THE UNIVERSITY *of* LIVERPOOL

Development of New Digital Signal Processing Procedures and Applications To Speech, Electromyography and Image Processing

Thesis submitted in accordance with the
requirements of the University of Liverpool
for the degree of Doctor in Philosophy

in

Department of Electrical Engineering and Electronics

by

Nan Lu, B.Eng., M.Sc.(Eng.)

March 2008

Development of New Digital Signal Processing Procedures and Applications To Speech, Electromyography and Image Processing

by

Nan Lu

To my parents

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Jihong Wang, for her invaluable support and intellectual guidance. Without her help and encouragement, this project would not be done.

I am grateful to Professor Q.H. Wu for many helpful discussions and much valuable advice along the way.

My thank goes to all colleagues working in our research group for their kindly help during my studying.

Many thanks goes to the Dr. Randeva and Dr. Vatish from Walsgrave Hospital, Coventry, UK, for providing the research data for me.

Many thanks also goes to BlueStar Technology Co., Ltd., for their financial and equipment supports that make it possible for me to conduct this research.

Last but not least, I am greatly indebted to my parents, for their encouragement, guidance and love during the period of my postgraduate life. Without their support, I would not have had this great studying opportunity and this thesis would certainly not have existed.

Abstract

Development of New Digital Signal Processing Procedures and Applications
To Speech, Electromyography and Image Processing

by

Nan Lu

The design of digital signal processing (DSP) procedures combines the areas of mathematics, technology, art and intuition. With recent advances in computer technology, DSP has been applied to many fields in science and technology. To extract useful information from different signals under varying circumstances will have different demands on DSP procedures. The procedures have to extract useful information from large amounts of the data, restore the missing information from damaged signals, or recognize specific information from a noisy environment. The more advanced device certainly need more advanced procedure to be integrated to achieve the best performance, especially for the real-time applications. The thesis summarizes research work in the development of new and advanced procedures for real-time DSP with applications in speech, electromyography (EMG) and image signal processing.

The work on DSP procedures was initially inspired by the field of Chaos Theory. The beginning of the thesis introduces chaos theory, explaining how the theory leads to the development of advanced tools for DSP with a new developed noise reduction procedure applied to sinusoidal experimental signal. Generalised from this theory, a new procedure was proposed and applied to the processing of distorted speech signals, in which a phase space reconstruction-based method is developed to restore missing information to the original speech signal. The experimental results show that the recovered speech obviously more speech recognizable. The signal processing techniques generalised from chaos theory combined with other signal processing methods are also applied to EMG signal processing. A new procedure for EMG signal classification was

developed for pattern recognition to distinguish term and preterm labor signals. The classification accuracy can reach 64.1% which is a very encouraging result.

Real-time detection of moving objects within a complex environment is a fundamental problem faced by image processing. The optical flow based approach is a possible solution. A study to develop motion detection procedures has been carried out with sponsorship from the BlurStar Technology Co. Ltd. Two new advanced motion detection procedures were developed for the real-time video surveillance under a noisy environment. One works by calculation of accumulative optical flow and double background filtering. Another is based on a parallel motion detection strategy which combines temporal difference and pyramidal structure-based optical flow. Experimental results show that both of the procedures can separate the background interference and foreground moving object successfully and raise the alarm when a suspicious object appears. Both procedures have a high capability of anti-interference and preserves high accurate rate detection at the same time, and they are also much faster than conventional detection procedures. Both procedures are now in the test stage to be implemented in industrial applications for real-time video surveillance.

The contributions of the PhD project can be summarized as follows: 1) Development of a new noise reduction procedure based on chaos theory to improve signal-to-noise ratio. 2) Development of a new speech processing procedure based on phase space reconstruction to recover missing information from distorted speech signals. 3) Initial investigation of a new classification procedure for EMG to differentiate normal and abnormal signals. 4) Development of two new advanced motion detection procedures based on an optical flow calculation which detect moving objects accurately and quickly in a noisy environment.

Contents

List of Figures

List of Tables

List of abbreviations

ADC	Analog to Digital Converter
ANN	Artificial Neuron Network
BLT	Bilinear Transform
*Bior*1.5	Biorthogonal Wavelet
DAC	Digital to Analog Converter
DBF	Double Background Filtering
DFT	Discrete Fourier Transform
DSP	Digital Signal Processor
DWT	Discrete Wavelet Transform
EMG	Electromyography
FT	Fourier Transform
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FNN	False Nearest Neighbor
G3	Third-Generation
GPS	Global Positioning System
GSVD	Global Singular Value Decomposition
IIR	Infinite Impulse Response
IUPC	Intrauterine Pressure Catheter
LSA	Log Spectral Amplitude
LSM	Least Mean Square
LSVD	Local Singular Value Decomposition
MATLAB	A programming language for scientific research
MM	Mathematical Morphology
MMSE	Minimum Mean Square Error
OF	Optical Flow
PSR	Phase Space Reconstruction
PTL	Pre-term Labor

RF	Radio Frequency
RMS	Root Mean Square
SE	Structure Element
SNR	Signal to Noise Ratio
STSA	Short Time Spectral Amplitude
SVD	Singular Value Decomposition
TL	Term Labor
VLSI	Very-Large-Scale Integration
VoIP	Voice over Internet Protocol
WPE	Wavelet Packet Energy
WPT	Wavelet Packet Transform
WT	Wavelet Transform

Chapter 1

Introduction

1.1 Overview of digital signal processing

Digital signal processing (DSP) is an area of science and engineering that has developed rapidly over the past 30 years. This rapid development is a result of the significant advances in digital computer technology and integrated-circuit fabrication. The rapid developments in integrated-circuit technology, starting with medium-scale integration (MSI) and progressing to large-scale integration (LSI), and now, very-large-scale integration (VLSI) of electronic circuits has spurred the development of powerful, smaller, faster and cheaper digital computers and special-purpose digital hardware [1]. These inexpensive and relatively fast digital circuits have made it possible to construct highly sophisticated digital systems capable of performing complex digital signal processing functions and tasks.

Not only do digital circuits yield cheaper and more reliable systems for signal processing, they have other advantages as well. In particular, digital processing hardware allows programmable operations. The implementation of DSP procedures in the devices embedded hardware enabled many exciting technology developments in our life. The development of new DSP procedures is largely driven by the need in different areas and tasks of applications. Through procedures, one can more easily modify the signal processing functions to be performed by the hardware. Thus digital hardware and associated procedures

provide a greater degree of flexibility in system design. The work described in the thesis is motivated by searching solutions or better DSP procedures for solving the problems arisen from the real world. The description of the research work will start from the historical review of DSP and fundamental theory required for the work described in the thesis.

1.1.1 The history of digital signal processing

Since the Second World War, if not earlier, engineers and scientists have speculated on the applicability of digital techniques to perform signal processing tasks. For example, at the end of the 1940s, Shannon, Bode and other researchers in Bell Telephone Laboratories discussed the possibility of using digital circuit elements to implement filter functions [2]. At this time, there was, unfortunately, no appropriate hardware available for implementation of digital signal processing procedures. Hence, the cost, the size and the reliability are strongly favored of conventional analog implementations to filters.

During the middle of 1950s, Professor Linville at Massachusetts Institute of Technology (MIT) discussed digital filtering at graduate seminars [3]. By then, control theory, based partly on works by Hurewicz become established as a subject discipline, and sampling and its spectral effects were well understood [4]. A number of mathematical tools, such as z-transform, which had existed since Laplace's time, were now used in the electronic engineering community. Technology at that point, however, was only able to deal with low-frequency control problems or low-frequency seismic signal processing problems. While seismic scientists made notable use of digital filter concepts to solve problems, it was not until the middle of the 1960s that a more formal theory of digital signal processing (DSP) began to emerge. During this period, the advent of the silicon integrated circuit technology made complete digital systems possible, but still quite expensive.

The first major contribution in the area of digital filter synthesis was made by Kaiser at Bell laboratories [5]. His work showed how to design useful filters using bilinear transform (BLT). Further, in about 1965, the famous paper by

Cooley and Turkey was published [6]. In this paper, fast Fourier transform (FFT), an efficient and fast way of performing the discrete Fourier transform (DFT), was demonstrated.

At this time, hardware better suited for implementing digital filters was developed and affordable circuits started to be commercially available. Long finite impulse response (FIR) filters could now be implemented efficiently, thereby becoming a serious competitor to the infinite impulse response (IIR) filters, having better passband properties for a given number of delays. At the same time, new opportunities emerged. It was now possible to achieve time varying, adaptive and non-linear filters that could not be built using conventional analog techniques. One such filter is Kalman filter named after R.E. Kalman [7]. The Kalman filter is a model-based filter that filters the signal according to its statistical rather than its spectral properties.

In the area of adaptive filters, B. Widrow is an important name, especially when talking about the least mean square (LMS) algorithm [8]. Widrow also made significant contributions in the area of neural networks as early as in the 1960s and 1970s. Today, there are many commercial products around which utilize the advantages of digital signal processing including essentially perfect reproducibility, guaranteed accuracy and suitable for large volume production.

1.1.2 Applications of digital signal processing

It is a well known fact that digital signal processing is used in many areas. Here lists a few of them:

1. *Telecommunications.* Wireless or mobile phones are rapidly replacing wired (landline) telephones, both of which are connected to a large-scale telecommunication networks. They are used for voice communication as well as data communications. So also are the computers connected to a different network that is used for data and information processing. Computers are used to generate, transmit, and receive an enormous amount of information through the Internet and will be used more extensively over the same network. In the coming years for voice communications also. This technology

is known as *voice over Internet protocol* (VoIP) or *Internet telephony.* At present, we can transmit and receive a limited amount of text, graphics, pictures and video images from mobile phones, besides voice, music, and other audio signals – all of which are classified as multimedia – because of limited hardware in the mobile phones and not the software that has already been developed. However, the computers can be used to carry out the same functions more efficiently with greater memory and large bandwidth. We see a seamless integration of wireless telephones and computers already developing in the market at present. The new technologies being used in the above mentioned applications are known by such terms as CDMA, TDMA, spread spectrum, echo cancellation, channel coding, adaptive equalization, ADPCM coding and data encryption and decryption, some of which are used in the software to be introduced in the third-generation (G3) mobile phones.

2. *Speech Processing.* The quality of speech transmission in real time over telecommunication networks from wired (landline) telephones or wireless (cellular) telephones is very high. Speech recognition, speech synthesis, speaker verification, speech enhancement, text-to-speech translation, and speech-to-text dictation are some of the other applications of speech processing.

3. *Consumer Electronics.* We have already mentioned cellular or mobile phones. Then we have HDTV, digital cameras, digital phones, answering machines, fax and modem, music synthesizers, recording and mixing of music signals to produce CD and DVDs. Surround-sound entertainment systems including CD and DVD players, laser printers, copying machines, and scanner are found in many homes. But the TV set, PC, telephones, CD-DVD players, and scanners are present in our homes as separate systems. However, the TV set can be used to read email and access the Internet just like the PC; the PC can be used to tune and view TV channels, and record and play music as well as data on CD-DVD in addition to their use to make telephone calls on VoIP. This trend toward the development of fewer systems

with multiple applications is expected to accelerate in the near future.

4. *Biomedical Systems.* The variety of machines used in hospitals and biomedical applications is staggering. Included are X-ray machines, MRI, PET scanning, bone scanning, CT scanning, ultrasound imaging, fetal monitoring, patient monitoring, and ECG and EEC mapping. Another example of advanced digital signal processing is found in hearing aids and cardiac pacemakers.

5. *Image Processing.* Image enhancement, image restoration, image understanding, computer vision, radar and sonar processing, geophysical and seismic data processing, remote sensing, and weather monitoring are some of the applications of image processing. Reconstruction of two-dimensional (2D) images from several pictures taken at different angles and three-dimensional (3D) images from several contiguous slices has been used in many applications.

6. *Military Electronics.* The applications of digital signal processing in military and defense electronics systems use very advanced techniques. Some of the applications are GPS and navigation, radar and sonar image processing, detection and tracking of targets, missile guidance, secure communications, jamming and countermeasures, remote control of surveillance aircraft, and electronic warfare.

7. *Aerospace and Automotive Electronics.* Applications include control of aircraft and automotive engines, monitoring and control of flying performance of aircraft, navigation and communications, vibration analysis and antiskid control of cars, control of brakes in aircrafts, control of suspension, and riding comfort of cars.

8. *Industrial Applications.* Numerial control, robotics, control of engines and motors, manufacturing automation, security access, and videoconferencing are a few of the industrial applications.

Obviously there are some overlaps among these applications in terms of devices and systems. It is also true that basic mathematical operations are common in all the applications and systems. Needless to say, the number of new applications to the existing applications will continue to grow at a very rapid rate in the near future.

1.1.3 Filters

Filtering [9] is the most common form of signal processing used in all the applications, to remove unwanted frequency components and to improve magnitude, phase, or group delays in the parts of the spectrum of a signal. The literature reports on filters can be classified into two sets: (1) to derive the transfer function of the filter such that the magnitude, phase, or group delay approximates the given frequency response specifications and (2) procedures to design the filters using the hardware components.

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. The following block diagram illustrates the basic idea.

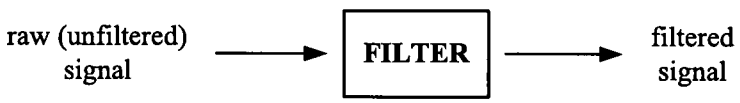


Figure 1.1: Filter block diagram

There are two main kinds of filter, *analog* and *digital*. They are quite different in their physical makeup and in how they work.

An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and operational amplifier (op amps) to produce the required filtering effect. Such filter circuits are widely used in applications such as noise reduction, video signal enhancement, graphic equalisers in hi-fi systems, and many other areas. There are well-established standard techniques for designing an analog filter circuit to satisfy the given requirement. At all

stages, the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved.

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialised DSP (Digital Signal Processor) chip. The analog input signal must first be sampled and digitised using an ADC (analog to digital converter). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a DAC (digital to analog converter) to convert the signal back to analog form. Note that in a digital filter, the signal is represented by a sequence of numbers, rather than a voltage or current. The following diagram shows the basic setup of such a system.

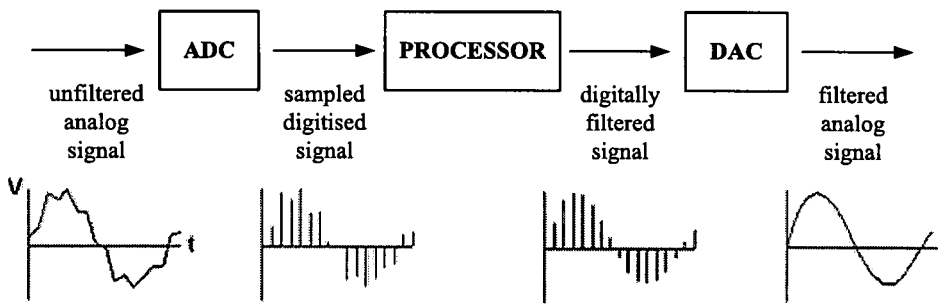


Figure 1.2: Digital filter system

Advantages of digital filters

The following list gives some of the main advantages of digital over analog filters.

1. A digital filter is *programmable*, i.e. its operation is determined by a program

stored in the processor's memory. This means that the digital filter can easily be changed without affecting the circuitry (hardware). An analog filter can only be changed by redesigning the filter circuit.

2. Digital filters are easily *designed*, *tested* and *implemented* on a general-purpose computer or workstation.

3. The characteristics of analog filter circuits (particularly those containing active components) are subject to drift and are dependent on temperature. Digital filters do not suffer from these problems, and so are extremely *stable* with respect both to time and temperature.

4. Unlike their analog counterparts, digital filters can handle *low frequency signals* accurately. As the speed of DSP technology continues to increase, digital filters are being applied to high frequency signals in the RF (radio frequency) domain, which in the past was the exclusive preserve of analog technology.

5. Digital filters are very much more *versatile* in their ability to process signals in a variety of ways; this includes the ability of some types of digital filter to adapt to changes in the characteristics of the signal.

Categories of digital filters

Since digital filters consist of computer program code, there are an infinite number of possible filter architectures and variations. The common filter architectures can be divided into three categories: linear, adaptive, and non-linear filters.

1. *Linear filter*

   Much of the available theory deals with linear filters, where the filter output is a (possibly time-varying) linear function of the filter input. There are basically two distinct theoretical approaches to the design of such filters: classical and optimal.

   The classical approach aimed at designing frequency-selective filters such as

lowpass/bandpass/notch filters etc. For a noise reduction application, for example, it is based on knowledge of the gross frequency spectrum of both the useful signal and the noise components. It is applicable mainly when the signal and noise occupy clearly different frequency bands. A classic filter example is that of bandpass filtering a frequency rich chirp signal. The frequency components of the chirp within a selected band can be extracted through a number of linear filtering methods. This example shows that linear methods in signal processing can indeed be markedly effective.

Optimal filter design [10], on the other hand, is based on optimization theory, where the filter is designed to be best (in some sense). If the signal and noise are viewed as stochastic processes, based on their statistical parameters, an optimal filter is designed that, for example, minimizes the effects of the noise at the filter output according to some statistical criterion. This is mostly based on minimizing the mean-square value of the difference between the actual filter output and some desired output.

In fact, linear signal processing enjoys the rich theory of linear systems, and in many applications linear signal processing algorithms prove to be optimal. Most importantly, linear filters are inherently simple to implement, perhaps the dominant reason for their widespread use.

2. *Adaptive filter*

   The theory of adaptive filter [11] is derived from optimal filter design. The adaptive filter is a self-designing filter and an alternative method to linear filter. An adaptive filter has an adaptation algorithm that is meant to monitor the environment and vary the filter transfer function accordingly. The algorithm starts from a set of initial conditions that may correspond to complete ignorance about the environment and attempts to find the optimum filter design based on the actual signals received. In a stationary environment the filter is expected to be like optimal filter. In a non-stationary environment, the filter is expected to track time variations and vary its filter coefficients accordingly.

3. *Non-linear filter*

Nonlinear methods in signal and image processing [12] have become increasingly popular over the past thirty years. there are two general families of nonlinear filters: the homomorphic and polynomial filters, and the order statistic and morphological filters [13]. Analysis and design of homomorphic and polynomial filters resemble traditional methods used for linear systems and filters in many ways. Other statistic and morphological filters can not be analyzed efficiently using generalizations of linear techniques. The median filter is an example of an order statistic filter, and is probably the oldest and most widely used order statistic filter. Morphological filters are based on a form of set algebra known as mathematical morphology. Most morphological filters use extreme order statistic (minimum and maximum values) within a filter window, so they are closely related to order statistic filters.

Although linear filters will continue to play an important role in signal processing, nonlinear filters are emerging as viable alternative solutions. The major forces that motivate the implementation of nonlinear signal processing procedures are the growth of increasingly challenging applications and the development of more powerful computers. Emerging multimedia and communications applications are becoming significantly more complex. Consequently, they require the use of increasingly sophisticated signal processing procedure. At the same time, the ongoing advances of computers and digital signal processors, in terms of speed, size and cost, makes the implementation of sophisticated procedures practical and cost effective.

1.2 Methodologies in digital signal processing

In digital signal processing, engineers usually study digital signals in one of the following domains: time domain (one-dimensional signals), spatial domain (multidimensional signals), frequency domain, and phase space domain. They choose the domain in which to process a signal by making an informed guess (or by trying different possibilities) as to which domain best represents the essential characteristics of the signal. A sequence of samples from a measuring

device produces a time or spatial domain representation, whereas a discrete Fourier transform produces the frequency domain information, that is the frequency spectrum. Chaotic method transforms the signals from time domain into phase space domain to show their dynamical information, called as phase space reconstruction.

1.2.1 Time and space domains

The most common processing approach in the time or space domain is enhancement of the input signal through the method of filtering. Filtering generally consists of some transformation of a number of surrounding samples around the current sample of the input or output signal. There are various ways to characterize filters, for example:

- A "linear" filter is a linear transformation of input samples; other filters are "non-linear". Linear filters satisfy the superposition condition, i.e. if an input is a weighted linear combination of different signals, the output is an equally weighted linear combination of the corresponding output signals.

- A "causal" filter uses only previous samples of the input or output signals; while a "non-causal" filter uses future input samples. A non-causal filter can usually be changed into a causal filter by adding a delay to it.

- A "time-invariant" filter has constant properties over time; other filters such as adaptive filters change in time.

- Some filters are "stable", others are "unstable". A stable filter produces an output that converges to a constant value with time, or remains bounded within a finite interval. An unstable filter produces output which diverges.

- A "finite impulse response" (FIR) filter uses only the input signal, while an "infinite impulse response" filter (IIR) uses both the input signal and

previous samples of the output signal. FIR filters are always stable, while IIR filters may be unstable.

Classic filters

Two of the most commonly used classical filters are Butterworth filters and Chebyshev filters.

- Butterworth filters. A smooth response at all frequencies and a monotonic decrease from the specified cutoff frequencies characterize the frequency response of Butterworth filters. Butterworth filters are maximally flat – the ideal response of unity in the passband and zero in the stopband. The half power frequency or the 3dB down frequency corresponds to the specified cutoff frequencies.

- Chebyshev filters. Chebyshev filters minimize peak error in the passband by accounting for the maximum absolute value of the difference between the ideal filter and the filter response. The frequency response characteristics of Chebyshev filters have an equal ripple magnitude response in the passband, monotonically decreasing magnitude response in the stopband, and a sharper roll-off than Butterworth filters. The cutoff frequency for Chebyshev filters is defined as the end of the passband.

The advantage of Chebyshev filters over Butterworth filters is that Chebyshev filters have a sharper transition between the passband and the stopband with a lower order filter. This produces smaller absolute errors and higher execution speed.

Mathematical morphology

Mathematical morphology (MM) is a branch of digital signal processing and a non-linear approach developed from set theory and geometry [14]. In contrast with the integral transform algorithm like Fourier and Wavelet analysis, the MM filtering technique handles a signal in a completely different way. It was widely used in the area of image processing, machine vision and pattern

recognition due to its robustness in preserving the main shape while suppressing noise. When acting upon complex shapes, morphological operations are able to decompose them into meaningful parts and separate them from the background, as well as preserve the main shape characteristics. Furthermore, the mathematical calculation involved in MM includes only addition, subtraction and maximum and minimum operations without any multiplication and division.

The underlying basis of the MM filtering technique is to process signals by a function known as the structure element (SE) in general or structuring function for a 1-D signal. With various combinations of the fundamental morphological operators, Dilation and Erosion, the technique is able to preserve or suppress the feature represented by the SE and obtain a signal with only components of interest. Therefore, such an algebraic system with its operators and their combinations allows the underlying shapes to be identified, reconstructed and enhanced from their noisy, distorted forms.

MM has several advantages over other techniques especially when applied to signal/image processing [15] which can preserve the edge information, use shape-based processing, design to be idempotent and compute efficiently.

1.2.2 Frequency domain

Signals are converted from time or space domain to the frequency domain usually through Fourier transform. Fourier transform converts the signal information to a magnitude and phase component of each frequency. Often the Fourier transform is converted to the power spectrum, which is the magnitude of each frequency component squared. The most common purpose for analysis of signals in the frequency domain is analysis of signal properties. The engineer can study the spectrum to get information of which frequencies are present in the input signal and which are missing.

Fourier transform

Fourier transform was proposed by Joseph Fourier in 1807 [16]. The Fourier transform's utility lies in its ability to analyze a signal in the time domain for its frequency content. The transform works by translating a function in the time domain into a function in the frequency domain. Fourier transforms are based on the concept that real world signals can be approximated by a sum of sinusoids, each at a different frequency. The more sinusoids included in the sum. the better the approximation. Such a sum of sinusoids is called a trigonometric Fourier series. The terms of the Fourier series for simple waveforms can be found using calculus. The frequency of each sinusoid in the series is an integer multiple of the frequency of the signal being approximated. These are referred to as the harmonics of the original waveform.

Fast fourier transform

The most popular computer algorithm for generating a frequency spectrum is the Fast Fourier Transform (FFT), which was proposed by Cooley and Tukey in 1965 [17]. FFT is a milestone for digital signal processing. As the name implies, the FFT is very efficient but it does have one quirk that affects the way it is used. The FFT reduces the number of computations needed for N points from $2N^2$ to $2Nlog_2N$.

Wavelet transform

Wavelet transform provides an excellent time-frequency description for signal processing applications. Wavelets [18] are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology.

Wavelets are functions that satisfy certain mathematical requirements and

are used in representing data or other functions. The fundamental idea behind wavelets is to analyze according to scale which plays a special role in wavelet analysis. Wavelet algorithms process data at different scales or resolutions [19]. If we look at a signal with a large "window" which is for the low-frequency components, we would notice gross features. Similarly, if we look at a signal with a small "window" which is for the high-frequency components, we would notice small features.

The wavelet analysis procedure is to adopt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequent analysis is performed with a dilated, low-frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion (using coefficients in a linear combination of the wavelet functions), data operations can be performed using just the corresponding wavelet coefficients. And if you further choose the best wavelets adapted to your data, or truncate the coefficients below a threshold, your data is sparsely represented. This sparse coding makes wavelets an excellent tool in the field of data compression.

1.2.3 Phase space domain

In mathematics and physics, a phase space, introduced by Willard Gibbs [20] in 1901, is a space in which all possible states of a system are represented, with each possible state of the system corresponding to one unique point in the phase space. In a phase space, every degree of freedom or parameter of the system is represented as an axis of a multidimensional space. For every possible state of the system, or allowed combination of values of the system's parameters, a point is plotted in the multidimensional space. Often this succession of plotted points is analogous to the system's state evolving over time. In the end, the phase diagram represents all that the system can be, and its shape can easily elucidate qualities of the system that might not be obvious otherwise. A phase space may contain very many dimensions.

Chaos theory

Much of what is known about digital signal processing methods has been learned using linear system theory. However, many signals are apparently random or aperiodic in time. Traditionally, the randomness in signals has been ascribed to noise or interactions between very large numbers of constituent components.

One of the most important mathematical discoveries of the past few decades is that random behavior can arise in deterministic nonlinear system with just a few degrees of freedom. This discovery gives new hope to providing simple mathematical models for analyzing, and ultimately controlling, signal processing systems.

In mathematics and physics, chaos theory [21] describes the behavior of certain nonlinear dynamical systems that under certain conditions exhibit a phenomenon known as chaos. Among the characteristics of chaotic systems is sensitivity to initial conditions (popularly referred to as the butterfly effect). As a result of this sensitivity, the behavior of systems that exhibit chaos appears to be random, even though the system is deterministic in the sense that it is well defined and contains no random parameters. Examples of such systems include the atmosphere, the solar system, plate tectonics, turbulent fluids, economics, and population growth.

Chaos theory is applied in many scientific disciplines: mathematics, biology, computer science, economics, engineering, philosophy, physics, politics, population dynamics, psychology, signal processing,robotics, etc.

1.3 Motivation and objectives

In contrast with traditional signal processing methods, the technique based on chaos theory processes signals from a different perspective and it is a non-linear approach that is developed based on embedding theory and attractor reconstruction. The chaos theory generalised methods attracted a great attentions in signal processing research due to its advantage in high sensitiveness to the minimal changing of the signal. When acting upon the regular signal contaminated by the very low level noise which is difficult to be detected by other methods, chaos theory generalised techniques are able to decompose the signal into meaningful parts and separate the noise component from the original signal in the phase space, as well as preserve the main clear signal characteristics. This distinct advantage motivated me to study chaos theory and to investigate more efficient ways for signal processing to reveal the characteristics of the noise embedded in signal by applying the chaos theory principles.

Real-time detection of moving object is very important for video surveillance. It has attracted a great interest from computer vision researchers due to its promising applications in many areas. The research in this area is still in its early developmental stage and leaves many unsolved interesting topics for researchers to explore, especially in the area of improving the robustness when it is applied in a complex environment. I started to be interested in this area by helping processing a set of video signal from police station. Afterwards, with the encouragement and support from the company, I extended my project to the topic of moving object detection with a variable moving background.

The overall goal of the project is to develop new advanced digital signal processing procedures with applications to EMG, speech and image signals. The goal is to be achieved by realizing the following objectives.

Based on the chaos theory, a new noise reduction procedure has been proposed for the sinusoidal experimental signal. Although many other methods can effectively reduce the noise, the new procedure has a considerable advantage in computation time and instant response to signal changes in comparison with other techniques. The objective of this learning is to process the signal

for noise reduction using chaos theory generalised techniques.

A procedure with its application discussed in this thesis is speech signal restoration whose main idea is based on chaos theory. With the developing of communications technologies, it requires reliable transmission and reproduction of audio and speech signals. However, in many cases the speech signals received may be partly corrupted. It is very importance to separate the clear speech signal from the background noise and improve its quality before any processing. The objective of this research work is to reduce the corrupting noise component of a noisy speech signal while preserving the original clean speech quality as much as possible.

Another procedure with its application also discussed in this thesis is the Electromyography signal classification which uses the chaos theory generalised techniques combined with other signal processing methods. Proper diagnosis of labor is one of the major challenges faced by obstetricians. There are no accurate and objective methods to predict the onset of labor, to differentiate true and false labor both for term and for preterm patients, and to determine whether false labor will progress to true labor and on what time scale. The objective of this studying is to find some ways to distinguish the abnormal preterm labour (PTL) signals from normal term labour (TL) signals.

Two procedures discussed in this thesis are about motion detection. Motion detection under noisy environment background is a fundamental problem of video processing. The noisy environment is normally caused by some interferences from the nature world. Separation of the background interference noise and foreground moving object is an important issue and has been an very challenge topic. The objective of this research work is to detect moving object efficiently under noisy environment background.

1.4 Major contributions of this thesis

The major contributions of this thesis are the development of new digital signal processing procedures for speech, electromyography and image processing which include noise reduction, information restoration, feature extraction and classification, and motion detection, which are summarised as:

- Chaos theory is studied in the field of digital signal processing in order to develop a new noise reduction procedure. Improved signal-to-noise ratio (SNR) value has been achieved for the sinusoidal signal contaminated by low level noise. This study proves that it is feasible to use chaos theory generalised techniques in digital signal processing.

- A new speech signal processing procedure based on chaos theory is developed for speech signal restoration. A phase space reconstruction-based method is proposed to effectively identify the distorted segments of the speech signal and to recover the missing information of the signal in phase space. The recovered signal obviously more speech recognizable by using the new procedure. [22, 23]

- A procedure for electromyography signal processing using chaos theory generalised techniques combined with other signal processing methods is developed for signal classification. A synthesized method is proposed for signal preprocessing, feature extraction and classification. Although the result of classification accuracy is not high as expected, the result has demonstrated the potential of the method and it convinced that the method should be studied further. [24, 25]

- Two motion detection procedures which integrate different image processing methods and strategies are developed. Both procedures were designed to detect moving objects under noisy environment. The results show that two procedures can separate the background interference and foreground moving objects successfully and are suitable for the application of real-time video surveillance. [26, 27, 28, 29, 30, 31]

1.5 Thesis outline

The thesis is organised as follows:

Chapter 2 This chapter introduces the chaos theory. The essential concepts of dynamical systems are given and some examples of chaotic dynamical systems are presented. The chaos theory generalised techniques used for phase space reconstruction are introduced, such as attractor, time delay, embedding dimension, embedding method and singular value decomposition method. The definitions and calculations for two invariant characteristics of chaotic dynamics systems, fractal dimension and Lyapunov exponents are discussed.

