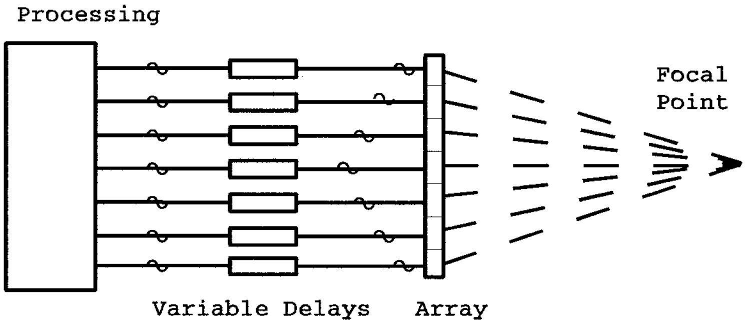

Figure 2.2: The directed wavefront is formed by delaying the pulses from the individual transducers that make up the aperture of the ultrasound machine. Copied with permission from www.analog.com.

frequency domain:

$$E(f,z) = P^2(f,z)D^2(f,z)T^2(f,z)S(f) \tag{2.1}$$

where

f is frequency

$E(f,z)$ is the Fourier transform of $e(t,z)$

$P(f,z)$ is the transducer transfer function,

$D(f,z)$ is the beam diffraction transfer function,

$T(f,z)$ is the tissue transfer function,

$S(f)$ is the backscatter transfer function.

This is not the only model of the returned signal, but it is presented here as it provides a logical breakdown of components. The return signal model illustrates the fact that while tissue properties affect the return radio-frequency signal, the design of ultrasound-machine components also has a significant effect.

As ultrasound pulses are applied to the body, they interact with many different-sized structures. They encounter veins, capillaries, tissue boundaries, tissue substructures and so forth. The traveling waves are absorbed, scattered and reflected by structures. The returned signal can be considered to be composed of signal attenuation, diffuse signal backscattering and non-diffuse reflected signal. Diffuse reflection occurs when there are many small boundaries with respect to size of the wavefront. Specular reflection indicates the location of a large boundary. As the size of the object increases, there is a gradual transition from diffuse reflection to specular reflection.

Fully Developed Speckle Formation

Fully developed speckle (FDS) is present in both the ultrasound RF return signal and the envelope-detected data [14]. In the literature, some authors use the term "speckle" to describe only FDS, while others use the term for the grainy appearance of the US image texture in general. In this thesis, "speckle" is used to describe the basic units of the ultrasound image texture , or primitive, including FDS.

FDS is the diffuse backscattering of the signal within a uniform medium with a high density ($>$ 10 per resolution cell) of small sub-components of size $<<\lambda$ where λ is the wavelength [15]. A simple description is that FDS is the interference pattern by the echoes from structures that are not resolvable by the ultrasonic system. While this simple model is easy to understand, it must be stressed that the reflected signal is actually a complicated phenomenon that is affected by local acoustic impedance boundaries within a complex medium.

To elaborate, acoustic media contain sub-particles with varying mechanical properties of density and compression. If the sound wave impinges upon a substructure in a medium that has identical density and compressibility (acoustic impedance) to the surrounding medium, then there is no resulting scattering. Faran proposed a complex but accurate model to describe the scattering phenomenon in a medium which contains spherical scatterers in a fluid [16]. This model has been verified experimentally by Faran and other researchers [17]. This model provides interesting insight into the amplitude of the backscattered sound [14]. For scatterers much smaller than the wavelength, the intensity is proportional to f^4. The intensity of the back scattered signal can be observed as a function of the diameter of the sphere, a, and the wavenumber of the incident sound wave ($k = 2\pi/\lambda$). The relationship between ka and the intensity of the backscatter is complex with many peaks and nulls that are caused by the effects of sound penetrating the sphere and reverberation within the sphere. While biological tissues are not comprised of spheres, these experiments serve to remind researchers that any simple model will likely leave out some effects contributing to the RF and B-mode image measurements.

The reflections from structures vary continuously from fully developed speckle to specular reflection. The question remains: Where does speckle noise stop and texture begin? For B-mode images that are envelope-detected and without logarithmic compression, FDS is considered to be present when SNR = 1.91 due to the Rayleigh-shaped intensity distribution of multiplicative noise [18]. FDS is considered to be particularly troublesome when the processing goal is the extraction of edges [19],[20],[21]. However, computer-aided detection systems have incorporated speckle with improved results [22]. Thus, the debate over the information content of speckle continues.

2.1.3 Signal Processing

Both the return signal and the constructed image may be modified to improve the visibility of desired features. The signal processing occurs on many different levels. The processing can occur on the signal (pre- or post-sampling) that forms an image line, a group of lines, or the entire image. It can occur prior to, during or after image construction [11]. Some of this manipulation is within the operator's control, while most is not. To the user of the images, the US machine is frequently a black box.

Manufacturer-controlled

The signal may be processed through beam-forming, single-line RF processing and multi-line RF processing. Beam forming is the process of delaying the output of the individual piezoelectric transducers to combine the signal in such a way as to reduce noise [23]. Some manufacturer-controlled single-line RF processing includes bandpass filtering and echo line-signal averaging. Multi-line RF processing may be in the form of lateral gain and interpolation between RF lines [11]. And finally, when the image is created, the dynamic range of the signal is compressed through logarithmic compression and, in the case of a phased-array transducer, the coordinate system of the signal is converted from a polar coordinate system to a rectangular coordinate system.

Operator Controlled

Operators have the ability to influence the processing of the signal by adjusting machine parameters including gain and time-gain compensation (TGC). Gain addresses the difficulty obtaining good contact at the skin surface, and the depth and impedance of the tissue of interest. The TGC compensates for the continual attenuation of the signal with increasing depth. There are frequently several spots in the system where the gain may influence the signal. These may be interspersed with non-linear processing blocks. The single gain knob may affect any or all of these [24].

The operator can influence the image processing. Ultrasound machine manufacturers offer components for processing images in an attempt to improve the image quality. Since most ultrasound images are interpreted qualitatively by a human observer, the amount of image improvement is a subjective assessment. As an example, signal processing techniques in the Sequoia 512 platform include Tissue Equalization^{TM} Technology which is an enhancement claiming to "automatically equalize tissue gain and brightness in two dimensions, providing consistent, reproducible image quality [25] . However, the algorithms which comprise these processing components are proprietary so their specifics are not available to the operators.

Since the operators are not usually experts in image processing, the use of image processing settings is based on the operators' judgment of image quality and manufacturer recommendation. Fortunately, some processing techniques, such as persistence (frame averaging), are common among manufacturers. To facilitate its use, post-processing is often controlled by a selection of a number which represents the type of filtering performed. Manufacturers often provide suggestions for image processing settings based on the type of scan being performed. Unfortunately, the use of these settings may be inconsistent as the final control is based on the operator's judgment.

Properties of the Constructed Image

The constructed image is subject to restrictions in axial and angular resolution. Prior to sampling the RF signal, these properties are limited by the center frequency, transducer bandwidth, focus depth, and the time between transmitted pulses [12]. In modern digital ultrasound images, additional resolution constraints may be caused by filtering and sub-

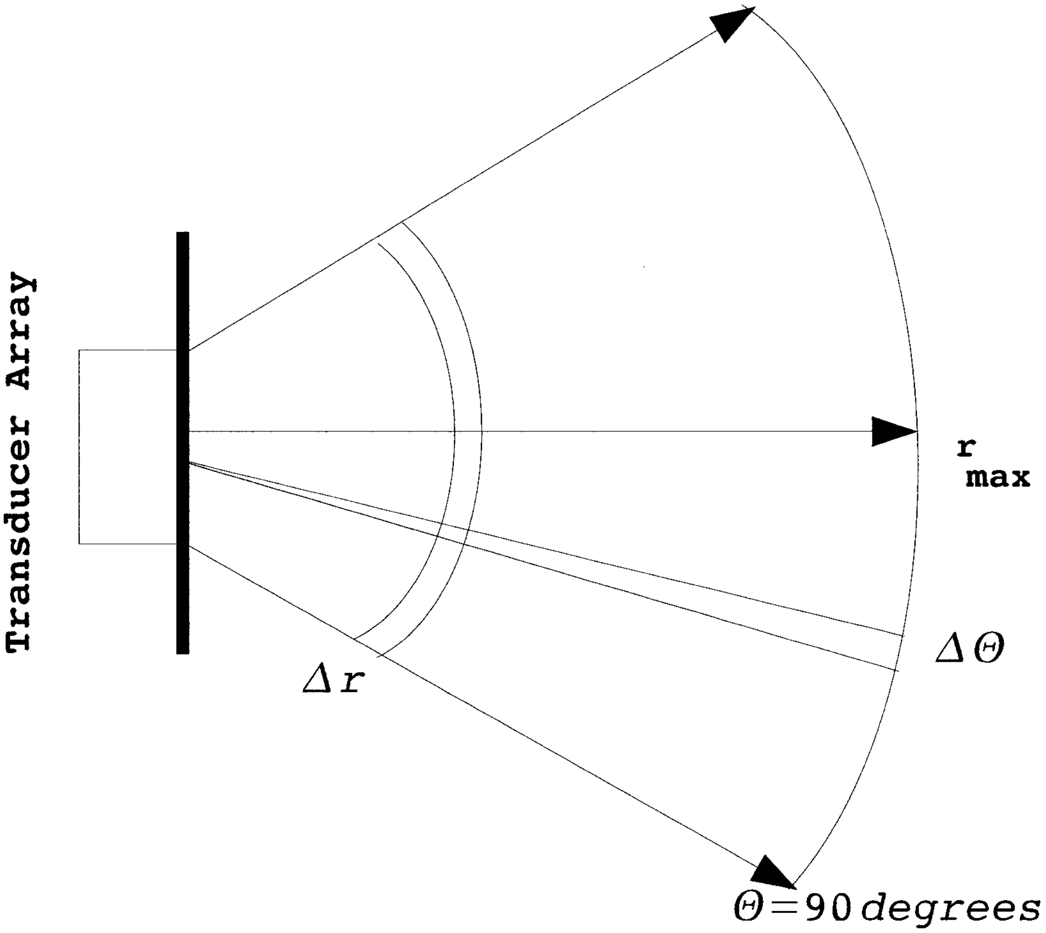

Figure 2.3: The resolution is described in terms of its axial (r) and lateral (θ) components.

sampling the signal before forming the image. The lateral and axial resolution restrictions, before sub-sampling, are now discussed.

Axial Resolution

For simplicity, assume that the transmitted pulse is formed by a rectangular gate function in time. Let B be the bandwidth of the transmitted pulse, T be the delay between the pulses, c be the speed of sound in the tissue, Δt be the duration of the pulse and Δr be the axial resolution (figure 2.3). Generally, the maximum depth that can be imaged by an ultrasound system has the limitations of 1) increasing r increases T, and 2) the pulse energy sets an outer limit to the distance that can be travelled due to attenuation. The relationship between B, Δr, c, and Δt can be derived by the shape of the spectrum, assuming a sinc function, and the location of its zeroes [12], leading to $B = 2\Delta f = 2\Delta t = 2c/\Delta r$. Thus, the axial resolution is dependent on the bandwidth of the pulse as well as its center frequency.

Lateral Resolution

The lateral resolution is dependent on the center frequency and the aperture size A. Aperture is the face of the transducer that is in contact with the body and determines the beam size. In an ideal situation, the beam is highly concentrated at the focus, so that one can be sure that the returning echoes come from the focal point. With a square aperture, side lobes exist, causing the beam to be less focused. To reduce this problem, the energy of the pulses sent from each transducer is tapered (apodized) to remove the square window and thus the side lobes. Based on a square aperture and the assumption that the beam is focused in the far field, the relationship $\Delta\theta = 2sin^{-1}(c/A * f_c)$ can be derived [12]. Apodization improves contrast but at the expense of lateral resolution [14].

2.2 Ultrasound and White Matter Damage

While understanding the physics of the imaging system is important, equally important is the mechanism of the disease. It is only through understanding the disease process that it can be suggested that there is information worth pursuing in US images.

2.2.1 The Disease Process

Cerebral palsy is a lifelong condition that results in weakness and spasticity. It is caused by brain damage. It is the end result of periventricular leukomalacia (PVL), which is the softening of the white matter tissue adjacent to the ventricles in the brain [26]. White matter is particularly susceptible to damage caused by complications of premature birth. Some of the complications include maternal infection, asphyxiation during birth, and respiratory distress after birth [27]. Healthy white matter is anisotropic, having a distinctive orientation indicated by white matter tracts, (Figure 2.4). White matter consists of nerve fibers tracts which are covered by a fatty sheath of myelin that assists in the conduction of electrochemical impulses. White matter gets its name from the appearance of the fatty layer, as opposed to grey matter which consists only of nerve cell bodies.

White matter of the preterm infant differs from that of the term infant. Myelination of the nerve fibers forming the white matter occurs throughout infancy, with the fibers becoming more compact as the brain develops. This has been evidenced in studies using MRI to assess the amount of myelination in preterm infants, over the gestational ages of 28 weeks to term [28]. The white matter of the preterm infant's brain contains a larger percentage of water due to the lack of myelination.

WMD is caused by non-hemorrhagic infarct, which is a blockage or reduction in blood supply leading to cell death. The myelin sheath is formed by cells called oligodendrocytes. The precursor cells to the oligodendrocytes are very susceptible to injury from the lack of blood supply. Injury to these cells cause a reduction in myelin sheath development of the white matter.

The details of the changes immediately after the injury are as follows. After the first 12 hours of non-hemorrhagic infarct, damage to the cells cause ischemic neuronal change, cytotoxic edema, and vasogenic edema predominate [29]. Cytotoxic edema is an increase in fluid within the cells. Vasogenic edema is caused by a breakdown of the blood brain barrier [30]. From 48 hours to up to 3 weeks later, macrophages, as part of the inflammatory response, become the predominant cell type consuming the products of myelin breakdown and blood. Liquefaction and phagocytosis (ingestion of micro-organisms or debris) continue. Astrocytes, cells that control the blood-brain barrier and form scar tissue, enlarge and form a network at the lesion perimeter in a process called gliosis. The astrocytes can be present at one week after the insult.

After the insult occurs, macrophages proliferate. While myelin and macrophages have very similar chemical properties, their structures are very different. Myelin cells follow the shape of the neuronal processes, in this case relatively long white matter tracts. Macrophages are small single cells, their size being in the tens of micrometres.

Thus, due to cell death, edema, and an inflammatory cellular infiltration, the structure of the white matter of a neonate having undergone an insult leading to brain injury will differ from other preterm infants. The structure will also differ from normal term infants, who have more developed white matter and no inflammatory changes. The end result is predicted to be a change in acoustic scatterer density and scatterer structure and a change in the anisotropic structure of white matter. The anisotropic nature of white matter has been shown to effect the acoustic wave velocity. This differs significantly in the longitudinal and transverse directions in experiments with a 90 MHz pulse [31]. These changes in the structure of the white matter fibers, may be detectable with appropriate B-mode image measures. As well, based on the timeline of the changes, the change may be detectable within days of the injury.

PVL consists of both diffuse and focal components [1]. Cystic PVL represents the focal damage and the end-stage of the body's response to it. This damage is typically detectable in US images. The diffuse injury is a distributed injury to the oligodendrocytes. Typically, diffuse disease is more difficult to detect in US evaluation. There is evidence that changes in scatterer density, types and sizes may be detectable as changes in B-mode image texture. This evidence, presented in the following sections, suggests that changes in the acoustic properties of media result in changes in B-Mode texture. However, before discussing B-mode texture experiments, some of the relevant texture and image processing terminology is presented.

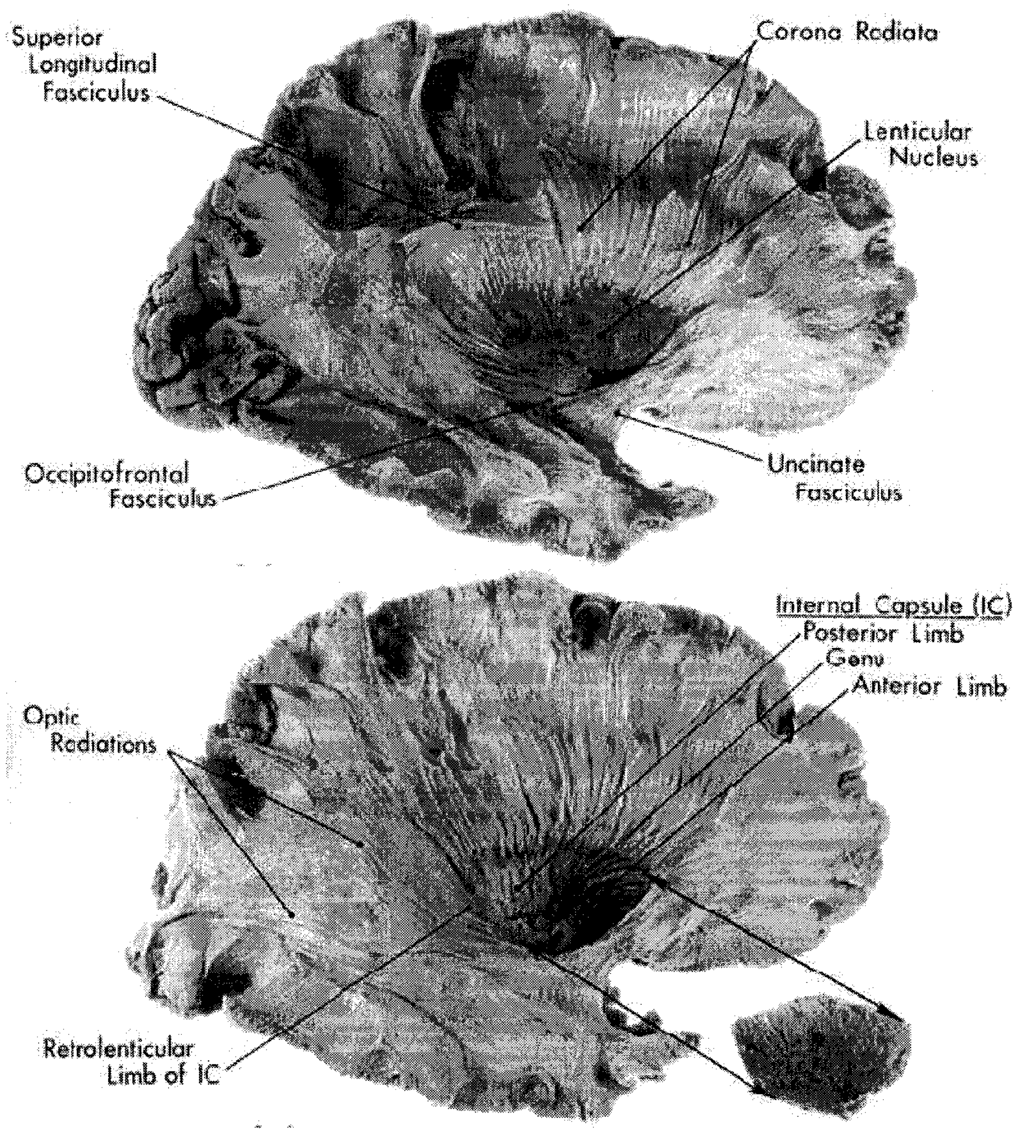

Figure 2.4: The directionality of white matter tracts is evident in these pictures. All regions labelled, except the Lenticular Nucleus, are regions of white matter. Reproduced with permission from Lippincott Williams and Wilkins.

2.3 Texture Description and Other Image Processing Techniques

Texture is the inter-relationship of grey levels within a region of an image. By definition, it is a spatially dependent quantity that cannot be measured at a point; rather it must be measured over an area. Texture can be thought of as having stochastic and deterministic components. The stochastic component is most frequently described in terms of probability distribution functions. The deterministic component can be thought of in terms of texture primitives which repeat. Examples of textures with primitives include brick walls or tile floors. All non-artificially created patterns are a combination of the two components. Figure 2.5 shows an example consisting of stochastic and deterministic components and the final texture.

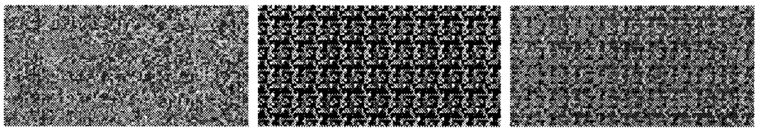

Figure 2.5: Left to right: Stochastic component, deterministic component, and final texture.

As texture is spatially dependent, the area over which to evaluate texture must be selected with care. A region over which an image measure is obtained is known as a region of support. Investigating a window that is too small results in the minimum repeated pattern (texel or primitive) not being identified. Evaluating a region of support that is too large may mean that smaller details may be overlooked, or that excess time is spent processing. In most real-world scenes, the evaluation of texture is a multi-resolution task requiring investigation over more than one support size to capture the full information.

There are many ways to describe texture. First- and second-order statistics capture some of the probabilistic and spatial relationships in texture. Texture can also be described in terms of basis functions or the output response of a set of filters designed to span a particular space. Measures are also available to capture the self-similar nature of texture over many levels of resolution. As yet there is no one group of measures that captures all needed information for all computer vision problems. The relevance of a particular texture measure is problem-specific and therefore it is in the researcher's interest to extract relevant features rather than to try to span the space of all texture. As texture descriptors and image processing is a large and varied subject, the discussion is limited to the texture measures and processing methods relevant to the review being presented and the methods used in this work.

2.3.1 First-Order Statistics

First-order statistics are a description of the shape of the histogram of a population, in this case the frequency of intensities within a window in an image. These statistics include mean, median, mode, variance, standard deviation, maximum, minimum, percentile value, skewness, and kurtosis. The mean is the geometric center of the distribution, the median is the value which splits the population in half, and the mode is the value that occurs most often. All three measures are indicators of central tendency. The variance and standard deviation measure the spread of a population, with standard deviation being the square root of the variance. Variance is the sum of the squares of the difference of each point from the mean divided by the number of samples in the population:

$$\sigma^2 = \sum (X - \mu)^2 / N \tag{2.2}$$

where μ =mean and σ =standard deviation. Maximum and minimum are simple point measures that can provide spread information by their difference which represents the range. A percentile value indicates where the tail of a given percentage of the population occurs. For example, the tenth percentile is the intensity level below which the lowest 1/10 of the distribution lies. Kurtosis provides a measure of the size of the populations tails:

$$kurtosis = \frac{\sum (X - \mu)^4}{N\sigma^4} - 3, \tag{2.3}$$

while skewness provides a measure of the asymmetry in a population:

$$skewness = \frac{\sum (X - \mu)^3}{\sigma^3}. \tag{2.4}$$

SNR is another common measure and is defined as $\frac{\mu}{\sigma}$. Glossaries of these and other statistical measures are available online [32], [33]. With first-order statistics, all information about the relative location of the intensities is lost. However, these measures can be cascaded in a two-stage system where the first stage emphasizes spatial relationships. Thus, first-order statistics can form part of an important texture measure.

2.3.2 Second-Order Statistics

Second-order statistics capture information about the occurrence of pairs of intensities. A traditional texture measure that captures this information is the co-occurrence matrix and its measures [34]. The co-occurence matrices contain counts of the number of times that a pair of grey levels occur in an image with a given displacement vector. Typically the row and column positions represent a grey level in the pair and the user of the texture analysis system chooses the displacement vector of the matrix (say x,y=1,0 for side by side along the x axis) and the number of matrices required. The resulting co-occurrence matrices can then be described by fourteen textural features. These features include angular second moment, contrast, correlation, variance, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, information measures of correlation (2), and maximal correlation coefficient. From these 14 measures, Haralick *et al.* propose that 28 features can be obtained. For a given distance d, there are 4 angular matrices $(0^o, 45^o, 90^o, 135^o)$. The 14 textural features are obtained over the 4 matrices. From each of the 4 values of textural features, the mean and range are obtained to provide 28 features.

A subset of the co-occurrence features is frequently used in textural analysis of images; however, part of the difficulty in their use is the determination of which matrices to obtain and which textural features to calculate. Over-describing the space results in excess computation time and potential difficulty when designing a classifier. Under-describing the space can lead to crucial analysis information being missed. Co-occurrence matrix parameters are popular in the analysis of US images. This popularity may be warranted, as there has been success in their use in ultrasound tissue characterization (UTC) and in the classification of tissue mimicking objects with similar properties [35].

Another second-order measure is the autocovariance function (ACVF). The autocovariance function is a measure of self-similarity.

$$ACVF = \int \int (A(x', z') - \bar{A})(A(x' + x, z' + z) - \bar{A})dx'dz' \tag{2.5}$$

where

A is the amplitude and

$\bar{A}$ is the mean amplitude over the integral region.

After obtaining the autocorrelation function, the resulting ACVF peaks can be described. Some chose to describe them by full-width-at-half-maximum height (FWHM) [36].

2.3.3 Change of Basis Functions

The most commonly known change of basis functions in computer vision, is the application of the Fourier Transform. The 2-D array of grey-level intensities is transformed to its 2-D sinusoidal components of varying magnitude and phase. This simple change in basis can reveal that a texture has dominant frequency components in a particular band, possibly helping to distinguish it from other textures. The Fourier Transform results in precision in frequency, but a complete loss of location information. For example, an impulse within an image has the same Fourier Transform regardless of its placement within the original image.

Wavelet transforms are another way to change to a different basis [37]. The appeal of wavelets is that they are localized in both the frequency and spatial domain. The discrete wavelet transform (DWT) successively decomposes the low frequency components, separating the lower frequencies into successively smaller groups. On the other hand, discrete wavelet packet analysis (DWPA) successively decomposes both the high and low frequency components. The wavelet coefficients at each decomposition can be evaluated and used for texture discrimination.

2.3.4 Searching for Primitives

Often the primitives that form the deterministic portion of the texture are difficult to isolate. However, isolating a sub-component and describing it may be a reasonable option.

Sub-components may include gradients, low-intensity regions and high-intensity regions. As well, in an attempt to isolate texture primitives, Voronoi tessellation algorithms may be applied on a simplification of the primitives to extract simplified shape information.

2.3.5 Groups of Filters

By the selection of the appropriate filter, textures with subtle differences can appear markedly different after filtering. This enhancement of texture differences can be used to allow for texture segmentation with the appropriate measures and thresholds. Groups of filters are often implemented. The groups may consist of linear filters, non-linear filters, or a combination of the two types.

In the application of linear filters, the group of filters may consist of the same type of filter applied at different spatial resolutions. With the addition of subsampling, multi-resolution techniques arise. The linear filters may be identical in size but vary in the spatial frequencies they target. This technique has been used to identify different textures with the purpose of locating weak edges that are formed in US images by texture changes [38]. Frequency separation methods are referred to as split-spectrum analysis. However, often the entire spectrum is not investigated, as the features of interest fall within a specific bandwidth.

In a manner similar to linear filters, a group of non-linear filters may be applied that performs the operation but at differing spatial scales. Some examples of non-linear filters include histogram-based filters (median, mode) and morphological operators. As an example, the effect of dilating a texture with identical kernels of varying size might be of interest. Again, this can be considered to be a multi-resolution analysis technique. Alternatively, a single non-linear filter may be applied successively and the change analyzed. The pectrum (pattern spectrum) is derived through the repeated application of "opening" or "closing" operators with a single kernel [39]. The result can be a measure of the texture at each repetition. A graph of the change may offer a texture measure of value.

When there is prior knowledge about the importance of various texture characteristics, the filters that form the group can be selected to emphasize those characteristics. The filter group can then comprise a reduced set of linear and non-linear filters with the size and features, shape, and sub-components refined for improved results. The same statement can be made about all of the groups of texture measures presented here. The key is in understanding the features that are important and identifying the appropriate methods to measure them.

2.3.6 Fractals

Fractals are patterns that are self-similar over a variety of scales. They may not have identical components over all scales but the shapes are similar over all scales [40]. The plot of a measured fractal quantity verses scale on a log-log graph is a straight line whose slope is the fractal dimension. The fractal dimension of an image may be estimated through box counting, fractal Brownian motion, and power spectrum techniques.

2.4 Image Processing

There are some image processing-related topics that are important in UTC that have not been covered in the texture discussion.

2.4.1 Resolution Reduction and Multi-resolution Techniques

Frequently, the image will contain detail or high-frequency components that are not relevant to the classification or segmentation problem at hand, or they represent unwanted noise. The most common method to deal with this issue is through resolution reduction, which is the filtering of an image and then sub-sampling the result. Resolution reduction is often preferred over filtering alone as it results in a smaller image and thus reduces computation time in subsequent processing steps. Filtering is a vital component of resolution reduction as without it aliasing may occur. Based on the Nyquist rate, aliasing will not occur if a baseband-bandlimited signal is sampled at more than twice the frequency of the highest frequency component of the signal. For a non-baseband signal, the signal can be recovered if it is sampled at a frequency of at least twice the bandwidth of the signal [41]. Therefore, filtering is applied before sub-sampling to reduce the bandwidth of the signal to less than 1/2 the sampling rate. There are options available in the application of the low-pass anti-aliasing filter. The filter is chosen with the understanding that there is always a trade-off between the filter properties in the spatial domain and its properties in the frequency domain. Common filter choices for filtering prior to resolution reduction include Gaussian, Laplacian and wavelets.

Repeated applications of filtering and resolution reduction lead to a multi-resolution respresentation of the images. The Burt-Adelson [42] pyramid is a multi-resolution representation using the Laplacian filter. Also popular is the application of wavelets, including the Daubechies family of wavelets [43]. A resolution pyramid using the Daubechies wavelets offers the advantage of allowing perfect reconstruction from the multi-resolution components, a feature not shared by all multi-resolution pyramids. Sajada *et al.* [44] provide a review of multiresolution and wavelet representations for the identification of disease signatures. This group stresses the need for analyses that identify short-duration unique signals that frequently identify the presence of disease. This is one of the desirable features of wavelet-based pyramids.

2.4.2 Adaptive filtering

The term adaptive filtering refers to a local modification of a filter due to image parameters within the filter window. The filter is frequently modified according to the local signal strength (SNR value, presence of edge gradients, etc.) or a classification of the local texture based on the problem at hand. Sometimes the modification of the filter is its complete suppression or the selection of an alternative algorithm. Adaptive filters may also be implemented such that the size or shape of the kernel or window under investigation is modified.

2.5 Experiments: Relationships Between Image Texture and Media Structures

Experiments have demonstrated that changes in scatterer density, size, and acoustic properties can change US B-mode image texture. While some of researchers have studied the effects of changing these variables on image texture, others have restricted their studies to FDS [50]. The traditional division between FDS and texture occurs when the scatterer density exceeds 10 per resolution volume [15] or when SNR $= 1.91$ [18]. First, a brief discussion of the properties of FDS is presented. Then the experimental findings showing the relation between US texture properties and the acoustic medium are discussed.

2.5.1 Fully Developed Speckle Characteristics

FDS is dependent upon media properties such as density and size. FDS characteristics are also dependent on the machine and transducer properties including center frequency, focus location, bandwidth of the transducer and distance from the transducer face.

In 1983, Wagner *et al.* [18] investigated FDS characteristics using two tissue-mimicking phantoms with high particle densities, randomly dispersed. The experiments on each phantom used three frequencies. They reported that FDS is a function of machine characteristics, as well as phantom characteristics, and that the intensity distribution follows a Rayleigh probability distribution function (pdf). The value of SNR for the Rayleigh distribution is 1.91. The Rayleigh distribution is a special case of the Rician distribution. As opposed to media with completely random particles, media with a structural component have image textures characterized by a Rician pdf and an SNR over 1.91. The first- and second-order characteristics of the Rayleigh and Rician pdfs differ and may reveal information about the media.

Speckle, including FDS statistics, have also been described as K-probability distribution functions [46] and Nakagami distribution functions [47]. The K-distribution is suggested when the number of scatterers is low or when the scattering cross-sections vary. The Nakagami model is proposed as a less computationally expensive alternative to the K-distribution.

Studies have been performed to observe the effect of an imaging system's impulse response on image speckle and texture statistics. The presence of FDS is dependent on the number of scatterers per resolution cell. The volume of material under investigation is dependent on the impulse response of the system and the location of the tissue with respect to the beam focus. In experiments by Rao *et al.* [48], the effect of varying system parameters on the outcome of image texture, whether it be FDS or texture, was examined. The SNRs of 3 different tissue models were obtained for 7 different pulse parameters. The authors concluded that SNR is dependent on the effective volume of the system, which is a measure of the system point-spread function. SNR values for some systems acted as a measure for discrimination while others did not. The authors concluded that care must be taken in extracting measures from different imaging systems.

FDS has also been observed to be dependent upon tissue properties. Wagner *et al.* [49] state " ...either speckle contrast ... or speckle-cell size, would serve as a tissue signature corresponding to the value of the specular-to-diffuse scattering ratio. This is only true when there are many diffuse scatterers per resolution cell." Conventional wisdom suggests that FDS occurs when the density of scatterers is large with respect to the resolution cell and scatterer size is small with respect to the wavelength [50]. However, the incorporation of FDS measures in UTC should be done with caution. The relationship between FDS and tissue structure is not unique in B-mode scans. Recently, in 2005, Dantas *et al.* [51] published an article in which they showed that the speckle patterns from a large number of randomly distributed scatterers can be replicated by a medium containing a much smaller number of periodic scatterers. They reported that it is possible to reduce the number of scatterers to 6.4% of the original phantom and maintain the original speckle pattern. This conclusion is based on the error between the original envelope detected signal and those created with the equivalent scatterers. The error level was found to be 0.5%.

2.5.2 Texture Properties as a Function of Media Structure

One of the most significant publications in this area is by Oosterveld *et al.* [36] in 1985 . This group demonstrated the variability of speckle statistics with transducer properties and scatterer density through experiments incorporating simulations and phantoms. They measured the first-order statistics of mean-intensity amplitude and SNR. The second-order statistics include a measure of the speckle size in the lateral and axial directions. The measure of speckle size is provided by generating the autocovariance function and then obtaining the FWHM. For a single experiment with scatterer density of 1000 per cm^3 and constant transducer characteristics, the mean amplitude, SNR, and lateral FWHM are depth dependent. The axial FWHM is independent of distance to the transducer. In experiments where the scatterer density varied from 100 to 19000 per cm^3 , the first- and second-order statistics were found to be dependent on density.

In 1986, Wagner *et al.* [52] presented experiments in which US texture was created through simulations with varying amounts of structured components with a constant diffuse component. They found that second-order measures, based on the autocorrelation function and the Fourier power spectrum, provide a means to classify whether texture in an image originated from the same medium as a reference image.

In 1987, Wagner *et al.* [53] continue researching the relationship between media structure and texture properties. They propose that the type of acoustic medium, whether it contain few diffuse scatterers, many diffuse scatterers, unresolved coherent component or resolved coherent component, can be determined from first- and second-order analysis of B-mode images. They also note that this holds true only if trends and inhomogeneities have been removed (such as blood vessels).

US texture as information is supported again by Thijssen *et al.* in 1990 [50]. The aim of this work was to "... enhance the transfer of scientific results to the medical field." While some of it is review and some tutorial, new results are presented. Of primary significance to this research is the demonstration that the relative scattering strength of scatterers changes the B-mode texture. Thijssen *et al.* reinforce the concept that while speckle is not an image of the histologic structure of tissue, characteristics of the tissue under investigation affect image texture properties. They also make it clear that the machine characteristics of frequency, TGC, gain, and the transducer characteristics such as bandwidth all have an effect on B-mode texture.

Furthering the connection between B-mode texture and underlying structure is a paper by Jacobs *et. al.* in 1991 [54] . In this work, tissue was modeled as having a diffuse scattering structure (7500/cm^3) and a structural component (1 mm spacing). Then, the model was subsequently modified to observe the correlation between media properties and B-mode texture measures. The first- and second-order statistics of mean amplitude, SNR and FWHM in the axial direction were obtained as texture measures. The simulations employed medium with the above diffuse and structural components and then they modified it by adding a degree of randomness to the structure portion. The randomness was added by introducing positional uncertainty in the structural component. This uncertainty was expressed as a percentage of the 1 mm spacing, and the experiments covered 0, 5, 10, and 20 percent. Beyond the effect of regularity of spacing, the effect of relative scattering strengths was changed. In the experiments, the structural component's relative scattering strength was varied from 0.3 to 1.50 as compared to the background scattering strength. The authors observed that these changes affect texture. They provided graphs illustrating the amplitude of each measure for various combinations of positional noise and relative scattering strength. The structural-component spacings are within the resolution of a system that is operating at a 3.5 MHz center frequency. Jacobs *et al.* also discussed the dependence of the orientation of the pulses with respect to the structure with the finding that texture measures were dependent on orientation. This is highly relevant to studies of organs such as liver and kidney, where the examination is made up of images with the transducer located at several locations on the body's surface.

Finally, there is a thorough review of texture properties as a function of media structure by Thijssen in 2003 [6]. This provides an overview of the physics of B-mode imaging, texture and its relation to media structure, processing in the formation of images and a basic understanding of UTC. This work would be very helpful to anyone new to UTC.

The mechanism of the disease, the physics of the US image formation, and the evidence provided by past experiments all suggest that there is diagnostic information to be exploited within the texture of US images. Others have had some success in the application of UTC to diseases in other areas of the body and in the brain. A review of this work follows.

2.6 Ultrasound Tissue Characterization in B-mode images

2.6.1 Parenchyma, Other than Brain

B-mode US image texture has been investigated for disease detection in many soft tissues. UTC is an area that includes analysis of the RF signal only, B-mode data only, or a combination of RF and B-mode information. As this research incorporates B-mode image data only, discussions of UTC will be limited to those experiments that use B-Mode data only.

In this section there is a review of the research in other tissues. While there are non-trivial differences between the neonatal brain and these other organs, all are soft tissues, and disease processes within them disrupt cell function and tissue architecture. An overview of the work in these other areas provides insight into which techniques have potential and those that may not.

Breast cancer is frequently a candidate disease for ultrasound tissue characterization. Huber *et al.* [55] investigated texture measures and B-mode characteristics for their ability to assist in the differential diagnosis of solid breast masses. The study included 77 patients, 27 of whom had malignant lesions. The authors concluded that the qualitative B-mode characteristics outperform the UTC measures; however, the contribution of each measure was evaluated through statistical tests, not classifier design. The study was performed at 2 centres and the results varied between the two centres. One point to note is that the correlation measure from the co-occurence matrix consistently contributed information to diagnosis.

Kutay *et al.* [56] [57] used parameters from two models to describe ultrasound scattering. The first model is the narrow-band, power-law, shot-noise model which treats the tissue as a collection of point scatters embedded in a uniform medium. Attenuation of the ultrasound signal in this model is a power-law decay $1/t^{\nu}$, that persists in the RF envelope and allows parameters of this model to be extracted from the B-scan images. The second model, the K distribution, is used to model tissue as if it contains scatterers with variable concentration and non-uniform cross-section. Four measures are used to describe the tissue, two from the shot noise model, one from the K distribution model, and SNR.

SNR is measured to indicate whether the histogram of intensities over a region is better described as a Rayleigh or K distribution.

For each of 100 clinical images, 20 ROIs were selected within the tumors and the 4 parameters were estimated. The value of these measures for discriminating between benign and malignant tumors was assessed through the use of receiver operator characteristic (ROC) curves. While this group includes the calculation of some power-law parameters from the RF signal, the best measures, according to the ROC curves, were extracted from the B-scan image only. The SNR and the power-law decay combine to give an area under the ROC curve of 0.889. An ideal ROC curve has an area of one.

Chang *et al.* [22] proposed a system to discriminate breast tumours using texture analysis through the use of support vector machines (SVM). The experiments utilized 250 ultrasonic images that included 140 benign breast tumor images and 110 carcinoma images. The same US machine was used to acquire all images in this work. ROIs for the tumors were extracted by a physician and then the texture measure extraction and analysis was performed by the computer-aided diagnosis (CAD) system. All areas within the tumor images were classified as speckle or not according to the ratio of mean intensity to standard deviation. For each pixel that was considered speckle, the autocovariance coefficients were calculated according to:

$$A(\Delta m, \Delta n) = \frac{1}{N_S} \sum_{x=0}^{M-1-\Delta m} \sum_{y=0}^{N-1-\Delta n} S_1(f(x,y) - \bar{f}) S_2(f(x, +\Delta m, y + \Delta n) - \bar{f}) \qquad (2.6)$$

where

S_1 is 0 if $f(x, y)$ is not a speckle and 1 otherwise,

S_2 is 0 if $f(x + \delta m, y + \delta n)$ is not a speckle and 1 otherwise,

$\bar{f}$ is the mean value of $f(x, y)$,

N_S is the total number of speckles.

For each ROI, a 5×5 speckle covariance matrix was created. A non-linear support vector machine (SVM) with a Gaussian radial-basis kernel was used as the classification system. The system was designed with the 250 images and then an error estimate was derived using an N=5 cross-validation. This group reported a 93.2 % accuracy. They also reported that their system, using texture measures with speckle emphasis, performs better than texture measures of all pixels or texture measures with non-speckle emphasis. They argued that for CAD systems, speckle contributes information.

Alacam *et al.* [58] modeled the ultrasonic tissue response as a fractional differencing process (FD). The FD model contains a correlation parameter, d, that is calculated from the image data. The data set consisted of breast scans of 60 patients, 29 malignant and 31 benign. For each image, 30 scanlines were obtained from inside and outside the tumor, and from each d_{FD} is estimated. Using d_{FD} and the patient's age, a quadratic classifier was created for classification. The area under the ROC curve was calculated as 0.8334.

Chang *et al.* [59] proposed a method to classify breast lesions using fractal analysis of texture. The algorithm consists of image pre-processing to reduce noise, standardization of intensity levels, extraction of the fractal dimension based on fractal Brownian motion, and classification using a k-means classifier. This group used 250 images: 140 benign and 110 malignant, one image obtained from each patient, all obtained from the same US machine. Using cross validation to measure the performance, the system provides a 88.8% accuracy.

As demonstrated by this review, there is significant research into the application of UTC for breast cancer in journals. For the diagnosis of breast cancer, there is now a commercial CAD system available using UTC [60]. The recent integration of CAD using UTC for breast cancer diagnosis suggests that UTC has the potential to provide more tools to assist medical practitioners.

Diseases of the liver have also received a significant amount of attention. Kadah *et. al.* [61] investigate texture measures to classify liver images into either normal or displaying cirrhotic or fatty liver diseases. The experiments were performed using images from 120 cases taken before needle biopsy. The texture parameter-set consisted of 8 measures from each image ROI including 2 first-order measures (mean, first percentile of grey-level), 4 co-occurrence matrix measures (contrast, entropy, correlation, angular second moment), an attenuation estimate, and a scatterer separation distance measure. The group then investigated 8 different classification systems, applying all 8 texture measures in each case. The classifiers included: a minimum distance classifier, Bayes quadratic classifier, a kNN classifier, single-layer perceptron network, a multi-layer perceptron network, a single-layer perceptron network with functional link inputs, and a multilayer perceptron trained with cluster centers only. They concluded that the single layer perceptron network with functional link inputs and the kNN systems perform the best. While this group made reference to an improvement in classification being obtained by systems that can create complex decision surfaces, no further comment on the classifier selection process was provided.

Mojsilovic *et al.* [62] applied the quincunx nonseparable wavelet transform to US images of the liver for the classification of diffuse diseases. The spectral decomposition differences in this wavelet transform offer some advantages for UTC in B-Mode images. The separable sampling with other wavelet transforms provides rectangular divisions with increased sensitivity to horizontal and vertical edges (so they are rotationally sensitive). Mojsilovic *et al.* claim that most natural textures have their energy concentrated in the mid-frequencies, which the quincunx preserves better, and they claim that the diamond shape of the low-pass filter removes more of the noise. This group used images from one machine with 122 images for training and 122 images for testing. The 244 images were obtained from 30 subjects, and thus the samples are not independent. Each data set contained the members of the 3 classes equally. From each wavelet channel, a pdf is calculated from which the variances are calculated and used to train the Bayes classifier. An accuracy of 90% was obtained. To check the dependence of their measure on rotation, the image samples were rotated at 5, 23, 45 and 90 degrees and an accuracy of 88, 88, 80, 90 was still obtained without re-training the classifier.

Gangeh *et al.* [63] presented a fuzzy-based texture analysis of diffuse liver diseases. Diffuse disease is of special interest, as there is frequently no healthy reference tissue within the same organ. This causes difficulty in establishing the absence or presence of disease. Since texture in B-scan images is dependent upon the ultrasound machine properties, Gangeh *et al.* maintain the machine settings for all images in their experiments. Their algorithm begins by subdividing the image into smaller subimages, and then it assesses the membership of every pixel with respect to its neighbor based on grey level and a triangular fuzzy membership function. After filtering, each texture sub-image is reduced to a set of descriptors that include maximum, entropy, and uniformity. After obtaining the texture measures, fuzzy C-means clustering is used for classification. Unfortunately the authors did not provide a quantitative analysis of their results and use only limited data, so an evaluation of their system's performance is not possible.

Akiyama *et al.* [64] investigated fractal dimension from Brownian Motion as a means to identify liver as normal, fatty or exhibiting cirrhotic. The authors reported favorable results, however the data set is small (21 in total) and the evaluation was based on the separation of the mean fractal measures of each group rather than the development and evaluation of a model.

Lee *et al.* [65] used an estimation of fractal dimension to classify liver images as normal, or displaying cirrhosis or hepatoma. They employ a modification to a differential box-counting technique that reduces the effect of noise. This group designed and tested a Bayes classifier using 216 images in each of the training and test sets. They report a 95.4% accuracy for their system.

Yoshida *et al.* [66] investigated wavelet packet-based texture analysis for differentiation between benign and malignant liver tumors. The images consisted of 17 benign and 27 cancerous tumours, from which 50 benign and 145 cancerous ROIs were extracted. The texture features were calculated to measure the homogeneity of the textural pattern with multiscale analysis. For classification, artificial neural networks (ANNs) were applied. Speckle was reduced from the ROIs and the image was "background-trend" corrected to reduce dependencies on machine settings. From each wavelet decomposition, the texture features that were extracted from the ROIs included Shannon entropy, the root-mean-square (RMS) of intensity, and the first moment of the power spectrum. The number of texture features were reduced via a backward elimination method, that successively eliminates features as long as discrimination performance is not reduced. This group reported an accuracy of 92%. The error measure was calculated using the jackknife method.

There are many other organs analyzed through quantitative US techniques. Just a few are included to demonstrate the variety of medical conditions where this field of study can have an impact.

Smutuk *et al.* [67] investigated texture measures of the thyroid gland for the detection of Hashimoto's lymphocytic thyroiditis (HLT). Thirty sonograms were obtained for each of the 60 patients and 39 controls, using the same machine and identical console parameters for all, for a total of 2970 images. An interactive tool was used to extract the ROI containing the gland from each image. Each ROI was sub-divided into windows sized 41 × 41, 31 × 31, and 21 × 21 pixels and labelled according to patient diagnosis. They extracted 129 texture measures from each window, 21 spatial features and 108 co-occurrence matrix features. This group designed a Bayes classifier using a design set of 81 patients and a test set of 18. As part of the design process, the group reduced the number of texture features through a compactness and separability criterion, and a minimal classification error criterion. The group identified three texture measures as optimal, a different texture measure from each of the three window sizes. This group presented a result of 100% classification success confirmed with both an independent test set and cross-validation.

Maeda *et al.* [68] investigated the ability to obtain quantitative measurements of placenta from 3 different sonographic machines to detect maternal and fetal disorders. This group used the grey-level histogram width (GLHW), which is the range of the grey-levels within the ROI divided by the full grey-scale, expressed as a percentage. Before investigating placental measures, the group investigated the effect on GLHW of the various device gains, contrasts, and the depth from the transducer face to the ROI. The GLHW remained constant over varying gain and distance of the ROI from the phantom surface, while it had a linear relationship with the contrast setting which can be corrected. This group obtained GLHW measures from historical data in the form of 222 normal sonograms. They then went on to obtain GLHW measures from 44 new subjects (37 normal and 7 abnormal). The abnormal measures fell outside the normal measures from the historical data, based on a mean $+1.5\times$ standard deviation criterion.

Christodoulou *et al.* [69] investigated the use of texture measures in cartoid artery sonograms to identify the risk of stroke. Their work consists of obtaining and standardizing images of carotid plaques, segmenting areas of interest, extracting texture measures, and building classification systems. The images were obtained with a single ultrasound machine, but the operator-controlled machine settings were freely adjusted to obtain the desired image quality. To compensate for the variation, the gray-levels of the manually extracted areas of interest were adjusted so that the intensity of the blood was within 15-20 and the artery wall was within 180-200, out of the image range of 0-255. The 61 texture measures obtained were grouped into 10 feature sets based on the methods obtained to extract them. The methods included first-order statistics, co-occurrence matrices, grey level differences, neighborhood gray tone matrix, statistical feature matrix, Laws texture energy measure, fractal dimension, Fourier spectrum, and shape parameters of the segmented images. The authors selected the top-ranked 15 individual texture measures, graded according to their ability to separate the two classes. They also attempted to reduce the number of features through the sequential elimination of features that did not contribute well to classifier performance. In the end, they found that these methods performed sub-optimally compared to creating a single classifier for each of the 10 groups and allowing the classifiers to vote. Two multiple classification schemes were investigated, one using artificial neural networks and one using K-nearest neighbors. They reported that the multiple classifier using artificial neural networks performs the best, with an accuracy of 73.1 %. The accuracy was obtained with a separate test set and with 5 bootstrap sets used confirm the results.

2.6.2 Non-PVL Brain Investigations

In relation to this research, the advances in UTC in the brain hold the most interest. While the focus of these investigations may differ, infant brain studies are more closely linked to this work due to the similarity in the types of images obtained and the tissue under investigation.

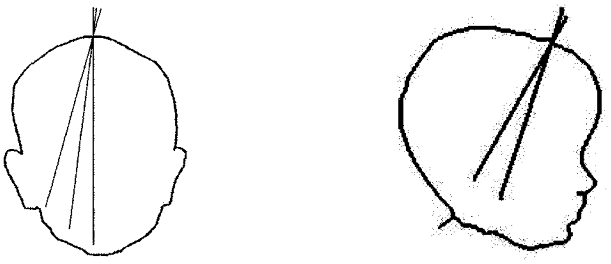

Figure 2.6: Insonation paths for neonatal cranial US. Left: The sagittal planes. Right: The coronal planes.

Barr *et al.* [70] investigated texture measures to determine if they are indicators of term infants having had clinical hypoxic episodes. Their data set consists of sagittal and coronal images from 25 patients, 9 of whom had an hypoxic episode. See Figure 2.6 for an illustration of coronal and sagittal planes. A single sonographic machine was used for all images, but the sonographer was free to modify the machine settings as he or she saw fit. To obtain a measure of the effects of the settings for each patient, a tissue-mimicking phantom was imaged prior to obtaining the images from each patient. Forty-five candidate texture measures were extracted which consisted of both first- and second-order grey-level statistics. The first-order statistics included mean intensity, upper and lower 10th percentile ranges, variance, and skewness. The second-order statistics included gradient distribution analysis, co-occurrence matrix analysis, run-length histographic analysis, and fractal features analysis. Run-length histographic analysis is a measure of the homogeneity of an ROI based on the length of a grey-level run. In each patient, four to six ROIs, sized 40 × 40 pixels, were extracted. For each patient, the average textural value was obtained for a given anatomical location. This group then continued to evaluate the texture measures and brain ROI locations for discrimination ability through the use of logistic regression. Nine of the 45 measures differed with patient population, with $P < .04$. Four of the texture measures are first order: mean, skewness, and upper and lower 10th percentile. The remaining five are from the co-occurence matrix: 1 entropy measure, 1 contrast measure and 3 correlation measures. The average grey levels, or echogenicity, of certain anatomical areas of the brain was found to provide the greatest difference between patient populations.

Mullaart *et al.* [71] investigated quantitative US measures related to maturity of the neonatal brain. This study investigated properties of white matter and grey matter regions taken from coronal views of 39 term and preterm infants. From these regions, four texture measures were investigated for their correlation with gestational age. The measures included mean grey level, SNR, axial correlation, and lateral correlation. This group obtained images of phantoms with the same machine settings to allow images to be re-scaled for comparison. The Wilcoxon rank test and linear regression were the statistical methods used to assess the correlation of the image measures with the maturity of the brain. The group found that gestational age affects image parameters. They recommended mean intensity, the left right brain mean-intensity ratio and the white grey matter intensity ratio as measures suitable for clinical use.

In 2000 Valckx *et al.* [72] reported on a calibrated parametric imaging technique. Much of the paper discusses the method rather than its application. The method standardizes the images with respect to the transducer and beam-forming characteristics, then obtains textures features within sliding windows and outputs each feature to a separate parametric image. This paper is included in this section because all of the clinical images used are of the neonatal brain and discussion is presented on how these parametric images could be combined with classifiers to detect brain pathology. Searches for follow-up work by this group that incorporates these ideas have been unsuccessful.

2.6.3 Detection of PVL

Stippel *et al.* have published a number of reports on their work in the area of improved PVL detection through the automated extraction of flare outlines within cranial ultrasound images. As the first three papers [73], [74], [75] are intermediate reports on their work as it evolved, this discussion is limited to their methodology as presented in the final 2 papers [76], [77].

The work published in 2002 [76] concentrated on an adaptive denoising technique. Generally, their procedure consists of segmenting speckles separately, adaptively filtering the speckle, and then segmenting the flares using an active contour technique. Prior to segmenting the speckle, this group standardizes the images to correct for scanner settings. They use an algorithm proposed by Simaeys *et al.* [45] which uses an US model incorporating the frequency, the power, the gain, the TGC, and the dynamic range of the logarithmic compression scheme. From this model, an inverse model is created that allows image standardization. Once Stippel *et al.* standardize the images and mean filter to suppress outliers, seed pixels for speckles are identified from the image. Individual speckles are segmented through a region-growing technique. The region growing is guided by a top-hat-transformed image using a speckle-shaped morphological operator. The top-hat transform is an image which has been morphologically opened and subtracted from the original image. After segmentation, each speckle is classified as noise or feature based on the texture parameters of the image in the surrounding neighborhood. These parameters include contrast and mean grey level. If the speckle is considered noise, it is replaced by the mean of the surrounding area; if it is not, the speckle is left unchanged. This group compared the performance of the filter and subsequent application of a snake algorithm for flare extraction to the Lee and Frost filters. In this work, the comparison was performed on only two images which does not allow for an assessment of its ability.

The work published in 2005 by Stippel *et al.* [77] proposed a tissue-specific adaptive-texture filter (ATF) that can be used as a visual aid for diagnosis by clinicians, and as a preprocessing technique prior to the automatic extraction of flares. The goal of the applied filter is to enhance the difference between the texture of healthy and PVL-affected neonatal brain by adaptively modifying the image texture. As with previous work, the images are first standardized using the algorithm by Simaeys *et al.* [45], and then mean filtered to suppress outliers. Using the region growing technique outlined in [76], segments (previously referred to as speckles) are identified. The segments are designated as being within pathological, normal, or transition zones as determined by the mean and contrast measures of local windows. Based on a pathological classification, the region is "lightened" (intensity added). The classification of healthy or transitional leads to a "darkening procedure" (intensity subtracted), with differing algorithms depending on the classification. Each of these filtering methods contains a manually controlled parameter to adjust the degree of intensity change.

After the filtering is complete, flares are segmented using two separate contour algorithms. The contours are compared to both manual segmentation and contours grown after pre-processing with other filters. The comparison is performed using measures of match between contours Φ_M and precision Φ_p. The authors measure precision by

$$\Phi_p = \left(1 - \left(\frac{N_{diff}}{N_M}\right)\right) \times 100\% \tag{2.7}$$

where

Φ_p is the precision ratio,

N_{diff} is the number of pixels located between the expert contour and the automatically created contour,

N_M is the total number of pixels enclosed by the expert contour.

The match rate is measured by

$$\Phi_M = \left(1 - \frac{|O_M - O_W|}{O_M}\right)\% \tag{2.8}$$

where

Φ_M is the match rate

O_M is the area enclosed by the expert contour,

O_W is the area generated by the automatically generated contour.

After application of all techniques to 10 cranial ultrasound images, the authors concluded that the preprocessing technique improves the extraction of white matter flares. The average precision rate is 45% and the average match rate is 65%. These values are increased from 9% and 24%, respectively, before filtering.

An evaluation of texture as a predictor of WMD has been published by Vansteenkiste *et al.* [78], who is from the same laboratory as Stippel. This work investigates texture measures from non-standardized images, using rectangular ROIs selected manually from images containing flares. This group reports good results, however, feature selection, classifier design and validation are all performed using a single data set. The concern with using a single data set for all steps is that the results are over-fit and thus do not generalize well.

2.7 Image Processing and Coherent Imaging Techniques

This work investigates texture processing methods in coherent imaging. Many researchers have considered FDS to be noise and thus significant effort has been made to isolate and remove FDS and speckle. Many authors use the term speckle to refer to the grainy nature of the image. In this review of image processing applications in coherent imaging, the term speckle refers to the grainy nature of the image. This was chosen because in many cases the image statistics have not been analyzed to determine if speckle as texture or speckle as FDS is being isolated by the filters.

Much of the work presented in this section is for the purpose of speckle reduction without removing other image features. In this research, the goal is to extract texture measures, including some specific to speckle. Some methods that successfully remove speckle may provide insights into the best processing methods to preserve speckle characteristics.

2.7.1 Review Papers

In 2000, Dong et. al. [79] reviewed speckle filters for synthetic aperature radar (SAR) images for the purpose of texture preservation. While SAR images the far field and medical ultrasound images the near field, filtering techniques can be useful for processing both types of images. They discuss Lee, Kuan, Frost, mean, median and edge-sharpening filters. Lee and Kuan filters are considered adaptive-mean filters since for both, the value of the central pixel in the moving window is the mean plus an adaptive value times the difference between the mean and the center pixel. The equation of the two filters can be expressed as

$$\hat{x} = \bar{x} + K(I - \bar{x}) \tag{2.9}$$

where

$\bar{x}$ is the local mean of the moving window,

I is the value of the pixel in question,

K is a value between 0 and 1.

These filters vary in how they calculate the adaptive value but for both it approaches 0 for uniform windows and 1 in areas where there are edges. The Frost filter is an adaptive weighted-mean filter in which the influence of the neighboring pixels decays exponentially with distance, and with the parameters of the decay dependent upon the local statistics in the window. The mean and median filters are self explanatory, with the center-pixel value being replaced by the mean and median pixel intensity values respectively. The edge-sharpening filter first locates edge crossings and then assigns to the center pixel the value of the edge or the mean value of the moving window depending on whether the edge is present or not. The authors found that all filters distort the texture at the expense of removing the speckle from the SAR images. They determine that the median filter causes the greatest distortion and is not suitable for SAR images.

2.7.2 Wavelets

In 2000, Duskunovic *et. al.* [80] present two wavelet-based techniques to suppress speckle, both of which are based on selective reduction of the detail coefficients after discrete wavelet decomposition. Both methods make a judgment about the presence of "real" or "false" edges and adjust the detail coefficients based on the decision. The first method incorporates prior probability information for the decision, while the second method uses spatial filtering techniques. The methods are intended to suppress speckle noise for further analysis of neonatal cranial ultrasound images. Limited experimental data are provided to support this algorithm, so it is difficult to assess its contribution.

In 2001, Fulin *et. al.* [81] propose a wavelet-based filtering technique to reduce speckle noise in SAR images. This technique may be applicable to ultrasound images because the authors perform a logarithmic transformation prior to manipulating the images to cause the speckle noise to be converted from a multiplicative to additive form. (In clinical ultrasound systems, the data are logarithmically compressed prior to presentation to the end user.) In their paper, they propose an algorithm in which the logarithmically compressed images are wavelet decomposed and the coefficients of the detail-images are attenuated if they do not belong to edges. An initial threshold determines the attenuation coefficient used on non-edge pixels. The authors determine if a detail image pixel belongs to an edge through the use of directional masks that vary depending upon the subimage being investigated. An interesting conclusion that they reach is that while seven different wavelets were used in the algorithm, all produce very similar results in the application of speckle reduction.

Wavelet-based techniques, like all multiresolution techniques, are very useful for texture evaluation. These groups use wavelets as part of adaptive filtering, using edge detection as filter control. The methods described here were not pursued due to this reliance on edge detection.

2.7.3 Adaptive Filters Based on Local Image Statistics

Kotropoulos *et. al.* [82] proposed nonlinear image processing of ultrasound images implemented through the use of signal-adaptive filters applied to portions of images segmented by neural networks. This group looks at the difference between speckle properties for both raw B-mode data (Rayleigh random variable) and displayed US image data that has undergone logarithmic compression (zero mean Gaussian random variable). The output of the signal-adaptive filter is

$$s(\hat{k},l) = \hat{s}_M L(k,l) + \beta[x(k,l) - \hat{s}_M L(k,l)] \tag{2.10}$$

where

s is the original image without corruption,

$\hat{s}$ is the output of the filter at k,l,

$\hat{s}_M L(k,l)$ is the maximum likelihood estimate of $s(k,l)$, which they consider to be the low pass component of the image,

β is a weighting factor approximating SNR in the window,

$x(k,l)$ is the image intensity at k,l.

Kotropoulos *et. al.* conclude that for both types of ultrasound data, the maximum likelihood estimate is the L_2 mean scaled by a factor of $\frac{\sqrt{\pi}}{2}$. Explicitly, the filter is

$$\hat{s}_M L = \frac{\sqrt{\pi}}{2}\sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2}. \tag{2.11}$$

β varies in magnitude and determines the size of the filter window. (Note: The definition for the L_2 mean provided by the authors is identical to the RMS value of the region.) The authors conclude that this adaptive filter outperforms some simple methods to detect lesions such as preprocessing with mean or median filters and then thresholding, however their best results indicate positive predictive values of only 61 %.

