



THESIS

2



This is to certify that the

dissertation entitled

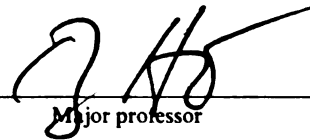
A PARALLEL ALGORITHM FOR
ULTRASONIC MATERIAL CHARACTERIZATION

presented by

CHIPAI CHOU

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Engineering



Major professor

Date May 30, 1995

**LIBRARY
Michigan State
University**

**PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.**

DATE DUE	DATE DUE	DATE DUE
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

**A PARALLEL ALGORITHM FOR ULTRASONIC
MATERIAL CHARACTERIZATION**

**By
Chipai Chou**

A DISSERTATION

**Submitted to
Michigan State University
in partial fulfillment of requirements
for the degree of**

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1995

ABSTRACT

A PARALLEL ALGORITHM FOR ULTRASONIC MATERIAL CHARACTERIZATION

By

Chipai Chou

Attenuation coefficient has been considered to be an important feature in material characterization. However, estimation of an attenuation coefficient in material, especially in biological tissues, is still a difficult task. Unlike conventional approaches, the proposed approach extracts several features from the return echoes, then applies a clustering technique to the extracted features. A modified Hopfield neural network, called the maximum neural network, is adopted to process the extracted data set. The advantages and convergence property of the maximum neural network are discussed in this dissertation. Due to the inherent parallelism in the maximum neural network, material characterization using parallel processing becomes possible. Both synthetic data sets and a data set taken from a human sample are used to demonstrate the capability of the proposed system. Some results were obtained from the proposed scheme which could be achieved by the traditional methods.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Bong Ho, for his guidance and support. Thanks also given to the guidance committee members, Dr. H. Roland Zapp, Dr. James A. Resh, and Dr. Wei-Eihn Kuan, for taking time to serve in the committee. Most importantly, I would like to thank my parents and my wife. Because of their encouragement and support, I was able to complete this dissertation.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
Chapter 1 INTRODUCTION	1
1.1 Ultrasonic imaging	1
1.2 Artificial neural network	4
1.3 Research objective	7
1.4 Thesis organization	8
Chapter 2 BACKGROUND	9
2.1 The acoustic plane wave equations	9
2.2 Transmission and reflection coefficients	12
2.3 The processing element of artificial neural network	16
2.4 The structure of Hopfield neural network	19
Chapter 3 TIME-DOMAIN AND FREQUENCY-DOMAIN ANALYSES	23
3.1 Time-domain technique	23
3.2 Advantages and limitations of time-domain technique	32
3.3 Advantages and limitations of frequency-domain technique	35
3.4 Frequency-domain technique	36
Chapter 4 MATERIAL CHARACTERIZATION USING PARALLEL PROCESSING	43
4.1 Material characterization	43
4.2 Methodology	45
4.3 Feature selection	46
4.4 The structure of maximum neural network	51
4.5 Classification using maximum cut	58
Chapter 5 SIMULATION EXPERIMENTAL RESULTS	64
5.1 Maximum independent set problem	64

5.2	Simulation results	71
5.3	Ultrasonic tissue characterization	80
Chapter 6 CONCLUSIONS		87
6.1	Summary	87
6.2	Directions for future work	88
APPENDIX		
A	Program list for the maximum independent set problem	90
B	Program list for data acquisition and displaying	96
BIBLIOGRAPHY		104

LIST OF TABLES

Table 5.1	Comparison of simulation results for 10-vertex graphs.	74
Table 5.2	Comparison of simulation results for 25-vertex graphs.	75
Table 5.3	Comparison of simulation results for 50-vertex graphs.	76
Table 5.4	Comparison of simulation results for 75-vertex graphs.	77
Table 5.5	Summary of simulation results to demonstrate solution qualities.	78
Table 5.6	Comparison of the computation time for 50%-density graph problems.	79

LIST OF FIGURES

Figure 2.1	Acoustic wave at interface of two different media.	13
Figure 2.2	Schematic diagram of a neuron.	17
Figure 2.3	Input/output functions of neurons.	18
Figure 2.4	Architecture of the Hopfield neural network.	20
Figure 3.1	Bidirectional interrogation for layered model.	24
Figure 3.2	The impulse response pairs of dual interrogation.	27
Figure 3.3	Experimental setup for measuring r_1 .	30
Figure 3.4	Structure of the multi-layered model.	37
Figure 3.5	Reflected signal gated by windows.	38
Figure 4.1	System diagram.	47
Figure 4.2	Stages of data clustering.	48
Figure 4.3	Features extracted from the spectrum of a return echo.	50
Figure 4.4	Architecture of maximum neural network.	53
Figure 5.1	The test result of the proposed algorithm (a) the given graph G (b) the found independent set S of the graph G.	72
Figure 5.2	The test result of the proposed algorithm (a) the given graph G (b) the found independent set S of the graph G.	73
Figure 5.3	Picture of human brain sample with hemorrhaged tumor.	81
Figure 5.4	Schematic diagram of the data acquisition system.	82
Figure 5.5	Projection of five features in a two-dimensional space.	84
Figure 5.6	C-scan image of human brain sample.	85

Figure 5.7 Reconstructed image of human brain sample.

86

CHAPTER 1

INTRODUCTION

1.1 Ultrasonic imaging

Pulsed-echo ultrasound is an important and valuable tool in non-destructive evaluation (NDE) [1], non-invasive clinical diagnostics [2], and many other applications [3]. It employs high frequency acoustic wave to extract information about the internal structure and characteristics of materials. The attractive features of using ultrasound are that it provides safety of operation and acquires low examination cost as compared to other detection techniques. The commonly used x-ray computerized tomography (CT) uses a high energy x-ray beam to obtain images of material structure. Basically, transmission technique based on material absorption characteristics is utilized in x-ray systems and image reconstruction. Quite often, injection of a contrast medium such as iodine, for visualization of non-bony tissue, is necessary and the procedure is invasive and hazardous. The nuclear magnetic resonance (NMR) techniques, which are very popular in recent years, measure the selective resonances of radioactive isotopes in particular organs

and provide a significant medical imaging technique. Since they use extremely strong magnetic fields, several teslas, the possible bioeffects on the human body are not well determined at the present time. The cost of equipment and the expense of examination for both of these techniques are extremely high. Short-term and long-term exposure risks for both operators and patients are also of great concerns. Although ultrasonic imaging systems, which use much lower frequency, can not provide high resolution images as those from CT and MRI, yet the lower cost and safety of operation continue to favor ultrasound for material evaluation.

The research work in this dissertation tries to apply neural network technique to ultrasonic detection in such a way that the imaging resolution can be improved. Non-destructive evaluation of materials by ultrasound has shown rapid growth in recent years, especially in the testing of composite materials which have become a major construction material in aerospace and automotive industries. A great deal of information about the mechanical properties of materials can be retrieved from the acoustic return echoes. A C-scan imaging system has been implemented to display two dimensional images for defects and flaws inside composite materials [1]. However, in order to assure the success and consistency of non-destructive evaluation, acoustic properties such as impedance, attenuation, and reflection coefficient have to be obtained with high accuracy. Several techniques have been proposed to improve the lateral resolution and the range resolution for better image quality. Lateral resolution is the ability to distinguish nearby objects in the transverse direction with respect to the ultrasound beam. Due to the inherent drawbacks of dispersive nature and the spatial profile of acoustic beams, the lateral resolution of ultrasonic imaging systems is poor. Techniques have been proposed to improve lateral resolution. Correlation technique has been used in C-scan imaging [4]. Digital filtering technique has also been implemented [5]. A method of firing a transducer array

repetitively to obtain an adaptive focusing effect for resolution enhancement has also been reported [6]. Range resolution is the ability to distinguish two different acoustic return echoes along the beam direction. Theoretically, range resolution can be improved by using broad-bandwidth and low-Q transducers which can provide narrow transmitting pulses. Due to fabrication difficulties, such transducers are not available at the present time. Beretsky employed frequency deconvolution to improve ultrasonic imaging [7]. Steiner proposed a generalized cross-correlation to improve image quality [8]. Yamada presented an on-line deconvolution for high resolution ultrasonic pulse-echo measurements under constraints of narrow-band transducer [10]. However, range resolution in ultrasonic imaging still has much room for improvement.

Differences in acoustic impedance, velocity, and attenuation of various normal and abnormal tissue were studied under a variety of controlled ultrasonic field conditions. These were found to be quantitatively significant and could be correlated with differences in tissue structure and pathological changes. In the past two decades, there were many techniques proposed for estimating these quantities. Jones utilized the return echoes deconvolved with the transmitted wave to produce the impulse response which yields impedance profiles under the assumption that the wave propagates through non-attenuating media [11]. Cobo-Parra determined the impedance profile of a multi-layered ocean floor. However, their treatment of attenuation is overly simplified by the hypothesis of linear frequency dependency and the same acoustic thickness for each layer [12]. The propagation velocity in tissue is an important property of material. Kossoff [13] and Lin [14] demonstrated that this parameter correlates with pathological characteristics of tissue. Greenleaf used transmission technique [15] and time-of-flight tomography [16] to produce two dimensional sound speed images of female breast tissue. Propagation velocity can also be measured by transmission methods [10, 17, 18] and pulse echo methods

[19,20,21]. However, the propagation velocity has not been extensively used in medical diagnosis because the pathology related changes in propagation velocity are too small to be accurately measured.

Attenuation has been considered an important characteristic capable of forming the basis for a tissue differentiation scheme [22]. Transmission methods [23, 24, 25] are simple and straightforward. However, when these approaches are limited to in-vivo measurements, very few human organs can be accessed in such a way. Kuc [26, 27, 28] proposed an approach to estimate the attenuation from return echoes on the assumption that attenuation coefficient is a strong function of frequency. Based on this assumption, both spectral-shift and spectral-difference approaches were used to estimate attenuation coefficients. The spectral-shift approach estimates attenuation coefficient from the downshift of the echo spectrum as compared with that of the incident pulse. The spectral-difference approach estimates attenuation coefficient from the deviation of the mean log spectrum of the return echoes. Several researchers have also proposed time-domain methods [29, 30]. Although time-domain methods are straightforward, difficulties such as signal distortion and echo overlapping remain to be solved. Because of the difficulties in estimating acoustic parameters by traditional methods, image processing and pattern recognition techniques have been adopted to enhance the results [31, 32, 33]. Almost all of these techniques are used in the post-processing stage.

1.2 Artificial neural network

The study of artificial neural network systems began in the early 1940s. An artificial neural network system is a parallel distributed information processing system which consists of neurons and synapses. Neurons are the processing elements and synapses link

neurons together. Each neuron receives and transmits signals to a number of neurons via synapses. The function of the neural network system depends on the function of neurons and the structure of how neurons and synapses are connected. McCulloch and Pitts modeled a neuron as a simple threshold device which performs a binary logic function [34]. Later, the neuron model was refined by the use of Rosenblatt's Perceptron [35] and Widrow's Adaline [36].

The neural network architectures can be classified into three categories: feedback model, feedforward model, and recurrent model. In the feedback neural networks, neurons are connected to one another by feedback paths (synapses) from the outputs to the inputs of neurons. Continuous-valued neurons are normally implemented as electrical circuits and the network dynamics are described by differential equations. A key issue of this type of network is to define an energy function which always decreases during the dynamical evolution. The Hopfield neural network is a one-layer feedback network which consists of interconnected nonlinear analog neurons [37]. In the Hopfield neural network, a gradient descent method is used to seek a local minimum of the given energy function. Generally, the energy function is derived from the given constraints and a cost function. Another canonical circuit model with feedback was proposed by Chua and Kennedy [38, 39]. This model uses integrators as neurons. The parameters of the Chua-Kennedy neural network correspond to the coefficients of the objective function and constraints. The Chua-Kennedy model requires more hardware than the Hopfield model does. However the model is superior to the Hopfield model in solving linear programming problems because it guarantees a stable equilibrium point while the Hopfield model does not.

In feedforward neural networks, synapses only exist between consecutive layers of neurons or between peers. Sample data are needed to train the neural network. Layered

feedforward neural networks were first studied by Rosenblatt [41]. Since then, feedforward multilayered structures and learning algorithms for the training of the neural network have been developed. The most well-known learning algorithm is the error back-propagation algorithm for multilayered feedforward neural networks proposed by Rumelhart [42]. In recurrent neural networks, connections (synapses) in both directions between a pair of layers and within a layer are allowed. The Boltzmann machine is a well-known recurrent neural network with symmetric connections [43]. The Boltzmann machine consists of visible and hidden units where visible units can be divided into input and output units. In the training phase, the Boltzmann machine adjusts the connections such that the states of the visible units have a desired probability distribution. The disadvantage of the Boltzmann machine is that the training requires an extremely long convergence time. Simulated annealing procedures have been proposed to shorten the training convergence time [95].

Since the outputs of a neural network system are the result of cooperative work of all neurons, the system can still produce useful result even when some connections are damaged and/or faults exist in some neurons. In other words, neural network systems exhibit fault tolerance. Because each neuron can function independently, parallelism is inherent in neural networks. Some massively parallel computational ability is essential for many applications requiring high computation capacity. Over the years, neural network models have been applied to various areas. The feedback model has been applied to several combinatorial optimization problems such as the travelling salesman problem [37] and the bipartite subgraph problem [44]. The feedforward model has been applied to robotics [45] and control problems [46]. The recurrent model has been applied to statistical pattern recognition [47] and constraint satisfaction problems [48].

1.3 Research objective

Material characterization using ultrasound can be achieved by extracting acoustic parameters from the return echoes as well as the transmitted signals. There are methods to estimate various acoustic parameters. Echoes returned from non-homogeneous material, such as biological tissues, are the result of collective scattering inside the target. Due to the nature of biological tissues, the return echoes from such target are very complex. No quantitative scheme for estimating the acoustic parameters of non-homogeneous material has been well developed at the present time.

Instead of solving the tissue characterization problem quantitatively, qualitative methods can be used to distinguish normal tissue from abnormal tissue. Frequency domain techniques, especially the dispersive characteristics of the attenuation coefficient, have been widely used in tissue characterization. Features related to the attenuation coefficient, which is strongly frequency dependent, are extracted from the return echoes. Each transducer location on the scanning plane is represented by a feature vector whose dimension equals the number of features. Then, the data set consisting of all feature vectors may have enough information to characterize the target. A modified neural network will be used to classify the data set into two clusters. One cluster represents the normal tissue, while the other represents abnormal tissue. To accomplish this goal, the following steps are taken.

- (1) Time domain return echoes are sampled and stored.
- (2) Various features are extracted from the stored return echoes.
- (3) Develop an appropriate algorithm for the modified neural network to classify the data set.

- (4) Classify the data set and display the clustering result in graphics.

Images reconstructed from the clustering result can be further improved by using image processing techniques.

1.4 Thesis organization

The organization of this dissertation is as follows: In chapter two, some background material of ultrasound is presented. Important acoustic parameters, such as attenuation coefficient and reflection coefficient, are defined. Some discussions of artificial neural network are also included. In chapter three, methods for estimating acoustic parameters in both the time domain and frequency domain are presented. Advantages and limitations of various methods will be examined. In chapter four, features used in feature vectors are presented. Algorithms for improving the existing neural network processing are developed. In chapter five, several synthetic data sets and a data set taken from a real sample are used to test the validity of the theory developed. Results are also compared with those produced by other algorithms. Finally, some conclusions and suggested future research are stated in chapter six.

CHAPTER 2

BACKGROUND

2.1 The acoustic plane wave equations

Ultrasonic waves are pressure waves with frequencies above the audible range. Ultrasonic waves consist of propagating periodic vibrations in an elastic medium where the particles of the medium oscillate about their equilibrium position on either perpendicular or parallel to the direction of propagation. Since the longitudinal vibrations are dominant in the non-destructive applications [49], only those variation are considered in this dissertation. The propagation of the resulting motion-strain effects from the source results in a longitudinal compression wave that transmits acoustic energy away from the source. Pressure and particle velocity are two observable parameters of a propagating acoustic wave.

Assume that the particle at location x_0 in a homogeneous medium undergoes a small compressional force at time t_0 , the relationships between particle velocity v and pressure p are described as

$$\frac{\partial v}{\partial x} = \left(-\frac{1}{k}\right) \frac{\partial p}{\partial t} \quad (2.1)$$

and

$$\frac{\partial p}{\partial x} = (-\rho) \frac{\partial v}{\partial t} \quad (2.2)$$

where k is the elastic coefficient and ρ is the medium density.

Equation 2.1 is the mass continuity equation, while equation 2.2 is the momentum equation. Equations 2.1 and 2.2 show the coupling between particle velocity and pressure. By decoupling these equations, the acoustic plane wave equations can be obtained as

$$\frac{\partial^2 p}{\partial t^2} = \frac{k}{\rho} \frac{\partial^2 p}{\partial x^2} \quad (2.3)$$

and

$$\frac{\partial^2 v}{\partial t^2} = \frac{k}{\rho} \frac{\partial^2 v}{\partial x^2} \quad (2.4)$$

The general solution forms for pressure and particle velocity are

$$p = p(0)e^{j(\omega t - Kx)} \quad (2.5)$$

and

$$v = v(0)e^{j(\omega t - Kx)} \quad (2.6)$$

where K is the wave number given by

$$K = \omega \sqrt{\frac{\rho}{k}} \quad (2.7)$$

The wave number is in general a complex quantity. It consists of the phase constant β and the attenuation constant α .

$$K = \beta - j\alpha \quad (2.8)$$

From equations 2.2, 2.5, and 2.6, the pressure and particle velocity can be related by

$$p = \frac{\omega\rho}{K}v \quad (2.9)$$

The characteristic acoustic impedance is defined as the ratio of pressure to particle velocity. From equation 2.9, the acoustic impedance Z can be expressed as

$$Z \equiv \frac{p}{v} = \frac{\omega\rho}{K} \quad (2.10)$$

For a lossy medium, the acoustic impedance is a complex quantity. For a lossless medium,

the acoustic impedance is a real quantity since there is no attenuation for the propagating wave and the wave number K becomes a real quantity. For a lossless medium, the phase velocity v_p is

$$v_p = \frac{\omega}{K} = \frac{\omega}{\beta} . \quad (2.11)$$

As a result, the acoustic impedance of a lossless medium can be expressed as

$$Z = \rho v_p . \quad (2.12)$$

2.2 Transmission and reflection coefficients

Material characterization using ultrasound is either based on the signal reflected from or transmitted through an interface between two different media. Therefore, it is important to determine the magnitudes of reflection coefficient and transmission coefficient. When an acoustic plane wave arrived at a boundary between two different media, it will be partially reflected. Consider an acoustic plane wave propagating from medium 1 to medium 2, as shown in Figure 2.1, the law of reflection states that the angle of incidence θ_i and the angle of reflection θ_r are equal when the wavelength of the wave is small compared to the physical dimensions of the reflector. That is

$$\theta_i = \theta_r . \quad (2.13)$$

The relationship between the incident angle θ_i and the angle of transmission θ_t is

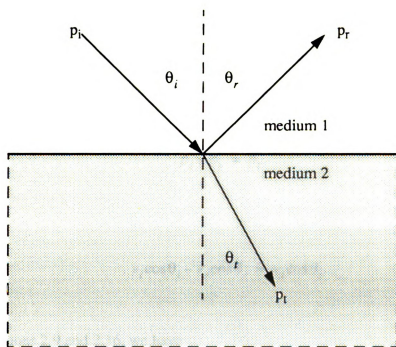


Figure 2.1 Acoustic wave at interface of two different media.

described by the Snell's law,

$$\frac{\sin\theta_i}{\sin\theta_t} = \frac{v_1}{v_2} \quad (2.14)$$

In the equilibrium state, the pressure on both sides remains the same in order to maintain a stationary boundary and the normal component of the particle velocity on both sides must be equal in order to keep the media in contact. Thus,

$$p_i + p_r = p_t \quad (2.15)$$

and

$$v_i \cos\theta_i - v_r \cos\theta_r = v_t \cos\theta_t \quad (2.16)$$

From equations 2.9 and 2.16, we have

$$\frac{p_i K_1 \cos\theta_i}{\rho_1} - \frac{p_r K_1 \cos\theta_r}{\rho_1} = \frac{p_t K_2 \cos\theta_t}{\rho_2} \quad (2.17)$$

From equations 2.15 and 2.17, the pressure reflection coefficient r and transmission coefficient t can be obtained as

$$r \equiv \frac{p_r}{p_i} = \frac{\frac{K_1}{\rho_1} \cos\theta_i - \frac{K_2}{\rho_2} \cos\theta_t}{\frac{K_1}{\rho_1} \cos\theta_i + \frac{K_2}{\rho_2} \cos\theta_t} \quad (2.18)$$

and

$$t \equiv \frac{p_t}{p_i} = \frac{2 \frac{K_2}{\rho_2} \cos \theta_i}{\frac{K_1}{\rho_1} \cos \theta_i + \frac{K_2}{\rho_2} \cos \theta_i} \quad (2.19)$$

For normal incidence, $\theta_i = \theta_r = 0$, the reflection and transmission coefficients are reduced to

$$r = \frac{\frac{K_1}{\rho_1} - \frac{K_2}{\rho_2}}{\frac{K_1}{\rho_1} + \frac{K_2}{\rho_2}} \quad (2.20)$$

and

$$t = \frac{2 \frac{K_2}{\rho_2}}{\frac{K_1}{\rho_1} + \frac{K_2}{\rho_2}} \quad (2.21)$$

Using the acoustic impedance definition given by equation 2.10, these coefficients are simplified to

$$r = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (2.22)$$

and

$$t = \frac{2Z_2}{Z_2 + Z_1} \quad (2.23)$$

2.3 The processing element of artificial neural network

The artificial neural network is a nonlinear network which consist of a massive number of simple processing elements where they are tightly interconnected. The processing element is called an artificial neuron because it performs a similar simplified function of a biological neuron. The mathematical model of an artificial neural network is composed of two important components: neurons and synaptic links. The output signal generated by one neuron propagates to the others through the synaptic links. The new state of the neuron is determined by the linear sum of the weighted input signals and a threshold value. This model is shown in Figure 2.2 where X_i 's is the inputs from other neurons, W_{ij} is the weight between neuron i and neuron j , $f(\cdot)$ is the neuron input/output function, θ_i is the threshold value of neuron i , and Y_i is the output of neuron i .