Chapter 3 In this chapter, a new noise reduction procedure based on chaos theory is developed. In this procedure, the noisy signal is transformed from the time domain to the phase space domain and the noise components can be eliminated by projecting the signal from the main phase space into a subspace which can separate the noise component from the clear signal. In this way, the signal is transformed back to the time domain. The experimental results show that the better noise reduction result can be achieved by using this new procedure.

Chapter 4 Chapter 4 contributes to an application for speech signal processing based on chaos theory. A phase space reconstruction-based procedure is developed for restoration of noise contaminated speech signal. The method embeds the noisy signal into a high dimensional reconstructed phase space where the singular value decomposition (SVD) method is used to identify the distorted segments and the space interpolation method is used to reconstruct the signal trajectory in the phase space. Then, a time aligned weighted average method is used to transform the signal back to the time domain. The experimental results show that the recovered speech signal is obviously more speech recognizable.

Chapter 5 In this chapter, the chaos theory generalised techniques are applied to electromyograph signal processing and a combined signal processing method is introduced. A new method is developed which integrates numbers of signal processing methods. Pattern recognition techniques are developed for the EMG signal classification. In this method, the signal is preprocessed by using wavelet transform to get rid of the noise. Then, the correlation dimension value is calculated along the signal to extract the contraction patterns of the signal and the average wavelet packet energy of contraction patterns is calculated by using wavelet packet transform method. Finally, the artificial neural network (ANN) method is used to classify the normal and abnormal signals. The experimental results show that the accuracy of classification result is very encouraging.

Chapter 6 Chapter 6 describes the work in application of image processing for motion detection. A procedure based on accumulative optical flow and double background filtering is developed for real-time video surveillance under noisy environment background. The temporal difference method is used firstly to detect initial possible motion area. Secondly, the optical flow is calculated based on the result of the temporal difference method to obtain the accurate motion area. Thirdly, a double background filtering method with morphological processing is used to get rid of the background interference. Finally, motion area detection methods are used to detect the moving object. The experimental results show that the procedure proposed can separate the background interference and foreground moving object very well.

Chapter 7 In this chapter, further development on image processing for motion detection is introduced. A procedure based on bidirectional motion detection strategy using temporal difference and pyramidal structure-based optical flow is developed. The temporal difference method is used initially for the whole image to detect coarse motion areas. The image is then decomposed into coarse levels based on Guassian pyramid construction and the optical flow method is performed independently at each resolution level of the image

pyramidal structure for the motion area estimation. Finally, the moving object can be detected by combining the results of parallel motion detection strategy. The result shows that the procedure is much faster than conventional mono-resolution detection methods and it also preserves high accurate rate detection at the same time.

Chapter 8 The conclusion of the thesis is made and some recommendations are given for future work.

1.6 Auto-bibliography

Conference papers

1. **N. Lu**, J. Wang and Q.H. Wu, Automatic vehicle license plate recognition using digital image processing methods, *Proceedings of the 11th Chinese Automation and Computing Society Conference*, Sheffield, UK, pp.155-160, September 2005. (Published)

2. **N. Lu**, J. Wang and Q.H. Wu, Speech signal restoration using phase space reconstruction, *XVIII International Conference on Systems Engineering*, Coventry, UK, pp.155-160, September 2006. (Published)

3. **N. Lu** and J. Wang, Uterine electromyography features extraction for classification of term and pre-term signals, *Proceedings of The 2nd International Conference on Complex Systems and Applications*, Jinan, China, 8-10 June, 2007. (Published)

4. **N. Lu**, J. Wang, Q.H. Wu and L. Yang, Motion detection based on accumulative optical flow and double background filtering, *Proceedings of World Congress on Engineering 2007*, London, UK, vol.1, pp.602-607, 2-4 July, 2007. (Published, The Best Paper Award, See Appendix D)

5. **N. Lu**, J. Wang and Q.H. Wu, Motion detection using temporal difference and pyramidal structure-based optical flow. *American Control Conference 2008*, Seattle, Washington, USA, 11-13 June, 2008. (Submitted)

6. **N. Lu**, J. Wang and Q.H. Wu, Motion detection using optical flow and inter-frame block-based histogram correlation, *International Conference on Modelling, Identification and Control 2008*, Shanghai, China, 29 June - 2 July, 2008. (Submitted)

7. **N. Lu**, J. Wang, Q.H. Wu and J.L. Wei, Bidirectional motion detection strategy for real-time surveillance. *UKACC Control Conference 2008*, Manchester, UK, 2-4 September, 2008. (Submitted)

Journal papers

1. **N. Lu**, J. Wang, Q.H. Wu and L. Yang, An improved motion detection method for real-time surveillance, *IAENG International Journal of Computer Science*, 35(1):119-128, Mar., 2008. (Published)

2. **N. Lu**, J. Wang and Q.H. Wu, Uterine electromyography signal feature extraction and classification, *International Journal of Modelling, Identification and Control.* (Submitted)

3. **N. Lu**, J. Wang and Q.H. Wu, Bidirectional motion detection strategy for real-time surveillance, *International Journal of Automation and Control.* (Submitted)

4. **N. Lu**, J. Wang and Q.H. Wu, Signal restoration using phase space reconstruction method, *IEEE Transaction On Signal Processing.* (Submitted)

Chapter 2

Chaos Theory

2.1 Introduction to chaos theory

Chaos theory has been developed from its obscure roots in the seventies and become one of the most fascinating fields in existence. At the forefront of much research on physical systems, the approaches generalised from chaos theory has already been applied to fields such as arrhythmic pacemakers, image compression, and fluid dynamics. The main ideas of some procedures developed in this thesis have been inspired by chaos theory.

Formally, chaos theory is defined as the study of complex nonlinear dynamic systems. The dynamical system concept is a mathematical formalization for any fixed "rule" which describes the time dependence of a point's position in its ambient space. Examples include the mathematical models that describe the swinging of a clock pendulum, the flow of water in a pipe, and the number of fish each spring in a lake. A dynamical system has a state determined by a collection of real numbers, or more generally by a set of points in an appropriate state space. Small changes in the state of the system correspond to small changes in the numbers. The numbers are also the coordinates of a geometrical spacea manifold. The evolution rule of the dynamical system is a fixed rule that describes what future states follow from the current state. The rule is deterministic: for a given time interval only one future state follows from the current state. Thus chaos theory is, very generally, the study of

forever changing complex systems based on mathematical concepts of recursion, whether in the form of a recursive process or a set of differential equations modelling a physical system.

The most commonly held misconception about chaos theory is that chaos theory is about disorder. The "chaos" in chaos theory is order–not simply order, but the very ESSENCE of order. It is true that chaos theory dictates that minor changes can cause huge fluctuations. But one of the central concepts of chaos theory is that while it is impossible to exactly predict the state of a system, it is generally quite possible, even easy, to model the overall behavior of a system. Thus, chaos theory lays emphasis not on the disorder of the system–the inherent unpredictability of a system–but on the order inherent in the system–the universal behavior of similar systems.

Chaos theory predicts that complex nonlinear systems are inherently unpredictable, but, at the same time, chaos theory also insures that the way to express such an unpredictable system lies not in exact equations, but in representations of the behavior of a system. Thus, chaos theory, which many think is about unpredictability, is at the same time about predictability in even the most unstable systems.

The techniques generalised from chaos theory have been used to model biological systems, which are of course some of the most chaotic systems imaginable. Systems of dynamic equations have been used to model everything from population growth to epidemics to arrhythmic heart palpitations. In fact, almost any chaotic system can be readily modeled–the stock market provides trends which can be analyzed with strange attractors more readily than with conventional explicit equations; a dripping faucet seems random to the untrained ear, but when plotted as a strange attractor, reveals an eerie order unexpected by conventional means. In a word, chaos theory gives people a wonderfully interesting way to become more interested in mathematics.

The term chaos as used in mathematics was coined by the applied mathematician James A. Yorke [32]. The availability of cheaper, more powerful computers broadens the applicability of chaos theory. Currently, chaos theory continues to be a very active area of research.

2.2 History of chaos theory

The first discoverer of chaos can plausibly be argued to be Jacques Hadamard, who in 1898 published an influential study of the chaotic motion of a free particle gliding frictionlessly on a surface of constant negative curvature [33]. In the system studied, Hadamard was able to show that all trajectories are unstable, in that all particle trajectories diverge exponentially from one-another, with positive Lyapunov exponent. In the early 1900s, Henri Poincar while studying the three-body problem [34], found that there can be orbits which are nonperiodic, and yet not forever increasing nor approaching a fixed point. Much of the early theory was developed almost entirely by mathematicians, under the name of ergodic theory [35]. Later studies, also on the topic of nonlinear differential equations, were carried out by G.D. Birkhoff [36], A.N. Kolmogorov [37], M.L. Cartwright [38], J.E. Littlewood [39], and Stephen Smale [40]. Except for Smale, these studies were all directly inspired by physics: the three-body problem in the case of Birkhoff, turbulence and astronomical problems in the case of Kolmogorov, and radio engineering in the case of Cartwright and Littlewood. Although chaotic planetary motion had not been observed, experimentalists had encountered turbulence in fluid motion and nonperiodic oscillation in radio circuits without the benefit of a theory to explain what they were seeing.

Chaos theory progressed more rapidly after 1950s, when it first became evident for some scientists that linear theory, the prevailing system theory at that time, simply could not explain the observed behavior of certain experiments like that of the logistic map [41]. The main catalyst for the development of chaos theory was the electronic computer. Much of the mathematics of chaos theory involves the repeated iteration of simple mathematical formulas, which would be impractical to do by hand. Electronic computers made these repeated calculations practical. One of the earliest electronic digital computers, ENIAC, was used to run simple weather forecasting models.

An early pioneer of the theory was Edward Lorenz whose interest in chaos came about accidentally through his work on weather prediction in 1961 [42]. Lorenz was using a basic computer, a Royal McBee LGP-30, to run his weather

simulation. He wanted to see a sequence of data again and to save time he started the simulation in the middle of its course. He was able to do this by entering a printout of the data corresponding to conditions in the middle of his simulation which he had calculated last time.

To his surprise the weather that the machine began to predict was completely different from the weather calculated before. Lorenz tracked this down to the computer printout. The printout rounded variables off to a 3-digit number, but the computer worked with 6-digit numbers. This difference is tiny and the consensus at the time would have been that it should have had practically no effect. However Lorenz had discovered that small changes in initial conditions produced large changes in the long-term outcome.

Yoshisuke Ueda independently identified a chaotic phenomenon as such by using an analog computer on November 27, 1961 [43]. The chaos exhibited by an analog computer is truly a natural phenomenon, in contrast with those discovered by a digital computer. Ueda's supervising professor, Hayashi, did not believe in chaos throughout his life, and thus he prohibited Ueda from publishing his findings until 1970 [44].

2.3 Introduction to dynamical systems

Dynamics is the study of change, and a dynamical system is any system that evolves in time. When we speak of dynamical systems, we are talking about a system of equations that describe how each variable changes with time. Dynamical systems whose behavior changes continuously in time are mathematically described by a coupled set of first-order autonomous ordinary differential equations as follows:

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2, \ldots, x_n, t) \\ \frac{dx_2}{dt} &= f_2(x_1, x_2, \ldots, x_n, t) \\ &\vdots \\ \frac{dx_n}{dt} &= f_n(x_1, x_2, \ldots, x_n, t) \end{aligned} \tag{2.3.1}$$

or in a more compact form as:

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}, \vec{\mu}) \tag{2.3.2}$$

where the components of the vector $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ are the dynamical variables of the system, the components of the vector $\vec{\mu}$ are parameters, and the components of the vector field $\vec{F}$ are the dynamical rules governing the behavior of the dynamical variables. Dynamical systems whose behavior changes at discrete time intervals are described by a coupled set of first-order autonomous ordinary difference equations as follows::

$$\vec{x}(n+1) = \vec{G}(\vec{x}(n), \vec{\mu}) \tag{2.3.3}$$

In this equation $\vec{G}$ describes the dynamical rules and time is represented by the integer n. A discrete dynamical system may be obtained from a continuous dynamical system (2.3.2) by sampling the solution of the continuous dynamical system at regular time intervals. Under modest smoothness assumptions about the dynamical rules, the solutions of dynamical systems are unique and the dynamical system is deterministic; that is, the state of the dynamical system for all times is uniquely determined by the state at any one time.

One of the surprising and far-reaching mathematical discoveries of the past few decades has been that the solutions of deterministic nonlinear dynamical systems may be random. This behavior is called deterministic chaos. The discovery of deterministic chaos is surprising because randomness has been traditionally associated with unknown external disturbances (noise). What makes the discovery far reaching is that most dynamical systems are nonlinear and most nonlinear systems have random solutions. Deterministic chaos has immediate ramifications for constructing mathematical models for systems characterized by random signals. A fundamental question in this regard is: Are all random signals equally random? It turns out that they are not. Random signals generated by noise are fundamentally different from random signals generated by deterministic dynamics with small numbers of dynamical variables. The difference is not revealed by statistical analysis but is instead revealed by dynamical analysis based on phase space reconstruction.

2.4 Phase space

Phase space is an abstract mathematical space spanned by the dynamical variables of the system. The state of the dynamical system at a given instant in time can be represented by a point in this phase space. If there are n dynamical variables, then the state at a given time can be represented by a point in the Euclidean space $\mathbb{E}^n$. As the dynamical variables change their values in time, the representative point traces out a path in the phase space which is a continuous curve in the case of a continuous dynamical system and a sequence of points in the case of a discrete dynamical system.

2.4.1 Attractors

An attractor is an attracting set that contains a dense orbit. A system must be dissipative in order to have an attractor. Since phase space volume elements decrease in time in dissipative systems, it follows that attractors must occupy zero volume in phase space. A limit cycle attractor is a periodic attractor.

A strange attractor is an aperiodic attractor with additional properties that (1) phase space paths through all points on the attractor diverge on average at an exponential rate and (2) the dimension of the set of points comprised by the attractor is not an integer.

With a dynamic system, a phase space is a space spanned by the state variables of a dynamical system. The trajectory is the solution of a dynamical system in phase space. For a dissipative nonlinear system, the solution of a d-dimensional nonlinear dynamical system is usually confined to a relatively low dimensional phase space. This is one of the reasons for the great interest shown in low dimensional chaotic strange attractors.

The Lorenz attractor [45] and the Rössler attractor [46] are two examples of low dimensional chaotic strange attractors. The Lorenz equations [45] are defined as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma & 0 \\ \gamma & -1 & -x \\ 0 & x & -\beta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2.4.1}$$

where (σ, γ, β) are the system parameters. Figure 2.1 shows the time series for the equation (2.4.1) and Fig.2.2 shows the attractor of Lorenz system in a 3-dimensional phase space when $(\sigma, \gamma, \beta) = (10, 28, 8/3)$.

The Rössler system consists of three differential equations as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} 0 & -1 & -1 \\ 1 & \sigma & 0 \\ z & 0 & -\beta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \gamma \end{pmatrix} \tag{2.4.2}$$

where (σ, γ, β) are the system parameters. Figure 2.3 shows the time series for the equation (2.4.2) and Fig.2.4 shows the attractor of the Rössler system in a 3-dimensional phase space when $(\sigma, \gamma, \beta) = (0.2, 0.2, 5.7)$.

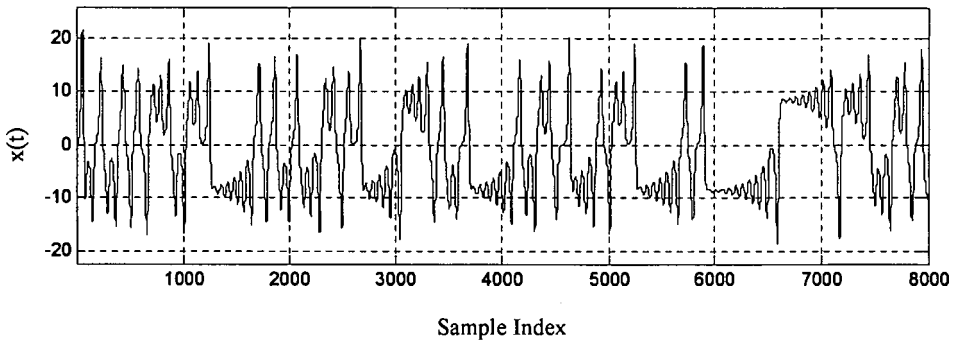


Figure 2.1: The time series of the $x(t)$ component (8000 points) from the Lorenz model

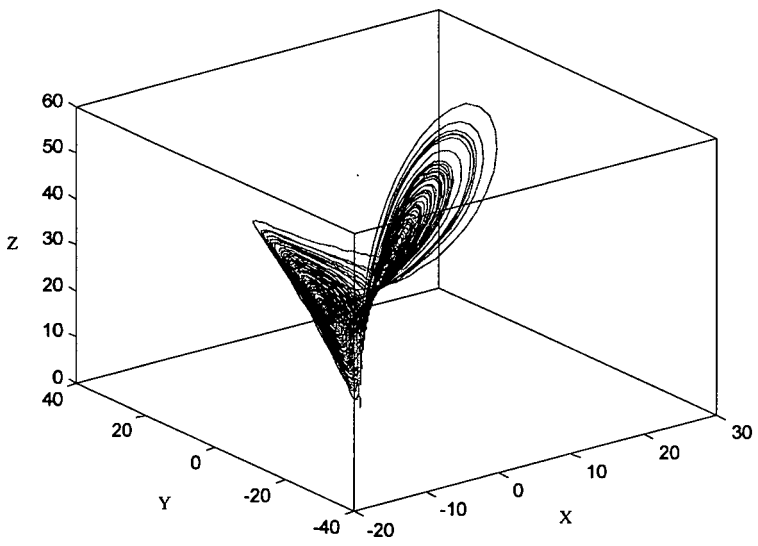


Figure 2.2: Lorenz attractor, $(\sigma, \gamma, \beta) = (10, 28, 8/3)$.

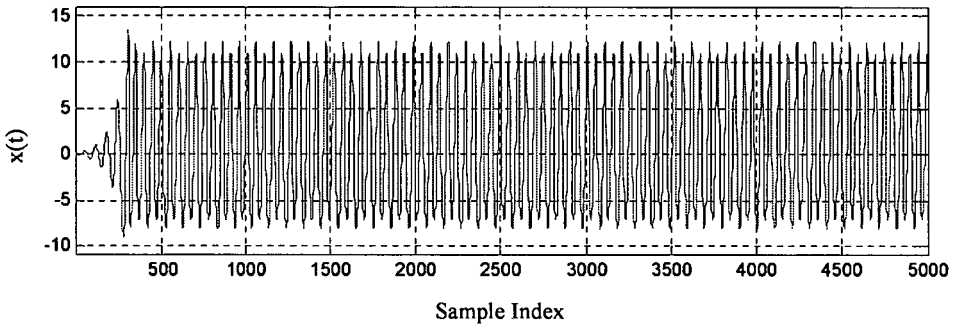


Figure 2.3: The time series of the $x(t)$ component (5000 points) from the Rössler model

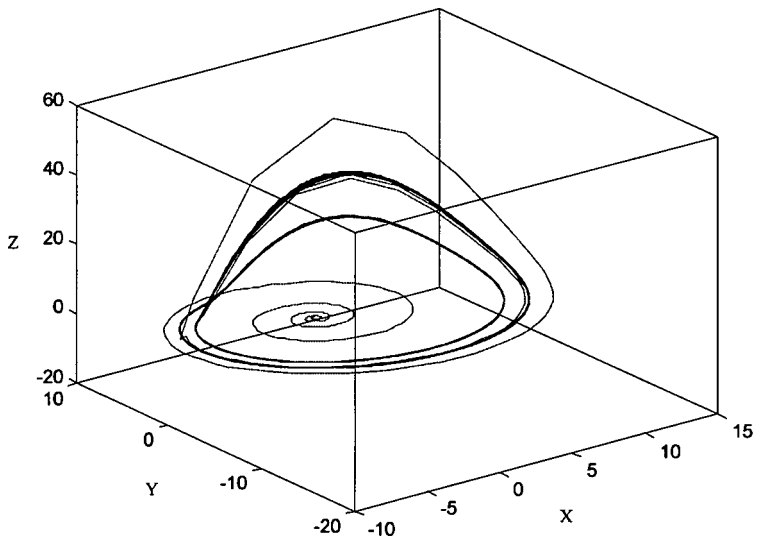


Figure 2.4: Rössler attractor, $(\sigma, \gamma, \beta) = (0.2, 0.2, 5.7)$.

2.4.2 Low dimensional discrete dynamical systems

In contrast to nonlinear continuous autonomous system, the minimum dimension for the existence of chaos for discrete dynamical systems (2.3.3) is $d = 1$ [47]. The famous logistic map [47] is a one-dimensional discrete dynamical system defined as follows:

$$s_{n+1} = r s_n (1 - s_n) \tag{2.4.3}$$

where r is the system parameter. The logistic map is one of the most simple form of a chaotic process. Basically, this map, like any one-dimensional map, is a rule for getting a number from a number which receives a real number between 0 and 1, and returns a real number in [0,1] again. The parameter r is fixed, but if one studies the map for different values of r (up to 4, else the unit interval is no longer invariant) it is found that r is the catalyst for chaos. Figure 2.5 shows the logistic map for various values of r when the initial value s is 0.4. The horizontal axis shows the values of the parameter r while the vertical axis shows the possible long-term values of s.

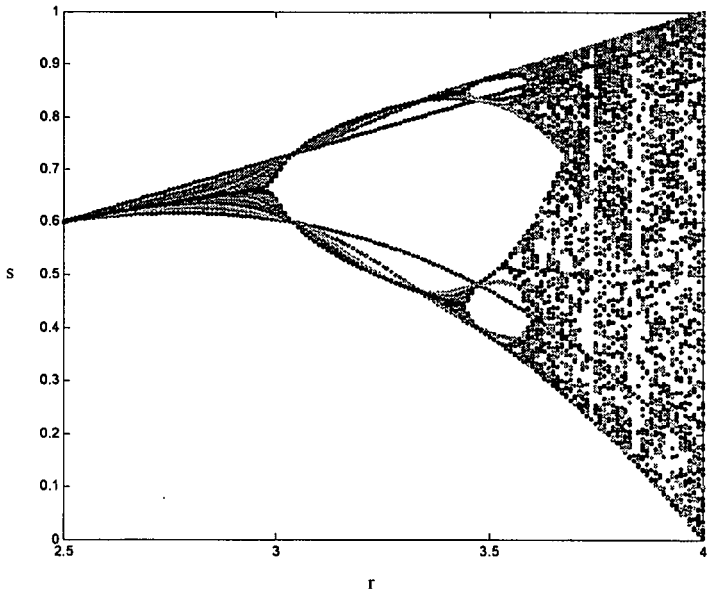


Figure 2.5: The logistic map from $r = 2.5$ to $r = 4.0$

There is two-dimensional discrete dynamical system such as the Hénon map [47]. The Hénon map is given by:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} -ax_n & 1 \\ \cdot b & 0 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.4.4}$$

where a and b are the constants. When $a = 1.4$ and $b = 0.3$, the Hénon map gives a two-dimensional strange attractor as shown in Fig.2.6.

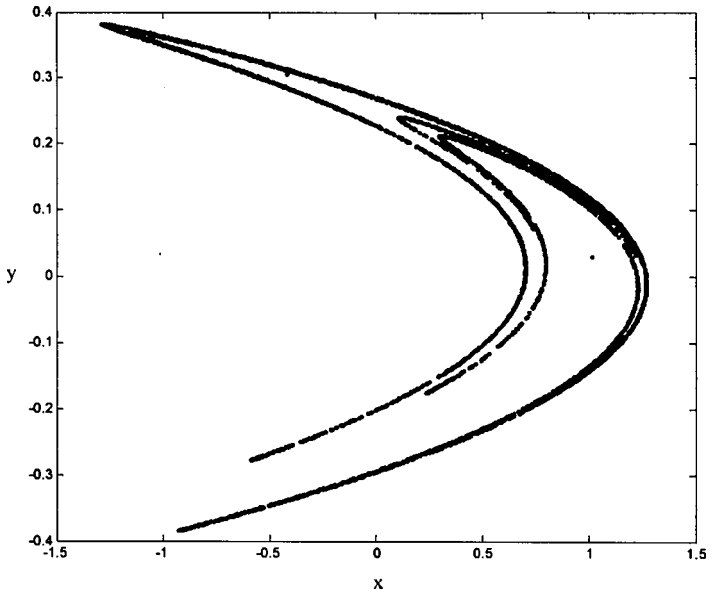


Figure 2.6: The Hénon map for $a = 1.4$ and $b = 0.3$

2.5 Phase space reconstruction

The essential problem in nonlinear time series analysis is to determine whether or not a given time series is a deterministic signal from a low-dimensional dynamical system. If it is, then further questions of interest are: What is the dimension of the phase space supporting the data set? Is the data set chaotic? The key to answering these questions is embodied in the method of phase space reconstruction, which has been rigorously justified by the embedding theorem of Takens [48].

2.5.1 Embedding theory

A fundamental idea in nonlinear analysis is that the dynamics of a system can be studied in a phase space [49]. A point in this space characterizes the state of the system at any moment of time. The phase space can be constructed by a procedure that starts from raw data and builds vectors by iteration of a time delay. Takens' embedding theorem asserts that if a time series is one component of an attractor that can be represented by a smooth d-dimensional manifold (with d an integer) then the properties of the attractor are equivalent to the properties of the embedding formed by the m-dimensional phase space vectors. Specifically, a scalar time series $x_n, n = 1, \ldots, N$ can be unfolded in a multidimensional phase space using time delay coordinates. The delay coordinate construction approach, based on Takens' theorem [48], is applied to a series of data:

$$X_i = [x_i, x_{i+\tau}, \ldots, x_{i+(m-1)\tau}] \tag{2.5.1}$$

where τ is called the delay time and m is the embedding dimension. Different choices of τ and m yield different reconstructed trajectories.

A reconstructed phase space matrix Y of dimension m and lag τ is called

a trajectory matrix and defined by:

$$Y = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_M \end{bmatrix} = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_M & x_{M+\tau} & \cdots & x_{M+(m-1)\tau} \end{bmatrix} \tag{2.5.2}$$

where the row vectors X_i, with $i = 1, \ldots, M$, represent individual points in the reconstructed phase space. The number of the points is $M = N - (m-1)\tau$.

2.5.2 Optimal delay time

A one-to-one embedding can be obtained for any value of the delay time $\tau > 0$. However, a very small delay time will result in near-linear reconstructions with high correlations between consecutive phase space points and very large delays might obscure the deterministic structure linking points along a single degree of freedom. If the delay time is commensurate with a characteristic time in the underlying dynamics, then this too may result in a distorted reconstruction.

Figure 2.7 shows the phase space reconstructions using six different time delays of $\tau = 1, 3, 5, 7, 9, 11$ for the y coordinate from the Lorenz equation (2.4.1).

From Fig.2.7(a), the delay time τ is too small, the dynamics is restricted to the diagonal of the embedding space. From Fig.2.7(b)-(c), the delay time τ leads to a well-utilized embedding space. For a larger delay time the characteristic structures tend to disappear. From Fig.2.7(d)-(f), the delay time τ is so large that causes the coordinates to disjoin by stretching and folding.

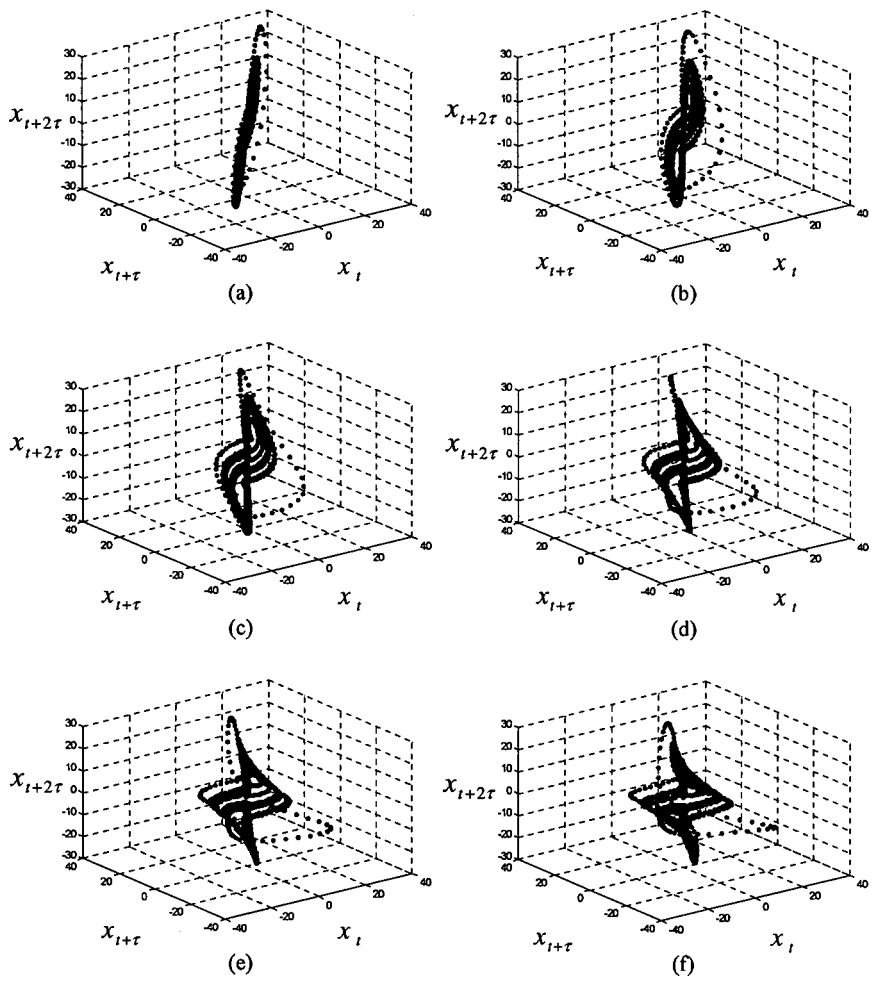


Figure 2.7: Phase space reconstruction of the Lorenz system using the y coordinate for six values of the delay time τ. (a)-(f) with embedding dimension three and $\tau = 1, 3, 5, 7, 9, 11$.

Autocorrelation function

There have been various proposals for choosing an optimal delay time based on the behavior of the autocorrelation function [50]. These include the earliest time τ at which the autocorrelation drops to a fraction of its initial value or has a point of inflection. These definitions seek to find times where linear correlations between different points in the time series are negligible, but they do not rule out the possibility of more general correlations. The autocorrelation function can be used for two purposes: detect non-randomness in data and identify an appropriate time series model if the data are not random.

Given measurements, $Y_1, Y_2, \ldots, Y_N$ at time $t_1, t_2, \ldots, t_N$, the lag k autocorrelation function is defined as:

$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2} \tag{2.5.3}$$

where $\bar{Y}$ is the mean value of Y_i. Instead of correlation between two different variables, the autocorrelation is the correlation which is between two values of the same variable at times t_i and t_{i+k}. Figure 2.8 shows the autocorrelation function for the signal $x(t)$ component of the Lorenz model with the parameters $(\sigma, \gamma, \beta) = (16, 45.92, 4)$.