Evans *et al.* [83] present a truncated median filter. The output of this filter is skewed toward the mode of the pixel intensity values, within the window of the image under investigation. The authors chose the mode as a basis for their filter because of their derivation that the maximum likelihood estimator of an image degraded by noise with a Rayleigh distribution is the mode. They also argue that the mode filter reinforces edges. The output of the truncated median filter results from first truncating the window histogram so that the median bisects the range of the remaining values and then a new median is calculated. This routine may be applied as an iterative process but the authors have chosen to apply it only once. Following the processing, Evans *et al.* use a Canny operator to identify edges as the basis for further comparison using both *in vivo* and phantom test images. They compare the detected edges after processing with their filter as well as the images pre-processed with median, adaptive weighted and unsharp-masking filters. Unfortunately, there is no gold standard for the *in vivo* images so that analysis is of limited use. Using the phantom with its known edge locations, the truncated median filter outperforms the other filtering methods.

Kofidis *et al.* [84] continued along the same theme as Kotropoulis *et al.* [82] with neural network segmentation and adaptive filters. They used the L_2 learning vector quantizer (LVQ) neural network, which is a self-organizing network. They then applied L-filtering to the approximately uniform sub-regions. They used Bovik's definition of the L-filter as a linear combination of input order statistics. One of the reasons they choose this filter is that it can cope with nonlinear models, which is appropriate since they model speckle noise as multiplicative Rayleigh-distributed noise. The L_2 LVQ ANN segments, based on a 7 × 7 window, and the filtering occurs in a 3 × 3 window over the entire image. This group compares their work to a number of other speckle filters including the homomorphic filter, Frost filter, sigma filter, variable-length median filter, and Taylor filter. The homomorphic filter is presented as

$$\ln x = \ln s + \ln n \tag{2.12}$$

where x is the image,
s is the original signal,
n is the multiplicative noise. The noise is then additive and averaging several images reduces the effect of the noise.

The Sigma filter averages those pixels which are within 2× the standard deviation away from the central pixel, rather than all pixels within a window. The variable-length median filter has a window size of 2L × 2L, where L is dependent upon the standard deviation and mean intensity values within the window. The Taylor filter is $\hat{s} = x^a$ where $\hat{s}$ is an approximation to s, the original uncorrupted signal, and

$$a = \frac{E\{s \ln s\} - E\{\ln x\}}{E\{ln^2 x\}}. \tag{2.13}$$

The method presented in this work outperforms all filters except the Frost filter when assessed using ROC curves and the SNR after processing. The ROC curves are created with respect to the ability of the system to correctly classify pixels as lesion or background. However, based on qualitative assessment of images, they conclude that their filter outperforms the Frost filter since the Frost filter reduces lesion contrast and adds blurring.

In 1996, Dutt *et el.* [85] investigated how log-compression affects the statistics of the echo envelope. They are concerned about this as many adaptive filters rely on the model of speckle statistics used to determine how to vary the amount of smoothing performed. This group argues that, based on what is known about speckle formation, the Rayleigh distribution is appropriate when there are many scatterers per resolution cell. Conversely, when the scatterer density is small, the K distribution more closely approximates the statistics of the echo envelope. The problem with the K distribution is that there does not exist a closed-form expression when the logarithmic compression of the image is considered. They proceed to derive an approximation that can be solved and then use this to develop a controlling parameter for an unsharp masking filter. They test their filter on a phantom and on abdominal images and they conclude that the filter preserved the object boundaries.

Ghofrani *et al.* [47] proposed an adaptive unsharp masking filter using the local statistics to control the filter parameter. In this case, the authors argue that the statistics of the echo envelope follow a Nakagami distribution rather than a Rayleigh or K distribution. Based on their assumption of a Nakagami distribution, the authors determine that an appropriate control parameter for their unsharp masking filter is

$$k = 1 - \hat{f}_m \tag{2.14}$$

where

$$\hat{f}_m = \frac{\hat{D}/4}{variance} \tag{2.15}$$

and $\hat{D}$ is the estimated dynamic range of the imaging system's logarithmic compression.

Suvichakorn *et al.* [86] proposed a new filter based on fitting a least-squares polynomial to the underlying image intensities within a window. The authors compare the output of this filter to the output of a median filter, both of which are then processed with the Canny edge detector for the extraction of thyroid edge-boundaries in US images. The results appear to be comparable. The significant difference between the two filters is that the processing time required by the polynominal fitting filter increases linearly rather than as a function of $Nlog_2N$.

In 2000, Tsubai *et.al.* [87] proposed a filter that uses morphological operations with adaptive structuring elements. They introduce an alternating sequential filter (ASF) that applies opening and closing to the image iteratively k times using a round structuring element with a radius that varies from 1 to k. The individual pixel weights within the kernel are dependent on the range of image-pixel intensities in the kernel neighborhood. As with most other suppression techniques, their goal is to remove noise while preserving edges. It is difficult to comment on performance as the results on only a single lesion phantom are provided.

2.7.4 Adaptive Filters Guided by Detected Edges

Dong *et al.* [19] proposed a new filter for SAR speckle reduction and edge sharpening. This filter smooths local variations, through the use of the mean, in areas where there are no edges. It does not perform averaging across edge boundaries. The edges are detected using wavelet-transform techniques with a second-order derivative of Gaussian function. Due to the discontinuous nature of SAR images, edge detection is performed by 1D convolutions in the horizontal, vertical, and both diagonal directions rather than horizontal and vertical directions alone. The authors compare the performance of this filter, using a 7 × 7 window for averaging operations, with the Lee, Lee-refined, Kuan, Frost, Mean, and Median filters. They claim that edges are not only preserved but that they are enhanced with their algorithm.

Wang *et al.* [20] used an adaptive, diffusion-based filter. To identify edges in noise in the US image, they used the difference of local averages rather than the gradient operator. They use eight directions for controlling the diffusion and use statistics to determine when to stop the process. The goal in this work is to preserve edges as part of their ongoing research into preprocessing fetal ultrasound data for improved visual assessment of fetal images.

Abd-Elmoniem *et al.* [21] proposed to suppress FDS based on an adapted, non-linear diffusion model. This group believes that some speckle patterns (texture) are not simply characteristic of the imaging system, and that deviations in speckle could be used to classify local regions. The classification of the local region determines the bandwidth of a smoothing filter so that FDS is replaced by a local mean and that regions that are dissimilar to FDS are not smoothed. The goal is to enhance coherence to overcome the ambiguity of using SNR alone for filter adaption and to enhance tissue texture and larger structure edges.

Generally, researchers that are concentrating on edge preservation within coherent imaging techniques tend not to over-filter texture, as this also tends to degrade edges. However, these three groups control the amount of speckle suppression based on the detection of edges, and thus it cannot be assumed that their methods do not remove relevant details in areas without detected edges. For this reason, methods based on adaptive filters controlled by edges were not pursued in this research.

2.7.5 Adaptive Filtering Based on Frequency Components of Noise

Uslu *et al.* [88] proposed algorithms for enhancing the detection of tumors that begins by split-spectrum processing (SSP) of the noise. The input image is bandpass-filtered using Gaussian windows and each band is scaled to unity. They believe that FDS is frequency dependent. SSP is implemented in both 1-D (filtering each A-scan line) and 2-D, and then the images are filtered to enhance the contrast of the tumors. This group found that SSP processing improves the supression of noise (speckle). Based on qualitative assessment of tumour contrast, they determine that implementing the SSP using 1-D filters out-performs the 2-D filter method.

2.8 Random Forests and Imaging

The Random Forest algorithm, in which ensembles of decision trees are created, was introduced in 2001 [89]. The Random Forest classifier (RFC) is non-parametric, handles high-dimensional data, and tends not to overfit. A complete introduction to its features and uses is discussed in the theory section. The advantage of RFC is that it provides an accurate error estimate without a separate test set, and all references to accuracy or error in this review are based on that error measure. Random Forests is an emerging technique for classification problems in computer vision, and thus a review of its computer-vision applications is short.

2.8.1 Land Cover

Random forests classifiers have been applied to the greatest extent in land-cover classification problems. Some cases apply the technique to one type of image while others use multisource data. In all cases, the problem is the classification of visually similar land cover into one of several classes.

Pal [90] uses landsat data to classify 7 types of crops. The experimental data, consisting of 4737 pixels, were selected from all seven classes with 2700 pixels used for training and 2037 pixels used for an additional test set. Like all land-cover problems presented here, the classification is performed on pixel intensities. The author found that RFC outperformed boosting and bagging, which are older, more established ensemble methods, even when 8% of the class labels for the training data were changed. Using the same data set, Pal [91] performed another study comparing the performance of RFC to support vector machines (SVMs). Pal found that the two classifiers have similar performance. However, the RFC has advantages over SVMs in that it requires little tuning by the designer, can handle unbalanced data, and can handle missing data values. Pal also demonstrates that the settings of the user-defined parameters in the RFC do not have a large effect on the final classifier performance.

Ham *et al.* [92] used two separate hyperspectral data sets to compare the performance of the RFC algorithm to a modified RFC algorithm (RF-BHC). Hyperspectral images use many narrow sections of the electromagnetic spectrum in remote sensing [93]. Hyperspectral data have the properties of being of high dimension (around 200), having a limited amount and quality of ground truth, and the quantity of training data is small. In RF-BHC, the classifier that forms the ensemble is a binary hierarchical classifier which decomposes the C-class problem, where $C > 2$, to a hierarchy of C-1 simpler 2 class problems. The RF-BHC error rates are somewhat lower than the RF algorithm for the data presented, but the difference in one experiment is 3% (94.9% vs. 92 %) while the other is 5% (75% vs. 70%).

Joelsson *et al.* [94] also compared the performance of RFC and RF-BHC against each other and against a Gaussian maximum likelihood classifier. The images were taken in an urban area in northern Italy with 9 classes represented. Both RFC techniques outperformed the traditional classifier. RF-BHC was reported to perform somewhat better, but the difference was in the range of 1%.

Gislson *et al.* [95] presented work with multisource data for identifying 10 classes of land cover, 9 of which are different forest types. They use 2019 samples where 1008 are used for training and 1011 are provided as an additional test set. The RFC algorithm was run with a variety of values for algorithm parameters and it was found that these changes had little effect on the final accuracy of the classifier. The highest accuracy obtained was 82.7%. This group found the error estimate by RFC to be pessimistic as compared to the test-set values. The RFC was compared to a number of other ensemble classifiers with improved or very similar results.

2.8.2 Image Classification

Frequently, multimedia retrieval systems will incorporate learning algorithms to improve the accuracy and speed of the images retrieved to the individual making the inquiry. Wu *et al.* [96] use an iterative algorithm incorporating the RFC and user feedback to determine the relevance of the returned images. The image features for classification include colour descriptors, texture features, edge information and shape descriptors. The performance of their algorithm is compared to retrieval systems incorporating SVM and another ensemble method, with an improvement of 24% using RFC. Later work [97] reports an improvement through a reduction in the number of iterations by using RF to identify non-relevant objects, rather than relevant ones in the initial searches.

Calleja *et al.* [98] compare the performance of machine learning techniques to the problem of galaxy classification. The techniques include a naive Bayes classifier, a decision tree, RFC, and another ensemble (bagging) method. The problem consists of image analysis, data compression, and machine learning. In the image analysis portion, a binary threshold is applied and the images are rotated, centered, and cropped to create images that are invariant to colour, position, orientation and size. The covariance matrix of the galaxy is obtained and from this, the principal components are identified and three are selected as descriptors. The data consists of 292 galaxy images and the classification problem is set up three times with 3, 5 and 7 classes. RF provided the highest accuracy of the four methods with values of 91.6%, 54.7% and 48.6%.

Luo *et al.* [99] identified plankton in images obtained with a shadow-image particle profiling evaluation recorder which continuously sample plankton and suspended particles in the ocean. The binary images in the group include a total of 7285, 1285 of which are used for training. The gold standard is manual classification. The processing of the images included noise suppression and the extraction of 29 features. The features include shape and size descriptors. Six classifiers were compared for performance using the entire feature set. In this case, they found that a SVM outperformed everything including RFC, although the difference in the error estimate was 1.8% and in the test set was 0.6%. Feature reduction was applied by successively reducing the dataset and preserving the best group at each step, resulting in 15 features selected. Only the SVM and the neural network were compared at this stage, with the SVM being proclaimed the best classifier.

Geng *et al.* [100] classify C. elegans (worms) based on 16 classes (1 wild, 15 mutants) which are observed with a high magnification system that tracks an individual's movements. The image data consists of consecutive stills that are obtained at 5 minute intervals, with 1596 video sequences analyzed and roughly equal numbers of clips per worm type. This paper presents new automatic methods for feature extraction and phenotype classification based on shape descriptors and tracking of head and tail movements. The 253 features are acquired from both binary and grey-scale images. Groups of measures were expressed as average, maximum and minimum values, and 10th and 90th percentile values of a population. The RFC algorithm classified the worms with a 91% success rate. The paper presents some discussion on the effect of RF algorithm parameters and finds the performance insensitive to their selection. Manual classification of the various types ranged from 50% to 90%. The RFC provides a more consistent classification method over manual classification.

While the use of RFC for image segmentation and classification is still limited, some trends can be observed. Random forests have yielded good results in imaging problems where the measured features have high dimensions. The work reported here indicates that the random forest algorithm is robust to parameter settings. Where random forests have been compared to other classifiers, based on the error estimates, the performance is better or comparable to other classifiers in image analysis problems. The error estimate of the RFC has been demonstrated to be credible if not a bit pessimistic.

Chapter 3

Theory

The design of this UTC system required addressing two main issues: extracting the relevant texture measures and designing and evaluating the classification system. The texture measures have been selected to exploit unique properties observed in past experiments but not utilized for an UTC system to detect WMD. Once the measures were obtained from the images, the classification system needed to be selected, designed, and tested, (Figure 3.1). The following section outlines the considerations for the UTC algorithm design for detecting WMD and the final system as designed.

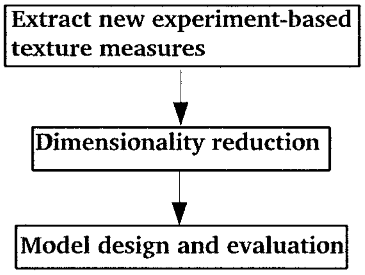
Extract new experiment-based texture measures
Dimensionality reduction
Model design and evaluation

Figure 3.1: Design of the model to test the hypothesis.

3.1 Preliminary Considerations

3.1.1 Experimental Limitations

When tasked with this problem, there were several limitations presented. The first is that the images would come from a normal clinical setting and it would be expected that the system would function with whatever images are clinically acceptable. This meant that there was to be no alteration of clinical practice. The images would be of neonates and taken within days of birth. However, the degree to which the infants are preterm would vary. The inclusion criteria are that the infants must be less than 31 weeks gestational age or have a mass less than 1500 g. Thus, the size of the cranial ultrasound images and the gestational age have significant variability.

The white matter in the periventricular region is the most susceptible to WMD. In the interest of managable data sets and computational time during design, one coronal US image of the periventricular white matter per patient was used as input to the new system. The image selected for each patient is of the same anatomical region, the region most likely to contain WMD.

The data sets used in this research consisted of EXP0, EXP1, and EXP2. Every image used in this research is film, scanned to digital with an Epson Perfection Photo 3200 DPI (hardware) scanner. EXP0 consists of 30 images that were used to assess texture as a feature on which to segment tissue types. EXP1 has 17 images, a subset of EXP0, where the clinical outcomes are known. These were used to identify texture measures as candidates for further evaluation. EXP2 consists of 69 images, with known clinical outcomes, that are used for the design and test of the model. The complete details of the EXP1 and EXP2 data sets are provided in Chapter 4.

3.1.2 Image Standardization

It has been established that operator-based machine settings alter the properties of B-mode images [101]. In UTC problems, researchers have traditionally addressed this problem by standardizing the images using one of two methods. Standardization can be performed by requiring that all images for a given experiment are obtained with the same machine with identical settings. Another method is to correct for the variations in the machine settings through the generation of a machine model and appropriate image processing methods to normalize the images [45].

In this work, the images are not standardized, but rather the patient's image is used as its own control. This is accomplished by comparing the white matter to a reference tissue, with the choroid plexus chosen as that reference. While this approach may not be appropriate for all UTC problems, in this case there are conditions that make this a reasonable choice.

1. During clinical evaluation, radiologists use the choroid plexus as the reference tissue when looking for increased echogenicity in the white matter [102].

2. All images are obtained through the anterior fontanel. The anterior fontanel is located at the midline, halfway between the forehead and the top of the head. The normal size of the anterior fontanel is $20mm \pm 10mm$ [103]. The relative locations of the choroid plexus and the white matter are always the same, with the white matter closer to the transducer face.

3. The choroid plexus is located close to the white matter, and both are located roughly at the center of the image at the focus of the acoustic energy. Thus, the variation of texture properties due to distance from the focal point should be minimal.

The location of the tissues within the focal zone is important because the observations by earlier researchers [36],[50], which are implemented in the algorithms, are based on the location of the tissue within the focal zone.

The distance from the transducer face to the tissue affects texture properties [36], [104]. A variation in the depth of the white matter from the transducer will affect its texture measurement. Because a reference tissue is used, the distance between the white matter and the choroid plexus will also have an effect. For these reasons, measurements of the distance between the transducer face and the ventricle and between the tip of the ventricle and the bottom of choroid plexus were obtained. The tip of the ventricle was chosen as it is the basis for selecting the white matter region to be investigated. The measurements were obtained at a standard radiology station with distance measurement capability. These measures were obtained from a number of patients, allowing the calculation of the mean and the standard deviation of these distances. The distance from the ventricle to the

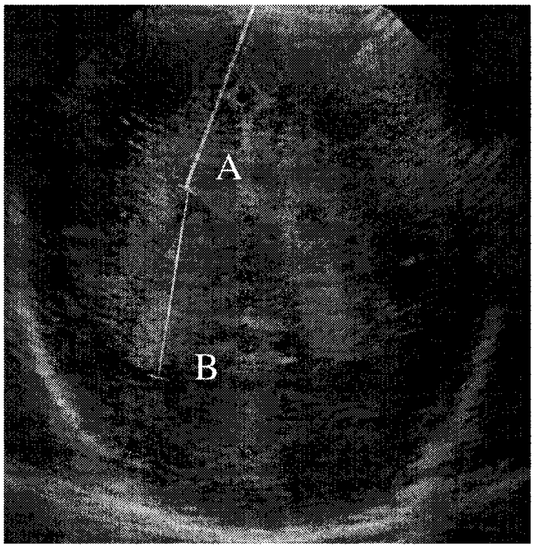

Figure 3.2: Measured distances. A: Distance from the transducer face to the tip of ventricle. B: Distance from the tip of the ventricle to the bottom of the choroid plexus.

transducer face was found to be 2.8 ± 0.2 cm. The distance between the tip of the ventricle and the bottom of the choroid plexus was found to be 2.7 ± 0.5. Figure 3.2 illustrates the distances measured. The variation in the distances was considered in the algorithm. Further details are provided in the discussion about the angle of insonation.

3.1.3 Resolution

The images are obtained at high spatial resolution. This was done to preserve as much detail as possible, including speckle. The justification for using high resolution is that the system is looking for diffuse tissue changes in patients without cystic echolucencies. At this early stage, the disease causes subtle changes in the white matter in the form of edema and an influx of immune-system components. These biological changes are expected to cause small-scale changes in the images. If the high resolution features had been found to be unnecessary, they could be removed later. However, the data could not be added if the data were not collected.

For every processing method, the size of the spatial support must be selected keeping in mind the size of the features of interest and the limitations of the imaging system.

Table 3.1: Average biparietal head measurements of fetuses.

Age (weeks)	Head size (cm)
24	6.0
28	7.3
31	8.1

The discussion first proceeds assuming that resolution of the images is the limiting factor. Based on standard growth charts, the biparietal diameter ranges from 6.0 cm to 8.1 cm for the ages 24 to 31 weeks, see Table 3.1. The average number of pixels comprising the biparietal measure for EXP2, is 5700 pixels. Choosing a mid-range value of 7.0 cm to represent the head size, this results in about 12 microns of tissue being represented by each pixel.

Ultrasound is not an ideal imaging system, however, and the frequency of the pulse has a significant effect on the resolution. The higher the carrier frequency of the pulse, the smaller the detail that can be resolved. Conventional wisdom is that detail down to half the wavelength can be resolved, although this is being challenged [10]. With a speed of sound in the human body assumed to be 1540 m/s [105], the wavelength for a 7.0MHz transducer is 220 microns. The frequencies range from 5 MHz to 8.5 MHz, and thus the wavelength ranged from 181 microns to 308 microns. A wavelength would then range in size from 15.1 to 25.7 pixels for a pixel size of 12 microns. All spatial supports and kernel sizes were chosen so that they are greater than 26 pixels at full image resolution. The regions are of a size containing both speckle and resolvable features. The spatial support of the processing techniques will be described in terms of this scale.

3.1.4 Image Analysis Platform

All algorithms for this thesis were written in C. They are written as commands within the image processing platform CVLab, written by Dr. P. Gregson. CVLab creates an interactive image-processing environment with some basic image manipulation routines. All user interface programs, image processing routines, and data extraction techniques were implemented in C in this environment. Algorithms created for this work are included

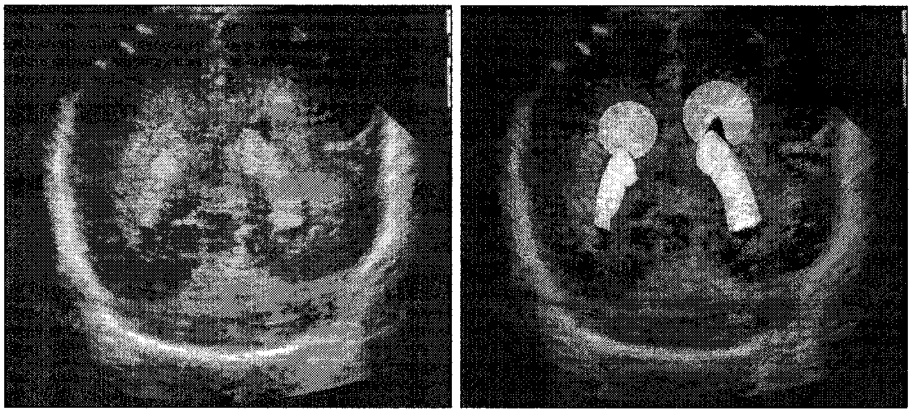

Figure 3.3: Left: A typical coronal image from our data set. Right: The periventricular white matter and choroid plexus ROIs chosen to be part of the image evaluation. The round upper areas are the white-matter ROIs, the longer shapes are the choroid plexus ROIs.

in Appendix A. All program descriptions are followed by the name given to the code to allow cross-referencing. Where the algorithm name is not provided, the necessary steps were performed by running script files containing the appropriate sequence of commands.

3.1.5 White Matter and Choroid Plexus Masks

Prior to any design, the tissue of interest had to identified. As the automatic segmentation of tissues from US images is a non-trivial task, a semi-automated method was chosen. The choroid plexi and ventricles in each patient image were manually extracted using a point and click tool (choroid.cmd). The white matter ROI was extracted through the manual identification of landmarks by the radiologist with the automated system calculating the remainder of the mask from these points. The complete details of the system are provided in Chapter 4. Figure 3.3 illustrates the tissues of interest. It should be noted that the white matter masks may contain germinal matrix as well as white matter within their boundaries. This could not be avoided as the two tissues cannot be segmented by radiologists using this imaging technique.

3.1.6 Noise Reduction

Prior to any image analysis, the level of noise in the images was reduced. The film images were scanned at 3200 DPI, resulting in images of about 50 MB ($\sim$ 7000 x 7000 pixels) after conversion to pgm format for use in CVLab. By digitally scanning the images with such high resolution, the speckle-texture features and the noise at the resolution of the film grain were preserved. Beyond the film grain noise, there exists unwanted signal associated with writing of the hard-copy films. The development process consisted of sending the signal from the ultrasound machine to a laser printer (Kodak 2180) and on to a developer (Kodak M6AN). The models of the machines did not change during the study period, but the laser printer was replaced in 1992. The pattern arising when writing the films differed slightly over the years but generally had a squared-cosine cross section. See figure 3.4 for a sample profile of the unwanted signal. The unwanted signal consists of components with spatial periodicities of both 18 pixels and 36 pixels for images from 1990-1991 and 10 pixels for the remainder. Median filtering the images with a window size of 37x37 pixels effectively removed the noise, including unwanted signal, in both groups with minimal signal degradation, as judged qualitatively. Figure 3.5 demonstrates the change in the images with noise reduction.

3.2 Extraction of New Texture Measures

B-mode texture properties have been observed to vary with acoustic scatterer density, size, periodicity, and scattering strength. The change in white matter due to the insult-causing-injury alters the scatterer density. The purpose of obtaining these texture measures is to capture the changes in image texture that are caused by the change in acoustic properties associated with the tissue changes. To increase the signal-to-noise ratio of those unique texture features, the images were first processed to enhance the speckle properties related to the changes and/or to remove those texture components suspected to be irrelevant. Following processing, the white matter and choroid plexus are compared, and the texture measures are extracted. The comparison results in an intermediate output image that is related to the texture properties and the shape of the ROIs. These intermediate images are referred to as "parametric maps". Figure 3.6 shows an overview of the steps in this

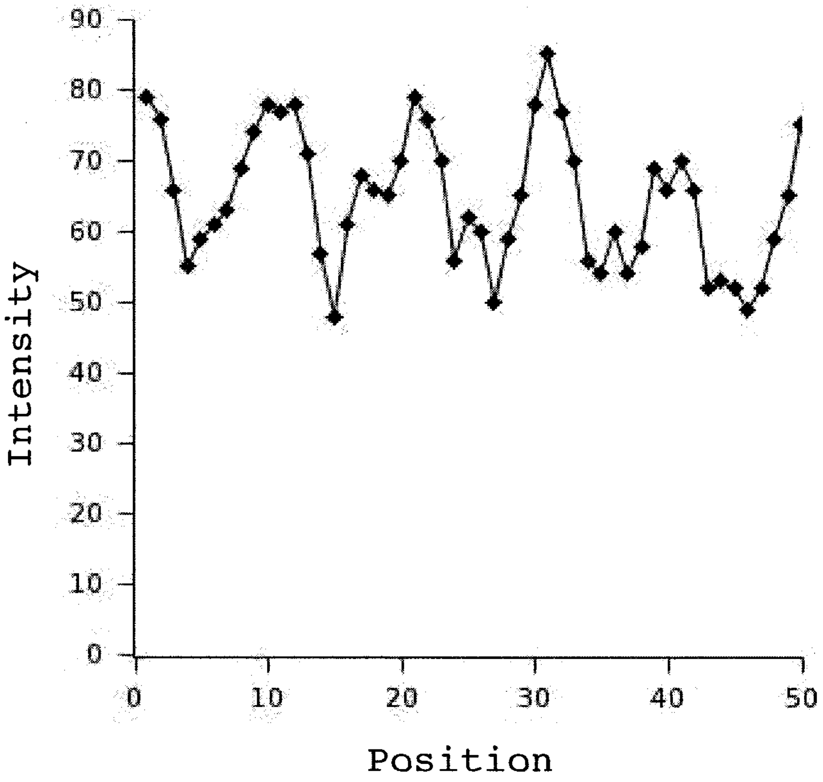

Figure 3.4: A sample noise profile obtained from the image background.

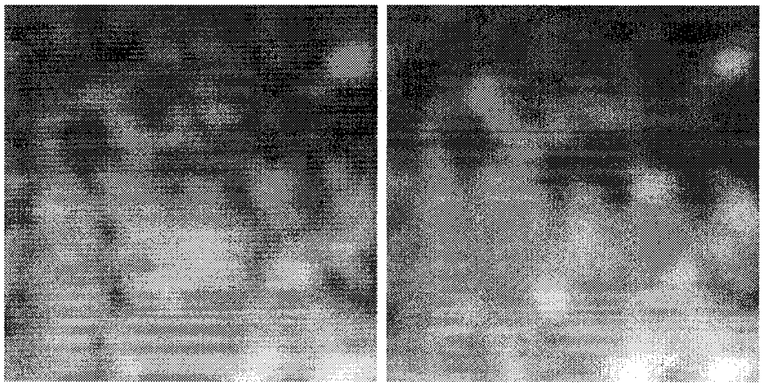

Figure 3.5: A small region of white matter. Left: original image. Right: median-filtered image.

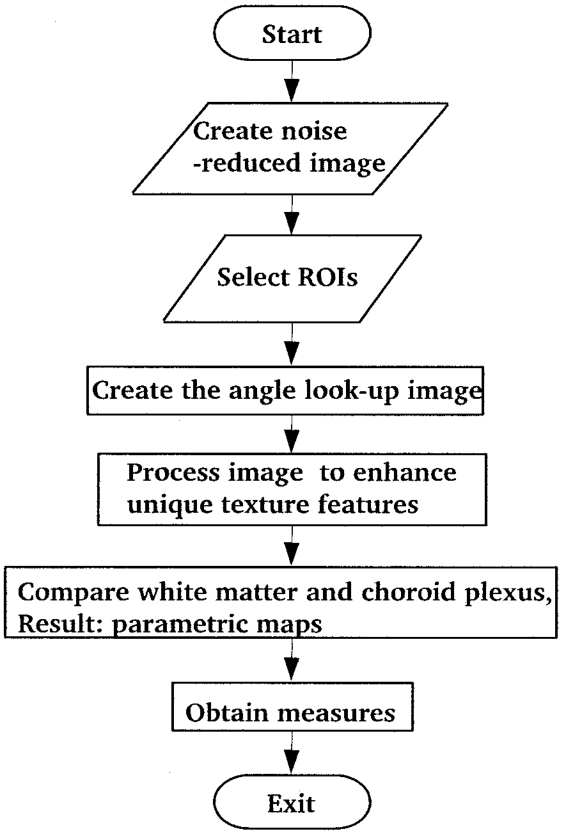
Start
Create noise -reduced image
Select ROIs
Create the angle look-up image
Process image to enhance unique texture features
Compare white matter and choroid plexus, Result: parametric maps
Obtain measures
Exit

Figure 3.6: Methodology of new, experiment-based texture measures.

section.

The processing that forms the method presented in this section, was developed through a series of experiments performed on the EXP0 and EXP1 data set. Initially, small samples of the choroid plexus and white matter were extracted by a radiologist. The images were processed and measures were extracted from these sample areas. The data were then assessed to determine if the measures separated the images into two groups, first based on tissue type, using EXP0, and then based on health outcome, using EXP1. The experiments assessing the measures' ability to distinguish between health outcomes are in print [106] [107] and are provided in Appendix B.

Once measures were found that separated the images by health outcome, a more-complete method to compare the two tissues was explored. The output of this stage

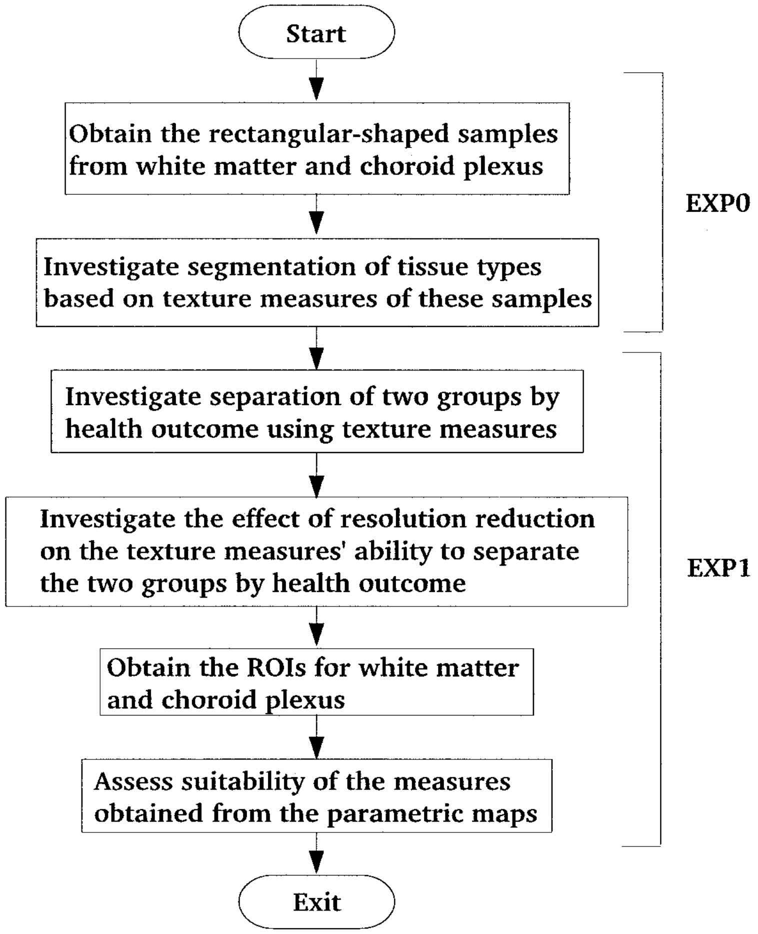
Start
Obtain the rectangular-shaped samples from white matter and choroid plexus
Investigate segmentation of tissue types based on texture measures of these samples
EXP0
Investigate separation of two groups by health outcome using texture measures
Investigate the effect of resolution reduction on the texture measures' ability to separate the two groups by health outcome
Obtain the ROIs for white matter and choroid plexus
Assess suitability of the measures obtained from the parametric maps
EXP1
Exit

Figure 3.7: Initial assessment of suitability of texture measures for diagnosis based on the EXP0 and EXP1 data sets.

is a group of parametric maps that contain many comparisons between the two ROIs. Descriptors of these maps were then assessed for their suitability to separate EXP1 data into two groups based on health outcome. Figure 3.7 illustrates the steps in the design of the texture measures.

3.2.1 Pre-processing

Angle Look-up Image

All coronal cranial US images in these experiments were obtained using a curved-array transducer. The center frequency ranged from 7.0 MHz to 8.5 MHz. This transducer shape results in the tissue being insonated in a fan pattern. As a result, the US machine collects data in a polar coordinate system and later converts them to a rectangular coordinate system to form the images. However, the texture information in the axial direction is less dependent on distance to the transducer and on the transducer properties than the lateral direction. Refer to Figure 2.3 in Chapter 2 for an illustration of these directions. It has been suggested that texture information in the axial direction has the most value for UTC [6].

The image processing and comparison methods contain either kernels or windows. Many of these are rectangular in shape and oriented with their long axis matching the axial direction at each location within the image. To accommodate all processing techniques that require the axial-direction information, the angle of insonation is calculated at each image location and saved as a look-up image. Figure 3.8 contains an example angle look-up image. The creation of this algorithm (ultgrid.cmd) was performed in collaboration with Ms. Zhengyan Sun.

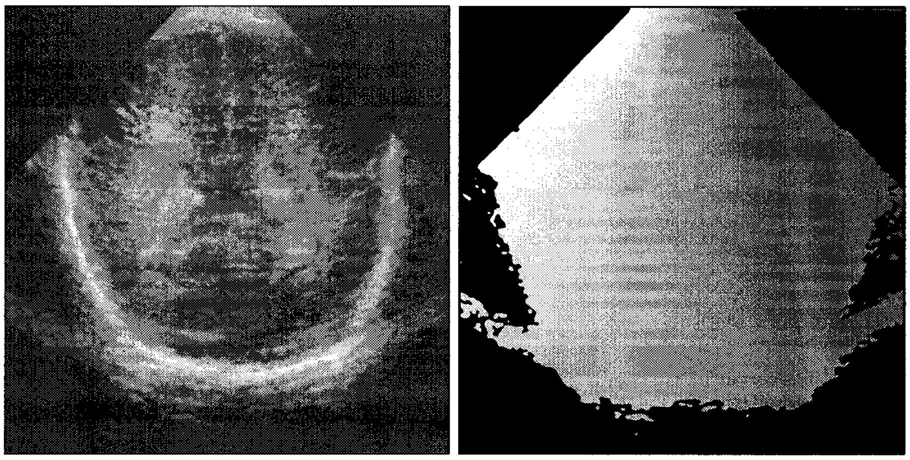

Figure 3.8: Left: Sample coronal image. Right: Angle look-up image created for the image. The intensity variations have been exaggerated for illustrative purposes.

Enhancement

First- and second-order statistics of B-mode images are a function of the density of acoustic scatterers [36], [50]. Speckle can be regarded as the texture primitive of US, with a size and orientation that varies throughout the image. The changes in US image texture may be caused by a number of changes in speckle properties such as an increased density of speckles or an increase in the individual intensity of each speckle. The processing techniques are an attempt to capture the changes in speckle properties that may provide information about tissue changes.

Twelve processing techniques were investigated, each with a variety of window sizes or kernels types as appropriate. Of those investigated, four filtering methods were selected based on simple separation of the two groups of EXP1 according to outcome. The four filters are: morphology-based filtering (IDD30), edge enhancement (DM), orientation specific band-pass filtering (GABT19), and only noise-reduced filtering. These techniques are explained in detail below. In this discussion, the terms enhancement and processing refer to the four filtering methods.

Resolution Reduction To reduce the computation time in the successive steps of the algorithm, the images are reduced twice using the Burt-Adelson pyramid method [42]. For the GAB19 and DM methods, this was done before processing. Although these methods were selected based on full-sized images, it was confirmed that the parameters for these techniques still allowed for separation according to outcome using reduced images. For the IDD30 technique, the images are reduced after processing.

The reduction was limited to twice, as there was evidence in early experiments (Figure 3.7) that a reduction beyond this level reduces information content. See Figure 3.9 for a graph illustrating the change in ability to discriminate with resolution reduction. The texture measure in this graph was obtained after processing with IDD30. The mean ±(2× standard error) for both the Control and CP outcome groups are plotted against each other for one, two and three resolution-reduction steps. At three times resolution, the measures overlap. Appendix C contains the data gathered at each level of resolution and the calculations that lead to this graph.

While all images were reduced twice, this does not guarantee that the images are

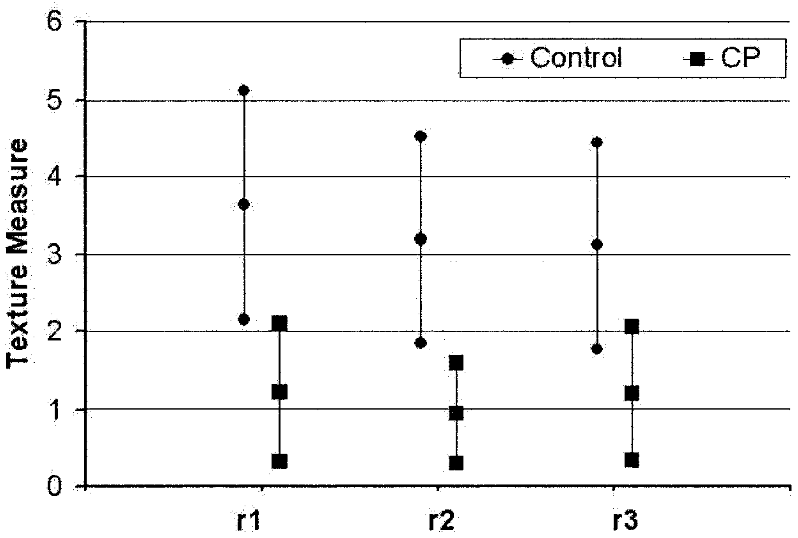

Figure 3.9: Mean $\pm(2 \times standard\,error)$ for the control and CP groups. From left to right: One, two and three times resolution reduction.

resticted to the same resolution level. This is only true if the frequency content of the original images is such that all have a bandwidth greater than the cumulative cut-off frequency from low-pass filtering prior to sub-sampling. To confirm that this is the case, samples were obtained from the white matter of all EXP1 images. The samples are 256×8 pixels and oriented along the axial direction. The frequency spectra were obtained for each sample and the area under the spectrum curve up to the cut-off frequency was calculated. The positive portions of the spectra are 128 points long, so the area under the curve up to point 32 is of interest. This region is then compared to the entire area under the positive portion of the curve. Figure 3.10 illustrates the positive spectrum from one image sample, and the cut-off. The ratios of these two areas for all images ranged from 0.97 to 0.99, suggesting that the amount of signal below the cut-off frequency is nearly equal for all. The DC magnitude was not included in this calculation because it is typically significantly larger than the magnitudes at the other frequencies and thus would reduce the ability of this test to identify differences between the spectra. The frequency spectra for all samples are provided in Appendix D.

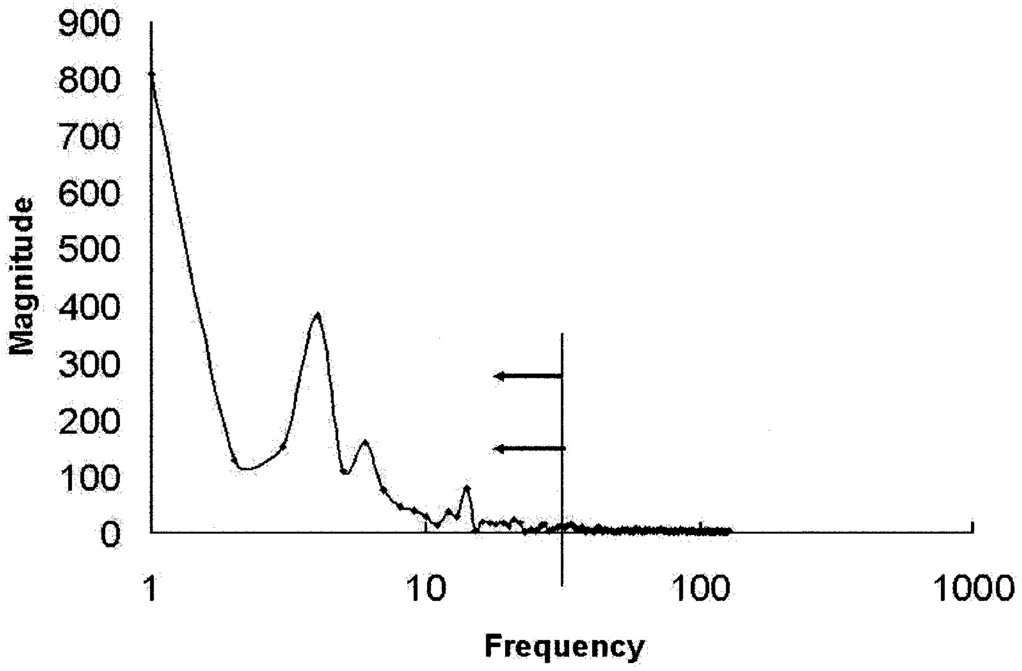

Figure 3.10: An example of a positive spectrum. The line and arrows indicate the cut-off frequency and the area calculated.

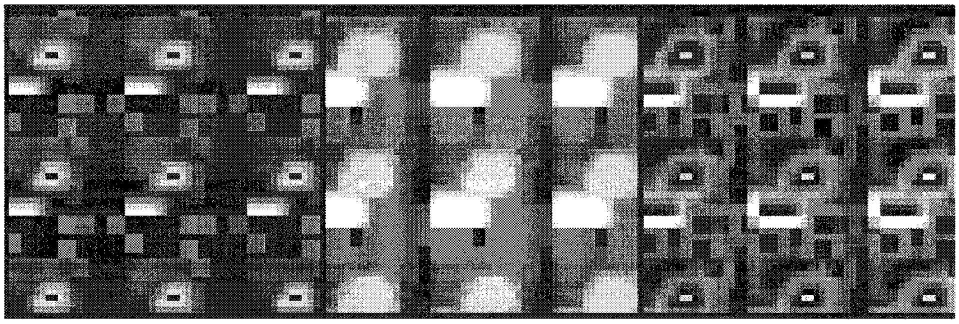

Figure 3.11: Effect of grey-level dilation and subsequent subtraction on a deterministic pattern. Left to right: Pattern, pattern grey-level dilated with a disc having radius = 3 pixels dilated, final processed image.

Morphology-Based Filtering The disease process of WMD may decrease the density of scatterers due to edema. Experiments indicate that the number of speckles varies with the density of scatterers [36]. If the density of speckles change, then the spaces between speckle may also vary. Enhancing the small spaces between the speckle "lumps" may reveal information about tissue properties. Grey-level dilation is a morphological operation that, among other things, fills low-intensity areas that are smaller in size than the structuring element used for the dilation. The structuring element may be flat or it may have an intensity variation. Dilating an image with a structuring element having a flat or binary intensity profile, and then subtracting the original from the dilated image causes small intensity holes to be filled and emphasized. Figure 3.11 illustrates this effect.

Dilation and the subsequent subtraction of the original image was chosen as a means of enhancing the spacing between the speckle primitives. It was chosen because it is simple to implement and separated patient images by health outcome in earlier experiments [106]. The kernel selected was a flat disc having a radius of 30 pixels. A disc, rather than a square or rectangular kernel, was chosen as it is orientation independent. Thus, it should work equally well for the various insonation angles throughout the image. Other kernel shapes and intensity profiles were investigated without improvement in results. The size of the disc was selected empirically and the same disc was used for all images. Continuing with the assumption of 12 microns per pixel, a disc sized 60 pixels in diameter is 720 microns which is 3.3 wavelengths for a 7.0MHz transducer. Figure 3.12 shows an image containing

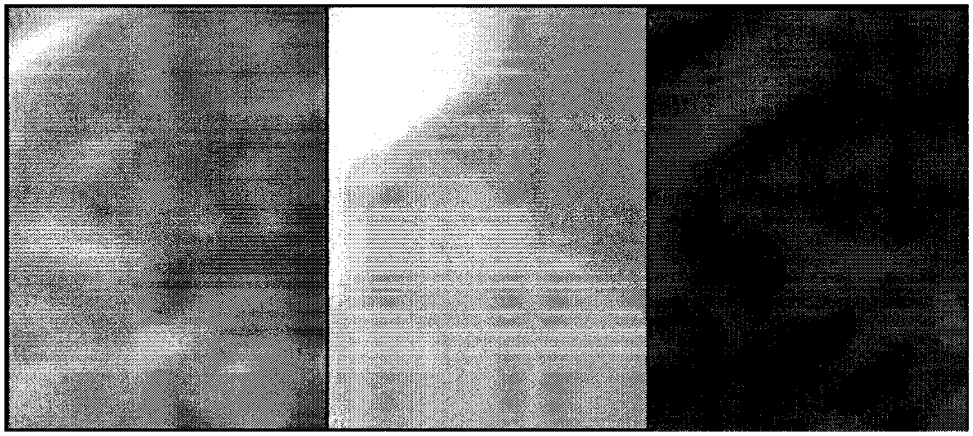

Figure 3.12: White matter sample. Left to right: Original, grey-level dilated with a disc radius = 30 pixels, resultant image created by subtraction of the previous two images.

only white matter processed by this technique (IDD30).

Edge Enhancement Speckle can be regarded as the texture primitive of US. Speckles are high-intensity patches that are roughly oblong in shape and oriented with their long axis perpendicular to the path of insonation. The size and orientation of speckle varies throughout the image (Figure 3.13). The intensity of the US image changes with the density of acoustic scatterers [36]. Intensity is also affected by structure of the acoustic media [54]. The disease process of WMD causes a change from order to disorder of the structure of white matter. Intensity changes in US images may be caused by increased density of speckles or an increase in the individual intensity of each speckle.

Rather than detect the speckle primitive and look for changes in its properties, detection of edges was chosen as a substitute. Because speckles are high-intensity areas, the detection of edges would include the boundaries of speckles. Once the locations of the speckle boundaries are known, their location relative to each other can be determined. This is relevant because in regions where there are more speckles it is expected that the edges of the speckles will be closer together. In addition, in the areas where speckles have an increased intensity profile, it follows that the gradient magnitude of those edges would be greater. Therefore, to obtain the maximum information about speckle using an edge-enhancing technique, it is important that an edge detection scheme includes information

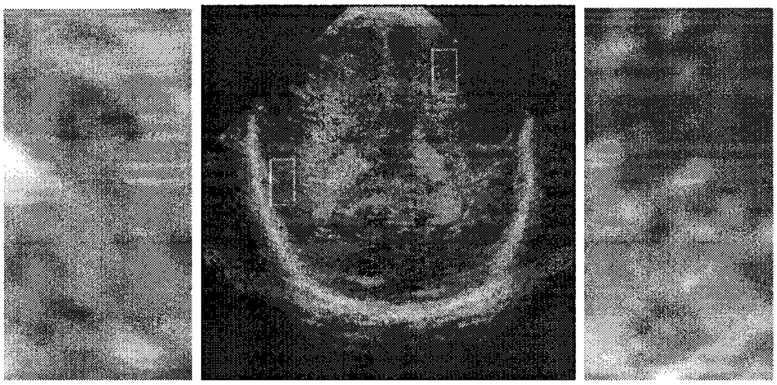

Figure 3.13: The variation of speckle primitives within a coronal US image.

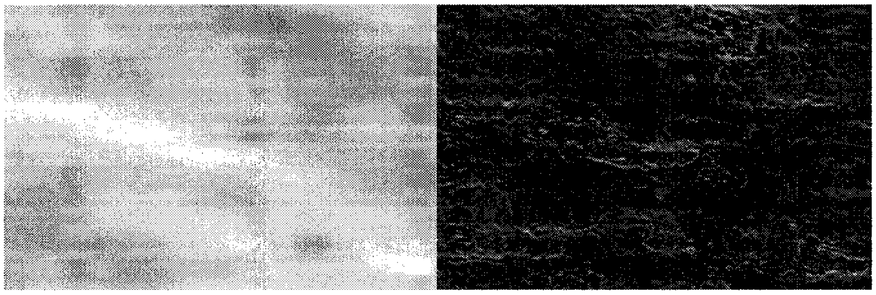

Figure 3.14: Left to right: White matter sample, Sobel magnitude.

about the intensity of the edges.

The problem with using a gradient-based edge detector is that speckle edges are very diffuse and traditional gradient-based edge detectors do not perform well. Traditional gradient-based detectors are implemented with a 3 × 3 kernel which is a region of support too small to include the spatial support of the speckle edges (Figure 3.14). The size of the kernel can be increased, but as with all linear filters, increasing the support of the Sobel kernel causes a decrease in localization of the edge. Figure 3.15 illustrates the effect. This figure shows a diffuse edge within Gaussian noise. A Gaussian approximation to film grain noise was chosen. This approximation was created from the histograms of samples of cerebrospinal fluid, which has very few scatterers, and so film grain noise predominates.

The "Difference of image and mode" filter (DM) is used to enhance the speckle edges. First, the edge is converted to a step edge using a local-mode filter (mode5.cmd). This image is then subtracted from the original image. The resulting image has zero crossings

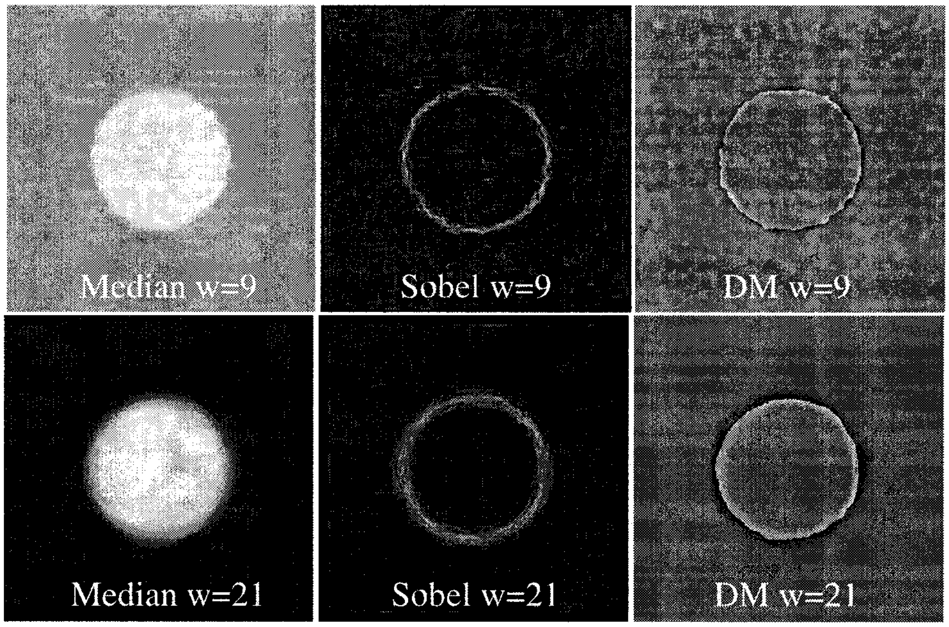

Figure 3.15: As the size of the Sobel kernel increases from 9 × 9 to 21 × 21, its localization decreases. The same effect is not observed in DM. Median filtering is performed for noise reduction for each technique.

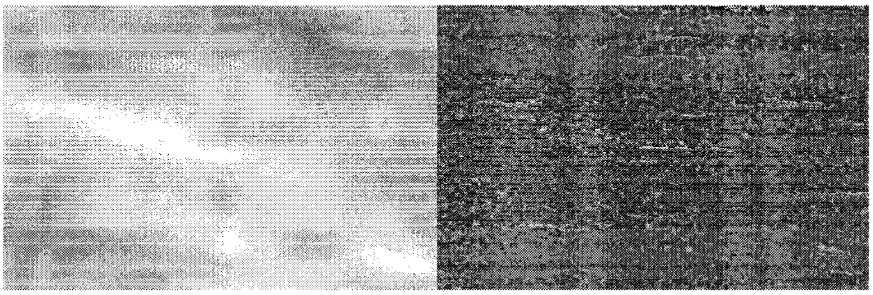

Figure 3.16: Left to right: White matter sample, DM processed.

at the center of edges with positive and negative peaks on either side of the edge centers. The size of the peaks depends on the gradient of the original edge and the distance to the neighboring edge. For this work, a 9×9 window was selected based on the width of the edge transition for speckle (Figure 3.16). The images are twice reduced prior to processing with DM. This leads to a pixel representing 48 microns, rather than 12 as stated earlier. Thus, the 9 pixel dimension of the DM support represents a 432 micron region or approximately two wavelengths for a 7.0MHz transducer. Appendix E contains the paper [108] describing the behavior of this technique.

DM and IDD30 are based on differences. Both methods remove the DC component, and thus additive changes in the image intensity have no effect on the post-processed images. The gains of US imagers are non-linear, with an additive component. These processing techniques serve to remove the additive effect of gain variations. Both techniques were evaluated for their ability to separate EXP1 into two populations. The analyses consisted of t-tests [106]. Later these measures were tested using Mann-Whitney U tests, once they were incorporated into the full algorithm.

The Mann-Whitney test is a non-parametric test. It is most appropriate for data that is non-Gaussian and that is of a small size so the central limit theorem does not apply. The data obtained during the design is of a small-sized data set and cannot be assumed to have a Gaussian distribution, therefore a non-parametric tests is more appropriate in this case.

Band-Pass Filtering Work by others in split-spectrum processing of US images suggests that speckle features are frequency dependent. Split-spectrum techniques have been

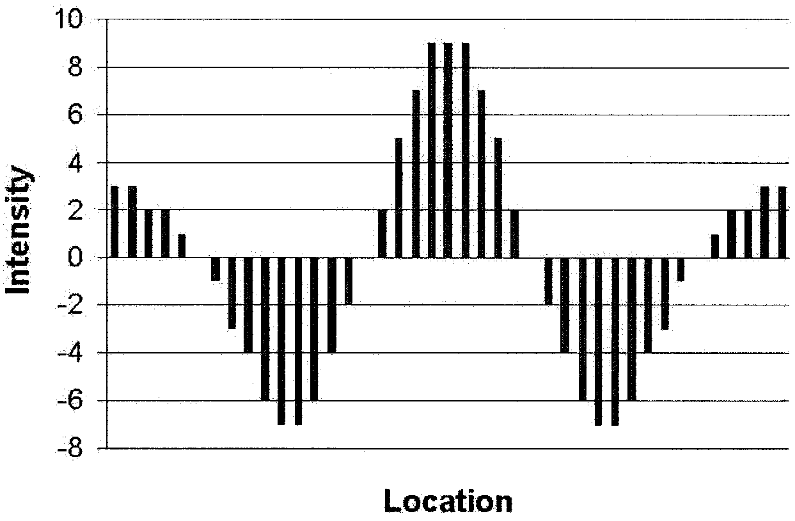

Figure 3.17: Gabor kernel used for variably-oriented band-pass filtering.

applied to A-scan ultrasound data for image enhancement through the use of 1-D Gaussian filters [109], [88]. This method ensures that the filter is always aligned in the direction of the acoustic pulses. In this research, a variable-orientation Gabor filter (GABT19) was applied to select specific frequency components along the axial direction. The Gabor filter was chosen because it is compact in frequency and space, and allows tuning to a mid-band frequency with a single convolution operation. The GABT19 filter is designed to have a cosine period of 20 pixels, a kernel size of 41 x 1 pixels, and zero DC component. The Gaussian and the cosine functions used each had maximum amplitudes of 10, however the kernel was scaled down by 0.1 before convolution to remove the problem of numerical overflow. Figure 3.17 illustrates the kernel intensities. The center frequency of the Gabor filter was selected, after examination of EXP1 spectrum samples, to be the frequency at which the populations separated according to outcome [107]. In the convolution process, the kernel is rotated so that the long axis is aligned with the axis of insonation based on the angle look-up image at the coordinates of the center of the kernel. The algorithm (vconv2.cmd) is used to implement the convolution, with rotateker4.cmd used to rotate the GABT19 kernel.

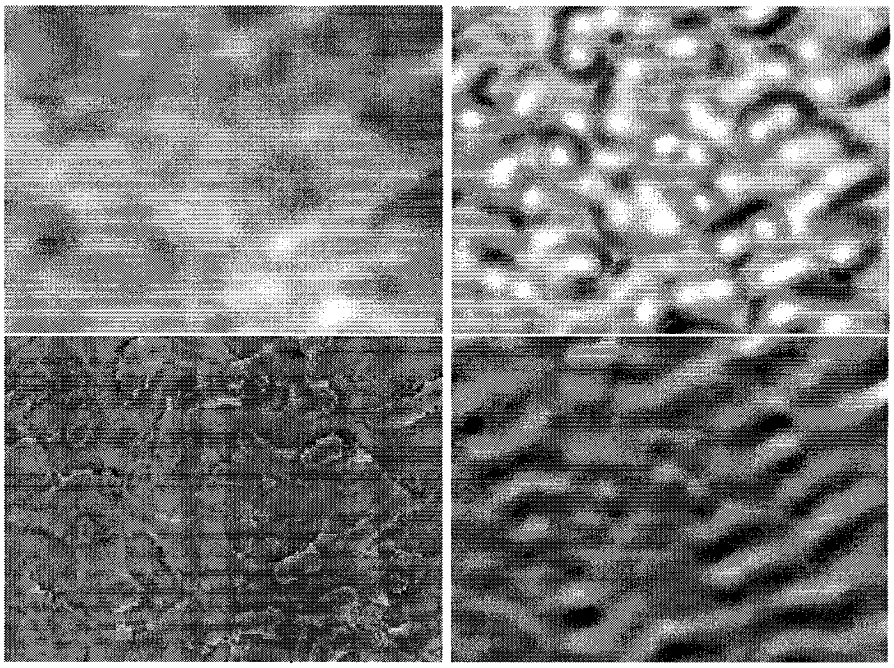

Figure 3.18: Top left: unprocessed except for noise reduction and reduced 2×. Top right: IDD30 and then 2× reduction. Bottom left: 2× reduction then DM processed. Bottom right: 2× reduction followed by filtering with GABT19.

Noise-reduced The final "technique" (ORIG), consists of no enhancement beyond the noise removal and 2× resolution reduction. The unprocessed images were included as it is possible that the most significant disease markers are removed in the other image-processing methods. The four types of processing are illustrated in Figure 3.18.

3.2.2 Comparison of Two Tissues

Because the images are not standardized, comparison must be made between the white matter and the reference tissue, the choroid plexus. The comparison is made after the images are processed. Even though processing is expected to enhance disease markers, it is expected that the disease signatures will be outliers within the population of the white matter ROIs. This is expected as the images are being analyzed during the injury period or before injury, in which case the susceptibility is being observed. Therefore, the biological changes will be slight.

One can expect the markers of any disease to be short conspicuous signal segments [44] which may not be observed within a very large image area. Therefore, comparison of the entire white matter ROI to the entire choroid plexus ROI would likely not capture the changes associated with the early disease process. Figure 3.3 illustrates some sample ROIs To capture the small-scale variations in the US image, the white matter was examined over small subregions and compared to the entire choroid plexus. The processes of comparing these two tissues consisted of identifying the appropriate size of sample, and determining what features to compare and how to compare them.

The earliest experiments within this research form the basis for the selection of the size of the white matter samples used in this work (EXP0, Figure 3.7). The first experiments concentrated on texture measures for segmenting the choroid plexus and white matter. In these experiments, four rectangular strips consisting of white matter and choroid plexus on each side of the patient's image were extracted from the full-size images. The strips were oriented along the axial direction and sized to provide a good sample in the opinion of the radiologist (Dr. Schmidt). Figure 3.19 shows an image and its tissue samples.

Using these samples as masks, statistics were obtained from these regions after enhancement. There was some suggestion that the tissues would segment but more importantly, the tissue measures varied with patient outcome. It was observed that the measures for white matter alone did not separate into two populations based on outcome, but the comparison of white matter to choroid plexus did. The graph in Figure 3.20 illustrates the differences in the texture measures from EXP0, suggesting segmentation may be possible. The graph in Figure 3.21 illustrates the change in the tissues measures based on outcome for EXP1. Both graphs use STDV as the measure after processing with DM, the tissue samples are as illustrated in Figure 3.19. These observations demonstrated the usefulness of a reference tissue. It also formed the basis for determining the appropriate resolution for the search for disease markers.