McCulloch and Pitts proposed a simple mathematical neuron model based on biological computation in 1943 [34]. Their neuron model has a binary input/output function. Since then, several neuron models have been proposed [50]. There are two most commonly used neuron input/output functions: linear function and sigmoid function. These input/output functions are shown in Figure 2.3.

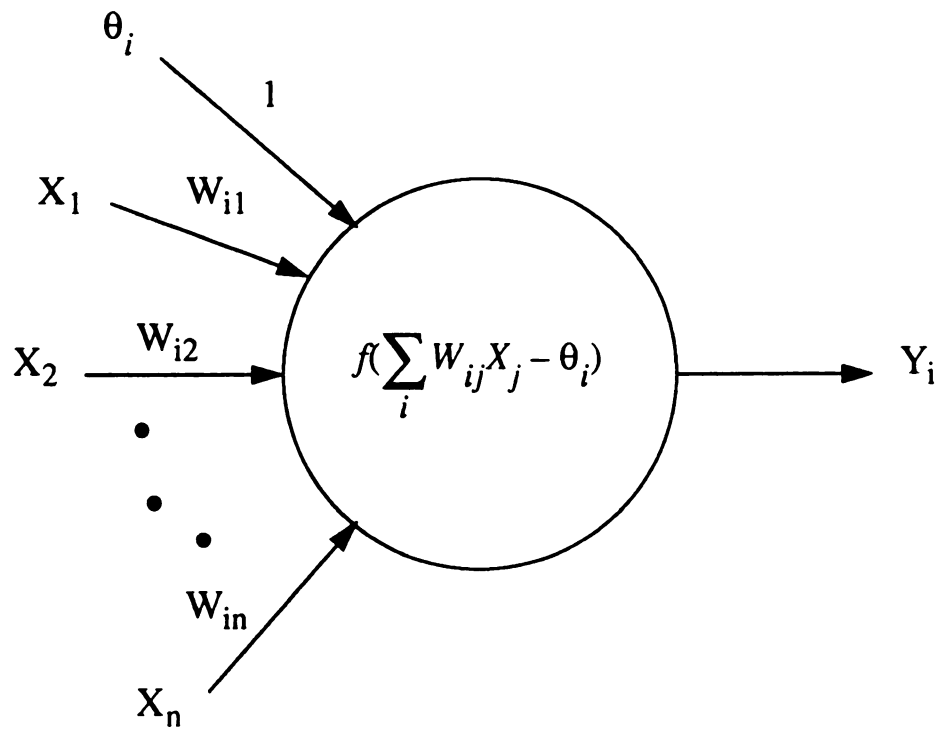


Figure 2.2 Schematic diagram of a neuron.

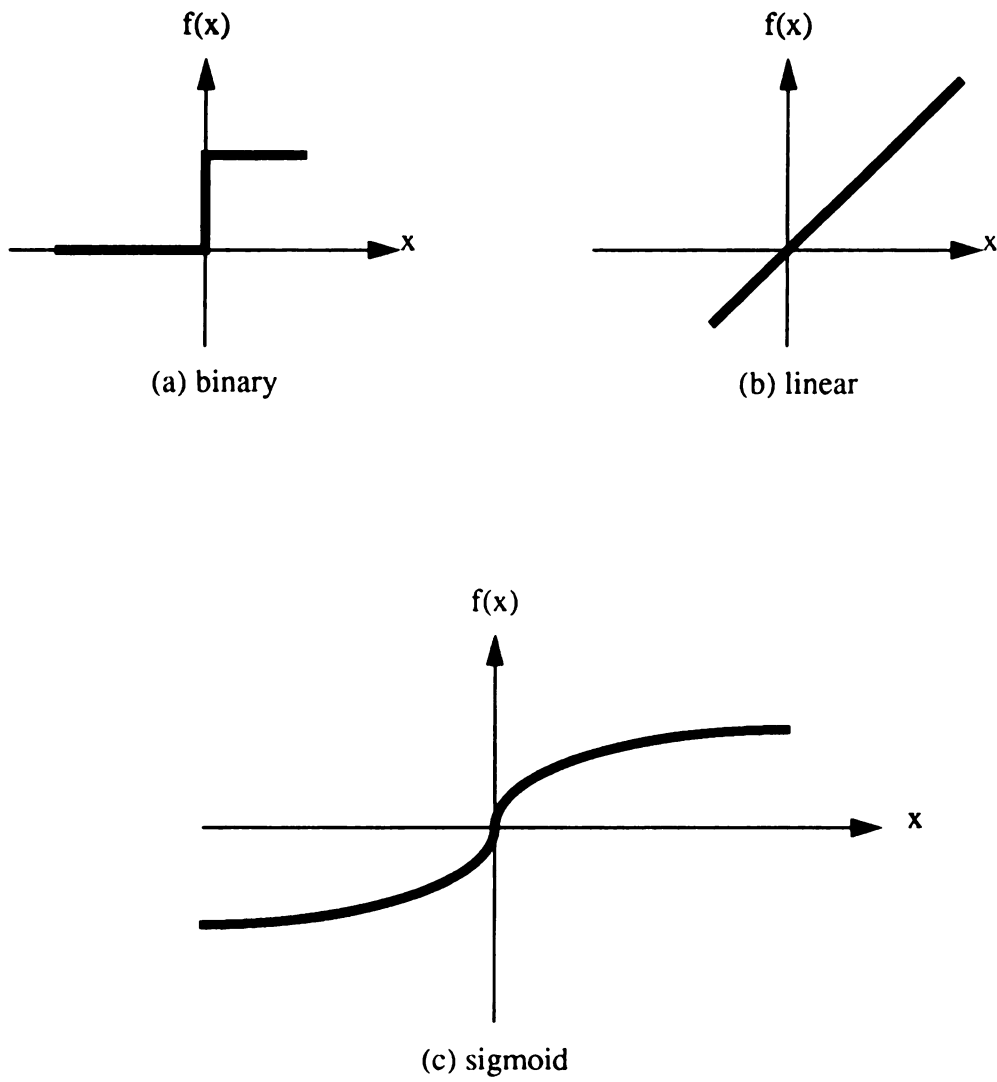


Figure 2.3 Input/output functions of neurons.

Since neurons can process information independently, the parallel processing ability is inherent in artificial neural network. Because neurons are simple processing elements, the fabrication of such parallel system with hundreds of neurons is possible [51, 52, 53]. The output of an artificial neural network is the result of collective efforts of all neurons. It has been shown a few inaccuracies and faults in the hardware will not paralyze the system as a whole. This provides an attractive feature of artificial neural network, the fault tolerance property.

2.4 The structure of Hopfield neural network

The artificial neural network for solving combinatorial optimization problems was first introduced by Hopfield [37]. The differentiable, continuous, and nondecreasing neuron model, using sigmoid function, is used in the Hopfield neural network. Figure 2.4 shows the architecture of the Hopfield neural network. The relationship between output V_i and input U_i of neuron i is given by

$$V_i = \frac{1}{2} (1 + \tanh(\lambda U_i)) \quad (2.24)$$

Note that λ represents a constant gain which changes the steepness of the sigmoid curve. One key issue in the Hopfield neural network is to find a suitable energy function E . In general, the energy function is derived from the given constraints and the cost function. The conductance matrix W in figure 2.4 is determined by the energy function.

In the Hopfield neural network, the symmetric conductance matrix W with zero diagonal elements must be used to guarantee the convergence to the local minimum [54].

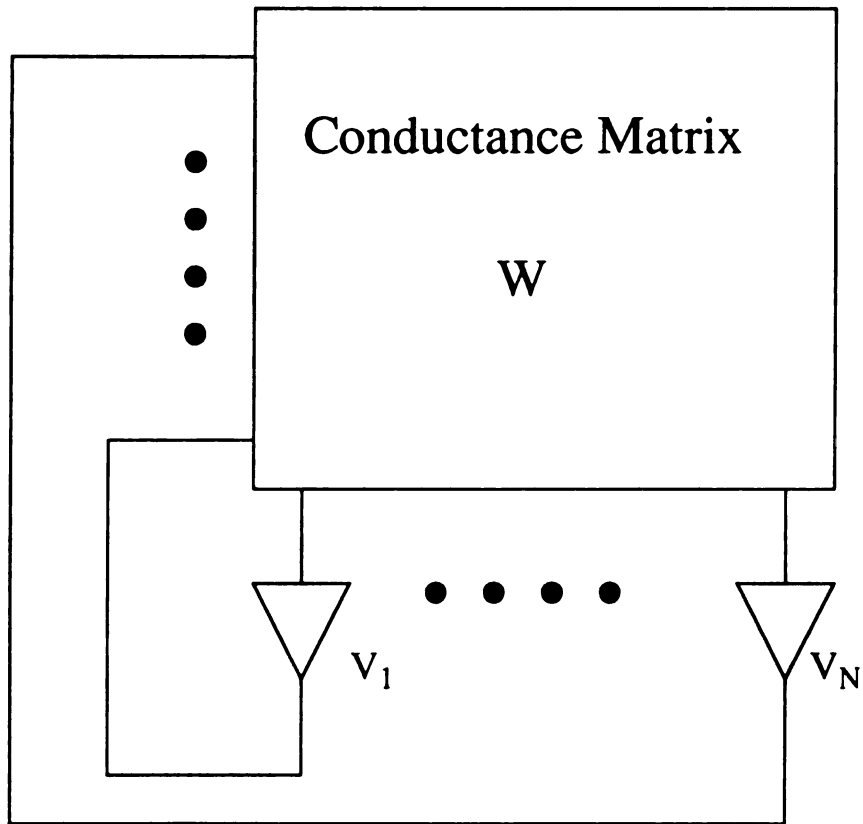


Figure 2.4 Architecture of the Hopfield neural network.

Hopfield mapped the travelling salesman problem onto an $N \times N$ artificial neural network.

The energy function E for an N -city problem is given by

$$E = \frac{A}{2} \sum_{x=1}^N \sum_{i=1}^N \sum_{j \neq i}^N V_{xi} V_{xj} + \frac{B}{2} \sum_{i=1}^N \sum_{x=1}^N \sum_{x \neq y}^N V_{xi} V_{yi} + \frac{C}{2} \left(\sum_{i=1}^N \sum_{x=1}^N V_{xi} - N \right)^2$$

$$+ \frac{D}{2} \sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^N d_{xy} V_{xi} (V_{y,i+1} + V_{y,i-1}) \quad (2.25)$$

where d_{xy} is the distance between city x and city y and A , B , C , and D are constants.

The conductance matrix W is given by

$$W_{x_i, y_j} = -A \delta_{xy} (1 - \delta_{ij}) - B \delta_{ij} (1 - \delta_{xy}) - C - D d_{xy} (\delta_{j, i+1} + \delta_{j, i-1}) \quad (2.26)$$

The gradient descent method is used in Hopfield neural network to seek the local minimum of the given energy function. Consequently, the dynamics of the system follows the motion equation which is given by

$$\frac{dU_i}{dt} = \frac{\partial E}{\partial V_i} \quad (2.27)$$

The Hopfield neural network is guarantee to converge to the local minimum. Unfortunately, the local minimum is not always equivalent to the feasible solution. In other words, the proof of the local minimum convergence in Hopfield neural network does

not always guarantee the feasible solution even through its fast convergence speed is very attractive. There are no systematic methods to find the constants in the energy function at the present time. We will discuss further about these problems and propose a modified Hopfield neural network in chapter four.

CHAPTER 3

TIME-DOMAIN AND FREQUENCY-DOMAIN ANALYSES

3.1 Time-domain technique

The time domain technique extracts material property information, such as attenuation coefficient, acoustic impedance, propagation velocity, from a one-dimensional echo sequence (A-mode signals). Consider the object under investigation consists of multiple homogeneous layers, as shown in Figure 3.1. The parameters α_i , Z_i , r_i , and d_i in Figure 3.1 are the attenuation coefficient, acoustic impedance, reflection coefficient, and layer thickness respectively. By the use of the convolution theorem, the relationship between the received signals $Y_i(t)$ and the incident signal $X(t)$ for the left transducer is

$$Y_1(t) = \int_{-\infty}^{\infty} X(\tau) h_1(t - \tau) d\tau \quad (3.1)$$

and for the right transducer is

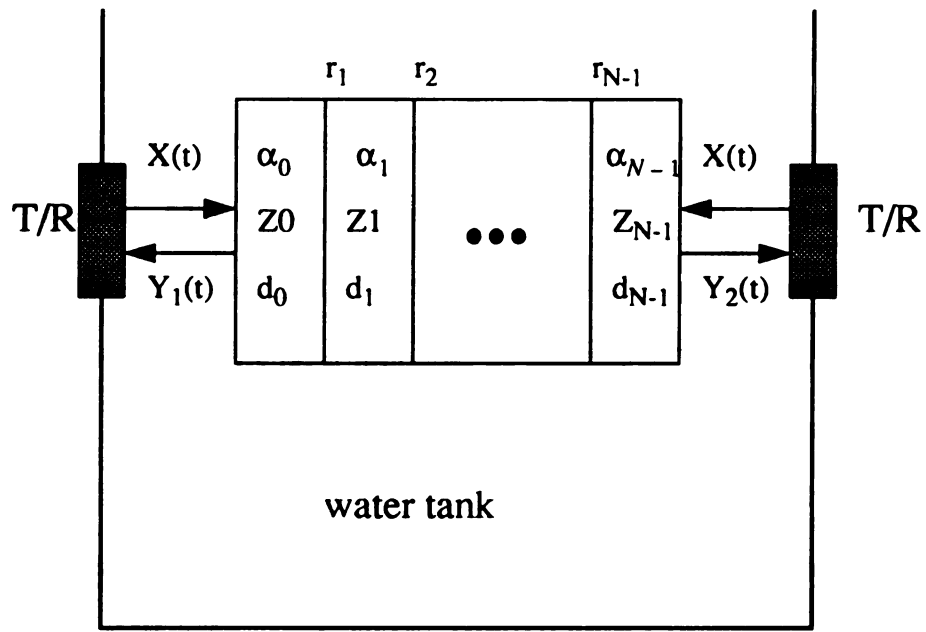


Figure 3.1 Bidirectional interrogation for layered model.

$$Y_2(t) = \int_{-\infty}^{\infty} X(\tau) h_2(t - \tau) d\tau \quad (3.2)$$

$Y_1(t)$ and $Y_2(t)$ are the received signals from the left side and right side respectively, and $h_1(t)$ and $h_2(t)$ are the left side and right side impulse responses of the test object respectively. In general, the process of evaluating the $h_1(t)$ and $h_2(t)$ from the measurements $Y_1(t)$ and $Y_2(t)$ is called the deconvolution process. There are many deconvolution algorithms been proposed [5, 55, 56, 57], but only a few algorithms for obtaining the attenuation properties [58]. The impulse response can be expressed as a sequence of delta functions under the assumption that the acoustic wave propagates through a non-attenuating medium. When a narrow band transducer is used, the attenuation process can be modeled as $e^{-\alpha x}$ where α is the attenuation coefficient at the central frequency of the incident signal and x is the distance travelled. Assuming a narrow band transducer is used, the received signals $Y_i(t)$ can be expressed as a sequence of delayed incident signals. Therefore, the left side and right side impulse responses can be expressed as

$$h_1(t) = \sum_i a_i \delta(t - t_i) \quad (3.3)$$

and

$$h_2(t) = \sum_i b_i \delta(t - t_i) \quad (3.4)$$

where a_i is the amplitude of the echo received by the left transducer which is reflected from the boundary between (i-1)th and ith layers while b_i is the amplitude of the echo received on the right side and reflected from the boundary between (i-1)th and ith layers. The amplitude a_i has the following form,

$$a_i = e^{-\alpha_0 d_0} r_i \prod_{k=1}^{i-1} (1 - r_k^2) e^{-2\alpha_k d_k} \quad (3.5)$$

Similarly, the amplitude b_i has a form of

$$b_i = -e^{-2\alpha_i d_i} r_i \prod_{k=i+1}^{N-1} (1 - r_k^2) e^{-2\alpha_k d_k} \quad (3.6)$$

The minus sign in b_i accounts for the fact that the reflection coefficient changes sign when the incident signal is from the opposite side of the object. The magnitudes a_i and b_i can be read directly from the impulse responses $h_1(t)$ and $h_2(t)$. The impulse responses of dual interrogation is shown in Figure 3.2. The amplitude ratio of successive echoes can be expressed as

$$\frac{a_i}{a_{i+1}} = \frac{r_i}{r_{i+1} (1 - r_i^2) e^{-2\alpha_i d_i}} \quad (3.7)$$

and

$$\frac{b_i}{b_{i+1}} = \frac{r_i (1 - r_{i+1}^2) e^{-2\alpha_i d_i}}{r_{i+1}} \quad (3.8)$$