If we used the first zero crossing of this autocorrelation function to select the time delay τ for time delay embedding, $\tau \simeq 40$.

Average mutual information

Fraser and Swinney [51] argue that a better value for τ is the value that corresponds to the first local minimum of the mutual information where the mutual information is a measure of how much information can be predicted about one time series points given full information about the other. The mutual information between measurement a_i drawn from a set $A = \{a_i\}$ and measurement b_j drawn from a set $B = \{b_j\}$ is the amount learned by the

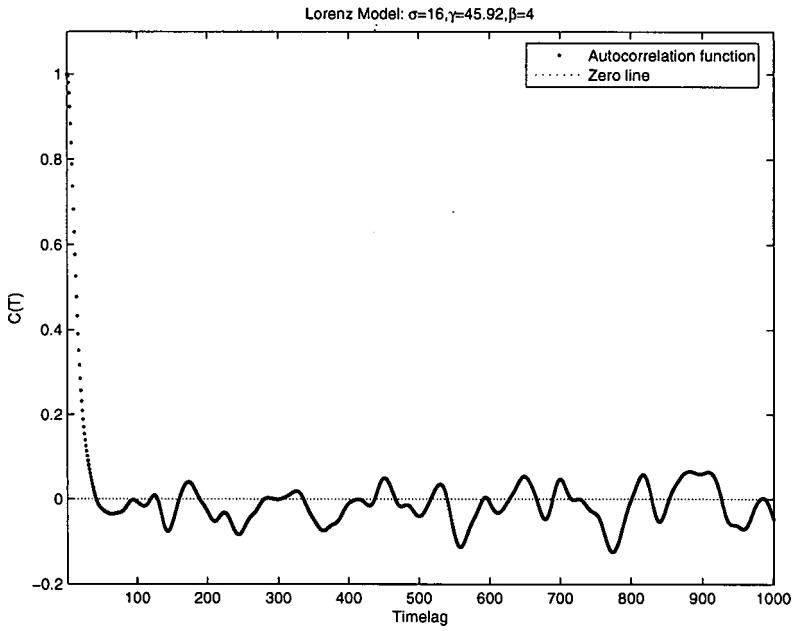


Figure 2.8: Autocorrelation function for the $x(t)$ component of the Lorenz model

measurement of a_i about the measurement of b_j. In *bits*, it is:

$$\log_2 \left[\frac{P_{AB}(a_i, b_j)}{P_A(a_i) P_B(b_j)} \right] \tag{2.5.4}$$

where $P_{AB}(a, b)$ is the joint probability density for measurements A and B resulting in values a and b. $P_A(a)$ and $P_B(b)$ are the individual probability densities for the measurements of A and of B.

If the measurement of a value from A resulting in a_i is completely independent of the measurement of a value from B resulting in b_j, then $P_{AB}(a, b)$ factorizes: $P_{AB}(a, b) = P_A(a) P_B(b)$ and the amount of information between the measurements, the mutual information, is zero, as it should be. The average over all measurements of this information statistic, called the average mutual information between A measurements and B measurements, is:

$$I_{AB} = \sum_{a_i, b_j} P_{AB}(a_i, b_j) \log_2 \left[\frac{P_{AB}(a_i, b_j)}{P_A(a_i) P_B(b_j)} \right] \tag{2.5.5}$$

Figure 2.9 shows the average mutual information $I(T)$ for the signal $x(t)$ component of the Lorenz model with the parameters $(\sigma, \gamma, \beta) = (16, 45.92, 4)$.

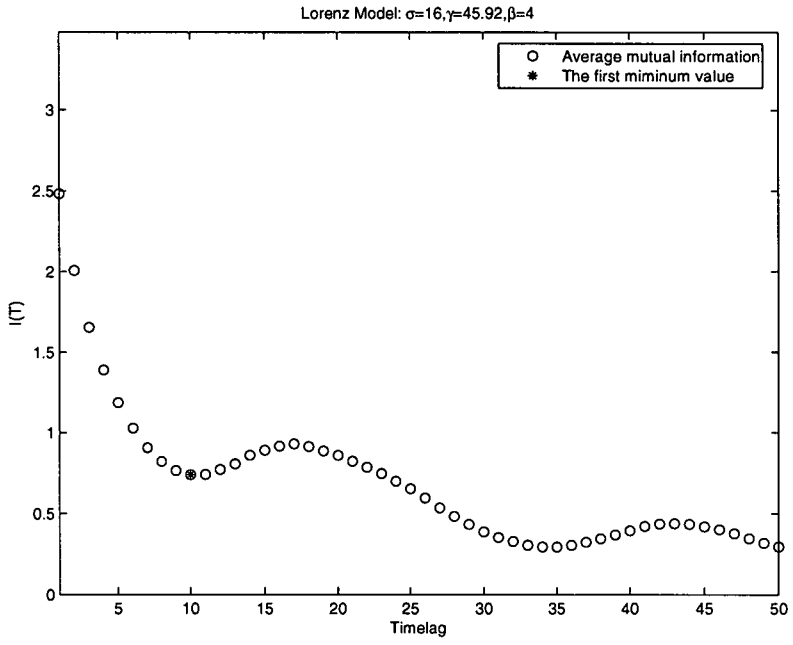


Figure 2.9: Average mutual information for the $x(t)$ component of the Lorenz model

The first minimum of this function is at $\tau = 10$.

2.5.3 Optimal embedding dimension

There have been many discussions on how to determine the optimal embedding dimension from a scalar time series based on Takens' theorem or its extensions [52, 53, 54]. The crucial problem is how to select a minimal embedding dimension for the phase space. If the embedding dimension is too small, one cannot unfold the geometry of the (possible strange) attractor, and if one uses an embedding dimension too high, most numerical methods characterizing the basic dynamical properties can produce unreliable or spurious results.

False nearest neighbor method

A method to determine the minimal sufficient embedding dimension m was proposed by Kennel et al. [52]. It is called the false nearest neighbor (FNN) method. The idea is quite intuitive. Suppose that the minimal embedding

dimension for a given time series x_i is m_0. This means that in a m_0-dimensional delay space the reconstructed attractor is a one-to-one image of the attractor in the original phase space. Thus the neighbors of a given point are mapped onto neighbors in the delay space. Due to the assumed smoothness of the dynamics, neighborhoods of the points are mapped onto neighborhoods again. Of course the shape and the diameter of the neighborhoods is changed. But suppose now you embed in an m-dimensional space with $m < m_0$. Points are projected into neighborhoods of other points to which they wouldn't belong to higher dimensions. These points are called false neighbors. If the dynamics is applied now, these false neighbors are not typically mapped into the image of the neighborhood, but somewhere else, so that the average "diameter" becomes quite large.

The idea of the false nearest neighbors method is the following. Each point $\vec{x}_i$ in the time series look for its nearest neighbor $\vec{x}_j$ in a m-dimensional space. Calculate the distance $\|\vec{x}_i - \vec{x}_j\|$. Iterate both points and compute using:

$$R_i = \frac{\|\vec{x}_{i+1} - \vec{x}_{j+1}\|}{\|\vec{x}_i - \vec{x}_j\|} \tag{2.5.6}$$

If R_i exceeds a given threshold R_t, this point is marked as having a false nearest neighbor. The criterion that the embedding dimension is high enough is that the fraction of points for which $R_i > R_t$ is zero, or at least sufficiently small. Figure 2.10 shows the percentage of false nearest neighbors for the dynamical variable $x(t)$ of the Lorenz model with the parameters $(\sigma, \gamma, \beta) = (16, 45.92, 4)$.

The percentage of false nearest neighbors goes to zero at $m = 3$, and as expected remains zero from there on. So $m = 3$ is the necessary integer dimension for unfolding the attractor from $x(t)$ data.

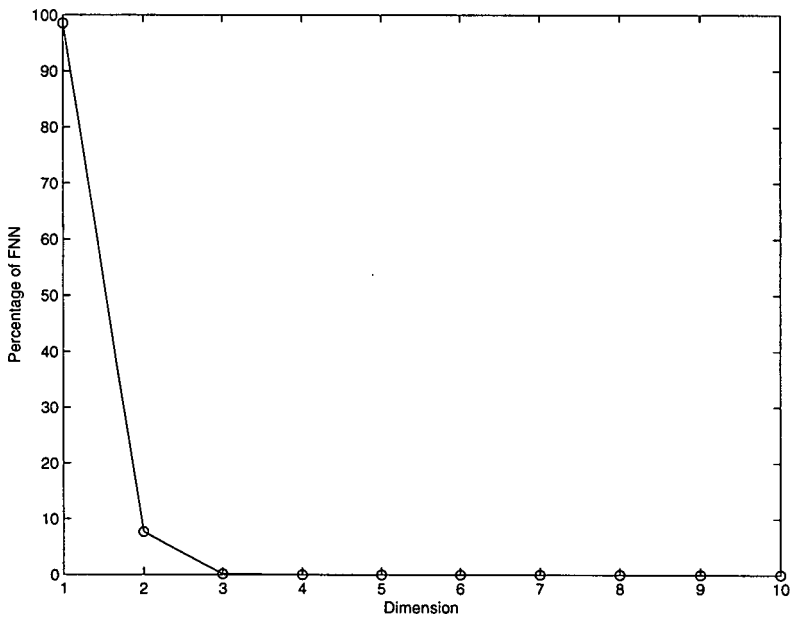


Figure 2.10: The percentage of false nearest neighbors for the dynamical variable $x(t)$ of the Lorenz model.

2.6 Invariant characteristics of dynamics

Classifying dynamical systems is a critical part of the analysis of measured signals. There are two major invariant characteristics of dynamics which have emerged as classifiers: fractal dimensions and Lyapunov exponents, respectively.

Fractal dimensions are characteristic of the geometric figure of the attractor and relate to the way that points on the attractor are distributed in a m-dimensional space. Lyapunov exponents tell how orbits on the attractor move apart (or together) under the evolution of the dynamics. Both are invariant under the evolution operator of the system, and thus are independent of changes in the initial conditions of the orbit, and both are independent of the coordinate system in which the attractor is observed. This means we can evaluate them reliably in the reconstructed phase space.

2.6.1 Fractal dimensions

A geometrical object can be fully represented by a set of points in a Euclidean space $\mathbb{E}^n$ provided that n is sufficiently large to be able to uniquely locate the position of each point in the object. Each set in $\mathbb{E}^n$ has a dimension d that is an integer in the range [0,n]. If the set is all of $\mathbb{E}^n$, then $d = n$. In Euclidean geometry, points have dimension $d = 0$, lines have dimension $d = 1$, plane surfaces have dimension $d = 2$, solid have dimension $d = 3$, etc.

A fractal dimension D is any dimension measurement that allows noninteger values [55]. A fractal is a set with a noninteger fractal dimension. Standard objects in Euclidean geometry are not fractals but have integer fractal dimensions $D = d$. The primary important of fractals in dynamics is that strange attractors are fractals and their fractal dimension D is simply related to the minimum number of dynamical variables needed to model the dynamics of the strange attractor.

Grassberger-Procaccia algorithm

The Grassberger-Procaccia algorithm [56, 57] is used for estimating the correlation dimension from a given set of points randomly distributed. It is based on the following approximation: The probability that two points of the set are in the same cell of size r is approximately equal to the probability that two points of the set are separated by a distance $|\cdot|$ less than or equal to r. Thus $C(r)$ is approximately given by:

$$C(r) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \theta(R - |x_i - x_j|) \qquad (i \neq j) \qquad (2.6.1)$$

where x_i and x_j are two arbitrary points of the phase space, and $C(r)$ is the standard correlation integral, and θ is the Heaviside function as defined by:

$$\theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \qquad (2.6.2)$$

The approximation in Eq.(2.6.1) is exact in the limit $N \to \infty$; however, this limit cannot be realized in practical applications. The limit $r \to 0$ used in the

definition of D_2 is also not possible in practice. instead, Procaccia and Grassberger propose the (approximate) evaluation of $C(r)$ over a range of values of r and then deduce D_2 from the slope of the straight line of best fit in the linear scaling region of a plot of $\log[C(r)]$ versus $\log[r]$.

The most common metric employed to measure the distance $|\cdot|$ in Eq.(2.6.1) is the Euclidean metric,

$$|x_i - x_j| = \sqrt{\sum_{k=1}^{m}(x_i(k) - x_j(k))^2} \tag{2.6.3}$$

We then assume that for small r, $C(r)$ behaves as follows:

$$C(r) \approx r^{D_c} \tag{2.6.4}$$

D_c is then called the *correlation dimension* and it can be estimated using:

$$D_c = \lim_{r \to 0} \frac{\log[C(r)]}{\log[r]} \tag{2.6.5}$$

where $C(r)$ is the correlation function of the attractor, i.e. a measure of the probability that two points on the attractor are separated by a distance of r. For small r, the function $\log[C(r)]$ is approximately linear in $\log[r]$ and thus, D_c is simply given by the slope of the log-log curve. For a system to be chaotic, D_c should be fractal and should converge with an increasing embedding dimension.

Figure 2.11 shows the curves $\log[C(r)]$ versus $\log[r]$ evaluated from each of the one variables $x(n)$ from the Lorenz system. The curves are shown using vectors in both $m = 3$ and $m = 4$.

Ten thousand data points were used and $\tau = 10$ selected by average mutual information was utilized in making time delay vectors for spaces of dimension $m = 3$ and $m = 4$. We already know from false nearest neighbors that this attractor is unfolded in $m = 3$, so $m = 4$ is a consistency check. The slope of this graph yields the correlation dimension D_2.

Figure 2.12 is a plot of

$$\frac{d\log[C(r)]}{d\log[r]} \tag{2.6.6}$$

versus $\log[r]$ for each of the data sets $x(n)$ from the Lorenz system.

We see consistency in a range of $\log(r)$ with a slope slightly around two.

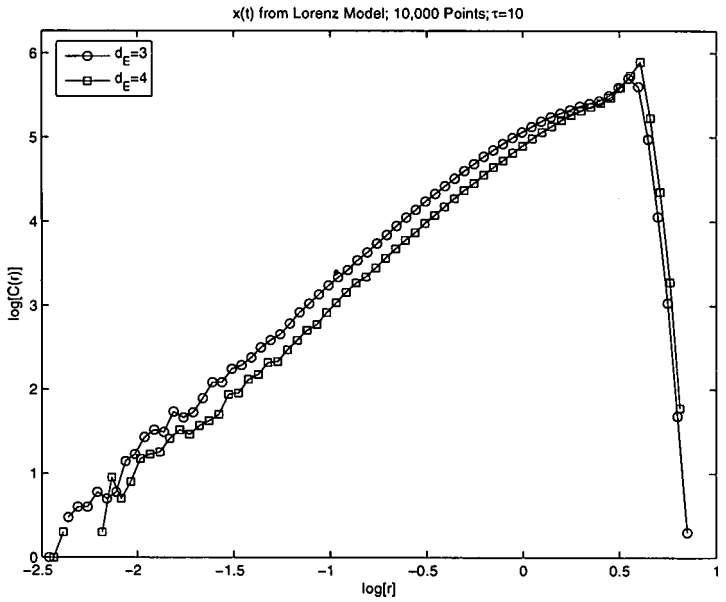


Figure 2.11: The correlation function $C(r)$ as a function of r for data from the $x(t)$ component of the Lorenz attractor.

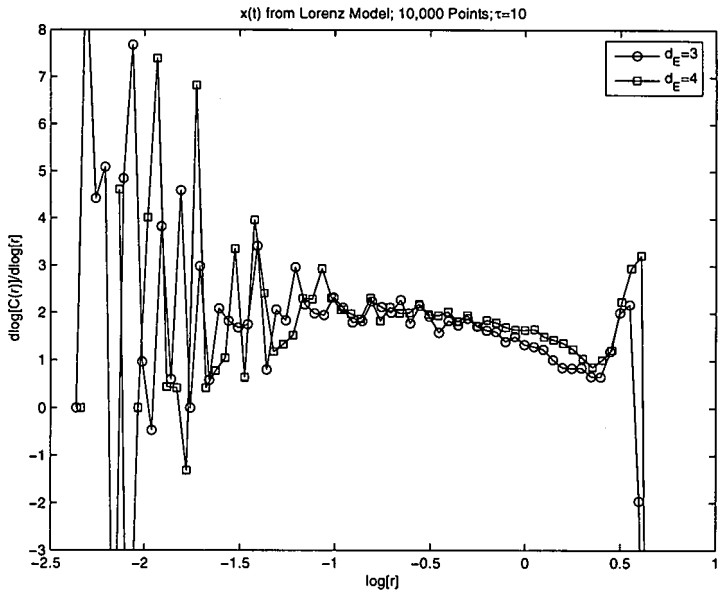


Figure 2.12: The derivative of the correlation function $\log[C(r)]$ created from $x(t)$ taken from the Lorenz model with respect to $\log(r)$.

2.6.2 Lyapunov exponents

Lyapunov exponents quantify the average exponential separation between nearby phase space trajectories. An exponential divergence of initial nearby trajectories in the phase space coupled with folding of trajectories is the generic mechanism for generating deterministic randomness and unpredictability. Indeed, the existence of a positive Lyapunov exponent for almost all initial conditions in a bounded dynamical system is a widely used definition of deterministic chaos.

Calculating Lyapunov exponents from a time series

The first algorithm for calculating Lyapunov exponents from a time series was proposed independently in 1985 [58]. The first step in these method is to construct an appropriate embedding of the experimental time series using the method of time delays described in Section 2.5. The maximum Lyapunov exponent can now be calculated as follows [59, 60, 61, 62]. Choose a reference point labeled $\vec{X}(0)$ and the "closest" neighboring point labeled $\vec{X}^{(1)}(0)$ from the set of reconstructed phase space vectors and calculate

$$\|\Delta\vec{X}_0(0)\| = \|\vec{X}(0) - \vec{X}^{(1)}(0)\| \tag{2.6.7}$$

Evolve the two points $\vec{X}(0)$ and $\vec{X}^{(1)}(0)$ forward in the reconstructed phase space for a time T_1 and calculate the new separation distance

$$\|\Delta\vec{X}(T_1)\| = \|\vec{X}(T_1) - \vec{X}^{(1)}(T_1)\| \tag{2.6.8}$$

An approximate renormalization is now performed by finding a point $\vec{X}^{(2)}(0)$ that satisfies the dual requirement that (1) $\vec{X}^{(2)}(0)$ is a neighboring point to $\vec{X}(T_1)$ and (2)

$$\Delta\vec{X}_0(T_1) = \vec{X}(T_1) - \vec{X}^{(2)}(0) \tag{2.6.9}$$

and $\Delta\vec{X}(T_1)$ are in approximately the same direction. The two points $\vec{X}(T_1)$ and $\vec{X}^{(2)}(0)$ are now evolved for a time T_2 in the reconstructed phase space to calculate

$$\|\Delta\vec{X}(T_1+T_2)\| = \|\vec{X}(T_1+T_2) - \vec{X}^{(2)}(T_2)\| \tag{2.6.10}$$

The renormalization process of finding a neighboring point to the current point that has a similar orientation to the previous replacement point is repeated N times and then the maximum Lyapunov exponent is calculated as:

$$\lambda = \frac{1}{\sum_{k=1}^{N} T_k} \sum_{k=1}^{N} \log \frac{\|\Delta \vec{X}(\sum_{j=1}^{k} T_j)\|}{\|\Delta \vec{X}_0(T_k)\|} \tag{2.6.11}$$

This calculation should then be averaged over several different initial starting points. In implementing the method, it is important to not accept as the "closest" neighboring point that is temporally separated by a distance less than the delay time τ. This is to avoid choosing adjacent points on the same trajectory. Thus the times $T_1, \ldots, T_N$ should be greater than τ. On the other hand, these times should be small enough to obtain exponential separations.

Figure 2.13 shows the illustration diagram about the Lyapunov exponent calculation from experimental data [60].

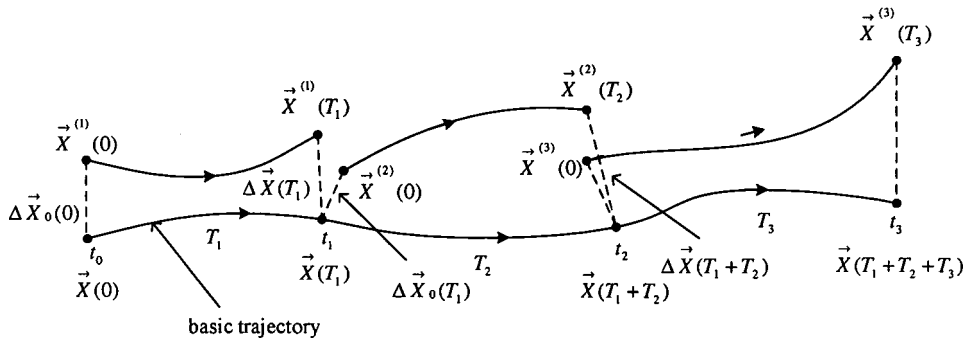


Figure 2.13: A schematic representation of the evolution and replacement procedure used to estimate Lyapunov exponents from experimental data.

The largest Lyapunov exponent is computed from the growth of length elements. When the length of the vector between two points becomes large, a new point is chosen near to the reference trajectory.

2.7 Summary

This chapter gives an introduction to chaos theory including theoretical background and mathematical definitions. The chaos theory is a theory describing erratic behavior in certain nonlinear dynamical systems. In mathematics and physics, chaos theory describes the behavior of certain nonlinear dynamical systems that under certain conditions exhibit a phenomenon known as chaos. In other word, chaos theory is a kind of method which is used to find out the invariable characteristics from the random signal for control and prediction. The embedding theory is used to transform the signal from time domain into phase space domain where the attractor of the signal is constructed. Time delay and embedding dimension are two parameters that influence the phase space reconstruction. The optimal choice for both of them can easily and correctly explore the dynamical features of signals in phase space. The fractal dimensions and lyapunov exponent are two invariant characteristics of the signal attractor which can help us determine, control and predict the changing trend of a signal. The research results have demonstrated that this technique can be used as effective means of digital signal processing.

The methods and concepts introduced in this chapter give a new way for signal processing. In the following three chapters, three different procedures are developed for noise reduction, speech signal processing and electromyography signal processing which are based on the basic theory introduced in this chapter.

Chapter 3

A Noise Reduction Procedure In Phase Space

The overall objective of noise reduction is to find the clear original signal that is consistent with the data. In this chapter, a phase space reconstruction based procedure is proposed for restoration of signals which are contaminated by noise. The procedure embeds the noisy signal into a high-dimensional reconstructed phase space and follows three steps. The first step is phase space reconstruction in which Takens' Theorem is applied to map the signal from time domain into a high-dimensional phase space domain. The second step is phase space projection which includes local singular value decomposition (LSVD) and global singular value decomposition (GSVD) for noise elimination in phase space. The third step is signal inverse transformation and a time-aligned average method is used to transform the signal from the phase space domain back to the time domain. The procedure has been applied to a sinusoidal signal with added random noise and the experimental results show that the noise can be eliminated and the SNR is increased to an impressive high value.

3.1 Introduction

Different methods have been developed for optimal separation of the wanted signals from noise, which include linear signal processing techniques such as Wiener filtering and Kalman filtering as well as nonlinear methods such as manifold decomposition and phase space projection [63]. Every method of noise reduction assumes that it is possible to distinguish between noise and a clean signal on the basis of some objective criteria. Conventional methods such as linear filters use power spectrums for this purpose. Low pass filters assume that a clean signal has some typical low frequencies, and also it is true for high pass filters. It follows that these methods are convenient for a periodic or a quasi-periodic signal. In the case of chaotic signals, linear filters cannot be used for noise reduction for the signals without a substantial disturbance to the clear signal because chaotic signals have a broad-band spectrum which overlaps noise and the clean signal. New methods of noise reduction have recently been developed and they are based on the theory of nonlinear dynamical systems [64, 65, 66, 67].

Experimental time series are a mixture of deterministic component and random noise. A method for separation of the noise contribution directly in the phase space is proposed. In fact, the philosophy for the method is that it is not necessary to always consider solving the problem only arising from separating a deterministic signal from random fluctuations. It is possible that the "noise" may arise from a high dimension, deterministic dynamical system. The noise reduction problem, therefore, is a question of how to separate the low-dimensional dynamics from a complex signal. A relationship is derived between phase space projection methods commonly used in nonlinear dynamical systems and subspace decomposition methods commonly used in linear signal processing. From this relationship, it has been shown that phase space techniques such as local singular value decomposition (LSVD) and global singular value decomposition (GSVD) can remove noise in phase space and preserve the clean signal.

The flowchart of the procedure is shown in the Fig.3.1:

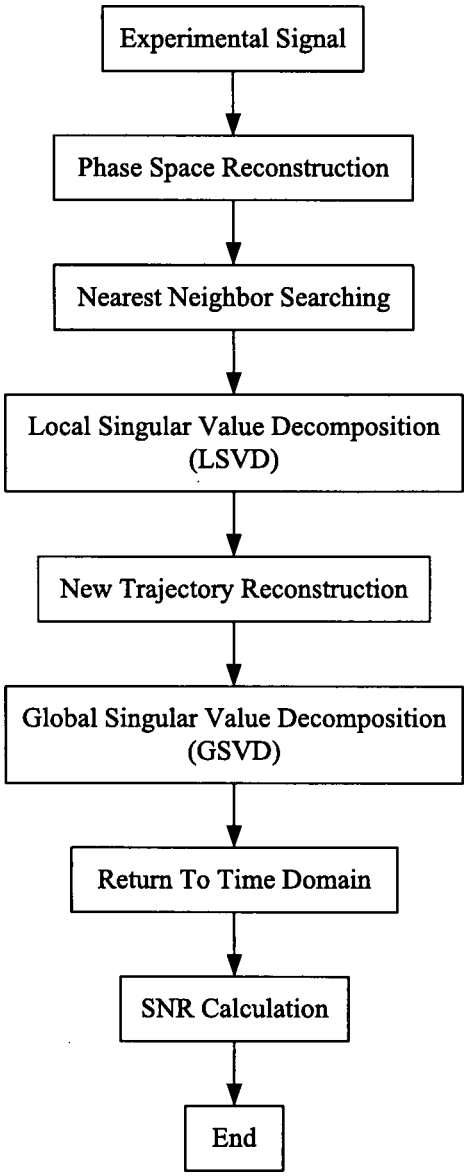


Figure 3.1: The flowchart of the nonlinear noise reduction (NNR) procedure

As can be seen, the procedure consists of seven steps:

(1) Phase Space Reconstruction: attractor reconstruction from the noise-corrupted time series, mapping the data into phase space;

(2) Nearest Neighbor Searching: searching the nearest neighbor points within the fixed radius for each point in the phase space;

(3) Local Singular Value Decomposition (LSVD): the LSVD method is used to project the points onto the primary subspace;

(4) New Trajectory Reconstruction: the median point of the nearest neighbor points is selected as the output in the phase space.

(5) Global Singular Value Decomposition (GSVD): the GSVD method is used to project the all tractory points onto the primary and secondary subspaces in the phase space;

(6) Transform Back To Time Domain: the time-aligned median method is used to transform the signal back from phase space domain to time domain.

(7) SNR Calculation: the Signal-Noise-Ratio (SNR) value is calculated.

3.2 Description of noise reduction procedure

The main feature of the procedure is to reconstruct the attractor of the contaminated signal in the phase space for noise elimination. The signal is projected onto a subspace spanned by eigenvectors associated with the higher eigenvalues. The noise can be reduced by projecting the trajectory points onto the subspace spanned by a suitable collection of singular vectors on the attractor. The combination of local and global nonlinear filters, LSVD and GSVD, is applied. The local filter is used for each point on the trajectory within a small region (neighborhoods) of the attractor and the global filter is used to all the points along the attractor. At last, the signal is transformed back to the original time domain which is the reversed procedure of the phase space reconstruction.

3.2.1 Signal composition

To demonstrate the development of the procedure, a periodic sinusoidal signal contaminated by noise is chosen as the initial studying. The noise is a random process of which the elements are uniformly distributed in the interval (-0.03, +0.03). Fig.3.2 shows the original clear signal, the noise and the composite signal.

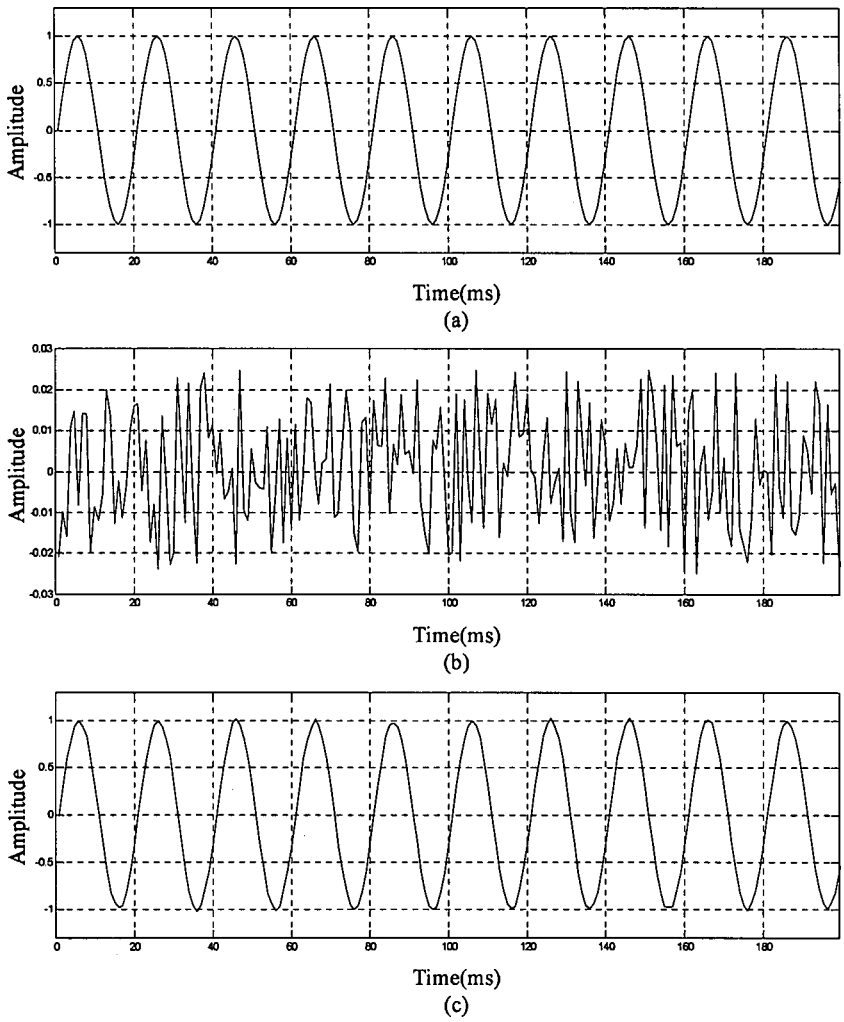


Figure 3.2: Signal simulation. (a) Original signal; (b) Noise; (c) Composite signal.