Determination of the Size of the Examined Tissues

As the size of the images and the orientation of the coronal slice both varied, the sizes of the rectangular tissue samples also varied. This led to the questions: "What is the minimum size of sample required to identify the texture features unique to WMD?" and "Does the

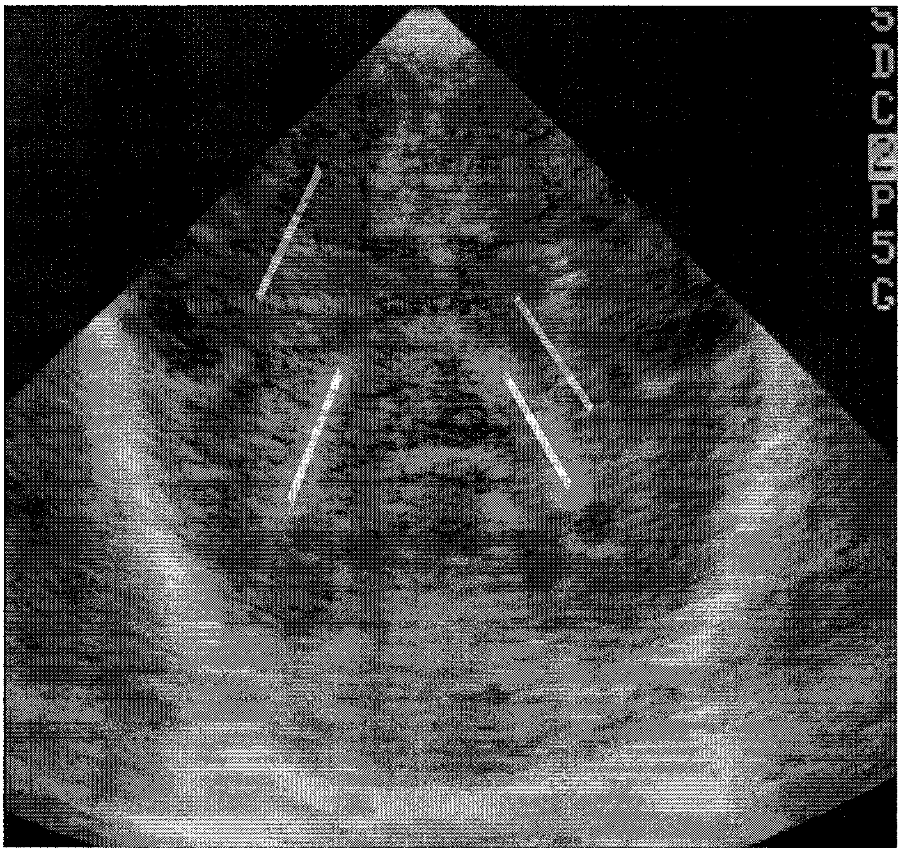

Figure 3.19: A coronal image indicating the tissue samples used in early experiments.

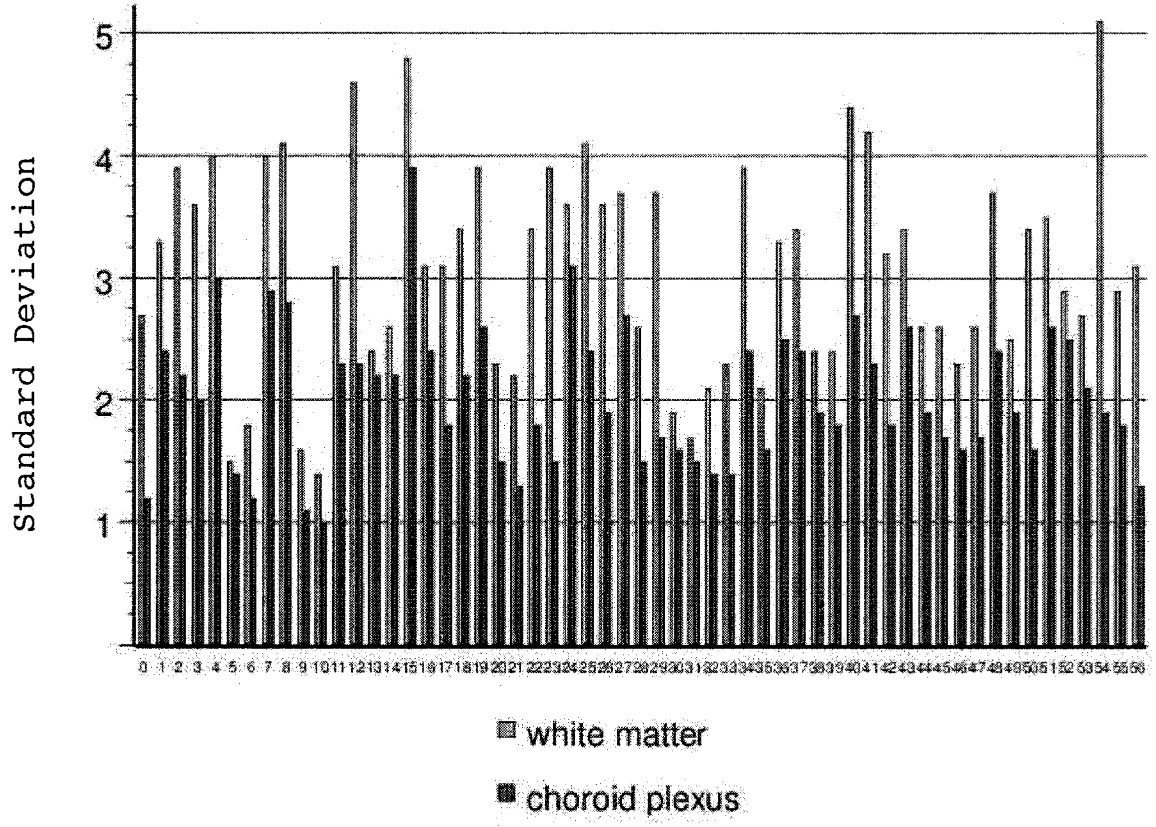

Figure 3.20: Segmentation of tissue type based on texture. The values for the white matter are greater than the choroid plexus for 55 of 57 sides. The number on the x-axis indicates a pair of choroid plexus and white matter measures from one side of a patient image.

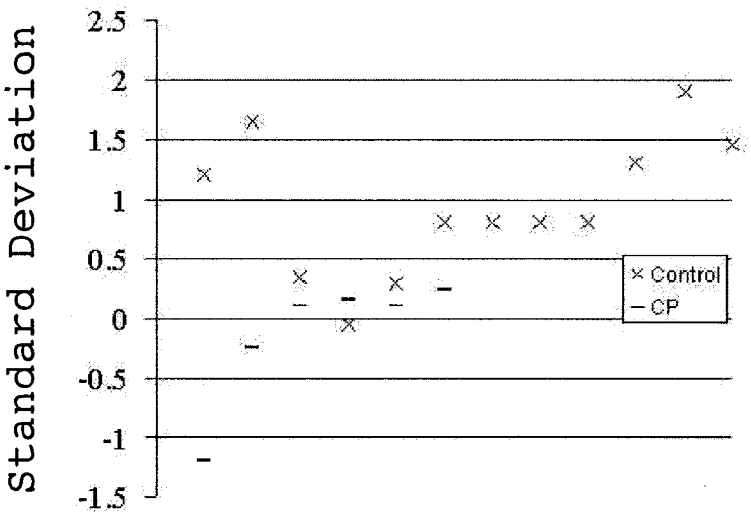

Figure 3.21: Separation of texture measures according to health outcome.

shape of the sample really matter?" To answer these, an analysis was performed in which the size of the rectangular white matter samples was reduced in three different ways. The samples were reduced so as to keep the ratio of length to width constant down to 1/16 their original size, a 1/4 reduction in each dimension. The widths were also successively reduced by 10 pixels at a time, to a maximum reduction of 30 pixels while keeping the original length constant. Keeping the original width constant, the length was reduced by 100 pixels successively, up to a reduction of 800 pixels where the original size permitted this much reduction. The standard deviation of all the white matter samples and the choroid plexus samples were compared after processing with DM. Both the white matter and the choroid plexus samples varied in size. The size of the white matter samples do not appear to influence the measure's ability to separate the patients by health outcome at an $\alpha = 0.05$ (probability of mis-classification) using the Mann-Whitney U test until the length is reduced by 400 pixels or more (Appendix F). This observation is based on all However, at this size the samples are still rectangular but the length-to-width ratio is smaller. For this reason, square samples were excluded as an option in the comparison. As well, work by others indicates the information is in the axial direction [104].

Based on these observations, rectangular tissue samples are preserved in the comparison of the two tissue types. Texture measures are dependent upon resolution but this analysis did not identify the limits of the required sample size.

To reduce computation time, the white matter sample was chosen to be small compared to the range of sample sizes in the above analyses. A reasonable value of 80x20 pixels was selected in the full-size images, which results in 20 × 5 pixels in twice-reduced images. With a 7.0MHz transducer, this is a support of 1040 microns × 260 microns or approximately 4.7 × 1.2 wavelengths. Regions of this size could contain both speckle and resolvable features.

Comparison Leading to Parametric Maps

The white matter is systematically examined within 20 × 5 pixel regions (80×20 reduced 2×) oriented along the axial direction within the white matter ROI (Figure 3.22). All of the white matter is relevant so the entire white matter ROI is examined in small windows by moving the rectangular region throughout it. The center of the examined region is located at every point within the ROI providing the entire rectangular region fits within the ROI. A comparison between the white matter and the choroid plexus is made at every location and the result placed in an output image at the coordinates of the center of the rectangular sample being examined. The values obtained from the choroid plexus and the white matter are compared by taking the difference or the ratio of the two numbers. The result is hundreds of comparisons per patient per white matter region. These output images are referred to as parametric maps. Figure 3.23 illustrates some sample parametric maps.

Texture Measures on the Enhanced Images

In order to compare the two tissues, a texture measure must be obtained for each of the regions. There are many ways that the tissue texture properties could be measured. There are several criteria that guided the choice. With four processing techniques (IDD30, DM, GABT19, ORIG) and two comparison methods (ratios, differences), there are eight maps for every texture measure used in the comparison. Each additional texture measure added, increases the number of maps by eight. It was desired to keep the number parametric maps created as small as possible to reduce computation time and to keep the size of the data

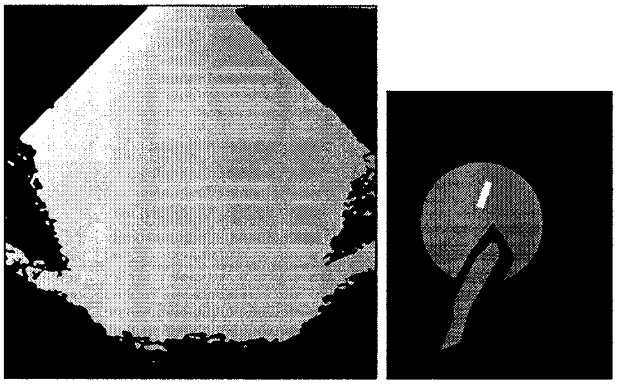

Figure 3.22: Left image: An example of an angle look-up image. Right image: White matter and choroid plexus ROIs from the patient's right side with a rectangular sample illustrated within the white matter ROI. The rectangular sample is moved throughout the ROI and a comparison is made between the rectangular sample and the choroid plexus ROI at every location. The rectangular region is oriented according to the insonation angle at the region's center pixel.

set per patient from being excessive. The measures had to be selected while keeping in mind the output of each type of enhancement performed. Assuming that the processing removed much of the irrelevant data, simple measures were expected to provide sufficient information. It was also desired to keep the texture measures simple in concept to help reduce processing time. Image enhancement and comparison of image tissue samples were expected to take a significant amount of processing time, in the order of 8-12 hours per output image. Because of the constant variation of the orientation of the small white matter samples, it was necessary for the measures to be rotationally invariant. It would be highly undesirable to build a detector biased to the location of the tissue rather than its health.

The combination of these criteria made first-order measures a good choice. The pdf parameters used to model acoustic media, such as Rayleigh or Nakagami, were not explored because most enhancement would alter the image intensity properties. As well, these models would not apply to the ORIG group as it was expected that the simple underlying assumptions on which the models are based would not capture the subtle changes early in the disease process.

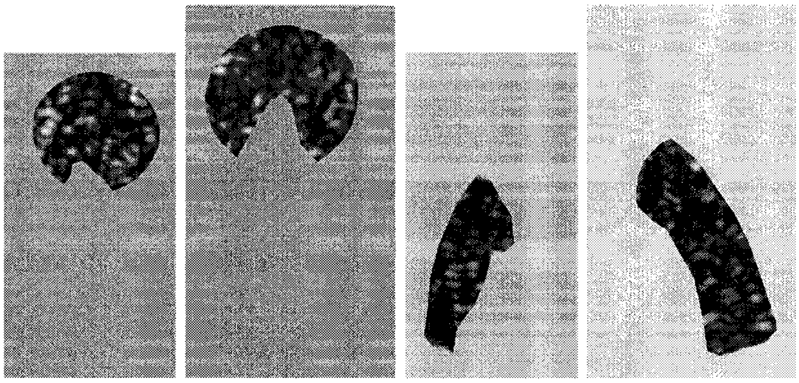

Figure 3.23: For each texture measure, there are four maps created for each patient for each processing technique. The two on the left are white matter parametric maps. The two on the right are choroid plexus maps. In this case the processing method is ORIG and the texture measure is STDV.

The Selected Texture Measures The texture measures selected were the standard deviation (STDV) and the normalized surface area (NSA). STDV was chosen as it is fast and easy to compute and the experiments investigating texture for tissue segmentation showed its value (EXP0 experiments). NSA was chosen to complement STDV. If one were to think of the intensities of an image as hills and valleys, the NSA is the surface area of the "terrain" divided by its projection on the x,y plane. As an illustration, two sinewave gratings may have the same amplitude but different frequencies so their STDVs would be equal but their NSAs differ. Additional measures were not chosen as it was desired to keep the number of measures to a minimum.

Parametric maps were created with small regions of white matter compared to the entire choroid plexus region on the same side of the patient's head. In those early experiments [106], it was observed that some discriminating features came from changes in the choroid plexus rather than from the white matter. In order to capture this information, texture maps were also created in which small regions of choroid plexus are compared to the entire white matter region. Sample parametric maps for one patient, using STDV as the measure and using ORIG as the processing technique, are shown in figure 3.23.

To summarize the variety of maps obtained, Table 3.2 lists the 16 maps that are created for each processing technique. With four processing techniques, there are thus

Table 3.2: The parametric maps created for each patient for one processing technique.

Examined tissue (samples)	Reference Tissue (entire region)	Measure	Comparison	Side
white matter	choroid plexus	STDV	ratio	right
white matter	choroid plexus	STDV	ratio	left
white matter	choroid plexus	STDV	difference	right
white matter	choroid plexus	STDV	difference	left
white matter	choroid plexus	NSA	ratio	right
white matter	choroid plexus	NSA	ratio	left
white matter	choroid plexus	NSA	difference	right
white matter	choroid plexus	NSA	difference	left
choroid plexus	white matter	STDV	ratio	right
choroid plexus	white matter	STDV	ratio	left
choroid plexus	white matter	STDV	difference	right
choroid plexus	white matter	STDV	difference	left
choroid plexus	white matter	NSA	ratio	right
choroid plexus	white matter	NSA	ratio	left
choroid plexus	white matter	NSA	difference	right
choroid plexus	white matter	NSA	difference	left

64 maps generated per patient. The parametric maps were obtained with the algorithms texmapsd2.cmd and texmapnsa2.cmd, with the algorithm rotateker4.cmd used to rotate the rectangular sample.

3.2.3 Measures Derived from Parametric Maps

These parametric maps are the outputs of a group of comparisons made at a relatively fine scale. Based on the body's inflammatory response, it can be expected that in the early stage of WMD, the damage has a significant diffuse component. For this reason and in the interest of reducing the data per patient to a reasonable level, the parametric maps are summarized. The maps are described through the two texture measures used earlier, STDV and NSA, as well as the first-order statistics of median, mean, maximum, minimum, skewness, and kurtosis. The normalized surface area is the same as described above, but normalized to the map size rather than the ROIs. Once these measures were obtained, the left and right versions were averaged per patient. For example, the mean of the NSA ratio map on the left was averaged with the mean of the NSA ratio map on the right, both for white matter. Therefore, the parametric maps result in 256 measures (64 maps x 8 measures /2 halves = 256) for each patient.

To complete the design stage of the parametric map-based measures, the individual measures obtained from the parametric maps (created from the EXP1 data set) were analyzed for their ability to separate the patients according to outcome. Mann-Whitney tests were performed on every measure. Spreadsheets showing a sample of the Mann-Whitney tests performed are included in Appendix G. However, this method assesses only the individual measure and not its interaction with other measures. Therefore, this assessment was not used for feature reduction but rather to confirm that there existed some measures within the group that likely contained diagnostic information.

3.3 Design and Evaluation of a System Model

Now that the texture measures are gathered, the hypothesis that there exists diagnostic information within them must be tested. To do this, a model was created and tested to evaluate its ability to identify the classes defined by the patients' health outcomes of CP or no CP. In general, the design of a model system consists of the following:

1. Define the properties of the data set and the classification problem under investigation.
2. Understand the different types of models and the problems that they are suited for.
3. Find the best fit between 1 and 2.

3.3.1 The Data Set

Before selecting a method for building a model, there needs to be an understanding of the data set.

The texture measures were extracted on the basis of prior knowledge of the physics of the imaging system, an understanding of the disease process, and previous experimental results (Figure 3.7). While this knowledge assists in the choice of texture measures, there is still a strong degree of uncertainty about the diagnostic information in the measures. It is expected that each measure will have a small amount of information. The data set consists of a large number of measures which are highly correlated. These factors contribute to a data set with low SNR. This low SNR is in addition to the imaging limitations due to the subtle nature of the changes early in the disease process.

The number of images from which the texture information was obtained is small. Small data sets are typical in biomedical research. Early research in a field, such as this investigation, is rarely able to obtain input from several health centres, so a single health centre is used. When one can obtain data from a large health centre, available numbers can still be low. Typically, inclusion issues will reduce the number of images that can be used. Inclusion issues can include problems obtaining consent, varying image quality, changing clinical protocol, patient relocation, and changing technology. In this work, the patient images are limited to those available from the IWK Health Centre from 1990 to 2000.

The image data reflects biological properties so it cannot be assumed that it is normally distributed. Also, the small number of samples that form the data sets, 69 patients from EXP2, mean that the central limit theorem does not apply. The texture measures exploited in this work are new and so we do not have prior information or likelihood approximations. These factors impact the type of classifier chosen, particularly if the classifier model makes assumptions about the underlying probability distribution of the data set.

3.3.2 The Available Classifiers

Classification systems are generally divided into two groups. There are those from classical statistics and those from machine learning. At times the two views about classification systems are at odds [110].

Classifiers based on classical statistics assume that the class populations have easy-to-model relationships within and between the classes, making them easy to interpret. Unfortunately, the creation of a mathematically tractable model can take precedence over the classifier's performance. Thus, the relationships described by the models are easy to see but may be incomplete.

Classifiers from the machine learning world are more like black boxes. The relationships within the classification system are complex and can be hidden from the user. Prediction problems in complex systems, such as biological systems, may be solved more effectively by using machine learning techniques. Unfortunately, their structure can be impossible to interpret and thus it may not be possible to obtain information about the real world system from them.

No one type of classification system will perform optimally in all situations. The strengths and limitations of various classification methodologies must be considered and matched with the properties of the data and the classification goals. To elaborate, because the purpose of designing the classifier is to test a hypothesis, care must be taken to ensure that the classifier model can reveal the information content of the measures. Regardless of the information within the measures, the performance of a classifier can be poor due to incompatibility with the problem. As well, the performance measures may be overly optimistic because the classifier is overfit to the data and thus is not generalizable to the entire population. First, the means by which the performance is measured will be presented. A review of various classifiers and their performance issues is presented, after this the best fit to this problem is selected.

Measuring the Performance of a Classifier

Types of Error Measures Predictive error is often considered to be the most important measure of performance. However, when discussing an error measure, one must also have a measure of confidence in its value. The predictive error of classifiers can be described in terms of their bias and variance. The accuracy of the predictive error depends on the stability of the classifier.

Medical diagnosis is typically a two-class problem. Either the disease is present (true) or the disease is absent (false). Diagnostic error is usually presented in terms of several descriptive measures. When calculating these measures for a given test sample, one needs to know the number of positives that are classified as positive correctly, TP, the number of negatives that are classified as negative correctly, TN, the number falsely classified as positive, FP, and the number falsely classified as negative, FN. Overall prediction error is the number of mis-classifications divided by the total number of samples classified, given by:

$$error = \frac{(FP + FN)}{(FP + FN + TP + TN)}. \tag{3.1}$$

Accuracy is described by 1−error, which is also referred to as efficiency. Sensitivity:

$$sensitivity = \frac{TP}{(TP + FN)}, \tag{3.2}$$

is a measure of how often a test detects a disease when it is truly present. Specificity:

$$specificity = \frac{TN}{(TN + FP)}, \tag{3.3}$$

is a measure of how often a disease is correctly indicated as absent. Each classification error, FP and FN, has a different cost associated with it. The cost depends on the severity of the disease and the potential for harm if treatment is withheld or applied unnecessarily.

Additionally, diagnostic tests are described in terms of positive predictive value, PPV,

$$PPV = \frac{TP}{(TP + FP)} \tag{3.4}$$

and negative predictive value, NPV, [111]

$$NPV = \frac{TN}{(TN + FN)}. \tag{3.5}$$

For the remaining discussion about classifier selection, statements about error are limited to predictive error or accuracy.

Methods to Obtain Error Measures Like most statistical measures, predictive-error measures are estimates of the true error for the entire population. An estimate of predictive error can be obtained several ways. They will be presented in order of the least to the most representative of the classifier's performance for a population.

The least valuable is to design and validate the model on all members of the same data set. This is also known as the resubstitution error estimate. An alternative is to design and validate the system on two separate data sets. While this approach offers improvement, the selection of the two separate sets can be arbitrary and the samples included in the sets can significantly influence the confidence in the error measure. Also, for small data sets the creation of a separate test set requires reducing the size of the design set [112].

The remainder of the validation methods consist of the use of a single data set for both design and test. The methods discussed are performed in a manner to achieve relatively low-bias measures of performance and allow the learning stage to be performed on the maximum number of samples.

Cross-validation offers significant improvement over the resubstitution estimate. The classifier is designed with the entire data set. To obtain a measure of the error, the sample is divided into N equal subsets, the design is again performed with N-1 subsets, and is validated with the left-out subset. Design and validation continues until all N subsets are used for validation. The N measures of error are averaged to provide a measure of the error for the system designed by the entire number of samples.

Additional improvement comes with bootstrapping, in which the samples are drawn with replacement from the original data set. The error measure is obtained by using those samples not included in the design process. The bootstrap was introduced as a method for estimating the standard error of any population statistic estimated from a random sample. As the number of bootstrap samples approaches infinity, the estimated standard deviation approaches the population standard deviation [113]. In error analysis, bootstrapping is an extension of cross-validation with the addition of randomization of the included and left-out samples.

Once there is an estimate of the predictive error of the classifier, we need to consider the cause of the error and the variability of the measure of predictive error. Assuming that our samples provide an adequate representation of the population, the main source of error in our designed classifier is its bias. Bias provides a measure of the match between the classifier model and the real-world relationships between the variables. Bias is described as the distance between these two entities. The bias of a model is dependent on underlying assumptions about the data. If there are assumptions about the data that are largely in error, the classifier may have a large bias. For example, classifiers that assume that the measured data has a Gaussian distribution have large bias and may not predict well when the data do not have a Gaussian distribution.

Variance provides a measure of how much the classifier would change if the training sample changes. Therefore, it provides a measure of confidence in the error estimate. With a low-variance classifier, the performance is consistent. A high-variance classifier will tend to over-fit the sample data. Bias and variance measures are related in a non-linear fashion and the tendency is a trade off between them [114].

Another related descriptor of classifier performance is stability. In the training process, if the loss of a small percentage of the data points results in a markedly different classifier, then the classifier is said to be unstable. Instability can be caused by the algorithm that creates the classifier or by the use of a small data set to train the system. Either way, unstable classifiers do not generalize well.

3.3.3 The Theoretically Best Classifier

A Bayesian decision system puts the classification problem into probabilistic terms. Let there be $K_i = 1,..i$ classes and let $\bar{x}$ be the input vector of predictor variables. The probability that a sample belongs to class K_i having measured $\bar{x}$ is given by:

$$p(K_i/\bar{x}) = \frac{p(\bar{x}/K_i)p(K_i)}{p(\bar{x})} \tag{3.6}$$

where:

$p(\bar{x}/K_i)$ = the probability of measuring $(\bar{x})$ for class K_i,

$p(K_i)$ = the prior probability of class K_i ;

$p(\bar{x})$ is the probability of the feature vector space.

These three probabilities are also known as the likelihood, the prior, and the evidence, respectively. Assuming that the cost of misclassification is the same for all errors, the Bayes classifier assigns the input to class K_j when $p(K_j/\bar{x})$ is the maximum of all $p(K_i/\bar{x})$. The Bayes error is the lowest that a classification system can achieve. The problem with this classification method is that the conditional probabilities must be known but may be impossible to obtain. The likelihoods can be estimated by obtaining samples and assuming that the underlying population follows a known probability density function. Some classifiers in classical statistics are based on Bayes theory and the assumption of a normal distribution.

3.3.4 Classification Techniques from Classical Statistics

Some classical statistical techniques include linear discriminant function, quadratic discriminant function, and logistic regression.

Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are derived from Bayes theorem and the assumption of a normal population. LDA assumes identical covariance matrices of the classes and that the difference between the classes can be defined by hyperplanes. QDA assumes differing covariance matrices and that the classes can be separated by a hyperquadratic [115].

Logistic regression is similar to linear regression except that the data are modeled by a a sigmoid rather than a line. The dependent variable represents the predicted class. The formula for logistic regression is in the form

$$C = \frac{1}{1 + e^{-(a+b\bar{x})}} \tag{3.7}$$

where

C is the predicted class,

a and b are the parameters derived during training [116].

Figure 3.24 shows a classifier for a single variable, this can be extended into a multidimensional space. Logistic regression does not require that the predictor variables be multivariate normal but there must be at least as many cases as features.

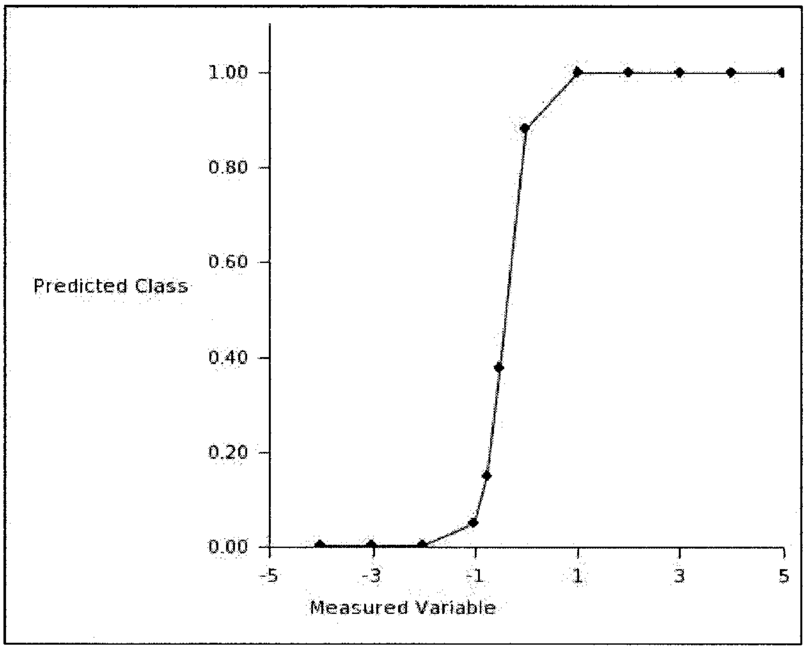

Figure 3.24: A sample logistic regression curve used as a classifier. The x axis is the measured variable, the y axis is the predicted class.

The "curse of dimensionality" is a term first introduced by Bellman [117]. As the number of dimensions increases, the number of cases required to span the space grows exponentially. Besides the limitation in predictive ability caused by the underlying model, classifiers from classical statistics suffer from the curse of dimensionality. The mathematical formulas used to create the statistical models require that the number of measured features be much less than the number of samples. For systems that are inherently high-dimension spaces, methods are required to reduce the number of features prior to the design of the model.

The techniques from classical statistics tend to have high bias and low variance. The classifiers tend to be stable but the models fit poorly due to the bias. With these negatives, why would one choose classifier methods from statistics? There are many positive aspects to these classifiers. They are easy to understand and have standard, well-understood performance measures. In situations where the model assumptions are not violated and even in some cases where they are, they have performed well. Problems with large data sets and few variables tend to lend themselves to these methods.

3.3.5 Classification Techniques from Machine Learning

Machine learning classifiers (MLC) include genetic algorithms, k nearest neighbors (kNN), ANNs, and decision trees. These models tend to be low bias; however, they require many training cases, sometimes numbering in the thousands, and may not converge to the best model. After the creation of the classifier, the underlying relationships between the feature variables can be difficult if not impossible to extract. MLCs can be sensitive to user-specified components and the design process tends to be *ad hoc*. Initial conditions of the variables can have a significant effect and are frequently set by random number generators. These models tend to be unstable and have low bias and high variance.

Genetic Algorithms and kNNs

Genetic Algorithms are iterative processes where a trial classifier is evaluated and subsequent changes made to it through mutation, addition, deletion, and preservation of the sub-rules of the classifier. The stopping point of the process is determined by goodness-of-fit measures.

kNN is a system where a new sample is classified based on the relative location of the training set within the feature space. The majority class, within the neighborhood of the new sample determines the class that is predicted.

ANNs

An ANN is a non-linear classifier based on an idealized model of neural networks. A single-layer ANN is illustrated in Figure 3.25. In the single-node ANN, the inputs are combined via an equation for a hyper-plane and then the output is passed through a non-linear function that performs a thresholding operation. A multi-layer ANN is formed by using the basic structure as a building block and forming a much larger network with them. The outputs of one single-layer network, or node in a large ANN, are used as inputs to the next stage. All nodes at the same depth from the input data form a layer as shown in Figure 3.26. The neural network must be trained to determine the weights and bias values. Typically, thousands of samples are required to train the network. There are

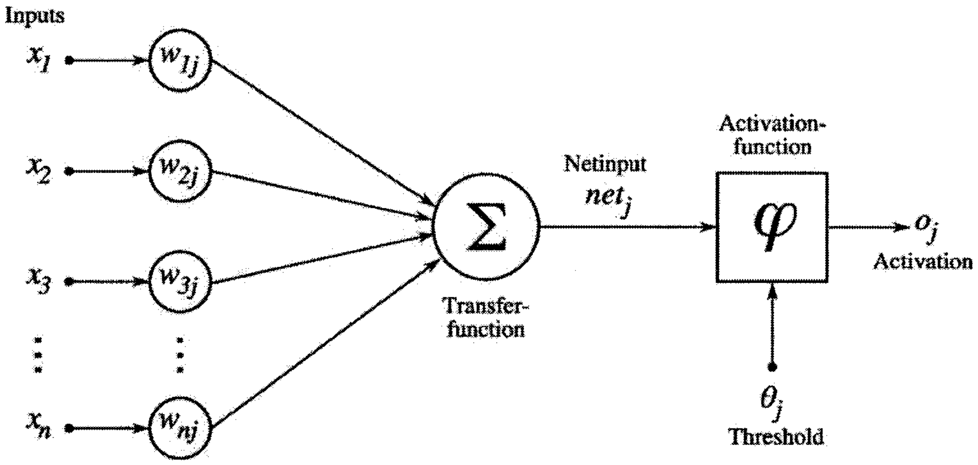

Figure 3.25: Single layer ANN classifier. Image obtained from www.commons.wikimedia.org under GFDL.

several algorithm options for training the network including gradient descent and back propagation. A common problem in neural networks is that the training algorithms are not stable and may not converge to the optimal model. As well, the addition of many non-linear components makes the extraction of the relationships between the predictor variables impossible for large, complex ANNs.

Decision tree

In this classification system, a series of decisions is made by thresholding a feature or a combination of features until the sample is classified. The structure consists of nodes, branches, and leaves with a shape not unlike a tree (Figure 3.27). The nodes within the tree represent the locations of decisions causing splits in the logic path. The branches indicate the order of the nodes and the leaves represent a class designation. In the tree-creation process, nodes are added until all members of the training set are classified. The feature for decision making and the choice of threshold at the node is based on information gain. Two common methods of measuring information gain are entropy and homogeneity of the group after split. Entropy, H, is used by the algorithm C4.5 and calculated by [118]

$$H = \sum_{i=1}^{n} P_i log_2 \frac{1}{P_i} \tag{3.8}$$

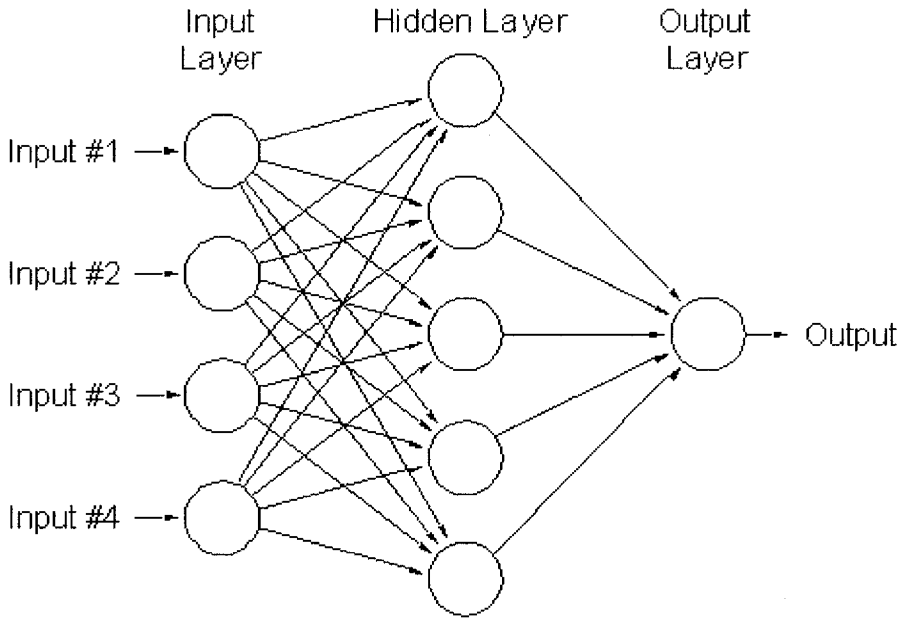

Figure 3.26: A multi-layer ANN classifier.

where

P_i is the probability of event i.

Homogeneity is implemented by CART using the GINI index [119]. The GINI index, I, is

$$I = \sum_{i \neq j} p(i/t)p(j/t) \tag{3.9}$$

where

t is the node being created,

i, j are the classes. For a two class problem, this reduces to

$$I = 2p(1/t)p(2/t). \tag{3.10}$$

Some algorithms incorporate tree pruning to reduce the complexity of the resulting classifier.

3.3.6 Ensembles of Classifiers

An ideal classifier has low bias and low variance. Towards this goal, ensembles of classifiers have been investigated. By aggregating many classifiers in an appropriate way, the variance

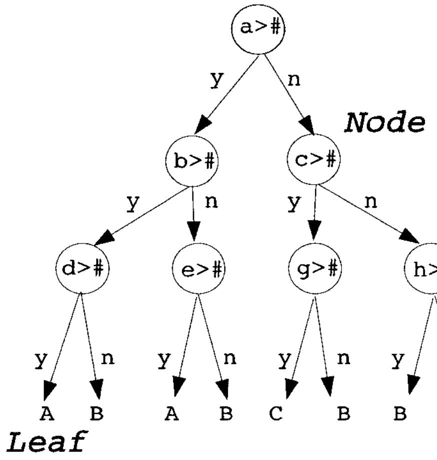

Figure 3.27: Decision tree classifier.

can be reduced so that the model's error reflects the error that would be obtained by applying the model to the true population. By using a low-bias model as the base unit of the ensemble, it is possible to create a system closer to the ideal. Decision trees are ideal for forming ensemble classifiers as they have low bias and the methodology for creating them is well understood. There are many ways to form ensemble classifiers but generally they are formed by perturbing and combining. Perturbing and combining (P&C) is a method to create multiple versions of a predictor by perturbing the training data or the construction method and then combining the group into a single predictor through voting. There are two P&C methods, adaptive re-sampling and combining (arcing), and randomization algorithms.

Arcing

In the creation of the trees forming the ensemble, extra weight is added to those difficult-to-classify points. Also known as adaptive boosting, this type of algorithm has been shown to perform better than some randomization algorithms. The trees of the ensemble are grown sequentially. The more frequently a sample is misclassified, the larger its subsequent weighting becomes so as to favor the eventual correct classification of the sample. There are questions as to when to stop the training process. It has been shown that if the learning algorithm continues past the point of zero training-set error, the designed classifier will perform better than if the algorithm stops at zero [120], as tested on separate data sets.

Randomization Algorithms

Randomization algorithms are ensemble classifiers that are based on the injection of randomness. This reduces the over-fitting associated with many classifiers. The randomness may be injected in several ways including bootstrapping and the random selection of a subset of features for decision at each node.

Bootstrapping is the term used to describe the process for generating a sample of data by sampling with replacement from the original data set. It has been shown that the tendency is for bootstrapped samples to consist of approximately 2/3 of the cases from the original data set with 1/3 of the cases from the original data set left out [121]. Bootstrapping is performed many times to create an ensemble of classifiers, one classifier from each of the bootstrapped samples. This process is referred to as bagging. Bagging has been shown to perform well when there is classification noise [122]. Typically, decision trees, created using the binary recursive partitioning algorithm, form the components of the ensembles.

Generally speaking, there are outliers in any data set which are more difficult to classify. With a boosting algorithm, the difficult-to-classify samples are additionally weighted during the iterative learning algorithm. In this way, the unique relationships between the outliers and the feature variables are ensured inclusion into the ensemble. The algorithm still incorporates bootstrapping for creating the ensemble but the re-weighting improves the classification accuracy. Adaboost is a well known boosting algorithm which performs well compared to other randomization algorithms [123].

The random forest method of Breiman and Cutler [89], [124] adds an extra level of randomness. While bootstrapping is used to form a group of trees with low-correlation, node creation is based on a random subset of the available features that form the data set. At each split, a random subset of the features is selected from the entire set and is polled to determine the best for the node split based on the CART algorithm. Where no individual measure performs best, a feature is picked at random. The number of features making up the subset can be determined by the user but the number chosen does not have a large effect on the ensemble performance [125]. The Random Forest algorithm compares very favorably to other ensemble methods [126] and the algorithm is readily available as a package, randomForest, in R system for statistical computing and graphics (R) [127], [128]. Beyond the model's performance, the algorithm has the advantage of providing measures of importance for all of the input variables and handles a large number of variables well.

3.3.7 The Choice: Random Forest Algorithm

The texture data, with 69 patients and 256 measures per patient, consists of a high-dimensional, small sample set. We have little prior information about the measures except the expectation that they have non-Gaussian distributions. The random forest algorithm was chosen because it is a non-parametric model and it has proven performance compared to other classifiers. In a study that benchmarks SVMs with 16 classification methods using 21 data sets, random forest performed comparably and outperformed all others in several instances based on test set errors [126]. The random forest algorithm has also exhibited good performance on high dimensional data sets. Li *et al.* and Izmirlian study the use of classifiers and gene expression profile data for cancer markers [129] [125]. While Li *et al.* are investigating a novel method of gene selection and classifier creation, their report indicates that the RFC often performs comparably in this high dimensional problem. For these reasons, the choice was to implement a decision-tree ensemble classifier created through the random forest algorithm. The algorithm is implemented in R, which is a freely available statistics and graphics language and environment. The random forest algorithm was originally written in Fortran77 by Breiman and Cutler [124] and was ported to R, as the randomForest package, by Liaw and Weiner [127].

As discussed previously, a forest of decision trees is created. Each individual tree is created through the CART algorithm. A summary of the algorithm is as follows:

Let N be the number of samples available from the population. Let F be the number of measured features for each sample. Let B_k be the bootstrap samples created from the original data set, ntree be the number of bootstrap samples, so $k = 1, \dots, ntree$. A tree is created for each bootstrap. Each bootstrap sample is made up of "in-bag" cases while those cases not selected to be included in the bootstrap sample are called "out-of-bag" (OOB) cases. The growth of each tree in the forest consists of finding optimal binary splits at each node based on a randomly selected subset of all candidate features, mtry.

During the creation of each node, the feature that provides the best split is selected as the decision criterion in the logic path. The CART algorithm uses the GINI index as the basis for selecting the best feature. When there is no best feature, a feature is chosen from the candidates at random. Unrestricted splitting continues until there is exactly one case per leaf of the tree, or all cases in each of the leaves are identical. These conditions indicate that further splitting is not possible. Once the random forest classifier is created, new samples are classified via voting by the ensemble.

An advantage of the random forest methodology is that a separate data set is not required for validation. For each tree, an estimate of predictive error is calculated using those samples not used in the growth of the tree and the classification is compared to the ground truth. This error estimate is known as the OOB error. The final error estimate for the entire forest is the average of the OOB errors over all of the trees in the forest. In this way, blind testing is incorporated in the algorithm. The forest OOB error estimate has been found to provide the upper bound to the error obtained with a separate test set [130], [89]. Specifically, in a 1996 publication Breiman [130] provided theoretical justification for its accuracy and provides results of experiments comparing separate test-set error measures to OOB error measures. In all cases the OOB was equal to the separate test-set error or was slightly pessimistic.

When using the randomForest package, the user has the option to modify a number of model parameters. These include the number of trees in the forest, the number of features selected as candidates at each node, and the cutoff which specifies the voting scheme by the trees. The default number of trees is 500, and the user needs to consider increasing this in order to ensure convergence if the sample size is small or if the signal in the measures is weak. There are no strict rules for the size of ntree, but some guidelines have been noted. The number of trees grows with the number of predictors [128], and the number of trees should be large enough that the ensemble statistic of interest has stabilized [131]. The user should not be concerned about negative effects caused by setting ntree too large. While increasing the size of ntree will increase computation time, it will not cause the forest to over-fit [89]. The variation in the classification error with ntree has been demonstrated to converge by Pal [91].

When the randomForest algorithm is being used for classification (rather than regression or clustering), the default for mtry is $\sqrt{F}$. It has been suggested that the algorithm is relatively robust to the setting of mtry [124]. However, others suggest that the algorithm is robust to a variety of mtry values as long as it is not set at 1 or F (the maximum or minimum values) [131]. This is because because setting $mtry = 1$ creates an ensemble of random trees, removing the ability of the algorithm to pick the best feature from a random group. Setting $mtry = F$ reduces the algorithm to bagging, removing the random component out of the node creation. Svetnik *et al.* suggest that the optimal choice of mtry depends on the proportion of irrelevant variables in the training data [131]. There is a function within the randomForest library, called tuneRF, which selects the optimal mtry value for the data in question. The cutoff allows the user to determine the voting scheme of the forest. The default value of the cutoff is a majority voting scheme where 50% + 1 votes determine the predicted class. In a two class problem, this default is represented as a vector with values (0.5,0.5).

The manual pages, from randomforest.pdf, showing the call to the randomForest algorithm within R is provided in Appendix H, complete with the parameter default values.

3.4 Feature Dimension Reduction

When using classical statistical techniques to create a model, feature dimension is essential if the sample is over-parametrized (the number of measured features is larger than the number of cases). While it may not be essential for machine learning techniques, they can benefit by improved performance if the noisy features are removed prior to classifier design.

Feature dimension reduction can be accomplished through feature selection (FS) or feature extraction (FE). Feature extraction transforms the data to a new coordinate system, resulting in features which are combinations of the original features. Principal component analysis (PCA) is one commonly used method for feature extraction. This method proposes a linear combination of the input variables based on the eigenvectors of the data. However, it is designed to preserve those combinations that span the variance of the sample set and does not necessarily preserve the features which discriminate between the classes [132],[133]. Other FE methods include singular-value decomposition and independent component analysis.

Feature extraction is generally considered to be more effective than feature selection [134]. However, FE methods have the disadvantage of losing the meaning of the variables that were measured. This can be a problem when it is desirable to relate the features to a physical or biological phenomenon.

Feature selection identifies a subset of the original features based on some performance criteria. Feature selection is particularly valuable when the original measures have a meaning that needs to be retained for interpretability. Filtering is a FS technique in which the performance criteria is separate and independent from the classifier methodology. The features may be graded independently or in subgroups of the original set. For example, individual features may be assessed by their between-class distances. The distance between classes for each feature is measured, used to rank features, and the lowest ranked features are dropped. Often the number of features retained is based on some threshold.

Unfortunately, FS methods can lead to sub-optimal solutions. Assessing features in groups can be prohibitively computationally expensive. This leads to evaluating features individually. Grading features individually can lead to sub-optimal solutions by overlooking combinations that perform well but whose constituents are all weak individually. Additionally, groups of strong individual features may not perform better together because they may be correlated.

Feature selection can also be performed using wrappers or embedded methods. These methods apply the classifier algorithm to a subset of the measured features and then grade the subset according to the classifier's performance. The subset that results in the best-performing classifier is selected as the feature set. Wrapper methods learn the solution through an exhaustive analysis of all feature combinations. Embedded methods use a subset based on a ranking criteria that is part of the learning algorithm [135]. For example, the classifier may be designed with all features and then the lower-ranked features successively removed and the classifier performance re-evaluated. This is a backward elimination method. The reverse, keeping only the best ranked variable and adding to it, is known as forward elimination. It should be noted that in the literature, the term wrapper can refer to both wrapper methods and embedded methods as described here.

Feature selection techniques are popular because they are always less computationally complex than feature extraction techniques. Unfortunately, FS techniques do not always find the optimal solution. Another issue to consider when applying FS algorithms is multiplicity. Multiplicity refers to the phenomenon in which there may exist several optimal solutions to the FS problem. This situation will confound the extraction of meaning from the selected feature variables.

Because randomForest includes measures of variable importance, wrapper algorithms incorporating variable-importance measures are a natural extension of the modeling proceedure. There are two measures of variable importance provided with the randomForest package. The first type is the mean decrease in accuracy. This is measured as follows. Each predictor variable is permuted and the difference in each tree's OOB error caused by the permutation is calculated. The difference measures are averaged over all of the trees and normalized by the standard error. The second measure of variable importance is the decrease in node impurity caused by splitting on the variable. For classification, the measure of node impurity is the Gini index. Recent work in FS incorporating randomForest variable importance is presented.

Jiang *et al.* proposed a gene selection method incorporating variable importance as calculated by the mean decrease in accuracy [136]. Nested subsets were identified by repeatedly calculating the variable importance and dropping the lowest ranked 10% of the features. This was done until there is only one variable left. The best subset was chosen based on the OOB error estimate, which was calculated for each subset. The prediction error for the classifier, created with the selected genes, was calculated with an independent test set.

Svetnik *et al.* also propose a gene selection method based on the mean decrease in accuracy in a backward elimination method [131]. They do not use the OOB error estimate as the measure of performance with each variable subset. They argue that if variables have been selected based on the OOB error rate, the selected variables are over-tuned to the data set and thus the OOB error does not generalize to the true population. They demonstrate the over-fitting problem using a data set containing only noise. The model error with the noise-only data set is erroneously measured as 35% when using the OOB error estimate to assess the model performance and for variable selection. To address the issue of obtaining a generalizable error estimate, they use 5-fold cross validation to obtain the error of the predictor model.

Diaz-Uriarte *et al.* [137] propose a method in which they calculate the variable importance once and iteratively drop the lowest-ranked variables. The OOB is used to determine the best subset, but a bootstrap method is used to assess the prediction error of the final classifier. They argue that their procedure does not lead to overfitting and aggressively reduces the number of variables. To confirm that their algorithm does not overfit, they test their procedure on a data set without signal and obtain a larger number of variables with an error equal to betting on the most probable class.

The problem common to the use of wrapper algorithms is the need to create an unbiased estimate of the model's performance. This can be particularly difficult when an additional test set is not available or the sample size is small. As well, algorithms provide tests but they do not contain domain knowledge. Studies have been performed to assess the performance of variable-selection algorithms [138], [139], [140]. It is reported that forward- and backward-elimination techniques can provide sub-optimal solutions, with at least one study determining that noise is included in 20% to 74% of the selected variables and less than half of the actual predictors were selected [140]. For these reasons, a filtering method, unrelated to RFC, was applied to the EXP1 data using engineering knowledge and simple statistical tests. This allowed the use of EXP2 to design and test a model, with the OOB error able to be used as a low-bias estimate of the model's performance.

Chapter 4

Experiments

4.1 Extraction of the Texture measures

4.1.1 Preliminary

The experiments consisted of the extraction of texture measures and the design and testing of the classifier. Previous experiments, Figure 3.7, and knowledge of the disease process of WMD suggests that the texture measures will contribute to diagnostic capability. The RFC using the randomForest package in R was selected to design and test the model to verify this. The following sections will describe the computation of the measures, the extraction of the data, and the design of the RFC.

Experimental Data

The images were obtained from the library of images at the IWK, spanning the years 1990 to 2000. The use of digital ultrasound machines at the IWK Health Centre began in 1992. For this reason, the images are separated into 2 groups. Those created from 1990 and 1991 were identified as group EXP1 and the remainder as group EXP2. All images were scanned with the same Epson Perfection 3200 Photo scanner with 3200 DPI (hardware) at 2^{16} grey levels to maintain the original high resolution of the images. The scanner saved the images in tagged image file format which were than converted to portable grey map format, which has 256 grey levels.

If the quality of the image was deemed sufficient for clinical use, it was included in this study. To obtain a clinically suitable image, the operator had the ability to adjust various settings including gain, time gain compensation (TGC), and post-processing. While there are no quantitative measures used to assess clinical image quality, a qualitative assessment of quality does exist. The operator attempts to create an image which is side to side symmetrical, an image as large as possible, and the soft tissue should appear similar in brightness throughout the depth of the head. All patient images consist of a single coronal image through the atria of the lateral ventricles. None of the images contained cystic echoluecencies or flares.

The EXP1 data set consists of images produced with analog machines and recorded on film. This data set consists of seventeen patients, five of whom subsequently developed CP. The sample size is small for several reasons. The EXP1 data set is restricted to those patients born over a two year period, due to changes in imaging technology occurring in 1992. The population from which the images could be drawn is small, as cerebral palsy has a local incidence of 11% within the preterm population of about 50 children per year. As well, some eligible patients from this period had follow-up care elsewhere, making their records inaccessible. Three ultrasound machines were used to obtain these images. The machines include model XP128 made by Acuson, a Diasonics manufactured machine (model unknown), and model HDI 5000 made by Advanced Technology Laboratories (ATL). The center frequency of the acoustic pulses ranged from 5 to 7.5 MHz. This data set was used for the design of the appropriate image-processing and texture-measurement techniques.

The EXP2 data set consists of digital images stored on film. The EXP2 group originally consisted of 74 patients, case matched and selected by an independent third party, M.J. Vincer, Director of the Perinatal Centre at the IWK Health Centre. The case-matching criterion included gestational age and gender, and no patient was matched to his or her twin. However, this study included restrictions on the age of the first scan, and therefore the numbers were reduced from the original group. The result is that EXP2 consisted of 37 patients with normal outcomes and 32 patients with CP. The final diagnosis of CP is a result of multi-disciplinary follow-up, as is the normal clinical practice at the Perinatal Centre within the IWK Health Centre. The images in the EXP2 data set were obtained from two ultrasound machines. The images were obtained with two ultrasound machines. One was manufactured by ATL, now part of Philips, model number HDI 5000, at a center frequency of 8.5MHz. The other is an Acuson ultrasound machine, model number 128XP, at 7.0 MHz. The ATL machine was used for 30 cases: 16 CP, 14 no CP. The Acuson machine was used to obtain the images for 39 cases: 18 CP, 21 no CP.

Extraction of White Matter and Choroid Plexus ROIs

The extraction of the white matter and choroid plexus ROIs was performed with a semi-automated system. This was chosen as segmentation of tissue types and the selection of landmarks within cranial US images are non-trivial tasks. The methods used to extract the ROIs were selected with the goal of standardizing the method through automated methods in the future. Landmarks, intensity levels, and standard geometric shapes were used within the algorithms where possible.

The choroid plexus was identified by a radiologist using an interactive program (choroid.cmd) and a mouse to identify points within the choroid plexus boundary. The program joined the points with straight lines to form a closed boundary that was filled to form the final mask. The choroid plexus is relatively easy to locate by its increased echogenicity and with the radiologist's anatomical knowledge. Care was taken to ensure that no other tissue types were included in the sample. There are two choroid plexus ROIs identified per patient, also identified as masks, one on each side of the brain. Figure 3.3 illustrates the choroid plexus and white matter masks.

The periventricular white matter was identified by first locating the upper-most and peripheral tip of the ventricle within the image. Once this point was located, the radiologist located the inter-hemispheric fissure and identified a point on it, having the same vertical coordinate as the ventricle tip. Once these two points were identified, a circle was constructed with its center at the ventricle tip and its radius equal to the distance between ventricle tip and the point on the inter-hemispheric fissure (whm.cmd). This circle contains a sample of white matter and ventricle. The boundary of the ventricle was then identified with the same methodology as the choroid plexus. The traced ventricle included cerebral spinal fluid and choroid plexus completely within its boundaries. The ventricle region was then subtracted from the white matter circle leaving a white matter mask containing only white matter and germinal matrix. Two white matter masks were identified per patient, one on each side of the brain. See Figure 4.1 for an illustration of

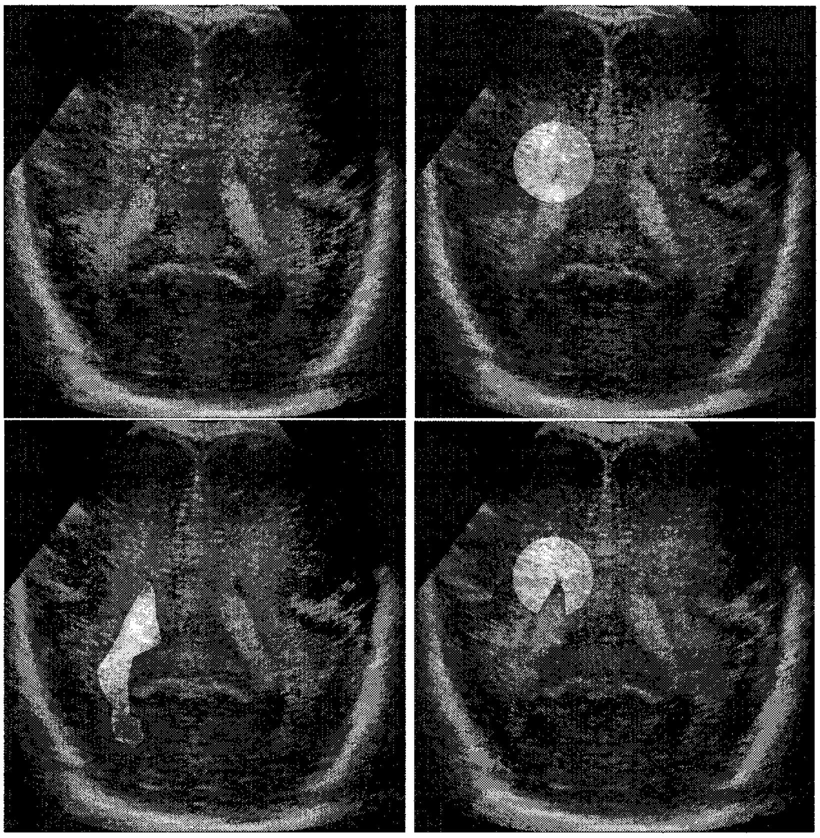

Figure 4.1: Top, left to right: Two points extracted by the radiologist, resulting circle. Bottom, left to right: Traced ventricle, final white matter ROI.

the process that leads to the white matter ROIs.

Creation of Angle Look-up Image

The angle look-up image was created for each patient image. To reduce computation time, it was obtained from the image after two-reduction steps of the Burt-Adelson method of reduction. There are two main steps to the calculation of the angle look-up image; the identification of the signal region of the image and the calculation of the theoretical point-source of the pulses.

An approximation to the signal portion of the image is obtained by the application of a threshold. The threshold is based upon the statistics of the image, with the threshold as a percentage of the mean and standard deviation. The thresholds were chosen manually to suit the individual, with the goal of extracting the signal-only region of the image, (Figure 4.2). Due to varying image quality and generally decreased echogenicity of the white matter, this resulted in some gaps in the side boundary. These gaps were filled by manually inserting a line to complete the side boundary and then filling in the resulting hole. Automated methods to obtain the signal portion of the image were not pursued as the signal portion is more clearly defined in modern, digital US equipment.

Once the signal portion of the image was identified, the angles at every point were located with the algorithm ultgrid.cmd. As a simplifying assumption, the transducer is treated as a point source. The algorithm for creating the angle look-up has three options available for the calculation of the point source and they are presented in order of use by the algorithm:

1. If the bottom of the signal area in the image has a distinct boundary from the rest of the image, it is considered to be an arc of a circle. The center of arc curvature is assumed to be the insonation point.

2. If this is unavailable and the top of the image is a circular arc, then this is used to calculate the insonation point.

3. If neither arc is available, then the two sides are extended until they meet at a point, and this is considered to be the insonation point.

See Figure 4.3 for an illustration of these options. Each method resulted in a slightly different set of coordinates, however the effect of this on the look-up image is a variation of only a few degrees for each point in the image. While the goal is precise identification of the direction of isonation, this slight variation is not expected to have a significant effect.

After calculation of the theoretical point of insonation, all points in the image with the same x coordinate as the insonation point were designated as having an angle of 180^o and the angles were incremented in the clock-wise direction with respect to the viewer. This image was stored for later use by a number of processing techniques.

Noise Reduction

The median filtering (37×37 window) for noise reduction was performed on three separate personal computers. Each of the computers is equipped with 2G RAM and a 1.6GHz

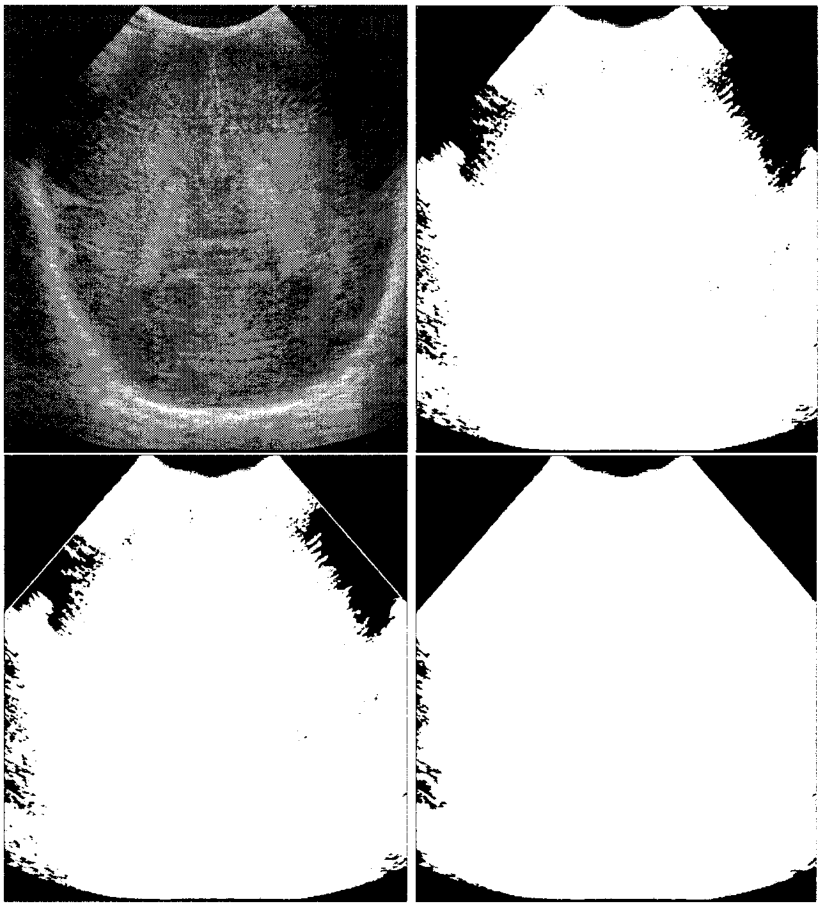

Figure 4.2: Top: Original image and after thresholding. Bottom: Image after manually connecting the outer boundary and the final mask of the signal region within the US image.

processor as a minimum. By using these computers, each image could be entered and median filtered in its entirety. The quantity of RAM is the limiting factor in this operation as filtering is performed using the full-size images. The median filtering required several hours per image, depending on the size of the image and which of the three machines were used.

4.1.2 Parametric Maps

The image processing required to obtain each of the parametric maps required significant processor time, on the order of several days per map per patient. To be able to complete this project in a reasonable amount of time, additional resources were required. The

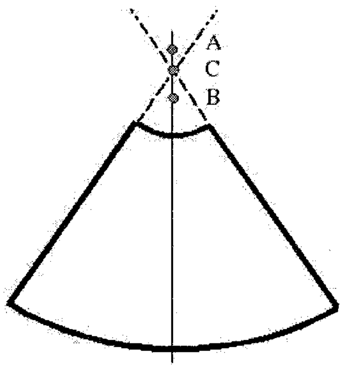

Figure 4.3: Three sets of coordinates that may be located as the theoretical insonation point. A: Based on bottom arc. B: Based on top arc. C: Based on sides of the signal area. The variations in the three points have been exaggerated for the illustration.

Department of Astronomy at Saint Mary's University has a cluster, Pluto, of 24 computers each with parallel processors. The Department had some processing time available on the machines and kindly allowed their use in this project. Some modifications to CVLab had to be made before the algorithms could be run on Pluto. As the algorithms would be initiated remotely using the command prompt, all functions that incorporated the window display features had to be removed from CVLab before loading onto Pluto.

On Pluto, each process is limited to 33M RAM. As each process had at least one input image, and created one output image and typically several intermediate images, the complete image could not be processed as a whole. The patient images were cropped by identifying the minimum region required to contain the 2 white matter ROIs and the two choroid plexus ROIs. With these modifications, the preprocessing and subsequent map creation were performed on Pluto, requiring approximately 4 months of continuous processing time, with a range of 8 to 20 batches running at any given time. See Appendix I for sample script files for processing and map creation, and the batch files used for control. The DM and GABT19 processing, and the creation of all STDV and NSA parametric maps were performed on Pluto. IDD30 and ORIG processing were performed on three PCs. These techniques were performed on the full-size images which required more RAM than the 33M RAM available on Pluto.