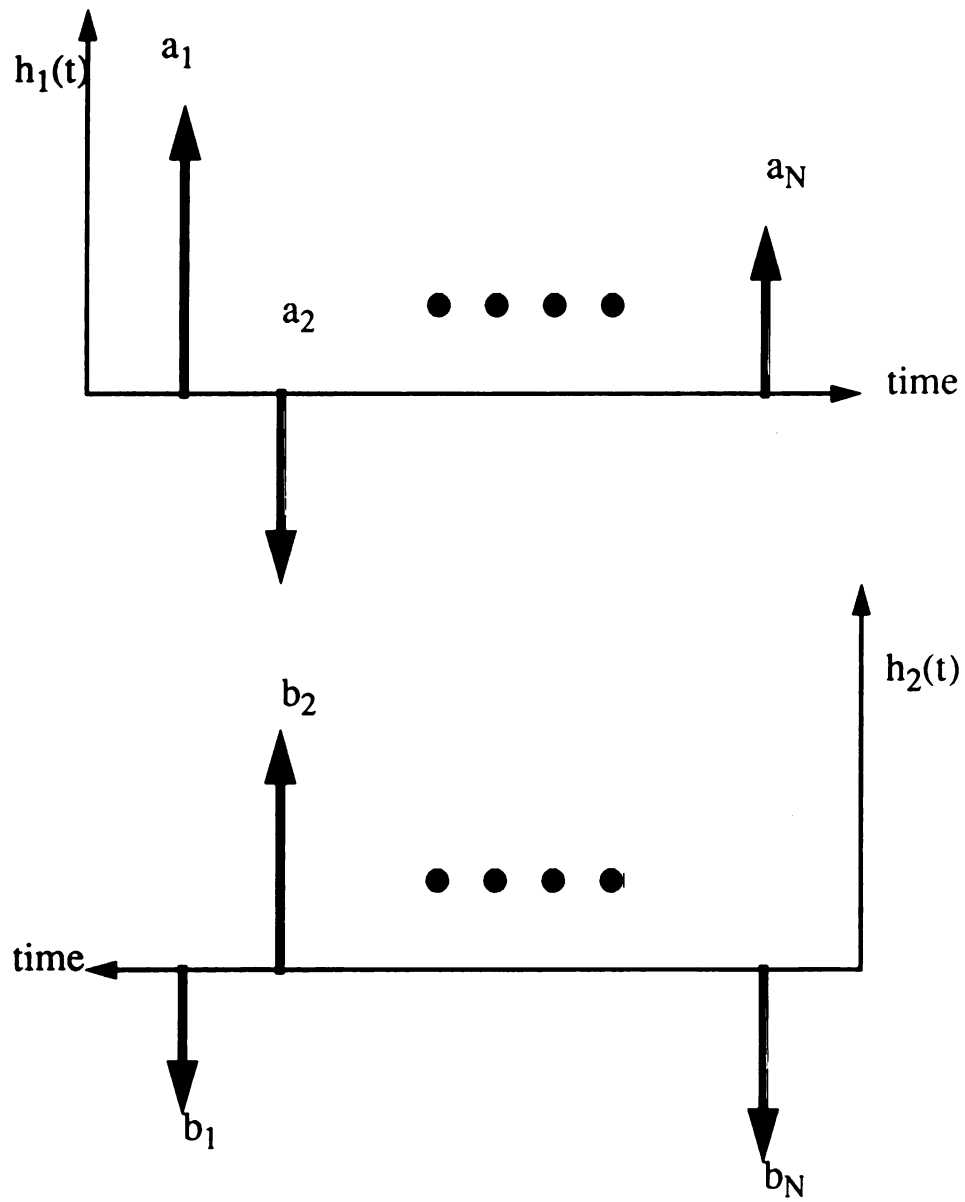


Figure 3.2 The impulse response pairs of dual interrogation.

From equations 3.7 and 3.8, one can obtain

$$\frac{a_i b_i}{a_{i+1} b_{i+1}} = \frac{r_i^2 (1 - r_{i+1}^2)}{(1 - r_i^2) r_{i+1}} \quad (3.9)$$

To simplify the notation, a new parameter R_i is defined as

$$R_i = \frac{r_i^2}{1 - r_i^2} \quad (3.10)$$

Equation 3.9 can then be reduced to

$$\frac{a_i b_i}{a_{i+1} b_{i+1}} = \frac{R_i}{R_{i+1}} \quad (3.11)$$

The R parameters for successive layers are related by

$$R_{i+1} = R_i \left(\frac{a_{i+1} b_{i+1}}{a_i b_i} \right) \quad (3.12)$$

The reflection coefficient r_i can be expressed in terms of R_i as follows

$$r_i = \operatorname{sgn}(a_i) \sqrt{\frac{R_i}{1 + R_i}} \quad (3.13)$$

where

$$\text{sgn}(a_i) = \begin{cases} 1 & \text{for } a_i > 0 \\ -1 & \text{for } a_i < 0 \end{cases} .$$

Therefore, the acoustic impedances for successive layer can be related by

$$Z_i = \frac{1 + r_i}{1 - r_i} Z_{i-1} . \quad (3.14)$$

From equations 3.7 and 3.8, one can obtain

$$\frac{a_i b_{i+1}}{a_{i+1} b_i} = \frac{1}{\left(1 - r_i^2\right) \left(1 - r_{i+1}^2\right) e^{-4\alpha_i d_i}} . \quad (3.15)$$

From equation 3.15, the attenuation coefficient α_i of the i th layer can be expressed as

$$\alpha_i = \frac{1}{4d_i} \ln \left(\frac{a_i b_{i+1}}{a_{i+1} b_i} \left(1 - r_i^2\right) \left(1 - r_{i+1}^2\right) \right) . \quad (3.16)$$

Once the reflection coefficients are determined and the layer thicknesses are available, the attenuation coefficient for each layer can be obtained from equation 3.16. However, the layer thicknesses d_i can not be measured directly from the A-mode echo sequence. It is a distance inferred from the measurement of time delay between two consecutive echoes. A precise measurement of d_i will require knowledge of the mean sound velocity in each

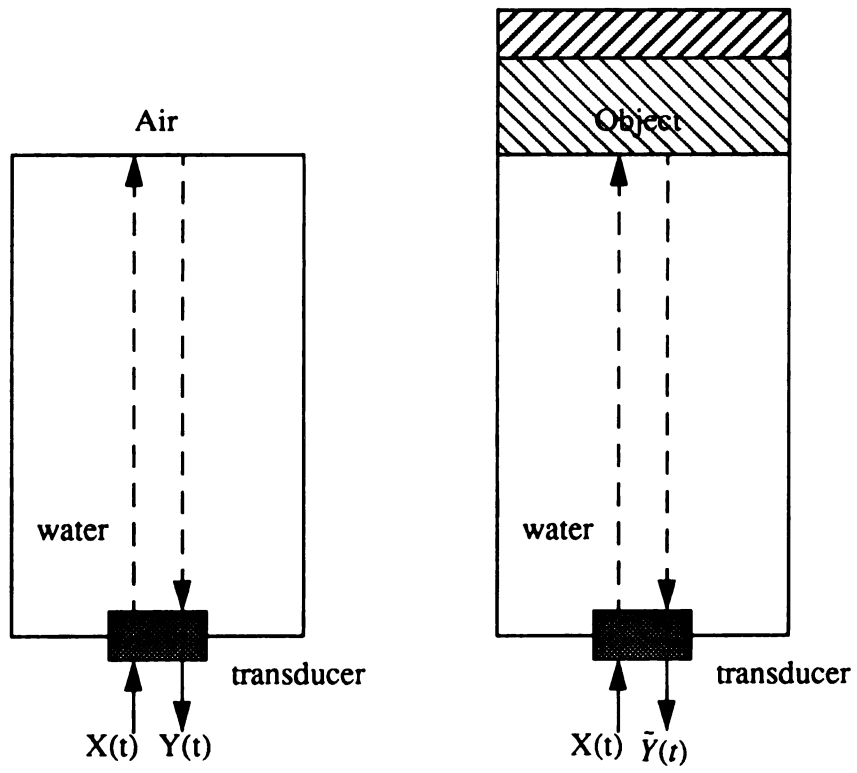


Figure 3.3 Experimental setup for measuring r_1 .

layer. Therefore, we can only obtain the attenuation-velocity product (αv) from the experimental data [90].

The reflection coefficient of the first interface is

$$r_1 = a_1 e^{2\alpha_0 d_0} \quad (3.17)$$

where a_1 is obtained from the impulse response function. Typically the transducer is immersed in water, so the attenuation coefficient α_0 is a known quantity. The distance d_0 between the transducer and the test object can easily be measured. In practice, it is rather difficult to get a replica of the incident pulse from the transducer. A water-air/water-object interface setup, as shown in Figure 3.3, can be used to measure the reflection coefficient r_1 . Since the reflection coefficient at the water-air interface is approximately -1, the amplitudes of $Y(t)$ and $\tilde{Y}(t)$ in Figure 3.3 can be expressed as

$$Y_{peak}(t) = -A_0 e^{-\alpha_0 d_0} \quad (3.18)$$

and

$$\tilde{Y}_{peak}(t) = r_1 A_0 e^{-\alpha_0 d_0} \quad (3.19)$$

From equations 3.18 and 3.19, the reflection coefficient r_1 is then

$$r_1 = -\frac{\tilde{Y}_{peak}(t)}{Y_{peak}(t)} \quad (3.20)$$

Once r_1 is obtained, equations 3.12 and 3.13 can be used to find other reflection coefficients. Consequently, the attenuation-velocity products can be evaluated from equation 3.16.

3.2 Advantages and limitations of time-domain technique

The time domain technique described in the previous section provides a simple way to determine both the reflection coefficient and the attenuation-velocity product simultaneously. Because of its short processing time, the real-time ultrasonic imaging becomes possible by using the time domain technique. Needless to say there are drawbacks in the time domain technique.

- (1) Since the incident signal of a given layer is the transmitted signal from the previous layer, whatever error incurred in the current layer will be carried to the next layer. So the error will be accumulative along the propagation path.
- (2) The time domain technique requires a very precise alignment between the transducer and the object. A minor misalignment could cause a significant error in the end.
- (3) In reality, the incident signal is not truly a narrow band signal. This can cause some analytical error which will be discussed next.

Assuming the incident signal has a Gaussian shape in the time domain,

$$x(t) = e^{j2\pi f_0 t} e^{-\frac{t^2}{2\sigma^2}} \quad (3.21)$$

where f_0 is the central frequency and σ is the standard deviation. The signal and its spectrum are related by the Fourier transform pair [59].

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad (3.22)$$

and

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} df \quad (3.23)$$

The frequency spectrum of the incident signal can thus be obtained as

$$X(f) = \sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(f-f_0)^2} \quad (3.24)$$

The transfer function of the medium can be characterized by

$$H(f) = e^{-\alpha f^n d} e^{-jkd} \quad (3.25)$$

where α is the attenuation coefficient, d is the distance travelled, k is the wave number, and n is the frequency dependent factor. The output signal frequency spectrum can be expressed as

$$Y(f) = X(f)H(f) \quad . \quad (3.26)$$

In reality, most materials have a linear frequency dependency, i.e. $n=1$, equation 3.26 can be reduced to

$$Y(f) = \sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(f-f_0)^2} e^{-\alpha fd} e^{-jkd} \quad (3.27)$$

and

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} Y(f) e^{j2\pi ft} df \\ &= \text{EXP}\left(\frac{\alpha^2 d^2}{8\pi^2 \sigma^2} - \alpha d f_0\right) * \text{EXP}\left(-\frac{2\pi ft - kd}{4\pi f \sigma^2}\right) * \\ &\quad \text{EXP}\left(j2\pi\left(f_0 - \frac{\alpha d}{4\pi^2 \sigma^2}\right)\left(t - \frac{kd}{2\pi f}\right)\right) \quad . \quad (3.28) \end{aligned}$$

The peak amplitude of $y(t)$ occurs at $t = \frac{kd}{2\pi f}$,

$$y_{\text{peak}}(t) = \text{EXP}\left(\frac{\alpha^2 d^2}{8\pi^2 \sigma^2} - \alpha d f_0\right) \quad . \quad (3.29)$$

Hence, it is apparent that the decay of the peak amplitude in the time domain will not just

follow the usual exponential $e^{-\alpha d f_0}$ manner. As a result, the attenuation measurement based on the time-domain analysis becomes inaccurate.

3.3 Advantages and limitations of frequency-domain techniques

If a broadband transducer is used in the ultrasonic imaging system, the attenuation property of the object can be estimated by using the spectral distributions of the incident and reflected signals. The frequency domain approaches can be broadly divided into two categories: the spectral difference [2, 26, 28, 60] and the spectral shift [61, 62, 63] methods. The spectral difference method estimate the attenuation coefficient, $\alpha(f) = \alpha_0 f^n$, by evaluating the spectral difference between the input and output signals. The advantage of the spectral difference method is that no spectral form of the incident signal is required and the attenuation factor n is not restricted to be an integer. Kuc estimates α_0 for liver by comparing the spectra of signals reflected from a planar interface with / without a volume of liver interposed under the assumption of linear frequency dependent attenuation in the liver [61]. Insana modified the spectral difference method to improve the overall measurement accuracy [64]. Insana included the transducer beam diffraction pattern into the data analysis by using empirically determined correction factors. In general, the spectral difference between the incident and reflected signals contains information on both the attenuation coefficient and the reflection coefficient. These material properties can not be determined simply by using the signal information in a single trace of return echo. Since the frequency difference method does not consider reflection and attenuation as separate factors, the estimation results are contaminated and the accuracy is not as good as that of the spectral shift method which will be described next.

When an acoustic signal passes through an medium with attenuation, it experiences a frequency dependent attenuation. The attenuation experienced at higher frequencies is larger than that at lower frequencies. The spectral shift method was first suggested by Serabian [63]. Serabian showed the downshift of the central frequency for a signal propagating through different thicknesses of graphite. Kuc applied this concept to evaluate attenuation in linear frequency dependent soft tissue [61, 65]. Ophir extended this method to nonlinear frequency dependent media [66]. The spectral shift method does not require the knowledge of the reflection coefficient or transmittance to estimate the attenuation coefficient. However, the spectral shift method does require a Gaussian-shaped spectrum for the interrogating pulse. For most ultrasonic imaging system, a Gaussian-shaped pulse with a corresponding Gaussian power spectrum can be produced by slightly modifying the transducer driving voltage and the impedance loading. For a linear frequency dependent attenuation medium, the power spectrum of the signal after going through the medium has also a Gaussian shape, except that the peak is down shifted to a lower frequency. For a nonlinear frequency dependent attenuation medium, the power spectrum of the signal after going through the medium remains in Gaussian shape, but the width becomes narrower. Narayana derived the theoretical relationship between the central frequency downshift and the spectral bandwidth of a signal with a sinc(f) spectrum propagating through attenuating media. The spectral shift method will be further examined in next section.

3.4 Frequency-domain technique

Assuming that the incident signal has a Gaussian-shaped spectrum, the incident signal in the time domain can be expressed as

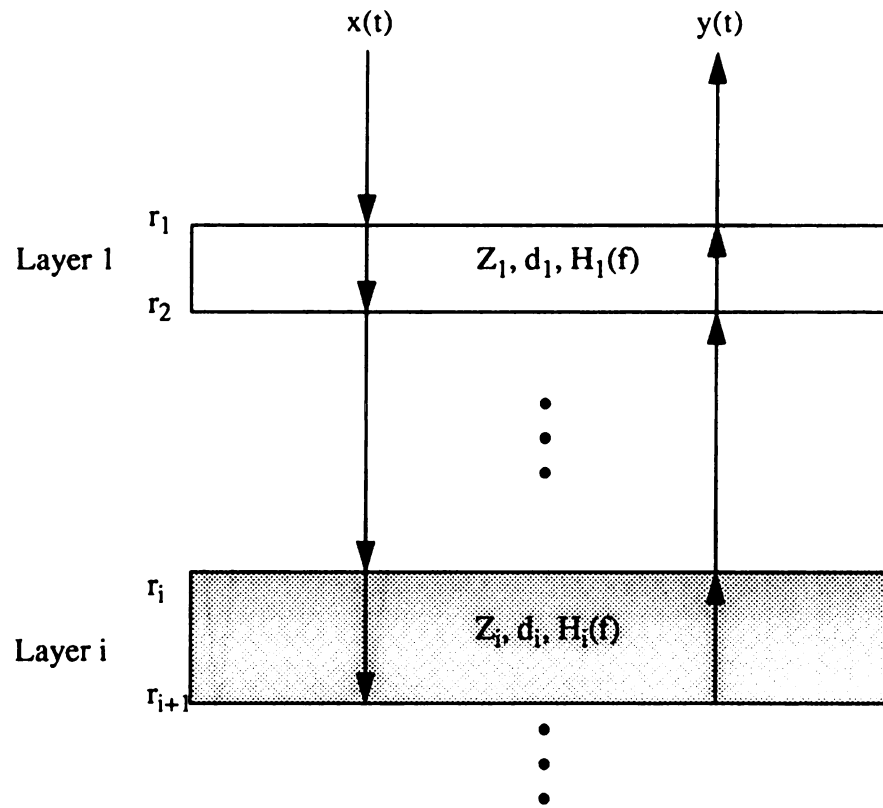


Figure 3.4 Structure of the multi-layered model.

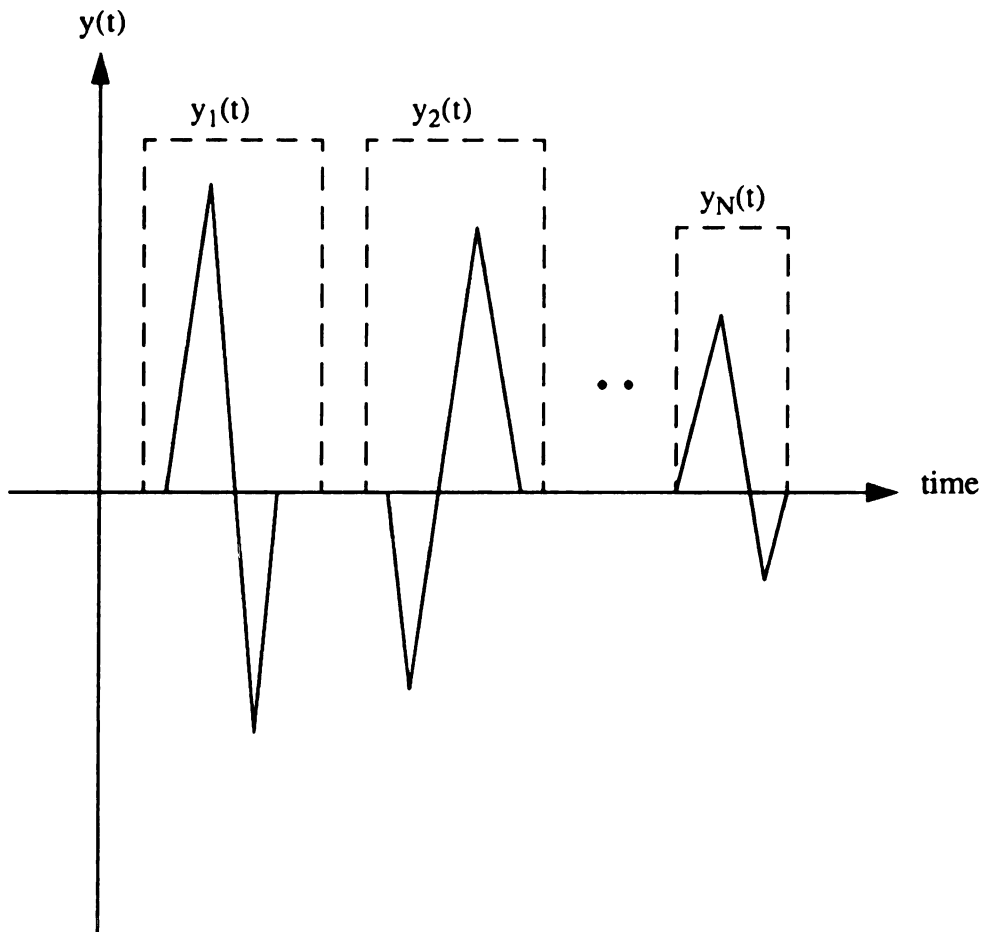


Figure 3.5 Reflected signal gated by windows.

$$x(t) = \text{EXP}\left(-\frac{t^2}{2\sigma_0}\right) \sin(2\pi f_0 t) \quad (3.30)$$

where f_0 is the central frequency and σ_0 is the standard deviation of the Gaussian-shaped envelop. The magnitude of the incident signal spectrum $|X(f)|$ is

$$\begin{aligned} |X(f)| &= \left| \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \right| \\ &= C \cdot \text{EXP}\left(-\frac{(f-f_0)^2}{2\sigma_f^2}\right) \end{aligned} \quad (3.31)$$

where $C = \sqrt{2\pi}\sigma_0$ is a constant and $\sigma_f^2 = \frac{1}{4\pi^2\sigma_0}$ is the variance.