3.2.2 Phase space reconstruction

The embedding method introduced in section 2.5.1 is adopted. In this procedure, the embedding dimension m is set as 3 and the time delay τ is set as 1. Figure 3.3 shows the phase space reconstruction in three dimensional phase space for both clear and composite signal.

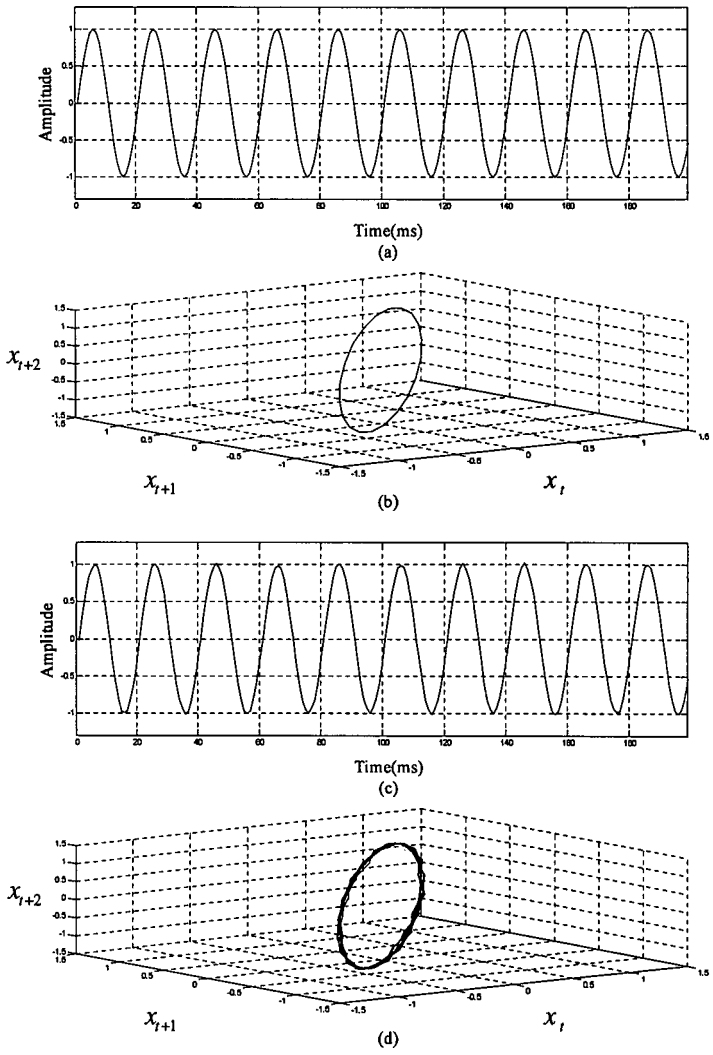


Figure 3.3: Phase space reconstruction. (a) Original signal; (b) Phase space reconstruction of (a), $\tau = 1, m = 3$; (c) Composite signal; (d) Phase space reconstruction of (c), $\tau = 1, m = 3$.

From the results we can see that the trajectory points of the clear sinusoidal signal in phase space are formed into a single ellipse. But for the contaminated signal, the shape of the trajectory points in phase space is very different. So the phase space reconstruction can make the difference between clear and contaminated signals and this difference is difficult to be detected in time domain but more obviously in phase space.

3.2.3 Nearest neighbor searching

A neighborhood is a local subspace of attractor global phase space. By taking a reference point x_{ref} on any trajectory, the neighborhood is defined as the n-dimensional subspace that includes the number of points nearest to x_{ref} within a neighborhood radius r. If the first reference point is x_0, its n nearest points form the neighborhood will be:

$$U = \{x_1, \ldots, x_n| : |x_i - x_0| \leq r, i = 1, 2, \ldots, n\} \tag{3.2.1}$$

where n is the maximum number of points in each neighborhood.

In the task of range searching, after the reference point x_{ref} is selected, all points in the data set X that have the distance r or smaller than r from x_{ref} are chosen as the nearest neighborhood points. Sometimes range searching is called a fixed size approach. Fig.3.4 illustrates how the neighbor points are searched for the attractor in a two dimensional space.

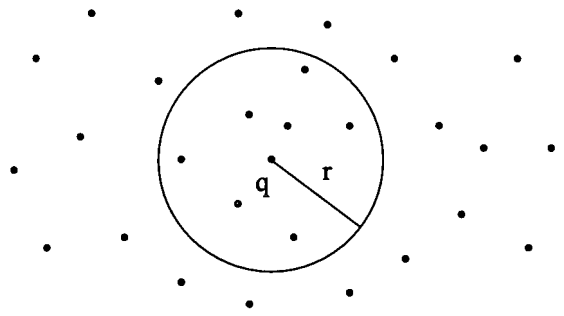


Figure 3.4: The Range Searching Map

3.2.4 Subspace projection for noise reduction

Phase space projection method for signal noise reduction has been proposed in recent years [68, 69, 70]. The main idea behind the method can be described as follows: Firstly, noisy signal is mapped onto an appropriate phase space where the signal should form an attractor. Because of the existence of noise, the orbits of contaminated signal in phase space will depart from the original attractor. Shadowing theorem [71] says that an accurate trajectory always exists along with an inaccurate trajectory. Then singular value decomposition (SVD) method is used to adjust the orbit in phase space and let the orbit be more close to the true attractor's orbit [72, 73]. From the results of Fig.3.3, it can be seen that the trajectory of contaminated signal deviated from the trajectory of the clean signal. So if the trajectory of the contaminated signal can be adjusted closely to the trajectory of the clean signal, the noise should be eliminated through this procedure.

Local singular value decomposition (LSVD)

Singular value decomposition (SVD) is the widely-used multivariate statistical technique in digital signal processing. The purpose of singular value decomposition is to reduce the size of a dataset containing a large number of values to a dataset containing significantly fewer data numbers. The reduced dataset still contains the variability features presented in the original data. One application of the SVD method is the subspace projection for noise reduction in the phase space. The subspace projection of a given signal data matrix contains information about the signal energy and the noise sources. By using a subspace projection, it is thus possible to divide approximately the observed noisy data into the subspaces of the original signal and additional noise. A summary of the noise reduction technique using the subspace projection is given in the following part and the corresponding Matlab code can be found in Appendix E.1.

Let X be the available data in the form of an $L \times M$ matrix:

$$X = [X_1, X_2, \ldots, X_M] \tag{3.2.2}$$

where the column vector $X_i (i = 1, 2, \ldots, M)$ is written as:

$$X_i = [x_i(0), x_i(1), \ldots, x_i(L-1)]^T \qquad (T : transpose). \tag{3.2.3}$$

Then, the singular value decomposition (SVD) of the autocorrelation matrix of X (for $M \leq L$) is given by:

$$X^T X = V \Lambda V^T \tag{3.2.4}$$

where the matrix $V = [v_1, v_2, \ldots, v_M]$ is orthogonal such that $V^T V = I_M$ and $\Lambda = diag(\sigma_1, \sigma_2, \ldots, \sigma_M)$, with eigenvalues $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0$. The columns in V are the eigenvectors of $X^T X$. The eigenvalues in Λ contain some information about the number of signals, signal energy, and the noise level. It is well known that if the signal-to-noise ratio (SNR) is sufficiently high, the eigenvalues can be arranged in an order of the following:

$$\sigma_1 > \sigma_2 > \cdots > \sigma_s \gg \sigma_{s+1} > \sigma_{s+2} \cdots > \sigma_M \tag{3.2.5}$$

and the autocorrelation matrix $X^T X$ can be decomposed as:

$$X^T X = \begin{bmatrix} V_s & V_{M-s} \end{bmatrix} \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_{M-s} \end{bmatrix} \begin{bmatrix} V_s & V_{M-s} \end{bmatrix}^T \tag{3.2.6}$$

where Λ_s contains the s largest eigenvalues associated with s signal fractions with the highest energy (i.e., $\sigma_1, \sigma_2, \ldots, \sigma_s$) and Λ_{M-s} contains $(M-s)$ eigenvalues $\sigma_{s+1}, \sigma_{s+2}, \ldots, \sigma_M$. It is then considered that V_s contains s eigenvectors associated with the main signal energy, whereas V_{M-s} contains $(M-s)$ eigenvectors associated with the minor signal energy. The subspace spanned by the columns of V_s is thus referred to as the signal principal subspace, whereas that spanned by the columns of V_{M-s} corresponds to the signal secondary subspace.

Then, the signal primary and secondary subspaces are mutually orthogonal and orthogonally projecting the observed noisy data onto the signal primary subspace leads to noise reduction. The data matrix after the noise reduction is written as $Y = [Y_1, Y_2, \ldots, Y_M]^T$, where $Y_i = [y_i(0), y_i(1), \ldots, y_i(L-1)]$, is given by:

$$Y = X V_s V_s^T \tag{3.2.7}$$

which describes the orthogonal projection onto the signal primary space.

Neighborhood projection

For each point on the trajectory, there must be a neighborhood and the SVD method is used for each point until the entire attractor is covered. Fig.3.5 shows the whole procedure of trajectory projection by using the local SVD method in phase space.

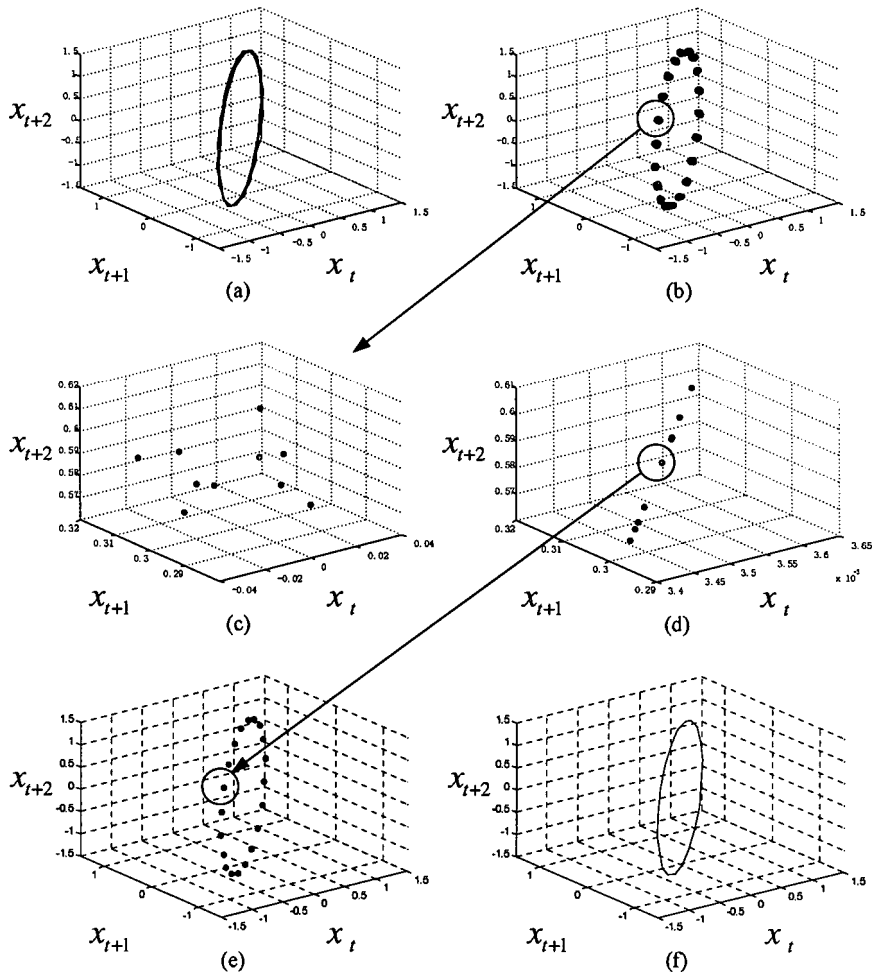


Figure 3.5: Local singular value decomposition (LSVD). (a) Signal trajectory in phase space, $\tau = 1, m = 3$; (b) Signal trajectory points in phase space; (c) A neighborhood for a reference point; (d) Result of neighborhood projection using LSVD method; (e) New signal trajectory points in phase space; (f) New signal trajectory in phase space.

In this procedure, we just take the first eigenvector in V_s which is associated with largest eigenvalue in Λ_s, by using (3.2.7) to project noisy signal onto the signal primary subspace within a neighborhood. The result is that the random points in the neighborhood will form a straight line in a three dimensional phase space. Fig.3.5(d) shows the result of neighborhood projection using LSVD method. Then the median point of this space line is selected as the output point for each point in this neighborhood. Fig.3.5(e) shows the result of new trajectory points.

Global singular value decomposition (GSVD)

After the local SVD method is applied, the noise still can not be eliminated entirely. By comparing with the trajectory of the clear signal in phase space, the points in the three dimensional phase space are located on the same plane. To clean noise from the signal further, the global SVD method is used and this time all the points in the phase space are projected onto the primary and secondary subspace so that all the points are on the same plane in the phase space. Fig.3.6(b) shows the result of trajectory of de-noised signal after local SVD method from an angle. Fig.3.6(d) shows the result of trajectory of de-noised signal after global SVD method from the same angle as Fig.3.6(b).

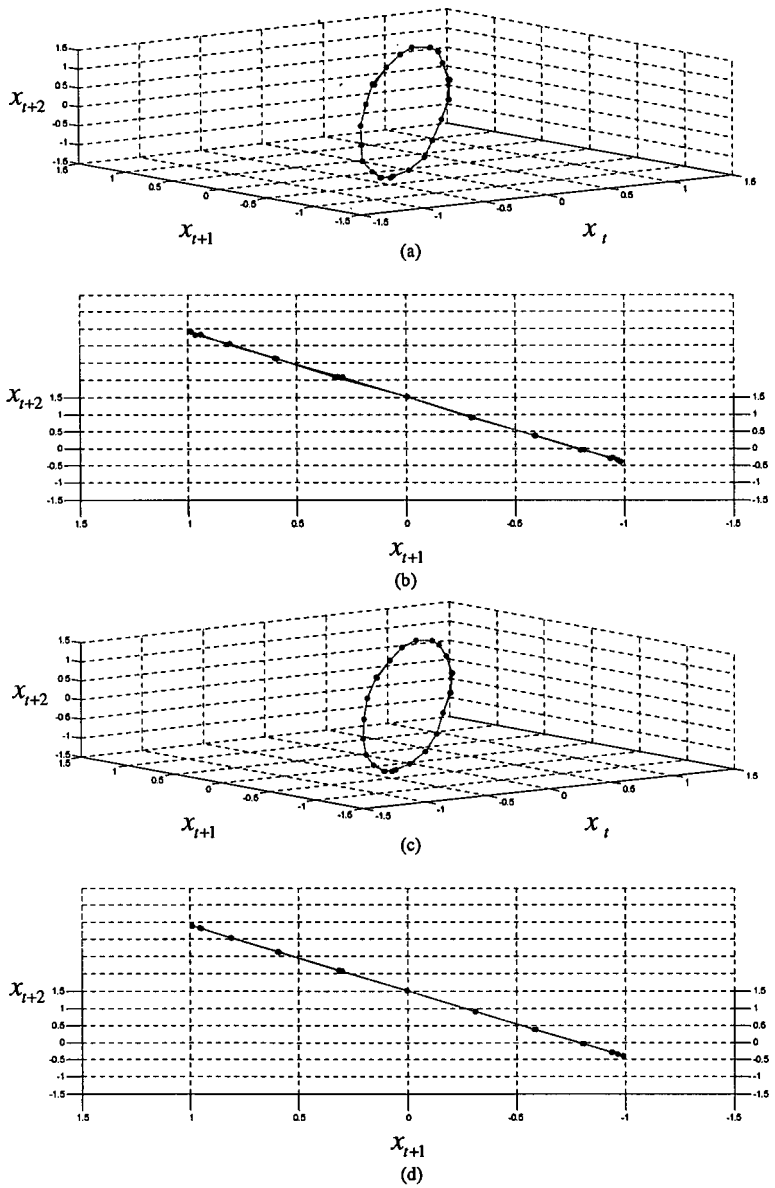


Figure 3.6: Global singular value decomposition (GSVD). (a) Trajectory of de-noised signal after LSVD method; (b) Trajectory of de-noised signal after LSVD method from an angle; (c) Trajectory of de-noised signal after GSVD method; (d) Trajectory of de-noised signal after GSVD method from the same angle as (b).

3.2.5 Transform back from phase space to time domain

Once transformed, the new trajectory matrix Y no longer corresponds to a time-delay embedding in a single time series, and there is no unique mapping back to the one-dimensional signal. Each column of the transformed matrix represents a possible enhanced signal output. There are several methods for creating an enhanced time-series from the new trajectory matrix, including selecting a single column from it or performing a time-aligned averaging of the columns.

In this procedure, a time-aligned median method is used, in which the median value is selected for each column of the matrix as the result for output [63]. To illustrate this process, an example trajectory matrix Y and aligned trajectory matrix Y_{aligned} are shown below for the case of $\tau = 1, m = 3$, with 50 points in the time-series.

$$Y^T = \begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & 45 & 46 & 47 & 48 \\ 2 & 3 & 4 & 5 & \cdots & 46 & 47 & 48 & 49 \\ 3 & 4 & 5 & 6 & \cdots & 47 & 48 & 49 & 50 \end{bmatrix} \tag{3.2.8}$$

$$Y^T_{\text{aligned}} = \begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & 47 & 48 & & \\ & 2 & 3 & 4 & \cdots & 47 & 48 & 49 & \\ & & 3 & 4 & \cdots & 47 & 48 & 49 & 50 \end{bmatrix} \tag{3.2.9}$$

The resulting output time-series is given by:

$$y(i) = \text{median}\{Y_{\text{aligned}}(i)\} \quad i = 1, 2, \ldots, 50 \tag{3.2.10}$$

where y is the time series transformed back from phase space, $Y_{\text{aligned}}(i)$ is the ith column from Y_{aligned}. Fig.3.7 shows the result of time-aligned median method.

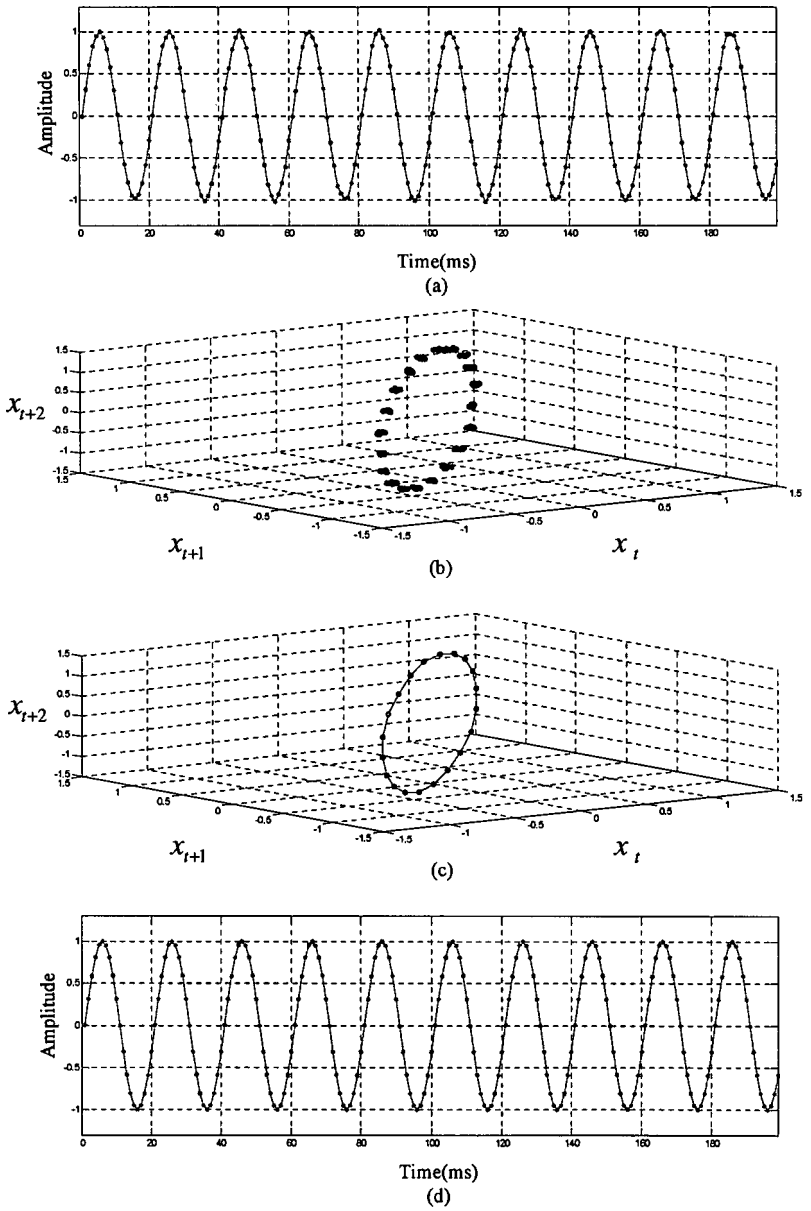


Figure 3.7: Signal return to time domain. (a) Composite signal in time domain; (b) Trajectory of composite signal in phase space; (c) Result of local SVD and globe SVD methods; (d) Result of de-noised signal transforming back to time domain.

3.2.6 SNR calculation

As a measure of the noise-reduction performance, the signal-to-noise ratio (SNR) is used. SNR is an engineering term for the power ratio between a signal (meaningful information) and the background noise as:

$$SNR = \frac{P_{\text{signal}}}{P_{\text{noise}}} \tag{3.2.11}$$

Because many signals have a very wide dynamic range, SNRs are usually expressed in terms of the logarithmic decibel scale. In decibels, the SNR is 20 times the base-10 logarithm of the amplitude ratio, or 10 times the logarithm of the power ratio:

$$SNR(dB) = 10\log(\frac{P_{\text{signal}}}{P_{\text{noise}}}) = 20\log(\frac{A_{\text{signal}}}{A_{\text{noise}}}) \tag{3.2.12}$$

where P is average power and A is RMS amplitude. RMS is the square root of the mean of the squares of the values. The RMS for a collection of N values $\{x_1, x_2, \ldots, x_N\}$ is:

$$x_{\text{rms}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{3.2.13}$$

In this procedure, the SNR is defined as:

$$SNR = 10\log\left(\frac{\sum x_n^2}{\sum (x_n - \hat{x}_n)^2}\right) \tag{3.2.14}$$

where x is the time series of clean signal and $\hat{x}_n$ is the time series of a signal after noise reduction.

By calculation, the SNR value is 33.405 for the composite signal before any signal processing has been applied and it will increase to 43.117 after local SVD method and it will increase to 45.914 finally after global method.

3.3 Noise reduction procedure for higher dimensional phase space

The application of noise reduction procedure in three dimensional phase space has been discussed in previous sections. It can also be used in a higher dimensional phase space. Figure 3.8 shows the SNR results for the composite signal which is embedded into a higher dimensional phase space from 4 to 10, respectively, and the time delay $\tau = 1$. The both local and global SVD methods are used in three dimensional phase space. In the global SVD method, the signal is also projected onto primary and secondary subspaces.

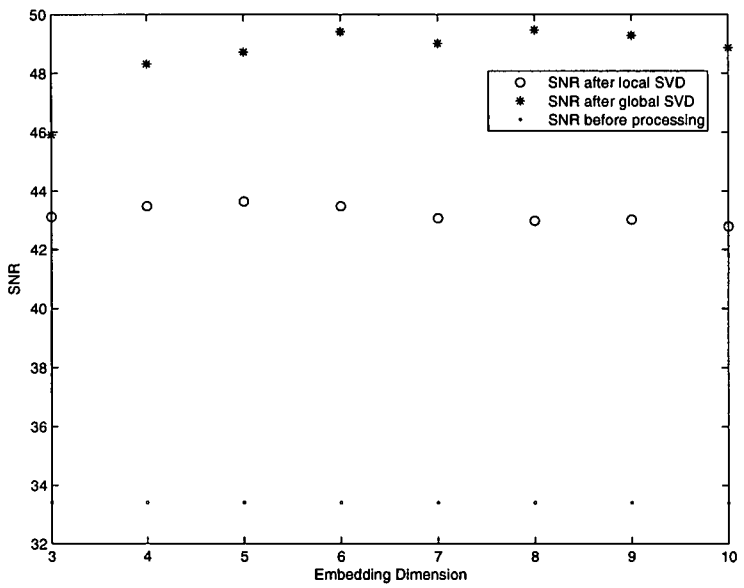


Figure 3.8: SNR results in higher dimensional phase space, $\tau = 1$

From the Fig.3.8, we can see that the different embedding dimensions can influence the SNR value. And another parameter in phase space reconstruction, the time delay τ, can also play an important role in the SNR calculation. Fig.3.9 and Fig.3.10 show the SNR results for the composite signal which is embedded into a higher dimensional phase space from 4 to 10 and the time delay $\tau = 2$ and $\tau = 3$, respectively.

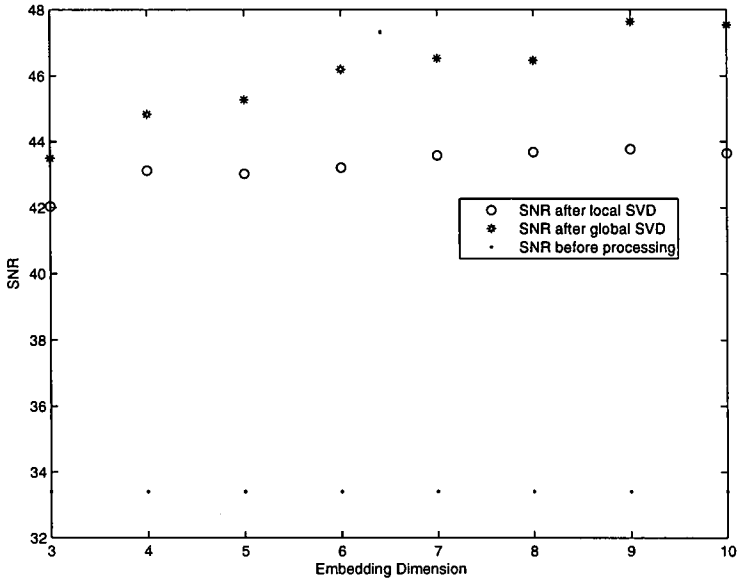


Figure 3.9: SNR results in higher dimensional phase space,$\tau = 2$

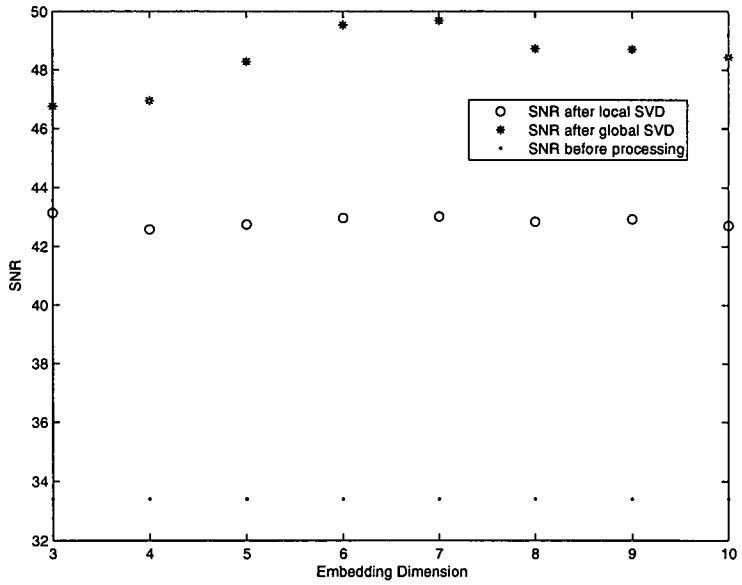


Figure 3.10: SNR results in higher dimensional phase space,$\tau = 3$

Table 3.1 and Table 3.2 show SNR results of local and global SVD processing for the composite signal which is embedded into a higher dimensional phase space with embedding dimension m from 4 to 10 and time delay τ from 1 to 10, in a tabular way.

Table 3.1: SNR values after local SVD

SNR	Embedding Dimensions							
Time Delay	3	4	5	6	7	8	9	10
1	43.12	43.48	43.64	43.48	43.07	42.98	43.02	42.79
2	42.03	43.13	43.03	43.22	43.59	43.69	43.78	43.65
3	43.14	42.58	42.75	42.97	43.02	42.85	42.93	42.71
4	41.50	41.68	42.47	43.07	**44.18**	44.10	43.59	43.00
5	42.04	42.84	42.66	42.69	43.15	42.35	41.78	41.42
6	43.66	43.19	43.72	43.15	42.92	42.38	42.02	41.62
7	42.31	42.60	42.69	42.47	42.09	41.77	41.72	41.28
8	41.52	41.84	41.98	41.92	42.09	41.10	40.76	40.53
9	41.79	42.02	41.73	41.61	41.16	40.84	40.50	40.32
10	34.61	37.64	40.17	41.19	41.50	41.55	41.18	41.13

From the table, we can see that the highest SNR value after local SVD processing is 44.18 when $m = 7$ and $\tau = 4$.

Table 3.2: SNR values after global SVD

SNR	Embedding Dimensions							
Time Delay	3	4	5	6	7	8	9	10
1	45.91	48.32	48.72	49.41	49.01	49.47	49.28	48.86
2	43.49	44.84	45.28	46.20	46.53	46.46	47.64	47.53
3	46.76	46.95	48.28	49.53	**49.68**	48.72	48.71	48.43
4	41.81	42.77	43.65	44.29	46.22	46.16	45.10	44.82
5	42.91	44.13	44.62	44.20	44.91	44.13	43.71	43.24
6	44.61	45.36	46.82	46.33	46.10	45.55	45.25	45.05
7	44.38	45.32	46.25	46.31	46.09	46.02	46.53	46.88
8	41.91	42.92	43.23	43.39	43.72	42.46	42.13	42.12
9	45.02	46.50	47.07	46.60	46.74	45.72	45.90	45.86
10	34.59	37.76	40.16	41.49	41.58	41.62	41.33	41.24

From the table, we can see that the highest SNR value after global SVD processing is 49.68 when $m = 7$ and $\tau = 3$ which is higher than the result of local SVD processing.

3.4 Summary

In this chapter, phase space reconstruction based method is proposed to address the problems of noise reduction in signal processing. A noise reduction procedure is developed from the inspiration of chaos theory. The procedure can be summarized into three main steps: Phase space reconstruction, which is used to map the signal from time domain into phase space domain; Phase space projection, which is used to get rid of noise components from the signal addressed in the phase space; Time domain transformation, which is used to transform signal back to time domain. The proposed procedure is an initial studying which is tested using a sinusoidal signal contaminated by noise. Results show that the procedure is effective for noise reduction in phase space and it is an important alternative way for signal noise reduction.

The noise reduction procedure proposed can also be applied to the higher dimensional phase space with different time delay to get the better result.

This noise reduction procedure requires no prior dynamical information for its implementation and no general properties for the time series. The procedure is purely geometric. The chaotic techniques and conceptions which are used in this procedure can also be applied to other types of signals, such as the speech signal and electromyography (EMG) signal.