The statistical measures and the NSA were obtained for each parametric map. The parametric maps are smaller than the original ROIs extracted by the radiologist due to the edge effects of the moving window within the ROI boundary. Through binary thresholding, using the value of 1 as the threshold, the ratio maps were used to create a new mask. This threshold is possible, as the maps were scaled by a factor of 100, to maintain two positions pat the decimal point when storing the output images in the integer format. The ratio maps always have non-negative values, unlike the difference maps. Using this new mask, the algorithm mapinfoavg.cmd extracted summary statistics and the NSA within the map regions, found the average of each value from the two sides, and wrote this information to a text file with the appropriate formatting for later use in R. See Appendix J for a sample script file extracting the data from the parametric maps.

4.2 Variable Selection

To start with, there are 256 variables obtained from the EXP1 data set. This number of candidate variables needs to be reduced as model performance is affected by noise. If there are many non-contributing measures, the RFC can form decisions at nodes or entire decision trees based on noise only. As well, the presence of highly correlated variables will cause the effect of a single true information-containing measure to be split between many variables. It is therefore desirable to remove the noise and reduce the number of correlated measures. The selection of the variables was performed in several steps. The first step was to restrict the measures according to some simple engineering observations. The images undergo logarithmic compression. Therefore, any relationship between the texture measures of the white matter and the choroid plexus that could be expressed by a ratio, prior to logarithmic compression, should now be apparent as a difference. For that reason, all measures from the ratio-based comparison were discarded for this analysis, reducing the number of measures per patient to 128.

Originally, 8 measures were extracted per patient. The measures consist of the NSA, standard deviation, mean, median, maximum, minimum, skewness and kurtosis. The maximum and minimum were discarded because they are not stable measures. As well, error is introduced by rotating a kernel or area under investigation within a discrete space, causing the maximum and minimum values to be functions of location within the discrete

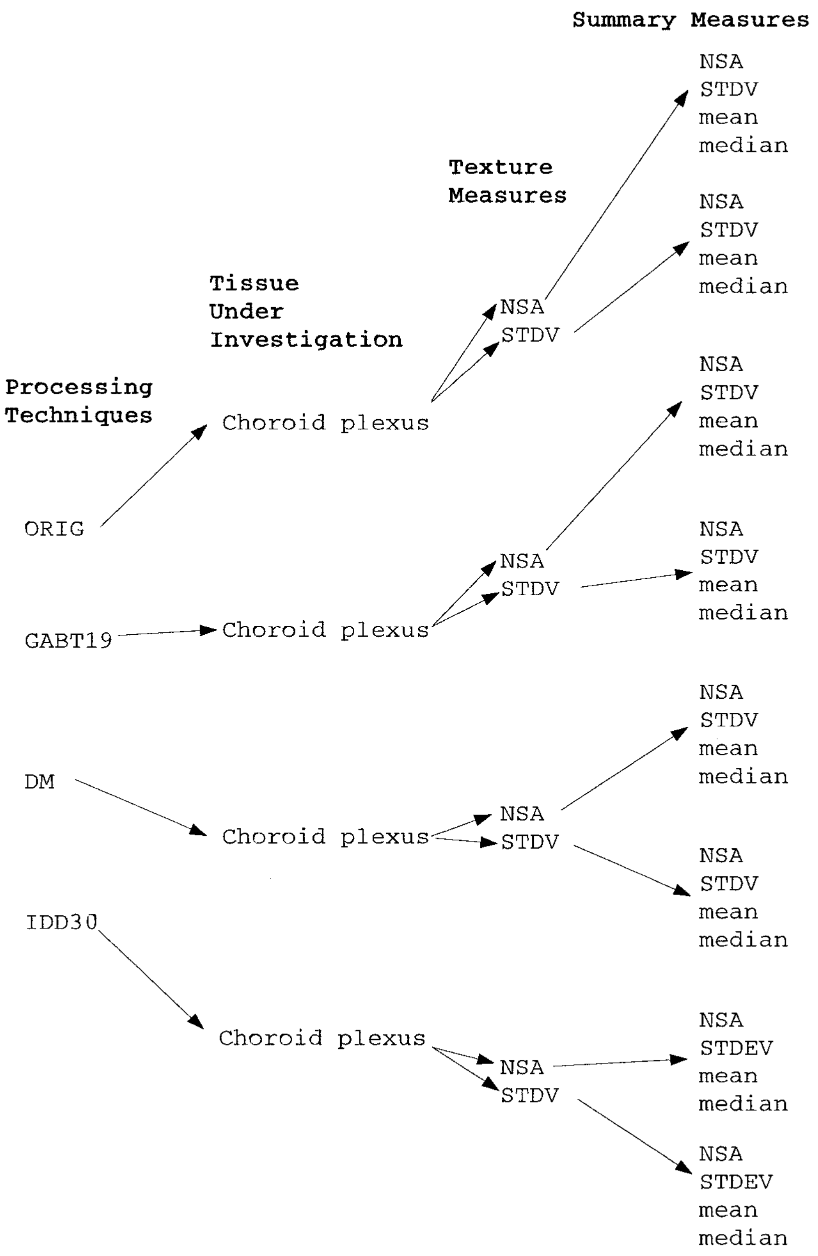

Figure 4.4: Preliminary selection of measures. This diagram shows 32 measures. The addition of the white matter as the tissue under investigation doubles this to 64 measures.

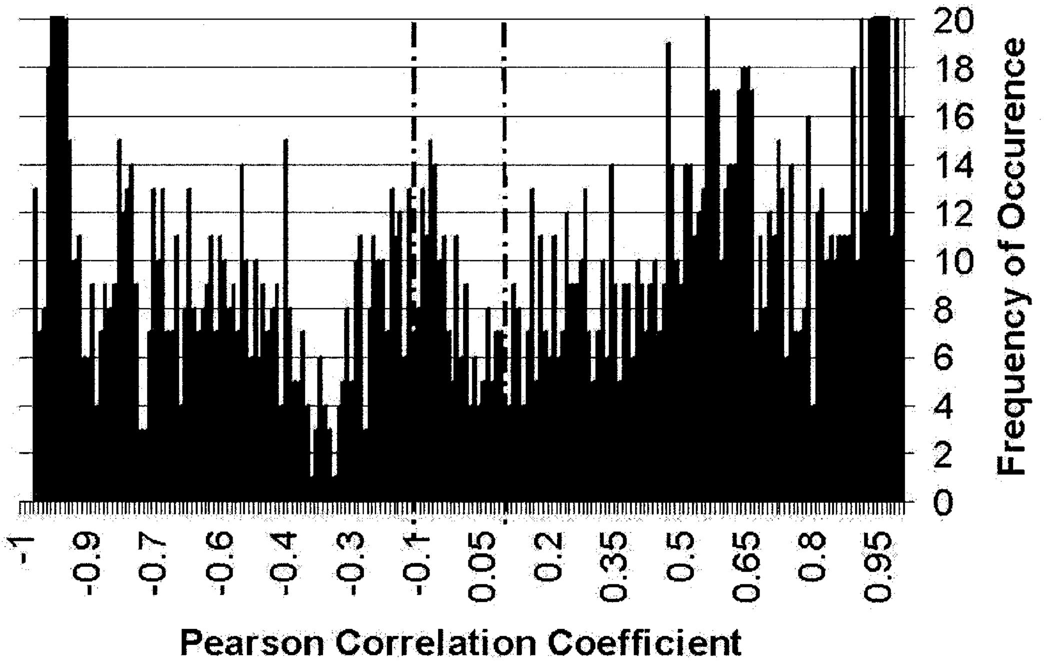

Figure 4.5: Histogram of very weakly correlated variables. The region 0 ± .02 is indicated by the two vertical dashed lines.

image. This provides additional motivation for their removal from further consideration. Skewness and kurtosis were observed to have highly fluctuating values, not related to the outcome of the members of EXP1. Therefore, these measures were considered to be unsuitable. These observations reduced the summary measures from 8 to 4. At this point, the variable set consists of 64 measures per patient, coming from 16 parametric maps described by the four measures NSA, STDV, mean and median. Figure 4.4 illustrates the 64 measures.

Although the processing techniques were chosen to highlight a particular aspect of the texture, it is expected that they are highly correlated. It is desirable to create a subset of the 64 measures that have low correlation between them. This will increase the likelihood that a separate characteristic is being measured by each variable. To assist in the creation of a low-correlation set of variables, the Pearson correlation coefficient was calculated for all pairs of the 64 measures. The pairs of measures with a correlation coefficient $< |.02|$

were identified. This cut-off was chosen as it was desired to capture all measures at or very close to zero correlation. Figure 4.5 contains a histogram of the very weakly correlated variables (0-0.19) with the initial 0.02 threshold shown. All correlation descriptors refer to a strength of correlation scale [141]. This initial search criteria identified 28 pairs of measures (Group A). Only three pairs consisted of 2 measures that came from maps created with the choroid plexus as the tissue under investigation and the white matter as the reference. It was observed in early experiments, on EXP0 and EXP1 in Figure 3.7, that the choroid plexus measures differed according to patient outcome. Therefore, one low-correlation pair that contained two choroid plexus-based measures were chosen as the starting point. The process for forming groups of variables based on a single low-correlation pair of choroid plexus measures was repeated for all three pairs initially identified.

Returning to the entire group of 64 measures, those measures which had a correlation coefficient $< |.39|$ with the pair from Group A were identified as potential candidates for inclusion, as this designates the cut-off between low-correlation and moderate correlation. This identified 10 variables (Group B). The one variable with the lowest correlation to both in the pair, from Group A, was selected. Finally, the correlations of these three measures were compared to the remainder from Group B. Those with correlation coefficients $| < .59|$ with all three were included, as this is the cut-off from moderate to strong correlation. See Figure 4.6 for a flowchart demonstrating this search method. The details of this search are also presented in Appendix K. This method resulted in a group of 5 variables to proceed to the model design and test phase. At the completion of the variable selection process, the use of EXP1 data ceased and EXP2 data was used for the design and validation of the model. The selected variables are listed in table 4.1.

When this statistical- and engineering-knowledge-based procedure was repeated with the other 2 pair of very-weak correlation choroid plexus pairs of variables, no additional measures met the statistical-based search criteria. Therefore, the other 2 pairs were not expanded into larger groups with this method. These two pair were separately used to create RFC with the EXP2 data set, the OOB error rate was nearly 50 % for both pairs. This OOB error estimate indicates that the performance of these pairs was equal to random chance.

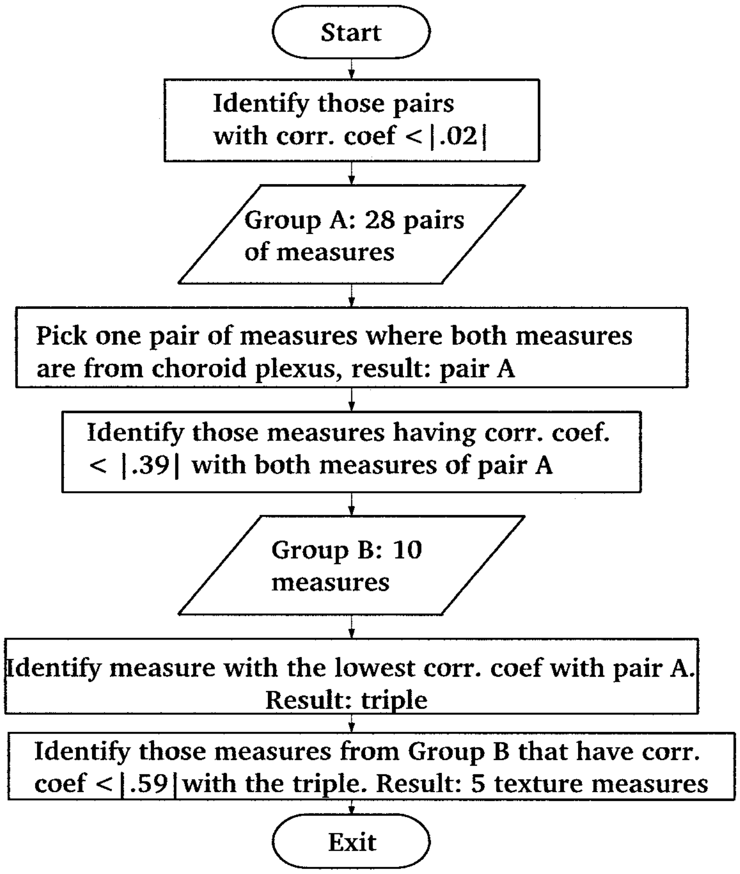
Start
Identify those pairs with corr. coef < |.02|
Group A: 28 pairs of measures
Pick one pair of measures where both measures are from choroid plexus, result: pair A
Identify those measures having corr. coef. < |.39| with both measures of pair A
Group B: 10 measures
Identify measure with the lowest corr. coef with pair A. Result: triple
Identify those measures from Group B that have corr. coef < |.59| with the triple. Result: 5 texture measures
Exit

Figure 4.6: Flowchart of the variable selection method.

4.3 Creation of the Random Forest Classifier

The hypothesis is tested by creating a model/classifier and testing its performance. If the performance is found to be better than chance, then the measures from which the model is created are deemed to contain information for the data set under investigation. The randomForest algorithm was run on the EXP2 data with the 5 previously selected variables based on EXP1.

The randomForest algorithm has several parameters that can be modified. Two parameters were adjusted from their default values in the creation of the classifier. The values of ntree and mtry were modified, while the cut-off vector was left at the default setting. The value of ntree determines the number of decision trees that are contained within the forest. The number of trees in the forest was increased from the default value of 500 to

Table 4.1: The 5 variables chosen based on engineering knowledge and statistical measures.

Processing	Examined tissue	Comparison	Measure	Summary
DM	choroid plexus	difference	NSA	standard deviation
ORIG	choroid plexus	difference	STDV	mean
DM	choroid plexus	difference	STDV	standard deviation
ORIG	white matter	difference	STDV	median
GABT19	choroid plexus	difference	STDV	NSA

ensure the convergence of the randomForest algorithm [131]. With the default value of 500, the OOB error rate varied widely and showed a greater dependence on the random number used by the algorithm than when the value of ntree was increased. The value of ntree= 20000 was chosen to provide a stable OOB error rate.

The value of mtry was investigated for its effect on the creation of the model. Increasing the value of mtry above its default level increases the number of candidate variables from which the algorithm can select during the creation of the nodes. Increasing the number of variables reduces the benefits of injecting randomness, but can be beneficial if the number of variables containing signal is relativeley small compared to the number of variables in the feature set. The randomForest package contains an algorithm, tuneRF, which identifies the optimal value of mtry for the data set used to create and validate the RFC. The algorithm tuneRF selected mtry= 4.

Setting ntree= $20,000$ and mtry= 4 results in an OOB error rate of 27.5 % for the model. The random number generator seed was set manually. The data from EXP2 and the calls to randomForest are within Appendix L. A sample tree from the forest is provided in Figure 4.7 using getTree function within randomForest. The node numbers correspond to the getTree output, provided in L. The texture measures are provided adjacent to the nodes which use them for spiltting in the decision tree.

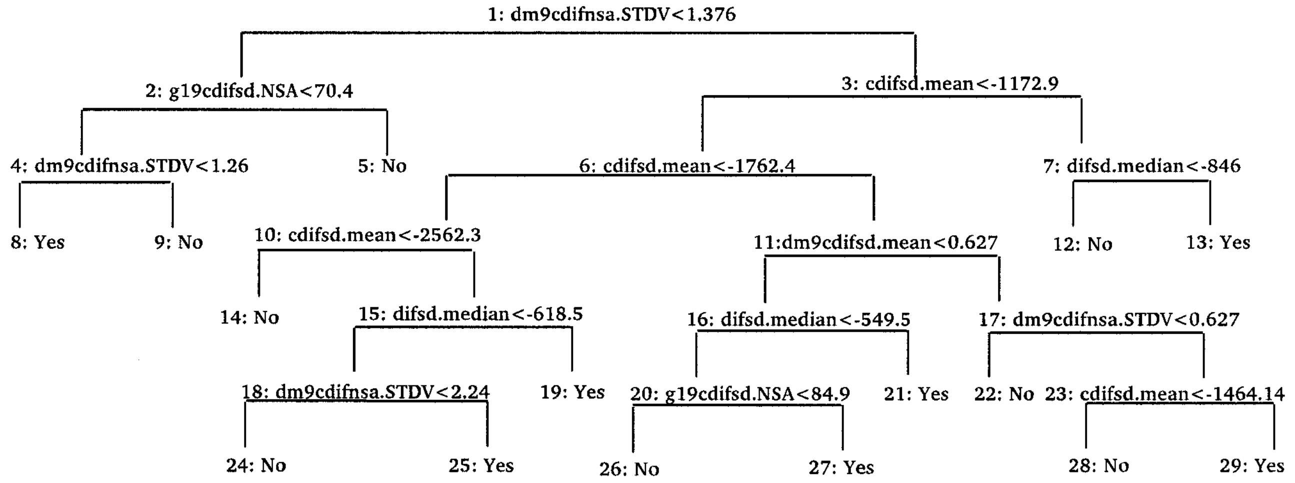

Figure 4.7: Sample tree from the designed RFC. Where the inequality is true, continue with the left branch.

Table 4.2: Radiologists assessment of the same patients.

Evaluator	Accuracy	Sensitivity	Specificity
Radiologist 1	63 %	42 %	81 %
Radiologist 2	64 %	58 %	70 %
Radiologist 3	60 %	45 %	73 %
Majority vote	64 %	48 %	78 %

4.4 Supporting Experiments

Coinciding with this work, a separate study was undertaken by an undergraduate student, Jeanette Evans [142]. The purpose of this study was to determine the ability of radiologists to diagnose the same patients (EXP2) using the earliest clinical US scans. A series of coronal and sagittal cranial US images are obtained at regular intervals in the neonatal period for assessment as part of normal clinical practice for very preterm infants. Blinded to the outcome and to all patient identification, three radiologists assessed each patient's earliest group of cranial US images and provided a diagnosis of normal or abnormal periventricular white matter. The location of the brain evaluated by the radiologists was not limited to the white matter and choroid plexus ROIs used in this research. Rather, the entire series of images and within the images, the entire brain was available for use during their assessment. The radiologists were not expected to provide a diagnosis of the type of abnormality. Resulting from the study are the accuracy, sensitivity and specificity of each radiologist and the same measures from a majority voting scheme, using CP as the positive outcome. The outcomes of this study are provided in Table 4.2. The high specificity values indicate that when radiologists diagnose the patient as having an abnormal brain, they are usually correct. The low sensitivity values indicate that when a patient is diagnosed as normal, there is a good chance that the radiologists have missed an abnormality.

Chapter 5

Results and Discussion

5.1 Performance

5.1.1 Model Performance

The designed RFC has an accuracy of 72.5%, which is 50 of 69 patients correctly classified. The reported accuracy is a result of the OOB error estimate of 27.5 % which is 19 of 69 patients incorrectly classified. The EXP2 data set contains 32 CP positive patients and 37 CP negative patients based on clinical diagnosis. The RFC predicted that 34 patients are negative and 35 are positive for CP. The confusion matrix for the classifier is provided in Table 5.1. The column labelled "Class Error" indicates the portion of those cases incorrectly classified. For example, 11 of 37 are incorrectly classified as having CP and therefore 0.297 or 29.7% of normal patients are identified as having CP. Calculating specificity and sensitivity, from the confusion matrix, provides values of 70.3% and 75% respectively. However, these values are provided for convenience only as the RFC is not a diagnostic test.

Table 5.1: Confusion Matrix for the designed RFC.

	Predicted No	Predicted Yes	Class Error
Diagnosis No	26	11	0.297
Diagnosis Yes	8	24	0.250

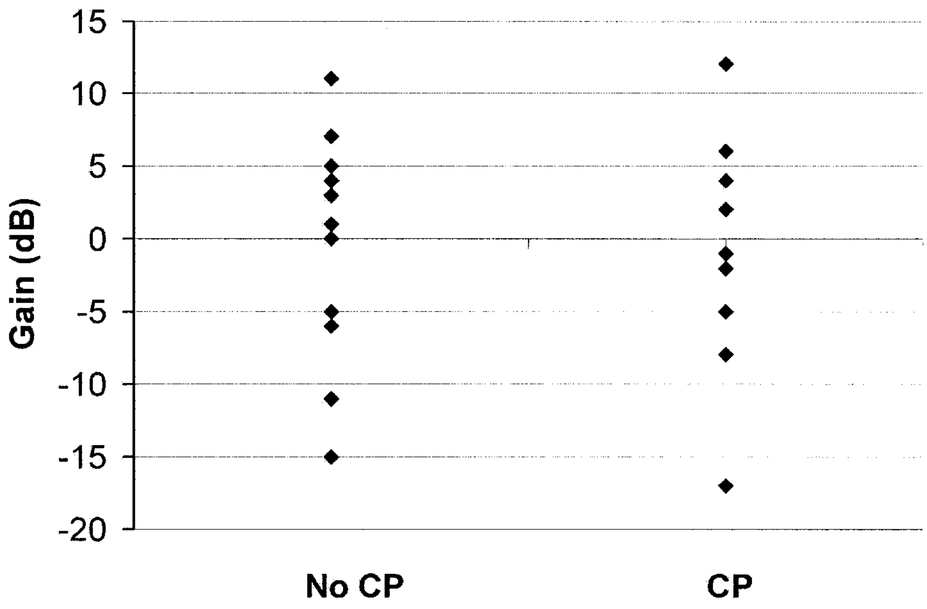

Figure 5.1: The range of gain settings versus the patient outcome.

The images in the EXP2 data set were obtained from two ultrasound machines. The images were obtained with an HDI 5000, at a center frequency of 8.5MHz and with an Acuson 128XP, at 7.0 MHz. The ATL machine was used for 32 cases: 16 CP, 14 no CP. The Acuson machine was used to obtain the images for 39 cases: 18 CP, 21 no CP. The gain setting for the Acuson was printed directly on the image, whereas this was not so for the ATL HDI 5000.

For the known gain values, the range of gain settings over the two outcomes was evaluated and the range is nearly identical (Figure 5.1). This suggests that the gain setting is not associated with the performance of the model. Of the 19 cases misclassified by the RFC, 11 were obtained with the Acuson machine and 8 were obtained with the ATL. This corresponds to a 72% accuracy rate for the images obtained with the Acuson machine, and 75% for the ATL manufactured machine. Figure 5.2 shows a comparison of the gain setting to the RFC performance. The texture measures appear to be associated with the outcome of CP or no CP independent of the machine used to obtain the images. The gain vs. RFC performance data are found in Appendix M.

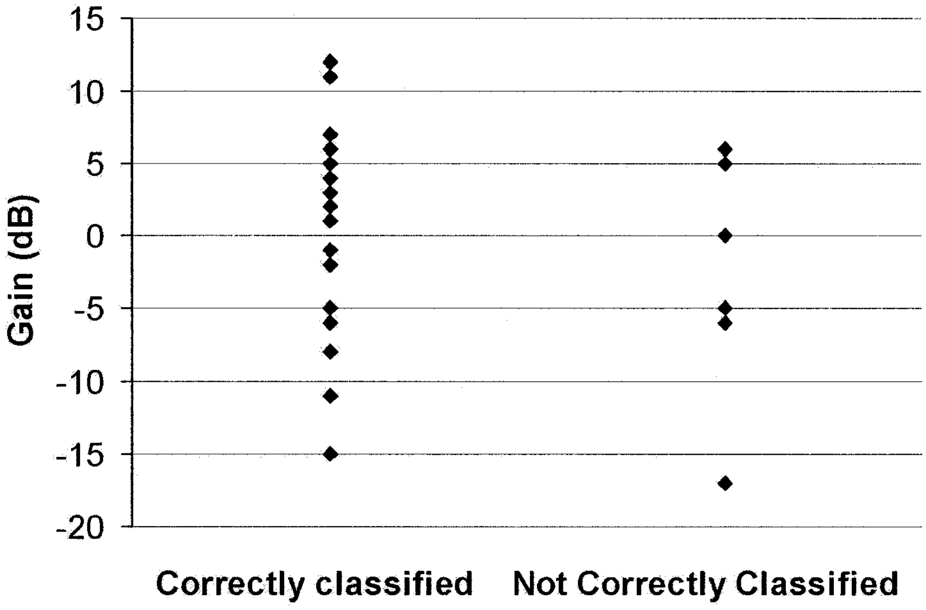

Figure 5.2: The range of gain settings versus the ability of the RFC.

5.2 Discussion

The objective of this work is to test the hypothesis that quantitative early texture measures are associated with patient outcome, not to build a classifier. The study design does not lend itself to comparison with radiologists for several reasons: the incidence of CP in EXP2 is much greater than the general population at 46% vs. 11%, the sample size in the study is small at 69 patients, and the investigation is limited to texture measures without incorporating all available clinical data, and the image analysis experience of the radiologists is not built into the RFC. These factors put the designed RFC at a disadvantage.

In spite of these weaknesses, it is interesting to note the differences in the type of error committed by the RFC as opposed to the radiologists. Unlike the radiologists, the errors made by the RFC are nearly identical in proportion to the number of cases within each class (Table 5.1). The majority voting classification error is calculated from the sensitivity and specificity values and number of cases per class as provided in section 4.4. The information in Table 5.2 shows that radiologists are more often correct when

Table 5.2: Classification errors. The radiologists' errors are derived from the report described in Section 4.4.

Evaluator	Overall Error	Classification error of CP cases	Classification error of No CP cases
Majority vote	.36	.52	.22
RFC	.275	.250	.297

they indicate that the patient has an abnormal brain but frequently incorrect when they indicate that the patient's brain appears normal. Comparing the RFC results, Table 5.1, suggests that the texture measures are obtaining information that is not currently part of the radiologists' assessment. The difference in the classification error of CP cases is particularly noteworthy, Table 5.2.

5.3 Discussion on the Measures

The measures obtained from the parametric texture maps are designed to capture previously unexploited information that has been suggested as important by earlier experiments. It is expected that only a subset of the measures contain diagnostic information. Not all measures were tested for their value, so there is room for future investigation. Only one set of variables was obtained and evaluated. However, some general comments about the final texture measures can be made.

Of the enhancement techniques applied to form the parametric maps, ORIG, DM, and GABT19 survived the variable selection process but IDD30 did not. There are several possible reasons for this. The texture measures were designed from EXP1, which is a small data set and has some technical differences compared to EXP2. Either of these reasons could cause IDD30 to be over-tuned to EXP1 data. The processing techniques and texture measures were selected prior to the removal of the noise. The noise pattern existed in the image different frequencies for the two data sets. The processing technique IDD30 may have been sensitive to the noise differences.

The measures STDV and NSA were chosen to be complementary measures. However, this is not a guarantee that both contain diagnostically relevant information. The first and third measures in Table 4.1 are identical except that one uses STDV and the other uses NSA in the comparison that creates the parametric maps. This suggests that each measure provides an unique feature for classification for this data set.

Four of the five measures come from choroid plexus parametric maps. The fact that the model performed well with these measures suggests that the choroid plexus may provide more cues to the eventual health of white matter than previously suspected. The importance of the choroid plexus measures may be related to the biological mechanism that causes WMD, may be related to risk factor, or may be a function of the way the operators are trained to obtain the US images.

Finally, all discussion about the importance of these particular measures should made with care. Feature selection techniques often select sub-optimal groups and are not unique. That is, there are possibly other subsets of the original measures which may also provide diagnostic information.

5.3.1 Potential Clinical Impact

There are many potential benefits to the addition of computer-based quantitative measures to the use of US images in the diagnosis of white matter damage. As ultrasound is the preferred imaging modality, it is desirable to maximize the information extracted from it. In this research, texture is studied as a new source of diagnostic information within US images. Texture is not well perceived by the human observer. The ability of a human observer to distinguish between two textures is often dependent upon which texture is presented as the foreground and which as the background [143]. Also, some texture combinations are easier to segment than others [144]. Computerized approaches to texture discrimination are not dependent upon location of texture within an image. As well, computers have the ability to obtain mathematical measures of texture subtleties that may be missed by the human visual system. Second, the model presented in this thesis generates quantitative texture measures. Quantitative measures allow for robust comparisons between textures. Thus, in this work, new information is being extracted in a non-subjective manner.

Even with the qualitative information from diagnostic imaging, preterm children do not obtain clinical diagnosis of CP or no CP until after assessment by a team of medical practitioners. The diagnosis may not occur until the child is over 2 years of age. Diagnosis at the Perinatal Follow-up Clinic from 1993 to 2000: 13.8 months on average, 5.3 months standard deviation, and an overall range of 9 to 41 months. (M. Vincer, personal communication.) At present, confident radiological diagnosis of WMD with US does not occur until it has lead to cystic periventricular echolucencies [4]. The model designed in this research, suggests that a system may be created to detect WMD in images not containing cystic periventricular echolucencies with improved accuracy. Improved accuracy of diagnostic imaging and its contribution to the patient's assessment may reduce the age of diagnosis of at-risk children.

This study is the first step in the research to determine if a computer-aided-detection system can be designed to identify those children at high risk. If this work holds for larger studies, there is the potential to improve the diagnosis of CP by identifying children at risk from images obtained at one week of age rather than at 12 to 18 months. Earlier diagnosis of CP with improved accuracy has the potential for a number of clinical benefits:

- Studies have shown that early diagnosis of disease may be beneficial, even when no effective remedy for the disease exists [145]. The benefit may be as simple as reduced parental stress.

- There are continual advancements in the knowledge of prevention and treatment of CP [146]. Improved and earlier diagnosis of CP would allow inclusion of those patients who would benefit the most from new treatments.

- Early diagnosis allows for prompt provision of treatments used in current clinical practice, such as physiotherapy.

- Early and accurate diagnosis may provide insight into the causes and evolution of WMD. This information could lead to modifications in care and treatment of very preterm neonates.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

An association between early quantitative ultrasonic texture measures and the image data used in this research was demonstrated. The input to the system was solely texture measures obtained from a single coronal US scan from each patient. The US image was obtained within days of birth and none contained cystic periventricular leukomalacia. The images were not standardized and as long as the patient met the inclusion criteria and the image quality was sufficient for clinical use, it was considered to be acceptable for this research.

The texture measures are based on measures found to contain information about acoustic media from previous experiments, both in this work and by others. The measures incorporated processing, extraction of texture measures and comparison to a reference, and data reduction. The comparison of the texture measures led to the formation of parametric maps, containing hundreds of comparisons in a format preserving spatial information. The data reduction of these maps was in the form of summary values.

All texture measures were evaluated and the high-dimensional data set of 256 measures was reduced to 5. The resulting RFC had an accuracy of 72.5% as indicated by its OOB error estimate.

Several conclusions result from this research:

- Diagnostic infomation is contained within texture measures of the choroid plexus and white matter. The model was created with measures from both choroid plexus and white matter.

- Not all diagnostically relevant information within coronal US images is used by radiologists. The model created from the texture measures demonstrated different classification errors than the radiologists. This suggests that the texture measures contribute information that differed from the information that the radiologists use in their qualitative assessment.

- Quantitative texture measures can contribute to the diagnosis of white matter damage when cystic periventricular leukomalacia is not present.

- It is possible to obtain quantitative texture measures that have low sensitivity to the machine type and settings used to obtain the images. By using the patient as his or her own control, it is possible to obtain diagnostically relevant texture measures from B-mode images without correcting for operator-dependent settings. Other experiments require either standard operator settings or software standardization of images. In this work, the ability of the model to correctly classify patients appears to be independent of the gain setting and the machine used to obtain the images.

6.2 Future Work

While this research is a good start in the application of UTC to the early detection of cerebral palsy, there is still much more that can be done. The following are identified as topics for future research:

- Further experiments with larger data sets need to be performed. The results must be replicated on other images to verify that the results obtained in this research are suitable for a clinical setting.

- Clinical practice uses the entire brain, whereas this work is restricted to one image and four regions within that image. Improved performance may be obtained if other imaged areas of the brain are assessed and incorporated in the classifier. Future work includes the incorporation of other image measures as well as other clinical data.

- The choroid plexus should be investigated in greater detail to determine if it can provide more diagnostic information. Four of the five measures used in the model are obtained by examining the choroid plexus in detail. The choroid plexus may provide indicators of conditions other than WMD.

- The measures obtained may not be the only, or the best, for this diagnostic task. Investigation of additional measures is required. Further investigation of the measures discarded during the variable selection process is recommended.

- The measures obtained to summarize the maps are combined by averaging. This was done as the left and right side of the patients' brains are not independent. However, if the damage has occurred to one side of the brain only, this method of combining the measures on the two sides will reduce the effect. The measures extracted from the parametric maps and the data reduction methods for each individual patient, needs to be investigated further.

- The selection of the white matter and choroid plexus ROIs needs to be automated to remove the inter- and intra- observer variability caused by the semi-automated approach.

- The sample size of the tissue under investigation, used during the comparison that creates the parametric maps, requires further assessment.

- The effect of gestational age and corrected gestational age on texture should be evaluated for normal white matter and choroid plexus.

- Investigate the earliest age at which texture can identify WMD in CP patients.

- The images used in this research are obtained at very high resolution and then reduced two times. Two of the five texture measures will not allow further resolution reduction due to the window size used in their calculation. Modern digital machines display images at much lower resolution than those used in this work. If the resolution-sensitive measures are proven to be diagnostically critical for an UTC system, the images will need to be obtained from radio frequency data prior to resolution reduction for use on today's display systems. The necessity of the high resolution texture measures needs to be evaluated.

Bibliography

[1] Volpe, J.J. "Neurobiology of Periventricular Leukomalacia in the Premature Infant," *Ped. Res.*, vol. 50, no. 4, pp. 553-62, 2001.

[2] Takashima, S., Armstrong, D., Becker, L.E., and Huber, J. "Cerebral White Matter Lesions in Sudden Infant Death Syndrome," *Pediatrics*, vol. 62, no. 2, pp. 155-159, 1978.

[3] Maalouf, E.F., Duggan, P.J., Counsell, S.J., Rutherford, M.A., Cowan, F., Azzopardi, D., and Edwards, A.D. "Comparison of Findings on Cranial Ultrasound and Magnetic Resonance Imaging in Preterm Infants," *Pediatrics*, vol. 107. no. 4, pp. 719-727, 2001.

[4] Debillon, T., Guyen, S.N., Muet, A., Quere, M.P., Moussaly, F., and Roze, J.C. "Limitations of Ultrasonography for Diagnosing White Matter Damage in Preterm Infants," *Arch Dis Child Fetal Neonatal Ed,* vol. 88, pp. F275-F279.

[5] Holling, E.E. "Characteristics of Cranial Ultrasound White-Matter Echolucencies that Predict Disability: A Review," *Dev. Med. and Child Neur.*, vol. 41, pp. 136-139, 1999.

[6] Thijssen, J.M. "Ultrasonic Speckle Formation, Analysis and Processing Applied to Tissue Characterization," *Patt. Rec. Let.*, vol. 24, pp. 659-675, Feb. 2003.

[7] Schmerr, L., Walbridge, R., and Crouse, B. (2001) NDT resource centre, Iowa State University [online]. Available: www.ndt-ed.org, [2006, 14 June].

[8] Humprey, V.F. and Duck, F.A. "Chapter 1. Ultrasonic Fields: Structure and Prediction," F. Duck, A. Baker, H. Starrett, eds., *Ultrasound in Medicine,* Taylor and Francis, CRC Press, 1998, pp. 3-22.

[9] Bronzino, J.D. "Chapter 65. Ultrasound," in *The Biomedical Engineering Handbook: Second Edition,* Boca Raton: CRC Press LLC, 2000.

[10] Clement. G.T. and Hynynen, K. "Superresolution Ultrasound for Imaging and Microscopy," in *Proc. UFFC Symp.*, 2004, pp. 1832-1835.

[11] Bamber, J.C. (1999) "Image formation and Image Processing in Ultrasound," Course notes, Oxford University, [online]. Available: http://mpss.iop.org/1999/pdf/bamber.pdf, [2006, 14 June].

[12] Seagar, A. and Liley, D. (2002) "Basic Principles of Ultrasound Imaging System Design," Lecture Notes, Biomedical Imaging, Swinburne University of Technology, Australia, [online]. Available: http://marr.bsee.swin.edu.au/lectures/het408/usisd.pdf, [2006, 14 June].

[13] Odom, W. (1999) "Ultrasound Analog Electronics Primer," Analog Dialogue, vol. 33, pp. 41-43, Fig. 3, [online]. Available: http://www.analog.com/library/analogDialogue/archives/33-05/ultrasound/ index.html, [2006, 14 June].

[14] Anderson, M. and Trahey, G. (2000) "A Seminar on k-space Applied to Medical Ultrasound," Duke University, [online]. Available: dukemil.egr.duke.edu/Ultrasound/k-space/bme265.html, [2006,13 June].

[15] Bamber, J.C. "Ultrasonic Properties of Tissues," in *Ultrasound in Medicine*, F. Duck, A. Baker, H. Starrett, eds. Taylor and Francis, CRC Press, 1998, pp. 57-88.

[16] Faran, J.J. "Sound Scattering by Solid Cylinders and Spheres," *J. Acoust. Soc. Am.*, vol. 23, no. 4, pp. 405-418, 1951.

[17] Hay, A.E. and Schaafsma, A.S. "Resonance Scattering in Suspensions," *J. Acoust. Soc. Am.*, vol. 85, no.3, pp. 1124-1138, Mar. 1989.

[18] Wagner, R.F. "Statistics of Speckle in Ultrasound B-scans," *IEEE Trans. on Sonics and Ultrasonics*, vol. 30, no. 3, pp. 156-163, 1983.

[19] Dong, Y., Milne, A.K., and Forster, B.C. "Toward Edge Sharpening: A SAR Speckle Filtering Algorithm," *IEEE Trans. Geosci. Rem. Sens*, 2001, vol.39, no. 4, pp. 851-863.

[20] Wang, S.R. and Sun, Y-N. "Adaptive Diffusion Filter and Absorbing Light Function for Effective Preprocessing in Fetal Ultrasound," in *Proc. First Joint BMES/EMBS Conf. Serving Humanity, Advancing Technology*, 1999.

[21] Abd-Elmoniem, K.Z., Kadah, Y.M., and Youssef, A-B.M. "Real Time Adaptive Ultrasound Speckle Reduction and Coherence Enhancement in Ultrasound Imaging via Nonlinear Anisotropic Diffusion," *IEEE Trans. Biom. Eng*, vol. 49, no. 9, pp. 997-1014, 2002.

[22] Chang, R., Wu, W., Moon, W., and Chen, D. "Improvement in Breast Tumor Discrimination by Support Vector Machines and Speckle-Emphasis Texture Analysis," *Ult. Med. and Biol.*, vol. 29, no. 5, pp. 679-686, 2003.

[23] Ganse, A. (2004) "An Introduction to Beamforming", Applied Physics Laboratory, University of Washington, [online]. Available: http://staff.washington.edu/aganse/beamforming/beamforming.html, [2006, 14 June].

[24] Quistgaard, J.U. "Ultrasonic Image Formation: Implications for the Image Processing Practitioner," *Proc. IEEE Int. Conf. Image Proc.*, 1994, vol. 3, pp. 533 - 537.

[25] "Acuson Sequoia 512 Ultrasound Platform," (2002) Acuson Corporation, A Siemens Company, Equipment brochure, Order No. A91004-M2420-F492-2-4A00, [online]. Available: www.medical.siemens.com/siemens/zh_CN/gg_us_FBAs/files/ brochures/ Acuson/Sequoia_GI_new_2003.pdf, [2005, 1 May].

[26] Children's Hospital at Montefiore, "High-Risk Newborn." [online]. Available: http://www.montekids.org/healthlibrary/peds/hrnewborn/pvl/, [2006, 14 June].

[27] Nelson, K. and Grether, J.K. "Causes of Cerebral Palsy", *Current Opinion in Pediatrics*, vol. 11, no.6, pp. 487, December 1999.

[28] Inder, T.E., Huppi, P.S., Warfield, S., Kikinis, R., Zientara, G.P., Barnes, P.D., Jolesz, F., and Volpe, J.J. " Periventricular White Matter Injury in the Premature Infant is Followed by Reduced Cerebral Cortical Grey Matter Volume at Term," *Ann. Neurol.*, vol. 46, pp. 755-60, 1999.

[29] "The Central Nervous System.", Kumar,ABBAS,Fausto, eds., *Robbins and Cotran, Pathologic Basis of Disease*, Philadephia, Pennsylvania: Elesier Saunders, 2005, pp.1365.

[30] Molson Medical Informatics, Student Projects (2001). "Blood Brain Barrier and Edema Primer." McGill University [online]. Available: http://sprojects.mmi.mcgill.ca/braintumor/section1/subsection2,[2006, 14 June].

[31] Rang, E.M. and Grimm, M.J. "Ultrasonic Assessment of White Matter as a Function of Predominant Axon Orientation," in *Proc. Summer Bioengineering Conf.*, 2003, pp. 133-134.

[32] Lane, D. and Brott, E. (1993) Glossary, Rice Virtual Lab in Statistics, Rice University, Texas [online]. Available: http://davidmlane.com/hyperstat/glossary.html, [2006, 15 June].

[33] Contingency analysis(2006), [online]. Available: http://www.riskglossary.com/.

[34] Haralick, R.M., Shanmugam, K., and Dinstein, I. "Textural Features for Image Classification," *IEEE Trans. Sys. Man Cyb.*, vol. SMC-3, no. 6, pp. 610-621, 1973.

[35] Chan, K.L. "Adaptation of Ultrasound Image Texture Characterization Parameters," in *Proc. of 20th Conf. IEEE Eng. Med. Biol.*, 1998, vol. 20, no. 2, pp. 804-807.

[36] Oosterveld, B.J., Thijssen, J.M., and Verhoef, W.A. "Texture of B-Mode Echograms: 3-D Simulations and Experiments of the Effects of Diffraction and Scatterer Density," *Ultrasonic Imaging*, vol. 7, pp. 142-160, 1985.

[37] Strang, G. and Nguyen, T. "Bases and Frames," in *Wavelets and Filter Banks,* Revised ed., Wellesley, MA: Wellesley-Cambridge Press, 1997, pp. 69-80.

[38] Chen, C., Lu, H.H., and Han, K. "A Textural Approach Based on Gabor Functions for Textural Edge Detection in Ultrasound Images," *Ultrasound Med. Biol.*, vol. 27, no. 4, pp. 515-534, 2001.

[39] Anastassopoulos, V, and Venetsanopoulos, A.N. "The Classification Properties of the Pectrum and its Use for Pattern Identification," *Circ. Syst. Signal Process.*, vol. 10, no. 3, pp. 293-326, 1991.

[40] Weisstein, E.W. "Fractal" From MathWorld–A Wolfram Web Resource [online]. Available: http://mathworld.wolfram.com/Fractal.html,[2006, 15 June].

[41] Bateman, A., and Paterson-Stephens, I. *The DSP Handbook* Essex: Pearson Education Limited, 2002, pp. 252-261.

[42] Burt, P.J. and Adelson, E.H. "The Laplacian Pyramid as Compact Image Code," *IEEE Trans. Comm.*, COM-31(4), pp. 532-540, 1983.

[43] Daubechies, I. "Orthonormal Bases of Compactly supported Wavelets," *Comm. Pure Appl. Math.*, vol. XLI, pp. 909-996, 1988.

[44] Sajda, P., Laine, A., and Zeevi, Y. "Multiresolution and Wavelet Representations for Identifying Signatures of Disease," *Disease Markers*, vol. 18, pp. 339-363, 2002.

[45] Simaeys, B., Philips, W., Lemahieu, I., and Govaert, P. "Quantitative Analysis of the Neonatal Brain by Ultrasound," *Computerized Medical Imaging and Graphics.*, vol. 24, pp. 11-18, 2000.

[46] Shankar, P.M., Reid, J.M., Ortega, H., Piccoli, C.W., and Goldberg, B.B. "Use of Non-Rayleigh Statistics for the Identification of Tumors in Ultrasonic B-scans of the Breast," *IEEE Trans. Med. Imag.*, vol. 12, no. 4, pp. 687-692, 1993.

[47] Ghofrani, S., Jahed-Motlagh, M.R., and Ayatollahi, A. "An Adaptive Speckle Suppression Filter Based on Nakagami Distribution," in *Proceedings of Eurocon, Inter. Conf. Trends in Comm.*, 2001, vol. 1, pp. 84-87.

[48] Rao, N., Mehra, S., and Zhu, H. "Ultrasound Speckle Statistics Variations with Imaging Systems Impulse Response," in *Proc. IEEE Ultra. Symp.*, 1990, vol. 3, pp. 1435-1440.

[49] Wagner, R.F., Insana, M.I., and Smith, S.W. "Fundamental Correlation Lengths of Coherent Speckle in Medical Ultrasound Images," *IEEE Trans. UFFC.*, vol. 35, no. 1, pp. 34-44, 1988.

[50] Thijssen, J.M. and Oosterveld, B.J. "Texture in Tissue Echograms. Speckle or Information? " *J. Ultrasound Med.*, vol. 9, pp. 215-229, 1990.

[51] Dantas, R.G., Costa, E.T., and Leeman, S. "Ultrasound Speckle and Equivalent Scatterers," *Ultrasonics*, vol. 43, pp. 405-420, 2005.

[52] Wagner, R.F., Insana, M.F., and Brown, D.G. "Unified Approach to the Detection and Classification of speckle texture in Diagnostic Ultrasound," *Optical Engineering*, vol. 25, no. 6, pp. 738-742, 1986.

[53] Wagner, R.F., Insana, M.F., and Brown, D.G. "Statistical Properties of Radiofrequency and Envelope-Detected Signals with Applications to Medical Ultrasound," *J. Opt. Soc. Am. A.*, vol. 4, No. 5, May 1987.

[54] Jacobs, E.M.G.P. and Thijssen, J.M. "A Simulation Study of Echographic imaging of diffuse and structurally scattering media," *Ultrasonic Imaging*, vol. 13, pp. 316-333, 1991.

[55] Huber, S., Danes, J., Zuna, I., Teubner, J, Medl, M., and Delorme, S. "Relevance of Sonographic B-Mode Criteria and Computer-Aided Ultrasonic Tissue Characterization in Differential Diagnosis of Solid Breast Masses," *Ultrasound Med. Biol.*, vol. 26, no. 8, pp. 1243-1252, 2000.

[56] Kutay, M.A., Petropulu, A.P., Reid, J.M., and Piccoli, K. "Malignant verses Benign Tumor Classification Based on Ultrasonic B-scan Images of the Breast," *IEEE Ultra. Symp.*, 2000, pp. 1383-1386.

[57] Kutay, M.A. and Piccoli, K. "Breast Tissue Charcterization Based on Modeling of Ultrasonic Echos Using the Power Law Shot Noise Model," *Patt. Rec. Lett.*, 2000, pp. 1383-1386.

[58] Alacam, B., Yazici, B., and Bilgutay, N. "Breast Cancer Detection Based on Ultrasound B-scan Texture Analysis and Patient Age Information," in *Proc. IEEE 2003 Bioengineering Conference*, 2003, pp. 98-99.

[59] Chang, R., Chen, C., and Ho, M. "Breast Ultrasound Classification Using Fractal Analysis," in *Proc. 4th IEEE Symp. Bioinfo. Bioeng., 2004*, pp. 100-107, 2004.

[60] Medipattern Corporation(2005). Medipattern, [online]. Available: www.medipattern.com, [2006, 14 June].

[61] Kadah, Y.M., Faraq, A.A., Zurada, J.M., Badawi,A.M., and Youssef, A.M. "Classification Algorithms for Quantitative Tissue Characterization of Diffuse Liver Disease from Ultrasound Images," *IEEE Trans of Med. Im.*, vol. 5, no. 4, pp. 466-478, 1996.

[62] Mojsilovic, A., Popovic, M., Markovic, S., and Krstic, M. "Characterization of Visually Similar Diffuse Diseases from B-Scan Liver Images Using Nonseparable Wavelet Transform," *IEEE tran. Med. Im.*, vol. 17, no. 4, pp. 541-549, 1998.

[63] Gangeh, M.J., Hanmandlu, M, and Bister, M. "A Fuzzy-Based Texture Analysis for Tissue Characterization of Diffused Liver Disease on B-scan Images," in *Proc. 38th Ann. Rocky Mountain Bioeng. Sym.*, 2002, pp. 369-374.

[64] Akiyama, I., Saito, T., Nakamura, M., Taniguchi, N., and Itoh, K. "Tissue Characterization by Using Fractal Dimension of B-Scan Image," in *Proc. IEEE Ultra. Symp.*,vol. 3, 4-7 Dec 1990, pp. 1353-1355.

[65] Lee, W., Chen, Y., Hsieh, K. "Robust Calculation of Fractal Dimension of Images and its Applications to Classification of Ultrasonic Liver Images and Texture Images," in *IEEE Int. Symp. Cir. Sys.*, 2002, vol. 2, pp. II-656 - II-659.

[66] Yoshida, H., Casalino, D.D., Keserci, B., Coskun, A., Ozturk, O., and Savranlar, A. "Wavelet-packet-based Texture Analysis for Differentiation between Benign and

Malignant Liver Tumours in Ultrasound Images," *Physics in Medicine and Biology*, vol. 48, pp. 3735-3753, 2003.

[67] Smutek, D., Sara, R., Sucharda, P., Tjahjadi, T., and Svec, M. "Image Texture Analysis of Sonograms in Chronic Inflamations of the Thyroid Gland," *Ultrasound Med. Biol.*, vol. 29, no. 11, pp. 1531-1543, 2003.

[68] Maeda, K., Utsu, M., and Kihaile, P.E. "Quantification of Sonographic Echogenicity with Grey-Level Histogram Width: A Clinical Tissue Characterization." *Ultrasound Med. Biol.*, vol. 24, no. 2, pp. 225-234, 1998.

[69] Christodoulou, C.I., Pattichis,, C.S., Pantziaris, M., Nicolaides, A. "Texture-Based Classification of Atherosclerotic Carotid Plaques," *IEEE Trans. Med. Imag.*, vol. 22, no. 7, pp. 902-912, 2003.

[70] Barr, L.L., McCullough, P.J., Ball, W.S., Krasner, B.H., Garra, B.S., and Deddens, J.A. " Quantitative Sonographic Feature Analysis of Clinical Infant Hypoxia: A Pilot Study," *Am. J. of Neuroradiology*, vol. 17, pp. 1025-1031, June 1996.

[71] Mullaart, R.A., Thijssen, J.M., Rotteveel, J.J., Valckx, F.M., and Van Geemen, A.J. "Quantitative Ultrasonography of the Periventricular White and Grey Matter of the Developing Brain," *Ultrasound Med. Biol.*, vol. 25, no. 4, pp. 527-530, 1999.

[72] Valckx, F.M.J., Thijssen, J.M., vanGeemen, A.J., Rotteveel, J.J., and Mullaart, R. "Calibrated Parametric Medical Ultrasound Imaging," *Ultrasonic Imaging*, vol. 22, pp. 57-72, 2000.

[73] Stippel,G., Duskunovic, I., Philips, W., Zecic, A., and Lemahieu, I. "A New Filtering Method for Ultrasound Images Incorporating Prior Statistics Concerning Medical Features," in *Proc. IEEE ICIP Conf.*, 2001, pp. 821-824.

[74] Stippel, G., Vansteenkiste, E., Philips, W., and Lemahieu, I. "A Medical Feature Enhancing Speckle Suppression Technique Based on a Noise Model for the Speckle Noise as Respresented in the B-scan Image," in *Proc. ACIVS,* 2002, S00-1 - S00-5.

[75] Stippel, G., Philips, W., Lemahieu, I., and Govaert, P. "A Medical Feature Enhancing New Denoising Technique for Ultrasound Images," in *Proc. DICTA,* 2002, pp. 174-178.

[76] Stippel, G., Philips, W., and Lemahieu, I. " A New Denoising Technique for Ultrasound Images using Morphological Properties of Speckle Combined with Tissue Classifying Parameters," in *Proc. of SPIE/Medical Imaging Ultra. Imaging and Sig. Proc.,* February 2002, pp. 324-333.

[77] Stippel, G., W. Philips, W., and Govaert, P. "A Tissue-Specific Adaptive Texture Filter for Medical Ultrasound Images," *Ultrasound Med. Biol.,* vol. 31, no. 9, pp. 1211-1223, 2005.

[78] Vansteenkiste, E., Huysmans, B., Philips, W. "An Evaluation of Brain Tissue Classification in Non-Compensated Ultrasound Images," in *Proc. EUSIPCO2005,* 2005, [online]. Available:http://www.ee.bilkent.edu.tr/ signal/defevent/papers/cr1387.pdf

[79] Dong, Y., Milne, A.K., and Forster B.C. "A Review of Speckle Filters: Texture Restoration and Preservation," in *Proc. IEEE Int. Geosci. Rem. Sens. Sym.,* 2000, vol. 2, pp. 633-635.

[80] Duskunovic, I., Pizurica, A., Stippel, G., Philips, W., and Lemahieu, I. "Wavelet Based Denoising Techniques for Ultrasound Images," in *Proc. 22nd EMBS Inter. Conf.,* 2000, pp. 2662-2556.

[81] Fulin, S., Jiang, W., Hongtao, G., Yong, Z. "A Speckle Reduction Algorithm for SAR Images," in *Proc. ICII,* 2001, vol. 1 , pp. 357 -362.

[82] Kotropoulos, C., Magnisalis, I., Pitas, I., and Strintzis, M.G. "Nonlinear Ultrasonic Image Processing Based on Signal-Adaptive Filters and Self-Organizing Neural Networks," *IEEE Trans. Im. Proc.,* vol. 3, no. 1, pp. 65-77, 1994.

[83] Evans, AN., and Nixon, M.S. "Mode Filtering to Reduce Ultrasound Speckle for Feature Extraction," in *IEE Proc. Vis. Image Sig. Proc.,* 1995, vol. 142, no. 2, pp. 87-94.

[84] Kofidis, E., Theodoridis, S., Kotropoulos, C., and Pitas, I. "Nonlinear Adaptive Filters for Speckle Suppression in Ultrasonic Images," *Signal Processing,* vol. 52, pp. 357-372, 1996.

[85] Dutt, V., and Greenleaf, J.F. "Adaptive Speckle Reduction Filter for Log-Compressed B-scan Images," *IEEE Trans. Med. Imag.*, vol. 15, no. 6, pp. 802-813, 1996.

[86] Suvichakorn, A. and Chinrunggrueng, C. "Speckle Noise Reduction Based on Least Squares Approximation," in *Proc. IEEE Asia-Pacific Conf. Cir. and Sys.*, 2000, pp. 430-433.

[87] Tsubai, M., Takemura, A., and Ito, M. "Morphological Operations for Ultrasound Images by Locally Variable Structuring Elements and their Analysis of Effective Parameters," in *Proc. 22nd Ann. EMBS Inter. Conf.,* 2000, pp. 2526-2528.

[88] Uslu, K.H., Bilgutay, N.M., Murthy, R., and Kaya, K. "Medical Image Enhancement Using Split Spectrum Processing," in *Proc. IEEE Ultra. Symp.*, 1993, pp. 993-997.

[89] Breiman, L. "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.

[90] Pal, M. "Random Forests for Land Cover Classification," in *Proc. Goesci. Rem. Sens. Sym.,* 2003, vol 6, pp. 3510-3512.

[91] Pal, M. "Random Forest Classifier for Remote Sensing Classification," *Int. J. Rem. Sens.*, vol. 26, no. 1, pp. 217-222, 2005.

[92] Ham, J., Chen, Y., Crawford, M.M., and Ghosh, J. "Investigation of the Random Forest Framework for the Classification of Hyperspectral Data," *IEEE Trans. Geosci. Rem.Sens.*, vol. 43, no. 3, pp. 492-501, 2005.

[93] Short Sr., N.M. (2006), Remote Sensing Tutorial, [online]. Available: http://rst.gsfc.nasa.gov/Front/tofc.html.

[94] Joelsson, S.R., Benediktson, J.A., and Sveinsson, J.R. "Random Forest Classifiers for Hyperspectral Data," in *Proc. IEEE Geosci. Rem. Sens. Sym.,* 2005, vol. 1, pp. 160-163.

[95] Gislason, P.O., Benediktsson, J.A., and Sveinsson, J.R. "Random Forests for Land Cover Classification," *Patt. Rec. Lett.*, vol. 27, pp. 294-300, 2006.

[96] Wu, Y. and Zhang, A. "An Adaptive Classification Method for Multimedia Retrieval," in *Proc. Int. Conf. Multimedia Expo,* 2003, vol. 1, pp. 757-760.

[97] Wu, Y. and Zhang, A. "PatternQuest: Learning Patterns of Interest Using Relevance Feedback in Multimedia Information Retieval," in *Proc. IEEE Int. Conf. Multimedia Expo,* 2004, vol. 1, pp. 261-264.

[98] Calleja, J. de la , and Fuentes, O. "Automated Classification of Galaxy Images," *Lecture Notes in Artificial Intelligence*, vol. 3215, Springer, pp 411-418, 2004.

[99] Luo, T., Kramer, K. Goldgof, D.B., Hall, L.O., Samson, S., Remsen, A., and Hopkins, T. "Recognizing Plankton Images from the Shadow Image Particle Profiling Evaluation Recorder," *IEEE Trans. Systems, Man and Cyb., Part B*, vol. 34, no. 4, pp. 1753-1762, 2004.

[100] Geng, W., Cosman, P., Berry, C.C., Zhaoyang, F., Schafer, W.R. "Automatic Tracking, Feature Extraction and Classification of C. Elegans Phenotypes," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 10, pp. 1811-1820, 2004.

[101] Thieme, G. "Chapter 2. Clinical Relevance of Scattering," in *Ultrasonic Scattering in Biological Tissues*, K.Shung and G. A. Thieme, eds., Boca Raton: CRC Press, 1993, pp. 22-34.

[102] Siegel, M.J. "Brain," in *Pediatric Sonography,* 2nd ed., M.J. Siegel, ed. New York: Raven Press, Ltd., 1995, pp. 29-101.

[103] Kliegman, R.M. "The Fetus and the Neonatal Infant," in *Nelson Textbook of Pediatrics,* 15th ed., R. E. Behrman, R. M. Kliegman, A.M. Arvin, eds. Philadelphia: W.B. Saunders Co., 1996, pp 435.

[104] Huisman, H.J. and Thijssen, J.M. "An In Vivo Ultrasonic Model of Liver Parenchyma."*IEEE Trans. UFFC*, vol.45, no.3, pp. 739 - 750, 1998.

[105] Goss, S.A., Johnston, R.L., and Dunn, F. "Comprehensive Compilation of Empirical Ultrasonic Properties of Mammalian Tissues," *J. Acoustic Soc. Am.*, vol. 64, no. 2, pp. 423-457, Aug. 1978.

[106] Hope, T., Gregson, P., Linney, N., and Schmidt, M. "Texture-based Tissue Characterization: A Novel Predictor for Brain Injury?" in *Proc. 2nd IASTED Int. Conf. Biomed. Eng.*, 2004, ACTA Press, pp. 135-140.

[107] Hope, T., Gregson, P., Linney, N., and Schmidt, M. " Ultrasonic Tissue Characterization as a Predictor of White Matter Damage: Results of a Preliminary Study," in *Proc. IEEE 50th Ann. Ultra. Symp.,* 2004, vol.3, pp. 2157-2160.

[108] Hope, T., Linney, N. and Gregson, P. "Using the Local Mode for Edge Detection in Ultrasound," in *Proc. IEEE CCECE,* 2005, pp. 348-351.

[109] Murthy, R. and Bilgutay, N.M. "Scaling Techniques for Medical Image Enhancement," in *Proc. IEEE Ultra. Symp.,* pp. 1389-1392, 1995.

[110] Breiman, L. "Statistical Modeling: The Two Cultures.", *Statistical Science*, vol. 16, no. 3, pp. 199-215, 2001.

[111] Quanjer, P. (2006) "Some Statistical Concepts," [online]. Available: www.spirxpert.com/statistical9.htm, [2006, 28 June].

[112] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. "Estimating Accuracy," in *Classification and Regression Trees,* Wadsworth, 1984, pp. 1-13.

[113] Efron, B. and R.J. Tibshirani, R.J. "The Bootstrap Estimate of Standard Error," in *An Introduction to the Bootstrap,* Boca Raton: Fl: CRC Press, 1993, pp. 45-59.

[114] Duda, R.O., Hart, P.E., and Stork, D.G. "Algorithm Independent Machine Learning," in *Pattern Classification,* Second edition, John Wiley and Sons, 2001, pp.465-471.

[115] Johnson, D.E. "Discriminant Analysis," in *Applied Multivariate Methods for Data Analysts,* Brooks/Cole Publishing Company, 1998, pp. 217-234.

[116] Brannick, M. (2000) "Logistic Regression Course Notes," University of South Florida, [online]. Available:
http://luna.cas.usf.edu/ mbrannic/files/regression/Logistic.html, [2006, 14 June].

[117] Bellman, R. "Chapter 15. Adaptive Control Processes," in *Adaptive Control Processes: A guided tour,* Princeton University Press, Princeton New Jersey. 1961, pp. 194-203.

[118] Stremler, F.G. "Information and Digital Transmission," in *Introduction to communications systems.* Addison-Wesley Publishing Company, Philippines, 1982, pp. 499-570.

[119] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. "Introduction to Tree Classification," in *Classification and Regression Trees,* Wadsworth, 1984, pp.18-36.

[120] Breiman, L. "Arcing Classifiers," *The Annals of Statistics*, vol. 26, no. 3, pp. 801-824, 1998.

[121] Efron, B., and Tibshirani, R. (1995), "Cross-validation and the Bootstrap: Estimating the Error Rate of a Prediction Rule," Technical Report (TR-477), Dept. of Statistics, Stanford University, [online]. Available: citeseer.ist.psu.edu/47726.html, [2006, 28 June].

[122] Dietterich, T. "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization" *Machine Learning*, vol. 40, pp. 139-157,2000.

[123] Freund, Y. and Schapire, R.E. "Experiments with a New Boosting Algorithm," in *Proc of 13th Int. Conf. on Machine Learning*, 1996, pp. 148-156.

[124] Breiman, L. and Cutler, A. (2004) "On-line Random Forest Manual," [online]. Available: http://www.stat.berkeley.edu/users/breiman/RandomForests, [2006, 15 June]

[125] Izmirlian, G. "Application of the Random Forest Classification Algorithm to a SELDI-TOF Proteomics Study in a Setting of a Cancer Prevention Trial," *Ann. N.Y. Acad. Sci.*, vol. 1020, pp. 154-174.