Consider a multi-layered structure, as shown in Figure 3.4, where r_i , Z_i , d_i , and $H_i(f)$ are the reflection coefficient, the acoustic impedance, the layer thickness, and the transfer function of the i th layer respectively. The magnitude of the transfer function can be characterized by

$$|H_i(f)| = \text{EXP}(-\alpha_i f^n d_i) \quad (3.32)$$

where α_i is the attenuation coefficient of the i th layer.

After the incident signal propagates through the structure, echoes come back from various boundary. We can use a window to gate a nonoverlapped sequence of pulses, $y_1(t)$,

$y_2(t), \dots, y_N(t)$, from the reflected signal $y(t)$, as shown in Figure 3.5. The magnitude of each gated signal $|Y_i(f)|$ can be expressed as

$$|Y_1(f)| = |X(f)| |r_1| \quad (3.33)$$

and

$$|Y_{i+1}(f)| = |X(f)| |R_{i+1}(r_1, \dots, r_{i+1})| \prod_{k=1}^i |H_k(f)|^2 \quad (3.34)$$

where $R_{i+1}(r_1, \dots, r_{i+1})$ is the reflection function and can be expressed as

$$R_{i+1}(r_1, \dots, r_{i+1}) = r_{i+1} \prod_{k=1}^i (1 - r_k^2) \quad (3.35)$$

Assuming that the object has a linear frequency dependency and the coupling medium is water whose attenuation is practically negligible. The spectrum of the signal reflected from the $(i+1)$ th interface is

$$|Y_{i+1}(f)| = C_{i+1} |R_{i+1}| \text{EXP} \left(-\frac{(f - f_{i+1})^2}{2\sigma_f^2} \right) \quad (3.36)$$

where C_{i+1} is a frequency independent constant and

$$f_{i+1} = f_0 - 2\sigma_f^2 \sum_{k=1}^i \alpha_k d_k \quad (3.37)$$

From equation 3.36, we observe that the reflected signal spectrum maintains a Gaussian shape as that of the incident signal, but the central frequency has been down shifted by Δf_i .

$$\Delta f_i = 2\sigma_f^2 \alpha_i d_i \quad (3.38)$$

From equation 3.38, the attenuation coefficient is then

$$\alpha_i = \frac{\Delta f_i}{2\sigma_f^2 d_i} \quad (3.39)$$

In order to obtain the reflection coefficient of each layer, the peak spectral amplitudes p_i are needed and can be obtained as

$$p_1 = |X(f_1)| |r_1| \quad (3.40)$$

and

$$p_{i+1} = |X(f_{i+1})| |R_{i+1}| \prod_{k=1}^i |H_k(f_{i+1})|^2 \quad (3.41)$$

The amplitude ratio of successive return echoes is

$$\begin{aligned}
\frac{p_{i+1}}{p_i} &= \frac{|X(f_{i+1})| |R_{i+1}| \prod_{k=1}^i |H_k(f_{i+1})|^2}{|X(f_i)| |R_i| \prod_{k=1}^{i-1} |H_k(f_i)|^2} \\
&= \frac{|X(f_{i+1})| \prod_{k=1}^i |H_k(f_{i+1})|^2}{|X(f_i)| \prod_{k=1}^{i-1} |H_k(f_i)|^2} \left| \frac{r_{i+1} (1 - r_i^2)}{r_i} \right| \quad (3.42)
\end{aligned}$$

From equation 3.42, the reflection coefficient of each layer can be obtained as

$$|r_{i+1}| = \left(\frac{|r_i|}{1 - r_i^2} \right) \cdot \frac{p_{i+1}}{p_i} \cdot \frac{|X(f_i)| \prod_{k=1}^{i-1} |H_k(f_i)|^2}{|X(f_{i+1})| \prod_{k=1}^i |H_k(f_{i+1})|^2} \quad (3.43)$$

Once r_1 is known, equation 3.42 can be used to find the successive reflection coefficients. r_1 can be measured by a simple experimental setup as shown in Figure 3.3 and explained in section 3.1. Once the reflection coefficients r_i are determined, the acoustic impedance can be evaluated as

$$Z_i = Z_{i-1} \cdot \frac{1 + r_i}{1 - r_i} \quad (3.44)$$

CHAPTER 4

MATERIAL CHARACTERIZATION USING PARALLEL PROCESSING

4.1 Material characterization

The measurement and estimation of the attenuation coefficient of biological tissue have received much attention in the field of ultrasonic material characterization. From chapter two, the pressure of the sound wave propagating in the x direction can be expressed as

$$p(x,t) = p(0) \cos (2\pi ft - kx) \quad (4.1)$$

where f is the frequency and k is the wave number. When a sound wave propagates through an attenuating medium, the wave will experience an exponential decay $e^{-\alpha_f x}$ in its amplitude where α_f is the attenuation coefficient at the operation frequency and x is the distance the wave travelled. Therefore, the pressure of a sound wave propagating through an attenuating medium can be expressed as

$$p = p(0)e^{-\alpha_r x} \cos(2\pi f t - kx) \quad (4.2)$$

In an ultrasonic imaging system, the transducer sends out a pulse which composes of a band of frequency components. The pressure P of the pulse propagating through an attenuating medium consists of many frequency components which can be expressed as

$$\begin{aligned} P &= p(f_1)EXP(-\alpha_{f_1} x) \cos(2\pi f_1 t - kx) + \\ & p(f_2)EXP(-\alpha_{f_2} x) \cos(2\pi f_2 t - kx) + \dots \\ &= \sum_i p(f_i)EXP(-\alpha_{f_i} x) \cos(2\pi f_i t - kx) \end{aligned} \quad (4.3)$$

where $p(f_i)$ is the amplitude of the excitation at frequency f_i .

As described in chapter three, the high frequency components will be attenuated more than the lower frequency components. As a result, the spectrum of the pulse is down shifted in frequency domain. Based on this phenomenon, Dines [67] and Schattner [68] have proposed techniques for ultrasonic imaging. Clinical evaluations on liver, breast, and other biological tissues have demonstrated the correlation between pathological status and tissue attenuation properties [69, 70, 71]. The correlation implies that the attenuation measurement of biological tissues may provide a unique method of clinical diagnosis. However, the tissue attenuation measurement remains a difficult task [72, 93]. Difficulties arise from the fact that the acoustic wave is scattered by the biological structure, which is

very complex and dispersive in nature. Although some improvement can be made in data acquisition, tissue characterization based on information obtained from the back-scattered ultrasonic signals is not a trivial task.

4.2 Methodology

Methods for estimating attenuation coefficient can generally be grouped into two categories: time-domain methods and frequency-domain methods. Time-domain methods have the advantage of being easily implemented and thus are suitable for real-time imaging applications. However, the time-domain method can only provide limited amount of information. On the other hand, the frequency-domain method can retrieve more information and provide better accuracy. The trade-off is that the frequency-domain method requires more processing procedure such as windowing, sampling and Fourier transform.

Material characterization utilizing the conventional estimation of attenuation property is not reliable at the present time. In order to improve the accuracy, we proposed a method of applying a clustering technique based on features extracted from the return echoes. To speed up the processing, a parallel algorithm based on a modified Hopfield neural network is proposed. The modified Hopfield neural network is called maximum neural network and will be described in detail in section 4.4. The maximum neural network is used to classify the multi-dimensional data set into clusters after features are extracted from the return echoes. The spatial information is then added to the clustering result for image reconstruction. Several general features of our proposed scheme are outlined as follows:

- (1) For good material identification, both time-domain and frequency-domain information should be utilized.
- (2) Features can be added or deleted from the feature space for optimal classification.
- (3) Knowledge of input spectrum or relationship between attenuation coefficient and frequency is not required. Conventional frequency-domain techniques, however, require such information for material characterization.

The rest of this chapter are as follows. Features extracted from the return echoes are given in section 4.3. The method for selecting independent features in the feature space is also given in section 4.3. The structure of the maximum neural network and its convergence property are described in section 4.4. Clustering using the maximum neural network is explained in section 4.5. The system diagram is shown in Figure 4.1. The stages of data clustering are shown in Figure 4.2.

4.3 Feature selection

Five different features are extracted from each reflected signal at the initial phase of the process. That is, at each scanning position, five frequency-dependent features are extracted from each echo return. These features are the total energy, central frequency, peak frequency, 3-dB bandwidth of echo spectrum, and correlation coefficient between incident and reflected signals.

Total energy

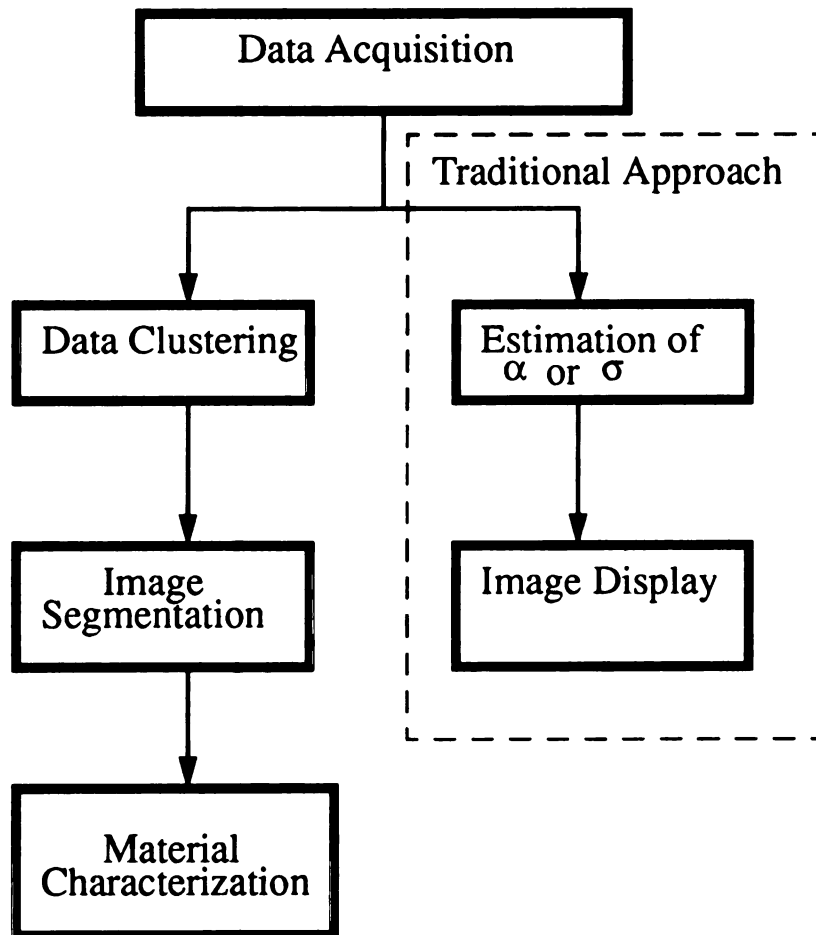


Figure 4.1 System diagram.

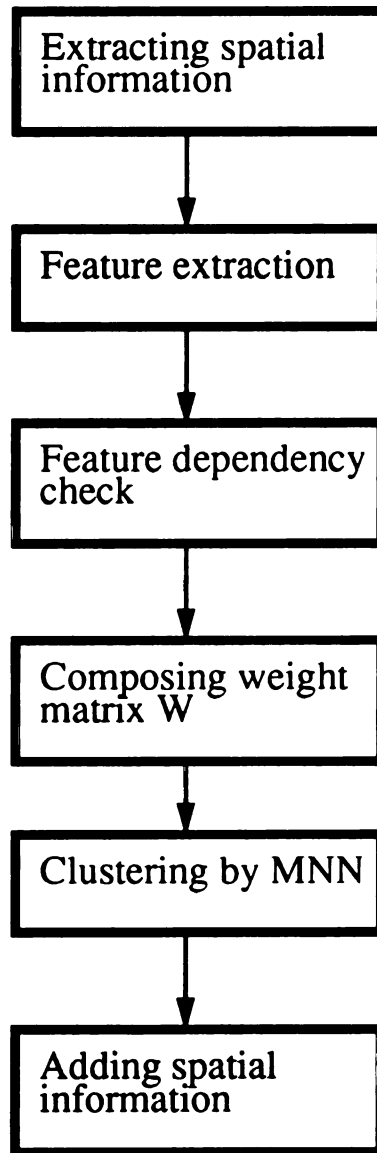


Figure 4.2 Stages of data clustering.

Total energy of the reflected signal is related to the reflection coefficient which contains the information of acoustic impedance of the medium. Let the sampled return echo be $s(i)$, the total energy E_t can be put in the form of

$$E_t = \sum_{i=1}^N |s(i)|^2 \quad (4.4)$$

where N is the number of sampling points of each return echo.

Central frequency, peak frequency, and 3-dB bandwidth

Attenuation coefficient has been shown to be highly related to spectral-shift and spectral-difference of the echo spectrum [73]. This phenomenon cause central frequency and peak frequency shifted downward and the 3-dB bandwidth of the echo spectrum widened. Because of the Gaussian-shaped spectrum, the central frequency can be estimated by the mean frequency F_m and is given as [74]

$$F_m = \frac{\sum_{i=1}^N F_i P(F_i)}{\sum_{i=1}^N P(F_i)} \quad (4.5)$$

where $P(F_i)$ is the i th element of N -point FFT which ranges from the lower 3-dB to the upper 3-dB level. Peak frequency $PK(F)$ is the frequency having the maximum magnitude within the 3-dB bandwidth. Figure 4.3 shows how these three features extracted from an echo return are defined.

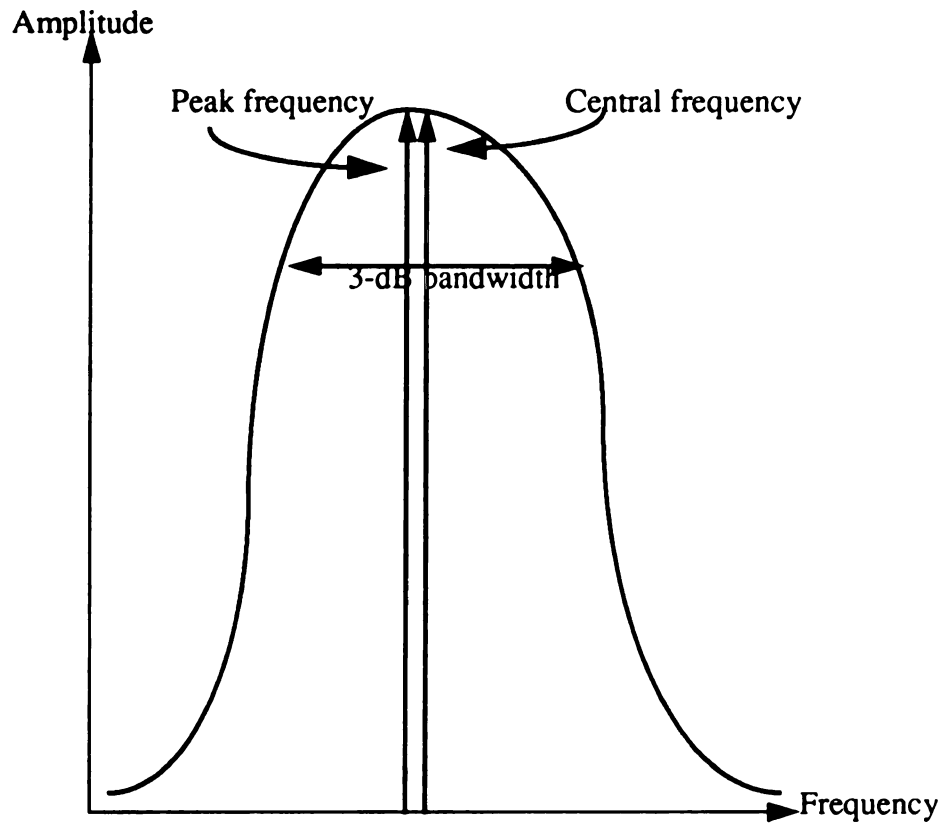


Figure 4.3 Features extracted from the spectrum of a return echo.

Correlation between incident and reflected signals

Correlation between incident and reflected signals at a given interface can provide useful information about the medium under interrogation. Properties such as elasticity, stiffness, velocity, and attenuation are all embedded in this feature.

Feature dependence test

In order to avoid redundancy of features, linear dependence test is used to measure the degree of dependence between features. The linear dependence between two features, i and j , is measured by [94]

$$d(i, j) = \left| \frac{\left(\frac{1}{N} \right) \sum_{k=1}^N (x_{ki} - m_i) (x_{kj} - m_j)}{s_i s_j} \right| \quad (4.6)$$

where s_i and m_i are sample variance and sample mean for feature i respectively. The absolute value is required because the correlation could have either positive or negative value. Nevertheless, the magnitude is used as an index for dependence. If $d(i, j) = 0$, the features i and j are linearly independent. On the other hand, when $d(i, j)$ approaches unity, one of the features can be discarded.

4.4 The structure of maximum neural network

The NP-complete problems are classified as hard problems. It is believed at the present time that no polynomial time algorithm exists for those problems. Therefore, a near-optimum solution is acceptable as long as it can be obtained in a reasonable

computation time. In 1985, Hopfield [37] used an artificial neural network to solve optimization problems such as travelling salesman problems. In a Hopfield neural network, the gradient descent method seeks a local minimum of the given energy function. The energy function is usually composed of two parts. When the first part is minimized the network output represents a valid solution. The second part is described as the cost of the solution. The sum of these two terms forms a specific energy function. In general, the state of the system in the Hopfield neural network is guaranteed to converge to one of the local minima instead of the global minimum [54]. However, the local minimum is not always equivalent to the feasible solution. In other words, the proof of the local minimum convergence in the Hopfield neural network does not always guarantee a feasible solution although its fast convergence speed is very attractive. Furthermore, the determination of the parameters in the energy function is usually based on a process of trial and error. No theoretical analysis has been reported for the general case at the present time.