Chapter 4

Speech Signal Restoration Using Phase Space Reconstruction

In this chapter, a phase space reconstruction procedure is proposed for restoration of speech signals contaminated by noise. The procedure embeds the noisy signal into a high-dimensional reconstructed phase space which consists of two steps. The first step is Singular Value Decomposition (SVD) which identifies the distorted segments of the signal and the second step is Space Interpolation to recover the missing information of the distorted signal segments. Then, the signal is transformed back from phase space to time domain. The experimental results show that SVD method can identify the distorted segments successfully and the recovered speech signal is obviously more speech recognizable. The procedure described in the chapter is also suitable for real-time implementation which may be used to improve the sound quality of microphones.

4.1 Introduction

With the development of communication technology, it demands to have reliable transmission and reproduction of audio and speech signals. However, in many cases, the speech signals may be partially corrupted or contaminated. This is mainly because the transmission channels often introduce additive noise

to the signals transmitted. Sometimes, the recorded speech signal is already distorted before they were transmitted, for example, conversations recorded in a noisy environment. It is very important to separate the clear speech signal from the background noise and improve its quality before further processing applied.

There are some popular methods for speech signal enhancement and noise reduction, such as Ephraim-Malah minimum mean-square error (E-M MMSE) [74], log spectral amplitude (LSA) estimation [75], spectral subtraction [76] and beta-order MMSE [77]. One of the main approaches of these algorithms is to obtain the best possible estimates of the short time spectra amplitude (STSA) of a speech signal in a noisy environment.

The main problem with the application of these algorithms for filtering noise from corrupted speech signals is that they can not recover the useful information missing from the distorted signal. For example, the missing information is caused by speech saturation. Sometimes, recovering the missing information is more difficult than cleaning the contamination from a signal. Some research work has been studied for speech signal recovery [78, 79, 80]. The objective of this research work is to restore the missing information for the corrupted signal and then remove the noise without distorting the original speech signal. A novel procedure based on phase space reconstruction is proposed to achieve the above goal. Some related research work has been reported in [81, 82, 83, 84, 85].

The whole process of the procedure is shown in the flowchart in Fig.4.1.

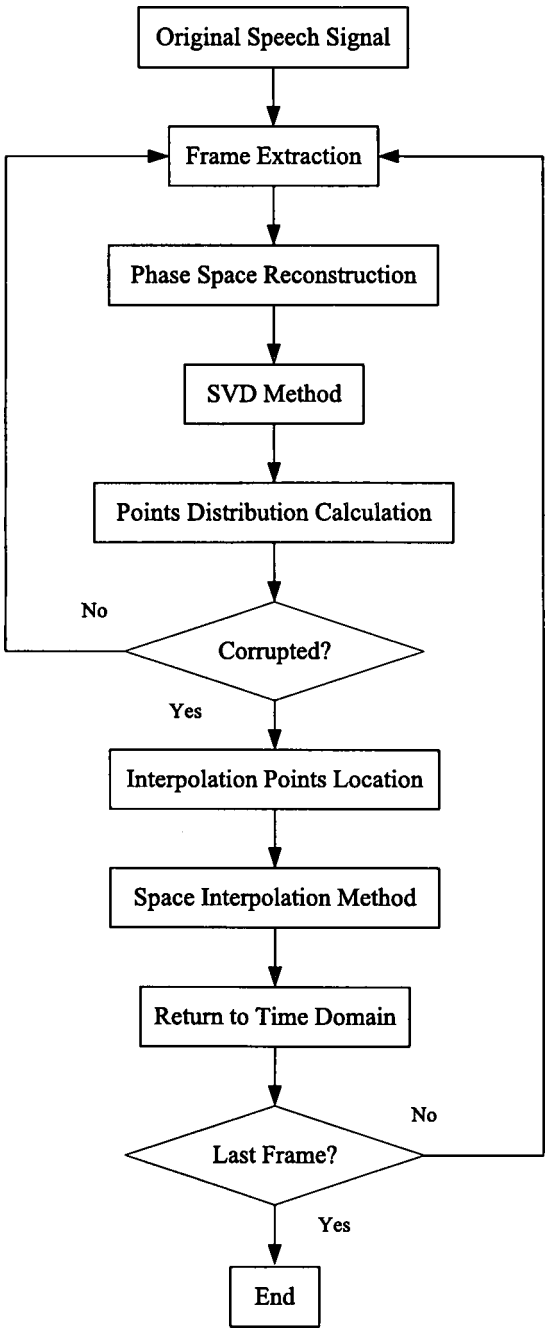


Figure 4.1: The flowchart of the procedure for speech signal restoration

As can be seen, the procedure is comprised of three parts:

(1) Detection of corrupted signal segments, including four steps: frame extraction, phase space reconstruction, SVD calculation and points distribution calculation, which is used to identify the corrupted part of the signal.

(2) Signal restoration, including two steps: interpolation points location and space interpolation, which is used to recover the missing information of a corrupted signal in phase space.

(3) Return to the time domain, the time-aligned weighted average method is used to transform the recovery signal from phase space back into the time domain.

4.2 Detection of corrupted speech signal segments

For most cases, noise reduction methods could be effectively used to remove noise from contaminated speech signals. However, in many cases, the distorted signal is not purely caused by add-on noise contamination but by losses of information from the signal. So the signal can not be restored by filtering. This motivated the work reported in this chapter and also the procedure development is inspired by the methods introduced in Chapter 3. The proposed procedure works in the way of that the damaged part of the speech signal is restored first and the filtering method is applied afterwards. There are two main steps in this procedure to recover the corrupted speech signal, namely, detection and restoration. The step of detection includes frame extraction, phase space reconstruction, singular value decomposition (SVD) and point distribution calculation.

4.2.1 Frame extraction and phase space reconstruction

To identify the corrupted part of a speech signal automatically, the original signal is divided into frames at the very first step. One frame is defined as a waveform segment which includes only one peak or crest and also all the data covered in the frame must be either positive or negative. Fig.4.2(a) shows the experimental signal and one particular frame extracted. The embedding method introduced in Chapter 2 is used to reconstruct the phase space of the speech signal with the embedding dimension $m = 3$ and the time delay $\tau = 1$. Fig.4.2(b) shows the example of phase space reconstruction for one typical frame.

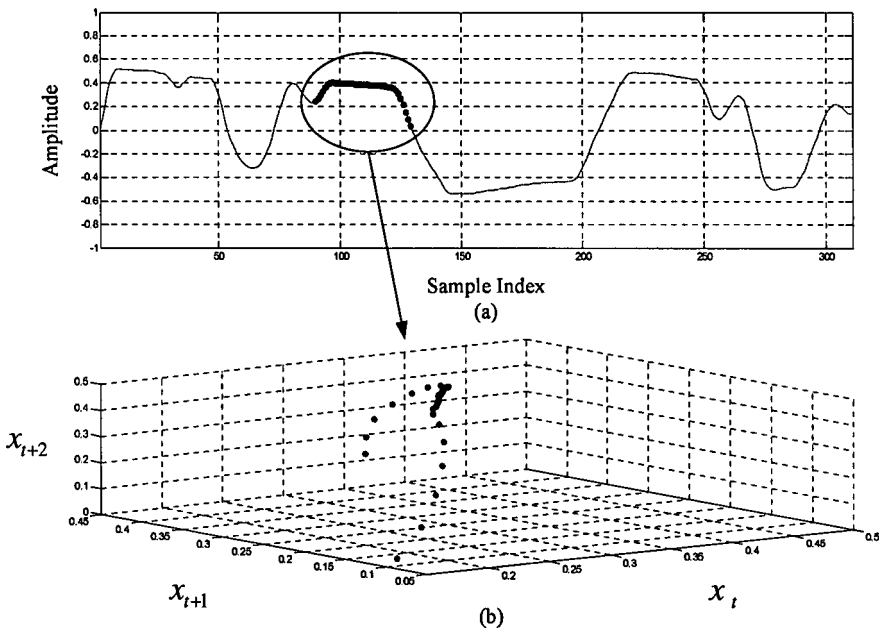


Figure 4.2: Phase space reconstruction. (a) Original speech signal and one frame extraction. (b) Phase space reconstruction of the frame, $m = 3, \tau = 1$.

4.2.2 Singular value decomposition

Singular Value Decomposition (SVD) method introduced in Chapter 3 is used to project the signal onto a signal primary subspace. In this procedure, SVD method is used for detection of the corrupted segment of the signal. Fig.4.3 shows the result of trajectory projection by using SVD method in the reconstructed phase space ($m = 3, \tau = 1$) for one frame of the speech signal.

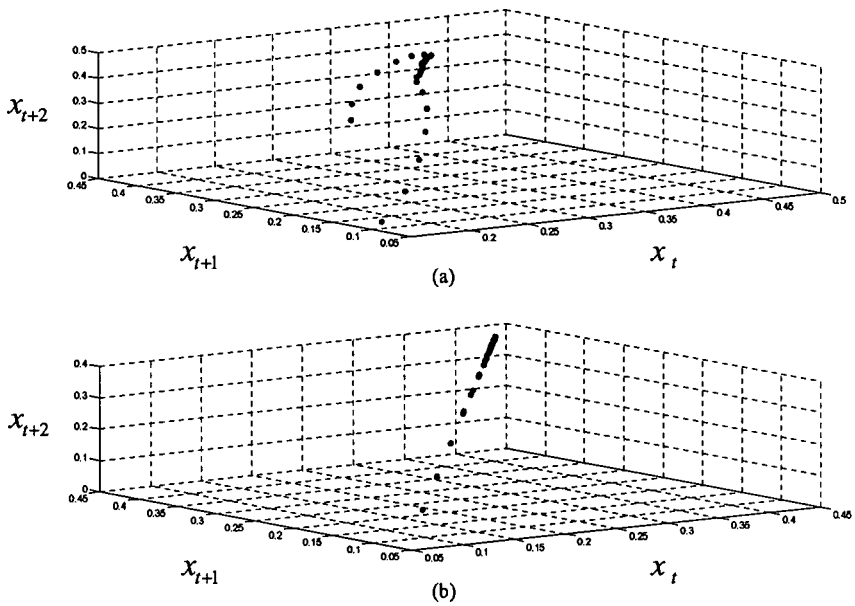


Figure 4.3: Trajectory projection. (a) Phase space reconstruction of the frame, $m = 3, \tau = 1$. (b) Trajectory projection by using SVD.

4.2.3 Points distribution calculation

Because only one eigenvector is used for projection, all points of trajectory after SVD method are then formed into a line in the phase space and the sum of distances between the trajectory points and this line is minimal. Fig.4.4(a) shows the space line. It can be seen that for the corrupted part of the signal, the distribution of the points along this space line is not uniform and most of them concentrate within a narrow scope. By calculating the distribution

ratio of the trajectory points along this line, we can tell whether this part of the signal is corrupted or not. The line is initially divided into L number of equal intervals. The distribution ratio is the maximal ratio of the number of points at an interval (POL_i) to the number of all points (M) along the line. Fig.4.4(b) shows the point distribution along the space line. The distribution ratio for each interval is calculated by:

$$R_i = \frac{POL_i}{M} \quad i = 1, 2, \ldots, L \tag{4.2.1}$$

where i is the index of intervals along the space line, R_i is the points distribution value for each interval, POL_i is the number of points in each interval and M is total number of points along the space line. So the maximal point distribution value $R_{\max}$ can be obtained by:

$$R_{\max} = \max\{R_i\} \quad i = 1, 2, \ldots, L \tag{4.2.2}$$

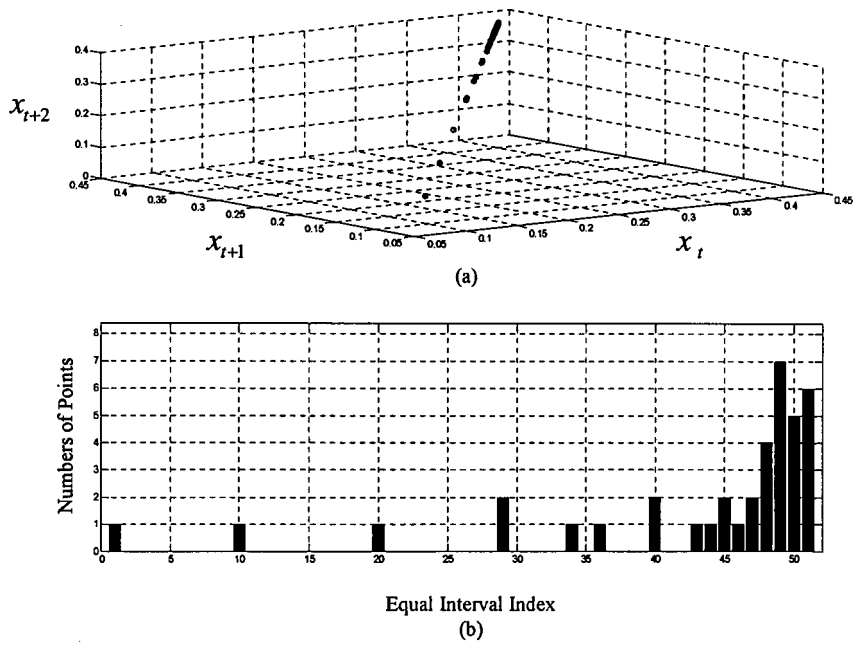


Figure 4.4: Distribution ratio of the trajectory points. (a) Trajectory projection by using SVD. (b) Points distribution calculation.

If $R_{\max}$ is higher than the threshold preset, the corruption degree of the signal segment is considered as serious. According to this criterion, the corrupted part of signal can be detected. Fig.4.5 shows the result of distribution ratio of all the frames. The frame whose the maximal distribution ratio above the threshold (60%) is identified as a corrupted frame.

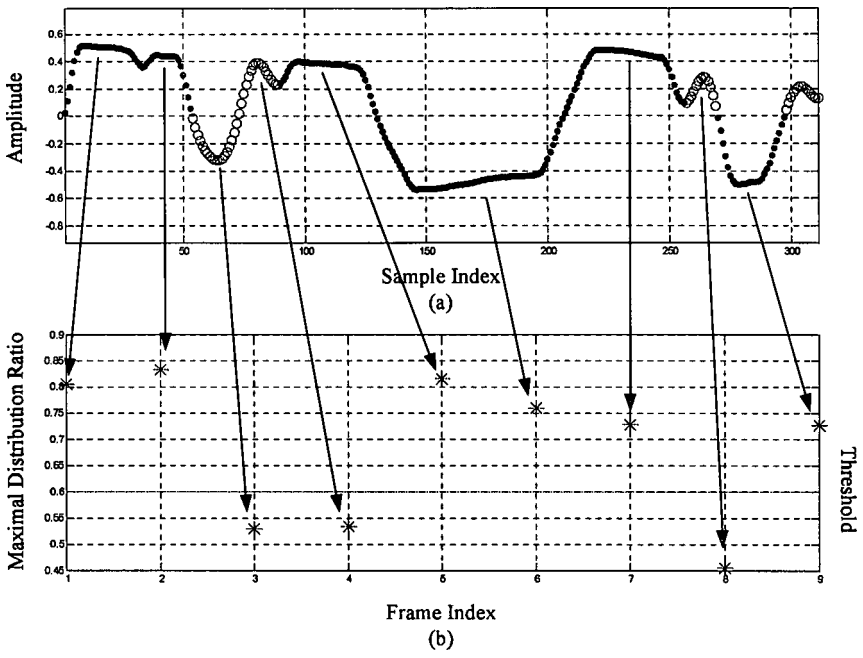


Figure 4.5: Distribution ratio of all frames. (a) Original speech signal and frames. (b) Points distribution ratio of all frames.

4.3 Signal restoration in phase space

In this section, the second main step, restoration, is discussed. The space interpolation method is used in phase space for this work.

Interpolation is a method of constructing a new data point set from a known discrete data point set. There are a number of interpolation methods reported such as linear interpolation, polynomial interpolation, spline interpolation and multivariate interpolation. Because of the multi-dimensional nature of the phase space, the interpolation method used in this procedure is the multivariate interpolation in which the interpolation function has more than one variable, cubic interpolation in two dimensions [86].

4.3.1 Locations of interpolation points

To use space interpolation method in phase space, there are three points at least should be identified first: the start point, the end point and the middle point. But for a better recovery result, the points used are based on the signal waveform changing trend. Four points are initially selected: two start points and two end points. Firstly, the slope at each point is calculated by using the equation:

$$x_{i\_\text{slope}} = \frac{x_{i+1} - x_{i-1}}{2} \qquad (i = 2, 3 \ldots, N-1) \tag{4.3.1}$$

The changing value of the slope is defined as:

$$CVS = \sum_{i=2}^{N-2} |(x_{(i)\_\text{slope}} - x_{(i+1)\_\text{slope}}| \tag{4.3.2}$$

where x_{i_slope} is the point slope value, N is the length of the signal and CVS represents the absolute difference between adjacent elements of x_{i_slope}.

According to the characteristic of the corrupted part of the signal, the interpolation points should be those points whose slope values have a sharp change. If the CVS value changes sharply at the beginning or by the end of the frame, the points in those positions should be considered as the segment terminal points for interpolation. In this procedure, the sharp change is set as 20% changing between two adjacent slopes. Fig.4.6 illustrates the results of interpolation point location.

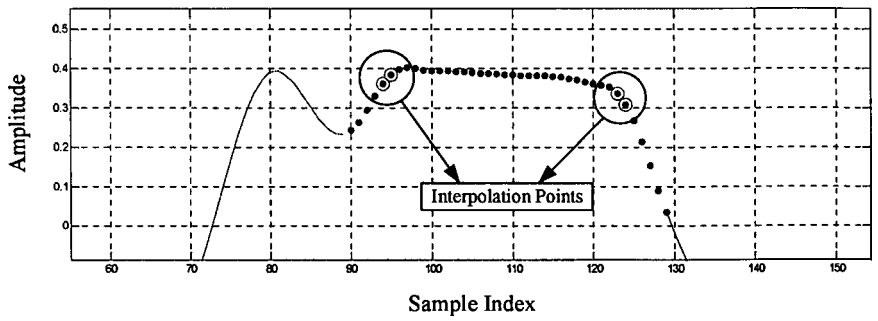


Figure 4.6: Interpolation points location

4.3.2 Results from application of the space interpolation method

Fig.4.7 shows the results after using the interpolation method in phase space.

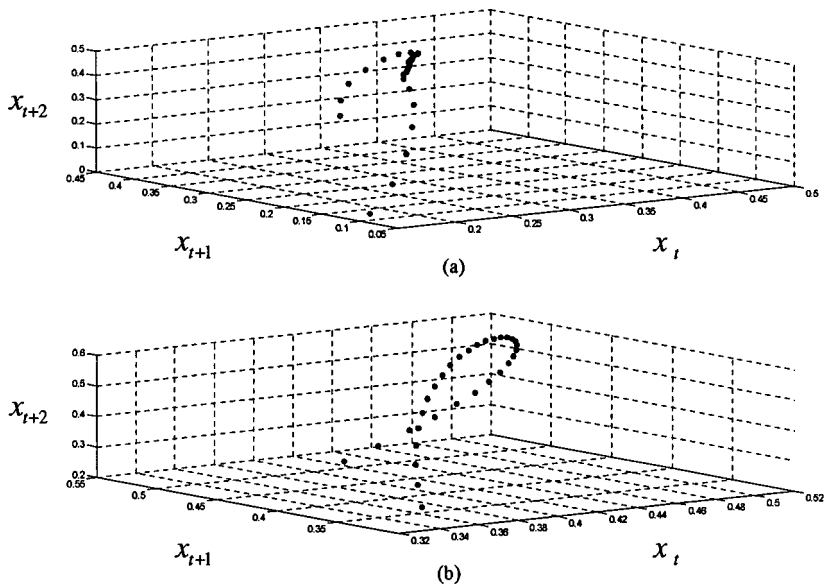


Figure 4.7: Space interpolation method. (a) Result before using space interpolation method. (b) Result after using space interpolation method.

4.4 Transform back from phase space to time domain

Once interpolation method has been applied, the new trajectory matrix no longer corresponds to a time-delay embedding in a single time series, and there is no unique mapping back to a one-dimensional signal. Each column of the transformed matrix represents a possible enhanced signal output. There are several methods for creating an enhanced time-series from the new trajectory matrix, including selecting a single column from it or calculating a time-aligned averaging of the columns which is introduced in Chapter 3.

In this procedure, a time-aligned weighted average method is chosen [63] which is a slightly different from the time-aligned median method we introduced in Chapter 3. The new method uses higher weight given to the values in the center columns of the matrix and lower weight given to the values in the left-most and right-most columns. This corresponds to emphasizing the time-centered value of each projected point. Signal points near the beginning or end of the trajectory matrix have fewer representatives and are weighted accordingly. To illustrate this process, an example trajectory matrix X, aligned trajectory matrix X_{aligned}, and weighting matrix P are shown below for the case of $d = 3$, $\tau = 1$ with n=50 points in the time-series.

$$X^T = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 45 & 46 & 47 & 48 \\ 2 & 3 & 4 & 5 & 6 & \cdots & 46 & 47 & 48 & 49 \\ 3 & 4 & 5 & 6 & 7 & \cdots & 47 & 48 & 49 & 50 \end{bmatrix} \tag{4.4.1}$$

$$X^T_{\text{aligned}} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 48 & & \\ & 2 & 3 & 4 & 5 & \cdots & 48 & 49 & \\ & & 3 & 4 & 5 & \cdots & 48 & 49 & 50 \end{bmatrix} \tag{4.4.2}$$

$$P^T = \begin{bmatrix} 1 & 0.5 & 0.25 & 0.25 & \cdots & 0.25 & 0.25 & & \\ & 0.5 & 0.5 & 0.5 & \cdots & 0.5 & 0.5 & 0.5 & \\ & & 0.25 & 0.25 & \cdots & 0.25 & 0.25 & 0.5 & 1 \end{bmatrix} \tag{4.4.3}$$

The resulting output time-series is given by:

$$Y(m) = \sum_{i=1}^{d} p_i(m) x_i(m) \quad (m = 1, 2, \ldots, n + d - 1) \tag{4.4.4}$$

where x_i is the ith column from X_{aligned}, p_i is the corresponding weight vector from P. Each row of the weighting matrix sums to one. m is the index of output time-series. Fig.4.8 shows the result of time-aligned weighted average method.

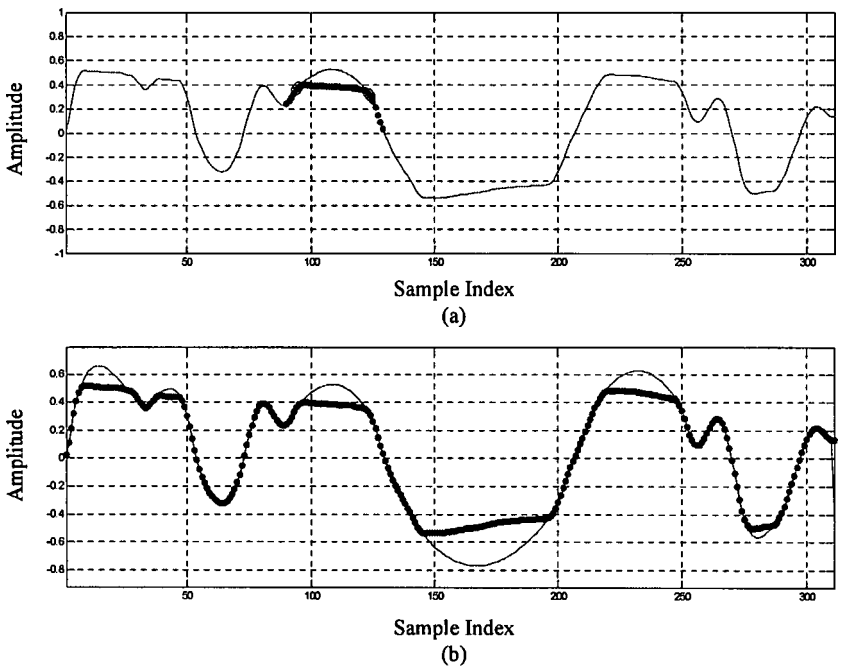


Figure 4.8: Signal return to time domain. (a) Restoration result of one frame. (b) Restoration result of all frames.

4.5 MMSE LSA estimator

MMSE LSA estimator is a short-time spectral amplitude (STSA) estimator for speech signals which minimizes the mean square error of the log spectra. The estimator is very effective in dealing with noisy speech signals, and it significantly improves the quality of speech.

MMSE LSA estimator is derived from an algorithm [74] which is proposed for enhancing speech degraded by uncorrelated additive noise when the noisy speech alone is available. This algorithm capitalizes on the major importance of the short-time spectral amplitude (STSA) of the speech signal in its perception, and utilizes a minimum mean-square error (MMSE) STSA estimator for enhancing the noisy speech.

While the distortion measure of mean-square error of the spectra (i.e., the original STSA and its estimator) used in [74] is mathematically tractable, and leads also to good results, it is not the most subjectively meaningful one. It is well known that a distortion measure which is based on the mean-square error of the log-spectra is more suitable for speech processing. Such a distortion measure is therefore extensively used for speech analysis and recognition. For this reason, it is of great interest to examine the STSA estimator which minimizes the mean-square error of the log-spectra in enhancing noisy speech. The derivation of the above STSA estimator and its comparison with the MMSE STSA estimator derived in [75].

We found that MMSE LSA estimator is superior to the MMSE STSA estimator since it results in a much lower residual noise level without further affecting the speech itself. So the MMSE LSA estimator is used in this algorithm. The whole computation process of MMSE LSA estimator is explained below and the corresponding Matlab code can be found in Appendix E.2.

4.5.1 Gain function of MMSE LSA estimator

An observed noisy speech signal $x(t)$ is assumed to be a clean speech signal $s(t)$ degraded by uncorrelated additive noise $n(t)$, i.e.,

$$x(t) = s(t) + n(t) \quad 0 < t < T \tag{4.5.1}$$

Let $S_k = A_k e^{j\alpha_k}$, N_k and $X_k = R_k e^{j\theta_k}$ denote the kth spectral component of the clean speech signal $s(t)$, noise $n(t)$, and the observed noisy speech $x(t)$, respectively, in the analysis interval $[0, T]$. We are looking for the estimate of A_k, $\hat{A}_k$, which minimizes the mean-square error of log-spectra between the speech spectral amplitude and the estimated speech spectral amplitude:

$$E\{(\log A_k - \log \hat{A}_k)^2\} \tag{4.5.2}$$

given the observed signal $\{x(t), 0 \leq t \leq T\}$, where E indicates the expectation operator. Obviously, the estimator is given by:

$$\hat{A}_k = \exp\{E[\ln A_k | X_t, 0 \leq t \leq T]\} \tag{4.5.3}$$

Assuming the individual spectral components are statistically independent of one another, the expected value of A_k given $\{x(t), 0 \leq t \leq T\}$ is equal to the expected value of A_k given X_k only. We therefore have

$$\hat{A}_k = \exp\{E[\ln A_k | X_k\} \tag{4.5.4}$$

The evaluation of $E[\ln A_k | X_k]$ for the Gaussian model assumed here is conveniently done by utilizing the moment generating function of $\ln A_k$ given X_k. Let $Z_k = \ln A_k$. Then the moment generating function $\Phi_{Z_k|X_k}(\mu)$ of Z_k given X_k equals:

$$\begin{aligned} \Phi_{Z_k|X_k}(\mu) &= E\{\exp(\mu Z_k)|X_k\} \\ &= E\{A_k^{\mu}|X_k\} \end{aligned} \tag{4.5.5}$$

$E\{\ln A_k | X_k\}$ is obtained from $\Phi_{Z_k|X_k}(\mu)$ by:

$$E\{\ln A_k | X_k\} = \frac{d}{d\mu} \Phi_{Z_k|X_k}(\mu)|_{\mu=0} \tag{4.5.6}$$

Therefore, our task is now to calculate $\Phi_{Z_k|X_k}(\mu)$ and then to obtain $E\{\ln A_k|X_k\}$ by using (4.5.6). From (4.5.5), $\Phi_{Z_k|X_k}(\mu)$ is given by:

$$\begin{aligned}\Phi_{Z_k|X_k}(\mu) &= E\{A_k^\mu|X_k\} \\ &= \frac{\int_0^\infty \int_0^{2\pi} a_k^\mu p(X_k|a_k,\alpha_k)p(a_k,\alpha_k)d\alpha_k da_k}{\int_0^\infty \int_0^{2\pi} p(X_k|a_k,\alpha_k)p(a_k,\alpha_k)d\alpha_k da_k}\end{aligned} \tag{4.5.7}$$

where the symbol a_k denotes the sample value of A_k. With the complex Gaussian assumption of each individual spectral component of speech and noise, the conditional probability density function (PDF) of the observed spectral component given a_k and α_k, $p(X_k|a_k,\alpha_k)$ is given by:

$$p(X_k|a_k,\alpha_k) = \frac{1}{\pi\eta_n(k)}\exp\{-\frac{|X_k - a_k e^{j\alpha_k}|^2}{\eta_n(k)}\} \tag{4.5.8}$$

and the joint PDF of the speech spectral amplitude and phase, $p(a_k,\alpha_k)$, is given by:

$$p(a_k,\alpha_k) = \frac{a_k}{\pi\eta_s(k)}\exp\{-\frac{a_k^2}{\eta_s(k)}\} \tag{4.5.9}$$

where $\eta_n(k) = E\{|N_k|^2\}$, $\eta_s(k) = E\{|S_k|^2\}$ and are the variances of the kth spectral components of noise and speech signals, respectively. Substituting (4.5.8) and (4.5.9) into (4.5.7), and using the integral representation of the modified Bessel function of zero order, we have (4.5.10):

$$\Phi_{Z_k|X_k}(\mu) = \eta_k^{\mu/2}\Gamma(\frac{\mu}{2}+1)M(-\frac{\mu}{2};1;-v_k) \tag{4.5.10}$$

where $\Gamma(h)$ is the gamma function and $M(\alpha,\gamma,z)$ is the confluent hypergeometric function, i.e.,

$$\Gamma(h) = \int_0^\infty t^{h-1}e^{-t}dt \tag{4.5.11}$$

$$M(\alpha,\gamma,z) = 1 + \frac{\alpha}{\gamma}\frac{z}{1!} + \frac{\alpha(\alpha+1)}{\gamma(\gamma+1)}\frac{z^2}{2!} + \frac{\alpha(\alpha+1)(\alpha+2)}{\gamma(\gamma+1)(\gamma+2)}\frac{z^3}{3!} + \ldots \tag{4.5.12}$$

where $\eta(k)$ and v_k are defined as follows:

$$\eta(k) = [\frac{1}{\eta_s(k)} + \frac{1}{\eta_n(k)}]^{-1} \tag{4.5.13}$$

$$v_k = \frac{\xi_k}{1+\xi_k}\gamma_k \tag{4.5.14}$$

where ξ_k and γ_k represent the a priori SNR and a posteriori SNR respectively, i.e.,

$$\xi_k = \frac{\eta_s(k)}{\eta_n(k)}, \quad \gamma_k = \frac{R_k^2}{\eta_n(k)} \tag{4.5.15}$$

So, the derivative of $M(-\mu/2; 1; v_k)$ at $\mu = 0$ is obtained by:

$$\frac{\partial}{\partial \mu} M(-\mu/2; 1; v_k)|_{\mu=0} = -\frac{1}{2}\sum_{r=1}^{\infty} \frac{(-v)^r}{r!}\frac{1}{r} \tag{4.5.16}$$

The derivative of $\Gamma(\mu/2+1)$ is conveniently obtained through the derivative of $ln\Gamma(\mu/2+1)$ by using:

$$\frac{\partial}{\partial \mu}\Gamma(\frac{\mu}{2}+1) = \Gamma(\frac{\mu}{2}+1)\frac{\partial}{\partial \mu}\ln\Gamma(\frac{\mu}{2}+1) \tag{4.5.17}$$

The derivative of $ln\Gamma(\mu/2+1)$ is obtained by utilizing its series expansion given by:

$$\ln\Gamma(\frac{\mu}{2}+1) = -c\frac{\mu}{2} + \sum_{r=2}^{\infty}\frac{(-\mu)^r}{2^r r}\alpha_r \tag{4.5.18}$$

where

$$\alpha_r = \sum_{n=1}^{\infty}\frac{1}{n^r} \tag{4.5.19}$$

and $c = 0.57721566490$ is the Euler constant. Differentiating (4.5.18) term by term, and using (4.5.17) gives:

$$\frac{\partial}{\partial \mu}\Gamma(\frac{\mu}{2}+1)|_{\mu=0} = -\frac{c}{2} \tag{4.5.20}$$

Now, by using (4.5.16) and (4.5.20) we obtain from (4.5.10):

$$\begin{aligned}\frac{\partial}{\partial \mu}\Phi_{Z_k|X_k}(\mu)|_{\mu=0} &= \frac{1}{2}\ln\lambda_k - \frac{1}{2}(c + \sum_{r=1}^{\infty}\frac{(-v_k)^r}{r!}\frac{1}{r}) \\ &= \frac{1}{2}\ln\lambda_k + \frac{1}{2}(\ln v_k + \int_{v_k}^{\infty}\frac{e^{-t}}{t}dt)\end{aligned} \tag{4.5.21}$$

The integral in (4.5.21) is known as the exponential integral of v_k, and can be efficiently calculated. On substituting (4.5.21) into (4.5.6) and using (4.5.15) and (4.5.4), we get the desired amplitude estimator:

$$\hat{A}_k = \frac{\xi_k}{1+\xi_k}\exp\{\frac{1}{2}\int_{v_k}^{\infty}\frac{e^{-t}}{t}dt\}R_k \tag{4.5.22}$$

It is useful to consider $\hat{A}_k$ as being obtained from R_k, by a multiplicative nonlinear gain function which depends only on the *a priori* and the *a posteriori* SNR ξ_k and γ_k, respectively. This gain function is defined by:

$$G(\xi_k, \gamma_k) = \frac{\hat{A}_k}{R_k} \tag{4.5.23}$$

It is interesting to note that the new gain function which results from (4.5.22) always gives a lower gain than the one which results from the estimator of [74]. This is easy to prove by using Jensens inequality:

$$\hat{A}_k = \exp\{E[\ln A_k|X_k\} \leq \exp\{\ln E[A_k|X_k\} = E[A_k|X_k] \tag{4.5.24}$$

The STSA estimator which minimizes the mean-square error of the log-spectra is derived and is examined in enhancing noisy speech. The MMSE LSA estimator is superior to the MMSE STSA estimator derived in [74] since it results in a much lower residual noise level without further affecting the speech itself. In fact, the new estimator results in a very similar enhanced speech quality as that obtained with the MMSE STSA estimator of [74], which takes into account the signal presence of uncertainty.