[126] Meyer, D., Leisch, F., and Hornik, K. "The Support Vector Machine under Test," *Neurocomputing*, vol. 55, pp. 169-186, 2003.

[127] R Development Core Team,(2006). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, [online]. Available: http://www.R-project.org, [2006, 14 June].

[128] Liaw, A. and Weiner, M. (2002). "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18-22, [online]. Available: http://CRAN.R-project.org/doc/Rnews/Rnews_2002-3.pdf, [2006, 15 June].

[129] Li, H., Zhang, K., and Jiang, T. "Robust and Accurate Cancer Classification with Gene Expression Profiling." in *Proc. IEEE Comp. Syst. Bioinfo. Conf.*, 2005, pp. 310-321.

[130] Breiman, L. (1996), "Out-of-Bag Estimation", Technical Report, Statistics Department, University of California, [online]. Available:
http://citeseer.ist.psu.edu/224001.html, [2006, 15 June].

[131] Svetnik, V., Liaw, A., Tong, C., and Wang, T. "Application of Breiman's Random Forest to Modeling Structure-Activity Relationships of Pharmaceutical Molecules," *Lecture Notes in Computer Science*, vol. 3077, pp. 334-343, 2004.

[132] Duda, R.O., Hart, P.E. and Stork, D.G. "Maximum-Likelihood and Bayesian Parameter Estimation," in *Pattern Classification,* Second edition, John Wiley and Sons, 2001, pp. 117.

[133] Cheriyadat, A. "Limitations of Principal Component Analysis for Dimensionality-Reduction for Classification of Hyperspectral Data," *Master's Thesis*, Mississippi State University, 2003.

[134] Yan, J., Zhang, B., Liu, N., Yan, S., Cheng, Q., Fan, W., Yang, Q., Xi, W., and Chen, Z. "Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing." *IEEE Trans. Know. Data Eng.*, vol. 18, no. 3, pp. 320-333, 2006.

[135] Liu, H., Dougherty, E.R., Dy, J.G., Torkkola, K., Tuv, E., Peng, H., Ding, C. Long, F., Berens, M., Parsons, L., Zhao, Z., Yu, L., and Forman, G. "Evolving Feature Selection," *IEEE Intelligent Systems*, vol. 20, no. 6, pp. 64-76, 2005.

[136] Jiang, H., Deng, Y., Chen, H., Tao, L., Sha, Q., Chen, J., Tsai C., and Zhang, S. (2004) "Joint Analysis of Two Microarray Gene-Expression Data Sets to Select Lung Adenocarcinoma Marker Genes," *BMC Bioinformatics*, vol. 5, article 81, [online]. Available: http://bmc.ub.uni-potsdam.de/1471-2105-6-128/1471-2105-6-128.pdf, [2006, 28 June]

[137] Daz-Uriarte, R., Alvarez de Andres, S. (2006) "Gene Selection and Classification of Microarray Data using Random Forest," *BMC Bioinformatics*, vol. 7, 2006, [online]. Available: http://www.biomedcentral.com/1471-2105/7/3, [2006, 15 June].

[138] Yang, Y. and Pedersen, J.O. "A Comparative Study on Feature Selection in Text Categorization," in *Proc. of the 14th Int. Conf. Mac. Learn.*, 1997, pp. 412-420.

[139] Liu, H., Li, J., and Wong, L. "A comparative Study on Feature Selection and Classification Methods using Gene Expression Profiles and Proteomic Patterns," *Genome Informatics*, vol. 13, pp. 51-60, 2002.

[140] Derksen, S. and Keselman, H.J. "Backwards, Forward and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables," *Brit. J. Math. Stat. Psych.*, vol. 45, pp. 265-282, 1992.

[141] Swinscow, T.D.V. (1997) "Statistics at Square One. Correlation and regression. ", BMJ, [online]. Available: http://bmj.bmjjournals.com/statsbk/11.shtml, [2006, 15 June].

[142] Evans, J.W. "Prediction of White Matter Damage in Premature Infants based on Imaging Features of the Periventricular White Matter on the First Cranial Ultrasound Examination," NESC 3000, Department of Pshychology, Dalhousie University, April 16, 2004.

[143] Gurnsey,R. and Browse, R.A. “Aspects of Visual Texture Discrimination,” in *Computational Processes in Human Vision: An Interdisciplinary Perspective,* Z. Pylyshyn, ed. Norwood, New Jersey: Ablex Publishing Company, 1988, pp. 27-43.

[144] Landy, M.S. “Visual Perception of Texture” in *The Visual Neurosciences,* L.M. Chalupa, J.S. Werner, eds. Cambridge, MA: MIT Press, 2004, pp. 1106-1118.

[145] Merelle, M.E., Huisman, J., Alderden-van der Vecht, A., Taat, F., Bezemer, D., Griffioen, R., Brinkhorst, G., and Dankert-Roelse, J.E. “Early Versus Late Diagnosis: Pychological Imapact on Parents of Children with Cystic Fibrosis.” *Pediatrics,* vol. 111, no. 2, pp. 346-350, 2003.

[146] Peterson, M.C., and Palmer, F.B. “Advances in Prevention and Treatment of Cerebral Palsy.” *MRDD Research Reviews.*, vol. 7, pp. 30-37, 2001.

Appendix A

Algorithms

```
/*          ultgrid   draw lines on ultrasound images, outputs angle image          */

/*      prompt:             source image, result image, min angle, increment angle,
        binary image, output angle image, output distance image

            format:             %d %d %d %d %d %d %d
            helpfile:
            menu:
            prototype:  int     ultgrid(int src, int rslt,int sang,int iang, int bi, int opang,
                                        int distim) ;
                */

#include "cvcmd.h"
#include <stdlib.h>
#include <stdio.h>

/* use the drawline function from CVLAB, use stats command to mean value as the
threshold of binary image, after used binary command then can use this command,
finally use combine command
curve=1 : upper curve;
curve=2 : bottom curve;
curve=3 : side edge  */


float pnt(int bi,int a, int b, int arr[]);

void  _ultgrid (void) {

        int     src, rslt,sang, iang,bi, opang, distim;

        src = *((int *) arglist[0]);
        rslt = *((int *) arglist[1]);
        sang = *((int *) arglist[2]);  /* starting angle */
        iang = *((int *) arglist[3]);  /* angle increment */
        bi = *((int *) arglist[4]);        /* input mask image */
        opang = *((int *) arglist[5]);  /* output image -angles of insonation */
        distim = *((int *) arglist[6]); /* output image -distance from point source */
        ultgrid(src, rslt,sang,iang,bi, opang, distim);
        }


int     ultgrid(int src, int rslt,int sang,int iang,int bi, int opang, int distim){

  int a, b,type,xur,yur,xll,yll,i,j,t,dx,dy,x,y,Q1,center,tt,cx,cy, c;
  float rad, m,tm;
  int arr[2], t1;
  rad= 3.1415926/180.0;
  a=0;
  b=0;
  getsize (src, &a, &b, &type);

  pnt(bi,a,b,arr);
  cx=arr[0];
  cy=arr[1];
  printf("center %d %d\n",cx,cy);

  form(rslt,0,a,b,1);
  t1=gettempimage();
  angles(bi,arr[0], arr[1], opang, 10, distim);
  stats(opang,0);
  c=max(sang,120);
  for(t=c; t<imtab[opang].maxpix/10; t=t+iang){
        for(i=0; i<a; i++){
              for(j=0; j<b; j++){
                    if (getpix(opang, i,j)==t*10) putpix(-50, rslt,i,j);
                    if ((getpix(opang, i,j)+1==t*10)&&(j>2*b/3)) putpix(-50, rslt,i,j);
                    if ((getpix(opang, i,j)+2==t*10)&&(j>3*b/4)) putpix(-50, rslt,i,j);
                    if ((getpix(opang, i,j)+1==t*10)&&(i>2*b/5)&&(i<3*b/5))
                                      putpix(-50, rslt,i,j);
                    if ((getpix(opang, i,j)+2==t*10)&&(i>2*b/5)&&(i<3*b/5))
                                      putpix(-50, rslt,i,j);
                    } /* end for j */
              } /* end for i */
        } /* end for t */
 add(src,rslt,t1);
 display(t1,1,"s");




 return (TRUE);
}

float pnt(int bi, int a, int b, int arr[])
{
   int      i,j,k=0,m,x3,y3,x1,y1,x2,y2,x4,y4, t1,t2,t3, t4, t5,t6,t7,t8;
   int      c,d,b1,b2;
   int      curve=2,n=0,kk=0,count=0,stop=0;
   float    sq1,sq2,sq3,da,dd,D,E,de,center,dummy;
   double   m2, m1;

      x1=0;
      x2=0;
      x3=0;
      y1=0;
      y2=0;
      y3=0;
      t1=gettempimage();
      t2=gettempimage();
      t3=gettempimage();
      t4=gettempimage();
      t5=gettempimage();
      t6=gettempimage();
      t7=gettempimage();
      t8=gettempimage();

/* first choice is to calculate the insonation point from the bottom curve*/
/* second choice is the top curve, third is the sides extended */


      setroi(0,a,0,(int) (b/5),2);
      crop(bi,t1,2);
      sobel(t1,t4,t3);
      thin(t4,t2);
      stats(t2,0);
      c=0;


      while((curve==2)&&(c==0)){
        printf("while 2 \n");
        for(i=a/10;i<5*a/6;i++){
             count=0;
             for(j=10; j<b/5; j++){
                   if (getpix(t2,i,j)==1)     count++;

              }
             if (count>1) curve=1;
          }
          c=1; /* checked all the way across the image */
      } /* end while */


     sobel(bi,t1,t2);
     if (curve == 2){     /* bottom curve */
            crop(t1,t2,2);
        for (j=0; j<b/5; j++){
                if (getpix(t2,a/6,j)==1){
                        x1=a/6;
                        y1=j;
                        }
                if(getpix(t2, 5*a/6, j)==1){
                        x2=5*a/6;
                        y2=j;
                        }
                if(getpix(t2, 4*a/5, j)==1){
                        x3=4*a/5;
                        y3=j;
                        }
                } /* end for j */
        } /* end if curve==2 */

if ((abs(y1-y2)<3)||(abs(y2-y3)<3)||(abs(y3-y1)<3)){
        /* the bottom is a straight line */
        y1=0;
        y2=0;
        y3=0;
        curve=1;
}  /* end if checking if bottom is straight line */

 setroi(0.3*a,0.6*a, 4*b/5, b,1);
 crop(bi,t1,1);
 sobel(t1,t2,t3);
 c=0;


if (curve == 1){      /* top curve */
        form(t3,0,a,b,1);
        mount(t2,t3,0.3*a,4*b/5);
        for (j=4*b/5; j<b; j++){
                if (getpix(t3,a/2,j)==1){
                        x1=a/2;
                        y1=j;
                        }
                if(getpix(t3, 0.48*a, j)==1){
                        x2=0.48*a;
                        y2=j;
                        }
                if(getpix(t3, 0.52*a, j)==1){
                        x3=0.52*a;
                        y3=j;
                        }
        } /* end for j */


        if ((abs(y1-y2)<3)||(abs(y2-y3)<3)||(abs(y3-y1)<3)){
            /* the top is a straight line */
            y1=0;
            y2=0;
            y3=0;
            curve=3;
            }
        if (y3<y2) curve=3;
    } /* end if curve==1 */


printf("curve %d \n", curve); /* identify curve used to calc. insonation pt.*/

sobel(bi,t1,t2);
if (curve==3){
    form(t3,0,a,b*2,1);
    setroi(0.10*a,0.3*a,0.6*b, 0.97*b, 3);
    setroi(0.5*a, 0.95*a, 0.6*b, 0.97*b, 4);
    crop(t1,t4,3); /* the sobel image left side */
    mount(t4,t3,0.1*a,0.6*b);
    display(t3,1,"s");
    hough(t3,t5,t6);
    form(t2,0,a,b*2,1);
    crop(t1,t4,4); /* the sobel image right side */
    mount(t4,t2,0.5*a,0.6*b);
    display(t2,2,"s");
    hough(t2,t7,t8);
    add(t6,t8,t5);
    display(t5,0,"s");
    for(i=0; i<a; i++){
     for(j=0; j<b*2; j++){
        if (getpix(t5,i,j)==2){
            arr[0]=i;
            arr[1]=j;
         } /* if */
       } /* for */
      } /* for i */
    }



if(curve!=3){
    sq1=x1*x1+y1*y1;
```

```
                sq2=x2*x2+y2*y2;
                sq3=x3*x3+y3*y3;
                da=x1*y2-x1*y3-y1*x2+y1*x3+x2*y3-x3*y2;
                dd=sq1*(y2-y3)+sq2*(y3-y1)+sq3*(y1-y2);
                de=sq2*(x1-x3)+sq3*(x2-x1)+sq1*(x3-x2);
                D=-dd/da;
                E=-de/da;
                arr[0]=(int)(-D/2);
                arr[1]=(int)(-E/2);
               }

        release(t1);
        release(t2);
        release(t3);
        release(t4);
        release(t5);
        release(t6);
        release(t7);
        release(t8);
return;
}
```

```
/*      whm.cmd   With user interface, extracts a circular arc of white matter


               prompt:          window, res, side
               format:          %d %d %d
               helpfile:
               menu:
               prototype:  int  whm (int w, int res, int side) ;                    */

#include "cvcmd.h"
#include <math.h>
#include    <stdlib.h>
#include    <stdio.h>

static void  curpoint(int arr[]);

static int xs, ys, itype, xrl, xru, yrl, yru;
static int im, w;

void  _whm (void) {

        int     res, wndw, side;
```

```
          wndw = *((int *) arglist[0]);
          res = *((int *) arglist[1]);
          side = *((int *) arglist[2]);
          whm (wndw, res,side);
          }



int       whm (int wndw, int res, int side) {

          int    xc,yc,x1,y1,xf,yf,t1,t2;
          int    t3,t4,t5,t6,t7, j,k, rad;
          int    xfp, yfp, x1p,y1p,arr[2],c;
          float  m,b,radf;


          assignedwindow (wndw);
          w = wndw;
          im = swindow[w].image;
          if (im == res) {
                 response ("Profile image cannot be the same as the displayed image");
                 longjmp (jumpbuf, 1);
                 }
          if((side != 1)&&(side != 2)){
                 printf("You need to specify the side of the image the sample is from.");
                 longjmp (jumpbuf, 1);
                 }

          getsize (im, &xs, &ys, &itype);
          t1=gettempimage();
          t2=gettempimage();
          t3=gettempimage();
          t4=gettempimage();
          t5=gettempimage();
          t6=gettempimage();
          t7=gettempimage();
          form(t1,0,xs,ys,itype);
          form(t5,0,xs,ys,itype);
          form(t7,0,xs,ys,itype);
          form(t6,0,xs,ys,itype);

          arr[0]=0;
          arr[1]=0;


          curpoint(arr); /* locate the center of the circle */
```

```
        xc=arr[0];
        yc=arr[1];
        printf("xc yc %d %d \n",xc,yc);

        for(j=0; j<xs; j++) putpix(50,t1,j,yc);
        /* draws an x-axis through the center of the circle*/


        add(t1,im,t2);

        display(t2,w,"s");

/* ---------------------------------------------------------------------------------------*/
        curpoint(arr); /* locate the boundary   */
        x1=arr[0];
        y1=arr[1];
        printf("x1 y1 %d %d \n",x1,y1);
        form(t3,0,xs,ys,itype);
        rad=abs(xc-x1);
        disc(t3,xs,ys,(float)(xc-xs/2),(float)(yc-ys/2),rad,40);

        add(t3,im,t4);
        copy(t3,res,0);

        display(t4,w,"s");

/*-------------------locate the end of the circular section --------------------*/
/*      curpoint(arr);
        xf=arr[0];
        yf=arr[1];

        form(t5,1,xs,ys,itype);

        if(xc!=xf){
                m=(float)(yf-yc)/(float)(xf-xc);
                b=(float)(yc-m*xc);
                printf("m b %5.2f %5.2f \n", m,b);


                for(j=0; j<xs; j++){
                        for(k=0; k<yc; k++){
                                if(k<(int)(m*j+b)) putpix(0,t5,j,k);
                                if(k==(int)(m*j+b)) putpix(1,t6,j,k);

                                }
                        }
        mask(t3,t5,res);
        }


    if(xc==xf){
        if (side==1){
            for(k=0; k<yc; k++){
                for(j=xc+1; j<xs; j++) putpix(0,t5,j,k);

                }
            }
        if (side==2){
            for(k=0; k<yc; k++){
                for(j=0; j<xc ; j++) putpix(0,t5,j,k);

                }
            }


        mask(t3,t5,res);
        }
                                  */
    release(t1);
    release(t2);
    release(t3);
    release(t4);
    release(t5);
    release(t6);
    release(t7);
    return (TRUE);
    }


static void curpoint(int arr[]){

    int    x,y;
    int    xpel, ypel, xpell, ypell, xpelu, ypelu;
    int    flag;

    printf("enter cursor point \n");
    x = (xs-1)/2;
    y = (ys-1)/2;
```

```
/*      Initialize the cursor */

        flag = cursorposn (w, &x, &y, 0);


/* move cursor to position of first point */

        do {
                pixeltopel (w, x, y, &xpel, &ypel);
                showxhair (w, xpel, ypel);
                flag = cursorposn (w, &x, &y, 1);
                erasexhair (w);
                } while (flag == 0);

        arr[0] = x;
        arr[1] = y;
        pixeltopel (w, arr[0], arr[1], &xpell, &ypell);
        xpelu = xpell;
        ypelu = ypell;

/*      wait for end of button press */

        while ((flag = cursorposn(w, &x, &y, 1)) >= 1) { }



}
```

```
/*      choroid.cmd             tool for tracing the choroid plexus


                prompt:         window, res
                format:         %d %d
                helpfile:
                menu:
                prototype:  int choroid (int w, int res) ;                    */

#include "cvcmd.h"
#include <math.h>
#include    <stdlib.h>
#include    <stdio.h>
```

```
void  _choroid (void) {

        int     res, windw;

        windw = *((int *) arglist[0]);
        res = *((int *) arglist[1]);
        choroid (windw, res);
        }


int     choroid (int windw, int res) {

        int     w,im,t1,t2, t3,t4,j,k,count,i,n,xs,ys,type,x,y;
        int     xarr[30], yarr[30], flag,xpel,ypel,xpell,ypell;
        int     x1,y1,x2,y2,c,d,g;
        float m,b;


        assignedwindow (windw);
        w = windw;
        im = swindow[w].image;
        if (im == res) {
                response ("Result image cannot be the same as the displayed image");
                longjmp (jumpbuf, 1);
                }

        getsize (im, &xs, &ys, &type);
        t1=gettempimage();
        t2=gettempimage();
        t3=gettempimage();


        form(t1,0,xs,ys,type);
        form(t2,0,xs,ys,type);
        form(t3,0,xs,ys,type);


        /*   initialize the array  */
        for(j=0; j<=29; j++){
                xarr[j]=0;
                yarr[j]=0;
                }
        count=0; /* count the number of points input */

        x1=0;
        y1=0;
        x2=0;
        y2=0;
        m=0.0;
        b=0.0;

        x = (xs-1)/2;
        y = (ys-1)/2;

/*      Initialize the cursor */

        flag = cursorposn (w, &x, &y, 0);


        /* move cursor to position of  point */

        do {

                do {
                        pixeltopel (w, x, y, &xpel, &ypel);
                        showxhair (w, xpel, ypel);
                        flag = cursorposn (w, &x, &y, 1);
                        erasexhair (w);
                        } while (flag == 0);


                if(flag == 1){
                        xarr[count] = x;
                        yarr[count] = y;
                        printf(" x y %d %d \n",x,y);
                        count++;
                        }


                m=0.0;
                b=0.0;
                if(count>=2){
                        x1=xarr[count-2];
                        y1=yarr[count-2];
                        x2=xarr[count-1];
                        y2=yarr[count-1];
                        if(x1==x2){
                                if(y1<y2){
                                        i=y1;
                                        n=y2;
                                        }
```

```
            else{
                i=y2;
                n=y1;
                }
            for(k=i; k<=n; k++)
                putpix(1,t1,x1,k);
        } /* end if x1==x2 */

        else{
            m=(float)(y1-y2)/(float)(x1-x2);
            b= y1-m*x1;
            if(y1<y2){
                i=y1;
                n=y2;
                }
            else{
                i=y2;
                n=y1;
                }
            for(k=i; k<=n; k++)
                if(((int)((float)(k-b)/m)>0)&&((int)((float)(k-b)/m)<xs))
                    putpix(1,t1,(int)((float)(k-b)/m)+.5,k);
            if(x1<x2){
                c=x1;
                d=x2;
                }
            else{
                c=x2;
                d=x1;
                }
            for(g=c; g<=d; g++)
                if(((int)(m*g+b+.5)>0)&&((int)(m*g+b+.5)<ys))
                    putpix(1,t1,g,(int)(m*g+b+.5));

            }
        scale(t1,t2,80,0);
        add(t2,im,t3);
        display(t3,w,"s");
        } /* end if count */

} while(flag >=0 );


/* connect all of the points obtained */
/* connect the first and last points */

        x1=xarr[0];
        y1=yarr[0];
        x2=xarr[count-1];
        y2=yarr[count-1];
        if(x1==x2){
                if(y1<y2){
                        i=y1;
                        n=y2;
                        }
                else{
                        i=y2;
                        n=y1;
                        }
                for(k=i; k<=n; k++)
                        putpix(1,t1,x1,k);
                } /* end if x1==x2 */

        else{
                m=(float)(y1-y2)/(float)(x1-x2);
                b= y1-m*x1;
                if(y1<y2){
                        i=y1;
                        n=y2;
                        }
                else{
                        i=y2;
                        n=y1;
                        }
                for(k=i; k<=n; k++)
                        if(((int) ((float)(k-b)/m)+.5>0)&&
                                ((int)((float)(k-b)/m)+.5<xs))
                                    Putpix(1,t1,(int)((float)(k-b)/m)+.5,k);

                if(x1<x2){
                        c=x1;
                        d=x2;
                        }
                else{
                        c=x2;
                        d=x1;
                        }
                for(g=c; g<=d; g++)
                        if(((int)(m*g+b+.5)>0)&&((int)(m*g+b+.5)<ys))
                        putpix(1,t1,g,(int)(m*g+b+.5));
                display(t1,0,"s");
                }

        copy(t1,res,0);
        release(t1);
        release(t2);
        release(t3);
        return (TRUE);
        }
```

```
/* rotateker4.cmd       Change orientation of any kernal, counterclockwise is
negative

                prompt:         source kernal,kernal origin wrt rotation
                                        x,y,ang,result kernal
                format:         %d %d %d %d %d
                menu:
                helpfile:
                prototype:  int rotateker4(int src, int xo, int yo,int ang, int         res);
                                */

#include "cvcmd.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

void  _rotateker4 (void) {

        int     src,xo,yo,ang,res;

        src = *((int *) arglist[0]);
        xo  = *((int *) arglist[1]);
        yo  = *((int *) arglist[2]);
        ang  = *((int *) arglist[3]);
        res = *((int *) arglist[4]);
        rotateker4 (src, xo,yo,ang,res);
        }

int rotateker4 (int src, int xo, int yo, int ang, int res) {

        int     x, y, xs, ys,x1,y1, type,t1, t2, t3, t4, t5, t6, t7;
        int     xf2,yf2,xl2,yl2, xf1,yf1,xl1,yl1;
        int     c,d, j,k, x2,y2,xp,yp,deg,a, xu, yu,xf,yf,xl,yl,xm,ym;
        int     xp2,yp2,pix,pixn,pixs,pixe,pixw;
        double  theta,dist,dist2,eta,phi,zeta, alpha,beta;
        float  xc,yc;
        int     minx,miny,maxx,maxy,t8,t9;


        t1=gettempimage();
        t2=gettempimage();
        t3=gettempimage();
        t4=gettempimage();
        t5=gettempimage();
        t6=gettempimage();
        t7=gettempimage();
        t8=gettempimage();
        t9=gettempimage();

        printf("t1,t2,t3,t4,t5,t6,t7 %d %d %d %d %d %d \n",t1,t2,t3,t4,t5,t6,t7);

        getsize(src,&xs,&ys,&type);
        c=max(xs,ys);
        form(res,0,3*c,3*c,type);
        form(t1,0,xs,ys,type);
        form(t5,0,3*c,3*c,type);
        form(t7, 0, 3*c,3*c,type);


        theta=(float)ang*pi/180;
        deg=ang;




        /* locate the origin of the scaled kernal in the result */
        x2=(int)(1.5*c);
        y2=(int)(1.5*c);


    /* now rotate the kernal so the orientation matches that defined by the target */

        xl=0;
        yl=0;
        ym=9999;
        xm=9999;



    /*   ************** performing vector addition - see notes for notation  ***       */
```

```
if(deg!=0){
    if(deg<0){                  /*  then   rotate counterclockwise*/
        for(x=0; x<xs; x++){
            for(y=0; y<ys; y++){
                    dist=sqrt((x-xo)*(x-xo)+(y-yo)*(y-yo));
        /* dist. from center of rot. and a kernal point*/
                    zeta=atan2((double)(y-yo),(double)(x-xo));
                    alpha=zeta-theta;
                    xp=x2+dist*cos(alpha)+.5;
                    yp=y2+dist*sin(alpha)+.5;
                    if((xp>0)&&(xp<3*c)&&(yp>0)&&(yp<3*c))
                    {
                        if(ym>yp) ym=yp;
                        if(xm>xp) xm=xp;
                        if(yl<yp) yl=yp;
                        if(xl<xp) xl=xp;
                        putpix(getpix(src,x,y),t5,xp,yp);
                        putpix(1,t7,xp,yp);
                    } /* end if xp yp */



            }  /* for y */
        } /* for x */
        fillholes(t7,t8);
        loadascii("5x5",t6);
        dilate(t8,t6,t4);
        /* the rotated kernal has holes, correct that */
        for(x=0; x<3*c; x++){
            for(y=0; y<3*c; y++){
                if((getpix(t4,x,y)>0)&&(getpix(t7,x,y)==0)){
                    dist=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
                    zeta=atan2((double)(y-y2),(double)(x-x2));
                    alpha=zeta+theta;
                    xp=(int)((dist*cos(alpha)+.5)+xo);
                    yp=(int)((dist*sin(alpha)+.5)+yo);

                    if((xp>=0)&&(xp<xs)&&(yp>=0)&&(yp<ys)){
                        putpix(getpix(src,xp,yp),t5,x,y);
                        putpix(2,t4,x,y);
                        }

if(((xp<0)||(xp>xs)||(yp<0)||(yp>ys))&&(getpix(t8,x,y)>0)){

                        pixn=getpix(t5,x,y+1);
                        pixs=getpix(t5,x,y-1);
                    pixe=getpix(t5,x+1,y);
                    pixw=getpix(t5,x-1,y);
                    putpix((pixe+pixn+pixw+pixs)/4,t5,x,y);
                    }
                } /* end if getpix */
            } /* end for y */
        }/* end for x */

}  /* endif deg<0 */


if(deg>0){ /* rotate clockwise */
    for(x=0; x<xs; x++){
        for(y=0;y<ys; y++){
                dist=sqrt((x-xo)*(x-xo)+(y-yo)*(y-yo));
    /* dist. from center of rot. and a kernal point*/
                zeta=atan2(y-yo,x-xo);
                alpha=zeta-theta;
                xp=x2+dist*cos(alpha)+.5;
                yp=y2+dist*sin(alpha)+.5;


            if((xp>0)&&(xp<3*c)&&(yp>0)&&(yp<3*c)) {
                    if(yl<yp) yl=yp;
                    if(xl<xp) xl=xp;
                    if(ym>yp) ym=yp;
                    if(xm>xp) xm=xp;
                    putpix(getpix(src,x,y),t5,xp,yp);
                    putpix(1,t7,xp,yp);
                } /* end if xp yp */



        } /* end for y */
    }/* end for x */
    fillholes(t7,t8);
    loadascii("5x5",t6);
    dilate(t8,t6,t4);
/* the rotated kernal has holes, correct that */
    for(x=0; x<3*c; x++){
        for(y=0; y<3*c; y++){
            if((getpix(t4,x,y)>0)&&(getpix(t7,x,y)==0)){

                dist=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
                zeta=atan2((double)(y-y2),(double)(x-x2));
                alpha=zeta+theta;
```

```
                    xp=(int)(dist*cos(alpha)+xo+.5);
                    yp=(int)(dist*sin(alpha)+yo+.5);
                    if((xp>=0)&&(xp<xs)&&(yp>=0)&&(yp<ys)){

                        putpix(getpix(src,xp,yp),t5,x,y);
                        putpix(4,t4,x,y);
                        }


if(((xp<0)||(xp>xs)||(yp<0)||(yp>ys))&&(getpix(t8,x,y)>0)){

                        pixn=getpix(t5,x,y+1);
                        pixs=getpix(t5,x,y-1);
                        pixe=getpix(t5,x+1,y);
                        pixw=getpix(t5,x-1,y);
                        putpix((pixe+pixn+pixw+pixs)/4,t5,x,y);
                        }

                    } /* end if getpix */
                } /* end for y */
            }/* end for x */
    }/* end if deg>0 */

} /* end if deg!=0  */

else{  /* if deg==0 (m1==0) then no rotation is necessary */
    mount(src,t5,x2,y2);
    form(t6,1,xs,ys,type);
    mount(t6,t7,x2,y2);
    yl=ys-1;
    xl=xs-1;
    ym=0;
    xm=0;

}

/* remove the excess around the rotated image */
    if (xm>=1) xm =xm-1;
    if (ym>=2) ym=ym-2;
    setroi(xm,xl+1,ym,yl+2,9);
    crop(t5,res,9);
```

```
        release(t1);
        release(t2);
        release(t3);
        release(t4);
        release(t5);
        release(t6);
        release(t7);
        release(t8);
        release(t9);



        return(TRUE);
        }   /* end rotateker4.cmd */
```

```
/* vconv2.cmd    Vary kernel for US based on location in image and perform
convolution

            prompt:         source image, source angx10 image,new kernal
                        length (odd #), source kernal, output
            format:         %d %d %d %d %d
            menu:
            helpfile:
            prototype:  int vconv2(int image,int ang10,int leng, int ker,
                                int res);            */

#include "cvcmd.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>



void  _vconv2 (void) {

        int     image,ang10,leng, ker,res;

        image = *((int *) arglist[0]);
        ang10 = *((int *) arglist[1]);
        leng = *((int *) arglist[2]);
        ker = *((int *) arglist[3]);
        res = *((int *) arglist[4]);
        vconv2 (image, ang10, leng, ker,res);
        }

int vconv2 (int image, int ang10, int leng,int ker, int res) {

        int     x, y, xs, ys, type, t1, t2, t3, t4, t5, t6, t7;
        int     xcoord[101],ycoord[101],xc,yc,j,xker,yker, kertype;
        int     xtemp,ytemp,temptype, theta,dI,k, plength,dx,dy;
        int     alpha, upsam;
        float  m1,b1,b2,weights[101],test[101],factor,step,sum;
        float  sum1,sum2,value;
        float  avgold, avgnew;


        t1=gettempimage();
        t2=gettempimage();
        t3=gettempimage();
        t4=gettempimage();
        t5=gettempimage();
        t7=gettempimage();

        getsize (image, &xs, &ys, &type);
        form(res,0,xs,ys,1);
        form(t1,0,xs,ys,1);




/* -------------------- variable convolution   portion --------------------- */

        for(x=0+leng/2; x<xs-leng/2; x++){
                for(y=0+leng/2; y<ys-leng/2; y++){
                        if(getpix(ang10,x,y)>0){
                                theta=(int)((float)getpix(ang10,x,y)/10);
                                if (theta!=180){
                                        m1=-tan((theta-90)*pi/180);
                                        }
                                if (theta==180){
                                        m1=0;
                                        }
                                b1=(float)(y-m1*x);
                if(y%300==0)
                printf(" x y  theta m1 b1 %d %d %d %5.2f %5.2f \n ",x,y,theta, m1, b1);
                plength=0;


      /*----------------- find the footprint of the rotated kernal ---------------------*/
                        form(t1,0,xs,ys,1);
                        if((m1<-0.001)||(m1>0.001)){
                            alpha=abs(180-theta);
                            dx=(int)(abs(leng*cos((90-alpha)*pi/180))/2);
                            dy=(int)(abs(leng*sin((90-alpha)*pi/180))/2);

                            if ((dx!=0)&&(dy!=0)){
                                for(k=y-dy; k<=y+dy; k++){
                                    for(j=x-dx; j<=x+dx; j++){
                                        if (j==(int)(((float)(k-b1)/m1)+.5)){
                                            putpix(1,t1,j,k);
                                            }
                                        } /* end for j */
                                    } /*end for k */

           /* t1 is a mask of the footprint image for the source */




      /* ----------  get final size and coords of the footprint of the new kernel ------*/
                                plength=0;
                                for(k=y-dy; k<=y+dy; k++){
                                    for(j=x-dx; j<=x+dx; j++){
                                        if (getpix(t1,j,k)>0){
                                            xcoord[plength]=j;
                                            ycoord[plength]=k;
                                            plength++;
                                            } /* end if */
                                        } /* end for j */
                                    } /* end for k */
                                } /* end if dx, dy */

      /* plength is the pixel length of the new kernal, xcoord, ycoord are the
                                coordinates of the mask  */

                            if (dx==0){ /* then the mask is nearly vertical */
                                for(k=y-leng/2; k<=y+leng/2; k++){
                                    xcoord[plength]=x;
                                    ycoord[plength]=k;
```

```
putpix(1,t1,xcoord[plength],ycoord[plength]);
                                        plength++;
                                        }
                                }
                        if (dy==0){  /* then the mask is horizontal */
                                for(k=x-leng/2; k<=x+leng/2; k++){
                                        xcoord[plength]=k;
                                        ycoord[plength]=y;

putpix(1,t1,xcoord[plength],ycoord[plength]);
                                        plength++;
                                        }
                                }
                        } /* end if m1<0.001 ... */

                if((m1<0.001)&&(m1>-0.001)){
                        for(j=-leng/2; j<=leng/2; j++){
                                xcoord[j+leng/2]=x;
                                ycoord[j+leng/2]=y+j;
                                plength++;
                                putpix(1,t1,xcoord[j+leng/2],ycoord[j+leng/2]);
                                }
                        } /* end if m1==0 */



                /* end finding the footprint */


        dI=0;
        step=0.0;
        avgold=0.0;
        avgnew=0.0;
        upsam=0;

/*  ------------------- determine the weights of the new kernel -----------------*/
        if (plength!=0){
                getsize(ker,&xker,&yker,&kertype);
                form(t2,0,xker,yker*plength,3);
                form(t3,0,xker,plength,3);
                form(t4,0,xker,plength,3);
                form(t5,0,xker,plength,1);
```

```
                                upsam=(int)(yker*(float)((plength-1))/(float)((yker-1)));


                                for (j=0; j<yker-1; j++){
                                        dI=getpix(ker,0,j+1)-getpix(ker,0,j);
                                        step=(float)dI/(float)((upsam+1));
                                        value=(float)(getpix(ker,0,j));
                                        putfpix((float)(value),t2,0,j*(upsam+1));
                                        avgold=value+avgold;
                                        for(k=1; k<=upsam+1; k++){

        putfpix((float)(value+step*k),t2,0,j*(upsam+1)+k);
                                                }

                                        }

                                putfpix((float)getpix(ker,0,yker-1), t2,0,yker*plength-1);



                                for (j=0; j<plength; j++){
                                        putfpix(getfpix(t2,0,j*yker),t3,0,j);
                                        avgnew=getfpix(t2,0,j*yker)+avgnew;
                                        }


        /* normalize the weights  - to have the same dc value*/

                                for (j=0; j<plength; j++){
                                        weights[j]=getfpix(t3,0,j)*avgold/avgnew;
                                        putpix((int)(weights[j]),t5,0,j);
                                        }

        /* ---------------------- end finding the weights ------------------- */




/* ------------------------------ perform the convolution  ------------------  */
                                sum=0.0;
                                for(j=0; j<plength; j++){
                                  sum=getpix(image,xcoord[j],ycoord[j])*weights[j]+sum;
                                  }

                                putpix((int)sum, res,x,y);
```

```
                        } /* end if plength != 0 */
                        } /* end  if getpix ang10 */


                } /* end for y */
        } /* end for x */


        release(t5);
        release(t1);


        return (TRUE);
        } /* end of main  */
```

```
/*      maskinfoavg             get statistics of an image mask and output to a file    */

/*              prompt:         src lft,  region lft, src rt, region rt, output
                                        filename, image name, mask name
                format:         %d %d %d  %d %s  %s %s
                helpfile:
                menu:
                prototype:  int maskinfoavg (int src1,int ma1,int src2,
                                int ma2, char *finame, char *imna, char *mna) ;
                */

#include "cvcmd.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void  _maskinfoavg (void) {

        int     src1,ma1,src2, ma2;
        char *finame, *imna, *mna;

        src1 = *((int *) arglist[0]);
        ma1 = *((int *) arglist[1]);
        src2= *((int *) arglist[2]);
        ma2= *((int *) arglist[3]);
        finame = ((char *) arglist[4]);
        imna = ((char *) arglist[5]);
        mna = ((char *) arglist[6]);

        maskinfoavg (src1, ma1,src2, ma2,finame, imna, mna);
        }


int maskinfoavg (int src1,int ma1,int src2, int ma2, char *finame, char *imna, char
*mna) {

        int    xs, ys, type,exist=0, max=0, min=0,med1=0,med2=0,med=0;
        float  sd=0.0, ave=0.0,sa1=0.0,sk1,kur1,sk2,kur2, sa2=0.0, sa=0.0,kur,sk;
        FILE *out, *in;
        char   fname[80],imn[80], mn[80],num[3];

        stats2 (src1, -ma1,&sk1,&kur1);
        stats2 (src2, -ma2,&sk2,&kur2);
        kur=0.5*(kur1+kur2);
        sk=0.5*(sk1+sk2);
        getsize (src1, &xs, &ys, &type);
        sd= 0.5*(imtab[src1].stddev+imtab[src2].stddev);
        ave=0.5*(imtab[src1].avgpix+imtab[src2].avgpix);
        max=0.5*(imtab[src1].maxpix+imtab[src2].maxpix);
        min=0.5*(imtab[src1].minpix+imtab[src2].minpix);

        normsurfarea2(src1,ma1,&sa1);
        normsurfarea2(src2,ma2,&sa2);
        sa=0.5*(sa1+sa2);
        maskmed(src1,ma1,&med1);

        maskmed(src2,ma2,&med2);
        printf("1 /n");
        med=0.5*(med1+med2);
        printf("2 /n");

        strcpy(fname,finame); /* output file */
        printf("3 /n");
        strcpy(imn,imna);  /* image name */
        printf("4 /n");
        strcpy(mn,mna); /* mask name */
        printf("5 /n");
        strncpy(num,imna,5);
        printf("6 /n");


        in=fopen(finame,"r");
        printf("7 /n");
        if(in==NULL) exist=1;
        printf("8 /n");
        out=fopen(finame, "a");
        printf("9 /n");
        if(exist==1){
          fprintf(out,"proc.im  pat.  mask avg NSA   avgstdev   avgmean   avgmax
        avgmin");
          fprintf(out,"     avgskavgkurt     avgmed\n");
              }
        printf("10 /n");
        fseek(out,0,2);
        fprintf(out, "%s  %s   %s   %f   %f   %f   %dn
                    %d",imn,num,mn,sa,sd,ave,max,min);
        fprintf(out, "      %f   %f   %d   \n", sk, kur,med);
        fclose(out);




        return (TRUE);
        }
```

```
/*          mode5             mode5 filter an image                  */

/*          prompt:           source image, result image, window size (<= 65)
            format:           %d %d %d
            helpfile:
            menu:
            prototype:  int   mode5 (int src, int rslt, int w) ;                  */

#include "cvcmd.h"
#include <stdlib.h>
#include <stdio.h>

static int    compare (const void *e1, const void *e2) ;

void  _mode5 (void) {

        int     src, rslt, w;

        src = *((int *) arglist[0]);
        rslt = *((int *) arglist[1]);
        w = *((int *) arglist[2]);
        mode5 (src, rslt, w);
        }


int     mode5 (int src, int rslt, int w) {

        int     p, i, j, x, y,q,l,m,v, xsize, ysize, type,value;
        int     list[4225], count[4225], list2[4225], maxcount,nomode,
        int     k, check[4225], med,c,d, f,h;

        getsize (src, &xsize, &ysize, &type);
        form (rslt, 0, xsize, ysize, type);

/* if a mode5 does not exist in the region of interest, insert the median in the output
instead */

        w = min(w, 65);
        for(v=0; v<4225; v++){
                count[v]=0;
                list[v]=0;
                }

        for (y = w/2; y < ysize-w/2; y++) {
                for (x = w/2; x < xsize-w/2; x++) {
                        q=0;
                        for(v=0; v<4225; v++) count[v]=0;

                        /* find the mode5 */
                        for (j = -w/2; j <= w/2; j++) {
                                for (i = -w/2; i <= w/2; i++) {
                                        list[q]=getpix(src,x+i,y+j);
                                        q++;
                                        }
                                }
                        for(l=0; l<q;l++){
                                value=list[l];
                                for( m=0;m<q;m++){
                                        if(value==list[m]) count[m]=count[m]+1;
                                        }
                                }

                        for(l=0; l<q;l++){
                                list2[l]=list[l];
                        }
```

```
            maxcount=0;
            p=0;
            for(l=0; l<q; l++){
                if (maxcount<count[l]){
                    maxcount=count[l];
                    p=l;
                    }
                }

/* check to see if all values occur equally but greater than 1 or if multimodal */
            k=0;
            c=0;
            d=1;
            f=0;
            for(v=0; v<4225; v++){
                check[v]=-99999;
                }

            for(l=0; l<q; l++){
                if (maxcount==count[l]) k++;
                }
            nomode=0;
            if(k== w*w) nomode=1;

/* if multimodal put the mode which is closest to the med pixel value in the result */
            if(( k>maxcount)&&(nomode==0)){
                qsort (&list2, w*w, sizeof(int), compare);
                med=list2[(w*w)/2];
                for(l=0; l<q; l++){
                    if (count[l]==maxcount){
                        check[l]=list[l];

                        if( list[l]==med) c=1;
                        }
                    }
                if(c==1) putpix(med,rslt,x,y);
                else{
                    while (f==0){

                        for(l=0; l<q; l++){

                            if ((check[l]==med+d)&&(f==0)){
                                putpix(med+d,rslt,x,y);
                                f=1;
                                }
                            if ((check[l]==med-d)&&(f==0)){
                                putpix(med-d,rslt,x,y);
                                f=1;
                                }
                            } /* end for l */
                        d++;

                        }  /* end while */
                    } /* end else  */
                d=1;
                } /* end if k */




            if (((maxcount>1)&&(nomode==0))&&(k<=maxcount))
            putpix(list[p],rslt,x,y);




            /* if mode5 does not exist, insert median */
            if(( maxcount<=1)||(nomode==1)){
                qsort (&list, w*w, sizeof(int), compare);
                putpix( list[(w*w)/2], rslt,x,y);
                } /* end if getpix */

            } /* end for x */
        } /* end for y */

    return (TRUE);
    }

static int compare (const void *e1, const void *e2) {

    return (max(-1, min(1, *(int *)e1 - *(int *)e2)));
    }
```

Appendix B

Reports on Early Experiments

Texture-based Tissue Characterization: A Novel Predictor for Brain Injury?

Tyna Hope
Sexton Campus, Dalhousie University
PO Box 1000, Halifax, Ca. B3J 2X4

Peter Gregson
Sexton Campus, Dalhousie University
PO Box 1000, Halifax, Ca. B3J 2X4

Norma Linney
Saint Mary's University
923 Robie Street, Halifax, Ca. B3H 3C3

Matthias Schmidt
IWK Health Centre
PO Box 3070, Halifax, Ca. B3J 3G9

Abstract

Premature infants are prone to white matter damage (WMD), which is associated with cerebral palsy (CP) and cognitive impairment. Ultrasound (US) is the preferred imaging modality to detect WMD. To improve on existing diagnostic rates, quantitative measures incorporating new information are needed. We are investigating US texture measures as new indicators of white matter health. We are developing techniques which enhance image texture differences that may reflect pathological changes.

Earlier experiments using data from 30 patients indicate that tissue types may be segmented based on texture measures. Present experiments using images from 18 patients (12 with normal outcome, 6 who developed CP) reveal that the measures from CP patients form separate populations from healthy patients. Texture measures were obtained without compensating for operator-dependent machine settings and without suppressing speckle. Digitized analogue ultrasound films are used as the input data since modern digital ultrasound machines have bandwidth restrictions. The effect of image resolution reduction on the texture-based methods is part of an on-going investigation.

Key Words

ultrasound tissue characterization, white matter damage, ultrasound image texture

Introduction

The purpose of our work is to detect white matter damage (WMD) in the cranial ultrasound (US) images of premature infants more reliably and earlier than is currently possible. Premature infants are prone to white matter damage (WMD) which is associated with subsequent cerebral palsy (CP) and cognitive impairment [1]. Ultrasound is commonly used to screen for WMD. Although MRI detects WMD with greater sensitivity [2], ultrasound remains the imaging modality of choice because it is portable, it requires no sedation and it is more readily available. Currently, diagnosis with ultrasound is qualitative. Radiologists depend on echolucencies and echodensities as cues to the presence of WMD [3]. Since US intensity information alone is not satisfactory for diagnosing WMD [2] [4], other measures are required.

Texture information is an additional measure that is explored in this work. A previous study indicates that texture may be useful for the detection of cerebral hypoxic insult in term infants [5]. We are investigating texture as a predictor of periventricular leukomalacia in preterm infants. In an earlier set of experiments, we found that tissue segmentation was possible by preprocessing to enhance texture and then obtaining a measure of the intensity over a single tissue sample. We are now assessing the ability of this method to discriminate between images of healthy patients and patients who went on to develop CP.

The outcome of this study is that preprocessing image data to enhance texture differences and then obtaining measures of that difference results in two populations corresponding to the patient outcome. This inference is supported by a two sample t-test on the experimental data. The existence of two populations within the experimental data is significant because the algorithms use images which are not corrected for operator-dependent settings. As well, the input images were not selected dependent upon the presence of visually obvious flares for the indication of the presence of periventricular leukomalacia by a radiologist. The encouraging results in this preliminary study suggests that additional diagnostic information may be present in ultrasound tissue and that further study on the correlation of US image texture and brain injury is warranted.

Experimental Methods

- The images used in this study are of premature infants who had cranial ultrasound scans between 1990 and 1991 at the IWK Health Centre in Halifax, Canada. The patients were identified through the Perinatal Follow-up Database. To be included in this study, the patients had to be less than 31 weeks gestational age, and the final outcome of the patient had to be known. Eighteen patients were included in this research. None of the images had unequivocal evidence of periventricular leukomalacia (i.e. cystic periventricular echolucencies). The group contained six patients who subsequently developed CP.

The sample size is small due to a number of factors. We require the images in this preliminary study to be analogue data stored on film. To obtain this type of image, we are restricted to US films from 1990 and 1991. The population from which the images could be drawn is small, and some eligible patients from this period had follow-up care elsewhere, making their records inaccessible. The images were scanned with a 3200 dpi scanner to maintain the high resolution of the images. The effect of the reduced spatial resolution on the information content of texture measures has yet to be determined.

For each patient, a single semi-coronal image at the level of the atria of the lateral ventricles was chosen. The data set contained images obtained with a variety of machines and various operator-dependent settings.

- Due to the large size of the cranial US images, they were first divided into halves, representing the left and right hemispheres. Therefore, each patient in this study is associated with two half US images. Samples (masks) of two tissue types, choroid plexus and white matter, were selected from each patient's half image by a radiologist, where it was possible to select the white matter with a high degree of certainty. For all but one patient, images for both sides of the patient's brain were used. The masks are rectangular in shape with their major axis oriented approximately along the path of insonation (Figure 1). The choroid plexus was chosen as the second tissue type as it is the traditional standard used to compare white matter echogenicity.

- The selected regions were processed to enhance the differences in texture between the two tissue types. The results presented in this paper include preprocessing with two non-linear techniques. These techniques are based upon intensity distribution based filters (DMM) and grey-level morphology (GM). No preprocessing was performed to compensate for the operator-dependent machine settings. Speckle was not suppressed as we are exploring speckle as a component of texture (Figure 2).

- The standard deviation (STDV) of the intensity was calculated within the choroid plexus and white matter masks for each patient after processing. For a typical patient, four STDV values were calculated after being processed with DMM. Two values were obtained from the white matter, one from the left side and one from the right side. Likewise, the left and right STDV values were calculated for the choroid plexus regions. The same values were obtained for each patient after processing with GM.

- The STDV values obtained for each tissue after processing were analyzed to determine if two populations exist corresponding to the two patient outcomes.

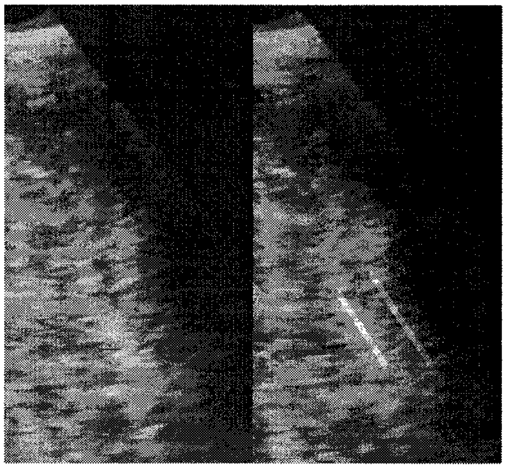

Figure 1: Left side of a patient's cranial sonogram. Right image demonstrates the location of the choroid plexus and white matter masks.

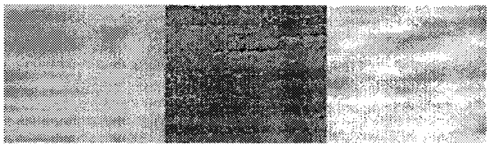

Figure 2: Portion of choroid plexus taken from the previous figure. It is shown unprocessed (left), after processing with DMM (middle), after GM (right). The intensity of the images have been scaled for display.

Theoretical Basis

We use cranial ultrasound images without preprocessing to remove speckle. When ultrasonic pulses are applied to biological tissues, they encounter many different-sized obstacles. Structures much larger than a wavelength result in specular reflection, while those much smaller than the ultrasonic wavelength result in fully developed speckle. Reflections due to microstructures may be correlated with pathology [6] and so we have included speckle in the texture analysis.

Ultrasound image texture has been investigated for disease detection in many soft tissues. Diseases of the liver have received a significant amount of attention [7],[8],[9],[10]. While there are non-trivial differences between the neonatal brain and liver, both are soft tissues. In both cases, disease processes disrupt cell function, then tissue architecture. Advances in the use of texture for the detection of liver disease suggest that texture should be explored as a predictor of brain injury. More importantly, previous work by others [5] indicates that US texture may be useful to indicate the occurence of hypoxic insult in term infants. Although premature and term infant brains differ at the histological level, this work suggests that further investigation of the application of US texture and brain injury in premature infants is warranted.

Stippel et. al. [11] have investigated speckle classification and suppression techniques as part of an algorithm to help radiologists assess white matter damage using flare (areas of increased echogenicity) symmetry. They adaptively filter speckle to assist a region-growing technique that outlines white matter. Symmetry of white matter flares is a useful feature for assessing the extent of white matter damage. We are developing a classification scheme independent of flare symmetry that incorporates speckle. Our texture measures are independent of the presence of visually obvious flares and thus may allow earlier diagnosis than is presently available.

We preprocess the images using two techniques. In the first technique, we enhance the edge information in the ultrasound images by obtaining the difference in response between two intensity distribution based filters (DMM). The technique is applied as a method to enhance the diffuse edges that are found within ultrasound images. Median and mode filters are dependent upon the local intensity histogram properties. These filters have been shown to respond differently to non-step edges [12]. The difference between the mode and the median filter (DMM) can be exploited as an edge detector. At step edge locations and constant intensity regions, the responses of the median and mode filters are the same, and so their difference is zero. For other types of edge profiles, such as sigmoid and linear profiles, the responses of the two filters differ. DMM produces an output which varies with edge profile. For a noise-free, linear edge profile, the output image has an odd profile containing a zero crossing at the input edge location.

To implement this technique, the mode algorithm was designed to compensate for the fact that the mode of the local histogram may not be uniquie or may be undefined. Three algorithms were investigated for a non-unique mode. These included selecting the first mode detected, the mode closest to the center value, and the mode closest to the median. For our purposes, all algorithms worked equally well, and the first mode detected was chosen to realize DMM for the data presented here. In the case where the histogram was flat, and thus the mode did not exist, the median value was calculated and thus the output of DMM is zero at these locations.

In the second technique, we further investigate the spatial relationships between local grey levels through the use of grey level morphology (GM). We dilate the image using a flat binary disc. This dilated image is then subtracted from the unprocessed image to reveal local intensity variations that are related to the size of the disc. We use a disc with a radius of 30 pixels based upon the results of our earlier experiments to segment tissue types.

Results and Discussion

In searching for a new method to describe white matter health, we had expectated that the values for the white matter for the Control and CP groups would be different, whereas the values for the choroid plexus would remain constant. We found the opposite to be true (Table 1). Shown are the descriptive statistics of the white matter texture measure and the choroid plexus texture measure after processing with DMM. Similar results were obtained for the morphology technique. Present qualitative assessment techniques compare the appearance of the white matter to the choroid plexus. This comparison may be relevant for texture measures as well as for intensity. It is for these reasons that we evaluate comparative measures between the white matter and the choroid plexus. We calculate the ratio and the difference between the white matter STDV and the choroid plexus STDV for each patient. We then average each of these values for the two sides of the brain to obtain a single ratio and a single difference for each patient. See figure 3 for an illustration of the data obtained during the experiments.

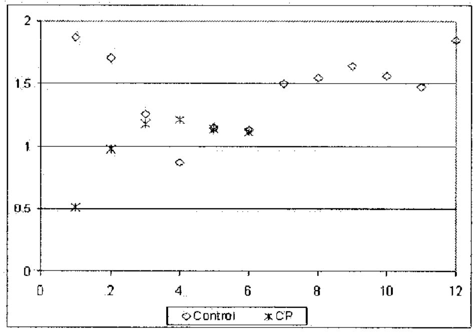

Figure 3: Results using the morphology technique. The average ratio for the two hemispheres of white matter vs. choroid plexus is plotted for each patient.

The values for the control and CP groups were assessed to determine how much, if any, difference existed between them. Table 2 provides statistics describing the experimental data. As a measure of the confidence of the sample means, the means ± twice the standard error was examined (Figure 5). In three of the texture measures these regions did not overlap. As a measure of the separation of the populations, for all texture measures the mean ± one standard deviation are plotted (Figure 6). The difference between the white matter and choroid plexus after processing with

Statistic	White Matter Control	Choroid Plexus Control	White Matter CP	Choroid Plexus CP
Mean	2.904	1.978	2.633	2.775
Median	2.7	1.9	2.75	2.65

Table 1: Statistics describing the measures of white matter and choroid plexus for both the control and CP groups. THe measures are obtained after DMM processing and calculated over areas containing a single tissue type.

Experiment	Processing	Group	Measure	Mean	Standard error	Standard Deviation	Low	High
1	Morphology	Control	Ratio	1.462	0.088	0.305	1.157	1.767
1	Morphology	CP	Ratio	1.019	0.107	0.262	0.757	1.281
2	Morphology	Control	Difference	3.746	0.758	2.627	1.119	6.373
2	Morphology	CP	Difference	-0.0917	1.404	3.439	-3.531	3.347
3	DMM	Control	Ratio	1.524	0.094	0.326	1.198	1.850
3	DMM	CP	Ratio	0.948	0.095	0.233	0.715	1.181
4	DMM	Control	Difference	0.942	0.167	0.580	0.362	1.522
4	DMM	CP	Difference	-0.142	0.223	0.545	-0.687	0.403

Table 2: Statistics describing the measures from the control and CP groups. The "Low" value indicates the mean minus the standard deviation. The "High" value is the mean plus the standard deviation.

DMM results in two groups where the means ± one standard deviation do not overlap. See figure 6 for a sample histogram of the measures obtained in these experiments.

The two sample t-test was performed for the results of each of the four experiments to determine if two populations exist, these separating into the texture measures for the control group and the texture measures for the CP group. For the t-test, it is assumed that the unknown variances are not equal. A cutoff of $\alpha = 0.05$ was used. The null hypothesis (H_0) is that the means of the two groups are equal. In all cases the test statistic was greater than t and thus H_0 was rejected. The P value is provided as an additional measure of the significance of the experimental results. See table 3 for details.

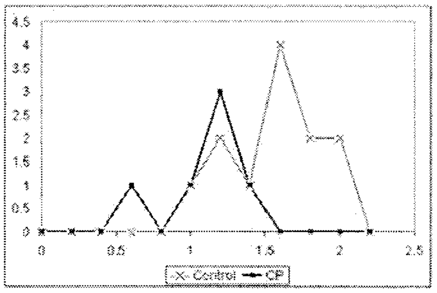

Figure 6: Histograms of the average of the difference between the white matter and the choroid plexus after morphology based processing. Refer to table 2 for statistics describing the populations.

Conclusions

Texture measures have been obtained without prior compensation for operator-based machine settings and without using speckle-reduction techniques. Statistical analysis of these texture measures indicate that two populations exist coresponding to the patient outcome. Since these results were obtained using uncompensated image data, this suggests that the texture components we are measuring are robust to the machine settings. It also suggests that it is not necessary to remove speckle prior to assessing ultrasound texture.

Our measures indicate a change in choroid plexus rather than white matter. This observation may be related to machine settings or may be related to pathology. The mechanism underlying this observation requires further attention.

The Control Group and CP Group separate into two populations using two different texture measurement techniques. While the studies performed to date are on a small population, the results suggest that additional studies with a larger population are warranted. The successful discovery of additional diagnostic measures may provide a tool for earlier medical intervention.

Acknowledgements

The authors would like to thank William Phillips of the Engineering Mathematics Department, Dalhousie University, for his assistance in the analysis of the data. We would also like to thank Michael Vincer, the Director of the IWK Perinatal Follow-up Clinic and Dora Stinson, Medical Director of the Special Care Nursery, IWK for their support. Financial support for this work has been provided by the Natural Science and Engineering Research Council of Canada, a Category B grant of the IWK Health Centre and an IWK Auxiliary Award.

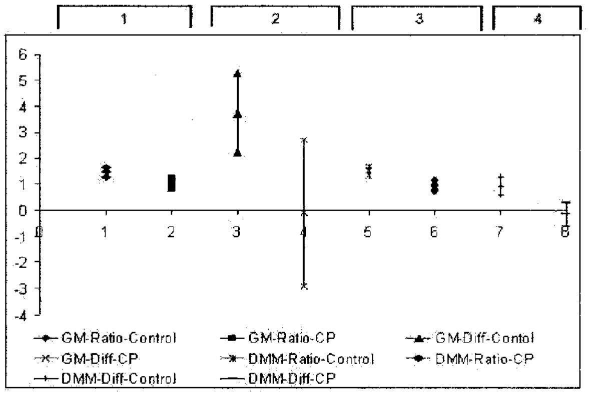

Figure 4: Graph indicating the mean ± twice the standard error. In three of four cases, the Control and CP Groups do not overlap. The pair numbering corresponds to the numbers in Table 2.

Processing method, texture measure	Calculated test statistic	Degrees of freedom	t	P
GM,ratio	3.20	12	2.179	.0076
GM,difference	2.405	8	2.306	0.0428
DMM,ratio	4.305	13	2.160	0.0008
DMM,difference	3.144	10	2.280	0.0104

Table 3: Results of the t-test for each experiment. In all cases the null hypothesis, which would be that the population means are equal, was rejected.

References

[1] J.J. Volpe. "Neurobiology of periventricular leukomalacia in the premature infant." *Pediactric Research*, 50(4),2001,553-62.

[2] E.F. Maalouf, P.J. Duggan, S.J. Counsell, M.A. Rutherford, F. Cowan, D. Azzopardi and A.D. Edwards. "Comparison of Findings on Cranial Ultrasound and Magnetic Resonance Imaging in Preterm Infants." *Pediatrics*, 107(4), April 2001,719-727.

[3] L.S. de Vries, P. Eken, L.M. Dubowitz. "The spectrum of leukomalacia using cranial ultrasound." *Behavioural Brain Research*, 49(1), 1992, 1-6.

[4] E.E. Holling."Characteristics of cranial ultrasound white-matter echolucencies that predict disability: a review." *Developmental Medicine and Child Neurology*, 41,1999,136-139.

[5] L.L. Barr, P.J. McCullough , W.S. Ball, B.H. Krasner, B.S. Garra and J.A. Deddens. " Quantitative Sonographic Feature Analysis of Clinical Infant Hypoxia: A Pilot Study." *American Journal of Neuroradiology*, 17, June 1996, 1025-1031.

[6] J. Thijssen. "Ultrasonic speckle formation, analysis and processing applied to tissue characterization." *Pattern Recognition Letters*, 24, Feb. 2003, 659-675.

[7] P.C. Hartman, B.J. Oosterveld, J.M. Thijssen and G.J.E. Rosenbusch. "Variability of Quantitative Echographic Parameters of the Liver : Inntra- and Interindividual Spread, Temporal and Age-related effects." *Ultrasound in Medicine and Biology*, 17(9), 1991, 857-867.

[8] Y.M. Kadah, A.A. Faraq, J.M. Zurada, A.M. Badawi and A.M. Youssef. "Classification Algorithms for Quantitative Tissue Characterization of Diffuse Liver Disease from Ultrasound Images." *IEEE Transactions ofMedical Imaging*, 15(4), August 1996, 466-478.

[9] M.H. Loew, R. Mia and Z. Guo. " An Approach to image Classification in Ultrasound." *Proceedings of the 29th Applied Imagery Pattern Recognition Workshop*, Washington, D.C., USA, 2000, 193-199.