A modified Hopfield neural network for classifying an N-point data set into two classes is proposed in this section. The modified Hopfield neural network is called the maximum neural network and is shown in Figure 4.4. The maximum neural network consists of N clusters and each cluster is composed of two processing elements. The total number of required processing elements is then 2N. Processing element of the i th neuron in the x th cluster has an input $U_{x,i}$ and an output $V_{x,i}$. The input/output function of the neuron xi is given by

$$V_{x,i} = \begin{cases} 1 & \text{if } U_{x,i} = \max\{U_{x,1}, U_{x,2}\} \\ 0 & \text{otherwise} \end{cases}$$

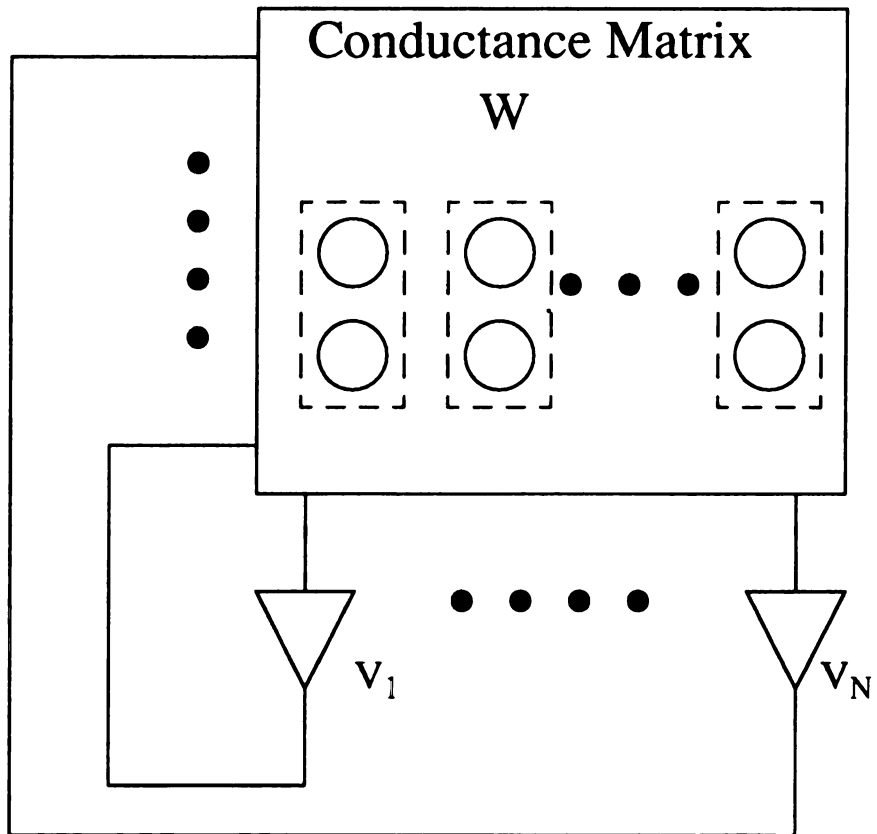


Figure 4.4 Architecture of maximum neural network.

Therefore, only one processing element per cluster is always encouraged to fire in the maximum neural network. The gradient descent method is also used in maximum neural network. The approach is very similar to the Hopfield neural network which seeks the local minimum of a given energy function. The dynamics of the system follows the motion equation which is given by

$$\frac{dU_{x,i}}{dt} = -\frac{\partial E}{\partial V_{x,i}} \quad (4.7)$$

Convergence property of the maximum neural network

Theorem 4.1 and Theorem 4.2 below are introduced to prove that the proposed system can always converge to a stable status. The convergence of the Hopfield neural network is given in [50] and is restated in Theorem 4.1 for completeness.

Theorem 4.1:

$\frac{dE}{dt} \leq 0$ is guaranteed by the conditions: $\frac{dU_i}{dt} = -\frac{\partial E}{\partial V_i}$ and $V_i = f(U_i)$ where $f(U_i)$ is a nondecreasing function.

Proof:

$$\frac{dE}{dt} = \sum_i \frac{dU_i}{dt} \frac{\partial E}{\partial U_i} \frac{\partial V_i}{\partial U_i}$$

$$= - \sum_i \left(\frac{dU_i}{dt} \right)^2 \frac{dV_i}{dU_i} \text{ where } \frac{\partial E}{\partial V_i} \text{ is replaced by } - \frac{dU_i}{dt} \text{ (condition 1)}$$

$$\leq 0 \text{ where } \frac{dV_i}{dU_i} > 0 \text{ (condition 2) Q.E.D.}$$

In Theorem 4.2, the convergence of the maximum neural network for the clustering problem is guaranteed.

Theorem 4.2:

$\frac{dE}{dt} \leq 0$ is guaranteed by the two conditions, they are

$$(1) \quad \frac{dU_{x,i}}{dt} = - \frac{\partial E}{\partial V_{x,i}} \text{ and}$$

$$(2) \quad V_{x,i} = 1 \text{ if } U_{x,i} = \max \{ U_{x,1}, U_{x,2} \}, 0 \text{ otherwise.}$$

Proof:

The derivative of the energy E with respect to time t is

$$\begin{aligned} \frac{dE}{dt} &= \sum_x \sum_i \frac{dU_{x,i}}{dt} \frac{dV_{x,i}}{dU_{x,i}} \frac{\partial E}{\partial V_{x,i}} \\ &= - \sum_x \sum_i \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}}, \text{ where } \frac{\partial E}{\partial V_{x,i}} \text{ is replaced by } - \frac{dU_{x,i}}{dt} \text{ (condition 1)}. \end{aligned}$$

Since $\frac{dU_{x,i}}{dt} \equiv \frac{U_{x,i}(t+dt) - U_{x,i}(t)}{dt}$ and $\frac{dV_{x,i}}{dU_{x,i}} \equiv \frac{V_{x,i}(t+dt) - V_{x,i}(t)}{U_{x,i}(t+dt) - U_{x,i}(t)}$, if we let

$U_{x,a}(t+dt)$ be the maximum at time $t+dt$ and $U_{x,b}(t)$ be the maximum at time t in the

cluster x , then

$$U_{x,a}(t+dt) = \max\{U_{x,1}(t+dt), U_{x,2}(t+dt)\}$$

and

$$U_{x,b}(t) = \max\{U_{x,1}(t), U_{x,2}(t)\}.$$

It is necessary and sufficient to consider the following two cases:

$$(1) a = b$$

$$(2) a \neq b .$$

If case (1) is satisfied, then there is no state change in the cluster x . As a result,

$\sum_i \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}}$ must be zero. If case (2) is satisfied, then

$$\begin{aligned}
\sum_i \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}} &= \left(\frac{U_{x,a}(t+dt) - U_{x,a}(t)}{dt} \right)^2 \frac{V_{x,a}(t+dt) - V_{x,a}(t)}{U_{x,a}(t+dt) - U_{x,a}(t)} + \\
&\quad \left(\frac{U_{x,b}(t+dt) - U_{x,b}(t)}{dt} \right)^2 \frac{V_{x,b}(t+dt) - V_{x,b}(t)}{U_{x,b}(t+dt) - U_{x,b}(t)} \\
&= \left(\frac{U_{x,a}(t+dt) - U_{x,a}(t)}{dt} \right)^2 \frac{1}{U_{x,a}(t+dt) - U_{x,a}(t)} + \\
&\quad \left(\frac{U_{x,b}(t+dt) - U_{x,b}(t)}{dt} \right)^2 \frac{-1}{U_{x,b}(t+dt) - U_{x,b}(t)} \\
&= \frac{U_{x,a}(t+dt) - U_{x,a}(t)}{(dt)^2} - \frac{U_{x,b}(t+dt) - U_{x,b}(t)}{(dt)^2} \\
&= \frac{1}{(dt)^2} (U_{x,a}(t+dt) - U_{x,b}(t+dt) + U_{x,b}(t) - U_{x,a}(t)) \\
&> 0.
\end{aligned}$$

Notice that $U_{x,a}(t+dt)$ is the maximum at time $t+dt$ and $U_{x,b}(t)$ is the maximum at time t in the cluster x . The contribution from each term is either zero or positive, therefore

$$\sum_i \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}} \geq 0 \text{ and } - \sum_x \sum_i \left(\frac{dU_{x,i}}{dt} \right)^2 \frac{dV_{x,i}}{dU_{x,i}} \leq 0 \Rightarrow \frac{dE}{dt} \leq 0 \text{ Q.E.D.}$$

Theorem 4.1 guarantees the convergence of the system. Theorem 4.2 states that the solution quality improves as the time elapses until no further improvement can be made. The maximum neural network performs a parallel improvement algorithm which can be implemented either on the sequential machine or on the parallel machine. The termination condition is given by the convergence state of the system. As long as the system reaches an equilibrium state, the execution is terminated. The equilibrium state is defined that all firing neurons have the smallest change rate of the input per cluster. The condition of the equilibrium state is as follows

$$V_{x,i}(t) = 1 \text{ and } \frac{d}{dt} U_{x,i}(t) = \min \left\{ \left| \frac{d}{dt} U_{x,1}(t) \right|, \left| \frac{d}{dt} U_{x,2}(t) \right| \right\} \text{ for } x = 1, \dots, N. \quad (4.8)$$

4.5 Classification using maximum cut

When an N-point data set is classified into two classes, we would like to have data points in the same class to be as similar as possible, while data points in different classes are as different as possible. If the N-point data set is mapped onto an N-vertex graph and the relationship between data points is regarded as weights on the edge between vertices, the clustering problem can be viewed as the maximum cut problem. As explained in section 4.4, the energy function E of the maximum neural network is the same as that of the Hopfield neural network. Those parameters in the energy function of the maximum

neural network need to be determined. If the clustering problem is mapped onto the maximum cut problem while using the maximum neural network, we will see that the number of the parameters will be reduced to one. The advantages of the maximum neural network will be presented next. First, the maximum cut problem is defined in the following manner [75].

Maximum Cut Problem:

Instance: Graph $G = (V, E)$, weight $W(e) \in Z^+$ for each $e \in E$, positive integer K .

Question: Is there a partition V into two disjoint sets V_1 and V_2 such that the sum of the weights of the edges from E that have one endpoint in V_1 and one endpoint in V_2 is at least K ?

Optimization description:

$$\text{maximize } \sum_{x=1}^N \sum_{y=1, y \neq x}^N \sum_{i=1}^2 \sum_{j=1, j \neq i}^2 W_{x,y} V_{x,i} V_{y,j} \quad V_{x,i} \in \{1,0\}$$

$$\text{subject } \sum_{i=1}^2 V_{x,i} = 1 \text{ for } x = 1 \text{ to } N \text{ where } N \text{ is the number of vertices.}$$

Note that $W_{x,y}$ represents the weight of the edge between vertex x and vertex y and $W_{x,x} = 0$. If $V_{x,i} = 1$, the vertex x belongs to the i th partition.

Remark:

To maximize $\sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 \sum_{j \neq i}^2 W_{x,y} V_{x,i} V_{y,j}$ is equivalent to minimize

$$\sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 W_{x,y} V_{x,i} V_{y,i}.$$

Proof:

$$\text{Let Sum} = \sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 \sum_{j=1}^2 W_{x,y} V_{x,i} V_{y,j}$$

Let M^* be the maximum of $\sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 \sum_{j \neq i}^2 W_{x,y} V_{x,i} V_{y,j}$ and

S^* be the minimum of $\sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 W_{x,y} V_{x,i} V_{y,i}$, then

$$\text{Sum} = \sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 \sum_{j \neq i}^2 W_{x,y} V_{x,i} V_{y,j} + \sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 W_{x,y} V_{x,i} V_{y,i} = M^* + S^*.$$

Since M^* is the maximum of the maximum cut problem, the maximum cut can be obtained by the minimization of S^* . Q.E.D.

The maximum neural network model is composed of $2N$ processing elements for solving the maximum cut problem of an N -vertex graph. The input/output function for

each neuron is given by

$$V_{x,i} = \begin{cases} 1 & \text{if } U_{x,i} = \max\{U_{x,1}, U_{x,2}\} \\ 0 & \text{otherwise} \end{cases}$$

where x and i represent the vertex and partition respectively. With this function, there is always one neuron fired for each vertex x . The maximum cut problem can be solved by minimizing the energy function E which can be expressed as

$$E = \frac{C}{2} \sum_{x=1}^N \sum_{y \neq x}^N \sum_{i=1}^2 W_{x,y} V_{x,i} V_{y,i}. \quad (4.9)$$

The motion equation for the x_i -th neuron is

$$\frac{dU_{x,i}}{dt} = -C \sum_{y \neq x}^N W_{x,y} V_{y,i}. \quad (4.10)$$

It should be noted that the parameter C does not affect the performance of the system. For simplicity, it is assigned to be unity hereafter.

In summary, the maximum neural network approach has the following features for the clustering problem:

- (1) The maximum neural network always generates a valid solution. Wherever or whenever the system converges, the corresponding output is a valid solution.

Note that none of the existing algorithms based on Hopfield neural networks can guarantee this property.

- (2) No trial-and-error parameters need to be evaluated in the maximum neural network. None of the existing algorithms based on Hopfield neural networks can avoid this trial and error process.
- (3) The maximum neural network has an exact condition for convergence listed in equation 4.8. Most of the algorithms used in the neural networks, the states of convergence are defined by a tolerance range. None of the existing algorithms based on Hopfield neural networks clearly defines the condition of the system convergence.

The parallel algorithm based on the maximum neural network has been implemented in C language on a sequential machine such as Sun SPARC 10. The following procedure describes the proposed parallel algorithm based on the first order Euler method.

step 0: Set $t = 0$

step 1: for $x = 1$ to N do

for $i = 1$ to 2 do in parallel

$$U_{x,i}(t) \leftarrow \text{random number} \in (-1, 1)$$

step 2: for $x = 1$ to N do

for $i = 1$ to 2 do in parallel

$$V_{x,i}(t) = 1 \text{ if } U_{x,i}(t) = \max \{ U_{x,1}(t), U_{x,2}(t) \}$$

0 otherwise

step 3: for $x = 1$ to N do

for $i = 1$ to 2 do in parallel

$$\Delta U_{x,i}(t) = - \sum_{y=0}^N W_{x,y} V_{y,i}$$

step 4: for $x = 1$ to N do

for $i = 1$ to 2 do in parallel

$$U_{x,i}(t+1) = U_{x,i}(t) + \Delta U_{x,i}(t)$$

step 5: Increment t by 1.

step 6: If the state of the system reaches an equilibrium state then stop, otherwise go to

step 2.

CHAPTER 5

SIMULATION AND EXPERIMENTAL RESULTS

5.1 Maximum independent set problem

In order to verify the validity of the proposed algorithm, two types of problem are solved by the maximum neural network: the maximum independent set problem and the tissue characterization problem. For the maximum independent set problem, a massive number of simulation runs are performed. The simulation results are compared with results published by other researchers. For the tissue characterization problem, the result is compared with the result of the conventional C-scan imaging. The experimental setup is also described.

An independent set in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that, for all $v, w \in S$, the edge (v, w) is not in E . The maximum independent set problem is to find the maximum size of the independent set in G . The problem is one of the fundamental problems in graph theory and has many useful applications such as RNA secondary structure prediction [76]. It is also known to be NP-complete [75]. Algorithms for the

maximum independent set problem have widely studied. Gavril developed an $O(m^3)$ time algorithm for the maximum independent set problem in a circle graph, where m is the number of edges in the graph [77]. Supowit proposed an $O(m^2)$ time algorithm on the circle graphs [79]. Masuda gave an $O(m \log m)$ time algorithm on the circle graphs [80]. Hsu gave an $O(n^4)$ time algorithm on the planar perfect graphs, where n is the number of vertices [81]. Several parallel algorithms have been proposed by Karp [82], Luby [83], and Goldberg [84], where the computation time is $O(\log^4 n)$. Takefuji gave a nearly $O(1)$ time parallel algorithm on circle graphs [76].

Given a graph G , a clique is a maximal complete subgraph of G . The maximum clique problem is to find the maximum complete subgraph of G . The problem is known to be NP-complete [75]. It is also useful and has been investigated in many areas including clustering analysis, classification theory, graph coloring, information retrieval systems, and VLSI circuit design [85]. Pardalos proposed an algorithm where the maximum clique problem is formulated and solved as a linear constrained indefinite quadratic global optimization problem. Although the supercomputer Cray 2 was used, their algorithm could not solve larger than 75-vertex graph problem [86]. Carraghan proposed an algorithm based on a partial enumeration [87]. Although it could solve up to 3,000-vertex graph problems on the mainframe IBM 3090, it required a prohibitively long computation time even for a middle-size graph.

In fact, the maximum independent set problem and the maximum clique problem are equivalent as pointed out in [75]. For clarity, the equivalence is restated here. Given $G = (V, E)$, two vertices v, w are adjacent if $(v, w) \in E$. A set S of vertices is independent if $(v, w) \notin E$ for all $v, w \in S$. A set of vertices is a clique if $(v, w) \in E$ for all pairs of distinct vertices $v, w \in S$. Clearly $S \subseteq V$ is an independent set of G if and only if S is a clique of \bar{G} , where \bar{G} is the complement of G . Therefore, finding the maximum independent set of

graph G is equivalent to finding the maximum clique of graph \bar{G} .

The equivalence between the maximum independent set problem and the unconstrained quadratic zero-one problem has been pointed out by Pardalos [86]. The equivalence between the quadratic zero-one problem and the maximum cut problem has also been pointed out by Hammer [88] and Barahona [89]. The equivalence between the maximum independent set problem and the maximum cut problem will be established in this section. The procedure of using the maximum neural network to solve the maximum independent set problem will also be presented in this section.

The unconstrained quadratic zero-one problem can be formulated as follows:

$$\text{minimize } f(\mathbf{X}) = \mathbf{C}^T \mathbf{X} + \mathbf{X}^T \mathbf{Q} \mathbf{X} \quad (5.1)$$

where \mathbf{Q} is an $N \times N$ symmetrical rational matrix with zero diagonal elements, \mathbf{X} is an integer N -vector and \mathbf{C} is a rational N -vector. Equation 5.1 can be reformulated as

$$\text{minimize } f(\mathbf{X}) = \sum_{i=1}^N c_i x_i + \sum_{i=1}^N \sum_{j=1}^N q_{i,j} x_i x_j \quad x_i \in \{0, 1\} \quad (5.2)$$

where $q_{i,j}$ is the ij -th element of \mathbf{Q} , x_i is the i th element of \mathbf{X} and c_i is the i th element of \mathbf{C} . If \mathbf{A}_G is the adjacency matrix of G and \mathbf{I} is the identity matrix, the minimization of the following unconstrained quadratic zero-one problem is equivalent to finding the maximum independent set in G [86]:

$$\text{minimize } f(\mathbf{X}) = \mathbf{X}^T (\mathbf{A}_G - \mathbf{I}) \mathbf{X} \quad x_i \in \{0, 1\} \quad (5.3)$$

Equation 5.3 is reduced to

$$\text{minimize } f(X) = \sum_{i=1}^N (-1)x_i + \sum_{i=1}^N \sum_{j=1}^N q_{i,j}x_ix_j \quad x_i \in \{0, 1\} \quad (5.4)$$

where $q_{i,j} = 1$ if there exists an edge between vertex i and vertex j in graph G , 0 otherwise.