4.6 Experimental results

The procedure proposed in this chapter has an advantage of without requiring spectral models for speech and noise signals. This procedure can also suppress the noise during the process of signal restoration. The procedure shows a better performance when it is combined with other noise reduction methods in signal processing, such as log spectral amplitude (LSA) estimation which is introduced in previous section. The experimental results show that the speech signal processing with restoration and noise reduction can result in an improved quality of the signal processing. Fig.4.9 and Fig.4.10 show the results of LSA method without and with signal restoration, respectively.

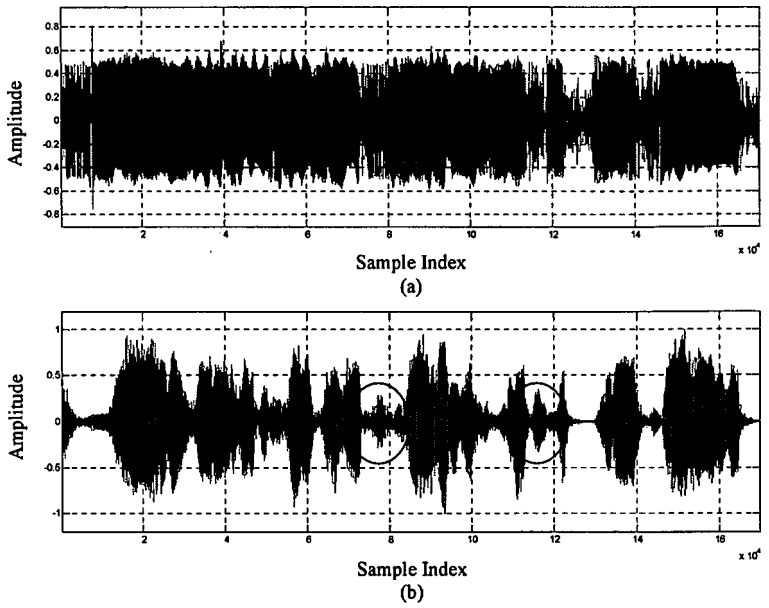


Figure 4.9: Noise reduction without signal restoration. (a) Original signal. (b) LSA method without restoration.

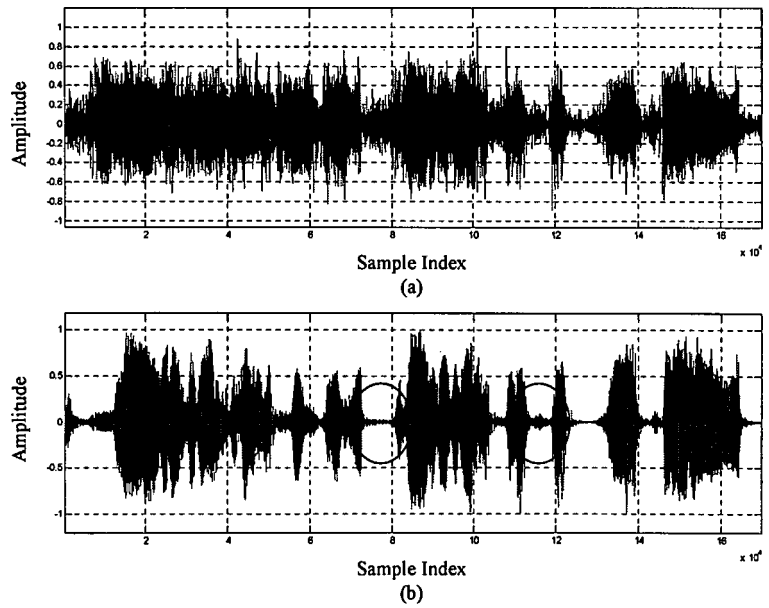


Figure 4.10: Noise reduction with signal restoration. (a) Signal after restoration. (b) LSA method with restoration.

The two circles in the Fig.4.9 and Fig.4.10 are the noise parts after using noise reduction method. From Fig.4.10, it can be seen that the noise part in the circles can be eliminated completely by using LSA method with signal restoration. In our procedure, the embedding dimension m is set as 3, time delay τ is set as 1 and the point distribution ratio is set as 60%.

The speech signal used in this experiment is a piece of telephone conversation provided by a police station. The conversation recorded on the CD is almost not recognizable before processing. After the procedure developed in this thesis is applied, the whole set of the conversation can be recognized by the police officer apart from a few missing words. Certainly, the understanding level of the speech depends on the listener, such as, the native English speaker or a person with English as his/her second language. We have asked a number of persons to listen the record and recite the contents of the record. The conversation can be understood by the listener with English as the second language.

4.7 Summary

In this chapter, a phase space trajectory reconstruction method for speech signal restoration is proposed. The procedure is based on the theory of nonlinear dynamic system analysis, in which the signal is mapped (projected) into a high dimensional phase space. Firstly, the SVD method is used to detect the corrupted signal segment. Secondly, the new trajectory of signal in phase space is reconstructed by using space interpolation method. Thirdly, the signal is transformed from phase space back into the time domain for the signal restoration. Finally, the restored signal is processed using noise reduction method. The experimental results demonstrate the effectiveness of this procedure to show that the result of noise elimination for restored signal is much better than that of signal without restoration. The main contributions of the procedure can be considered to have three folds:

(1) Finding an efficient way to detect the corrupted signal segments by using the SVD method.

(2) Using interpolation method in phase space to achieve the signal restoration.

(3) Proving the effectiveness the procedure that the better result of noise elimination can be obtained for the signal with restoration.

Chapter 5

Uterine Electromyography Signal Feature Extraction and Classification

Uterine electromyography (EMG) signal has a potential to be used for early diagnosis of preterm labor clinically. But it is difficult to differentiate the patterns of uterine contractions from EMG signals which will lead to preterm birth or not. In this chapter, a number of methods have been investigated with the aim of finding a possible effective procedure which includes a variety of signal processing methods to extract the features from uterine EMG signal and to classify the normal term labor from abnormal preterm labor signals. In this procedure, the signal is preprocessed to eliminate the noise and high frequency components using threshold de-noising and wavelet de-noising methods. Then, the fractal dimension value along the signal is calculated for the extraction of contraction patterns using techniques rooted from chaos theory which includes phase space reconstruction and singular value decomposition. Each contraction pattern of the signal is decomposed into sub-signals using wavelet packet transform method and the average wavelet packet energy of all the contraction patterns of each EMG signal is calculated. Afterwards, the signals are classified using artificial neural network method. The experimental results show that the classification accuracy of term labor signal and preterm labor signal can reach

64.1%. Although it is not a result of confidence, it is an very encouraging for further study heading the right direction for procedure development.

5.1 Introduction

Electromyogram (EMG) measures the electrical activity of muscles to gather information about muscular and nervous systems. As the use of EMG is convenient, it is now becoming increasingly a powerful measure to get information and to diagnose problems and diseases about the muscular and nervous system. For example, it can diagnose some causes of muscle weakness or paralysis, muscle or motor problems [87], sensory problems, nerve damage or injury. EMG is also used in other fields such as kinesiology, gait [88] and posture studies [89] and prosthesis design [90]. But up to now, most analysis methods of EMG are still based on linear and statistical analysis. Only a few people deal with nonlinear principle and method. In recent years, some research work have been reported in analysis of EMG signals using nonlinear methods which are based on chaos theory [91, 92, 93, 94].

Abdominal uterine electromyography is an electrophysiological signal representing uterine activity during women pregnancy which has been studied for many years. Proper diagnosis of labor is one of the major challenges faced by obstetricians. There are no accurate and objective methods to predict the onset of labor, to differentiate the true and false labor both for term and preterm patients, or to determine whether the false labor will progress to true labor and on what time scale. At present, the progress of labor is monitored by recording the changes in the cervical state and by measuring the rate, duration and amplitude of uterine contraction using a tocodynamometer or the intrauterine pressure catheter (IUPC). Owing to the poor predictive power of the tocodynamometer and the invasive nature of the IUPC, neither technique has been beneficial in prediction of preterm tabor or the diagnosis of true labor at term [95].

Characterization of these uterine electrical events is possible non-invasively through recording of uterine EMG signals from the abdominal surface. In ad-

dition to providing the usual information concerning the frequencies of uterine contractions, the analysis of the characteristics of electrical events measured by transabdominal EMG can assess the progression of uterine preparedness for labor. Recent studies have proved that it is able to provide reliable information about uterine contractions. More recent ones have shown that the characterization of the contractile activity during pregnancy would be of great value for the diagnosis of preterm delivery [96, 97, 98, 99, 100].

EMG signals have been processed in the steps as shown in the flowchart in Fig.5.1. As can be seen, the procedure is comprised of three main parts: Signal preprocessing, in which the threshold de-noising and wavelet transform methods are used to eliminate noise and get rid of the high frequency component of the signal; Signal feature extraction, in which the fractal dimension value along the signal is calculated for the contraction pattern extraction and the average wavelet packet energy of each contraction pattern is calculated to be used for the classification; Signal classification, the term labor signal and preterm labor signal are classified using artificial neural network (ANN).

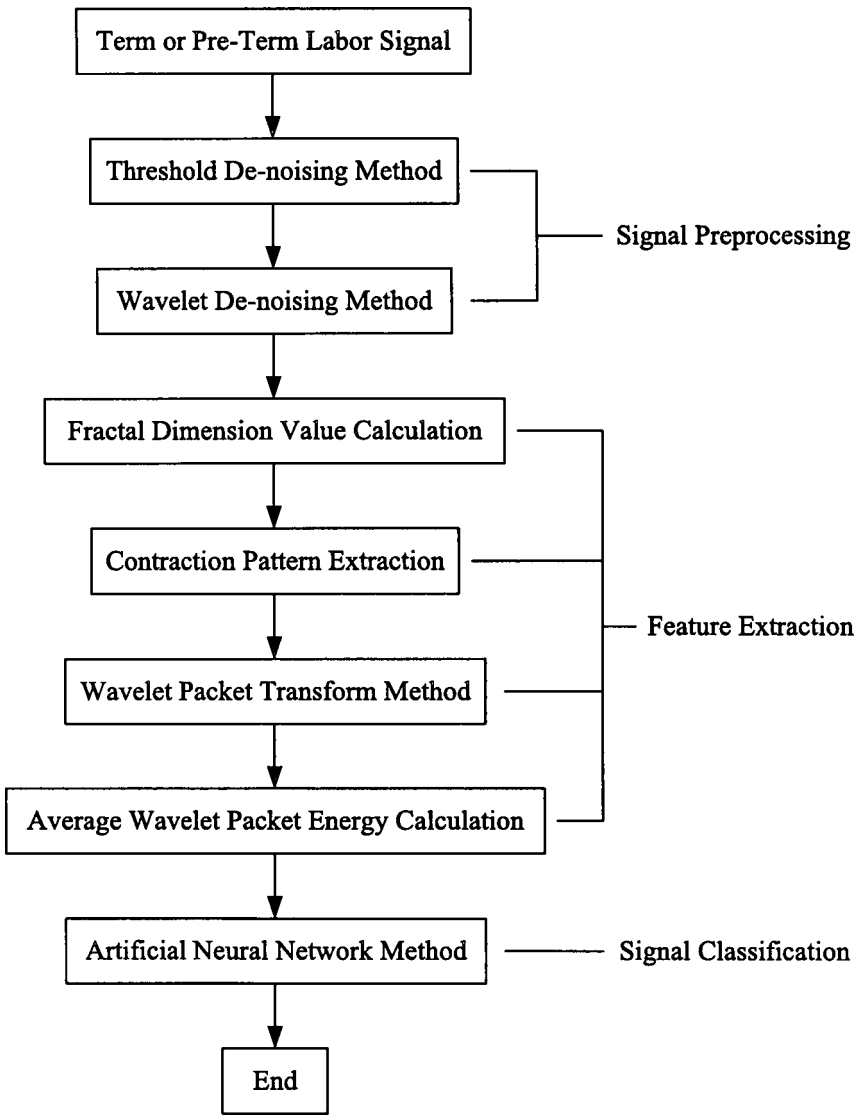


Figure 5.1: Flowchart of the procedure

5.2 Signal preprocessing

5.2.1 Threshold de-noising

As the original signal is always contaminated by noise, the pre-processing of the signal is necessary in most cases. A typical example of the original signal is shown in Fig.5.2(a). We can see that there are some high values of sharp sparks across the whole signal segment and the sparks have extreme large values which make it difficult to see the normal contraction patterns from the signal clearly. So the first step is to remove such kind of contaminations from the original signal. The method used here is called the threshold de-noising method in which a threshold value is set so that any signal values over the limit will be eliminated. Results are shown in Fig.5.2(b) and Fig.5.2(c).

5.2.2 Wavelet transform de-noising

The theory of wavelet de-noising [101, 102] is based on the multi-resolution analysis of wavelet transform which is a powerful time-frequency method for non-stationary signal analysis. The wavelet transform is a transformation of the original temporal signal into a wavelet basis space. The time-frequency wavelet representation is performed by repeatedly filtering the signal with a pair of filters that cut the frequency domain in the middle. Specifically, the wavelet transform decomposes a signal into an approximation signal and a detail signal. The approximation signal is subsequently divided into new approximation and detail signals. This process is carried out iteratively producing a set of approximation signals at different detail levels (scales) and a final gross approximation of the signal.

The detail D_j and the approximation A_j at the level j can be obtained by filtering the signal with an L-sample high pass filter g, and an L-sample low pass filter h. Both approximation and detail signals are down sampled by a factor of two. The whole procedure of wavelet transform can be found in Fig.5.3:

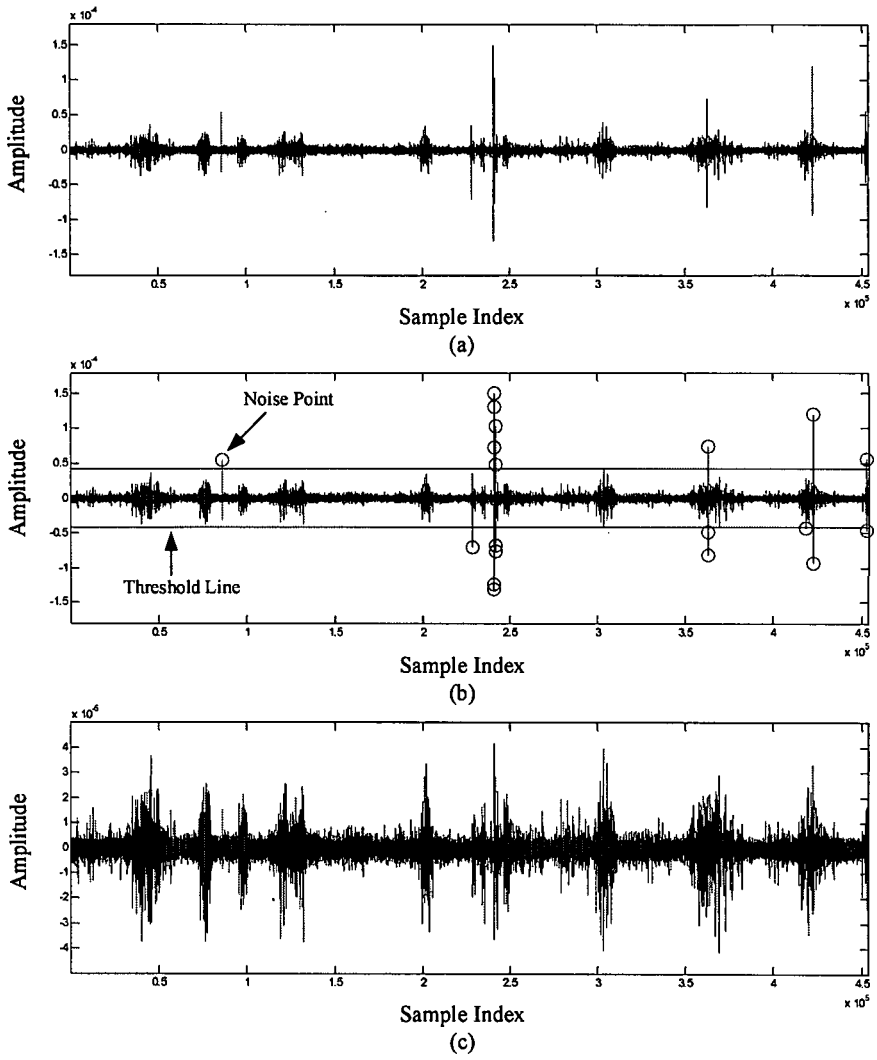


Figure 5.2: Threshold de-noising. (a) Original signal; (b) Threshold line and noise points; (c) Result of threshold de-noising.

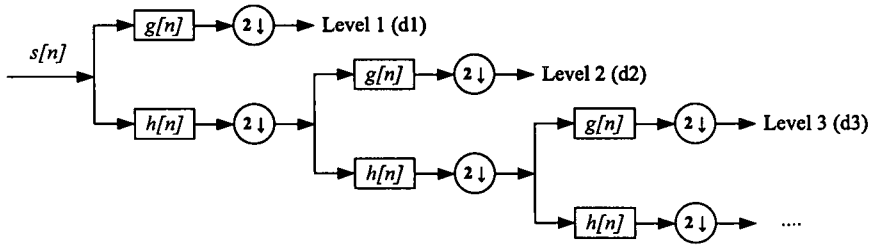


Figure 5.3: Wavelet transform decomposition.

The formulas to calculate the both approximation and detail coefficients can be expressed as follows:

$$A_j[n] = H\langle A_{j-1}[n]\rangle = \sum_{k=0}^{L-1} h[k]A_{j-1}[2n-k] \tag{5.2.1}$$

$$D_j[n] = G\langle D_{j-1}[n]\rangle = \sum_{k=0}^{L-1} g[k]A_{j-1}[2n-k] \tag{5.2.2}$$

where $A_0[n], n = 0, 1, \ldots, N-1$ is the original temporal sequence, while H and G represent the convolution/down sampling operators. Sequences $g[n]$ and $h[n]$ are associated with wavelet function $\psi(t)$ and the scaling function $\varphi(t)$ through inner products:

$$g[n] = \langle \psi(t), \sqrt{2}\psi(2t-n)\rangle \tag{5.2.3}$$

$$h[n] = \langle \varphi(t), \sqrt{2}\varphi(2t-n)\rangle \tag{5.2.4}$$

We can select a set of thresholds and apply them to the signals at each of the scales. Then we use the inverse wavelet transform to reconstruct a new signal. The final reconstruction of the original signal can be computed by the details and the approximations, as described by the following equation, for fixed N:

$$S(t) = A_N(t) + D_1(t) + D_1(t) + \ldots + D_N(t) \tag{5.2.5}$$

The wavelet transform can be used in order to perform artifact removal. Its application is based on the spectral separation between the original signal and the artifact Thus, by using the wavelet decomposition (5.2.5), it is possible to

remove the artifact by means of the de-noising procedure applied to the single wavelet details.

In our experiment, the *bior*1.5 wavelet is selected as the mother wavelet and the signal is decomposed into 8 levels. Only the detail coefficients for the $7th$ and $8th$ levels are kept. The results of decomposition are demonstrated in Fig.5.4.

The wavelet based de-noising method is summarized in the following three steps: (1) Choosing a wavelet and level N and computing the wavelet decomposition of the signal at level N; (2) For each level, selecting a threshold and apply the threshold to the detail coefficients; (3) Computing the wavelet reconstruction based on the original approximation coefficients of level N and the left detail coefficients of levels 1 to N.

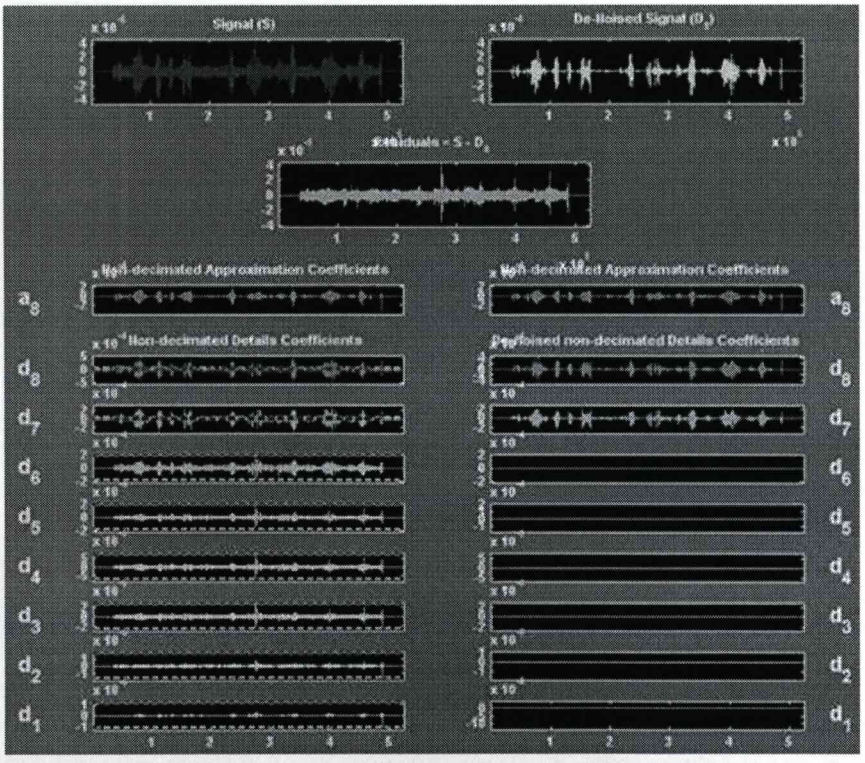


Figure 5.4: Wavelet de-noising.

After wavelet de-noising, the signal is normalized into a same amplitude scale for further processing. Fig.5.5 shows the result of signal normalization.

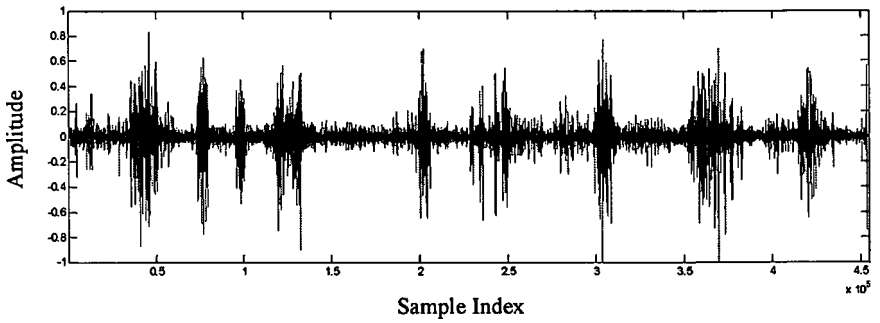


Figure 5.5: Signal normalization.

5.3 Feature extraction

5.3.1 Contraction pattern extraction

In this part, a different signal processing technique is investigated which is rooted from the nonlinear dynamic chaotic system theory. A fundamental idea in nonlinear analysis is that the dynamics of a system can be studied in a phase space. A point in this space characterizes the state of the system at any moment of time. EMG signals can be considered as a kind of chaotic signal or a time sequence, so it can be analyzed by using chaotic techniques and the generalised methods [103, 104]. Fractal dimension is one of the chaotic characterizations used in the chaotic system to calculate the dimension of an attractor. For different parts of a signal, the fractal dimension values are different. By calculating the fractal dimension value, it is possible to separate the wanted signal from the noise components [105, 106, 107]. There are three main steps, namely, phase space reconstruction, single value decomposition (SVD) and fractal dimension calculation whose basic conceptions have been introduced already in Chapter 2 and Chapter 3, respectively.

Phase space reconstruction

Two particular signal patterns, non-contraction and contraction patterns, have been taken from the EMG signals and the phase space is reconstructed. The results are shown in Fig.5.6. Here, the embedding dimension m is set as 3 and the time delay τ is set as 1.

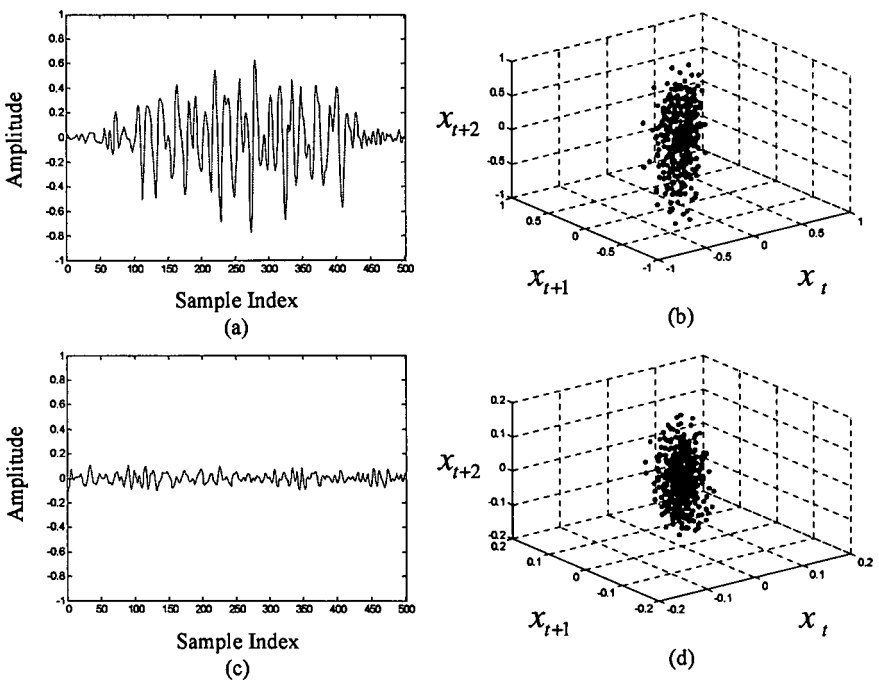


Figure 5.6: Phase space reconstruction. (a) Contraction pattern signal; (b) Phase space reconstruction of contraction pattern signal, $m = 3, \tau = 1$; (c) Non-contraction pattern signal; (d) Phase space reconstruction of non-contraction pattern signal, $m = 3, \tau = 1$.

Fractal dimension value calculation

Fractal dimensions have multiple definitions, but one thing is in common: their values are usually non-integer and fractional number, hence this dimension is referred to as fractal. The existing fractal dimensions include Hausdorff dimension, box dimension, information dimension, correlation dimension and

so on. In this procedure, the correlation dimension D_c of time series is used as fractal dimension.

D_c introduced by Grassberger and Procaccia [108] is widely used in many fields for characterization of strange attractors. In fractal analysis, D_c determines how the distribution of signal set scales up/down with decreasing/increasing radius of each hypersphere. In the GP algorithm for D_c, correlation integral function is applied as follows:

$$I_c(r) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \theta(d_{ij}, r) \tag{5.3.1}$$

where

$$\theta(d_{ij}, r) = \begin{cases} 1, & d_{ij} \leq r \\ 0, & d_{ij} > r \end{cases} \quad i \neq j \tag{5.3.2}$$

is Heaviside unit function, and d_{ij} is a Euclidean distance between points y_i and y_j in the reconstruction phase space. D_c can be estimated by:

$$D_c = \lim_{r \to 0} \frac{\log(I_c(r))}{\log(r)} \quad i \neq j \tag{5.3.3}$$

Phase space projection

The SVD method is used for each point on the trajectory within a neighborhood. After each point on an attractor is processed, the fractal dimension value of the new attractor can be recalculated. If this whole procedure repeats again and again, the fractal dimension value of the new attractor will change very different between contraction patterns and non-contraction patterns which show in Fig.5.7 and Fig.5.8.