[10] M.J. Gangeh, M. Hanmandlu and M. Bister. "A Fuzzy-Based Texture Analysis for Tissue Characterization of Diffused Liver Disease on B-scan Images." *Conference Proceedings of the 38th Annual Rocky*

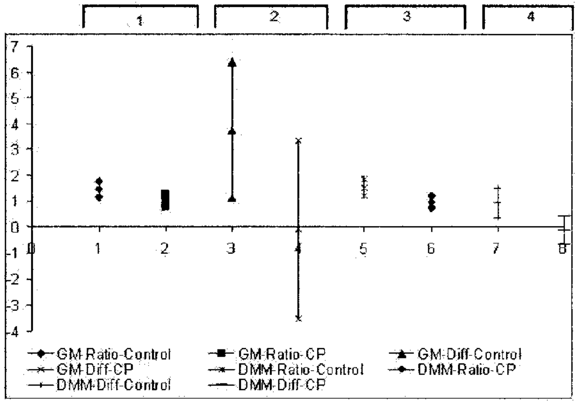

Figure 5: Graph indicating the mean ± the standard deviation. After processing with DMM, the STDV for the ratio of the white matter to the choroid plexus for the control and CP groups do not overlap (Pair 4).

Mountain Bioengineering Symposium, Copper Mountain Colorado, USA, 2002, 369-374.

[11] G. Stippel, W. Philips and I. Lemahieu,. " A new denoising technique for ultrasound images using morphological properties of speckle combined with tissue classifying parameters." *Proceedings of SPIE/Medical Imaging 2002 Ultrasonic Imaging and Signal Processing*, San Diego, USA, February 2002, 324-333.

[12] E. Davies. "Image distortions produced by mean, median and mode filters." *IEE Proceedings - Vision, Image and Signal Processing*, 146(5),1999,279-285.

Ultrasonic Tissue Characterization as a Predictor of White Matter Damage: Results of a Preliminary Study.

Tyna Hope*, Peter Gregson*, Norma Linney† and Matthias Schmidt‡
* Department of Electrical and Computer Engineering
Dalhousie University,PO Box 1000, Halifax, Ca. B3J 2X4
Email: tahope@dal.ca
†Saint Mary's University, 923 Robie Street, Halifax, Ca. B3H 3C3
‡IWK Health Centre, PO Box 3070, Halifax, Ca. B3J 3G9

Abstract—**Premature infants are prone to white matter damage (WMD), which is associated with cerebral palsy (CP) and cognitive impairment. Ultrasound (US) is the preferred imaging modality to detect WMD. To improve on existing diagnostic rates, quantitative measures incorporating new information are needed. We are investigating US texture measures as new indicators of white matter health.**

We have developed algorithms to enhance texture features and then obtain a measure of the tissue texture. Using our texture measures, data from 18 patients (12 with normal outcome, 6 who developed CP) form separate populations based on patient outcome. Our algorithms are applied to B-mode cranial US images without compensating for operator-dependent machine settings and without suppressing speckle. The results of the preliminary study will form the basis for the design of a computer aided diagnosis system for the early detection of white matter damage.

I. Introduction

The purpose of our work is to detect WMD in the cranial US images of premature infants more reliably and earlier than is currently possible. Premature infants are prone to WMD which is associated with subsequent CP and cognitive impairment [1]. Ultrasound is commonly used to screen for WMD. Although MRI detects WMD with greater sensitivity [2], US remains the imaging modality of choice because it is portable, it requires no sedation, and it is more readily available. Currently, diagnosis with US is qualitative. Radiologists depend on echolucencies and echodensities as cues to the presence of WMD [3]. Since US intensity information alone is not satisfactory for diagnosing WMD [2] [4], other measures are required.

Texture information is an additional measure that is explored in this work. A previous study indicates that texture may be useful for the detection of cerebral hypoxic insult in term infants [5]. We are investigating texture as a predictor of periventricular leukomalacia in preterm infants. In an earlier set of experiments, we found that tissue characterization of white matter within cranial US images is possible by enhancing texture characteristics and then obtaining a quantitative measure over a single tissue sample [6]. We are continuing from our previously reported work by expanding upon the preprocessing techniques and adding an additional quantitative texture descriptor.

II. Background

We use cranial US images without preprocessing to remove speckle. Structures much larger than the US wavelength result in specular reflection, while those much smaller result in fully developed speckle. Reflections due to microstructures may be correlated with pathology [7] and so speckle has been included in the analysis.

Ultrasound image texture has been investigated for disease detection in many soft tissues. Diseases of the liver have received a significant amount of attention [8],[9],[10]. While there are non-trivial differences between the neonatal brain and liver, both are soft tissues, and in both organs the disease process disrupts cell function and tissue architecture. More importantly, previous work by others [5] indicates that US texture may indicate the occurrence of hypoxic insult in term infants. Although premature and term infant brains differ at the histological level, this work suggests that further investigation of the application of US texture and brain injury in premature infants is warranted.

Stippel et. al. [11] have investigated speckle classification and suppression techniques as part of an algorithm to help radiologists assess white matter damage using flare (areas of increased echogenicity) symmetry. Symmetry of white matter flares is a useful feature for assessing the extent of white matter damage. We are developing a classification scheme independent of flare symmetry that incorporates speckle. Our texture measures are independent of the presence of visually obvious flares and thus may allow earlier diagnosis than is presently available.

III. Theoretical Basis

Our goal is to describe texture characteristics, including speckle. Speckle is not bandlimited but it is not white [12] suggesting that band-specific processing schemes are appropriate. We chose the Gabor filter due to its desirable band-pass

0-7803-8412-1/04/$20.00 (c)2004 IEEE.

properties of compact support in both the frequency and spatial domains. We truncate the filter for implementation. The Gabor filter has some intuitive template matching qualities with US image texture which adds to its appeal in this application.

We chose to apply the Gabor filter oriented along the theorectical path of insonation. Speckle characteristics in the lateral orientation are dependent on the distance to the transducer [7] so the information about speckle properties can be detected in the axial direction (direction of the pulses). In our application the transducers are curvilinear. We select the orientation of the Gabor kernel by locating a hypothetical insonation point and then orienting the Gabor filter so that the medial axis follows the line of insonation, identical to the mask orientation as shown in Figure 1. While the pulses do not come from a point source, the error caused by this approximation was not found to be significant. Similar filtering was performed by Murthy et al. but they applied Gaussian band-pass filters to A-mode data and then concatenated the result to form the final image, rather than applying it to B-mode image data [13].

To realize the Gabor kernel, we created a 2-D Gaussian and multiplied it with an image of a cosine. The Gaussian image was created to be 200x21 pixels, with the Gaussian radii of 100 and 10 pixels and having a maximum intensity at the center of 10 (out of a possible 256 grey levels). The cosine image was chosen to have a peak value of 10 with the first zero crossing located at 0,0 in the image. The period (T) of the cosine determines the center frequency of the bandpass filter. We examined some samples of white matter and choroid plexus in the frequency domain to identify the frequencies that appear to be the most characteristic of each tissue texture. Four frequencies were noted, however only two were pursued further. The two discarded frequencies are the same frequencies as the dominant noise frequencies in the US film images, which caused numerical overflow. In the end, cosine images having T=3 and T=7 were used to realize the Gabor filters and applied to the cranial US images.

An additional texture measure reported here is the normalized surface area (NSA). We obtained the surface area of the tissue samples (with the intensity of the image considered to be hills and valleys) and then normalized it to the pixel area of the tissue mask to account for the varying tissue mask sizes. Image intensity roughness, as a texture measure, has been applied to B-mode US images in the past for the detection of liver disease [14]. We obtained the texture measures from the Gabor filtered images and the images after processing with two non-linear techniques. One non-linear technique is based upon histogram characteristics (DMM) and the other is based on grey-level morphology (GM) (Figure 2). The non-linear techniques are explained in [6].

IV. Experimental methods

- The images used in this study are of premature infants who had cranial US scans between 1990 and 1991 at the IWK Health Centre in Halifax, Canada. The patients were identified through the Perinatal Follow-up Database. To be included in this study, the patients had to be less than 31 weeks gestational age, and the final outcome of the patient had to be known. Eighteen patients were included in this research. None of the images had unequivocal evidence of periventricular leukomalacia (i.e. cystic periventricular echolucencies). The group contained six patients who subsequently developed CP.

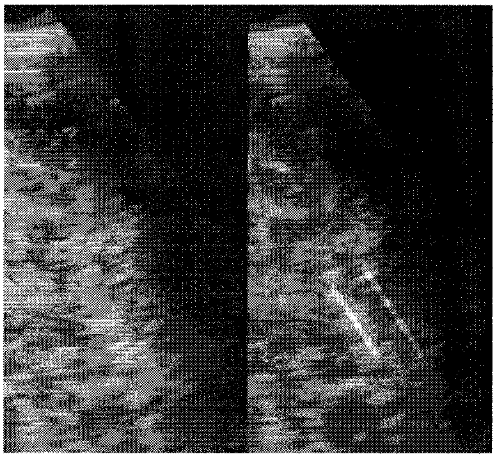

Fig. 1. Left side of a patient's cranial sonogram. Right image demonstrates the location of the choroid plexus and white matter masks.

The sample size is small due to a number of factors. The input data are digitized analogue US films since modern digital US machines have greater bandwidth restrictions than the analogue film images. This limits our data to US films from 1990 and 1991. The population from which the images could be drawn is small, and some eligible patients from this period had follow-up care elsewhere, making their records inaccessible. The images were scanned with a 3200 dpi scanner (Epson Perfection 3200) to maintain the high resolution of the images. The effect of reducing spatial resolution on the information content of texture measures has yet to be determined.

For each patient, a single semi-coronal image at the level of the atria of the lateral ventricles was chosen. The data set contained images obtained with a variety of machines and various operator-dependent settings.

- Due to the large size of the cranial US images, they were first divided into halves, representing the left and right hemispheres. Therefore, each patient in this study is associated with two half US images. Samples (masks) of two tissue types, choroid plexus and white matter, were selected from each patient's half image by a radiologist, where it was possible to select the white matter with a high degree of certainty. For all but one patient, images for both sides of the patient's brain were used. The masks are rectangular in shape with their major axis oriented approximately along the path of insonation (Figure 1).

The choroid plexus was chosen as the reference tissue type as it is the traditional standard used to compare white matter echogenicity.

- The images were processed to enhance the differences in texture between the two tissue types. The technique presented here in detail is a variably-oriented Gabor filter (vGa). No preprocessing was performed to compensate for the operator-dependent machine settings. Speckle was not suppressed as we are exploring speckle as a component of texture.
- After processing, a measure of the image intensity roughness (NSA) and the standard deviation of the intensity (STAT) were calculated within the choroid plexus and white matter masks for each patient. For a typical patient, four NSA values were calculated after being processed with vGa. Two values were obtained from the white matter, one from the left side and one from the right side. Likewise, the left and right NSA values were calculated for the choroid plexus regions. Ratios and differences of white matter and choroid plexus were obtained and the left and right halves were averaged. For example, each patient would have one average NSA ratio, $NSA_{ratio} = 1/2 * (NSA_{w_{left}}/NSA_{choroid_{left}} + NSA_{w_{right}}/NSA_{choroid_{right}})$. Similarly, the NSA difference was calculated, as well as the STAT ratio and difference for each patient image.
- Two tailed Student t-tests were performed on the ratio and difference measures to determine if two populations exist corresponding to the two patient outcomes.

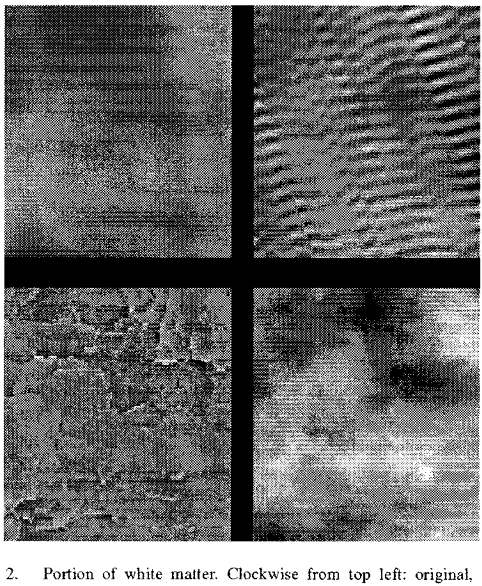

Fig. 2. Portion of white matter. Clockwise from top left: original, after variable gabor filtering, after applying DMM, after applying GM.

V. Results and Discussion

The texture measures for the control and CP groups were assessed to determine if a difference existed between them. In this work, we are reporting on 12 different experiments applied to 18 patient images. The experiments include: vGa T=7 NSA_{ratio}, vGa T=7 NSA_{diff},vGa T=7 $STAT_{ratio}$,vGa T=7 $STAT_{diff}$, vGa T=3 NSA_{ratio}, vGa T=3 NSA_{diff},vGa T=3 $STAT_{ratio}$,vGa T=3 $STAT_{diff}$, DMM NSA_{ratio}, DMM NSA_{diff}, GM NSA_{ratio}, and GM NSA_{diff}. For 7 of the 12 experiments, descriptive statistics indicate that the means of the texture measures differ based upon the patient outcome. Table I provides the mean ± twice the standard error for these seven experiments.

The two sample t-test was performed on the texture measures for each of the experiments to determine if two populations exist based on patient outcome (Table II). For the t-test, it is assumed that the unknown variances are not equal. A cutoff of $\alpha = 0.05$ was used. The null hypothesis (H_0) is that the means of the two groups are equal. In 9 of 12 cases, the test statistic was greater than t and thus H_0 was rejected. The P value is provided as an additional measure of the significance of the experimental results. While the student t-test is not the most robust test for this data, since biological data typically do not follow normal distributions, it provided a simple measure to determine which if any of our texture measures shuld be investigated further for this problem. The results of these experiments indicate that although individual texture measures do not always separate well into two populations, more than one measure may yield better results (Figure 3).

In these experiments, the images are used without compensating for operator dependent settings and without removing speckle. The favorable results suggest that using the patient image as its own control and including speckle in the data does not prevent the image data from revealing patient outcome. This means less image processing is required.

VI. Conclusions

Texture measures have been obtained without prior compensation for operator-based machine settings and without using speckle-reduction techniques. Statistical analysis of these texture measures indicate that two populations exist corresponding to the patient outcome. The encouraging results in this preliminary study suggest that that additional studies with a larger population are warranted. The successful discovery of additional diagnostic measures may provide a tool for earlier medical intervention.

Acknowledgments

The authors would like to thank William Phillips of the Engineering Mathematics Department, Dalhousie University, for his assistance in the analysis of the data. We would also like to thank Michael Vincer, the Director of the IWK Perinatal Follow-up Clinic and Dora Stinson, Medical Director of the Special Care Nursery, IWK for their support. Financial support for this work has been provided by the Natural Science and

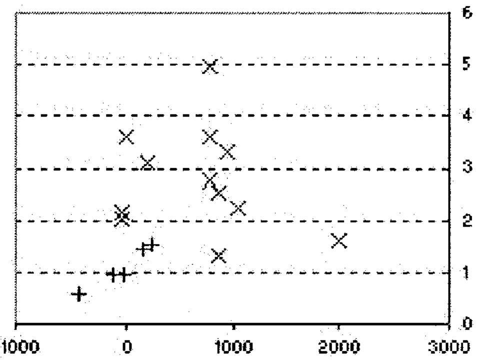

Fig. 3. The STAT ratio value vs the NSA difference for all patient images after processing with vGa t=7. + CP Group, X Normal Outcome Group.

Method, measure	Group	lower bound	upper bound
Gabor T=7, NSA_{diff}	Normal	351	1023
Gabor T=7, NSA_{diff}	CP	-231	149
Gabor T=7, $STAT_{ratio}$	Normal	2.190	3.350
Gabor T=7, $STAT_{ratio}$	CP	0.780	1.370
Gabor T=3, NSA_{ratio}	Normal	1.190	1.730
Gabor T=3, NSA_{ratio}	CP	1.020	1.140
GM, NSA_{ratio}	Normal	1.350	1.800
GM, NSA_{ratio}	CP	0.954	1.150
GM, NSA_{diff}	Normal	1.840	4.050
GM, NSA_{diff}	CP	-0.350	0.990
DMM, NSA_{ratio}	Normal	1.440	1.860
DMM, NSA_{ratio}	CP	0.740	1.330
DMM, NSA_{diff}	Normal	1.580	3.10
DMM, NSA_{diff}	CP	-0.450	0.650

TABLE I

CONFIDENCE OF THE SEPARATE MEANS, AS ILLUSTRATED BY $mean \pm 2 * standarderror$.

Engineering Research Council of Canada, a Category B grant of the IWK Health Centre and an IWK Auxiliary Award.

REFERENCES

[1] J. J. Volpe. "Neurobiology of periventricular leukomalacia in the premature infant." *Pediactric Research*, 50(4),2001,553-62.

Method, measure	t critical	df	t	P
Gabor T=7, NSA_{diff}	2.120	16	3.770	0.002
Gabor T=7, $STAT_{ratio}$	2.130	15	5.180	0.0001
Gabor T=7, $STAT_{diff}$	2.205	11	3.410	0.006
Gabor T=3, NSA_{ratio}	2.18	12	2.78	0.017
Gabor T=3, NSA_{diff}	2.13	15	2.37	0.03
GM, NSA_{ratio}	2.140	14	4.230	0.0008
GM, NSA_{diff}	2.120	16	4.060	0.0009
DMM, NSA_{ratio}	2.180	12	3.880	0.002
DMM, NSA_{diff}	2.245	10	3.360	0.008

TABLE II

RESULTS OF THE T-TESTS FOR THE EXPERIMENTS IN WHICH THE NULL HYPOTHESIS, EQUAL POPULATION MEANS, WAS REJECTED.

[2] E. F. Maalouf, P. J. Duggan, S. J. Counsell, M. A. Rutherford, F. Cowan, D. Azzopardi and A. D. Edwards. "Comparison of Findings on Cranial Ultrasound and Magnetic Resonance Imaging in Preterm Infants." *Pediatrics*, 107(4), April 2001,719-727.

[3] L. S. de Vries, P. Eken, L.M. Dubowitz. "The spectrum of leukomalacia using cranial ultrasound." *Behavioural Brain Research*, 49(1), 1992, 1-6.

[4] E. E. Holling."Characteristics of cranial ultrasound white-matter echolucencies that predict disability: a review." *Developmental Medicine and Child Neurology*, 41,1999,136-139.

[5] L. L.Barr, P. J. McCullough , W. S. Ball, B. H. Krasner, B. S. Garra and J. A. Deddens. " Quantitative Sonographic Feature Analysis of Clinical Infant Hypoxia: A Pilot Study." *American Journal of Neuroradiology*, 17, June 1996, 1025-1031.

[6] T. Hope, P. Gregson, N. Linney, M. Schmidt. "Texture-based Tissue Characterization: A Novel Predictor for Brain Injury?" *2nd IASTED International Conference on Biomedical Engineering*, 2004, ACTA Press, pp.

[7] J. Thijssen. "Ultrasonic speckle formation, analysis and processing applied to tissue characterization." *Pattern Recognition Letters*, 24, Feb. 2003, 659-675.

[8] P. C. Hartman, B. J. Oosterveld, J. M. Thijssen and G.J.E. Rosenbusch. "Variability of Quantitative Echographic Parameters of the Liver : Inntra- and Interindividual Spread, Temporal and Age-related effects." *Ultrasound in Medicine and Biology*, 17(9), 1991, 857-867.

[9] Y. M.Kadah, A.A. Faraq, J. M. Zurada, A.M. Badawi and A.M. Youssef. "Classification Algorithms for Quantitative Tissue Characterization of Diffuse Liver Disease from Ultrasound Images." *IEEE Transactions of Medical Imaging*, 15(4), August 1996, 466-478.

[10] M. H.Loew, R. Mia and Z. Guo. " An Approach to image Classification in Ultrasound." *Proceedings of the 29th Applied Imagery Pattern Recognition Workshop*, Washington, D.C., USA, 2000, 193-199.

[11] G. Stippel, W. Philips and I. Lemahieu,. " A new denoising technique for ultrasound images using morphological properties of speckle combined with tissue classifying parameters." *Proceedings of SPIE/Medical Imaging 2002 Ultrasonic Imaging and Signal Processing*, San Diego, USA, February 2002, 324-333.

[12] K.H. Uslu, N. M. Bilgutay, R. Murthy and K. Kaya. " Medical Image Enhancement using Split Spectrum Processing." *1993 Ultrasonics Symposium*, 1993.

[13] R. Murthy and N. M. Bilgutay. "Scaling Techniques for Medical Image Enhancement. " *1995 IEEE Ultrasonics Symposium*, 1995.

[14] I. Akiyama, T. Saito, M. Nakamura, N. Taniguchi and K. Itch. " Tissue Characterization by Using Fractal Dimension of B-Scan Image." *Ultrasonics Symposium*, 1990, pp 1353-1355.

Appendix C

Resolution Reduction

STDEV as measure for the rectangular samples

Data taken from book 3

	rt amt	lft amt	rt bbc	lft bbc	rt bc	lft bc	rt bgv001	lft bgv001	rt dev	lft dev
r1 w	14.37	15.59	15.1	14.79	8.78	9.84	15.82	11.48	11.82	12.28
r1 c	5.68	12.14	9.18	8.29	8.56	6.6	17.81	13.35	11.44	9.56
r2 w	12.61	14.2	13.35	13.19	7.99	8.91	14.54	10.53	10.86	11.1
r2 c	4.95	10.98	8.46	7.5	7.88	6.04	16.33	12.4	10.14	8.89
r3 w	11.38	14.44	12.56	12.42	7.89	8.75	14.57	10.38	10.5	10.85
r3 c	4.83	10.53	7.97	7.04	7.65	5.75	16.15	12.34	9.36	8.69
diff r1	8.69	3.45	5.92	6.5	0.22	3.24	-1.99	-1.87	0.38	2.72
diff r2	7.66	3.22	4.89	5.69	0.11	2.87	-1.79	-1.87	0.72	2.21
diff r3	6.55	3.91	4.59	5.38	0.24	3	-1.58	-1.96	1.14	2.16
ratio r1	2.529929577	1.284184514	1.644880174	1.784077201	1.025700935	1.490909	0.888265	0.859925094	1.033217	1.284519
ratio r2	2.547474747	1.293260474	1.578014184	1.758666667	1.013959391	1.475166	0.890386	0.849193548	1.071006	1.248594
ratio r3	2.35610766	1.371320038	1.575909661	1.764204545	1.031372549	1.521739	0.902167	0.841166937	1.121795	1.248562
average diff r1		6.07		6.21		1.73		-1.93		1.55
ave diff r2		5.44		5.29		1.49		-1.83		1.465
ave diff r3		5.23		4.985		1.62		-1.77		1.65
ave ratio r1		1.907057046		1.714478688		1.258305		0.874095057		1.158868
ave ratio r2		1.920367611		1.668340426		1.244562		0.869789671		1.1598
ave ratio r3		1.863713849		1.670057103		1.276556		0.87166706		1.185178

r1 differences		r1 differences		R2 diff		r2 diff		r3 diff		r3 diff
Control		CP		Normal		CP		Normal		
Mean	3.633333333	Mean	1.212	Mean	3.182083333	Mean	0.945	Mean	3.107917	Mean
Standard Error	0.741272537	Standard Error	0.443890752	Standard Error	0.664643177	Standard E	0.322106	Standard Error	0.67013	Standard E
Median	4.025	Median	1.315	Median	3.5775	Median	1.14	Median	3.54	Median
Mode	#N/A	Mode	1.315	Mode	#N/A	Mode	#N/A	Mode	#N/A	Mode
Standard Deviation	2.567843394	Standard Deviation	0.992569897	Standard Deviation	2.302391502	Standard D	0.720252	Standard Devial	2.321399	Standard D
Sample Variance	6.593819697	Sample Variance	0.985195	Sample Variance	5.301006629	Sample Va	0.518763	Sample Varianc	5.388893	Sample Va
Kurtosis	0.291472465	Kurtosis	3.046122239	Kurtosis	0.385893395	Kurtosis	3.568266	Kurtosis	0.090512	Kurtosis
Skewness	-0.892708821	Skewness	-1.645782145	Skewness	-0.879186757	Skewness	-1.77833	Skewness	-0.79785	Skewness
Range	8.3	Range	2.525	Range	7.725	Range	1.855	Range	7.695	Range
Minimum	-1.93	Minimum	-0.465	Minimum	-1.83	Minimum	-0.29	Minimum	-1.77	Minimum
Maximum	6.37	Maximum	2.06	Maximum	5.895	Maximum	1.565	Maximum	5.925	Maximum
Sum	43.6	Sum	6.06	Sum	38.185	Sum	4.725	Sum	37.295	Sum
Count	12	Count	5	Count	12	Count	5	Count	12	Count
mean + 2 std err	5.115878408		2.099781505		4.511369687		1.589213		4.448177	
mean -2 std err	2.150788258		0.324218495		1.85279698		0.300787		1.767656	
overlap (lowest normal - highest of CP)		-0.051006754				-0.26358				0.28705
mean + 2 stdev	8.769020122		3.197139794		7.786866337		2.385503		7.750715	
mean - 2 stdev	-1.502353455		-0.773139794		-1.422699671		-0.4955		-1.53488	
		4.699493249				3.808203				4.6473

rt	lft												
epat	epat	mec	mec	rm	rm	sad	sad	sdc	sdc	sm2	sm2	tap	tap
14.2	12.21	11.07	10.76	13.11	13.34	12.88	11.92	16.55		11.91	14.63	14.54	13.89
14.34	9.85	7.27	7.28	8.96	8.76	6.82	8.08	10.7		10.67	8.5	9.1	6.59
12.33	11.08	10.19	9.83	11.82	12	11.76	10.71	14.9		10.77	12.81	13.49	12.7
12.61	8.97	6.68	6.58	8.28	7.99	6.26	7.39	9.88		9.85	7.86	8.35	6.05
11.1	10.33	10.04	9.74	11.49	11.72	11.48	10.24	14.79		10.49	11.47	13.17	12.2
12.15	8.61	6.49	6.59	7.96	7.79	5.88	7.18	9.4		9.39	7.53	7.83	5.69
-0.14	2.36	3.8	3.48	4.15	4.58	6.06	3.84	5.85		1.24	6.13	5.44	7.3
-0.28	2.11	3.51	3.25	3.54	4.01	5.5	3.32	5.02		0.92	4.95	5.14	6.65
-1.05	1.72	3.55	3.15	3.53	3.93	5.6	3.06	5.39		1.1	3.94	5.34	6.51
0.990237099	1.239594	1.522696	1.478022	1.46317	1.522831	1.888563	1.475247525	1.546729		1.116213683	1.721176	1.597802198	2.107739
0.9777954	1.235229	1.525449	1.493921	1.427536	1.501877	1.878594	1.449255751	1.508097		1.093401015	1.629771	1.615568862	2.099174
0.913580247	1.199768	1.546995	1.477997	1.443467	1.504493	1.952381	1.426183844	1.573404		1.1171459	1.52324	1.681992337	2.144112
	1.11		3.64		4.365		4.95		5.85		3.685		6.37
	0.915		3.38		3.775		4.41		5.02		2.935		5.895
	0.335		3.35		3.73		4.33		5.39		2.52		5.925
	1.114916		1.500359		1.493		1.681905287		1.546729		1.418695		1.852771
	1.106512		1.509685		1.464707		1.663925		1.508097		1.361586		1.857371
	1.056674		1.512496		1.47398		1.689282398		1.573404		1.320193		1.913052

	r1 ratios		r1 ratios		r2 ratios		r2 ratios		r3 ratios		r3 ratios	
cp	Normal			cp	normal			CP	Normal			CP
1.199	Mean	1.460098	Mean	1.117951	Mean	1.444562	Mean	1.10054	Mean	1.450521213	Mean	1.132853981
0.427853363	Standard E	0.090194	Standard E	0.039443	Standard E	0.090298	Standard Error	0.031302	Standard E	0.091998092	Standard E	0.043324891
1.49	Median	1.49668	Median	1.13894	Median	1.486402	Median	1.129266	Median	1.493238155	Median	1.163853987
#N/A	Mode	#N/A	Mode	#N/A	Mode	#N/A	Mode	#N/A	Mode	#N/A	Mode	#N/A
0.956709203	Standard D	0.31244	Standard D	0.088197	Standard D	0.312802	Standard Deviation	0.069994	Standard D	0.31869074	Standard D	0.096877401
0.9152925	Sample Va	0.097619	Sample Va	0.007779	Sample Va	0.097845	Sample Variance	0.004899	Sample Va	0.101563788	Sample Va	0.009385231
4.342778396	Kurtosis	-0.52169	Kurtosis	2.497187	Kurtosis	-0.48517	Kurtosis	3.64127	Kurtosis	-0.62329018	Kurtosis	3.721075427
-2.016276997	Skewness	-0.35776	Skewness	-1.54772	Skewness	-0.21318	Skewness	-1.85631	Skewness	-0.28788769	Skewness	-1.77954569
2.415	Range	1.032962	Range	0.219883	Range	1.050578	Range	0.175893	Range	1.041385348	Range	0.253231169
-0.48	Minimum	0.874095	Minimum	0.971036	Minimum	0.86979	Minimum	0.979838	Minimum	0.87166706	Minimum	0.966442598
1.935	Maximum	1.907057	Maximum	1.190919	Maximum	1.920368	Maximum	1.155732	Maximum	1.913052408	Maximum	1.219673767
5.995	Sum	17.52118	Sum	5.589755	Sum	17.33474	Sum	5.502698	Sum	17.40625456	Sum	5.664269905
5	Count	12	Count	5	Count	12	Count	5	Count	12	Count	5
2.054706725		1.640485		1.196837		1.625158		1.163144		1.634517398		1.219503762
0.343293275		1.279711		1.039065		1.263965		1.037935		1.266525028		1.0462042
			-0.08287				-0.10082147				-0.04702	
3.112418407		2.084978		1.294345		2.070167		1.240527		2.087902694		1.326608782
-0.714418407		0.835219		0.941557		0.818957		0.960552		0.813139732		0.93909918
			0.459127				0.421570037				0.513469	

bcv001	bcv001	dm001	dm001	kls001	kls001	tw001	tw001	vam002	vam002
15.54	15.64	14.93	16.66	12.97	13.42	15.99	15.32	14.09	13.1
16.01	16.1	10.14	18.82	10.25	12.02	12.94	14.7	11.83	12.73
13.99	14.1	13.4	15.15	11.76	11.4	14.56	14.11	12.93	11.9
14.43	14.24	9.39	17.13	9.47	11.1	11.88	13.66	10.94	11.61
13.76	13.84	13.67	14.68	11.71	10.53	14.38	13.98	12.88	11.64
14.37	14.19	8.8	16.57	8.87	10.4	11.46	13.03	10.48	10.91
-0.47	-0.46	4.79	-2.16	2.72	1.4	3.05	0.62	2.26	0.37
-0.44	-0.14	4.01	-1.98	2.29	0.3	2.68	0.45	1.99	0.29
-0.61	-0.35	4.87	-1.89	2.84	0.13	2.92	0.95	2.4	0.73
0.970643	0.971429	1.472387	0.885228	1.265366	1.116473	1.235703	1.042177	1.19104	1.029065
0.969508	0.990169	1.42705	0.884413	1.241816	1.027027	1.225589	1.032943	1.181901	1.024978
0.95755	0.975335	1.553409	0.885938	1.32018	1.0125	1.254799	1.072909	1.229008	1.066911
	-0.465		1.315		2.06		1.835		1.315
	-0.29		1.015		1.295		1.565		1.14
	-0.48		1.49		1.485		1.935		1.565
	0.971036		1.178808		1.190919		1.13894		1.110052
	0.979838		1.155732		1.134422		1.129266		1.10344
	0.966443		1.219674		1.16634		1.163854		1.147959

Appendix D

Frequency Spectra

The area under the curve up to 32 was calculated and compared to the entire area under the curve.
The purpose is to determine if the spectrum for all 17 images of EXP1 are roughly equal for the portion that exists after twice resolution reduction.

frequency	log frequency	amtrtw256.ascii	area amt	bbcrtw256.ascii	area bbc	bcrtw256.ascii	area bc	bgv001rtw256.acsii	area bgv	devrtw256.ascii	area dev	epatrtw256.ascii	area epat	mecrtw256.ascii	area mec
128	2.10720997	3	0.008515622	2	0.005109373	0	0.003406249	5	0.017031243	3	0.01192187	1	0.005109373	5	0.015328119
127	2.103803721	2	0.00858294	1	0.003433176	2	0.005149764	5	0.015449291	4	0.013732703	2	0.006866352	4	0.015449291
126	2.100370545	3	0.00865133	1	0.003460532	1	0.00865133	4	0.013842128	4	0.015572394	2	0.006921064	5	0.013842128
125	2.096910013	2	0.00872082	1	0.003488328	4	0.00872082	4	0.013953311	5	0.017441639	2	0.00872082	3	0.010464984
124	2.093421685	3	0.010549721	1	0.001758287	1	0.005274861	4	0.014066295	5	0.015824582	3	0.012308008	3	0.012308008
123	2.089905111	3	0.010635842	0	0.003545281	2	0.007090562	4	0.014181123	4	0.015953763	4	0.008863202	4	0.014181123
122	2.086359831	3	0.012510611	2	0.005361691	2	0.007148921	4	0.014297841	5	0.017872302	1	0.005361691	4	0.014297841
121	2.08278537	4	0.014416497	1	0.005406186	2	0.005406186	4	0.016218559	5	0.016218559	2	0.005406186	4	0.014416497
120	2.079181246	4	0.014537139	2	0.003634285	1	0.005451427	5	0.018171423	4	0.016354281	1	0.005451427	4	0.016354281
119	2.075546961	4	0.012827339	0	0.001832477	2	0.005497431	5	0.016492293	5	0.01832477	2	0.007329908	5	0.016492293
118	2.071882007	3	0.009240364	1	0.001848073	1	0.005544218	4	0.014784582	5	0.016632655	2	0.005544218	4	0.012936509
117	2.068185862	2	0.007455745	0	0	2	0.005591809	4	0.016775426	4	0.01491149	1	0.005591809	3	0.013047554
116	2.064457989	2	0.007520298	0	0.003760149	1	0.005640223	5	0.01692067	4	0.015040595	2	0.005640223	4	0.015040595
115	2.06069784	2	0.007585978	2	0.005689484	2	0.007585978	4	0.015171956	4	0.015171956	1	0.007585978	4	0.015171956
114	2.056904851	2	0.00956602	1	0.005739612	2	0.007652816	4	0.015305631	4	0.015305631	3	0.00956602	4	0.015305631
113	2.053078443	3	0.011581262	2	0.005790631	2	0.007720842	4	0.017371894	4	0.017371894	2	0.009651052	4	0.015441683
112	2.049218023	3	0.013632654	1	0.005842566	2	0.007790088	5	0.021422741	5	0.019475219	3	0.007790088	4	0.015580176
111	2.045322979	4	0.013756028	2	0.00589544	2	0.007860587	6	0.019651468	5	0.019651468	1	0.003930294	4	0.013756028
110	2.041392685	3	0.009915468	1	0.005949281	2	0.007932374	4	0.017847842	5	0.017847842	1	0.005949281	3	0.015864749
109	2.037426498	2	0.008005485	2	0.004002742	2	0.010006856	5	0.018012341	4	0.018012341	2	0.006004114	5	0.018012341
108	2.033423755	2	0.012119933	0	0.002019989	3	0.010099945	4	0.016159911	5	0.0181799	1	0.008079956	4	0.016159911
107	2.029383778	4	0.012233737	1	0.006116869	2	0.006116869	4	0.01631165	4	0.01631165	3	0.008155825	4	0.01631165
106	2.025305865	2	0.012349699	2	0.008233132	1	0.006174849	4	0.016466265	4	0.018524548	1	0.004116566	4	0.016466265
105	2.021189299	4	0.014545859	2	0.00623394	2	0.010389899	4	0.016623839	5	0.018701819	1	0.00623394	4	0.016623839
104	2.017033339	3	0.010490286	1	0.004196115	3	0.010490286	4	0.014686401	4	0.016784458	2	0.004196115	4	0.018882516
103	2.012837225	2	0.012711159	1	0.006355579	2	0.010592632	3	0.016948212	4	0.019066738	0	0.002118526	5	0.021185265
102	2.008600172	4	0.014975793	2	0.006418197	3	0.010696995	5	0.019254591	5	0.02139399	1	0.006418197	5	0.02139399
101	2.004321374	3	0.010803434	1	0.004321374	2	0.008642748	4	0.017285495	5	0.019446182	2	0.008642748	5	0.019446182
100	2	2	0.010912014	1	0.004364805	2	0.008729611	4	0.019641624	4	0.019641624	2	0.008729611	4	0.017459222
99	1.995635195	3	0.013227357	1	0.006613678	2	0.008818238	5	0.022045595	5	0.019841035	2	0.008818238	4	0.019841035
98	1.991226076	3	0.013363024	2	0.004454341	2	0.008908683	5	0.022271707	4	0.020044536	2	0.008908683	5	0.022271707
97	1.986771734	3	0.011251253	0	0.004500501	2	0.009001002	5	0.024752757	5	0.022502506	2	0.009001002	5	0.020252256
96	1.982271233	2	0.011369069	2	0.009095256	2	0.011369069	6	0.022738139	5	0.020464325	2	0.011369069	4	0.020464325
95	1.977723605	3	0.016085131	2	0.006893628	3	0.011489379	4	0.018383007	4	0.020680883	3	0.011489379	5	0.025276634
94	1.973127854	4	0.016257168	1	0.004644905	2	0.00928981	4	0.020902073	5	0.020902073	2	0.006967358	6	0.023224525
93	1.968482949	3	0.011737803	1	0.004695121	2	0.011737803	5	0.021128045	4	0.021128045	1	0.009390242	4	0.021128045
92	1.963787827	2	0.00949287	1	0.004746435	3	0.011866088	4	0.021358958	5	0.023732175	3	0.014239305	5	0.023732175
91	1.959041392	2	0.014396649	1	0.007198324	2	0.009597766	5	0.023994414	5	0.021594973	3	0.011997207	5	0.023994414
90	1.954242509	4	0.01698376	2	0.007278754	2	0.009705006	5	0.024262514	4	0.019410011	2	0.007278754	5	0.026688765
89	1.949390007	3	0.017175671	1	0.007361002	2	0.012268336	5	0.02699034	4	0.019629338	1	0.007361002	6	0.022083005
88	1.944482672	4	0.017371968	2	0.009926839	3	0.012408549	6	0.024817098	4	0.017371968	2	0.009926839	3	0.019853678
87	1.939519253	3	0.017572805	2	0.007531202	2	0.010041603	4	0.022593606	3	0.020083206	2	0.007531202	5	0.025104007
86	1.934498451	4	0.020318102	1	0.007619288	2	0.007619288	5	0.025397628	5	0.025397628	1	0.005079526	5	0.025397628
85	1.929418926	4	0.020558559	2	0.010279279	1	0.007709459	5	0.025698198	5	0.025698198	1	0.010279279	5	0.025698198
84	1.924279286	4	0.020804775	2	0.007801791	2	0.010402387	5	0.026005968	5	0.023405372	3	0.013002984	5	0.028606565
83	1.919078092	4	0.01579272	1	0.01052848	2	0.0131606	5	0.02368908	4	0.02368908	2	0.00789636	6	0.0263212
82	1.913813852	2	0.010657667	3	0.010657667	3	0.013322084	4	0.023979751	5	0.029308584	1	0.00799325	4	0.026644168
81	1.908485019	2	0.010790064	1	0.005395032	2	0.010790064	5	0.029672675	6	0.029672675	2	0.01348758	6	0.026975159
80	1.903089987	2	0.021851583	1	0.002731448	2	0.013657239	6	0.030045926	5	0.030045926	3	0.016388687	4	0.024583031
79	1.897627091	6	0.024896199	0	0.002766244	3	0.016597466	5	0.030428687	6	0.030428687	3	0.013831221	5	0.033194932
78	1.892094603	3	0.033623265	1	0.011207755	3	0.011207755	6	0.030821326	5	0.030821326	2	0.02241551	7	0.036425204
77	1.886490725	9	0.034062797	3	0.011354266	1	0.011354266	5	0.031224231	6	0.034062797	6	0.025547098	6	0.034062797
76	1.880813592	3	0.008628493	1	0.005752329	3	0.020133151	6	0.034513973	6	0.028761644	3	0.011504658	6	0.037390138
75	1.875061263	0	0.008744315	1	0.005829544	4	0.014573859	6	0.03206249	4	0.03206249	1	0.014573859	7	0.040806806
74	1.86923172	3	0.023635438	1	0.008863289	1	0.011817719	5	0.035453158	7	0.035453158	4	0.014772149	7	0.038407587
73	1.86332286	5	0.026956637	2	0.008985546	3	0.017971091	7	0.029951818	5	0.032947	1	0.005990364	6	0.032947
72	1.857332496	4	0.018222443	1	0.009111222	3	0.018222443	3	0.030370739	6	0.027333665	1	0.021259517	5	0.033407812
71	1.851258349	2	0.027721389	2	0.012320617	3	0.015400772	7	0.036961852	3	0.027721389	6	0.027721389	6	0.033881698
70	1.84509804	7	0.034369221	2	0.012497899	2	0.012497899	5	0.037493696	6	0.037493696	3	0.018746848	5	0.034369221
69	1.838849091	4	0.019020534	2	0.009510267	2	0.012680356	7	0.041211157	6	0.041211157	3	0.028530801	6	0.034870979
68	1.832508913	2	0.009651165	1	0.022519385	2	0.016085275	6	0.03860466	7	0.035387605	6	0.02573644	5	0.035387605
67	1.826074803	1	0.029388902	6	0.026123469	3	0.022858035	6	0.042450637	4	0.029388902	2	0.019592601	6	0.039185203
66	1.819543936	8	0.049729342	2	0.013261158	4	0.023207026	7	0.039783473	5	0.033152894	4	0.023207026	6	0.036468184
65	1.812913357	7	0.057233753	2	0.016833457	3	0.020200148	5	0.037033605	5	0.037033605	3	0.020200148	5	0.037033605
64	1.806179974	10	0.078653382	3	0.017098561	3	0.023937986	6	0.047875972	6	0.037616835	3	0.03077741	6	0.037616835
63	1.799340549	13	0.07643746	2	0.01389772	4	0.02779544	8	0.05559088	5	0.04169316	6	0.03821873	5	0.03821873
62	1.792391689	9	0.06355669	2	0.014123709	4	0.021185563	8	0.045902054	7	0.056494836	5	0.052963909	6	0.045902054

61	1.785329835	9	0.064607262	2	0.017946462	2	0.021535754	5	0.043071508	9	0.053839385	10	0.053839385	7	0.053839385
60	1.77815125	9	0.047445052	3	0.021897716	4	0.029196955	7	0.040145813	6	0.047445052	5	0.047445052	8	0.047445052
59	1.770852012	4	0.051968127	3	0.018560045	4	0.018560045	4	0.033408081	7	0.044544108	8	0.044544108	5	0.044544108
58	1.763427994	10	0.052871965	2	0.015106276	1	0.011329707	5	0.037765689	5	0.041542258	4	0.052871965	7	0.052871965
57	1.755874856	4	0.038434143	2	0.019217072	2	0.015373657	5	0.046120972	6	0.046120972	10	0.057651215	7	0.061494629
56	1.748188027	6	0.046952025	3	0.023476013	2	0.023476013	7	0.054777363	6	0.043039356	5	0.054777363	9	0.058690031
55	1.740362689	6	0.051798043	3	0.023906789	4	0.035860184	7	0.055782508	5	0.051798043	9	0.051798043	6	0.047813578
54	1.73239376	7	0.064943122	3	0.024353671	5	0.024353671	7	0.040589451	8	0.056825232	4	0.052766286	6	0.056825232
53	1.72427587	9	0.066180208	3	0.033090104	1	0.016545052	3	0.049635156	6	0.062043945	9	0.066180208	8	0.045498893
52	1.716003344	7	0.059032173	5	0.03373267	3	0.025299503	9	0.080115092	9	0.063248757	7	0.075898508	3	0.054815589
51	1.707570176	7	0.051601031	3	0.034400687	3	0.025800515	10	0.07310146	6	0.051601031	11	0.086001718	10	0.07310146
50	1.698970004	5	0.052643546	5	0.039482659	3	0.035095697	7	0.070191394	6	0.06141747	9	0.052643546	7	0.065804432
49	1.69019608	7	0.058206477	4	0.022387107	5	0.035819371	9	0.107458112	8	0.053729056	3	0.053729056	8	0.071638741
48	1.681241237	6	0.045716897	1	0.022858449	3	0.041145207	15	0.105148864	4	0.045716897	9	0.077718725	8	0.050288587
47	1.672097858	4	0.070050197	4	0.032690092	6	0.046700131	8	0.084060236	6	0.065380184	8	0.070050197	3	0.051370144
46	1.662757832	11	0.095453179	3	0.033408613	4	0.04772659	10	0.09068052	8	0.104998497	7	0.04772659	8	0.066817225
45	1.653212514	9	0.112238129	4	0.043919268	6	0.034159431	9	0.082958617	14	0.102478292	3	0.034159431	6	0.078078698
44	1.643452676	14	0.10483432	5	0.054913215	1	0.039936884	8	0.089857988	7	0.079873767	4	0.059905325	10	0.069889546
43	1.633468456	7	0.051095826	6	0.051095826	7	0.061314991	10	0.107301234	9	0.081753321	8	0.051095826	4	0.056205408
42	1.62324929	3	0.073258036	4	0.041861735	5	0.062792602	11	0.104654337	7	0.073258036	2	0.041861735	7	0.073258036
41	1.612783857	11	0.101876721	4	0.048257394	7	0.091152856	9	0.107238654	7	0.101876721	6	0.048257394	7	0.101876721
40	1.602059991	8	0.07696769	5	0.060474614	10	0.065972306	11	0.142939996	12	0.126446919	3	0.087963074	12	0.082465382
39	1.591064607	6	0.050764547	6	0.045124042	2	0.050764547	15	0.14101263	11	0.157934146	13	0.118450609	3	0.090248083
38	1.579783597	3	0.098445917	2	0.034745618	7	0.09265498	10	0.110027789	17	0.173728088	8	0.104236853	13	0.127400598
37	1.568201724	14	0.196337184	4	0.095193786	9	0.083294563	9	0.095193786	13	0.184437961	10	0.113042621	9	0.160639515
36	1.556302501	19	0.232454672	12	0.097875651	5	0.079523967	7	0.06728951	18	0.165165162	9	0.17128239	18	0.177399618
35	1.544068044	19	0.302139055	4	0.1384804	8	0.0692402	4	0.081829328	9	0.125891273	19	0.302139055	11	0.132185837
34	1.531478917	29	0.220404612	18	0.194474657	3	0.077789863	9	0.116684794	11	0.129649772	29	0.278747009	10	0.129649772
33	1.51851394	5	0.046773865	12	0.120275654	9	0.106911692	9	0.1737315	9	0.140321596	14	0.200459423	10	0.120275654
32	1.505149978	2	0.096517991	6	0.151671129	7	0.110306276	17	0.21371841	12	0.165459414	16	0.227506694	8	0.151671129
31	1.491361694	12	0.128163952	16	0.16376505	9	0.092562854	14	0.16376505	12	0.15664483	17	0.270568343	14	0.142404391
30	1.477121255	6	0.242933738	7	0.110424426	4	0.103062798	9	0.154594197	10	0.132509311	21	0.272380251	6	0.095701169
29	1.462397998	27	0.304799331	8	0.160019649	10	0.144779682	12	0.167639632	8	0.160019649	16	0.220979515	7	0.205739549
28	1.447158031	13	0.244811141	13	0.19742834	9	0.142148405	10	0.284296809	13	0.236914008	13	0.244811141	20	0.260605409
27	1.431363764	18	0.188489786	12	0.114732913	9	0.13112333	26	0.245856243	17	0.229465827	18	0.229465827	13	0.221270619
26	1.414973348	5	0.127750045	2	0.110716705	7	0.127750045	4	0.212916741	11	0.23846675	10	0.315116777	14	0.187366732
25	1.397940009	10	0.132965752	11	0.17728767	8	0.132965752	21	0.292524655	17	0.38116849	27	0.301389038	8	0.159558903
24	1.380211242	5	0.166350651	9	0.138625543	7	0.101658731	12	0.184834057	26	0.38815152	7	0.286492788	10	0.231042571
23	1.361727836	13	0.308882483	6	0.154441242	4	0.222009285	8	0.135136086	16	0.337840216	24	0.250967018	15	0.318535061
22	1.342422681	19	0.484881266	10	0.323254177	19	0.303050791	6	0.313152484	19	0.303050791	2	0.151525396	18	0.303050791
21	1.322219295	29	0.434380631	22	0.349623435	11	0.264866238	25	0.381407383	11	0.296650187	13	0.243676939	12	0.328434136
20	1.301029996	12	0.278454934	11	0.133658368	14	0.300731329	11	0.345284118	17	0.4232515	10	0.367560513	19	0.389836907
19	1.278753601	13	0.363956986	1	0.187848767	13	0.305254246	20	0.410919177	21	0.469621917	23	0.5635463	16	0.481362465
18	1.255272505	18	0.53370705	15	0.39717734	13	0.248235837	15	0.570942426	19	0.583354218	25	0.409589131	25	0.844001847
17	1.230448921	25	0.789868162	17	0.434427489	7	0.144809163	31	0.658223468	28	0.658223468	8	0.40809855	43	0.921512855
16	1.204119983	35	0.518531387	16	0.378387769	4	0.294301598	19	0.54656011	22	0.85487607	23	1.023048411	27	0.574588834
15	1.176091259	2	1.677940509	11	1.692922121	17	1.093657653	20	0.853951866	39	1.632995674	50	3.056248784	14	0.719117361
14	1.146128036	110	2.140281444	102	2.285112519	56	1.078186893	37	1.33566436	70	1.480495435	154	3.218468337	34	0.852894109
13	1.113943352	23	0.83429055	40	0.990720028	11	0.590955806	46	1.181911613	22	0.643098966	46	1.008101082	19	0.730004231
12	1.079181246	25	0.623511255	17	1.076973985	23	0.699088376	22	0.83134834	15	1.095868266	12	0.566828413	23	1.171445388
11	1.041392685	8	1.138298842	40	0.952031759	14	0.745068333	22	1.552225693	43	1.738492777	18	1.28317324	39	1.552225693
10	1	47	1.853178368	6	1.304088481	22	0.983786047	53	2.173480802	41	2.516661981	44	2.013329585	36	1.853178368
9	0.954242509	34	2.659931167	51	1.867067069	21	1.636880718	42	2.557626122	69	3.197032653	44	1.585728196	45	2.173982204
8	0.903089987	70	4.494375891	22	2.145702038	43	3.131565137	58	3.334536951	56	2.638633587	18	5.335259122	40	2.522649694
7	0.84509804	85	7.23025328	52	2.410084427	65	4.58585509	57	2.811765164	35	1.841036715	166	7.933194571	47	4.184174352
6	0.77815125	131	8.749527688	20	4.830056009	72	5.186371616	27	10.09560887	20	6.532452799	71	3.088068596	78	7.047130898
5	0.698970004	90	12.30757165	102	17.44380234	59	5.572325748	228	29.89673901	145	10.03018635	7	11.43538153	100	12.45293667
4	0.602059991	164	28.54850131	258	39.79298761	56	20.92723838	389	32.48407152	62	6.24693683	229	32.48407152	157	26.6119509
3	0.477121255	293	54.76438157	379	48.16095935	279	34.51388677	131	69.99627547	38	7.571924139	291	36.18675374	269	50.00991757
2	0.301029996	329	82.63273381	168	151.8696328	113	95.57702362	664	106.1130735	48	65.47402406	120	79.17088886	299	57.64724417
1	0	220	0	841	0	522	0	41	0	387	0	406	0	84	0
0		4124		7090		8582		3491		8902		3375		7139	
frequency	log frequency	amtrtw256.ascii	area amt	bbcrtw256.ascii	area bbc	bcrtw256.ascii	area bc	bgv001rtw256.acsii	area bgv	devrtw256.ascii	area dev	epatrtw256.ascii	area epat	mecrtw256.ascii	area mec
total sum			218.8366551		282.4406538		181.6087794		274.5716306		122.7719699		197.5197391		179.2429044
sum 1 to 32			215.0002226		280.5056306		179.4915066		270.5000503		118.6555084		194.1522182		175.345535
ratio			0.982468967		0.993148921		0.988341572		0.985171154		0.966470673		0.982950965		0.978256492
min	0.966470673														
max	0.993148921														

rmrtw256.ascii	area rm	sadrtw256.ascii	area sad	sdcrtw256.ascii	area sdc	sm2rtw256.ascii	area sm2	taprtw256.ascii	area tap	bcv001w256.ascii	area bcv	dm002rtw256.ascii	area dm	kls002rtw256.ascii	area kls
2	0.006812497	1	0.005109373	0	0.003406249	1	0.005109373	4	0.01192187	2	0.006812497	3	0.01192187	0	0
2	0.013732703	2	0.005149764	2	0.00858294	2	0.010299528	3	0.010299528	2	0.006866352	4	0.012016115	0	0.001716588
6	0.013842128	1	0.006921064	3	0.013842128	4	0.010381596	3	0.010381596	2	0.006921064	3	0.010381596	1	0.003460532
2	0.00872082	3	0.006976656	5	0.017441639	2	0.00872082	3	0.012209147	2	0.005232492	3	0.010464984	1	0.003488328
3	0.015824582	1	0.003516574	5	0.015824582	3	0.008791434	4	0.014066295	1	0.005274861	3	0.012308008	1	0.001758287
6	0.019499044	1	0.00177264	4	0.010635842	2	0.005317921	4	0.014181123	2	0.005317921	4	0.014181123	0	0.00177264
5	0.017872302	0	0.00357446	2	0.014297841	1	0.005361691	4	0.012510611	1	0.00357446	4	0.014297841	1	0.00357446
5	0.012614435	2	0.005406186	6	0.018020621	2	0.007208249	3	0.012614435	1	0.005406186	4	0.012614435	1	0.003604124
2	0.005451427	1	0.003634285	4	0.012719996	2	0.007268569	4	0.012719996	2	0.005451427	3	0.012719996	1	0.003634285
1	0.005497431	1	0.003664954	3	0.010994862	2	0.007329908	3	0.012827339	1	0.005497431	4	0.012827339	1	0.001832477
2	0.009240364	1	0.003696146	3	0.009240364	2	0.011088437	4	0.014784582	2	0.007392291	3	0.012936509	0	0.001848073
3	0.005591809	1	0.003727873	2	0.005591809	4	0.01491149	4	0.013047554	2	0.009319681	4	0.013047554	1	0.003727873
0	0.005640223	1	0.001880074	1	0.005640223	4	0.015040595	3	0.013160521	3	0.009400372	3	0.013160521	1	0.003760149
3	0.009482473	0	0.001896495	2	0.009482473	4	0.011378967	4	0.015171956	2	0.007585978	4	0.013275462	1	0.003792989
2	0.007652816	1	0.003826408	3	0.015305631	2	0.00956602	4	0.015305631	2	0.005739612	3	0.013392427	1	0.003826408
2	0.009651052	1	0.003860421	5	0.019302104	3	0.009651052	4	0.015441683	1	0.005790631	4	0.015441683	1	0.003860421
3	0.007790088	1	0.003895044	5	0.021422741	2	0.007790088	4	0.015580176	2	0.005842566	4	0.015580176	1	0.001947522
1	0.00589544	1	0.003930294	6	0.021616615	2	0.007860587	4	0.013756028	1	0.00589544	4	0.013756028	0	0
2	0.007932374	1	0.005949281	5	0.017847842	2	0.007932374	3	0.011898562	2	0.007932374	3	0.013881655	0	0.001983094
2	0.010006856	2	0.006004114	4	0.014009599	2	0.008005485	3	0.014009599	2	0.008005485	4	0.018012341	1	0.006004114
3	0.014139922	1	0.004039978	3	0.010099945	2	0.008079956	4	0.016159911	2	0.008079956	5	0.016159911	2	0.004039978
4	0.01631165	1	0.004077912	2	0.006116869	2	0.010194781	4	0.014272693	2	0.008155825	3	0.014272693	0	0.002038956
4	0.014407982	1	0.006174849	1	0.006174849	3	0.012349699	3	0.010291415	2	0.006174849	4	0.018524548	1	0.004116566
3	0.012467879	2	0.00623394	2	0.00623394	3	0.010389899	2	0.012467879	1	0.00623394	5	0.020779799	1	0.00207798
3	0.008392229	1	0.006294172	1	0.010490286	2	0.008392229	4	0.012588344	2	0.010490286	5	0.018882516	0	0.002098057
1	0.004237053	2	0.006355579	4	0.016948212	2	0.010592632	2	0.014829685	3	0.008474106	4	0.014829685	1	0.002118526
1	0.004278798	1	0.004278798	4	0.012836394	3	0.012836394	5	0.019254591	1	0.006418197	3	0.017115192	0	0.002139399
1	0.008642748	1	0.004321374	2	0.012964121	3	0.010803434	4	0.015124808	2	0.008642748	5	0.021606869	1	0.004321374
3	0.013094416	1	0.004364805	4	0.017459222	2	0.008729611	3	0.010912014	2	0.008729611	5	0.017459222	1	0.006547208
3	0.008818238	1	0.004409119	4	0.019841035	2	0.008818238	2	0.011022797	2	0.006613678	3	0.017636476	2	0.006613678
1	0.004454341	1	0.004454341	5	0.020044536	2	0.011135854	3	0.013363024	1	0.006681512	5	0.020044536	1	0.004454341
1	0.011251253	1	0.004500501	4	0.018002005	3	0.013501504	3	0.013501504	2	0.011251253	4	0.018002005	1	0.006750752
4	0.018190511	1	0.006821442	4	0.015916697	3	0.013642883	3	0.015916697	3	0.011369069	4	0.018190511	2	0.006821442
4	0.018383007	2	0.006893628	3	0.013787255	3	0.013787255	4	0.018383007	2	0.011489379	4	0.018383007	1	0.002297876
4	0.01857962	1	0.006967358	3	0.00928981	3	0.011612263	4	0.016257168	3	0.011612263	4	0.01857962	0	0.004644905
4	0.018780485	2	0.009390242	1	0.007042682	2	0.014085364	3	0.016432924	2	0.009390242	4	0.018780485	2	0.007042682
4	0.014239305	2	0.00949287	2	0.011866088	4	0.016612523	4	0.016612523	2	0.014239305	4	0.021358958	1	0.004746435
2	0.011997207	2	0.009597766	3	0.019195532	3	0.011997207	3	0.01679609	4	0.009597766	5	0.023994414	1	0.004798883
3	0.01698376	2	0.007278754	5	0.024282514	2	0.012131257	4	0.01698376	0	0.002426251	5	0.024262514	1	0.004852503
4	0.017175671	1	0.004907334	5	0.02699034	3	0.012268336	3	0.019629338	1	0.009814669	5	0.02699034	1	0.007361002
3	0.012408549	1	0.00248171	6	0.027298807	2	0.009926839	5	0.022335388	3	0.014890259	6	0.029780517	2	0.009926839
2	0.010041603	0	0.005020801	5	0.025104007	2	0.010041603	4	0.020083206	3	0.010041603	6	0.030124808	2	0.010041603
2	0.007619288	2	0.010159051	5	0.022857865	2	0.010159051	4	0.017778339	1	0.007619288	6	0.025397628	2	0.010159051
1	0.015418919	2	0.010279279	4	0.020558559	2	0.010279279	3	0.015418919	2	0.012849099	4	0.023128378	2	0.010279279
5	0.020804775	2	0.007801791	4	0.018204178	2	0.010402387	3	0.020804775	3	0.013002984	5	0.026005968	2	0.007801791
3	0.01579272	1	0.00789636	3	0.01579272	2	0.01052848	5	0.02368908	2	0.01052848	5	0.02105696	1	0.00789636
3	0.015986501	2	0.00799325	3	0.010657667	2	0.010657667	4	0.023979751	2	0.010657667	3	0.021315334	2	0.010657667
3	0.018882612	1	0.008092548	1	0.005395032	2	0.01348758	5	0.018882612	2	0.016185096	5	0.024277643	2	0.008092548
4	0.024583031	2	0.008194344	1	0.008194344	3	0.019120135	2	0.016388687	4	0.013657239	4	0.019120135	1	0.008194344
5	0.022129954	1	0.005532489	2	0.016597466	4	0.01936371	4	0.024896199	1	0.008298733	3	0.024896199	2	0.011064977
3	0.016811633	1	0.008405816	4	0.025217449	3	0.019613571	5	0.025217449	2	0.014009694	6	0.028019388	2	0.008405816
3	0.017031399	2	0.014192832	5	0.036901364	4	0.022708532	4	0.019869965	3	0.017031399	4	0.025547098	1	0.008515699
3	0.023009316	3	0.014380822	8	0.034513973	4	0.023009316	3	0.017256987	3	0.011504658	5	0.020133151	2	0.008628493
5	0.020403403	2	0.014573859	4	0.03206249	4	0.017488631	3	0.020403403	1	0.008744315	2	0.014573859	1	0.008744315
2	0.014772149	3	0.011817719	7	0.035453158	2	0.011817719	4	0.029544298	2	0.011817719	3	0.020681009	2	0.008863289
3	0.017971091	1	0.008985546	5	0.026956637	2	0.014975909	6	0.032947	2	0.011980727	4	0.023961455	1	0.005990364
3	0.012148295	2	0.009111222	4	0.030370739	3	0.021259517	5	0.030370739	2	0.021259517	4	0.027333665	1	0.006074148
1	0.02156108	1	0.012320617	6	0.024641235	4	0.024641235	5	0.02156108	5	0.024641235	5	0.033881698	1	0.006160309
6	0.034369221	3	0.012497899	2	0.012497899	4	0.031244746	2	0.012497899	3	0.024995797	6	0.037493696	1	0.006248949
5	0.03170089	1	0.006340178	2	0.012680356	6	0.034870979	2	0.022190623	5	0.025360712	6	0.034870979	1	0.006340178
5	0.028953495	1	0.01286822	2	0.01930233	5	0.028953495	5	0.03860466	3	0.02573644	5	0.02573644	1	0.009651165
4	0.032654336	3	0.022858035	4	0.052246937	4	0.026123469	7	0.052246937	5	0.029388902	3	0.035919769	2	0.022858035
6	0.036468184	4	0.019891737	12	0.05967521	4	0.016576447	9	0.036468184	4	0.019891737	8	0.043098763	5	0.026522316
5	0.020200148	2	0.013466765	6	0.047133679	1	0.020200148	2	0.023566839	2	0.023566839	5	0.037033605	3	0.013466765
1	0.013678849	2	0.010259137	8	0.047875972	5	0.03077741	5	0.023937986	5	0.020518274	6	0.037616835	1	0.006839425
3	0.01389772	1	0.01042329	6	0.03126987	4	0.01737215	2	0.03126987	1	0.01389772	5	0.04516759	1	0.00694886
1	0.035309272	2	0.010592782	3	0.035309272	1	0.007061854	7	0.0388402	3	0.024716491	8	0.045902054	1	0.014123709