The quadratic problem also is one of the NP-complete problems. The equivalence between the quadratic zero-one problem and the maximum cut problem has been proven [88]. Barahona also proved the relationship [89]. For completeness, the reduction process is also presented here. Let $s_i = 2x_i - 1$, equation 5.4 can be reduced to the following form

$$\begin{aligned} f(s) &= \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j}s_is_j + \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j}s_i + \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j}s_j - \sum_{i=1}^N \frac{1}{2}s_i + C_1 \\ &= \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j}s_is_j + \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j}s_i + \sum_{j=1}^N \sum_{i=1}^N \frac{1}{4}q_{j,i}s_j - \sum_{i=1}^N \frac{1}{2}s_i + C_1 \\ &= \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j}s_is_j + \sum_{i=1}^N \left(\sum_{j=1}^N \frac{1}{2}q_{i,j} - \frac{1}{2} \right) s_i + C_1 \end{aligned}$$

where $s_i \in \{1, -1\}$ and $C_1 = \sum_{i=1}^N \sum_{j=1}^N \frac{1}{4}q_{i,j} - \frac{N}{2}$ with an additional variable $s_0 = 1$ and

$$W_{0,i} = W_{i,0} = \frac{1}{4} \left(\sum_{j=1}^N q_{i,j} - 1 \right), \quad 1 \leq i \leq N$$

$$W_{i,j} = W_{j,i} = \frac{1}{4} q_{i,j}, \quad 1 \leq i \leq N.$$

Equation 5.4 can then be transformed into

$$f(s) = \sum_{i=0}^N \sum_{j=0}^N W_{i,j} s_i s_j + C_1 \quad \text{with } s_0 = 1 \text{ and } s_i \in \{1, -1\} \quad (5.5)$$

A graph $G_M = (V, E)$ with node set $V = \{0, 1, 2, \dots, N\}$ and edge set E can be defined where $W_{i,j}$ represents the edge weight between vertex i and vertex j . The assignment of each s_i , either $+1$ or -1 , corresponds to a partition of V into $V^+ = \{i \in V \mid s_i = +1\}$ and $V^- = \{i \in V \mid s_i = -1\}$. Note that the vertex 0 has been assigned to the partition V^+ . Because C_1 is a constant, the minimization of equation 5.4 can be rewritten as

$$\begin{aligned} \text{minimize } f(s) &= \sum_{s_i, s_j \in V^+} W_{i,j} + \sum_{s_i, s_j \in V^-} W_{i,j} - \sum_{s_i \in V^+ \text{ and } s_j \in V^-} W_{i,j} + C_1 \\ &= C_1 + C_2 - 2 * \left[\sum_{s_i \in V^+ \text{ and } s_j \in V^-} W_{i,j} \right] \end{aligned} \quad (5.6)$$

where $C_2 = \sum_{i=0}^N \sum_{j=0}^N W_{i,j}$ and C_2 is a constant.

The minimization of equation 5.3 has been transformed into equation 5.6. Equation 5.6 is equivalent to the maximum cut problem. In other words, finding the maximum independent set in graph G and finding a maximum cut in graph G_M are equivalent. Therefore, the maximum independent set problem can be solved by substituting $W_{i,j}$ in equation 5.6 into $W_{x,y}$ in equation 4.9.

The relationship between the set of local minimum and the maximum independent set in a given graph is presented next.

Theorem 5.1:

Given a graph G , a graph G_M is defined by following equation 5.5. Then every local minimum in the maximum cut problem for the graph G_M corresponds to a maximum independent set in the graph G if the maximum neural network is used.

Proof:

Suppose that the system reaches an equilibrium state, i.e. local minimum, at $t = T$, then there exists a partition of V^+ and V^- in the graph G_M . Since the system stays in the equilibrium state and the maximum neural network is used, the following conditions must be satisfied

$$\text{for } i \in V^+, V_{i,1}(T) = 1 \text{ and } \frac{d}{dt}U_{i,1}(T) > \frac{d}{dt}U_{i,2}(T)$$

for $i \in V^-$, $V_{i,2}(T) = 1$ and $\frac{d}{dt}U_{i,1}(T) < \frac{d}{dt}U_{i,2}(T)$.

Suppose that the final state does not correspond to an independent set in the graph G , then \exists vertex a and vertex $b \in V^+$ and there exists an edge between vertex a and vertex b in G . It is defined that if there exists an edge between vertex i and vertex j then $(i, j) = 1, 0$ otherwise. Let $S = \{i \mid i \in V^+ \text{ and } (i, a) = 1 \text{ in } G\}$ and $Q = \{i \mid i \in V^- \text{ and } (i, a) = 1 \text{ in } G\}$.

$\Rightarrow |S| \geq 1$ where $|S|$ denotes the number of elements in S .

Since $a \in V^+$, then

$$\Rightarrow \frac{d}{dt}U_{a,1}(T) > \frac{d}{dt}U_{a,2}(T)$$

$$\Rightarrow \sum_{y=0}^N W_{a,y} V_{y,1} < \sum_{y=0}^N W_{a,y} V_{y,2} \text{ where } W_{i,j} \text{ is defined in equation 5.5}$$

$$\Rightarrow W_{a,0} + \sum_{y=1}^N W_{a,y} V_{y,1} < \sum_{y=1}^N W_{a,y} V_{y,2} \text{ w (because } s_0 \in V^+)$$

$$\Rightarrow \frac{1}{4}(\deg(a) - 1) + \frac{1}{4}|S| < \frac{1}{4}|Q| \text{ where } \deg(a) \text{ denotes the degree of the vertex } a \text{ in } G.$$

$$\Rightarrow \frac{1}{4}(|Q| + |S| - 1) + \frac{1}{4}|S| < \frac{1}{4}|Q|$$

$$\Rightarrow |S| < \frac{1}{2}$$

Since $|S| \geq 1$, we reach a contradiction.

Therefore, every local minimum for the maximum cut in the graph G_M corresponds to a maximum independent set in the graph G if the maximum neural network is used for solving the maximum independent set problem. Q.E.D.

5.2 Simulation results

A program in C language was written to test the validity of the algorithm. The program listing of the main program is given as Appendix A. More than a thousand examples including up to 1000-vertex problems have been examined. In order to demonstrate the performance of the proposed parallel algorithm, the graphs in Figure 5.1 (a) and Figure 5.2 (a) are first tested. Figure 5.1 (b) and Figure 5.2 (b) show the independent set found by the proposed algorithm.

Our algorithm is then compared with Pardalos's algorithm [86] where graphs with vertex size varying from 10 to 75 and with density varying from 0.1 to 0.9 were tested on the Cray 2 supercomputer using one processor. For the sake of comparison, the published results for the maximum clique problem have been converted to the results for the maximum independent set problem as given in section 5.1. The comparison of the results

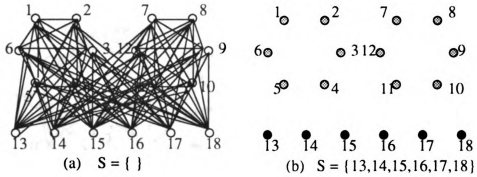


Figure 5.1 The test result of the proposed algorithm (a) the given graph G (b) the found independent set S of the graph G.

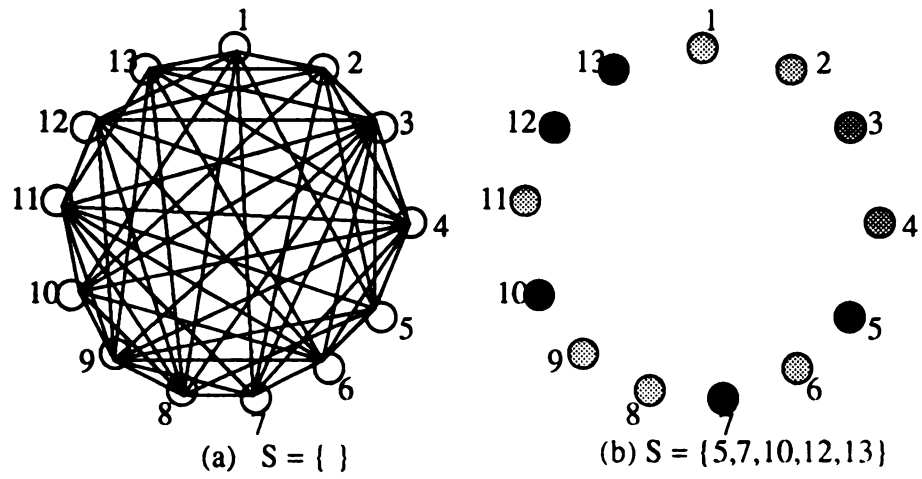


Figure 5.2 The test result of the proposed algorithm (a) the given graph G (b) the found independent set S of the graph G.

Table 5.1 Comparison of simulation results for 10-vertex graphs.

Cray 2			Sun SPARC 10		
Average density	Average set size	Average comp. time	Average density	Average set size	Average comp. time
0.88	2.1	0.3	0.91	2.0	0.021
0.75	2.9	0.5	0.76	2.7	0.028
0.50	3.7	0.6	0.51	3.8	0.022
0.22	5.5	0.4	0.26	5.4	0.026
0.10	7.0	0.2	0.11	7.5	0.025

Table 5.2 Comparison of simulation results for 25-vertex graphs.

Cray 2

Sun SPARC 10

Average density	Average set size	Average comp. time	Average density	Average set size	Average comp. time
0.91	2.6	6.7	0.91	2.7	0.18
0.75	3.4	7.3	0.76	3.5	0.19
0.49	5.5	7.3	0.51	5.4	0.14
0.24	8.7	6.7	0.26	9.2	0.19
0.09	14.2	4.7	0.11	14.5	0.17

Table 5.3 Comparison of simulation results for 50-vertex graphs.

Cray 2			Sun SPARC 10		
Average density	Average clique size	Average comp. time	Average density	Average clique size	Average comp. time
0.90	2.7	93.6	0.91	2.9	0.91
0.75	3.6	98.1	0.76	4.1	0.96
0.50	5.9	86.3	0.51	6.8	0.83
0.25	10.4	78.7	0.26	12.4	0.93
0.10	18.8	69.2	0.11	21.8	0.88

Table 5.4 Comparison of simulation results for 75-vertex graphs.

Cray 2

Sun SPARC 10

Average density	Average set size	Average comp. time	Average density	Average set size	Average comp. time
0.91	3.0	2283.0	0.91	3.1	2.29
0.75	4.0	3690.1	0.76	4.5	2.48
0.49	6.0	2837.1	0.51	7.6	1.98
0.24	11.0	2941.4	0.26	14.5	2.22
0.09	22.0	3879.6	0.11	26.7	2.45

Table 5.5 Summary of simulation results to demonstrate solution qualities.

	75% density		50% density		25% density	
Graph size	Avg. steps	Solution quality	Avg. steps	Solution quality	Avg. steps	Solution quality
100	214.7	4.9/6	180.6	8.2/10	264.7	15.3/17
200	251.0	4.7/6	177.5	8.9/10	114.8	18.5/21
300	246.9	5.6/7	224.2	9.4/12	226.3	19.6/22
400	287.0	5.3/7	209.0	11.0/13	292.2	20.6/22
500	316.6	5.3/7	376.8	9.2/12	276.3	21.2/26
600	259.7	5.7/7	364.6	10.2/11	260.9	22.2/24
700	282.8	4.7/5	353.2	11.7/13	347.2	21.8/24
800	388.8	5.3/6	355.0	10.2/13	283.5	24.7/26
900	392.2	5.8/7	449.0	10.6/12	388.3	24.2/27
1000	395.2	6.0/8	398.2	10.8/13	452.8	22.8/25

Table 5.6 Comparison of the computation time for 50%-density graph problems.

Graph size	IBM 3090	Sun SPARC 10
100	0.14	4.5
200	4.16	17.0
300	46.04	47.8
400	235.68	79.5
500	1114.78	227.4
600	---	362.0
700	---	407.5
800	---	534.2
900	---	817.6
1000	---	953.4

are shown in Tables 5.1 - 5.4. Each element in the tables represents by the average value of 100 simulation runs of randomly generated graphs with given sizes and densities. The maximum neural network converges within 200 iteration steps. Our computation times on the Sun SPARC 10 are shorter than theirs on the Cray 2 and our solution qualities are comparable to theirs.

Our algorithm is also compared with Carraghan's algorithm [87] on IBM 3090 mainframe where graphs with vertex size varying from 100 to 1000 and with density varying from 0.25 to 0.75 were tested. Table 5.5 shows the average number of iteration steps to converge to solution and the average/maximum sizes of independent sets found by our algorithm. The same conversion between the maximum independent set problem and the maximum clique problem as given in section 5.1 has been used. Each element in Table 5.5 represents by the average value of 10 simulation runs of randomly generated graphs with given sizes and densities. Table 5.6 compares the computation time between our algorithm on the Sun SPARC 10 and Carraghan's algorithm on the IBM 3090.

Tables 5.1 - 5.6 show that our algorithm is superior to the best existing algorithms in term of computation time while still maintains similar solution quality.

5.3 Ultrasonic tissue characterization

Experimental data sets obtained from a human brain sample with hemorrhaged tumor, as shown in Figure 5.3, are used in this section. The data set are taken from the data acquisition system in our Ultrasound Research Laboratory. A program written in C language is developed to do the data acquisition and displaying. The program listing of the main program is given in Appendix B. The experimental setup includes a PC-486, a PC-based A/D converter board, a Panametrics 5050 pulser, and a Panametrics V306 transducer, as shown in Figure 5.4. The pulser is used to trigger the transducer and receive



Figure 5.3 Picture of human brain sample with hemorrhaged tumor.

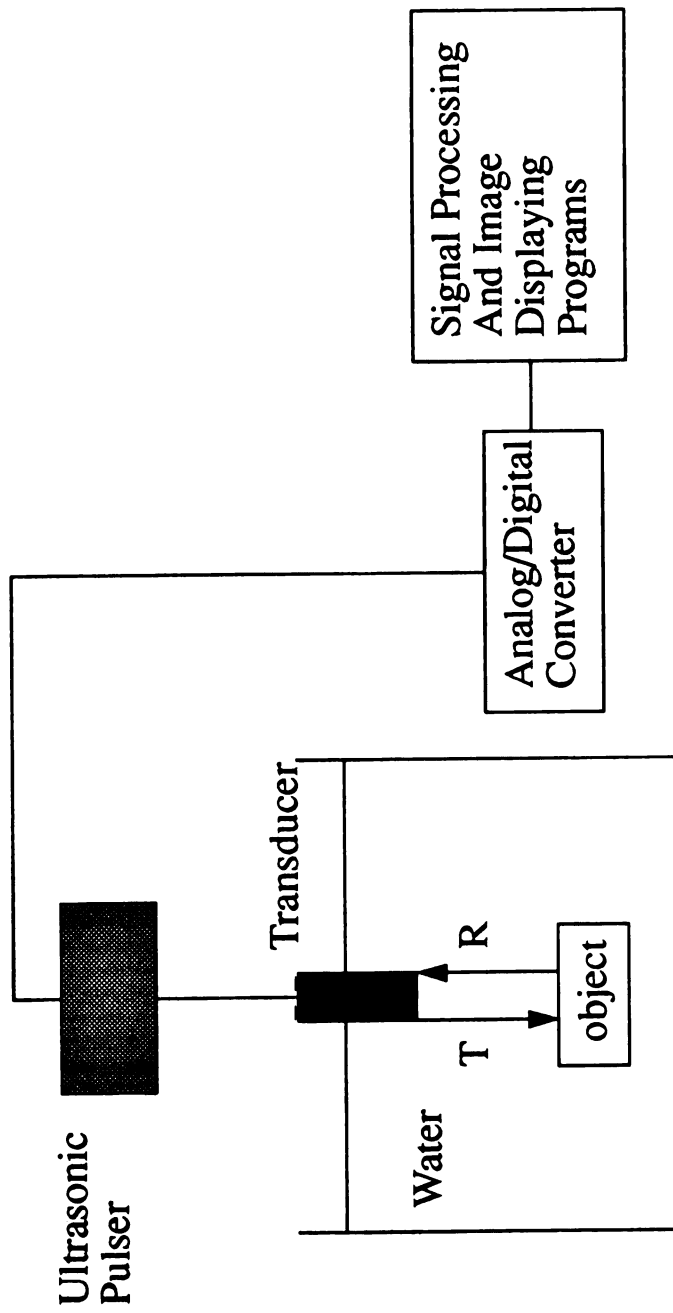


Figure 5.4 Schematic diagram of the data acquisition system.

reflected signals. The received signal is sampled at 40 MHz with 8-bit resolution by the A/D converter. The sampled data is stored in the computer for further analysis.

Five features, as mentioned in section 4.3, are extracted from returned echoes. The use of the multidimensional data is to catch as much as the information of the target tissue as possible. However, some of those features may contain similar information. In order to reduce the redundancy in the data set, the dependence between features is measured by equation 4.6. The dependence between features is shown by the eigenvector projection of the five features onto a two-dimensional space in Figure 5.5. Total energy and peak frequency are discarded due to their strong dependence to correlation coefficient and central frequency respectively.

Since the number of data points in the data set is large, it will take a long processing time and large memory capacity. The background data points are removed to reduce the size of the data set. The background data points can easily be removed because it is very uniform. The reduced three-feature data set is used to generate the conductance matrix $W_{x,y}$ for the proposed algorithm. The $W_{x,y}$ is given as

$$W_{x,y} = d(x, y) \quad (5.7)$$

where $d(.)$ is the Euclidean distance between data points x and y . The W matrix characterizes the maximum neural network. Spatial information is then added to the clustering result for subsequent image reconstruction. The C-scan image is shown in Figure 5.6 while the reconstructed image is shown in Figure 5.7. From Figure 5.7, the clustering result does show the abnormal tissue portion. However, a detail identification of the brain sample requires further investigation for conclusive result. Physicians could contribute their expertise to the eventual identification.

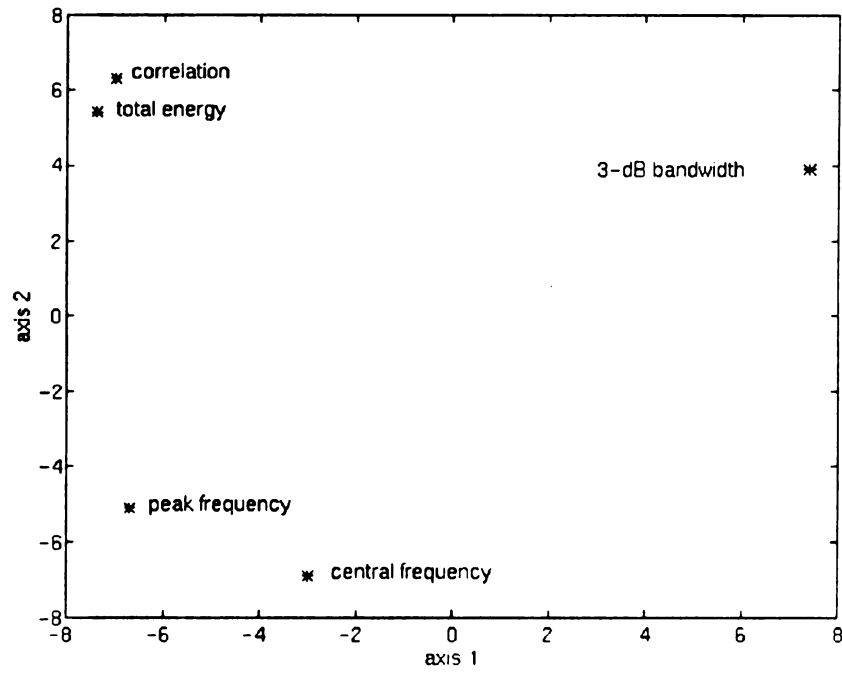


Figure 5.5 Projection of five features in a two-dimensional space.



Figure 5.6 C-scan image of human brain sample.