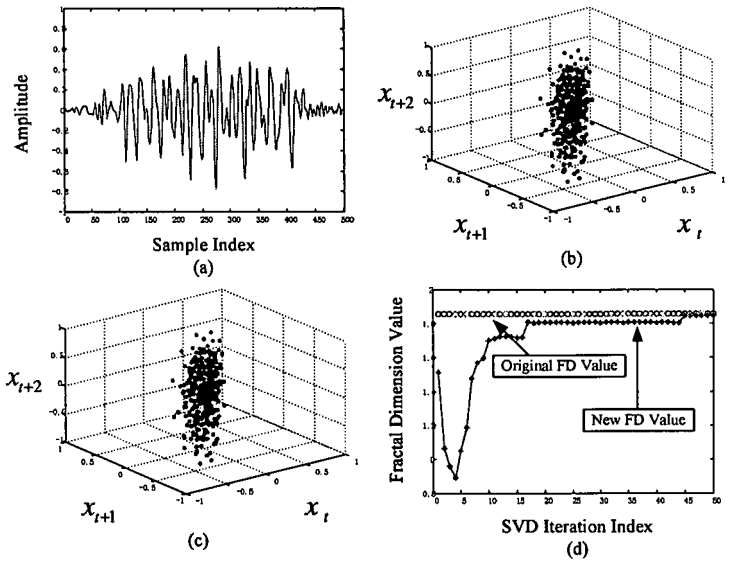


Figure 5.7: SVD method for contraction pattern. (a) Contraction pattern; (b) Phase space reconstruction; (c) New trajectory points after SVD method; (d) New fractal dimension value calculation.

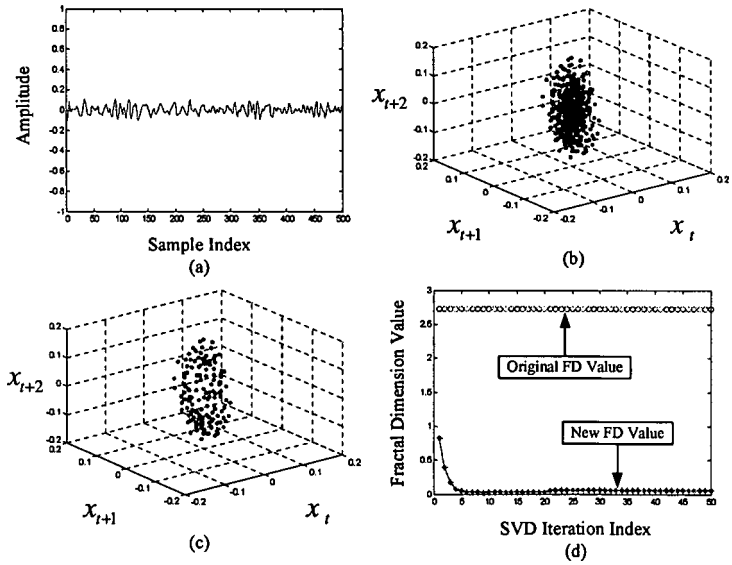


Figure 5.8: SVD method for non-contraction pattern. (a) Non-contraction pattern; (b) Phase space reconstruction; (c) New trajectory points after SVD method; (d) New fractal dimension value calculation.

From the results, we can see that there is a huge different between contraction pattern and non-contraction pattern for their fractal dimension value after iteration of SVD processing. For both contraction pattern signal and non-contraction pattern signal, the final fractal dimension values become stable after iteration of SVD processing, but the final fractal dimension value of contraction pattern signal is much higher than that of non-contraction pattern signal whose final fractal dimension value is down to zero.

Contraction pattern signal extraction

From the result of EMG signal fractal dimension value calculation, we know that the contraction pattern signal and noise signal have different fractal dimension values. So, the noise can be differentiated from the original EMG signals and the signal patterns due to contraction will be clearly seen. Our work focuses on the analysis of the contraction patterns, because we think they are caused by some factors which can be used to differentiate the preterm labor and term labor signals. The final fractal dimension value along the signal is calculated and is illustrated in Fig.5.9(b) and the contraction patterns extraction is shown in Fig.5.9(c).

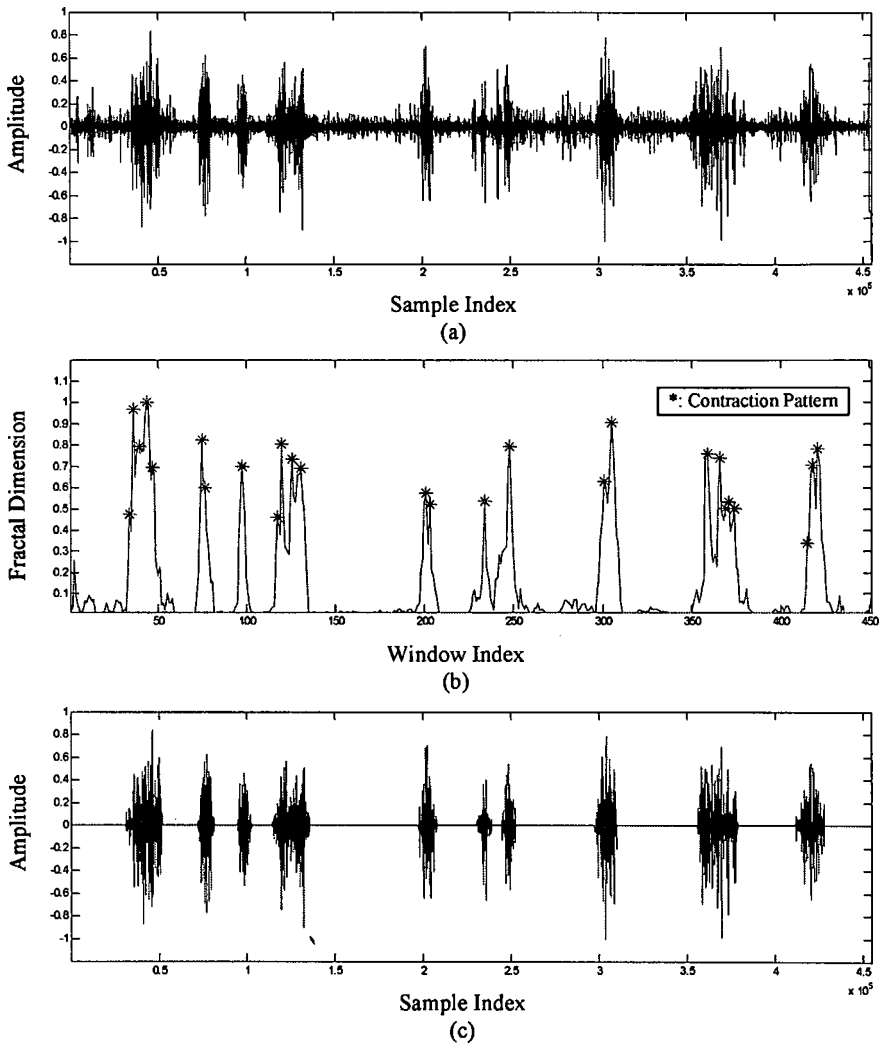


Figure 5.9: Contraction patterns extraction. (a) EMG signal after wavelet denoising; (b) Fractal dimension values along the signal; (c) Contraction patterns.

5.3.2 Wavelet packet transform

After contraction patterns were extracted from the original EMG signals, Wavelet packet transform is chosen for further analysis. In Section 5.3.2, we already used wavelet transform to process the signal, at that stage, the wavelet transform is used to get rid of the high frequency components of the signal. Wavelet packet transform is used to decompose the signal into different frequency ranges for further analysis. Wavelet packet method [109] is a generalisation of wavelet decomposition that offers a richer range of possibilities for revealing signal information. Unlike wavelet transform, wavelet packet transform is implemented by a basic two-channel filter bank which can be iterated over either a low-pass or a high-pass branch. So the information in high frequencies can be analyzed as well as that in low frequencies in wavelet packet transform. As a result, finer frequency bands can be gained by wavelet packet transform than by wavelet transform.

In wavelet analysis as shown in Fig.5.10(a), a signal is split into an approximation and a detail. The approximation is then itself split into a second-level approximation and detail, and the process is repeated. For n-level decomposition, there are $n + 1$ possible ways to decompose or encode the signal. In wavelet packet analysis, the details as well as the approximations can be split. This yields more than $2^{2^{n-1}}$ different ways to encode the signal. Wavelet packet decomposition is shown in Fig.5.10(b).

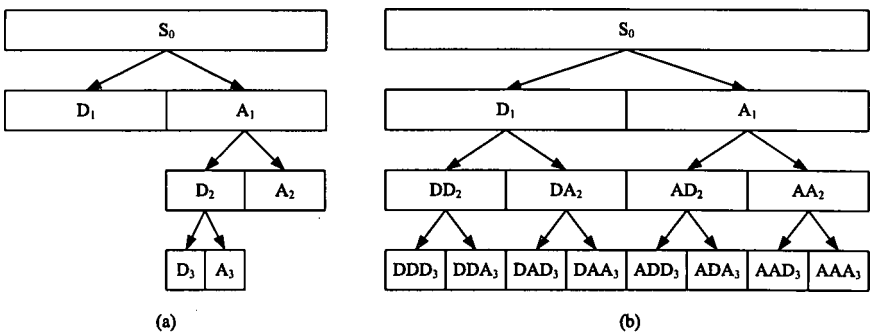


Figure 5.10: Wavelet and wavelet packet decomposition. (a) Wavelet decomposition; (b) Wavelet packet decomposition.

5.3.3 Wavelet packet energy

Given a finite energy signal whose scaling space is assumed as U_0^0, wavelet packet transform can decompose U_0^0 into small subspaces U_n^j in dichotomous way. When $s(t)$ is decomposed to nth resolution level with wavelet packet transform, the whole scaling space U_0^0 with frequencies in the interval $(0,2^{-1}f_s]$ is divided into 2^n subspaces with frequencies correspondingly in the interval $((n-1)2^{j-1}f_s, n2^{j-1}f_s], n = 1,2,\ldots,n$. The subsignal at U_j^{n-1}, the nth subspace on the jth level, can be reconstructed by:

$$s_j^n(t) = \sum_k D_k^{j,n}\psi_{j,k}(t) \tag{5.3.4}$$

where $D_k^{j,n}$ is the wavelet packet coefficients at U_j^{n-1}, $\psi_{j,k}(t)$ is the wavelet function. Thus, $s(t)$ can be rewritten as:

$$s(t) = \sum_{n=1}^{2^j} s_j^n(t) = \sum_{n=1}^{2^j}\sum_k D_k^{j,n}\psi_{j,k}(t) \tag{5.3.5}$$

Since the wavelet $\psi_{j,k}(t)$ is an orthogonal basis, the energy of the subsignal $s_j^n(t)$ is calculated by:

$$E_n = \sum_k |D_k^{j,n}|^2 \tag{5.3.6}$$

The total energy of $s(t)$ is:

$$E = \sum_n E_n \tag{5.3.7}$$

In consequence, wavelet packet energy (WPE) [110] which quantifies the probability distribution of the spectral energy is:

$$P_n = \frac{E_n}{E} \tag{5.3.8}$$

In our experiment, wavelet packet energy is calculated for each contraction pattern signal which is decomposed into 4th level using wavelet package transform and the first ten wavelet packet coefficients are selected for wavelet packet energy calculation. Fig.5.11 shows the results of wavelet packet energy calculation for each of contraction pattern signals.

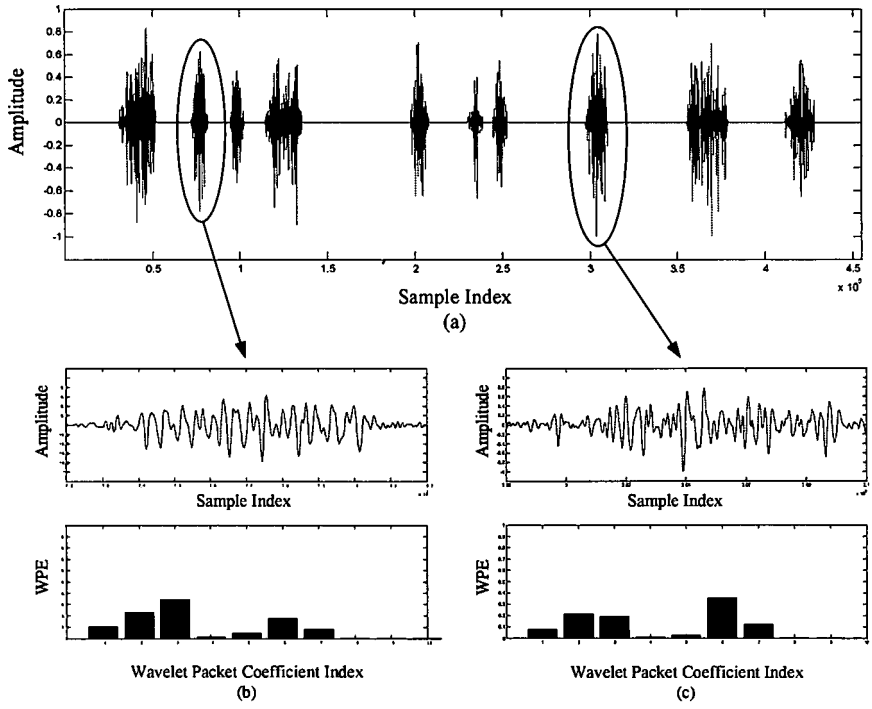


Figure 5.11: Wavelet packet energy calculation. (a) Contraction patterns; (b) and (c) Wavelet packet energy distribution for each of contraction pattern.

In order to find the general distribution rule of wavelet packet energy for every pattern of EMG signal, we evaluate the average wavelet packet energy of every EMG signal:

$$P_n = \frac{1}{l}\sum_{i=1}^{l} p_n^i \tag{5.3.9}$$

where p_n^i is WPE at the nth subspace of the ith subject and l is the total number of contraction patterns of every EMG signal. Fig.5.12 shows the result of average wavelet packet energy.

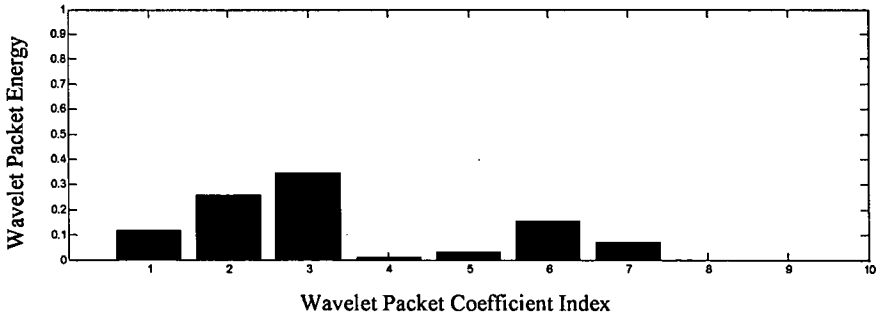


Figure 5.12: Average wavelet packet energy.

5.4 Signal classification

Some research work have been reported about EMG signal classification to help doctors differentiate the normal signals from abnormal signals [111, 112]. In this procedure, the artificial neural network (ANN) method is used [113, 114, 115].

From the result of wavelet packet energy calculation, we can see that the contraction pattern signal is decomposed into 10 sub-signals. For each contraction pattern signal, it can be represented by 10 wavelet packet energy values. For each EMG signal, it can be represented by 10 average wavelet packet energy values. So for the whole term labor signals and preterm labor signals, they form a wavelet packet energy matrix $X(n, 10)$ which can be used for the classification work, n is the number of data sets.

In our experimental data, there are total 39 sets of data, 11 PT signals and 28 TL signals. For each set of signal, there are 10 average wavelet packet energy distribution values. So the input of ANN is a matrix of 39×10.

We create a network with 10 input nodes, one hidden layer of 20 nodes and one output node. All neurons use the *tansig* transfer function, besides the output neuron which uses *logsig*. The training procedure is *trainlm*. Fig.5.13 shows the architecture of artificial neural network.

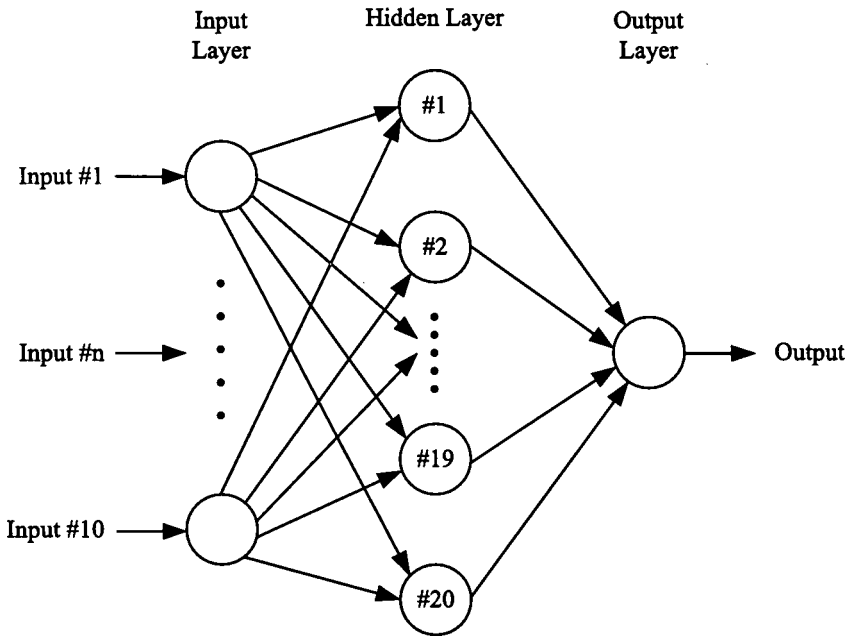


Figure 5.13: Architecture of artificial neural network.

All data sets are divided into two groups, one is for training and another is for testing. We use the Leave One Out (LOO) method that only one signal is taken out as the test signal and the rest of signal are put into the network to train. The network is trained to return 1 for the "TL"-class and 0 for the "PT"-class. Since there are only two classes we split this interval into two and let all values between 1 and 0.5 belong to the "TL"-class and between 0.5 and 0 to the "PT"-class. Finally, the trained network is used to classify the test signal. Table 5.1 shows the result of classification.

Table 5.1 Classification results

Index	File Name		Type	Test Result		
1	3rdApril03(1)	PT_B3C1	PT	0.9579	TL	Wrong
2	3rdJune03(1)	PT_B9C1	PT	0.0000	PT	Correct
3	3rdJune03(1)A	PT_B3C1	PT	0.0173	PT	Correct
4	10thDec03(2)	PT_B1C1	PT	0.6516	TL	Wrong
5	10thDec03(2)B	PT_B2C1	PT	0.9980	TL	Wrong
6	18thNov03(1)	PT_B1C2	PT	0.1699	PT	Correct
7	18thNov03(1)B	PT_B1C2	PT	0.9940	TL	Wrong
8	23rdJuly03(1)	PT_B8C2	PT	0.6910	TL	Wrong
9	23rdMay03(1)	PT_B6C1	PT	0.9552	TL	Wrong
10	27thFeb03(2)1G	PT_B1C1	PT	0.4312	PT	Correct
11	27thFeb03(2)2G	PT_B4C2	PT	0.9980	TL	Wrong
12	1stJuly03(1)	TL_B8C2	TL	0.7428	TL	Correct
13	3rdApril03(1)	TL_B3C1	TL	0.0042	PT	Wrong
14	4thJune04(1)	TL_B2C1	TL	0.7637	TL	Correct
15	4thJune04(2)	TL_B1C2	TL	1.0000	TL	Correct
16	6thApril04(1)	TL_B1C2	TL	0.9999	TL	Correct
17	8thMar04(1)	TL_B3C2	TL	0.9898	TL	Correct
18	11thMar03(1)2G	TL_B6C2	TL	0.9253	TL	Correct
19	11thMar03(1)G	TL_B5C1	TL	1.0000	TL	Correct
20	13thApril04(1)	TL_B1C1	TL	0.9932	TL	Correct
21	13thApril04(2)	TL_B1C1	TL	0.9118	TL	Correct
22	13thFeb03(1)G	TL_B3C1	TL	0.9987	TL	Correct
23	15thJan04(1)	TL_B1C2	TL	0.0000	PT	Wrong
24	15thJan04(2)	TL_B1C2	TL	0.7519	TL	Correct
25	17thDec03(1)	TL_B2C1	TL	1.0000	TL	Correct
26	19thMay04(1)	TL_B2C2	TL	0.5054	TL	Correct
27	21stJune04(1)	TL_B2C2	TL	0.9986	TL	Correct
28	23rdApril04(1)	TL_B1C1	TL	0.0413	PT	Wrong
29	24thApril03(1)	TL_B3C2	TL	0.4658	PT	Wrong
30	24thMar04(1)	TL_B3C1	TL	0.4609	PT	Wrong
31	25thFeb04(1)	TL_B1C1	TL	0.0041	PT	Wrong
32	26thApril04(2)	TL_B1C1	TL	0.8608	TL	Correct
33	26thJune03(1)	TL_B2C2	TL	0.9793	TL	Correct
34	27thFeb03(1)2P	TL_B6C2	TL	1.0000	TL	Correct
35	27thJan04(1)	TL_B1C1	TL	1.0000	PT	Wrong
36	30thDec03(1)	TL_B1C2	TL	0.0000	TL	Correct
37	30thDec03(1)B	TL_B1C2	TL	0.9946	TL	Correct
38	30thMay03(1)	TL_B6C1	TL	0.9301	TL	Correct
39	210104(1)	TL_B1C2	TL	1.0000	TL	Correct

From the table, we can see that the classification accuracy reaches 64.1%. Although the result is not good enough for classification, it is encouraging to continue the work under the conditions of getting more quality data.

Unfortunately, the funding to support a part-time researcher to collect data has been run out. It is impossible to obtain more organized test data. This study has indicated that further study using the methods should be continued. The work presented in this chapter developed a new method towards to uterine EMG signal processing.

5.5 Summary

There are three main steps in our procedure, signal preprocessing, signal feature extraction and signal classification. In signal preprocessing, the noise and high frequency components of the signal are eliminated by using threshold de-noising and wavelet de-noising methods. In the second step, each contraction pattern of the signal is extracted by calculating the fractal dimension values along the signal, then wavelet packet transform is used to decompose signal and the average wavelet packet energy of each EMG signal is calculated for feature extraction. At the last step, the artificial neural network method is used for the signal classification.

In the procedure, the main idea is that the difference between term labor signal and preterm labor signal can be the contraction patterns. So the analysis work is focused on the contraction patterns of the signal, try to extract some features for classification. From the result, we can see that the processing methods work well to separate the two kinds of signals. The classification accuracy of ANN is 64.1% which is encouraging, although it is still not good enough for this application. In this study, the data is from different patients observed at different time, so it is difficult to follow a patient from the beginning to the labor for the features extraction study. The initial study has shown some encouraging result, so the future work needs to get more organized and accurate experimental data from patients.

Chapter 6

Motion Detection Based On Accumulative Optical Flow and Double Background Filtering

Real-time detection of moving objects is very important for video surveillance. In this chapter, we present a novel real time motion detection procedure that is based on the integration of temporal difference, optical flow, double background filtering (DBF) and morphological processing methods to achieve enhanced performance. The temporal difference method is used to detect initial coarse motion areas for the optical flow calculation which is used for the real and accurate object motion detection. The DBF method is used to obtain and keep a stable background image to cope with variations on environmental changing conditions and is used to eliminate the background interference information and separate the moving object from it. The morphological processing methods are used and combined with the DBF to get the further improved results. The most attractive advantage of this procedure is that it does not need to learn the background model from hundreds of images and can handle quick image variations without prior knowledge about the object size and shape. The procedure has a high capability of anti-interference and preserves high accurate rate detection at the same time. It also needs less computation demands than other methods for real-time surveillance. The effectiveness of

the proposed procedure for motion detection is demonstrated in a simulation environment and the evaluation results are reported in this chapter.

6.1 Introduction

In recent years, motion detection has attracted a great interest from computer vision researchers due to its promising applications in many areas, such as video surveillance [116, 117], traffic monitoring [118] or sign language recognition. However, it is still in its early developmental stage and needs to improve its robustness when applied in a complex environment.

Several techniques for moving object detection have been proposed in [119, 120, 121, 122, 123], among them the three representative approaches are temporal difference, background subtraction and optical flow. Temporal difference based on frame differences, attempts to detect moving regions by making use of the difference of the consecutive frames (two or three) in a video sequence. This method is highly adaptive to dynamic environments, but generally does a poor job of extracting the complete shapes of certain types of moving objects. Background subtraction is the most commonly used approach in presence of still cameras. The principle of this method is to use a model of the background and compare the current image with a reference. In this way the foreground objects present in the scene are detected. The method of statistical model based on the background subtraction is flexible and fast, but the background scene and the camera are required to be stationary when this method is applied. Optical flow is an approximation of the local image motion and specifies how much each image pixel moves between adjacent images. It can achieve success of motion detection in the presence of camera motion or background changing. According to the smoothness constraint, the corresponding points in the two successive frames should not move more than a few pixels. For an uncertain environment, this indicates that the camera motion or background changing should be relatively small. The method based on optical flow is complex, but it can detect the motion accurately even without knowing the background. The main idea in the procedure proposed in the thesis is to integrate the advantages

of these three methods.

In this procedure, the temporal difference method, optical flow method and double background filtering method with morphological processing is integrated. The main goal of this procedure is to separate the background interference and foreground information effectively and detect the moving object accurately. In the first step, temporal difference method is used to detect the coarse motion object area for the optical flow calculation. Secondly, the DBF method is used to obtain and keep a stable background image to address variations on environmental changing conditions and is used to eliminate the background interference and separate the moving object from it. The morphological processing methods are used and combined with DBF to gain better results. Different from the paper [26], a new improved strategy is proposed which not only improves the capability of detecting the object in motion, but also reduces computation demands.

6.2 Overview of the motion detection procedure

The procedure is depicted in the flowchart of Fig.6.1.

As can be seen, the whole procedure is comprised of four steps: (1) Temporal difference method, which is used to detect the initial coarse object motion area; (2) Optical flow detection, which is used based on the result of (1) to calculate optical flow for each frame; (3) Double background filtering method with morphological processing, which is used to eliminate the background interference and keep the foreground moving information; (4) Motion area detection, which is used to detect the moving object and give the alarming in time.

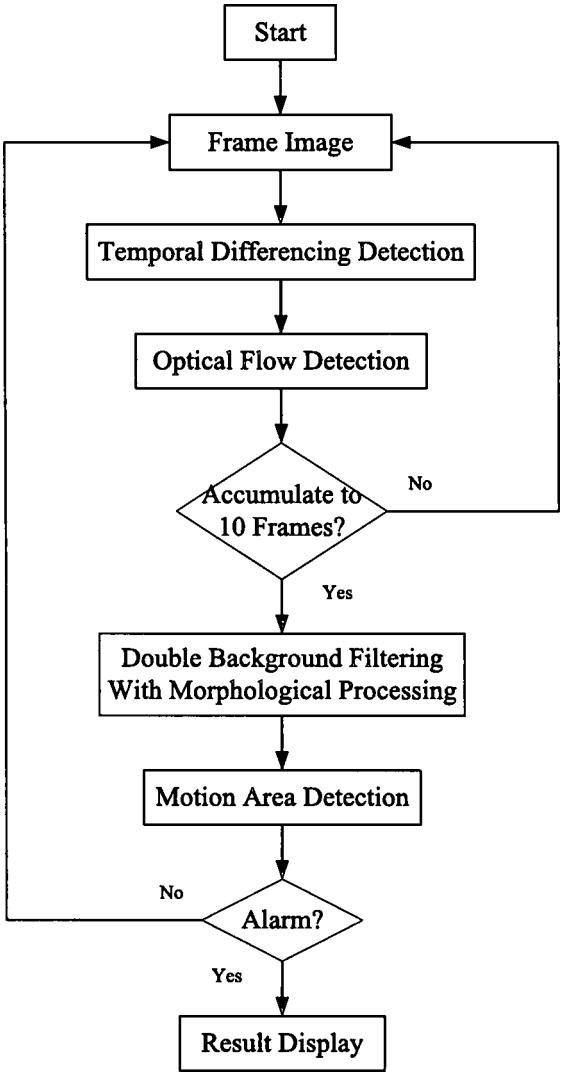
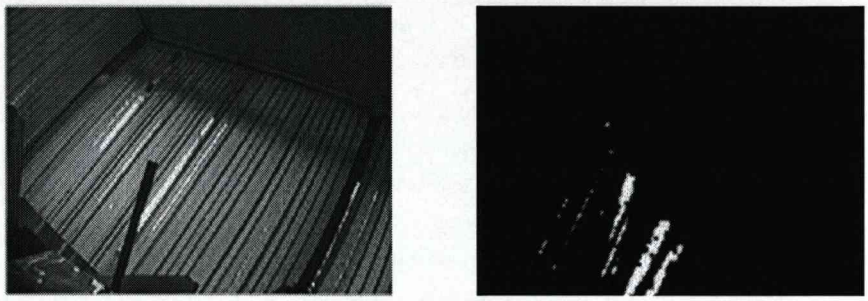


Figure 6.1: Flowchart of motion detection procedure

The final processing result is a binary image in which the background area and moving object area are shown as white color, the other areas are shown in black color. The experimental result in Fig.6.2 presents a set of images to help in understanding the processes achieved in the present method.



(a) (b)

Figure 6.2: Result of motion detection; (a)Original image with moving object; (b)Result of motion detect with moving object.

In this chapter, the effectiveness of the proposed procedure for motion detection is demonstrated for a simulation environment. The procedure is implemented in Matlab. The size of the input video image is 320×240 pixels and the sample rate is 25 frames per second. In our experiment, the simulation environment is the background of a column strip curtain which is swing caused by nature winds.

6.3 Temporal difference detection method

Temporal difference [124, 125] is based on frame difference which attempts to detect moving regions by making use of the difference of two consecutive frames in a video sequence. It is computed based on the gray level matching of pixels. Namely, the difference between two consecutive frames is considered as pixels movements from a position in the current frame to another position in the next frame. However, it also can be considered as the state transition of pixels. For example, in a 256 level gray image, each pixel has 256 states. Along with time, the pixels state would transit from one to another arbitrarily. The

diversity of state at each pixel indicates the intensity of motion at its position. This method is highly adaptive to static environment. So temporal difference is good at providing initial coarse motion areas.

In this method, the two subsequent 256 level gray images at time t and $t+1$, $I(x, y, t)$ and $I(x, y, t+1)$, are selected and the difference between images is calculated by setting the adaptive threshold to get the region of changes. The adaptive threshold T_d can be derived from image statistics. In order to detect cases of slow motion or temporally stopped objects, a weighted coefficient with a fixed weight for the new observation is used to compute the temporal difference image $I_d(x, y, t)$ as shown in following equations:

$$I_d(x, y, t+1) = \begin{cases} 255, & \text{if}(I_a(x, y, t+1) > T_d) \\ 0, & \text{otherwise} \end{cases} \tag{6.3.1}$$

and

$$\begin{aligned} I_a(x, y, t+1) = (1-w) I_a(x, y, t) \\ + w|I(x, y, t+1) - I(x, y, t)| \end{aligned} \tag{6.3.2}$$

where w is a real number between 0 and 1 which describes the temporal range for different images. $I_a(x, y, t-1)$ is initialized to an empty image, T_d is the threshold. In this procedure, after experimental test, the adaptive threshold T_d in (6.3.1) is set as three times of mean value of $I_a(x, y, t+1)$ in (6.3.2) can get best result.

$$T_d = 3 \times \text{mean}(I_a(x, y, t+1)) \tag{6.3.3}$$

And w in (6.3.2) is set as 0.5 which is always used in the most of situations for all the results. Figure.6.3 shows the results of temporal difference method under a simulation environment which has a static background of our laboratory.