9	0.053839385	1	0.017946462	7	0.039482215	1	0.017946462	4	0.039482215	4	0.025125046	5	0.039482215	3	0.017946462
6	0.036496194	4	0.021897716	4	0.051094671	4	0.021897716	7	0.054744291	3	0.021897716	6	0.032846574	2	0.014598477
4	0.029696072	2	0.014848036	10	0.040832099	2	0.014848036	8	0.063104154	3	0.03712009	3	0.033408081	2	0.018560045
4	0.018882845	2	0.018882845	1	0.037765689	2	0.011329707	9	0.064201672	7	0.037765689	6	0.026435983	3	0.018882845
1	0.011530243	3	0.019217072	9	0.049964386	1	0.011530243	8	0.049964386	3	0.023060486	1	0.015373657	2	0.015373657
2	0.054777363	2	0.023476013	4	0.019563344	2	0.019563344	5	0.027388681	3	0.015650675	3	0.043039356	2	0.023476013
12	0.083673762	4	0.031875719	1	0.031875719	3	0.027891254	2	0.023906789	1	0.019922324	8	0.051798043	4	0.019922324
9	0.064943122	4	0.032471561	7	0.097414683	4	0.024353671	4	0.052766286	4	0.024353671	5	0.044648396	1	0.01623578
7	0.045498893	4	0.028953841	17	0.099270312	2	0.024817578	9	0.062043945	2	0.024817578	6	0.062043945	3	0.012408789
4	0.042165838	3	0.025299503	7	0.063248757	4	0.029516086	6	0.037949254	4	0.029516086	9	0.06746534	0	0.012649751
6	0.098901975	3	0.030100601	8	0.090301804	3	0.034400687	3	0.043000859	3	0.030100601	7	0.068801374	3	0.025800515
17	0.10090013	4	0.035095697	13	0.074578357	5	0.035095697	7	0.078965319	4	0.035095697	9	0.065804432	3	0.017547849
6	0.035819371	4	0.040296792	4	0.06716132	3	0.044774213	11	0.080593584	4	0.049251635	6	0.058206477	1	0.013432264
2	0.018286759	5	0.032001828	11	0.096005484	7	0.059431966	7	0.064003656	7	0.045716897	7	0.077718725	2	0.032001828
2	0.037360105	2	0.028020079	10	0.084060236	6	0.032690092	7	0.051370144	3	0.032690092	10	0.060710171	5	0.051370144
6	0.038181272	4	0.042953931	8	0.04772659	1	0.038181272	4	0.038181272	4	0.062044566	3	0.062044566	6	0.033408613
2	0.043919268	5	0.058559024	2	0.082958617	7	0.048799186	4	0.058559024	9	0.063438942	10	0.087838536	1	0.014639756
7	0.064897436	7	0.044928994	15	0.139779093	3	0.044928994	8	0.064897436	4	0.039936884	8	0.074881657	2	0.024960552
6	0.071534156	2	0.025547913	13	0.102191652	6	0.076643739	5	0.071534156	4	0.045986243	7	0.091972487	3	0.025547913
8	0.073258036	3	0.031396301	7	0.141283355	9	0.083723469	9	0.052327168	5	0.057559885	11	0.094188903	2	0.015698151
6	0.091152856	3	0.026809663	20	0.155496048	7	0.075067058	1	0.016085798	6	0.042895462	7	0.069705125	1	0.042895462
11	0.137442304	2	0.032986153	9	0.082465382	7	0.082465382	2	0.038483845	2	0.021990769	6	0.065972306	7	0.060474614
14	0.124091115	4	0.067686062	6	0.118450609	8	0.101529094	5	0.112810104	2	0.056405052	6	0.107169599	4	0.045124042
8	0.086864044	8	0.086864044	15	0.214264642	10	0.110027789	15	0.127400598	8	0.063700299	13	0.098445917	4	0.034745618
7	0.077344951	7	0.059496116	22	0.208236408	9	0.077344951	7	0.083294563	3	0.053546505	4	0.059496116	2	0.07139534
6	0.091758423	3	0.06728951	13	0.116227336	4	0.097875651	7	0.085641195	6	0.073406739	6	0.122344564	10	0.091758423
9	0.151069528	8	0.088123891	6	0.062945637	12	0.1384804	7	0.075534764	6	0.094418455	14	0.125891273	5	0.075534764
15	0.149097237	6	0.058342397	4	0.149097237	10	0.077789863	5	0.09075484	9	0.116684794	6	0.123167283	7	0.077789863
8	0.140321596	3	0.026727923	19	0.247233289	2	0.093547731	9	0.106911692	9	0.113593673	13	0.160367539	5	0.08686575
13	0.117200418	1	0.020682427	18	0.137882845	12	0.172353556	7	0.096517991	8	0.137882845	11	0.179247698	8	0.151671129
4	0.192245928	2	0.056961756	2	0.242087465	13	0.178005489	7	0.128163952	12	0.135284172	15	0.185125708	14	0.142404391
23	0.272380251	6	0.066254656	32	0.257656994	12	0.154594197	11	0.176679082	7	0.073616284	11	0.117786055	6	0.110424426
14	0.144779682	3	0.053339883	3	0.114299749	9	0.129539716	13	0.175259615	3	0.099059783	5	0.091439799	9	0.091439799
5	0.142148405	4	0.055279935	12	0.244811141	8	0.134251271	10	0.150045538	10	0.102662737	7	0.19742834	3	0.08686847
13	0.229465827	3	0.073756873	19	0.295027491	9	0.139318538	9	0.262246659	3	0.090147289	18	0.196684994	8	0.114732913
15	0.306600107	6	0.076650027	17	0.272533429	8	0.119233375	23	0.315116777	8	0.144783384	6	0.161816723	6	0.204400072
21	0.345710956	3	0.044321917	15	0.221609587	6	0.141830136	14	0.212745204	9	0.141830136	13	0.23047397	18	0.212745204
18	0.443601737	2	0.06469192	10	0.268009383	10	0.184834057	10	0.203317463	7	0.175592354	13	0.19407576	6	0.110900434
30	0.472976302	5	0.115830931	19	0.299229906	10	0.24131444	12	0.376450526	12	0.24131444	8	0.222009285	6	0.125483509
19	0.212135554	7	0.10101693	12	0.313152484	15	0.292949098	27	0.414169415	13	0.212135554	15	0.23233894	7	0.181830475
2	0.190703692	3	0.158919743	19	0.339028785	14	0.307244837	14	0.497948528	8	0.22248764	8	0.148325093	11	0.201298341
16	0.311869526	12	0.222763947	13	0.222763947	15	0.4232515	33	0.54577167	13	0.155934763	6	0.144796566	8	0.21162575
12	0.352216438	8	0.211329863	7	0.375697534	23	0.469621917	16	0.399178629	1	0.187848767	7	0.258292054	11	0.187848767
18	0.422000923	10	0.260647629	25	0.39717734	17	0.359941964	18	0.471648091	15	0.322706588	15	0.471648091	5	0.273059421
16	0.355440673	11	0.381769611	7	0.460756428	12	0.289618326	20	0.394934081	11	0.19746704	23	0.539743244	17	0.315947265
11	0.616631919	18	0.378387769	28	0.574588834	10	0.560574472	10	0.714732452	4	0.210215427	18	0.448459578	7	0.378387769
33	0.898896701	9	0.928859925	13	1.962591131	30	1.093657653	41	1.348345052	11	0.928859925	14	1.528124392	20	0.809007031
27	0.643693667	53	1.126463918	118	2.108096761	43	1.255202651	49	1.480495435	51	1.190833285	88	1.979358027	34	0.579324301
13	0.83429055	17	0.469288434	13	0.382383169	35	1.181911613	43	1.494770569	23	0.730004231	35	1.129768453	2	0.069524213
35	1.794956642	10	0.585722694	9	1.473753875	33	1.190339668	43	1.454859594	19	0.717982657	30	1.26591679	2	0.264519926
60	2.255901341	21	0.807157361	69	3.311414813	30	1.345262268	34	1.448743981	19	0.869246388	37	0.848550046	12	0.393230509
49	2.173480802	18	1.2125735	91	4.232567877	35	1.738784641	36	2.676813198	23	0.846513575	4	1.441360953	7	1.121058519
46	2.5064736	35	2.020524637	94	4.322388147	41	2.736659951	81	2.429744816	14	2.046100898	59	2.046100898	42	2.225134726
52	3.769476554	44	2.087710091	75	3.769476554	66	3.044577216	14	1.536786595	66	2.4066658	21	2.348673853	45	2.551645667
78	6.795099148	28	4.385014721	55	6.594258779	39	2.410084427	39	7.029412911	17	7.698880808	60	3.079552323	43	5.221849591
125	9.739293264	103	6.928359029	142	11.08537445	33	1.583624921	171	10.13519949	213	9.937246379	32	7.324265259	113	10.92701195
121	8.818811184	72	9.497181275	138	9.448726268	7	6.735245904	85	22.72539805	38	8.673446164	153	26.16570351	163	11.24156151
61	18.17858618	124	25.79984911	57	41.85447676	132	13.9931385	384	45.16535328	141	17.36648439	387	48.4762298	69	19.74032038
230	40.06076144	289	57.14161356	613	67.35490659	92	24.38863938	339	66.73858718	137	24.91691316	389	83.64334805	247	22.97990931
225	59.45342414	360	72.54822896	152	94.82444863	185	138.624313	419	141.634613	146	69.537929	561	108.5213134	14	17.30922475
170	0	122	0	478	0	736	0	522	0	316	0	160	0	101	0
4182		8075		5821		7395		7516		8080		5581		4157	
rmrtw256.ascii	area rm	sadrtw256.ascii	area sad	sdcrtw256.ascii	area sdc	sm2rtw256.ascii	area sm2	taprtw256.ascii	area tap	bcv001w256.ascii	area bcv	dm002rtw256.ascii	area dm	kls002rtw256.ascii	area kls
	166.1331432		189.4909337		261.8577098		208.0800474		315.89769		152.8277051		297.3433719		99.95940732
	163.0512535		187.881153		257.7611771		205.6199187		312.8340488		150.7180759		293.8179577		98.53439052
	0.981449279		0.991504709		0.984355883		0.988177008		0.990301793		0.986196029		0.988143626		0.985744045

tw001rtw256.acsii	area tw	vam002rtw256.ascii	area vam
2	0.005109373	4	0.01192187
1	0.003433176	3	0.010299528
1	0.003460532	3	0.010381596
1	0.005232492	3	0.010464984
2	0.005274861	3	0.010549721
1	0.003545281	3	0.010635842
1	0.005361691	3	0.010723381
2	0.007208249	3	0.012614435
2	0.005451427	4	0.012719996
1	0.003664954	3	0.010994862
1	0.003696146	3	0.011088437
1	0.003727873	3	0.013047554
1	0.003760149	4	0.015040595
1	0.007585978	4	0.011378967
3	0.007652816	2	0.00956602
1	0.005790631	3	0.011581262
2	0.005842566	3	0.011685132
1	0.00589544	3	0.011790881
2	0.007932374	3	0.011898562
2	0.008005485	3	0.012008227
2	0.004039978	3	0.014139922
0	0.004077912	4	0.01631165
2	0.010291415	4	0.014407982
3	0.012467879	3	0.014545859
3	0.010490286	4	0.014686401
2	0.008474106	3	0.012711159
2	0.006418197	3	0.017115192
1	0.008642748	5	0.019446182
3	0.010912014	4	0.013094416
2	0.008818238	2	0.008818238
2	0.008908683	2	0.011135854
2	0.006750752	3	0.011251253
1	0.004547628	2	0.013642883
1	0.002297876	4	0.018383007
0	0.004644905	4	0.01857962
2	0.011737803	4	0.016432924
3	0.011866088	3	0.01898574
2	0.009597766	5	0.01679609
2	0.009705006	2	0.009705006
2	0.009814669	2	0.017175671
2	0.009926839	5	0.022335388
2	0.010041603	4	0.020083206
2	0.010159051	4	0.025397628
2	0.007709459	6	0.028268018
1	0.007801791	5	0.020804775
2	0.00789636	3	0.01842484
1	0.013322084	4	0.018650917
4	0.021580128	3	0.018882612
4	0.019120135	4	0.019120135
3	0.016597466	3	0.01936371
3	0.011207755	4	0.019613571
1	0.005677133	3	0.017031399
1	0.008628493	3	0.023009316
2	0.017488631	5	0.026232946
4	0.017726579	4	0.026589868
2	0.014975909	5	0.032947
3	0.021259517	6	0.036444886
4	0.018480926	6	0.033881698
2	0.009373424	5	0.031244746
1	0.006340178	5	0.028530801
1	0.02573644	4	0.02573644
7	0.029388902	4	0.026123469
2	0.013261158	4	0.033152894
2	0.016833457	6	0.037033605
3	0.023937986	5	0.041036547
4	0.03126987	7	0.0347443
5	0.028247418	3	0.017654636

3	0.032303631	2	0.028714339
6	0.032846574	6	0.040145813
3	0.033408081	5	0.051968127
6	0.033989121	9	0.045318827
3	0.019217072	3	0.030747315
2	0.023476013	5	0.027388681
4	0.027891254	2	0.035860184
3	0.036530506	7	0.044648396
6	0.033090104	4	0.04136263
2	0.025299503	6	0.037949254
4	0.047300945	3	0.030100601
7	0.057030508	4	0.026321773
6	0.058206477	2	0.022387107
7	0.054860277	3	0.032001828
5	0.042030118	4	0.037360105
4	0.033408613	4	0.04772659
3	0.063438942	6	0.039039349
10	0.084865878	2	0.059905325
7	0.066424574	10	0.097082069
6	0.083723469	9	0.057559885
10	0.117962519	2	0.037533529
12	0.082465382	5	0.060474614
3	0.062045557	6	0.050764547
8	0.127400598	3	0.075282172
14	0.124941845	10	0.095193786
7	0.085641195	6	0.10399288
7	0.0692402	11	0.169953219
4	0.058342397	16	0.187992169
5	0.08686575	13	0.153685558
8	0.103412134	10	0.165459414
7	0.071202196	14	0.170885269
3	0.088339541	10	0.125147683
9	0.144779682	7	0.091439799
10	0.150045538	5	0.165839805
9	0.17209937	16	0.21307541
12	0.144783384	10	0.119233375
5	0.212745204	4	0.088643835
19	0.249525977	6	0.073933623
8	0.20270413	2	0.20270413
13	0.242440633	19	0.444474494
11	0.392002033	25	0.402596682
26	0.345284118	13	0.356422315
5	0.422659725	19	0.410919177
31	0.595766009	16	0.422000923
17	0.355440673	18	0.500249836
10	0.462473939	20	0.336344683
23	0.464429962	4	1.24347377
8	0.28966215	79	1.754065244
10	0.504050541	30	1.199292666
19	1.417071033	39	1.001396864
56	1.84197449	14	0.931335416
33	1.876057113	31	1.624390915
49	2.941270041	40	2.199558465
66	2.638633587	46	3.566504739
25	3.54817985	77	7.933194571
81	4.988418501	160	10.68946822
45	16.32933719	110	23.8398632
292	38.48113088	382	33.42111204
324	52.03496705	153	24.91691316
267	58.24930416	130	141.183068
120	0	808	0
5609		6289	
tw001rtw256.acsii	area tw	vam002rtw256.ascii	area vam
	192.3605621		262.7915646
	189.9601908		259.7930077
	0.9875215		0.9885896

Appendix E

Using the Local Mode for Edge Detection

Using the Local Mode for Edge Detection in Ultrasound Images

Tyna Hope
Department of Electrical and Computer Engineering, Dalhousie University, Halifax
e-mail: tahope@dal.ca

Norma Linney
Department of Mathematics and Computer Science, Saint Mary's University, Halifax
e-mail: norma.linney@stmarys.ca

Peter Gregson
Department of Electrical and Computer Engineering, Dalhousie University, Halifax
e-mail: peter.gregson@dal.ca

Abstract

We are investigating quantitative ultrasound texture measures as an additional source of diagnostic information for the detection of white matter damage in very preterm infants. White matter damage is a form of brain damage which leads to cerebral palsy. Ultrasound speckle properties have been shown to correlate to scatterer properties in phantoms. The disease process alters the scatterer type and density in white matter. We are enhancing speckle edges in cranial ultrasound images to determine if local speckle gradients and speckle edge densities correlate to patient outcome. Speckle edges are very diffuse and traditional edge enhancing schemes, such as Sobel, do not perform well. To capture the speckle edge detail, we use film images scanned at a very high resolution. The digitization introduces a significant noise component and does not suppress the ultrasound film grain. We present a non-linear filter to enhance the speckle edge information. The filter (DM) exploits the changes to speckle edges that result from applying local mode filtering. This technique has the advantages of maintaining edge center localization with large window sizes and performing better than Sobel for diffuse edges. In this work, we discuss the filter, its parameters, and their selection for a given application.

Keywords— *Image processing; Biomedical Engineering; Digital Signal Processing.*

1 Introduction

White matter damage (WMD) is an important precursor to cerebral palsy in preterm infants. We are investigating ultrasound (US) texture as a means to detect WMD. We obtain texture descriptors in US images with speckle, as we hypothesize that there is information within the speckle properties. Ultrasound speckle properties have been shown to correlate to scatterer properties in phantoms [1]. The early stages of nonhemorrhagic infarct are characterized by edema of the tissue, which reduces the ultrasonic scatterer density [2]. It is believed that the cause of WMD leading to CP occurs during the neonatal period and therefore, we use the earliest available clinical images, obtained within days of birth.

We use analog film images digitized at 3200 dpi, as modern US equipment has bandwidth restrictions. Our goal is to measure speckle characteristics within the noisy film environment. US speckle edge profiles are diffuse and traditional edge detection schemes do not work well.

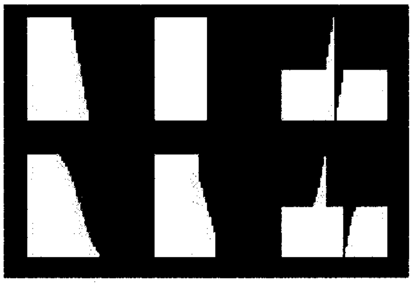

Figure 1. Components forming the DM output. Top row: linear profile, bottom row: sigmoidal edge profile. Left to right: Pre-mode filter edge profile, mode filtered edge profile, and the difference.

2 New Edge Enhancement Technique

We developed and applied a non-linear filter (DM) that exploits the changes due to local mode filtering. An ideal mode filter outputs the most commonly occurring value within a neighborhood, causing non-step edges to become step edges. We apply our mode filter to an image and then subtract the pre-filtered image. The output has an odd intensity profile and the zero crossing is located at the edge center (figure 1).

2.1 Local Mode Filter

To simplify the discussion about the mode filter, edges are assumed to be 1-D (figure 2). Let w be the sliding window of the mode filter centered at j. x_1 and x_2 designate the endpoints of the edge profile and i_1,i_2 the intensities at these points. X is the width of the edge profile ($X = x_2 - x_1$). Let x_c be the center of the edge and M_j be the output of the mode filter at point j.

Linear edge profiles, (Figure 2a): If $w > X$, a step edge is created centered at x_c (Figure 2b):

$$\begin{aligned} M_j &= i_1 \, \forall j < x_c \\ &= i_2 \, \forall j > x_c \end{aligned}$$

If $w < X$, (Figure 2c):

$$M_j = i_1 \, \forall j < x_1 + w/2$$

0-7803-8886-0/05/$20.00 ©2005 IEEE
CCECE/CCGEI, Saskatoon, May 2005

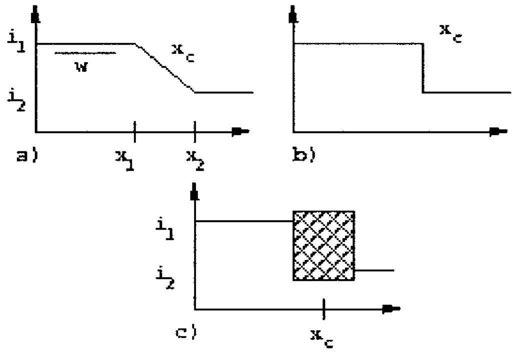

Figure 2. Mode filtering for a linear edge profile. a) Original b) Output when $w > X/2$ c) Output when $w <= X/2$. Crosshatching indicates where the mode is undefined.

$$
\begin{aligned}
&= i_2 \,\forall j > x_2 - w/2 \\
&= undefined\ otherwise.
\end{aligned}
$$

The mode filter has been considered difficult to implement because frequently the mode is not unique or does not exist. To ensure that the output of the filter is always defined, our modified mode filter outputs the median value for either of these cases. The median filter is a well-known rank-order filter [4].

2.2 DM Implementation

The original image I_{orig} is processed with our modified mode filter and then I_{orig} is subtracted resulting in I_{fin}. i.e. $I_{fin} = mode(I_{orig}) - I_{orig}$.

3 Application Considerations

The output of DM depends upon the length of the edge profile, the distance to the neighboring edge, the gradient of the edge, the noise within the image and the size of the window (neighbourhood). The magnitude of the response on either side of the zero crossing depends on the gradient of the edge and w.

To determine an appropriate window size, we first discuss an ideal image situation. Consider a single non-step edge within an infinite image of constant intensity without noise. Unlike linear filters, DM retains localization accuracy with increasing neighborhood size. In this paper, the window sizes are square and described as NxN. With increasing window size, the output reaches a limit since both the mode and median filter outputs approach stable values. However, as the window size reduces, the width of the zero crossing increases and there is a smaller response magnitude. This is caused by the area at the center of the edge profile (around x_c in Figure 2) becoming large compared to the window size.

TABLE I
PERFORMANCE OF DM VS.WINDOW (NOISY STEP EDGE).

NxN	% Match	NxN	% Match
5	13	15	70
9	58	21	88
11	66	29	91

TABLE II
THE DETECTED CENTERS OF TWO ADJACENT LINEAR PROFILE EDGES AS A FUNCTION OF WINDOW SIZE.

NxN	Zero Crossings	NxN	Zero Crossings
21	34.5, 49.5	27	36.5, 47.5
23	34.5, 49.5	29	37.5, 46.5
25	35.5, 48.5	31	42.5

Realistically, the maximum size of the window w_{max} depends upon edge density, and the minimum size of the window w_{min} depends upon noise levels. Artificially created edges with additive white Gaussian noise were used to obtain a measure of the performance of DM. A disc image with step edges is used for the first experiment. The image is 301x301, the disc center is the center of the image, and it has a radius of 50 pixels and an intensity of 50 on a zero background. Gaussian noise with zero mean and an intensity standard deviation of 50 is added. The detected edge is identified by thresholding the Sobel operator magnitude of the DM output. This edge is compared to the Sobel magnitude of the non-noisy step edge image, thresholded (at the mean + 2 standard deviations) and dilated with a 3x3 kernal. The standard is dilated because the DM output zero-crossing is greater than 1 pixel wide. Table 1 indicates the performance as a function of window size. The magnitude of Sobel operator produced a noise pattern only. Note that the computation time of DM increases with increasing window size.

Next, we investigate the detection of linear profile edges in a noisy environment. Figure 3 illustrates the linear edge and the performance of DM. Starting from the previously described disc image, linear edges were created by mean filtering the image and then Gaussian noise was added. The graph ((Figure 4) indicates that the window size providing good DM performance is dependent upon the SNR of the image. The Sobel operator could not detect the edge profile for any SNR.

In the final experiment, we investigated DM's ability to identify the center of two neighboring linear edge profiles as a function of window size. Theoretically, w_{max} is $2 * (d + X/2)$ where d is the distance between the two edges. This is confirmed with experimental results (Table 2) for which the edge centers are located at 34.5 and 49.5. The half pixel coordinates result from calculating the mid point between the peaks on either side of the edge center. In this experiment $d = 7$ and $X = 9$.

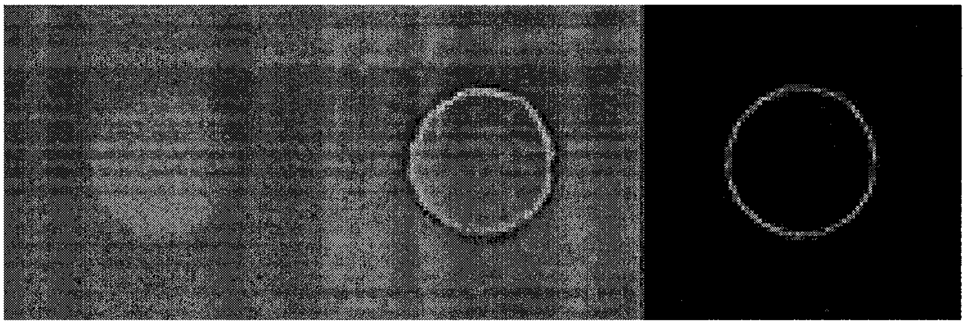

Figure 3. Typical experimental images. Left: Disc image, linear edge profiles, SNR = 1. Middle: DM output, w=21. Right: The detected edge and the true edge (found from the non-noisy step edge disc) added to indicate match.

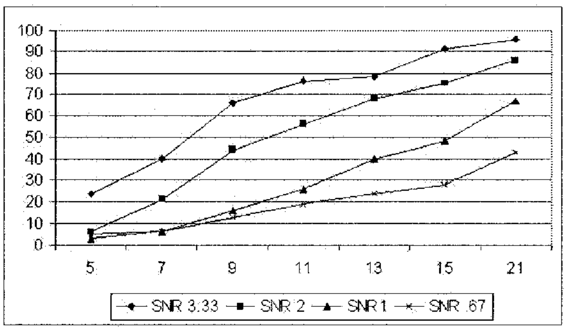

Figure 4. Percentage match vs window size over a range of SNR. Edges located from the noisy,linear-profile disc image was compared with the true edge of the step-edge disc.

Frequently there exists edge detail at many different resolutions simultaneously. Larger window sizes treat the small detail as noise. Smaller window sizes may cause edges to be missed. For images with many levels of detail, the output of the mode filter can be regarded as a step-wise approximation to the original image (Figure 5). DM is the difference between the approxmimation and the original.

We are investigating US speckle properties in the presence of film grain noise. The images are obtained with a frequency setting of 8.5 Mhz. Assuming a speed of sound in human soft tissue at 1540 m/s, the wavelength is in the order of 181 microns. It is generally accepted that US resolves structures in the order of a wavelength. The average biparietal head diameter of the gestational age under investigation is between 7 and 8 cm and our ultrasound images varied from 5000 to 6000 pixels. These numbers suggest that our image pixels correspond to a value in the order of 15 microns. To obtain a measure of the film grain noise, image samples were obtained from the ventricles of the cranial US images. The ventricles are fluid filled with a low density of scatters and provide a good approximation to the noise characteristics of the image. The histograms of the ventricle images are bell-shaped suggesting that a Gaussian noise model is reasonable. The median standard deviation of the noise intensity is 12. We sampled several speckle edge profiles to provide a measure of the intensity difference across speckles. Based on edges having a change in intensity of 50, the SNR is 4.1667. Based on the noise level of the image and the past experiments (figure 4), $w = 9$ is a reasonable choice. This choice is a compromise between accuracy and computation time. Using the equation for the maximum window size, the closest resolvable neighboring edges would have properties of $d + X/2 = 4.5$ pixels. Since in our US experiments $w/2$ corresponds to 67.5 microns, the filter highlights edge properties within the resolution of speckle edges.

4 Results

Figure 6 provides a typical sample of white matter tissue processed with DM, $w = 9$. We applied the technique to images from 17 patients (12 normal outcomes, 5 CP). DM was implemented as a processing technique prior to obtaining texture measures from white matter and a reference tissue for each patient. The final outcome is a texture measure from each of the 17 patients. The texture measures separate into two populations based on patient outcome as indicated by Mann-Whitney tests. This technique is a modified version of the method described in previously published experimental results [5].

5 Conclusions

DM can be used to identify edge centers and to enhance non-step edges in a noisy environment. Its output is dependent upon edge gradient, distance between edges, and image SNR.

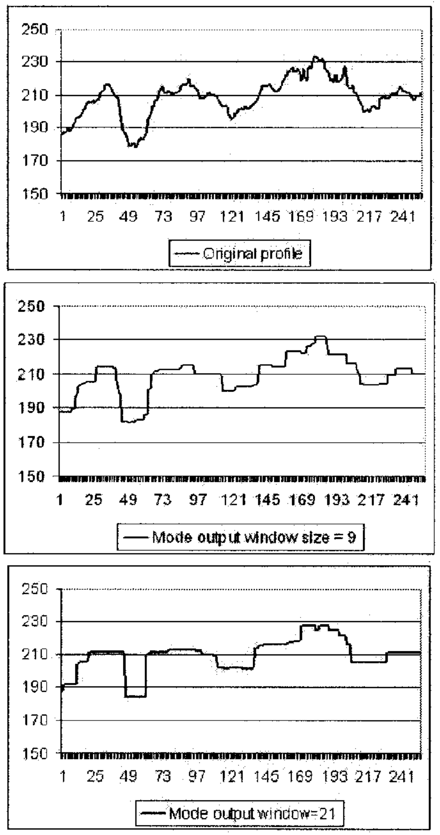

Figure 5. Comparison of original intensity profile with mode filter output for square windows sized of 9 and 21.

Acknowledgements

The authors would like to thank Dr. Matthias Schmidt, Diagnostic Imaging, and Dr. David Gaskin, Pathology, IWK Health Centre, Halifax, Nova Scotia for their assistance.

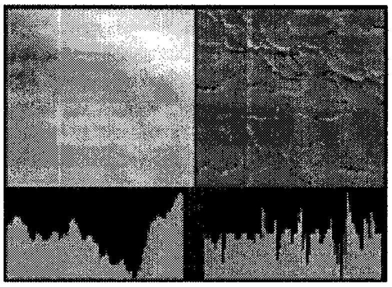

Figure 6. Portion of white matter, original and processed with DM w=9x9. The profiles along the lines in the images are shown below. Intensity ranges: 87 to 180 for original, -18 to 16 for DM.

This work was supported by the Natural Sciences and Engineering Research Council of Canada, Nova Scotia Health Research Foundation, and the IWK Children's Health Centre.

References

[1] J.M. Thijessen. " Ultrasonic speckle formation, analysis and processing applied to tissue characterization." *Patt. Rec. Lett.*, vol. 24, 659-675, 2003.

[2] Kumar V., A.K. Abbas, and N. Fausto [editors], *Robbins and Cotran Pathologic Basis of Disease.*- 7th ed. Philadelphia, Pennsylvania: Elsevier Saunders, 2005, pp 1365.

[3] L.D. Griffin " Mean, median and mode filtering of images " *Proc. Roy. Soc. Lond. A Mat.* vol. 456, 2995-3004, 2000.

[4] E.R. Davies " An Analysis of the Geometric Distortions Produced by Median and Related Image Processing Filters. " *Adv. Imag. Elect. Phys.*, vol. 126, 93-193, 2003.

[5] T. Hope, Gregson P., Linney N., Schmidt M. "Texture-based tissue characterization: A novel predictor for brain injury?" *Proc. IASTED Biomed 2004*, Innsbruck, Austria, February 2004,135-140.

Appendix F

Sample Sizes

From maskoutfilesthesis.xls

	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff
	1	2	3	4	5
	al001	bcv001	dm001	kls002	tw001
original	-0.5	-0.235	0.145	0.005	0.09
ratio=0.25	-0.92748	-0.385383	0.224767	-0.056836	0.055466
ratio=0.5	-0.6835495	-0.38547	0.4111935	-0.0499095	-0.01867
ratio=0.75	-0.751532	-0.482781	0.5106265	0.001462	-0.23726
ratio=0.85	-0.7866905	-0.690085	0.528793	-0.011436	-0.51777
shrinkwid=10times	-0.9969735	-0.239371	0.1346145	0.0040665	0.092978
shrinkwid=20times	-0.9756785	-0.246265	0.126433	0.0144915	0.10287
shrinkwid=30times	-0.9386505	-0.2612665	0.1292475	0.018469	0.107745
shrinklen=100times	-0.9750005	-0.2709265	0.087023	-0.0052945	0.075733
shrinklen=200times	-0.763537	-0.347675	0.1689065	-0.033017	0.05273
shrinklen=300times	-0.616692	-0.4287655	0.0785145	-0.044731	-0.00366
shrinklen=400times	-0.695002	-0.3832955	0.266387	-0.0417515	0.008374
shrinklen=500times		-0.342228	0.254876	-0.106087	-0.04658
shrinklen=600times		0.1274615	0.5869875	-0.153626	-0.24629
shrinklen=700times			0.5226025	0.1434265	-0.40537
shrinklen=800times					-0.43712

original size ranking	16	12	14	13
ratio=0.25, ranking	16	10	14	13
shrink 400, ranking	14	7	13	12
shrink 200, ranking	16	12	14	13
shrink width 30, ranking	16	12	14	13

	Sum of ranks,CP group	Sum of ranks Ctrl group	
original size	66	70	
ratio=0.25	65	71	
shrink 400	54	51	
shrink 200	66	70	
shrink width 30	66	70	
expected orig& ratio	42.5	93.5	
difference between actual and expected	23.5	-23.5	original
difference between actual and expected	22.5	-22.5	ratio=0.25
difference between actual and expected	23.5	-23.5	shrink length 200
difference between actual and expected	23.5	-23.5	shrink width 30
expected -- shrink length 400	37.5	67.5	
difference between actual and expected	16.5	-16.5	shrink length 400

Note: The difference between the actual and expected value in a Mann-Whitney test
doesn't vary until shrinklength =< 400

Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff	Ave diff
6	7	8	9	10	11	12	13	14	17	15	16
vam002	amt	bbc	bc	bgv	dev	epat	mec	rm	tap	sad	sm2
0.22	1.045	1.415	0.32	-0.065	0.28	0.675	0.71	0.635	1.185	0.715	1.57
0.1976365	0.948246	1.2974215	0.346021	-0.0780995	0.2223785	0.713075	0.6659305	0.5506825	1.206454	0.736302	1.570465
0.154987	0.8299855	1.306649	0.2785455	0.1300685	0.270003	0.6161085	0.599866	0.837766	1.235334	0.722373	1.636277
-0.0757055	0.601884	1.017577	0.494565	0.207071	-0.006551	0.482736	0.754111	1.1613485	1.089036	0.531324	1.666531
-0.554193	0.5918865	1.185377	0.6427425	0.028003	-0.2224225	0.4385035	0.7551525	1.2836885	0.751655	1.187542	1.813995
0.222661	1.0188045	1.3995685	0.311642	-0.0596925	0.271956	0.672615	0.709329	1.006252	1.182453	0.720906	1.6004
0.2286065	0.9765635	1.371692	0.312313	-0.0505955	0.262474	0.691952	0.7116185	1.006232	1.175329	0.724057	1.61649
0.2222695	0.9663635	1.347912	0.3161045	-0.051517	0.2434	0.698455	0.702313	1.001936	1.174927	0.718921	1.624546
0.2035875	1.0185385	1.3643365	0.322368	-0.0595605	0.237543	0.6718195	0.670717	0.6179955	1.17906	0.732606	1.575701
0.179825	0.9497645	1.332391	0.320671	-0.0799335	0.198554	0.7008975	0.66778	0.552163	1.133593	0.716104	1.515039
0.1841595	0.980791	1.187475	0.2838445	-0.0455385	0.1845085	0.671536	0.6161025	0.562028	1.166363	0.863715	
0.230182	0.8306345	0.9682475	0.2044565	0.1792695	0.2071435	0.686743	0.585463	0.7166535	1.127119		
0.064353	0.772347	1.7049315	0.228417	0.0954765	0.0719145	0.670651	0.5714265	0.872951	1.002234		
-0.12916	0.741965	0.5560425		0.2273525	-0.049504	0.6714575	0.608093	0.9707395	0.650251		
-0.0992055	0.5183715	0.490523		-0.028435	-0.132522	0.623028	0.719796	1.1122915	0.615926		
		0.648158				0.61107	0.7729435	1.356846	0.671868		
11	4	2	9	15	10	7	6	8	3	5	1
12	4	2	9	15	11	6	7	8	3	5	1
8	3	2	10	11	9	5	6	4	1		
11	4	2	9	15	10	6	7	8	3	5	1
11	5	2	9	15	10	8	7	4	3	6	1

Appendix G

Sample Mann-Whitney Tests on EXP1 Measures

proc.im	pat.	mask	avg NSA	avgstdev	avgmean	avgmax	avgmin	avgsk	avgkurt	avgmed
amtcdifnsa	amtcd	chf	71.337837	153.089752	-352.003967	364	-666	0.8062	1.194	-378
bbccdifnsa	bbccd	chf	81.856071	194.146729	-425.24823	314	-754	0.6927	0.0569	-456
bccdifnsa	bccdi	chf	72.682129	160.105988	-202.775604	467	-519	0.754	0.4835	-227
bgv001cdifnsa	bgv00	chf	143.355972	298.960876	40.166729	1035	-601	0.4409	-0.056	13
devcdifnsa	devcd	chf	83.379234	175.829819	-412.151917	141	-748	0.3334	-0.451	-426
epatcdifnsa	epatc	chf	111.481491	227.964569	-150.296844	672	-592	0.6777	0.28	-179
meccdifnsa	meccd	chf	73.618774	127.678375	-319.580872	288	-632	0.8229	1.2106	-337
rmcdifnsa	rmcdi	chf	94.08046	205.370743	-243.146042	620	-659	0.9214	1.2152	-272
sadcdifnsa	sadcd	chf	74.839417	149.986267	-424.609894	232	-724	0.7044	0.4931	-448
sdccdifnsa	sdccd	chf	107.653305	213.259659	-304.740967	415	-774	0.3742	-0.266	-325
sm2cdifnsa	sm2cd	chf	117.447891	186.025055	-99.268875	468	-466	0.4558	-0.278	-114
tapcdifnsa	tapcd	chf	103.807388	208.049622	-615.01355	646	-960	1.4616	4.7504	-651
bcv001cdifnsa	bcv00	chf	138.589493	293.661041	-102.412254	906	-724	0.6957	0.3993	-145
dm002cdifnsa	dm002	chf	116.13546	267.183075	-322.311005	784	-841	0.8345	0.7746	-358
kls002cdifnsa	kls00	chf	105.065201	193.791138	-155.190201	597	-589	0.5603	-0.001	-176
tw001cdifnsa	tw001	chf	147.206421	231.990738	-185.478882	687	-704	0.5547	0.603	-206
vam002cdifnsa	vam00	chf	111.623764	211.211761	-103.318443	746	-524	0.5759	0.2003	-118

		rank	rank	rank	rank	rank	rank	rank	rank
		avg NSA	avgstdev	avgmean	avgmax	avgmin	avgsk	avgkurt	avgmed
		17	15	13	13	9	5	4	13
		13	10	16	14	14	9	12	16
		16	14	8	11	2	6	8	8
		2	1	1	1	6	15	14	1
		12	13	14	17	13	17	17	14
		7	5	5	6	5	10	10	6
		15	17	11	15	7	4	3	11
		11	9	9	8	8	2	2	9
		14	16	15	16	11	7	7	15
		8	6	10	12	15	16	15	10
		4	12	2	10	1	14	16	2
		10	8	17	7	17	1	1	17
		3	2	3	2	11	8	9	4
		5	3	12	3	16	3	5	12
		9	11	6	9	4	12	13	5
		1	4	7	5	10	13	6	7
		6	7	4	4	3	11	11	3
sum of ranks - normal		129	126	121	130	108	106	109	122
sum of ranks - cp		24	27	32	23	44	47	44	31
expected sum - nor	108								
expected sum - cp	45								
difference normal		-21	-18	-13	-22	0	2	-1	-14
difference cp		21	18	13	22	1	-2	1	14

The critical value of U is 11 for N1=5 and N2=12
So all measures but minimum, skewness and kurtosis make these two
outcome groups separate into two populations.

Appendix H

Portion of randomForest Manual

The randomForest Package

January 24, 2006

Title Breiman and Cutler's random forests for classification and regression

Version 4.5-16

Date 2006-01-23

Depends R (>= 1.8.1)

Author Fortran original by Leo Breiman and Adele Cutler, R port by Andy Liaw and Matthew Wiener.

Description Classification and regression based on a forest of trees using random inputs.

Maintainer Andy Liaw <andy_liaw@merck.com>

License GPL version 2 or later

URL http://stat-www.berkeley.edu/users/breiman/RandomForests

R topics documented:

randomForest	*Classification and Regression with Random Forest*

Description

randomForest implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

Usage

```
## S3 method for class 'formula':
randomForest(formula, data=NULL, ..., subset, na.action=na.fail)
## Default S3 method:
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity=FALSE, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
             keep.inbag=FALSE, ...)
## S3 method for class 'randomForest':
print(x, ...)
```

Arguments

data	an optional data frame containing the variables in the model. By default the variables are taken from the environment which randomForest is called from.
subset	an index vector indicating which rows should be used. (NOTE: If given, this argument must be named.)
na.action	A function to specify the action to be taken if NAs are found. (NOTE: If given, this argument must be named.)
x, formula	a data frame or a matrix of predictors, or a formula describing the model to be fitted (for the print method, an randomForest object).
y	A response vector. If a factor, classification is assumed, otherwise regression is assumed. If omitted, randomForest will run in unsupervised mode.
xtest	a data frame or matrix (like x) containing predictors for the test set.
ytest	response for the test set.
ntree	Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.
mtry	Number of variables randomly sampled as candidates at each split. Note that the default values are different for classification (sqrt(p) where p is number of variables in x) and regression (p/3)
replace	Should sampling of cases be done with or without replacement?

`classwt` Priors of the classes. Need not add up to one. Ignored for regression.

`cutoff` (Classification only) A vector of length equal to number of classes. The 'winning' class for an observation is the one with the maximum ratio of proportion of votes to cutoff. Default is 1/k where k is the number of classes (i.e., majority vote wins).

`strata` A (factor) variable that is used for stratified sampling.

`sampsize` Size(s) of sample to draw. For classification, if sampsize is a vector of the length the number of strata, then sampling is stratified by strata, and the elements of sampsize indicate the numbers to be drawn from the strata.

`nodesize` Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`importance` Should importance of predictors be assessed?

`localImp` Should casewise importance measure be computed? (Setting this to `TRUE` will override `importance`.)

`nPerm` Number of times the OOB data are permuted per tree for assessing variable importance. Number larger than 1 gives slightly more stable estimate, but not very effective. Currently only implemented for regression.

`proximity` Should proximity measure among the rows be calculated?

`oob.prox` Should proximity be calculated only on "out-of-bag" data?

`norm.votes` If `TRUE` (default), the final result of votes are expressed as fractions. If `FALSE`, raw vote counts are returned (useful for combining results from different runs). Ignored for regression.

`do.trace` If set to `TRUE`, give a more verbose output as `randomForest` is run. If set to some integer, then running output is printed for every `do.trace` trees.

`keep.forest` If set to `FALSE`, the forest will not be retained in the output object. If `xtest` is given, defaults to `FALSE`.

`corr.bias` perform bias correction for regression? Note: Experimental. Use at your own risk.

`keep.inbag` Should an `n` by `ntree` matrix be returned that keeps track of which samples are "in-bag" in which trees (but not how many times, if sampling with replacement)

`...` optional parameters to be passed to the low level function `randomForest.default`.

Value

An object of class `randomForest`, which is a list with the following components:

`call` the original call to `randomForest`

`type` one of `regression`, `classification`, or unsupervised.

`predicted` the predicted values of the input data based on out-of-bag samples.

`importance` a matrix with `nclass` + 2 (for classification) or two (for regression) columns. For classification, the first `nclass` columns are the class-specific measures computed as mean descrease in accuracy. The `nclass` + 1st column is the mean descrease in accuracy over all classes. The last column is the mean decrease in Gini index. For Regression, the first column is the mean decrease in accuracy and the second the mean decrease in MSE. If `importance=FALSE`, the last measure is still returned as a vector.

`importanceSD` The "standard errors" of the permutation-based importance measure. For classification, a `p` by `nclass + 1` matrix corresponding to the first `nclass + 1` columns of the importance matrix. For regression, a length `p` vector.

`localImp` a `p` by `n` matrix containing the casewise importance measures, the [i,j] element of which is the importance of i-th variable on the j-th case. `NULL` if `localImp=FALSE`.

`ntree` number of trees grown.

`mtry` number of predictors sampled for spliting at each node.

`forest` (a list that contains the entire forest; `NULL` if `randomForest` is run in unsupervised mode or if `keep.forest=FALSE`.

`err.rate` (classification only) vector error rates of the prediction on the input data, the i-th element being the error rate for all trees up to the i-th.

`confusion` (classification only) the confusion matrix of the prediction.

`votes` (classification only) a matrix with one row for each input data point and one column for each class, giving the fraction or number of 'votes' from the random forest.

`oob.times` number of times cases are 'out-of-bag' (and thus used in computing OOB error estimate)

`proximity` if `proximity=TRUE` when `randomForest` is called, a matrix of proximity measures among the input (based on the frequency that pairs of data points are in the same terminal nodes).

`mse` (regression only) vector of mean square errors: sum of squared residuals divided by `n`.

`rsq` (regression only) "pseudo R-squared": 1 - `mse` / Var(y).

`test` if test set is given (through the `xtest` or additionally `ytest` arguments), this component is a list which contains the corresponding `predicted`, `err.rate`, `confusion`, `votes` (for classification) or `predicted`, `mse` and `rsq` (for regression) for the test set. If `proximity=TRUE`, there is also a component, `proximity`, which contains the proximity among the test set as well as proximity between test and training data.

Note

The `forest` structure is slightly different between classification and regression. For details on how the trees are stored, see the help page for `getTree`.

If `xtest` is given, prediction of the test set is done "in place" as the trees are grown. If `ytest` is also given, and `do.trace` is set to some positive integer, then for every `do.trace` trees, the test set error is printed. Results for the test set is returned in the `test` component of the resulting `randomForest` object. For classification, the `votes` component (for training or test set data) contain the votes the cases received for the classes. If `norm.votes=TRUE`, the fraction is given, which can be taken as predicted probabilities for the classes.

For large data sets, especially those with large number of variables, calling `randomForest` via the formula interface is not advised: There may be too much overhead in handling the formula.

The "local" (or casewise) variable importance is computed as follows: For classification, it is the increase in percent of times a case is OOB and misclassified when the variable is permuted. For regression, it is the average increase in squared OOB residuals when the variable is permuted.

Author(s)

Andy Liaw ⟨andy_liaw@merck.com⟩ and Matthew Wiener ⟨matthew_wiener@merck.com⟩, based on original Fortran code by Leo Breiman and Adele Cutler.

References

Breiman, L. (2001), *Random Forests*, Machine Learning 45(1), 5-32.

Breiman, L (2002), "Manual On Setting Up, Using, And Understanding Random Forests V3.1", `http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.1.pdf`.

See Also

`predict.randomForest`, `varImpPlot`

Examples

```
## Classification:
##data(iris)
set.seed(71)
iris.rf <- randomForest(Species ~ ., data=iris, importance=TRUE,
                        proximity=TRUE)
print(iris.rf)
## Look at variable importance:
round(importance(iris.rf), 2)
## Do MDS on 1 - proximity:
iris.mds <- cmdscale(1 - iris.rf$proximity, eig=TRUE)
op <- par(pty="s")
pairs(cbind(iris[,1:4], iris.mds$points), cex=0.6, gap=0,
      col=c("red", "green", "blue")[as.numeric(iris$Species)],
      main="Iris Data: Predictors and MDS of Proximity Based on RandomForest")
par(op)
print(iris.mds$GOF)

## The `unsupervised' case:
set.seed(17)
iris.urf <- randomForest(iris[, -5])
MDSplot(iris.urf, iris$Species)

## Regression:
## data(airquality)
set.seed(131)
ozone.rf <- randomForest(Ozone ~ ., data=airquality, mtry=3,
                         importance=TRUE, na.action=na.omit)
print(ozone.rf)
## Show "importance" of variables: higher value mean more important:
round(importance(ozone.rf), 2)
```

`rfImpute` *Missing Value Imputations by randomForest*

Description

Impute missing values in predictor data using proximity from randomForest.

Appendix I

Sample Script Files

dm9m37r2001

```
/* Script file to process with the DM method */

load /home/thope/exp2/m37r2/001-1m37r2.img 1

setroi 400 1400 500 1500 1  /* only process minimum area to contain all ROIs */

crop 1 10 1

form 1 0 1 1 1

mode5 10 11 9   /* the last input to mode5 is N in the NxN window */

subtract 11 10 12

save /disc15/tmp/thope/m37r2/001m37r2dm9.img 12

quit
```

gat19m37r2001

```
/* script file to variably convolve the noise and resolution reduced           */
/* image with the gabt19 kernel.                                                */

load /home/thope/exp2/m37r2/001-1m37r2.img 1

load /home/thope/exp2/001-1r2ang.img 2

load /home/thope/exp2/gabt19.img 3

scale 2 4 10 0          /* scale the angle image by 10 as vconv2 was designed   */
                        /*      for anglex10 images */

scale 3 6 .1 0          /* scale the gabt19 kernel to prevent numerical overflow*/

setroi 400 1400 500 1500 1      /* crop the image to the minimum size required  */
                                /*      to contain all ROIs                     */
crop 1 10 1

crop 4 5 1

form 1 0 1 1 1

form 4 0 1 1 1

vconv2 10 5 41 6 12
/* the number 41 is the length of the Gabor kernel, the width is assumed to be 1 */

save /disc15/tmp/thope/m37r2/001m37r2gabt19.img 12

quit
```

idd30

```
disc 10 61 61 0 0 30 2
addconstant 10 31 -1 0
load 005-1med37.img 1
gdilate 1 31 32
subtract 1 32 33
reduce 33 34
reduce 34 35
save 005-1m37idd30r2.img 35
```

g19cv17a

```
#!/usr/bin/tcsh
#
#$ -S /bin/tcsh
#
#$ -cwd
#
#Mail options
#$ -M tahope@dal.ca
#$ -m ea
#
#Where to place output and errors
#$ -o /home/thope/g19cv17aout.log
#$ -e /home/thope/g19cv17aerr.log
#
#Set the jobname in queue listings
#$ -N g19cv17a
#
cp /home/thope/exp2/gat19m37r2001.scr /home/thope/cvlab17/testimage17.scr
cd /home/thope/cvlab17
/home/thope/cvlab17/cv
cp /home/thope/exp2/gat19m37r2002.scr /home/thope/cvlab17/testimage17.scr
cd /home/thope/cvlab17
/home/thope/cvlab17/cv
cp /home/thope/exp2/gat19m37r2003.scr /home/thope/cvlab17/testimage17.scr
cd /home/thope/cvlab17
/home/thope/cvlab17/cv
cp /home/thope/exp2/gat19m37r2004.scr /home/thope/cvlab17/testimage17.scr
cd /home/thope/cvlab17
/home/thope/cvlab17/cv
cp /home/thope/exp2/gat19m37r2005.scr /home/thope/cvlab17/testimage17.scr
cd /home/thope/cvlab17
/home/thope/cvlab17/cv

/* The first step is to place the script file of choice within the script       */
/* file associated with the cvlab directory, as each call to cv on Pluto         */
/* automatically starts the script file.                                         */
                                                                        */
```

```
                                   ctexmap071m37n
load /home/thope/exp2/m37r2/071-1m37r2.img 1

load /home/thope/exp2/071-1r2ang.img 24

setroi 595 995 545 1365 1
/* The images were cropped to the minimum area containing all ROIs      */

crop 24 34 1

form 24 0 1 1 1

crop 1 2 1

form 1 0 1 1 1

load /home/thope/exp2/071-1rtr2chf.img 15

crop 15 25 1

form 15 0 1 1 1

load /home/thope/exp2/071-1rtr2w.img 23

crop 23 22 1

form 23 0 1 1 1

texmapnsa2 2 1 34 25 22 7 8 20 5 100 100

save /disc15/tmp/thope/m37r2/071-1rtm37r2cratnsa.img 7

save /disc15/tmp/thope/m37r2/071-1rtm37r2cdifnsa.img 8

form 7 0 10 10 1

form 8 0 10 10 1

form 2 0 10 10 1

form 34 0 10 10 1 /* The function "release" caused some difficulty on Pluto.    */

form 22 0 10 10 1 /* So, form was used as an alternative to release             */

form 21 0 10 10 1

form 25 0 10 10 1

form 35 0 1 1 1

form 36 0 1 1 1

form 37 0 1 1 1

form 38 0 1 1 1

form 39 0 1 1 1

form 30 0 1 1 1

load /home/thope/exp2/m37r2/071-1m37r2.img 1

load /home/thope/exp2/071-1r2ang.img 24
```

ctexmap071m37n

```
setroi 955 1315 475 1315 1

crop 24 34 1

form 24 0 1 1 1

crop 1 2 1

form 1 0 1 1 1

load /home/thope/exp2/071-1lftr2chf.img 15

crop 15 25 1

form 15 0 1 1 1

load /home/thope/exp2/071-1lftr2w.img 23

crop 23 22 1

form 23 0 1 1 1

texmapnsa2 2 2 34 25 22 7 8 20 5 100 100

save /disc15/tmp/thope/m37r2/071-1lftm37r2cratnsa.img 7

save /disc15/tmp/thope/m37r2/071-1lftm37r2cdifnsa.img 8

quit

/* The function texmapnsa2 created the ratio and difference maps using NSA as    */
/* the measure.                                                                   */
/* The parts of the call to texmapnsa2:                                           */
/* first number: input image                                                      */
/* second number: side of patient image, with 1=right and 2=left                 */
/* third number: angle look-up image, as the small sample under investigation    */
/*               was rotated                                                      */
/* fourth number: image with mask of tissue under investigation                   */
/* fifth number: image with mask of the reference tissue                          */
/* sixth and seventh numbers: the output images                                   */
/* eighth and nineth numbers: the size of the samples under investigation in     */
/*                            the primary tissue                                  */
/* tenth and eleventh numbers:  the multiplier of the values obtained in the     */
/*                            comparison                                          */
```

Appendix J

Data Extraction

Script file exp2m37g19c.scr

```
form 0 1 13 13 1
load 001-1lftm37r2gabt19cratsd.img 1
/* load the left side Gabor filtered choroid plexus ratio map, STDV compare*/

load 001-1lftm37r2gabt19cratsd.img 2

load 001-1r2d.img 9
/* mask of the signal area of the patient's image */

binary 9 10 1

binary 2 21 1
/* image 2 forms the left choroid plexus mask after thresholding*/

setroi 400 1400 500 1500 1
/* image crop used for the Gabor t=19 convolution */

setroi 505 895 215 915 2
/* image region which includes choroid and white mater ROIs */

crop 10 12 1

crop 12 11 2

erode 11 0 16
/* shrink the size of the signal mask slightly */

mask 21 16 22
/* make sure the choroid mask is completely within the signal area,
this step is more relevant for the white matter masks,
as some exceed the signal boundaries */

load 001-1rtm37r2gabt19cratsd.img 3
/* load the right side, Gabor filtered choroid plexus ratio map, STDV compare*/

load 001-1rtm37r2gabt19cratsd.img 4

binary 4 25 1

setroi 135 455 215 825 2
/* image region which includes choroid and white mater ROIs */

crop 10 12 1

crop 12 11 2

erode 11 0 16

mask 25 16 24

maskinfoavg 1 22 3 24 exp2m37r2gabt19c.txt 001-1m37r2gabt19cratsd chf

/* maskinfoavg calculates the stats and NSA on each side, averages the values,
and then places the output into file exp2m37r2gabt19c.txt.
The input consists of the 4 input images, the output text file name, map name,
and a designation for the region name under investigation. */
```

Appendix K

Variable Selection and EXP1 Data Set

After the pair of measures were selected from Group A, measures
with low correlation to either were identified from the 64.
The value of .3 was chosen as the cut-off

	Group A, 1	Group A, 2	
Group B,1	0.295	0.699	
Group B,2	-0.19	0.834	
Group B,3	-0.16	-0.16	
Group B,4	0.141	0.141	
Group B,5	0.098	0.098	
Group B,6	0.182	0.182	
Group B,7	0.151	0.151	
Group B,8	-0.027	-0.027	lowest correlation
Group B,9	0.172	0.172	
Group B,10	-0.095	0.095	
Group B,11	-0.144	0.144	
Group B,12	0.05	0.05	

Group B,1 and Group B,2 discarded due to their high correlation with Group A,2

	Group A,1	Group A, 2	Group B,8	
Group B,3	-0.16	-0.16	-0.42	less than .5
Group B,4	0.141	0.141	0.779	
Group B,5	0.098	0.098	0.916	
Group B,6	0.182	0.182	0.89	
Group B,7	0.151	0.151	0.743	
Group B,9	0.172	0.172	0.765	
Group B,10	-0.095	0.095	0.347	less than .5
Group B,11	-0.144	0.144	0.596	
Group B,12	0.05	0.05	0.895	

Result: Group A,1; Group A,2; Group B,8; Group B,3; Group B,10

Designations in spreadsheet containing Pearson correlations

Group A,1	C138
Group A,2	C19
Group B,1	C96
Group B,2	C24
Group B,3	C48
Group B,4	C81
Group B,5	C82
Group B,6	C90
Group B,7	C137

Group B,8	C146
Group B,9	C202
Group B,10	C209
Group B,11	C210
Group B,12	C2

Pearson Correlation Coefficient

	C1	C2	C3
C2	0.865		
C3	0.597	0.483	
C8	0.58	0.451	0.999
C17	0.948	0.864	0.76
C18	0.832	0.95	0.677
C19	0.16	0.05	0.624
C20	0.474	0.514	0.676
C21	-0.172	-0.316	0.359
C22	-0.244	-0.101	-0.426
C23	-0.401	-0.322	-0.534
C24	0.15	-0.073	0.553
C41	0.357	0.474	-0.472
C42	0.519	0.627	-0.095
C43	-0.549	-0.522	-0.697
C48	-0.569	-0.543	-0.693
C49	0.468	0.471	-0.382
C50	0.606	0.711	-0.072
C51	-0.755	-0.627	-0.975
C56	-0.765	-0.64	-0.971
C65	0.507	0.538	-0.328
C66	0.552	0.662	-0.059
C67	-0.69	-0.601	-0.98
C72	-0.694	-0.609	-0.977
C81	0.958	0.91	0.658
C82	0.777	0.948	0.577
C83	0.505	0.443	0.982
C88	0.475	0.392	0.979
C89	0.961	0.89	0.735
C90	0.781	0.934	0.659
C91	0.477	0.407	0.967
C96	0.437	0.333	0.958
C121	0.455	0.492	-0.392
C122	0.578	0.672	-0.06
C123	-0.758	-0.735	-0.926
C128	-0.771	-0.75	-0.919
C137	0.939	0.819	0.551
C138	0.816	0.929	0.426
C139	0.5	0.373	0.962
C144	0.455	0.363	0.948
C145	0.904	0.897	0.495
C146	0.676	0.895	0.352
C147	0.513	0.389	0.945
C152	0.442	0.18	0.858
C177	0.477	0.474	-0.362
C178	0.545	0.653	-0.135
C179	-0.671	-0.532	-0.956
C184	-0.665	-0.589	-0.941
C185	0.46	0.555	-0.309
C186	0.433	0.603	-0.064
C187	-0.652	-0.554	-0.949
C192	-0.548	-0.652	-0.766
C201	0.909	0.805	0.632
C202	0.816	0.839	0.658
C203	0.631	0.554	0.954
C208	0.614	0.536	0.945
C209	0.666	0.467	0.417
C210	0.62	0.606	0.516
C211	0.621	0.545	0.929
C216	0.607	0.531	0.904
C225	0.532	0.483	-0.31
C226	0.66	0.692	-0.047
C227	-0.683	-0.616	-0.953
C232	-0.712	-0.652	-0.942
C249	0.511	0.473	-0.312
C250	0.648	0.702	-0.033
C251	-0.746	-0.64	-0.933
C256	-0.766	-0.666	-0.924

	C8
C17	0.743
C18	0.65
C19	0.631
C24	0.567
C41	-0.5
C42	-0.118
C43	-0.692
C48	-0.687
C49	-0.404
C50	-0.097
C51	-0.97
C56	-0.965
C65	-0.35
C66	-0.079
C67	-0.975
C72	-0.971
C81	0.64
C82	0.55
C83	0.981
C88	0.981
C89	0.716
C90	0.632
C91	0.967
C96	0.964
C121	-0.416
C122	-0.085
C123	-0.914
C128	-0.907
C137	0.533
C138	0.397
C139	0.963
C144	0.947
C145	0.471
C146	0.318
C147	0.946
C152	0.868
C177	-0.385
C178	-0.16
C179	-0.953
C184	-0.936
C185	-0.335
C186	-0.09
C187	-0.944
C192	-0.751
C201	0.62
C202	0.642
C203	0.952
C208	0.944
C209	0.404
C210	0.498
C211	0.928
C216	0.904
C225	-0.328
C226	-0.065
C227	-0.947
C232	-0.935
C249	-0.33
C250	-0.052
C251	-0.929
C256	-0.92