Figure 5.7 Reconstructed image of human brain sample.

CHAPTER 6

CONCLUSIONS

6.1 Summary

In this dissertation, some basic principles of linear acoustic waves have been reviewed. Time domain and frequency domain techniques for acoustic parameter estimation were described. Their advantages and limitations were also discussed.

Ultrasound is a useful tool in many applications, such as non-destructive evaluation of materials and tissue characterization. The conventional ultrasonic detection technique uses the A-mode signal directly for material characterization. The proposed approach is first to extract features from the return echoes. Then, use these features to construct the feature vectors from the data set. A modified Hopfield neural network is adopted for clustering the data set. We refer to this modified Hopfield neural network as the maximum neural network. It has several advantages: (1) The maximum neural network always generates a valid solution. (2) No trial-and error parameters are needed in the maximum neural network. (3) The maximum neural network has exact condition for convergence.

The maximum independent set problem and the tissue characterization problem have been used to test the proposed system. For the maximum independent set problem, shorter computation time has been achieved from the proposed algorithm. For the tissue characterization problem, the abnormal portion of the sample has been identified. Due to the inherent parallelism in the maximum neural network, parallel processing for the material characterization is also achieved.

6.2 Directions for future work

Modern acoustic imaging is still in its infancy, combining recent achievements in data acquisition and computer science techniques. Much improvement can be made in the areas of non-destructive evaluation and clinical applications. Some of the topics to be pursued in the future are as follows:

- (1) Pre-processing knowledge: Currently no preference has been given to any feature in the feature vector. However, some feature may be more relevant than others. Those features should have greater influence to the outcome of the characterization.
- (2) Initial state: Randomly generated initial state in the maximum neural network may not be appropriate for specific problems. Some pre-processing knowledge can be used to adjust the initial state. This adjustment can further shorten the processing time.
- (3) Post-processing knowledge: A knowledge base of acoustic response of various biological tissues should be established and incorporated into the proposed

system. Therefore, conclusive results could be obtained.

- (4) **Real-time system:** Most of the components in the proposed system are built in software and run in sequential computers. Hardware implementation of the processing elements or the use of parallel computers is needed to take full advantage of the proposed system. Since the basic processing elements in the maximum neural network are very simple, hardware implementation should be feasible and may be more economical than the use of parallel computers.

APPENDIX

APPENDIX A

PROGRAM LIST FOR THE MAXIMUM INDEPENDENT SET PROBLEM

```
/* **** */
/*
/* Maximum Neural Network approach for solving
/* Maximum Independent Set problem
/*
/* By C.P. Chou
/*
/* **** */

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

#define MaxN 2000
#define M (MaxN+1)
/* random generator for initial U */
#define random() ((rand()%(1000))/10.0)-50.0
/* random generator for adjacency matrix */
#define random1() ((rand()%(100))/50.0)-1.0

main(){
FILE *fp;
char Infile[40],Outfile[40];
int i,j,x,y,N,flag,iteration,min,count;
float q[M][M],W[M][M],V[M][2],U[M][2],deltaU[M][2],deltaT;
float exist[M],deg[M],set[M],min_deg;

/* man-machine interface */

N=0;
while ((N != 1) && (N != 2)){
printf("(1) Use user given adjacency matrix.\n");
printf("(2) Use randomly generated adjacency matrix.\n");
```

```

printf("Your choice: ");
scanf("%d",&N);
}

/* (1) Use user given adjacency matrix. */

if (N == 1) {
printf("Input adjacency matrix file name:\n");
scanf("%s",Infile);
fp=fopen(Infile,"r");
fscanf(fp,"%d",&N);
if (N > MaxN){
printf("Vertex number is too large.\n");
exit(1);
}
for (i=1;i<=N;i++){
for (j=1;j<=N;j++){
fscanf(fp,"%d",&x);
q[i][j]=(float)x;
}
}
fclose(fp);
}

/* (2) Use randomly generated adjacency matrix. */

else {
printf("Input vertex number = ");
scanf("%d",&N);
if (N > MaxN){
printf("Vertex number is too large.\n");
exit(1);
}

/* generat the adjacency matrix */
/* This graph has N vertices. */
/* q[i][j], 1<= i,j <= N */
/* q[][] is a symmetric matrix with diagonal zeros. */

for (i=2;i<=N;i++)
for (j=1;j<=(i-1);j++){
if (random1() <= 0.0)
q[i][j]=0.0;
else
q[i][j]=1.0;
q[j][i]=q[i][j];
}
for (i=1;i<=N;i++)
q[i][i]=0.0;
/*
printf("Generated adjacency matrix:\n");
for (i=1;i<=N;i++){
for (j=1;j<=N;j++){
printf("%d ",(int)q[i][j]);
}
printf("\n");
}

```

```

    }
*/
}
/*
printf("Output file name:\n");
scanf("%s",Outfile);
*/

/* calculate conductance matrix W[][] from q[][] */

for (i=1;i<=N;i++){
    W[0][i]=0.0;
    for (j=1;j<=N;j++){
        W[0][i]=W[0][i]+q[i][j];
        W[0][i]=W[0][i]-1.0;
        W[0][i]=W[0][i]/4.0;
        W[i][0]=W[0][i];
    }
}
for (i=1;i<=N;i++){
    for (j=1;j<=N;j++){
        W[i][j]=q[i][j]/4.0;
        W[j][i]=W[i][j];
    }
}
W[0][0]=0.0;

/* The main algorithm starts here */

/* generate initial U[][] (step 1) */

for (x=0;x<=N;x++)
    for (i=0;i<=1;i++)
        U[x][i]=random();

flag=0;
iteration=0;
printf("iteration = %d\n",iteration);
while(flag==0){

/* calculate the V[][] (output) (step 2)*/

    for (x=0;x<=N;x++)
        if (U[x][0] > U[x][1]){
            V[x][0]=1.0;
            V[x][1]=0.0;
        }
        else{
            V[x][0]=0.0;
            V[x][1]=1.0;
        }
}

/*
printf("V[x][i]\n");
for (i=0;i<=1;i++){
    for (x=0;x<=N;x++)
        printf("%d ",(int)V[x][i]);
}
*/

```



```

        printf("\n");
    }
*/

/* calculate the delta U[][] (step 3) */

    for (x=0;x<=N;x++)
        for (i=0;i<=1;i++){
            deltaU[x][i]=0.0;
            for (y=0;y<=N;y++)
                deltaU[x][i]=deltaU[x][i]+W[x][y]*V[y][i];
            deltaU[x][i]=(-1.0)*deltaU[x][i];
        }
/*
    printf("deltaU[x][i]\n");
    for (i=0;i<=1;i++){
        for (x=0;x<=N;x++)
            printf("%5.2f ",deltaU[x][i]);
        printf("\n");
    }
*/
deltaT=1.0;

/* update U[][] (input) (step 4) */

    for (x=0;x<=N;x++)
        for (i=0;i<=1;i++)
            U[x][i]=U[x][i]+deltaU[x][i]*deltaT;

    for (x=0;x<=N;x++){
        if ((U[x][0]-U[x][1]) > 100.0){
            U[x][0]=50.0;
            U[x][1]=-50.0;
        }
        if ((U[x][1]-U[x][0]) > 100.0){
            U[x][1]=50.0;
            U[x][0]=-50.0;
        }
    }
/*
    printf("U[x][i]\n");
    for (i=0;i<=1;i++){
        for (x=0;x<=N;x++)
            printf("%5.2f ",U[x][i]);
        printf("\n");
    }
*/

/* Is the sysyem in equilibrium state? (step 5) */
/* If any neuron is unstable, set flag = 0. */

    flag=1;
    for (x=1;x<=N;x++){
        if (deltaU[x][0] > deltaU[x][1])

```

```

        i=0;
    else
        i=1;
    if (V[x][i] == 0)
        flag=0;
    }
    iteration=iteration+1;
    printf("iteration = %d\n",iteration);
} /* end of while() */

/* End of the algorithm */

/* Output the result */

for (x=0;x<=N;x++)
    if (U[x][0] > U[x][1]){
        V[x][0]=1.0;
        V[x][1]=0.0;
    }
    else{
        V[x][0]=0.0;
        V[x][1]=1.0;
    }
/*
    printf("V[x][i]\n");
    for (i=0;i<=1;i++){
        for (x=0;x<=N;x++){
            printf("%d ",(int)V[x][i]);
            printf("\n");
        }
    }
*/
/*
    fp=fopen(Outfile,"w");
*/
/* Vertices in the same group as vertex 0 belong to Independent Set. */
if (V[0][0] == 1.0)
    j=0;
else
    j=1;
/*
    fprintf(fp,"V+(Max Independent Set):\n");
*/
printf("V+(Max Independent Set):\n");
i=0;
count=0;
for (x=1;x<=N;x++){
    if (V[x][j] == 1.0){
/*
        fprintf(fp,"%d ",x);
*/
        printf("%d ",x);
        i++;
        count++;
    }
}

```

```
    if (i >= 15){
/*      fprintf(fp, "\n");
*/
        printf("\n");
        i=0;
    }
}
/*
fprintf(fp, "\nsize = %d", count);
*/
printf("\nsize = %d", count);
if (j==0)
    j=1;
else
    j=0;
/*
fprintf(fp, "\nV-:\n");
*/
printf("\nV-:\n");
i=0;
for (x=1;x<=N;x++){
    if (V[x][j] == 1.0){
/*      fprintf(fp, "%d ", x);
*/
        printf("%d ", x);
        i++;
    }
    if (i >= 15){
/*      fprintf(fp, "\n");
*/
        printf("\n");
        i=0;
    }
}
/*
fprintf(fp, "\n");
*/
printf("\n");
/*
fclose(fp);
*/
}/* end of the main program */
```

APPENDIX B

PROGRAM LIST FOR DATA ACQUISITION AND DISPLAYING

```

/*****
/*
/* Data acquisition and displaying.
/*
/* By C.P. Chou
/*
*****/

#include <alloc.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <time.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <math.h>

#define MAXSIZE 131072L /* 128K memory */
#define SEGMENT 0xd000 /* fixed segment */
#define PORT0 0x178 /* default setting */
#define PORT1 0x179 /* all switches are off */
#define PORT2 0x17A
#define PORT3 0x17B
#define PULSER 0x301 /* port for triggering pulser */
#define CMD0 "C:change setting S:static D:dynamic ESC:exit"
#define ESC 27
#define YES 1
#define NO 0

unsigned int huge *sig; /* sampling signal */
unsigned int huge *sigold; /* old sampling signal */

```

```

int israte;                /* index of sampling rate */
long startpoint,endpoint;
float sx,sy;              /* scales of x and y */
int lx,ly;                /* size of window 1 */
int wlt,wlb,wlr,wll;     /* parameters of window 1 */
int w2t,w2b,w2r,w2l;     /* parameters of window 2 */
long secs_now;           /* current time in seconds */
char *str_now;          /* currnet time in string */
long cursor,oldcur;     /* cursor position */
long curscale;         /* cursor inc (dec) step size */
int drawn,curdrawn;

void main()
{
int done,goodans;
int cmdflag,initflag,plotflag,curflag;
char ans,ans1;

if ((sig = farmalloc(sizeof(unsigned int)*MAXSIZE)) == NULL) {
printf("Memory allocation failed!\n");
exit(1);
}
if ((sigold = farmalloc(sizeof(unsigned int)*MAXSIZE)) == NULL) {
printf("Memory allocation failed!\n");
exit(1);
}

outportb(PULSER + 2, 0x88);
outportb(PULSER, 0x00);

done = NO;
initflag = YES;
plotflag = YES;
curflag = NO;

do {
if (initflag == YES){
getndata();
init_graphics();
drawn = NO;
curdrawn = NO;
initflag = NO;
cmdflag = YES;
}
if (cmdflag == YES){
dispcmd();
cmdflag = NO;
}
if (plotflag == YES){
sampling_all(50);
plotsignal();
cmdflag = YES;
}
if (curflag == YES){

```

```

plotcursor();
curflag = NO;
cmdflag = YES;
}
do { /* do loop for command */
goodans = YES;
if (kbhit() !=0){
ans = getch();
ans = toupper(ans);
switch(ans)
{
case '\0': /* arrows */
ans1 = getch();
if (plotflag == NO){
switch(ans1)
{
case 'M':
case 'K':
oldcur = cursor;
if (cursor > curscale && ans1 == 'K') /* left
arrow */
cursor-=curscale;
if (cursor < (endpoint-startpoint-curscale) &&
ans1 == 'M')
/* right arrow */
cursor+=curscale;
plotcursor();
cmdflag = YES;
break;

default:
goodans = NO;
break;
}
} else {
goodans = NO;
}
break;

case 'C': /* change setting */
closegraph();
initflag = YES;
plotflag = YES;
curflag = NO;
break;

case 'S': /* static */
oldcur = cursor;
plotflag = NO;
curflag = YES;
break;

case 'D': /* dynamic */
plotflag = YES;

```

```

        if (curdrawn == YES){
            curdrawn = NO;
            plotcursor();
            curdrawn = NO;
        }
    break;

    case ESC:
        done = YES;
        break;

    default: /* any illegal command */
        goodans = NO;
        break;
    } /* do loop for command */
}
} while(goodans == NO);
} while(done == NO);
closegraph();
farfree(sig);
farfree(sigold);
} /* main */

getndata()
{
    long sampleno;

    clrscr();
    do {
        printf("\nChoose the sampling rate:\n");
        printf("\n          8: 40MHz");
        printf("\n          9:  4MHz");
        printf("\n         10: 400KHz");
        printf("\n         11:  40KHz");
        printf("\n         12:  4KHz");
        printf("\n\n          Sampling rate = ");
        scanf("%d",&israte);
    } while(israte > 12 || israte < 8);
    do {
        printf("\nStarting point of sampling (inside [0, %ld]) = ",MAXSIZE);
        scanf("%ld",&startpoint);
    } while (startpoint < 0 || startpoint > (MAXSIZE-1));
    do {
        printf("\nNumber of sampling points (inside [0, %ld]) = ",(MAXSIZE-
startpoint));
        scanf("%ld",&sampleno);
    } while (sampleno < 0 || sampleno > (MAXSIZE-startpoint));
    endpoint = startpoint + sampleno - 1;
    cursor = sampleno/2;
    curscale = cursor/10;
} /* get data */

init_graphics()
{

```

```

int graphdriver;          /* graphics card, eg. Hercules..*/
int graphmode;           /* graphics mode                */
int maxx,maxy;         /* max x and y in graphics     */
float dx,dy;            /* differences of x and y      */

detectgraph(&graphdriver,&graphmode);
initgraph(&graphdriver,&graphmode,"\\tc\\");

maxx = getmaxx();
maxy = getmaxy();
lx = (maxx - 40);        /* parameters for window 1     */
ly = (maxy - 40);
wll = 30;
wlt = 10;
wlr = wll + lx;
wlb = wlt + ly;
w2l = 0;                 /* parameters for window 2     */
w2t = wlb + 2;
w2r = maxx;
w2b = maxy;
dx = endpoint - startpoint; /* parameters for graphics     */
sx = lx/dx;
dy = 256.0;
sy = ly/dy;

setcolor(GREEN);
rectangle(wll-2,wlt-1,wlr+1,wlb+1);
} /* init graphics */

plotsignal()             /* display the signal */
{
int px1,py1,px2,py2,pyold1,pyold2;
long j;

setviewport(wll,wlt,wlr,wlb,1);
px1 = 0;
py1 = ly - sig[startpoint]*sy;
pyold1 = ly - sigold[startpoint]*sy;
sigold[startpoint] = sig[startpoint];
for (j=startpoint+1;j<=endpoint;j++){
    px2 = (j-startpoint)*sx;
    py2 = ly-sig[j]*sy;
    if (drawn == YES){
        pyold2 = ly - sigold[j]*sy;
        setcolor(BLACK);
        line(px1,pyold1,px2,pyold2);
    }
    setcolor(LIGHTCYAN);
    line(px1,py1,px2,py2);
    px1 = px2;
    pyold1 = pyold2;
    py1 = py2;
    sigold[j] = sig[j];
}
}

```



```

drawn=YES;
} /* plot signal */

sampling_all(troffset)
unsigned int troffset;
{
  unsigned int trigger,sample_all();
  unsigned int i,j,control;
  long k;
  /*
                                     Low byte : (Port2)          7654 3210
                                     bit 6,5,4 : sampling rates : xxx

                                     000 = 20MHz (israte= 0) or 40 MHz (israte= 8)
                                     001 =  2MHz (israte= 1) or  4 MHz (israte= 9)
                                     010 = .2MHz (israte= 2) or .4 MHz (israte=10)
                                     011 = 20KHz (israte= 3) or 40 KHz (israte=11)
                                     100 =  2KHz (israte= 4) or  4 KHz (israte=12)
                                     101 = external clock
                                     110 = no clock
  */
  control = 0xa00c | ((israte % 8) <<4 );
  load_trigger_offset(troffset); /* trigger offset */

  trigger = sample_all(control); /* data acquisition */
  trigger <<= 1; /* address = trigger*2 */

  i=0; /* access low 64K */
  output(PORT2,0x806f); /* enable WAGII RAM */
  for (k=0;k<(MAXSIZE/2);k++){
    j = trigger+i;
    sig[k] = (unsigned int)peekb(SEGMENT,j);
    i++;
  }
  i=0; /* access high 64K */
  output(PORT2,0x906f); /* enable WAGII RAM */
  for (k=(MAXSIZE/2);k<MAXSIZE;k++){
    j = trigger+i;
    sig[k] = (unsigned int)peekb(SEGMENT,j);
    i++;
  }
} /* end of sampling_all() */

unsigned int sample_all(c)
unsigned int c;
{
  int count,busy;

  output(PORT2,0x806f); /* initialize */
  output(PORT0,0xffff); /* clear counter */
  output(PORT0,0xffff);
  count = 0; /* count for 128K samples */
  output(PORT0,count); /* load count */
  output(PORT0,count);

```

```

outport(PORT2,c & 0xbffc); /* start counter */

outportb(PULSER, 0x01);
outportb(PULSER, 0x00);

busy = 1;
while (busy){ /* wait for sampling to be completed */
    if(inportb(PORT2) & 01) busy = 0; /* sampling is done */
}
return(inport(PORT0)); /* read trigger address */
} /* end of sample_all() */

load_trigger_offset(char x)
{
outport(PORT2,0xc06f); /* enable trigger offset */
pokeb(SEGMENT,00,x); /* load offset value */
outport(PORT2,0x806f);
}

plotcursor() /* display the cursor line */
{
int px,j;
int gcolor,pcolor; /* original and new colors */

setviewport(w1l,w1t,w1r,w1b,1);
if (curdrawn == YES){
    px = oldcur*sx;
    for (j=1;j<ly;j++){
        gcolor = getpixel(px,j);
        pcolor = abs(gcolor-15);
        putpixel(px,j,pcolor);
    }
}
px = cursor*sx;
for (j=1;j<ly;j++){
    gcolor = getpixel(px,j);
    pcolor = abs(gcolor-15);
    putpixel(px,j,pcolor);
}
curdrawn = YES;
}

dispcmd()
{
int x,y;
double samplerate;

setcolor(LIGHTCYAN);
setviewport(w2l,w2t,w2r,w2b,1);
clearviewport();

time(&secs_now);
str_now=ctime(&secs_now);
str_now[24]='\0'; /* delete '\n' in str_now */
}