	C17	C18	C19	C24
C18	0.905			
C24	0.309	0.119	0.834	
C41	0.169	0.242	-0.604	-0.587
C42	0.404	0.452	-0.686	-0.553
C43	-0.577	-0.612	-0.185	-0.013
C48	-0.592	-0.625	-0.16	0.001
C49	0.217	0.243	-0.528	-0.554
C50	0.47	0.551	-0.502	-0.451
C51	-0.88	-0.778	-0.551	-0.503
C56	-0.888	-0.788	-0.542	-0.502
C65	0.273	0.312	-0.52	-0.537
C66	0.431	0.524	-0.442	-0.361
C67	-0.846	-0.775	-0.58	-0.5
C72	-0.849	-0.783	-0.574	-0.502
C81	0.946	0.906	0.141	0.075
C82	0.826	0.965	0.098	-0.078
C83	0.702	0.659	0.638	0.529
C88	0.673	0.611	0.65	0.551
C89	0.991	0.915	0.244	0.236
C90	0.862	0.982	0.182	0.014
C91	0.68	0.629	0.693	0.551
C96	0.641	0.557	0.699	0.581
C121	0.218	0.26	-0.561	-0.557
C122	0.442	0.526	-0.43	-0.375
C123	-0.901	-0.861	-0.486	-0.402
C128	-0.909	-0.874	-0.474	-0.393
C137	0.882	0.786	0.151	0.06
C138	0.791	0.872	-0.017	-0.19
C139	0.675	0.58	0.687	0.573
C144	0.637	0.568	0.705	0.57
C145	0.855	0.855	0.086	-0.049
C146	0.69	0.864	-0.027	-0.218
C147	0.685	0.593	0.701	0.574
C152	0.561	0.393	0.628	0.598
C177	0.239	0.252	-0.528	-0.548
C178	0.406	0.488	-0.544	-0.499
C179	-0.81	-0.699	-0.621	-0.519
C184	-0.821	-0.751	-0.586	-0.463
C185	0.266	0.355	-0.551	-0.576
C186	0.346	0.477	-0.509	-0.486
C187	-0.81	-0.723	-0.615	-0.513
C192	-0.71	-0.792	-0.545	-0.283
C201	0.881	0.805	0.212	0.12
C202	0.84	0.872	0.172	-0.004
C203	0.763	0.728	0.583	0.424
C208	0.75	0.712	0.623	0.451
C209	0.589	0.413	-0.095	0.048
C210	0.618	0.599	-0.144	-0.127
C211	0.748	0.725	0.632	0.43
C216	0.733	0.71	0.68	0.456
C225	0.295	0.273	-0.442	-0.457
C226	0.51	0.542	-0.436	-0.421
C227	-0.819	-0.774	-0.541	-0.398
C232	-0.841	-0.802	-0.509	-0.369
C249	0.271	0.271	-0.435	-0.486
C250	0.509	0.568	-0.452	-0.441
C251	-0.854	-0.782	-0.521	-0.409
C256	-0.871	-0.802	-0.499	-0.392

Pearson Correlation Coefficient

	C41	C42	C43
C42	0.769		
C43	0.178	-0.16	
C48	0.151	-0.201	0.999
C49	0.925	0.704	0.054
C50	0.788	0.871	-0.126
C51	0.29	-0.063	0.717
C56	0.275	-0.084	0.713
C65	0.907	0.757	0.008
C66	0.69	0.829	-0.122
C67	0.338	-0.007	0.701
C72	0.33	-0.025	0.695
C81	0.298	0.544	-0.629
C82	0.304	0.529	-0.626
C83	-0.515	-0.155	-0.66
C88	-0.56	-0.189	-0.64
C89	0.226	0.484	-0.605
C90	0.227	0.462	-0.661
C91	-0.536	-0.232	-0.661
C96	-0.6	-0.274	-0.636
C121	0.943	0.757	0.084
C122	0.741	0.813	-0.088
C123	0.2	-0.149	0.725
C128	0.178	-0.177	0.718
C137	0.391	0.514	-0.475
C138	0.511	0.65	-0.466
C139	-0.515	-0.196	-0.594
C144	-0.523	-0.225	-0.585
C145	0.468	0.573	-0.483
C146	0.507	0.588	-0.406
C147	-0.48	-0.195	-0.591
C152	-0.499	-0.249	-0.564
C177	0.92	0.711	0.081
C178	0.804	0.859	-0.045
C179	0.341	0.036	0.63
C184	0.311	-0.005	0.603
C185	0.889	0.762	0.05
C186	0.645	0.778	-0.067
C187	0.338	0.031	0.633
C192	0.171	-0.02	0.504
C201	0.23	0.423	-0.704
C202	0.178	0.404	-0.789
C203	-0.385	-0.038	-0.782
C208	-0.405	-0.072	-0.741
C209	0.261	0.481	-0.383
C210	0.21	0.528	-0.572
C211	-0.356	-0.074	-0.77
C216	-0.364	-0.118	-0.729
C225	0.876	0.674	0.025
C226	0.736	0.786	-0.237
C227	0.315	-0.01	0.785
C232	0.272	-0.056	0.787
C249	0.865	0.645	0.026
C250	0.744	0.805	-0.238
C251	0.262	-0.068	0.785
C256	0.23	-0.099	0.78

	C48	C49	C50	C51
C49	0.032			
C50	-0.156	0.819		
C51	0.719	0.194	-0.102	
C56	0.716	0.179	-0.126	0.999
C65	-0.017	0.976	0.882	0.138
C66	-0.153	0.715	0.955	-0.097
C67	0.699	0.273	-0.044	0.986
C72	0.695	0.267	-0.065	0.983
C81	-0.647	0.387	0.641	-0.79
C82	-0.639	0.313	0.645	-0.678
C83	-0.653	-0.462	-0.136	-0.936
C88	-0.632	-0.504	-0.188	-0.925
C89	-0.624	0.28	0.538	-0.86
C90	-0.672	0.223	0.56	-0.75
C91	-0.65	-0.482	-0.2	-0.919
C96	-0.624	-0.538	-0.261	-0.902
C121	0.059	0.987	0.861	0.205
C122	-0.117	0.765	0.95	-0.096
C123	0.73	0.15	-0.202	0.965
C128	0.725	0.127	-0.236	0.962
C137	-0.494	0.509	0.61	-0.696
C138	-0.487	0.541	0.741	-0.561
C139	-0.587	-0.433	-0.169	-0.915
C144	-0.576	-0.465	-0.209	-0.889
C145	-0.5	0.557	0.723	-0.639
C146	-0.42	0.486	0.737	-0.465
C147	-0.583	-0.403	-0.171	-0.907
C152	-0.553	-0.365	-0.231	-0.817
C177	0.059	0.992	0.836	0.174
C178	-0.074	0.832	0.991	-0.034
C179	0.628	0.245	-0.013	0.959
C184	0.602	0.231	-0.09	0.946
C185	0.027	0.932	0.933	0.136
C186	-0.092	0.646	0.923	-0.055
C187	0.631	0.262	-0.03	0.952
C192	0.502	0.151	-0.145	0.768
C201	-0.719	0.335	0.47	-0.762
C202	-0.798	0.24	0.461	-0.757
C203	-0.779	-0.275	-0.011	-0.947
C208	-0.738	-0.293	-0.041	-0.936
C209	-0.403	0.329	0.37	-0.509
C210	-0.586	0.231	0.439	-0.578
C211	-0.764	-0.25	-0.017	-0.921
C216	-0.722	-0.25	-0.041	-0.899
C225	0.002	0.969	0.796	0.112
C226	-0.264	0.814	0.906	-0.147
C227	0.781	0.221	-0.065	0.963
C232	0.785	0.178	-0.115	0.963
C249	0.006	0.973	0.809	0.124
C250	-0.264	0.811	0.938	-0.153
C251	0.786	0.156	-0.107	0.964
C256	0.782	0.126	-0.143	0.963

	C56
C56	
C65	0.118
C66	-0.128
C67	0.983
C72	0.982
C81	-0.802
C82	-0.689
C83	-0.928
C88	-0.917
C89	-0.87
C90	-0.758
C91	-0.909
C96	-0.892
C120	0.965
C121	0.186
C122	-0.123
C123	0.967
C128	0.966
C137	-0.705
C138	-0.573
C139	-0.908
C144	-0.881
C145	-0.65
C146	-0.475
C147	-0.899
C152	-0.808
C177	0.159
C178	-0.057
C179	0.956
C184	0.945
C185	0.117
C186	-0.078
C187	0.949
C192	0.771
C201	-0.768
C202	-0.759
C203	-0.942
C208	-0.931
C209	-0.508
C210	-0.574
C211	-0.917
C216	-0.896
C225	0.096
C226	-0.168
C227	0.957
C232	0.958
C249	0.108
C250	-0.174
C251	0.961
C256	0.961

Pearson Correlation Coefficient

	C65	C66	C67	C72
C121	0.99	0.785	0.281	0.269
C122	0.847	0.966	-0.059	-0.09
C123	0.074	-0.192	0.961	0.96
C128	0.045	-0.235	0.959	0.961
C137	0.539	0.526	-0.625	-0.627
C138	0.595	0.656	-0.519	-0.525
C139	-0.391	-0.167	-0.924	-0.919
C144	-0.422	-0.2	-0.905	-0.9
C145	0.607	0.647	-0.587	-0.593
C146	0.544	0.656	-0.454	-0.461
C147	-0.376	-0.185	-0.911	-0.904
C152	-0.357	-0.242	-0.794	-0.786
C177	0.971	0.723	0.251	0.245
C178	0.887	0.932	0.021	0.002
C179	0.199	0.006	0.946	0.943
C184	0.162	-0.078	0.947	0.947
C185	0.959	0.861	0.188	0.173
C186	0.742	0.919	-0.039	-0.066
C187	0.21	-0.005	0.943	0.939
C192	0.075	-0.145	0.809	0.817
C201	0.375	0.431	-0.707	-0.705
C202	0.279	0.403	-0.746	-0.741
C203	-0.24	-0.02	-0.949	-0.943
C208	-0.257	-0.043	-0.937	-0.932
C209	0.283	0.248	-0.432	-0.424
C210	0.207	0.303	-0.55	-0.54
C211	-0.196	-0.002	-0.924	-0.921
C216	-0.195	-0.016	-0.902	-0.9
C225	0.954	0.703	0.2	0.194
C226	0.858	0.85	-0.079	-0.09
C227	0.182	-0.028	0.963	0.954
C232	0.137	-0.075	0.961	0.953
C249	0.967	0.724	0.201	0.193
C250	0.862	0.885	-0.095	-0.108
C251	0.117	-0.087	0.955	0.949
C256	0.086	-0.122	0.955	0.95
C66	0.825			
C67	0.218	-0.048		
C72	0.204	-0.082	0.998	
C81	0.464	0.618	-0.753	-0.763
C82	0.394	0.613	-0.683	-0.693
C83	-0.414	-0.124	-0.971	-0.967
C88	-0.456	-0.167	-0.96	-0.956
C89	0.344	0.505	-0.816	-0.822
C90	0.298	0.52	-0.757	-0.763
C91	-0.441	-0.202	-0.95	-0.941
C96	-0.493	-0.246	-0.932	-0.923

	C81
C121	0.397
C122	0.631
C123	-0.822
C128	-0.84
C137	0.909
C138	0.871
C139	0.559
C144	0.523
C145	0.929
C146	0.779
C147	0.556
C152	0.445
C177	0.4
C178	0.578
C179	-0.708
C184	-0.741
C185	0.466
C186	0.547
C187	-0.702
C192	-0.665
C201	0.898
C202	0.88
C203	0.692
C208	0.665
C209	0.575
C210	0.651
C211	0.696
C216	0.673
C225	0.43
C226	0.65
C227	-0.745
C232	-0.775
C249	0.437
C250	0.667
C251	-0.785
C256	-0.806
C82	0.894
C83	0.593
C88	0.557
C89	0.968
C90	0.888
C91	0.554
C96	0.511

	C82	C83	C88	C89	C90
C121	0.332	-0.471	-0.515	0.288	0.24
C122	0.602	-0.113	-0.161	0.513	0.513
C123	-0.796	-0.901	-0.881	-0.88	-0.858
C128	-0.812	-0.894	-0.874	-0.891	-0.869
C137	0.745	0.461	0.431	0.905	0.746
C138	0.898	0.375	0.326	0.832	0.875
C139	0.478	0.953	0.956	0.644	0.566
C144	0.461	0.945	0.946	0.611	0.552
C145	0.859	0.426	0.38	0.89	0.841
C146	0.916	0.337	0.273	0.728	0.89
C147	0.483	0.934	0.936	0.651	0.574
C152	0.269	0.815	0.822	0.54	0.364
C177	0.322	-0.438	-0.48	0.297	0.237
C178	0.587	-0.192	-0.242	0.472	0.501
C179	-0.599	-0.923	-0.918	-0.784	-0.677
C184	-0.682	-0.925	-0.914	-0.794	-0.747
C185	0.462	-0.363	-0.415	0.331	0.365
C186	0.603	-0.086	-0.136	0.414	0.517
C187	-0.636	-0.923	-0.914	-0.782	-0.713
C192	-0.762	-0.803	-0.779	-0.681	-0.804
C201	0.765	0.548	0.522	0.897	0.777
C202	0.886	0.626	0.587	0.851	0.894
C203	0.653	0.938	0.93	0.744	0.719
C208	0.627	0.929	0.926	0.725	0.695
C209	0.349	0.324	0.306	0.628	0.37
C210	0.613	0.48	0.443	0.661	0.625
C211	0.651	0.908	0.899	0.735	0.715
C216	0.629	0.885	0.879	0.713	0.692
C225	0.318	-0.401	-0.434	0.345	0.243
C226	0.618	-0.123	-0.163	0.554	0.546
C227	-0.713	-0.934	-0.918	-0.796	-0.781
C232	-0.746	-0.923	-0.905	-0.819	-0.809
C249	0.337	-0.393	-0.43	0.325	0.253
C250	0.654	-0.099	-0.145	0.561	0.579
C251	-0.708	-0.9	-0.888	-0.833	-0.769
C256	-0.732	-0.893	-0.878	-0.849	-0.789
C83	0.569				
C88	0.517	0.997			
C89	0.849	0.665	0.634		
C90	0.988	0.654	0.604	0.874	
C91	0.537	0.981	0.979	0.635	0.631
C96	0.468	0.97	0.978	0.592	0.559

Pearson Correlation Coefficient

	C91	C96
C121	-0.502	-0.559
C122	-0.211	-0.268
C123	-0.892	-0.863
C128	-0.879	-0.849
C137	0.457	0.413
C138	0.366	0.295
C139	0.962	0.959
C144	0.951	0.943
C145	0.414	0.352
C146	0.33	0.242
C147	0.957	0.95
C152	0.845	0.851
C177	-0.457	-0.514
C178	-0.253	-0.314
C179	-0.935	-0.922
C184	-0.93	-0.912
C185	-0.406	-0.471
C186	-0.18	-0.238
C187	-0.942	-0.925
C192	-0.799	-0.757
C200	0.837	0.807
C201	0.56	0.53
C202	0.634	0.589
C203	0.94	0.924
C208	0.937	0.925
C209	0.258	0.227
C210	0.415	0.365
C211	0.934	0.919
C216	0.919	0.907
C225	-0.403	-0.45
C226	-0.141	-0.186
C227	-0.944	-0.92
C232	-0.929	-0.902
C249	-0.4	-0.448
C250	-0.125	-0.176
C251	-0.902	-0.885
C256	-0.892	-0.872
C96	0.992	

	C121	C122	C123
C122	0.827		
C123	0.148	-0.168	
C128	0.121	-0.213	0.998
C137	0.491	0.552	-0.693
C138	0.552	0.656	-0.652
C139	-0.449	-0.164	-0.852
C144	-0.474	-0.184	-0.828
C145	0.556	0.671	-0.683
C146	0.51	0.659	-0.592
C147	-0.429	-0.187	-0.849
C152	-0.4	-0.235	-0.685
C177	0.983	0.774	0.126
C178	0.873	0.935	-0.132
C179	0.262	0.006	0.914
C184	0.233	-0.071	-0.936
C185	0.957	0.901	0.051
C186	0.718	0.938	-0.161
C187	0.275	0.009	0.932
C192	0.143	-0.152	0.835
C201	0.31	0.409	-0.774
C202	0.215	0.383	-0.82
C203	-0.309	-0.038	-0.92
C208	-0.328	-0.069	-0.91
C209	0.304	0.347	-0.422
C210	0.213	0.374	-0.568
C211	-0.274	-0.031	-0.892
C217	-0.121	0.093	-0.891
C226	0.819	0.802	-0.246
C227	0.25	-0.016	0.958
C232	0.206	-0.063	0.965
C249	0.958	0.757	0.083
C250	0.826	0.847	-0.249
C251	0.187	-0.072	0.948
C256	0.155	-0.108	0.952

	C128
C137	-0.708
C138	-0.669
C139	-0.846
C144	-0.821
C145	-0.703
C146	-0.61
C147	-0.841
C152	-0.676
C177	0.102
C178	-0.167
C179	0.911
C184	0.937
C185	0.019
C186	-0.202
C187	0.928
C192	0.841
C201	-0.781
C202	-0.825
C203	-0.914
C208	-0.905
C209	-0.427
C210	-0.572
C211	-0.888
C216	-0.871
C225	0.048
C226	-0.269
C227	0.95
C232	0.958
C249	0.058
C250	-0.276
C251	0.944
C256	0.95

Pearson Correlation Coefficient

	C137	C138	C139	C144
C138	0.892			
C139	0.537	0.396		
C144	0.491	0.374	0.988	
C145	0.961	0.959	0.465	0.434
C146	0.743	0.947	0.318	0.313
C147	0.562	0.422	0.993	0.979
C152	0.494	0.25	0.916	0.887
C177	0.537	0.561	-0.398	-0.438
C178	0.575	0.707	-0.214	-0.257
C179	-0.701	-0.554	-0.974	-0.948
C184	-0.688	-0.602	-0.947	-0.916
C185	0.5	0.618	-0.371	-0.399
C186	0.439	0.621	-0.155	-0.174
C187	-0.68	-0.572	-0.965	-0.939
C192	-0.569	-0.624	-0.76	-0.741
C201	0.884	0.793	0.566	0.536
C202	0.786	0.819	0.582	0.557
C203	0.617	0.53	0.927	0.908
C208	0.608	0.513	0.927	0.905
C209	0.665	0.502	0.382	0.377
C210	0.628	0.647	0.463	0.465
C211	0.632	0.546	0.923	0.91
C216	0.622	0.527	0.905	0.888
C225	0.576	0.55	-0.353	-0.404
C226	0.638	0.703	-0.159	-0.228
C227	-0.658	-0.587	-0.915	-0.894
C232	-0.682	-0.619	-0.897	-0.871
C249	0.563	0.551	-0.352	-0.4
C250	0.636	0.726	-0.141	-0.207
C251	-0.703	-0.596	-0.884	-0.856
C256	-0.718	-0.618	-0.87	-0.838

	C145	C146	C147	C152
C146	0.884			
C147	0.484	0.34		
C152	0.376	0.152	0.915	
C177	0.58	0.504	-0.371	-0.338
C178	0.685	0.708	-0.216	-0.269
C179	-0.628	-0.451	-0.976	-0.89
C184	-0.657	-0.535	-0.939	-0.816
C185	0.614	0.629	-0.367	-0.382
C186	0.598	0.677	-0.194	-0.273
C187	-0.63	-0.497	-0.97	-0.863
C192	-0.617	-0.631	-0.757	-0.546
C201	0.848	0.656	0.594	0.504
C202	0.822	0.765	0.606	0.454
C203	0.568	0.443	0.934	0.811
C208	0.547	0.413	0.937	0.801
C209	0.578	0.347	0.391	0.434
C210	0.645	0.596	0.468	0.41
C211	0.597	0.472	0.93	0.833
C216	0.579	0.446	0.917	0.806
C225	0.587	0.457	-0.318	-0.281
C226	0.693	0.645	-0.14	-0.19
C227	-0.63	-0.523	-0.92	-0.795
C232	-0.659	-0.556	-0.902	-0.771
C249	0.598	0.482	-0.326	-0.282
C250	0.716	0.697	-0.129	-0.168
C251	-0.652	-0.498	-0.892	-0.774
C256	-0.671	-0.522	-0.878	-0.757

Pearson Correlation Coefficient

	C177	C178	C179
C178	0.859		
C179	0.208	0.036	
C184	0.187	-0.046	0.977
C185	0.945	0.951	0.194
C186	0.678	0.931	0.028
C187	0.223	0.022	0.992
C192	0.115	-0.112	0.787
C201	0.331	0.397	-0.711
C202	0.237	0.388	-0.699
C203	-0.265	-0.079	-0.941
C208	-0.28	-0.104	-0.94
C209	0.335	0.332	-0.474
C210	0.238	0.395	-0.532
C211	-0.244	-0.085	-0.937
C216	-0.244	-0.106	-0.922
C225	0.973	0.809	0.153
C226	0.824	0.894	-0.047
C227	0.203	0.001	0.948
C232	0.159	-0.049	0.94
C249	0.979	0.827	0.16
C250	0.824	0.927	-0.056
C251	0.144	-0.034	0.935
C256	0.113	-0.071	0.928

	C184	C185	C186	C187
C185	0.122			
C186	-0.091	0.865		
C187	0.983	0.19	0.01	
C192	0.876	0.01	-0.193	0.814
C201	-0.67	0.311	0.279	-0.699
C202	-0.705	0.287	0.334	-0.717
C203	-0.913	-0.237	-0.062	-0.938
C208	-0.913	-0.263	-0.098	-0.935
C209	-0.378	0.249	0.227	-0.437
C210	-0.502	0.266	0.37	-0.538
C211	-0.915	-0.211	-0.064	-0.932
C216	-0.901	-0.222	-0.098	-0.915
C225	0.152	0.886	0.594	0.174
C226	-0.097	0.852	0.743	-0.056
C227	0.94	0.159	-0.017	0.958
C232	0.937	0.111	-0.061	0.951
C249	0.139	0.92	0.647	0.182
C250	-0.121	0.883	0.807	-0.068
C251	0.908	0.121	-0.023	0.932
C256	0.908	0.084	-0.058	0.927

	C192
C201	-0.498
C202	-0.626
C203	-0.761
C208	-0.775
C209	-0.107
C210	-0.327
C211	-0.782
C216	-0.789
C225	0.107
C226	-0.119
C227	0.787
C232	0.794
C249	0.067
C250	-0.157
C251	0.73
C256	0.739

Pearson Correlation Coefficient

	C201	C202	C203
C202	0.929		
C203	0.742	0.787	
C208	0.723	0.754	0.994
C209	0.59	0.483	0.422
C210	0.614	0.686	0.563
C211	0.751	0.78	0.969
C216	0.741	0.756	0.954
C225	0.421	0.282	-0.199
C226	0.602	0.559	0.07
C227	-0.766	-0.828	-0.98
C232	-0.786	-0.851	-0.975
C249	0.385	0.272	-0.211
C250	0.576	0.56	0.069
C251	-0.837	-0.857	-0.979
C256	-0.846	-0.869	-0.971

	C208	C209	C210	C211	C216
C209	0.372				
C210	0.493	0.867			
C211	0.964	0.323	0.461		
C216	0.962	0.25	0.369	0.992	
C225	-0.2	0.305	0.167	-0.166	-0.15
C226	0.058	0.275	0.314	0.069	0.074
C227	-0.964	-0.446	-0.61	-0.953	-0.929
C232	-0.958	-0.454	-0.622	-0.945	-0.921
C249	-0.217	0.256	0.149	-0.162	-0.148
C250	0.048	0.269	0.346	0.081	0.076
C251	-0.967	-0.491	-0.602	-0.946	-0.93
C256	-0.958	-0.493	-0.609	-0.936	-0.92

	C225
C226	0.86
C227	0.152
C232	0.109
C249	0.987
C250	0.837
C251	0.07
C256	0.042

Pearson Correlation Coefficient

	C226	C227	C232
C227	-0.134		
C232	-0.187	0.998	
C249	0.847	0.158	0.116
C250	0.988	-0.139	-0.193
C251	-0.209	0.982	0.985
C256	-0.245	0.979	0.984

	C249	C250	C251
C250	0.841		
C251	0.09	-0.2	
C256	0.062	-0.237	0.998

Designation	Parametric map	Summary measure
C 1	cdifnsa	avg NSA
C 2	cdifnsa	avgstdev
C 3	cdifnsa	avgmean
C 4	cdifnsa	avgmax
C 5	cdifnsa	avgmin
C 6	cdifnsa	avgsk
C 7	cdifnsa	avgkurt
C 8	cdifnsa	avgmed
C 9	cratnsa	avg NSA
C 10	cratnsa	avgstdev
C 11	cratnsa	avgmean
C 12	cratnsa	avgmax
C 13	cratnsa	avgmin
C 14	cratnsa	avgsk
C 15	cratnsa	avgkurt
C 16	cratnsa	avgmed
C 17	cdifsd	avg NSA
C 18	cdifsd	avgstdev
C 19	cdifsd	avgmean
C 20	cdifsd	avgmax
C 21	cdifsd	avgmin
C 22	cdifsd	avgsk
C 23	cdifsd	avgkurt
C 24	cdifsd	avgmed
C 25	cratsd	avg NSA
C 26	cratsd	avgstdev
C 27	cratsd	avgmean
C 28	cratsd	avgmax
C 29	cratsd	avgmin
C 30	cratsd	avgsk
C 31	cratsd	avgkurt
C 32	cratsd	avgmed
C 33	ratsd	avg NSA
C 34	ratsd	avgstdev
C 35	ratsd	avgmean
C 36	ratsd	avgmax
C 37	ratsd	avgmin
C 38	ratsd	avgsk
C 39	ratsd	avgkurt
C 40	ratsd	avgmed
C 41	difsd	avg NSA
C 42	difsd	avgstdev
C 43	difsd	avgmean
C 44	difsd	avgmax
C 45	difsd	avgmin
C 46	difsd	avgsk
C 47	difsd	avgkurt
C 48	difsd	avgmed
C 49	difnsa	avg NSA
C 50	difnsa	avgstdev
C 51	difnsa	avgmean
C 52	difnsa	avgmax
C 53	difnsa	avgmin
C 54	difnsa	avgsk
C 55	difnsa	avgkurt
C 56	difnsa	avgmed
C 57	ratnsa	avg NSA
C 58	ratnsa	avgstdev
C 59	ratnsa	avgmean
C 60	ratnsa	avgmax
C 61	ratnsa	avgmin
C 62	ratnsa	avgsk
C 63	ratnsa	avgkurt
C 64	ratnsa	avgmed

C 66	idd30difnsa	avgstdev
C 67	idd30difnsa	avgmean
C 68	idd30difnsa	avgmax
C 69	idd30difnsa	avgmin
C 70	idd30difnsa	avgsk
C 71	idd30difnsa	avgkurt
C 72	idd30difnsa	avgmed
C 73	idd30cratsd	avg NSA
C 74	idd30cratsd	avgstdev
C 75	idd30cratsd	avgmean
C 76	idd30cratsd	avgmax
C 77	idd30cratsd	avgmin
C 78	idd30cratsd	avgsk
C 79	idd30cratsd	avgkurt
C 80	idd30cratsd	avgmed
C 81	idd30cdifnsa	avg NSA
C 82	idd30cdifnsa	avgstdev
C 83	idd30cdifnsa	avgmean
C 84	idd30cdifnsa	avgmax
C 85	idd30cdifnsa	avgmin
C 86	idd30cdifnsa	avgsk
C 87	idd30cdifnsa	avgkurt
C 88	idd30cdifnsa	avgmed
C 89	idd30cdifsd	avg NSA
C 90	idd30cdifsd	avgstdev
C 91	idd30cdifsd	avgmean
C 92	idd30cdifsd	avgmax
C 93	idd30cdifsd	avgmin
C 94	idd30cdifsd	avgsk
C 95	idd30cdifsd	avgkurt
C 96	idd30cdifsd	avgmed
C 97	idd30cratnsa	avg NSA
C 98	idd30cratnsa	avgstdev
C 99	idd30cratnsa	avgmean
C 100	idd30cratnsa	avgmax
C 101	idd30cratnsa	avgmin
C 102	idd30cratnsa	avgsk
C 103	idd30cratnsa	avgkurt
C 104	idd30cratnsa	avgmed
C 105	idd30ratsd	avg NSA
C 106	idd30ratsd	avgstdev
C 107	idd30ratsd	avgmean
C 108	idd30ratsd	avgmax
C 109	idd30ratsd	avgmin
C 110	idd30ratsd	avgsk
C 111	idd30ratsd	avgkurt
C 112	idd30ratsd	avgmed
C 113	idd30ratnsa	avg NSA
C 114	idd30ratnsa	avgstdev
C 115	idd30ratnsa	avgmean
C 116	idd30ratnsa	avgmax
C 117	idd30ratnsa	avgmin
C 118	idd30ratnsa	avgsk
C 119	idd30ratnsa	avgkurt
C 120	idd30ratnsa	avgmed
C 121	idd30difsd	avg NSA
C 122	idd30difsd	avgstdev
C 123	idd30difsd	avgmean
C 124	idd30difsd	avgmax
C 125	idd30difsd	avgmin
C 126	idd30difsd	avgsk
C 127	idd30difsd	avgkurt
C 128	idd30difsd	avgmed
C 129	dm9ratsd	avg NSA
C 130	dm9ratsd	avgstdev

C 132	dm9ratsd	avgmax
C 133	dm9ratsd	avgmin
C 134	dm9ratsd	avgsk
C 135	dm9ratsd	avgkurt
C 136	dm9ratsd	avgmed
C 137	dm9cdifnsa	avg NSA
C 138	dm9cdifnsa	avgstdev
C 139	dm9cdifnsa	avgmean
C 140	dm9cdifnsa	avgmax
C 141	dm9cdifnsa	avgmin
C 142	dm9cdifnsa	avgsk
C 143	dm9cdifnsa	avgkurt
C 144	dm9cdifnsa	avgmed
C 145	dm9cdifsd	avg NSA
C 146	dm9cdifsd	avgstdev
C 147	dm9cdifsd	avgmean
C 148	dm9cdifsd	avgmax
C 149	dm9cdifsd	avgmin
C 150	dm9cdifsd	avgsk
C 151	dm9cdifsd	avgkurt
C 152	dm9cdifsd	avgmed
C 153	dm9cratnsa	avg NSA
C 154	dm9cratnsa	avgstdev
C 155	dm9cratnsa	avgmean
C 156	dm9cratnsa	avgmax
C 157	dm9cratnsa	avgmin
C 158	dm9cratnsa	avgsk
C 159	dm9cratnsa	avgkurt
C 160	dm9cratnsa	avgmed
C 161	dm9cratsd	avg NSA
C 162	dm9cratsd	avgstdev
C 163	dm9cratsd	avgmean
C 164	dm9cratsd	avgmax
C 165	dm9cratsd	avgmin
C 166	dm9cratsd	avgsk
C 167	dm9cratsd	avgkurt
C 168	dm9cratsd	avgmed
C 169	dm9ratnsa	avg NSA
C 170	dm9ratnsa	avgstdev
C 171	dm9ratnsa	avgmean
C 172	dm9ratnsa	avgmax
C 173	dm9ratnsa	avgmin
C 174	dm9ratnsa	avgsk
C 175	dm9ratnsa	avgkurt
C 176	dm9ratnsa	avgmed
C 177	dm9difnsa	avg NSA
C 178	dm9difnsa	avgstdev
C 179	dm9difnsa	avgmean
C 180	dm9difnsa	avgmax
C 181	dm9difnsa	avgmin
C 182	dm9difnsa	avgsk
C 183	dm9difnsa	avgkurt
C 184	dm9difnsa	avgmed
C 185	dm9difsd	avg NSA
C 186	dm9difsd	avgstdev
C 187	dm9difsd	avgmean
C 188	dm9difsd	avgmax
C 189	dm9difsd	avgmin
C 190	dm9difsd	avgsk
C 191	dm9difsd	avgkurt
C 192	dm9difsd	avgmed
C 193	g19cratsd	avg NSA
C 194	g19cratsd	avgstdev
C 195	g19cratsd	avgmean
C 196	g19cratsd	avgmax

C 198	g19cratsd	avgsk
C 199	g19cratsd	avgkurt
C 200	g19cratsd	avgmed
C 201	g19cdifnsa	avg NSA
C 202	g19cdifnsa	avgstdev
C 203	g19cdifnsa	avgmean
C 204	g19cdifnsa	avgmax
C 205	g19cdifnsa	avgmin
C 206	g19cdifnsa	avgsk
C 207	g19cdifnsa	avgkurt
C 208	g19cdifnsa	avgmed
C 209	g19cdifsd	avg NSA
C 210	g19cdifsd	avgstdev
C 211	g19cdifsd	avgmean
C 212	g19cdifsd	avgmax
C 213	g19cdifsd	avgmin
C 214	g19cdifsd	avgsk
C 215	g19cdifsd	avgkurt
C 216	g19cdifsd	avgmed
C 217	g19cratnsa	avg NSA
C 218	g19cratnsa	avgstdev
C 219	g19cratnsa	avgmean
C 220	g19cratnsa	avgmax
C 221	g19cratnsa	avgmin
C 222	g19cratnsa	avgsk
C 223	g19cratnsa	avgkurt
C 224	g19cratnsa	avgmed
C 225	g19difsd	avg NSA
C 226	g19difsd	avgstdev
C 227	g19difsd	avgmean
C 228	g19difsd	avgmax
C 229	g19difsd	avgmin
C 230	g19difsd	avgsk
C 231	g19difsd	avgkurt
C 232	g19difsd	avgmed
C 233	g19ratsd	avg NSA
C 234	g19ratsd	avgstdev
C 235	g19ratsd	avgmean
C 236	g19ratsd	avgmax
C 237	g19ratsd	avgmin
C 238	g19ratsd	avgsk
C 239	g19ratsd	avgkurt
C 240	g19ratsd	avgmed
C 241	g19ratnsa	avg NSA
C 242	g19ratnsa	avgstdev
C 243	g19ratnsa	avgmean
C 244	g19ratnsa	avgmax
C 245	g19ratnsa	avgmin
C 246	g19ratnsa	avgsk
C 247	g19ratnsa	avgkurt
C 248	g19ratnsa	avgmed
C 249	g19difnsa	avg NSA
C 250	g19difnsa	avgstdev
C 251	g19difnsa	avgmean
C 252	g19difnsa	avgmax
C 253	g19difnsa	avgmin
C 254	g19difnsa	avgsk
C 255	g19difnsa	avgkurt
C 256	g19difnsa	avgmed

CP	cdifnsa.avgNSA	cdifnsa.avgstdev	cdifnsa.avgmean	cdifnsa.avgmed	cdifsd.avgNSA	cdifsd.avgstdev	cdifsd.avgmean	cdifsd.avgmed
no	71.337837	153.089752	-352.003967	-378	144.64389	353.995514	-2054.727539	-2130
no	81.856071	194.146729	-425.24823	-456	163.506668	412.013062	-2285.181641	-2399
no	72.682129	160.105988	-202.775604	-227	149.393463	339.30304	-1124.748291	-1189
no	143.355972	298.960876	40.166729	13	314.207672	719.262085	-1006.311279	-1139
no	83.379234	175.829819	-412.151917	-426	162.782715	365.264435	-2247.02124	-2288
no	111.481491	227.964569	-150.296844	-179	229.417877	494.884888	-1972.106567	-2041
no	73.618774	127.678375	-319.580872	-337	151.947815	266.310425	-1499.166626	-1521
no	94.08046	205.370743	-243.146042	-272	207.665527	454.891327	-2252.210449	-2340
no	74.839417	149.986267	-424.609894	-448	140.82901	311.221436	-2362.855469	-2425
no	107.653305	213.259659	-304.740967	-325	207.14624	466.619904	-1687.382812	-1767
no	117.447891	186.025055	-99.268875	-114	257.853699	424.802856	-1524.858154	0
no	103.807388	208.049622	-615.01355	-651	173.77771	331.573792	-2932.647705	-2983
yes	138.589493	293.661041	-102.412254	-145	286.15155	676.040955	-1957.625244	-2097
yes	116.13546	267.183075	-322.311005	-358	231.846848	558.382263	-2246.293945	-2359
yes	105.065201	193.791138	-155.190201	-176	219.091919	460.419769	-1305.223877	0
yes	147.206421	231.990738	-185.478882	-206	259.309753	487.943298	-1807.480713	-1918
	111.623764	211.211761	-103.318443	-118	220.322876	461.175537	-2684.515137	-2760

difsd.avgNSA	difsd.avgstdev	difsd.avgmean	difsd.avgmed	difnsa.avgNSA	difnsa.avgstdev	difnsa.avgmean	difnsa.avgmed	idd30difnsa.avgNSA
240.39325	519.183838	-1220.863892	-1307	108.040352	224.440063	132.420425	106	110.168953
276.457336	598.743408	-387.947052	-486	121.667435	254.690033	180.532486	153	123.795746
185.979599	379.86084	-1055.422852	-1121	84.248177	165.384491	20.131252	3	85.768768
239.892319	573.661133	-1496.182861	-1614	113.306503	266.079346	-340.499542	-379	119.802521
264.54071	598.755066	-339.596802	-437	118.425278	250.966415	180.892548	152	119.352661
250.118286	650.969238	-821.511475	-960	112.218918	265.85498	-91.109993	-132	123.760803
229.171387	463.280762	-226.300293	-305	102.860886	197.051239	114.137344	93	103.670807
266.394104	646.784424	-1000.309753	-1132	103.339417	208.943298	-2.945379	-23	102.079285
254.06192	558.465088	-344.160706	-427	118.985237	251.158463	197.371063	175	119.577454
272.475037	585.279175	-1139.929077	-1251	123.622429	250.524323	35.741791	-1	134.607117
220.236755	529.406982	-783.013062	-887	93.986763	221.641525	-175.954132	-207	98.666275
341.457153	726.040649	-198.635284	-338	157.676758	298.792114	299.803528	267	161.483948
310.738068	699.82373	-1208.066406	-1316	140.445328	311.343079	-189.290955	-220	143.587296
305.129211	713.728516	-1005.204529	-1149	131.419678	297.84314	33.300182	-4	136.371796
229.98349	574.874939	-617.171265	-740	104.88559	237.313812	-93.272285	-137	109.310455
289.067993	616.867737	-1388.329346	-1506	149.37384	263.186951	-130.51622	-158	144.17749
227.68576	705.257629	-1842.834839	-1994	110.790375	269.87085	-146.811401	-182	118.848839

idd30difnsa.avgstdev	idd30difnsa.avgmean	idd30difnsa.avgmed	idd30cdifnsa.avgNSA	idd30cdifnsa.avgstdev	idd30cdifnsa.avgmean
243.504517	151.982086	115	70.569763	172.82814	-351.224915
273.130157	194.725098	156	83.411545	197.966187	-419.75177
180.227448	51.673225	29	69.710129	151.620605	-227.787354
280.728516	-282.369965	-329	145.038834	346.049438	-0.362511
271.612427	183.628845	147	85.383652	192.991302	-399.35321
303.731079	-67.573799	-130	119.251465	239.597656	-169.700073
205.698059	153.250458	123	68.907883	116.452026	-340.535095
216.089859	15.493185	-13	88.6427	194.817871	-241.65152
268.355591	216.257202	189	73.065567	146.379944	-427.860901
297.75061	74.878029	24	112.043175	221.584106	-352.090973
241.318726	-128.209259	-168	109.415176	181.33606	-148.813248
310.31543	396.633362	373	87.583145	160.54567	-683.800293
314.716644	-128.551865	-167	143.593597	328.20105	-151.467361
322.832336	59.665894	12	118.595177	283.722107	-339.11853
272.89502	-5.971813	-61	95.197433	176.566864	-232.784821
268.526001	-49.186543	-83	131.179428	219.49707	-257.910461
291.898712	-58.707825	-102	119.768066	244.685272	-183.22905

idd30cdifnsa.avgmed	idd30cdifsd.avgNSA	idd30cdifsd.avgstdev	idd30cdifsd.avgmean	idd30cdifsd.avgmed	idd30difsd.avgNSA
-389	107.294914	281.471313	-530.38562	-598	167.97049
-461	118.313141	296.778564	-658.71698	-722	193.175705
-253	108.498512	246.514404	-364.279724	-409	132.577698
-40	221.118179	542.329346	-21.403872	-101	174.080902
-422	119.833755	272.546143	-629.535156	-640	182.011505
-193	171.260239	364.861053	-403.262115	-448	184.538925
-355	108.865601	193.369049	-457.735352	-482	160.249298
-263	150.98262	328.075287	-393.093353	-448	162.728714
-454	103.731163	223.026428	-674.713623	-715	184.54361
-381	156.943542	332.746826	-504.828064	-535	197.422852
-172	178.09407	304.355896	-270.155457	-289	151.388123
-709	132.725952	234.05304	-975.127625	-1007	243.654236
-219	215.942902	511.038391	-283.784912	-407	218.989258
-386	173.2724	420.946228	-564.755249	-634	206.622833
-257	163.863739	286.372559	-421.069702	-468	168.47641
-289	190.499329	335.335541	-441.695068	-496	218.561798
-209	171.146439	369.423035	-331.985229	-356	175.203186

idd30difsd.avgstdev	idd30difsd.avgmean	idd30difsd.avgmed	dm9cdifnsa.avgNSA	dm9cdifnsa.avgstdev	dm9cdifnsa.avgmean
367.894073	43.026783	-18	1.231266	2.290749	-4.610636
425.962219	114.372894	51	1.417627	2.946607	-5.676377
283.896973	-59.350304	-93	1.18473	2.224452	-2.704933
404.560242	-646.162842	-726	2.502903	4.357616	0.584124
404.927368	118.571823	58	1.46565	2.782225	-5.365305
480.674286	-215.52092	-315	1.929021	3.272656	-2.028162
316.773865	100.786903	42	1.778767	2.796976	-1.820099
333.678925	-108.478325	-154	1.649491	2.995337	-3.091149
410.049347	189.609131	135	1.344453	2.356297	-5.744819
434.208191	-48.503376	-126	1.845172	3.166828	-4.002062
370.016541	-296.134369	-361	1.752025	2.458118	-1.589924
459.523621	223.83519	177	1.824151	3.202287	-8.464417
496.850677	-356.989716	-433	2.53429	4.561933	-1.224469
491.666565	-159.083603	-254	2.009354	3.88209	-4.183762
400.396423	-134.945526	-212	1.832937	2.894241	-1.867702
438.73526	-147.290344	-218	2.523948	3.537663	-2.494933
415.35675	-276.933807	-348	1.995495	3.529939	-1.535973

dm9cdifnsa.avgmed	dm9cdifsd.avgNSA	dm9cdifsd.avgstdev	dm9cdifsd.avgmean	dm9cdifsd.avgmed	dm9difnsa.avgNSA	dm9difnsa.avgstdev
-5	0.584722	1.214912	-1.692079	-1	1.900654	3.323183
-6	0.662342	1.513733	-2.181836	-2	2.142128	3.822843
-2	0.539811	1.067352	-0.943955	-1	1.386114	2.326902
0	1.070157	2.032747	0.604591	0	2.041501	3.902382
-6	0.687836	1.365644	-1.992142	-2	2.048029	3.701719
-2	0.842158	1.46185	-0.910394	-1	2.02969	3.97187
-2	0.715247	1.242527	-0.438973	0	1.888727	3.035513
-3	0.660398	1.321994	-0.944786	-1	1.798959	3.080779
-6	0.596141	1.14967	-2.276434	-2	2.125911	3.838679
-4	0.823922	1.49291	-1.476859	-1	2.111975	3.598088
-2	0.709101	1.130769	-0.539827	0	1.682173	3.197364
-9	0.778894	1.398043	-3.242606	-3	2.799597	4.492585
-1	1.168295	2.336045	-0.294666	0	2.501615	4.612999
-4	0.936458	1.908983	-1.433667	-2	2.307103	4.314308
-2	0.740265	1.273371	-0.499649	0	1.813659	3.426193
-3	0.990007	1.427639	-0.723998	0	2.60909	3.849916
-2	0.851804	1.551166	-0.52729	0	1.944529	3.932964

dm9difnsa.avgmean	dm9difnsa.avgmed	dm9difsd.avgNSA	dm9difsd.avgstdev	dm9difsd.avgmean	dm9difsd.avgmed	g19cdifnsa.avgNSA
1.791655	1	0.766106	1.5491	0.665284	0	72.579895
2.435122	1	0.923996	1.797934	0.985728	0	71.309288
0.374552	0	0.53663	1.089255	0.206663	0	81.829231
-4.584195	-5	0.843352	1.721402	-1.816081	-2	167.945053
2.393781	1	0.850752	1.705731	0.938944	0	75.811264
-1.055155	-2	0.892907	2.119173	-0.165013	-1	96.765778
-0.967326	-1	0.704664	1.294719	-0.244221	0	85.339912
0.035669	0	0.661161	1.298805	0.157554	0	111.282883
2.771108	2	0.899483	1.83383	1.258656	1	83.257248
0.565275	0	0.88734	1.680087	0.387298	0	136.133118
-1.774949	-2	0.677936	1.475336	-0.636271	0	117.76033
4.177056	3	1.102342	1.914565	1.667356	1	94.258972
-2.52498	-3	1.090434	2.212284	-0.978708	-1	144.499176
0.502094	0	0.998276	2.04727	0.153472	0	142.852325
-1.234485	-1	0.707314	1.47065	-0.322016	0	107.032745
-1.499764	-1	0.964505	1.583591	-0.20787	0	160.194916
-1.776626	-2	0.822678	1.853816	-0.615258	0	139.206604

g19cdifnsa.avgstdev	g19cdifnsa.avgmean	g19cdifnsa.avgmed	g19cdifsd.avgNSA	g19cdifsd.avgstdev	g19cdifsd.avgmean	g19cdifsd.avgmed
212.692505	-357.512939	-410	100.537827	287.924744	-448.582703	-516
207.909409	-444.510254	-490	106.139008	298.049561	-593.005249	-650
214.78479	-219.876556	-267	120.988632	310.521118	-277.276672	-336
420.929718	117.457611	104	118.109352	354.612732	187.740784	195
220.560608	-407.598083	-453	106.001236	317.388245	-522.110474	-581
230.635651	-240.829086	-273	153.078583	357.993744	-302.68573	-363
189.156082	-316.044739	-339	134.045166	291.309753	-333.904144	-373
276.946655	-228.703812	-269	164.847839	412.857727	-303.289215	-376
196.44693	-435.311707	-472	108.086594	272.689087	-586.044556	-633
302.103027	-294.65863	-338	86.373566	207.652496	-177.29155	-203
266.313538	-208.11203	-271	172.531464	371.633026	-280.89386	-361
181.450821	-648.681335	-673	145.800842	270.145752	-842.74646	-886
368.876343	-135.320435	-223	213.44017	574.89801	-88.042061	-209
361.034668	-238.054504	-299	188.421661	496.613464	-358.740875	-430
207.881317	-188.355209	-213	172.554749	337.027496	-243.570923	-287
345.843689	-119.798882	-174	236.979401	480.733398	-166.534576	-222
348.847473	-69.871887	-145	193.831482	527.198059	-104.826591	-219

g19difsd.avgNSA	g19difsd.avgstdev	g19difsd.avgmean	g19difsd.avgmed	g19difnsa.avgNSA	g19difnsa.avgstdev	g19difnsa.avgmean
184.551239	462.120361	152.091049	77	126.465012	318.311035	151.840271
189.359833	498.125305	276.242767	169	130.446594	345.385925	215.173492
141.784973	344.397278	-18.724121	-69	96.602753	231.55722	30.302505
205.084991	587.681885	-558.575073	-672	137.532898	389.042053	-420.542908
191.68927	489.429932	260.035767	168	132.787857	338.009766	202.504852
188.575302	481.12738	11.093086	-79	131.678864	332.879639	8.959461
180.453979	400.885803	119.965919	64	122.800369	274.624207	114.481194
178.814072	453.672668	4.604113	-81	115.001007	305.064789	-11.313993
202.571091	523.364746	299.984589	210	137.852448	353.766602	202.765884
209.22934	533.046631	96.631088	10	144.070679	364.434906	23.118992
169.526627	462.795044	-151.106522	-235	112.943237	312.880524	-128.171646
254.728485	598.421997	527.496704	432	169.072601	393.043274	379.519501
219.554932	556.585083	-273.233917	-371	152.283875	393.861816	-142.640656
211.544464	561.82373	-14.33621	-115	140.734024	375.950958	-60.670082
183.323456	459.748932	13.457382	-63	119.687607	311.574402	-31.776398
241.690506	560.834961	-145.588013	-247	163.359344	372.314941	-186.01152
185.534454	543.540771	-256.164612	-350	125.051483	369.92038	-193.652847

g19difnsa.avgmed
97
128
-6
-494
142
-56
71
-69
147
-37
-195
317
-211
-126
-86
-254
-254

Appendix L

Calls to randomForest and the EXP2 Data Set

Id	cdifsd avgmean	difsd avgmed	dm9cdifnsa avgstdev	dm9cdifsd avgstdev	g19cdifsd avg NSA
1	-1506.128662	-1044	2.110604	0.88217	110.869194
2	-1538.783447	-509	1.442073	0.614567	77.849075
3	-1473.279541	-970	2.116361	0.879258	109.097992
4	-1559.692139	-549	1.444225	0.711589	87.305214
5	-1142.889893	-346	1.139563	0.55073	75.02829
6	-1146.958496	-535	1.50332	0.577883	67.582108
7	-1371.343506	-596	1.753635	0.832026	111.226578
8	-1759.997314	-757	2.493272	1.029326	142.936752
9	-982.943848	-537	1.628055	0.636941	87.569855
10	-1835.611328	-614	2.116736	0.993954	122.701736
11	-994.506104	-555	1.34396	0.563583	92.557549
12	-1707.345947	-665	1.646893	0.725142	91.25058
13	-1647.885986	-421	1.679309	0.735915	92.752716
14	-2297.991211	-1098	2.894506	1.217765	126.077507
15	-1222.086914	-763	2.08546	0.942853	98.368423
16	-1227.549316	-751	1.872331	0.745132	88.468658
17	-1261.142334	-549	1.330224	0.60167	76.301331
18	-1001.200378	-863	1.979923	0.828044	99.204124
19	-1533.385498	-574	1.635515	0.70721	90.588783
20	-1047.294922	-554	1.172557	0.496547	69.119011
21	-1742.322021	-1088	2.008103	0.795238	121.346794
22	-1261.724609	-767	1.776805	0.745297	98.78186
23	-1591.834106	-620	1.865009	0.77031	107.562271
24	-1118.041992	-829	2.207051	0.865998	112.801552
25	-1361.311279	-646	2.069677	0.819761	108.77182
26	-1266.759521	-222	1.427723	0.632918	82.623367
27	-1314.871826	-590	1.429831	0.576181	84.091927
28	-1147.156616	-723	1.848984	0.792648	119.885406
29	-1459.019043	-1201	1.623105	0.691892	113.19603
30	-1172.120117	-376	1.336858	0.644733	71.683418
31	-2085.207275	-850	2.093622	0.909746	126.852791
32	-1554.790283	-757	1.942472	0.882474	105.123001
33	-1172.381592	-400	1.402386	0.578857	72.335373
34	-1034.872437	-407	1.50551	0.637341	96.518311
35	-1394.214111	-402	1.751501	0.752228	95.136551
36	-1954.264893	-729	2.210483	0.86981	111.88356
37	-1046.379028	-464	1.237144	0.588007	82.542244
38	-940.495972	-307	1.11331	0.48033	74.56662
39	-1885.580078	-508	2.712895	1.181125	137.493317
40	-1331.295776	-372	1.516364	0.652321	87.759979
41	-1764.869385	-337	1.559643	0.788998	90.457207
42	-1455.008057	-494	1.952307	0.846054	99.875046
43	-991.483887	-312	1.494628	0.648839	77.91304
44	-1536.418335	-737	2.271782	0.9758	119.48053
45	-1173.501831	-713	1.678449	0.743745	87.923126
46	-1674.406738	-775	2.29959	0.940526	122.614487
47	-1206.430176	-462	1.499628	0.690442	98.130951
48	-1495.001465	-677	1.659625	0.756781	88.427261
49	-2659.919922	-1007	1.778932	0.838271	93.632362
50	-1724.177734	-749	1.434435	0.608173	85.689705

51	-1732.682373	-833	1.635921	0.743045	105.542694
52	-1425.107666	-537	2.12202	0.928024	112.843094
53	-1388.341553	-689	1.339033	0.589895	87.631447
54	-1692.88623	-891	2.133099	0.97476	106.895523
55	-1859.151367	-675	2.066749	0.899295	123.11438
56	-1982.948853	-385	1.798364	0.837046	103.254234
57	-1643.14209	-769	2.355124	1.010187	137.532578
58	-1072.79126	-765	1.982152	0.841016	111.588287
59	-1526.696045	-1101	2.443818	1.094525	140.157227
60	-1273.188599	-507	1.407683	0.61628	75.21875
61	-1043.307983	-1006	1.718987	0.687699	103.192245
62	-1322.907471	-402	1.34909	0.612353	60.875015
65	-1098.013306	-506	1.4979	0.602236	93.985481
66	-2055.791748	-397	1.922874	0.822325	97.499161
67	-1337.285278	-372	1.38082	0.601793	80.104309
68	-1336.89502	-1067	2.148232	0.91543	119.104355
69	-1217.283447	-386	1.429737	0.6218	109.811523
70	-2464.615723	-835	2.27918	0.985707	137.77066
71	-1757.619385	-882	2.488728	1.049408	143.581604

ID	Cerebral Palsy
1	Yes
2	Yes
3	No
4	No
5	No
6	Yes
7	No
8	No
9	Yes
10	No
11	No
12	No
13	Yes
14	Yes
15	No
16	No
17	No
18	No
19	No
20	Yes
21	Yes
22	Yes
23	No
24	Yes
25	Yes
26	No
27	No
28	Yes
29	Yes
30	No
31	No
32	No
33	Yes
34	Yes
35	No
36	No
37	No
38	No
39	Yes
40	No
41	Yes
42	Yes
43	Yes
44	No
45	No
46	No
47	Yes
48	No
49	No
50	Yes

51	No
52	Yes
53	No
54	No
55	No
56	Yes
57	No
58	Yes
59	No
60	Yes
61	Yes
62	No
65	Yes
66	Yes
67	Yes
68	Yes
69	Yes
70	Yes
71	No

```
library(randomForest)

# variables selected based on coorelations  See pages 173 and 174 in
book 7 of PhD notes


#Use these variables for experiment 2 data set


basechv4 <-
read.table("/home/abdo/rules/TAHrf/basecv4fin.csv",sep=",",header=TRUE)
base.col1<-
read.table("/home/abdo/rules/TAHrf/Outcomesfin.csv",sep=",",header=TRUE)
basechv4B<-basechv4[,-1]
basechv4 <-cbind(base.col1,basechv4)
basechv4 <- basechv4[,-1]
basechv4 <- basechv4[,-2]

base.col1<-base.col1[,-1]

# use tuneRF to determine the best value of mtry
tuneRF(basechv4B,base.col1,ntreeTry=50000,increase=0)
mtry = 2  OOB error = 28.99%
Searching left ...
mtry = 1        OOB error = 34.78%
-0.2 0.05
Searching right ...
mtry = 4        OOB error = 27.54%
0.05 0.05
mtry = 5        OOB error = 28.99%
-0.05263158 0.05
  mtry  OOBError
1    1 0.3478261
2    2 0.2898551
4    4 0.2753623
5    5 0.2898551


set.seed(156)
rf.basecv4 <- randomForest(Cerebral.Palsy ~ ., data=basechv4, mtry=4,
importance=TRUE, ntree=20000)


> set.seed(156)
> rf.basecv4 <- randomForest(Cerebral.Palsy ~ ., data=basechv4, mtry=4,
importance=TRUE, ntree=20000)
> rf.basecv4
```

```
Call:
 randomForest(formula = Cerebral.Palsy ~ ., data = basechv4, mtry = 4,
importance = TRUE, ntree = 20000)
               Type of random forest: classification
                     Number of trees: 20000
No. of variables tried at each split: 4

        OOB estimate of  error rate: 27.54%
Confusion matrix:
    No Yes class.error
No  26  11   0.2972973
Yes  8  24   0.2500000
```

```
basecv4.pred<-predict(rf.basecv4,predict.all=TRUE)

> basecv4.pred
 [1] No  Yes Yes No  No  Yes Yes No  Yes No  No  No  Yes Yes Yes No  No
Yes No
[20] No  Yes No  No  Yes No  Yes Yes Yes Yes No  No  No  Yes Yes Yes No
No  No
[39] Yes Yes Yes Yes No  No  No  No  Yes No  Yes No  No  Yes No  No  No
Yes No
[58] Yes Yes Yes Yes Yes Yes Yes No  Yes Yes No  No
Levels: No Yes
```

```
sink("sampletreesfin.txt")
getTree(rf.basecv4,1,labelVar=TRUE)
sink()
```

```
   left daughter right daughter           split var  split point status
1              2              3 dm9cdifnsa.avgstdev     1.375738      1
2              4              5   g19cdifsd.avg.NSA    70.401215      1
3              6              7      cdifsd.avgmean -1172.941712      1
4              8              9 dm9cdifnsa.avgstdev     1.260823      1
5              0              0                <NA>     0.000000     -1
6             10             11      cdifsd.avgmean -1762.433350      1
7             12             13        difsd.avgmed  -846.000000      1
8              0              0                <NA>     0.000000     -1
9              0              0                <NA>     0.000000     -1
10            14             15      cdifsd.avgmean -2562.267822      1
11            16             17  dm9cdifsd.avgstdev     0.627359      1
12             0              0                <NA>     0.000000     -1
13             0              0                <NA>     0.000000     -1
14             0              0                <NA>     0.000000     -1
15            18             19        difsd.avgmed  -618.500000      1
16            20             21        difsd.avgmed  -549.500000      1
17            22             23 dm9cdifnsa.avgstdev     1.764153      1
18            24             25 dm9cdifnsa.avgstdev     2.244832      1
19             0              0                <NA>     0.000000     -1
20            26             27   g19cdifsd.avg.NSA    84.890816      1
21             0              0                <NA>     0.000000     -1
22             0              0                <NA>     0.000000     -1
23            28             29      cdifsd.avgmean -1464.143799      1
24             0              0                <NA>     0.000000     -1
25             0              0                <NA>     0.000000     -1
26             0              0                <NA>     0.000000     -1
27             0              0                <NA>     0.000000     -1
28             0              0                <NA>     0.000000     -1
29             0              0                <NA>     0.000000     -1
   prediction
1        <NA>
2        <NA>
3        <NA>
4        <NA>
5          No
6        <NA>
7        <NA>
8         Yes
9          No
10       <NA>
11       <NA>
12         No
13        Yes
14         No
15       <NA>
16       <NA>
17       <NA>
18       <NA>
19        Yes
20       <NA>
21        Yes
22         No
23       <NA>
24         No
25        Yes
26         No
```

27	Yes
28	No
29	Yes

Appendix M

Gain Settings

Patient #	Outcome		Gain (db)	RFC class	mis-classified by RFC	Machine	
1	Yes	1	-17	No	TRUE	Acuson	0
2	Yes	1	-1	Yes	FALSE	Acuson	1
3	No	0	-6	Yes	TRUE	Acuson	
4	No	0	5	No	FALSE	Acuson	
5	No	0	5	No	FALSE	Acuson	1
6	Yes	1	6	Yes	FALSE	Acuson	1
7	No	0		Yes	TRUE		
8	No	0	7	No	FALSE	Acuson	1
9	Yes	1	12	Yes	FALSE	Acuson	1
10	No	0	-15	No	FALSE	Acuson	1
11	No	0	4	No	FALSE	Acuson	1
12	No	0	3	No	FALSE	Acuson	1
13	Yes	1	2	Yes	FALSE	Acuson	1
14	Yes	1	-8	Yes	FALSE	Acuson	1
15	No	0		Yes	TRUE		
16	No	0	4	No	FALSE	Acuson	1
17	No	0	-11	No	FALSE	Acuson	1
18	No	0		Yes	TRUE		
19	No	0		No	FALSE		
20	Yes	1	6	No	TRUE	Acuson	
21	Yes	1	-5	Yes	FALSE	Acuson	1
22	Yes	1		No	TRUE		
23	No	0		No	FALSE		
24	Yes	1		Yes	FALSE		
25	Yes	1		No	TRUE		
26	No	0	0	Yes	TRUE	Acuson	
27	No	0	-5	Yes	TRUE	Acuson	
28	Yes	1		Yes	FALSE		
29	Yes	1		Yes	FALSE		
30	No	0	-6	No	FALSE	Acuson	1
31	No	0		No	FALSE		
32	No	0		No	FALSE		
33	Yes	1	4	Yes	FALSE	Acuson	1
34	Yes	1	-2	Yes	FALSE	Acuson	1
35	No	0		Yes	TRUE		
36	No	0		No	FALSE		
37	No	0	1	No	FALSE	Acuson	1
38	No	0	11	No	FALSE	Acuson	1
39	Yes	1		Yes	FALSE		
40	No	0	-4	Yes	TRUE	Acuson	
41	Yes	1	4	Yes	FALSE	Acuson	1
42	Yes	1		Yes	FALSE		
43	Yes	1		No	TRUE		
44	No	0	7	No	FALSE	Acuson	1
45	No	0		No	FALSE		
46	No	0		No	FALSE		
47	Yes	1	1	Yes	FALSE	Acuson	1
48	No	0	1	No	FALSE	Acuson	1

49	No	0		Yes	TRUE		
50	Yes	1	2	No	TRUE	Acuson	
51	No	0	1	No	FALSE	Acuson	1
52	Yes	1		Yes	FALSE		
53	No	0	-2	No	FALSE	Acuson	1
54	No	0		No	FALSE		
55	No	0		No	FALSE		
56	Yes	1		Yes	FALSE		
57	No	0		No	FALSE		
58	Yes	1		Yes	FALSE		
59	No	0		Yes	TRUE		
60	Yes	1	-4	Yes	FALSE	Acuson	1
61	Yes	1		Yes	FALSE		
62	No	0	9	Yes	TRUE	Acuson	
65	Yes	1		Yes	FALSE		
66	Yes	1	-3	Yes	FALSE	Acuson	1
67	Yes	1	-4	No	TRUE	Acuson	
68	Yes	1		Yes	FALSE		
69	Yes	1	-1	Yes	FALSE	Acuson	1
70	Yes	1	3	No	TRUE	Acuson	
71	No	0	-1	No	FALSE	Acuson	1

Number correctly classified, Acuson	28
Total Acuson	39
	0.71795