```

```

x = 28;
y = 4;
samplerate=((int) (israte/8)+1)*2.0*pow10(1-(israte % 8));
gprintf(&x,&y,"%6.3f MHZ",samplerate);
x = 120;
y = 4;
gprintf(&x,&y,"Display [%ld, %ld]",startpoint,endpoint);
x = 320;
y = 4;
gprintf(&x,&y,"Cursor=%ld",cursor+startpoint);
x = 435;
y = 4;
gprintf(&x,&y,"%s",str_now);
x = 120;
gprintf(&x,&y,"%s",CMD0);
} /* display command */

```

```

/*      GPRINTF: Used like PRINTF except the output is sent to
the      */
/*      screen in graphics mode at the specified co-
ordinate.      */

```

```

int gprintf( int *xloc, int *yloc, char *fmt, ... )
{
    va_list  argptr;          /* Argument list
pointer      */
    char str[140];          /* Buffer to build sting
into      */
    int cnt;                /* Result of SPRINTF for
return */

    va_start( argptr, format );          /* Initialize va_
functions      */
    cnt = vsprintf( str, fmt, argptr ); /* prints string to
buffer      */
    outtextxy( *xloc, *yloc, str );      /* Send string in graphics
mode */
    *yloc += textheight( "H" ) + 2;      /* Advance to next
line      */
    va_end( argptr );          /* Close va_
functions      */
    return( cnt );            /* Return the conversion
count */
}

```

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] T. E. Preuss and G. Clark, "Use of time-of-flight c-scanning for assessment of impact damage in composites," *Composites*, vol. 19, pp. 145-148, March 1988.
- [2] R. Kuc, M. Schwartz, and L. V. Minsky, "Parametric estimation of acoustic attenuation coefficient slope for soft tissue," *IEEE Ultrasonics Symposium Proceedings*, pp. 44-47, 1976.
- [3] D. Ensminger, "Ultrasonics: Fundamentals, Technology, and Applications," Marcel Dekker, New York, 1988.
- [4] N. H. Wang, B. Ho, and H. R. Zapp, "Attenuation and velocity imagings of biological tissues by broadband ultrasonic signals," *Fifteenth International Symposium on Ultrasonic Imaging and Tissue Characterization*, Arlington, Virginia, June 1990.
- [5] E. E. Hundt and E. A. Atrautenberg, "Digital processing of ultrasonic data by deconvolution," *IEEE Transactions on Sonics and Ultrasonics*, vol. SU-27, no. 5, pp. 249-252, Sep. 1980.
- [6] T. Yokota and Y. Sato, "Super-resolution ultrasonic imaging by using adaptive focusing," *Journal of Acoustical Society of America*, vol. 77(2), pp. 567-572, Feb. 1985.
- [7] I. Beretsky and G. A. Farrel, "Improvement of ultrasonic imaging and media characterization by frequency domain deconvolution: experimental study with non-biological models," *Ultrasound in Medicine*, vol. 38, pp. 1645-1665, 1977.
- [8] J.P. Steiner, E. S. Furgason, and W. L. Weeks, "Robust deconvolution of correlation functions," *IEEE Ultrasonics Symposium Proceedings*, pp. 1031-1035, 1987.
- [9] C. Chou and B. Ho, "Tissue characterization using adaptive fuzzy clustering," *Nineteenth International Symposium on Ultrasonic Imaging and Tissue Characterization*, Arlington, Virginia, June 1994.

- [10] T. Pialucha, C. C. Guyott, and P. Cawlew, "Amplitude spectrum method for the measurement of phase velocity," *Ultrasonics*, vol. 27, pp. 270-279, Sep. 1989.
- [11] J. P. Jones, "Ultrasonic impediography and its application to tissue characterization," *Recent Advance in Ultrasound in Biomedicine*, pp. 131-154, 1979.
- [12] P. Cobo-Parra and C. Ranz-Guerra, "Impedance profile and overall attenuation of layered sea bottom from their normal incident acoustic reflection response," *Journal of Acoustical Society of America*, vol. 85(6), pp. 2388-2393, June 1989.
- [13] G. Kossoff, E. K. Fry, and J. Jellins, "Average velocity of ultrasound in the human female breast," *Journal of Acoustical Society of America*, vol. 53, pp. 1730-1736, 1973.
- [14] T. Lin, J. Ophir, and G. D. Potter, "Correlation of sound speed with tissue constituents in normal and diffuse liver disease," *Ultrasonic Imaging*, vol. 9, pp. 29-40, 1987.
- [15] J. F. Greenleaf, S. A. Johnson, R. Bahn, and B. Rajagopalan, "Quantitative cross-sectional imaging of ultrasound parameters," *IEEE Ultrasonics Symposium*, pp. 989-995, 1977.
- [16] J. F. Greenleaf, J. Ylitalo, and J. J. Gisvold, "Ultrasonic computed tomography for breast examination," *IEEE Engineering in Medicine and Biology Magazine*, pp. 27-32, Dec. 1987.
- [17] J. Ophir and T. Lin, "A calibration-free method for measurement of sound speed in biological tissue samples," *IEEE Transaction on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. UFFC-35, no. 5, pp. 573-577, Sep. 1988.
- [18] J. Ophir and Y. Yazdi, "A transaxial compression technique for localized pulse echo estimation of sound speed in biological tissue," *Ultrasonic Imaging*, vol. 12, pp. 35-46, 1990.
- [19] N. Hayashi, "A new method measuring in vivo sound speed in reflection mode," *Journal of Clinical Ultrasound*, vol. 16, pp. 87-93, 1988.
- [20] T. Kontonassios and J. Ophir, "Variance reduction of speed of sound estimation in tissue using the beam tracking method," *IEEE Transaction on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. UFFC-34, no. 5, pp. 524-530, Sep. 1987.

- [21] D. E. Robinson, F. Chen, and L. S. Wilson, "Measurement of velocity propagation from ultrasonic pulse echo data," *Ultrasound in Medicine and Biology*, vol. 8, pp. 413-420, 1982.
- [22] J. W. Mimbs, M. O'Donnell, J. G. Miller, and B. E. Sobel, "Changes in ultrasonic attenuation indicative of early myocardium ischemic injury," *American Journal of Physiology*, vol. 236, pp. 340-344, 1979.
- [23] P. L. Carson, C. R. Mair, A. L. Scharzinger, and T. U. Oughton, "Breast imaging in coronal planes with simulation pulse echo and transmission ultrasound," *Science*, 214, pp. 1141-1143, 1981.
- [24] K. A. Dines and A. C. Kak, "Ultrasonic attenuation tomography of soft tissue," *Ultrasonic Imaging*, vol. 1, no. 1, pp. 16-33, 1979.
- [25] J. F. Greenleaf and R. Bahn, "Clinical imaging with transmissive ultrasonic computerized tomography," *IEEE Transaction on Biomedical Engineering*, vol. BME-28, no. 2, pp. 177-185, Feb. 1981.
- [26] R. Kuc, "Estimating acoustic attenuation from reflected ultrasound signals: comparison of spectral-shift and spectral-difference approaches," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 1, pp. 1-6, Feb. 1984.
- [27] R. Kuc, "Estimating reflected ultrasound spectra from quantized signals," *IEEE Transaction on Biomedical Engineering*, vol. BME-32, no. 2, pp. 105-112, Feb. 1985.
- [28] R. Kuc, "Bounds on estimating the acoustic attenuation of small tissue regions from reflected ultrasound," *IEEE Proceedings*, 73, no. 7, pp. 1159-1168, July 1985.
- [29] I. Claesson and G. Salomonsson, "Estimation of varying ultrasonic attenuation," *Ultrasound in Medicine and Biology*, vol. 11, pp. 131-145, 1985.
- [30] J. Ophir, R. E. McWrite, N. F. Maklad, and P. M. Jaeger, "A narrowband pulse-echo technique for in vivo ultrasound attenuation estimation," *IEEE Transaction on Biomedical Engineering*, vol. BME-32, no. 3, pp. 205-212, March 1985.
- [31] R. Momenan, "Application of pattern recognition techniques in ultrasound tissue characterization," *IEEE Engineering in Medicine and Biology Society Eleventh*

Annual International Conference, pp. 411-412, June, 1989.

- [32] J. Saniie and D. T. Nagle, "Pattern recognition in the ultrasonic imaging of reverberant multilayered structures," IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. UFFC-36, no. 1, pp. 80-92, Jan. 1989.
- [33] E. Walach, A. Shmulewitz, Y. Itzchak, and Z. Heyman, "Local tissue attenuation images based on pulse-echo ultrasound scans," IEEE Transaction on Biomedical Engineering, vol. BME-36, no. 2, pp. 211-221, Feb. 1989.
- [34] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.
- [35] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," Psychological Review, vol. 65, pp. 386-408, 1958.
- [36] B. Widrow and M. E. Hoff, "Adaptive switching circuits," IRE Convention Record, pp. 96-104, 1960.
- [37] J. J. Hopfield and D. W. Tank, "Neural computation decisions in optimization problems," Biological Cybernetics, vol. 52, pp. 141-152, 1985.
- [38] L. O. Chua and G. N. Lin, "Nonlinear programming without computation," IEEE Transaction on Circuits and Systems, vol. 31, pp. 182-188, Feb. 1984.
- [39] M. P. Kennedy and L. O. Chua, "Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin," IEEE Transaction on Circuits and Systems, vol. 34, pp. 210-214, Feb. 1987.
- [40] C. Chou, B. Ho, and A. K. Jain "Tissue characterization using back-propagation neural network," Twentyth International Symposium on Ultrasonic Imaging and Tissue Characterization, Arlington, Virginia, June 1995.
- [41] F. Rosenblatt, "Principles of Neurodynamics," Spartan, New York, 1962.
- [42] D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing, Vol. I: Foundations," MIT press, Cambridge, MA, 1986.
- [43] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," Parallel Distributed Processing, vol. 1, chap. 7, MIT press, Cambridge, MA, 1986.

- [44] K. C. Lee, N. Funabiki, and Y. Takefuji, "A parallel improvement algorithm for the bipartite subgraph problem," *IEEE Transaction on Neural Networks*, vol. 3, no. 1, pp. 139-145, Jan. 1992.
- [45] A. Guez and Z. Ahmad, "Solution to the inverse kinematics problem in robotics by neural networks," *Proceedings of International Joint Conference on Neural Networks*, vol. II, pp. 617-624, 1988.
- [46] M. Kuperstein and J. Wang, "Neural controller for adaptive movements with unforeseen payloads," *IEEE Transaction on Neural Networks*, vol. 1, no. 1, March 1990.
- [47] T. Kohonen, "Self-Organization and Associative Memory," Springer-Verlag, Berlin, 1988.
- [48] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Washington DC, pp. 448-453, 1983.
- [49] W. J. Fry and F. Dunn, "Ultrasound: Analysis and Experimental method in biological research," *Physical Techniques in Biological Research*, vol. 4, Academic press, New York, pp. 261-314, 1962.
- [50] J. Hertz, A. Krogh, and R. G. Palmer, "Introduction to the Theory of Neural Computation," Addison-Wesley, Redwood City, CA, 1991.
- [51] H. P. Graf, W. Hubbard, L. D. Jackel, and P. DeVegvar, "A CMOS associative memory with several hundreds of neurons," *American Institute of Physics, Snowbird, Utah*, pp. 182-187, 1986.
- [52] Y. Suzuki and L. Atlas, "A study of regular architecture for digital implementation of neural networks," *Proceedings of International Symposium on Circuits and Systems*, Portland, May 1989.
- [53] M. S. Tomlinson Jr., D. J. Walker, and M. A. Sivilotti, "A digital neural network architecture for VLSI," *Proceedings of International Joint Conference on Neural Networks*, vol. II, pp. 545-556, 1990.
- [54] J. J. Hopfield, "Neurons with graded responses have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of*

Sciences, USA, 81, pp. 3088-3092, 1984.

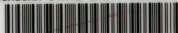
- [55] G. Hayward and J. E. Lewis, "Comparison of some non-adaptive deconvolution techniques for resolution enhancement of ultrasonic data," *Ultrasonics*, vol. 27, pp. 155-164, May 1989.
- [56] W. Kohl and R. Schwarz, "A simple method of realizing the deconvolution of ultrasonic images," *Ultrasonics*, pp. 273-278, Nov. 1981.
- [57] W. Vollmann, "Resolution enhancement of ultrasonic B-scan images by deconvolution," *IEEE Transaction on Sonics and Ultrasonics*, vol. SU-29, no. 2, pp. 78-83, March 1982.
- [58] A. C. Kak and K. A. Dines, "Signal processing of broadband pulsed ultrasound: Measurement of attenuation of soft biological tissue," *IEEE Transaction on Biomedical Engineering*, vol. BME-25, no. 4, pp. 321-343, July 1978.
- [59] J. G. Proakis and D. G. Manolakis, "Introduction to Digital Signal Processing," Mcmillan, New York, 1988.
- [60] L. S. Wilson, D. E. Robinson, and B. D. Doust, "Frequency domain processing for ultrasonic attenuation measurement in liver," *Ultrasonic Imaging*, vol. 6, pp. 278-292, 1984.
- [61] R. Kuc, "Clinical application of an ultrasound attenuation coefficient estimation technique for liver pathology characterization," *IEEE Transaction on Biomedical Engineering*, vol. BME-27, no. 6, pp. 312-319, June 1980.
- [62] P. A. Narayana and J. Ophir, "On the frequency dependence of attenuation in normal and fatty liver," *IEEE Transaction on Sonics and Ultrasonics*, vol. SU-30, no. 6, pp. 379-383, Nov. 1983.
- [63] S. Serbarian, "Influence of attenuation upon the frequency content of a stress wave packet in graphite," *Journal of Acoustical Society of America*, vol. 42(5), pp. 1052-1159, April 1967.
- [64] M. Insana, J. Zagzebski, and E. Madsen, "Improvement in the spectral difference method for measuring ultrasonic attenuation," *Ultrasonic Imaging*, vol. 5, pp. 331-345, 1983.
- [65] R. Kuc and M. Schwartz, "Estimating the acoustic attenuation coefficient slope for

- liver from reflected ultrasound signals," IEEE Transaction on Sonics and Ultrasonics, vol. SU-26, pp. 353-362, 1979.
- [66] J. Ophir and P. Jaeger, "Spectral shift of ultrasonic propagation through media with nonlinear dispersive attenuation," Ultrasonic Imaging, vol. 4, pp. 282-289, 1982.
- [67] K. A. Dines and S. A. Goss, "Computed ultrasonic reflection tomography," IEEE Transaction on Ultrasonics, Ferroelectrics, and Frequency Control, vol. UFFC-34, no. 3, pp. 309-317, May 1987.
- [68] P. Schattner, T. K. Whitehurst, J. F. Jensen, A. S. Shah, and P. S. Green "Three-dimensional ultrasonic reflection and attenuation imaging," IEEE Transaction on Ultrasonics, Ferroelectrics, and Frequency Control, vol. UFFC-21, no. 1, pp. 102-111, Jan. 1992.
- [69] L. Landini, R. Sarnelli, and F. Squartini, "Frequency-dependent attenuation in breast tissue characterization," Ultrasound in Medicine and Biology, vol. 11, no. 4, pp. 599-603, Aug. 1985.
- [70] J. G. Miller, J. E. Perez, J. G. Mottley, E. I. Madaras, P. H. Johnston, E. D. Blodgett, L. J. Thomas, and B. E. Sobel, "Myocardial tissue characterization: An approach based on quantitative backscatter and attenuation," Proceedings of IEEE Ultrasonics, pp. 782-793, 1983.
- [71] J. Ophir, T. H. Shawker, N. F. Maklad, J. G. Miller, S. W. Flax, P. A. Narayana, and J. P. Jones, "Attenuation estimation in reflection: Progress and prospects," Ultrasonic Imaging, vol. 6, pp. 349-395, 1984.
- [72] S. W. Flax, N. J. Pelc, G. H. Glover, F. D. Gutmann, and M. McLachlan, "Spectral characterization and attenuation measurement in ultrasound," Ultrasonic Imaging, vol. 5, pp. 95-116, 1983.
- [73] T. Chen, B. Ho, and H. R. Zapp, "Impedance and attenuation profile estimation of multilayered material from reflected ultrasound," IEEE Transaction on Instrumentation and Measurement, vol. 40, no. 4, pp. 787-791, 1991.
- [74] M. Fink, F. Hottier, and J. F. Gardoso, "Ultrasonic signal processing for in vivo attenuation measurement: short-time fourier analysis," Ultrasonic Imaging, vol. 5, pp. 117-135, 1983.

- [75] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman and Co., New York, 1979.
- [76] Y. Takefuji, L. Chen, K. Lee, and J. Huffman, "Parallel algorithms for finding a near-maximum independent set of a circle graph," *IEEE Transactions on Neural Networks*, vol. 1, pp. 263-267, 1990.
- [77] F. Gavril, "Algorithms on circular-arc graphs," *Networks*, vol. 4, pp. 357-369, 1974.
- [78] C. Chou, B. Ho, and J. T. Sheu, "Material characterization by ultrasonics using unsupervised competitive learning," To appear *Pattern Recognition Letters*.
- [79] K. J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Transactions on Circuits Devices*, vol. 6, 1987.
- [80] S. Masuda and K. Nakajima, "An optimal algorithm for finding a maximum independent set of a circular-arc graph," *SIAM Journal of Computing*, vol. 17, 1988.
- [81] W. L. Hsu, "The coloring and maximum independent set problems on planar perfect graphs," *Journal of ACM*, vol. 35, 1988.
- [82] R. M. Karp and A. Wigderson, "A fast algorithm for the maximum independent set problem," *Journal of ACM*, vol. 32, 1985.
- [83] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM Journal of Computing*, vol. 15, 1986.
- [84] M. Goldberg and T. Spencer, "A new parallel algorithm for the maximal independent set problem," *SIAM Journal of Computing*, vol. 18, 1989.
- [85] L. Gerhard and W. Lindenberg, "Clique detection for nondirected graphs: two new algorithms," *Computing*, vol. 21, pp. 295-322, 1979.
- [86] P. M. Pardalos and A. T. Phillips, "A global optimization approach for solving the maximum clique problem," *International Journal of Computational Mathematics*, vol. 33, pp. 209-216, 1990.
- [87] R. Carraghan and P. M. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Research Letters*, vol. 9, pp. 375-382, 1990.
- [88] P. L. Hammer, "Some network flow problems solved with pseudo-boolean programming," *Operations Research*, vol. 13, pp. 388-399, 1965.

- [89] F. Barahona, M. Junger, and G. Reinelt, "Experiments in quadratic 0-1 programming," *Mathematical Programming*, vol. 44, pp. 127-137, 1989.
- [90] J. Nodar and B. Ho, "Determination of attenuation-velocity products in a layered homogeneous medium," *International Journal of Imaging Systems and Technology*, vol. 3, pp. 27-32, 1991.
- [91] B. Ho, D. Ye, H. R. Zapp, and N. H. Wang, "Three-dimensional damage assessment in composites by ultrasonic imaging techniques," 43rd Annual Conference of Reinforced Plastics and Composites, 1988.
- [92] A. Yamada, "On-line deconvolution for the high resolution ultrasonic pulse-echo measurement with narrow-band transducer," *IEEE Ultrasonics Symposium Proceedings*, pp. 1027-1030, 1987.
- [93] R. Kuc and D. P. Regula, Jr., "Diffraction effects in reflected ultrasound spectral estimates," *IEEE Transaction on Biomedical Engineering*, vol. BME-31, no. 8, pp. 537-545, Aug. 1984.
- [94] A. K. Jain and R. C. Dubes, "Algorithms for clustering data," Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [95] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

MICHIGAN STATE UNIV. LIBRARIES



31293014172484