From the results, we can see that the temporal difference is a simple method for detecting moving objects in a static environment and the adaptive threshold T_d can restrain the noise very well.

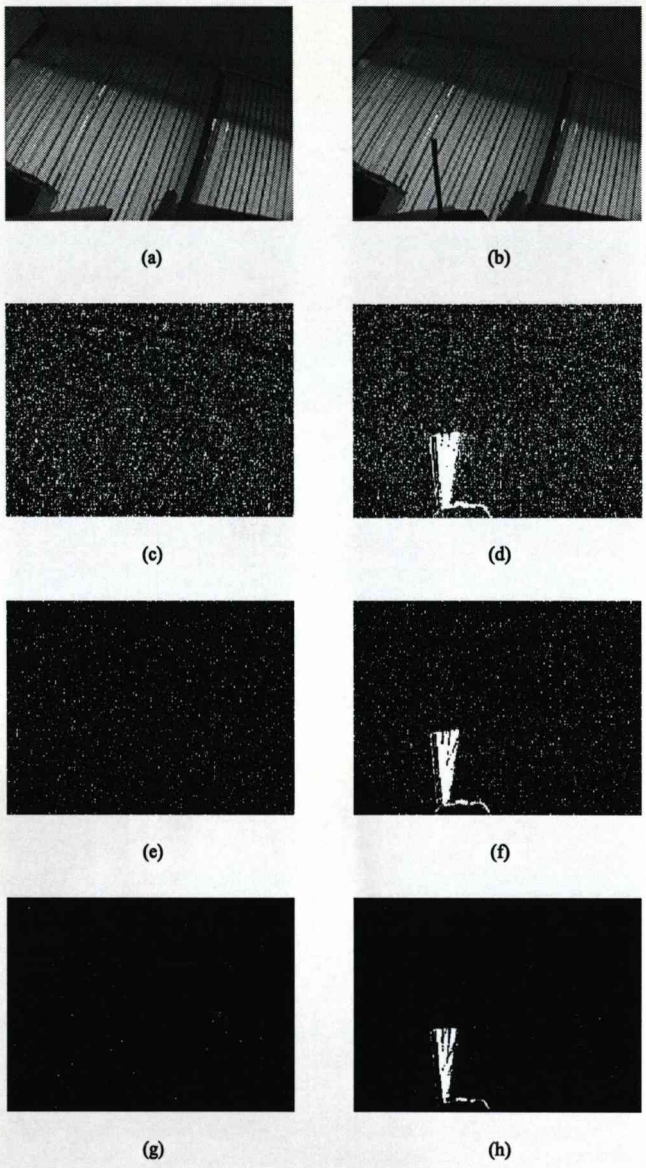


Figure 6.3: Results of temporal difference method. (a) Background image; (b) Background image with moving object; (c) The result of temporal difference for (a), $T_d = 1 \times mean(I_a(x, y, t+1))$; (c) The result of temporal difference for (b), $T_d = 1 \times mean(I_a(x, y, t+1))$; (c) The result of temporal difference for (a), $T_d = 2 \times mean(I_a(x, y, t+1))$; (c) The result of temporal difference for (b), $T_d = 2 \times mean(I_a(x, y, t+1))$; (c) The result of temporal difference for (a), $T_d = 3 \times mean(I_a(x, y, t+1))$; (c) The result of temporal difference for (b), $T_d = 3 \times mean(I_a(x, y, t+1))$.

6.4 Optical Flow Detection Method

Temporal difference is the simplest method to extract moving objects. But if the background is dynamic, it will become very sensitive to any movement and is difficult to differentiate the true and false movement. Compared with the temporal difference method, optical flow method is more accurate to the moving pixel, but its computation procedure is more complicated and requires more computing time. So in this procedure, the advantages of both methods are taken, firstly, the temporal difference method is used to detect possible motion area, secondly, the optical flow is calculated based on the result of temporal difference. The purpose of this integration way is to reduce the computation demands and keep the useful information at the same time.

Optical flow is a concept which is close to the motion of objects within a visual representation. The term optical flow denotes a vector field defined across the image plane. Optical flow calculation is a differential method for motion estimation. Such a method is to calculate the motion between two image frames which are taken at an interval t at every pixel position. Estimating the optical flow is useful in pattern recognition, computer vision, and other image processing applications [126, 127, 128]. In this procedure, an optical flow method entitled Lucas-Kanade is introduced. The whole computation process of optical flow is explained below and the corresponding Matlab code can be found in Appendix E.3.

6.4.1 Lucas-Kanade method

To extract a 2D motion field, Lucas-Kanade method is often employed to compute optical flow because of its accuracy and efficiency. Barron [129] compared the accuracy of different optical flow techniques on both real and synthetic image sequences. They found that the most reliable one was the first-order, local differential method of Lucas and Kanade. Liu [130] studied the accuracy and efficiency trade-offs in different optical flow algorithms. Liu's study has been focused on the motion algorithm implementations in real world tasks. Their results showed that Lucas Kanade method is pretty fast. Galvin

[131] evaluated eight optical flow algorithms. The Lucas-Kanade method consistently produces accurate depth maps, and has a low computational cost and good noise tolerance.

The Lucas-Kanade method [132] is trying to calculate the motion between two image frames which are taken at time t and $t+\delta t$ for every pixel position. As a pixel at location (x, y, t) with intensity $I(x, y, t)$ will have moved by δx, δy and δt between the two frames, the following image constraint equation can be given:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{6.4.1}$$

Assuming that the movement is small enough, the image constraint at $I(x, y, t)$ with Taylor series can be derived to give:

$$\begin{aligned} I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) \\ + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + H.O.T \end{aligned} \tag{6.4.2}$$

where H.O.T. means those higher order terms, which are small enough to be ignored. From (6.4.1) and (6.4.2), the following can be obtained:

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0 \tag{6.4.3}$$

or:

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t}\frac{\delta t}{\delta t} = 0 \tag{6.4.4}$$

which will result in,

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t}V_t = 0 \tag{6.4.5}$$

where V_x and V_y are the x and y components of the velocity or optical flow of $I(x, y, t)$ and $\partial I/\partial x$, $\partial I/\partial y$ and $\partial I/\partial t$ are the derivatives of the image at (x, y, t) in the corresponding directions.

Equation (6.4.5) is called the optical flow constraint equation since it expresses a constraint on the components V_x and V_y of the optical flow. The optical flow constraint equation can be rewritten as:

$$I_x V_x + I_y V_y = -I_t \tag{6.4.6}$$

or:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = -I_t \tag{6.4.7}$$

We wish to calculate V_x and V_y, but unfortunately the above constraint gives us only one equation for two unknowns, so this is not enough by itself. To find the optical flow, another set of equations is needed which should be given by some additional constraints. The solution as given by Lucas and Kanade is a non-iterative method which assumes a locally constant flow.

The Lucas-Kanade algorithm assumes that motion vectors in a any given region do not change but merely shift from one position to another. Assuming that the flow (V_x, V_y) is a constant in a small window of size $m \times m$ with $m > 1$, which is centered at (x, y) and numbering the pixels as $1 \ldots n$, a set of equations can be derived:

$$\begin{aligned} I_{x_1} V_x + I_{y_1} V_y &= -I_t \\ I_{x_2} V_x + I_{y_2} V_y &= -I_t \\ &\vdots \\ I_{x_n} V_x + I_{y_n} V_y &= -I_t \end{aligned} \tag{6.4.8}$$

With (6.4.8), there are more than three equations for the three unknowns and thus the system is over-determined. Hence:

$$\begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = -I_t \tag{6.4.9}$$

or:

$$A\vec{v} = -b \tag{6.4.10}$$

To solve the over-determined system of equations, the least squares method is used:

$$A^T A\vec{v} = A^T(-b) \tag{6.4.11}$$

$$\vec{v} = (A^T A)^{-1} A^T(-b) \tag{6.4.12}$$

or:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum I_{x_i}^2 & \sum I_{x_i} I_{y_i} \\ \sum I_{x_i} I_{y_i} & \sum I_{x_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{x_i} I_{t_i} \\ -\sum I_{y_i} I_{t_i} \end{bmatrix} \tag{6.4.13}$$

with the sums running from $i = 1$ to n. And there is a limit condition for the calculation of motion vector in (6.4.13) as:

$$A^T A = \begin{bmatrix} \sum I_{x_i}^2 & \sum I_{x_i} I_{y_i} \\ \sum I_{x_i} I_{y_i} & \sum I_{x_i}^2 \end{bmatrix} \tag{6.4.14}$$

Equation (6.4.14) must be an invertible matrix, which means that the optical flow can be found by calculating the derivatives of the image in all three dimensions: x-direction, y-direction and t-direction. One of the characteristics of the Lucas-Kanade algorithm is that it does not yield a very high density of flow vectors, i.e. the flow information fades out quickly across motion boundaries and the inner parts of large homogenous areas show little motion. The advantage for the method is its accuracy and robustness of detection in presence of noise.

6.4.2 Simplified Calculation

The theoretical calculation procedure of the Lucas-Kanade method is explained in the above section, but for the requirement of practical application, some operation characteristics between matrices can be used to simplify the complexity of calculation. For the calculation of invertible matrix in (6.4.13), the companion matrix method can be used:

$$M^{-1} = \frac{M^*}{|M|} = \frac{1}{\det M} \begin{bmatrix} M[1][1] & -M[1][0] \\ -M[0][1] & M[0][0] \end{bmatrix} \tag{6.4.15}$$

where M^* is the companion matrix of M and $|M|$ is the determinant of M.

6.4.3 Gradient Operator

From the operation expression of optical flow, the estimation of the gradients for x-direction, y-direction and t-direction, has a great influence on the final results of optical flow calculation. The most common gradient operators used in optical flow calculation are Horn, Robert, Sobel, Prewitt, Barron and so on. In an ordinary way, the calculation of gradient on both t and x directions uses the same template. For different operators, the number of frames required for calculation of the time gradient is different. For example, the Horn operator needs two frames and Barron operator needs five frames at least. In this step, a better 3D Sobel operator is used which was proposed in [133]. This operator uses three different templates to do the convolution calculation for three frames in a row along the directions of x, y and t and to calculate the gradient along three directions for central pixels of the template in the middle frame. Fig.6.4 shows the operators.

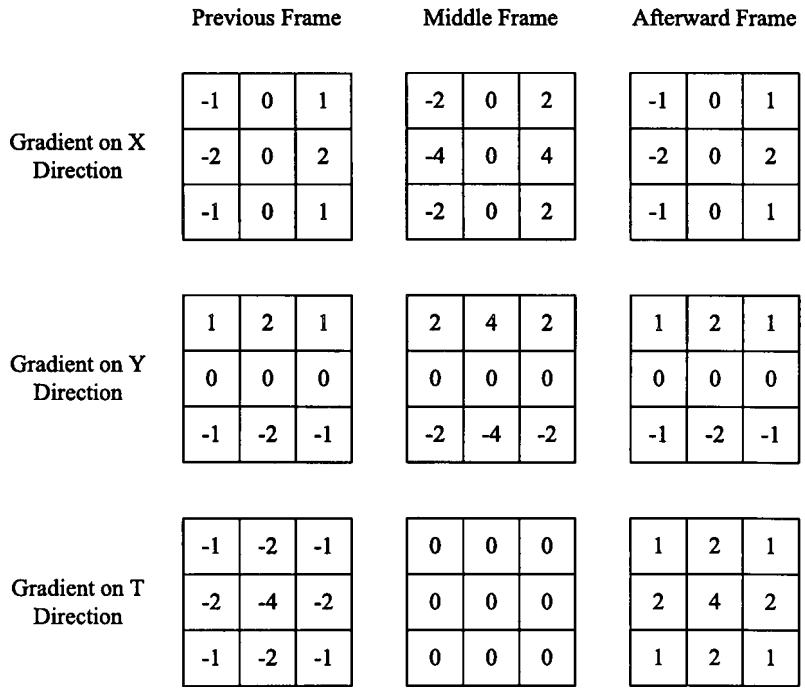


Figure 6.4: 3D Sobel operator for optical flow calculation

6.4.4 Display Results of Optical Flow Detection

The optical flow information for every frame of an image is calculated. As shown in Fig.6.5, the optical flow of frames $I_t, I_{t+1}, \ldots, I_{t+n}$ in a period time $[t, t+n]$ are represented as $F_1, F_2, \ldots, F_n$.

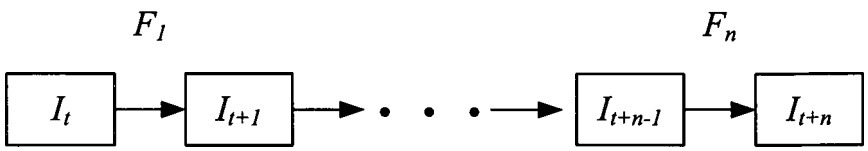


Figure 6.5: Diagram of optical flow calculation

The result of optical flow is shown as a binary image and the adaptive threshold is selected to distinguish the moving pixel from the still pixel. The pixels whose optical flow values are greater than the threshold will be considered as moving pixels and are shown in white. The optical flow value and the adaptive threshold formulaes that we used can be written as:

$$F_n(i,j) = \sqrt{V_x^2(i,j) + V_y^2(i,j)} \tag{6.4.16}$$

and

$$FD_n(i,j) = \begin{cases} 255, & \text{if } F_n(i,j) > T \\ 0, & \text{Otherwise} \end{cases} \tag{6.4.17}$$

and

$$T = \text{median}(F_n(i,j) > 0) \tag{6.4.18}$$

where $F_n(i,j)$ is the optical flow value, $FD_n(i,j)$ is the result of optical flow detection and the adaptive threshold T is select as median value of $F_n(i,j)$ whose value is above 0. Fig.6.6 shows the results of optical flow which is calculated based on the result of temporal difference. The simulation environment is a dynamical background which consists of swinging column bar curtains caused by winds.

From the results, we can see that the optical flow with adaptive threshold based on temporal difference result reserves the information of moving object

very well, but because of the background interference of the image, the real object movement still can not be separated from the background. So the method of double background filtering with morphological processing is introduced in the next section to deal with this problem.

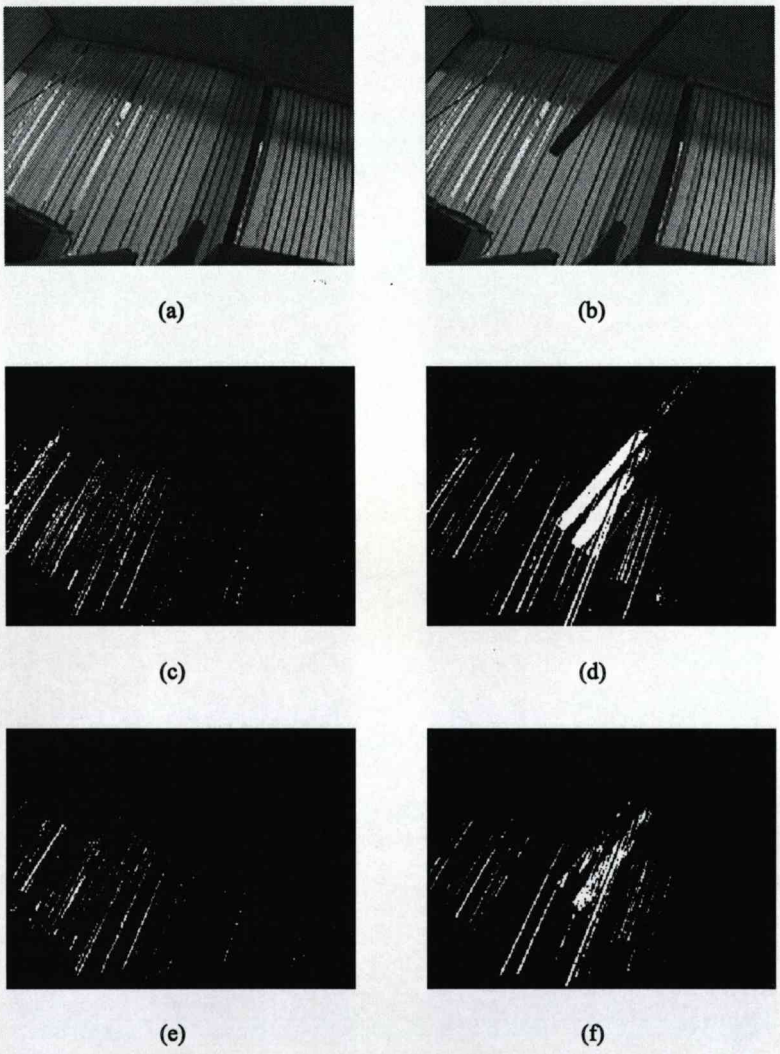


Figure 6.6: Results of optical flow based on temporal difference result. (a) Background image; (b) Background image with moving object; (c) Result of temporal difference for (a); (d) Result of temporal difference for (b); (e) Result of optical flow for (c); (f) Result of optical flow for (d).

6.5 Double background filtering with morphological processing

By using the optical flow method, two types of optical flow information are obtained, which are the interference information of image background and the information of image pixel with any possibility of real object movement. In the real situation, because of the environment such as light, vibration and etc., the interference information of the background still can be detected. Sometimes it is difficult for the real object movement to be differentiated from the background interference. In this section, the method of DBF with morphological processing is used to get rid of the background interference and separate the moving object from it. Firstly, the DBF method and its corresponding results are discussed. Then the morphological processing methods are introduced and the improved results are also demonstrated.

6.5.1 Double background filtering

In this chapter, a novel approach is developed to update the background. This approach is based on a double background principle [134, 135], long-term background and short-term background. For the long-term background, the background interference information which has happened in a long time is saved. For the short-term background, the most recent changes are saved. These two background images are modified to adequately update the background image and to detect and correct abnormal conditions.

During practical tests, we found that although the optical flow of background interference can be detected without moving object, it is relatively stable for some specific areas on the image and the amount of this optical flow doesn't change very much. For the area where the moving object appears, the amount of optical flow must change significantly in the specific area. According to these characteristics, the moving object should be easily detected if the information of the background and foreground can be separated. Fig.6.7 shown below explains the double background filtering method in a tabular way.

Processing	First five frames optical flow accumulation					Two Unused Frames		Last three frames optical flow accumulation		
Frame Index	1	2	3	4	5	6	7	8	9	10
Aim	To stable the optical flow information of background interference					To separate background and foreground information		To detect moving object		

Figure 6.7: Double background filtering method

The Double Background Filtering (DBF) method consists of four steps:

(1) **Optical Flow Accumulation For First Five Frames:** The optical flow information of the first five frames is accumulated for saving the optical flow information of the background interference. Let A^5 be the accumulation matrix, which is defined with the same size as the video images and set the initial value as zeros. To compute this matrix the formula below is applied:

$$A^5(i,j) = \begin{cases} A^5(i,j)+1, & \text{if } F_k(i,j) = 255 \\ A^5(i,j), & \text{if } F_k(i,j) = 0 \end{cases} \quad k = 1,2,3,4,5 \qquad (6.5.1)$$

(2) **Optical Flow Accumulation For Last Three Frames:** The optical flow information of the last three frames is accumulated for moving object detection. Let A^3 be the accumulation matrix and computed as follow:

$$A^3(i,j) = \begin{cases} A^3(i,j)+1, & \text{if } F_k(i,j) = 255 \\ A^3(i,j), & \text{if } F_k(i,j) = 0 \end{cases} \quad k = 8,9,10 \qquad (6.5.2)$$

(3) **Background Filtering:** By comparing the results of steps (1) and (2) and eliminating the overlap optical flow, the rest should be the optical flow which represents the real movement. The procedure to detect whether a pixel $B(i,j)$ belongs to an object with salient motion is described as follows

$$B(i,j) = \begin{cases} 0, & \text{if } A^5(i,j) > 0 \text{ and } A^3(i,j) > 0 \\ 255, & \text{if } A^5(i,j) = 0 \text{ and } A^3(i,j) > 0 \end{cases} \qquad (6.5.3)$$

(4) **Background Updating:** This step is an updating function of the new value of the accumulation matrix, both A^5 and A^3 are set to zero, with the new video frame input, the four steps above are then repeated.

In this method, there are always two unused frames during the process, the purpose of this is to separate the background and moving object effectively. When the moving object appears in the last three frames, the information of

moving object will not be lost while the background is updating. Fig.6.8 below shows the result of double background filtering method.

Fig.6.8(a) is the background image; Fig.6.8(b) is the background image with moving object; Fig.6.8(c) is the result of optical flow for (a); Fig.6.8(d) is the result of optical flow for (b); Fig.6.8(e) is the result of first five frames optical flow accumulation for (c); Fig.6.8(f) is the result of first five frames optical flow accumulation for (d); Fig.6.8(g) is the result of last three frames optical flow accumulation for (c); Fig.6.8(h) is the result of last three frames optical flow accumulation for (d); Fig.6.8(i) is the result of double background filtering for (c); Fig.6.8(j) is the result of double background filtering for (d);

From the results, we can see that for the background without moving object, the background interference can not be eliminated completely and for the background with moving objects, although the moving object area can be detected, the background interference still exit. So how to get rid of the background interference and preserve the information of moving object at the same time is most important challenge we are facing. The morphological processing method is introduced in next section to solve this problem.

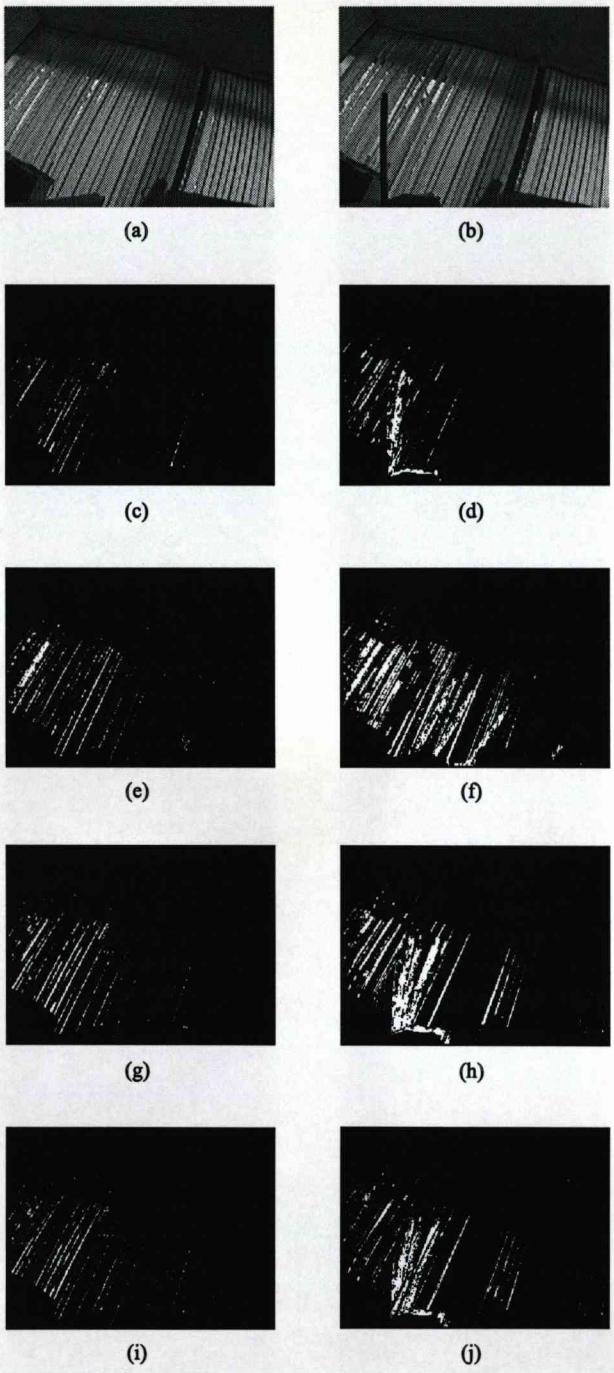


Figure 6.8: Results of double background filtering method.

6.5.2 Morphological image processing

Morphological image processing is a collection of techniques for digital image processing based on mathematical morphology which is a nonlinear approach that is developed based on set theory and geometry [136]. Morphological image processing techniques are widely used in the area of image processing, machine vision and pattern recognition due to its robustness in preserving the main shape while suppressing noise. When acting upon complex shapes, morphological operations are able to decompose them into meaningful parts and separate them from the background, as well as preserve the main shape characteristics. Furthermore, the mathematical calculation involved in mathematical morphology includes only addition, subtraction and maximum and minimum operations without any multiplication and division. There are two fundamental morphological operations which are dilation and erosion and many of the morphological algorithms are based on these two primitive operations.

Dilation of the set A by set B which is usually called as structure element, denoted by $A \oplus B$, is obtained by first reflecting B about its origin and then translating the result by x. All x such that A and reflected B translated by x that have at least one point in common form the dilated set.

$$A \oplus B = \{x|(\hat{B})_x \cap A \neq \varnothing\} \tag{6.5.4}$$

where $\hat{B}$ denotes the reflection of B and $(B)_x$ denotes the translation of B by x. Thus, dilation of A by B expands the boundary of A. Erosion of A by B, denoted by $A \ominus B$, is the set of all x such that B translated by x is completely contained in A.

$$A \ominus B = \{x|(\hat{B})_x \subseteq A\} \tag{6.5.5}$$

Thus, erosion of A by B shrinks the boundary of A. In this procedure, we also use two other important morphological operations which are defined in terms of erosion and dilation: opening and closing. A is said to be opened by B if the erosion of A by B is followed by a dilation of the result by B.

$$A \circ B = (A \ominus B) \oplus B\} \tag{6.5.6}$$

Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. Similarly, A is said to be closed by B if A is first dilated by B and the result is then eroded by B. Thus,

$$A \bullet B = (A \oplus B) \ominus B\} \tag{6.5.7}$$

Closing also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

In this procedure, three morphological operators, dilation, opening and closing are used. Dilation is applied firstly to the image with the accumulation optical flow of the first five frames. The dilation operator expands the area of background interference to make it eliminated efficiently in the third step of DBF method. The other two operators, opening and closing, are applied to the image with the accumulation optical flow of last three frames. The opening operator is used first to eliminate the noise which consists of isolated points and closing operator is used immediately after filling up the holes and gaps. The structure element in both operations is $SE = \{1,1,1;1,1,1;1,1,1\}$. Fig.6.9 shows the results of DBF with morphological processing.

Fig.6.8(a) is the background image; Fig.6.8(b) is the background image with moving object; Fig.6.8(c) is the result of optical flow for (a); Fig.6.8(d) is the result of optical flow for (b); Fig.6.8(e) is the result of first five frames optical flow accumulation after dilation for (c); Fig.6.8(f) is the result of first five frames optical flow accumulation after dilation for (d); Fig.6.8(g) is the result of last three frames optical flow accumulation after opening and closing for (c); Fig.6.8(h) is the result of last three frames optical flow accumulation after opening and closing for (d); Fig.6.8(i) is the result of double background filtering with morphological processing for (c); Fig.6.8(j) is the result of double background filtering with morphological processing for (d);

From the results, we can see that the DBF method with morphological processing can preserve the moving object area very well and eliminate the background interference completely. The result of this processing can be very helpful for further motion area detection.

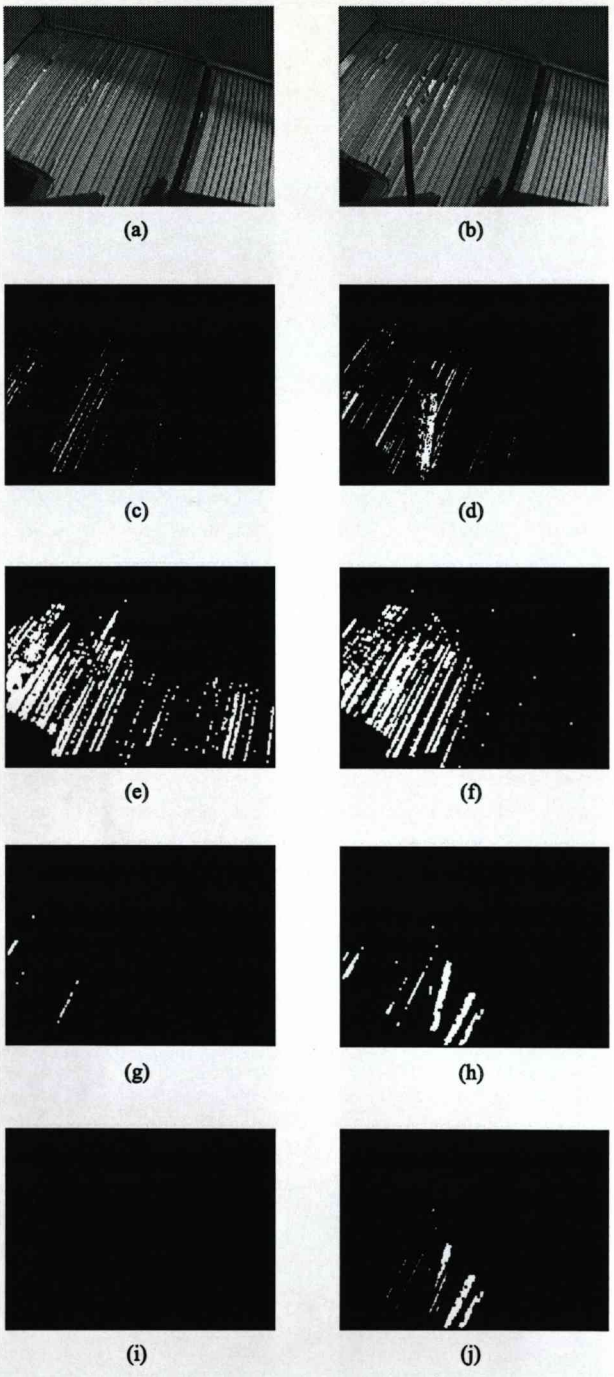


Figure 6.9: Results of double background filtering method with morphological processing.

6.6 Motion area detection

After applying the step of DBF method with morphological processing, the optical flow information of the background interference should be eliminated and only the optical flow information of real moving object is left. During the experimental test, we find that the appearance of a moving object can make a big influence on the instantaneous rate of change between the foreground motion information and the accumulative background optical flow information. In this procedure, we use the result of DBF method with morphological processing as the foreground motion information FM. Because the result of DBF method with morphological processing comes from the last three frames accumulative optical flow information so that the result of the first seven frames accumulative optical flow information is used as the accumulative background optical flow information $ABOF_7$. So we can define the instantaneous rate of change for the moving object appearance IRC_A as follows:

$$IRC_A = \frac{FM}{ABOF_7} \times 100\% \tag{6.6.1}$$

Fig.6.10 and Fig.6.11 show the result of IRC_A during the experiments for the background with and without moving object, respectively.

From the results, we can see that, for the background without moving object, IRC_A has a small value with little changing. But if there is moving object appearance, the value of IRC_A will increase sharply and last for several frames time. By taking advantage of this feature, we can use this IRC_A value to detect the movement of moving object and give the alarm without delay. In our experiment, the alarm threshold T_a is set as 0.25 and the abnormity alarm will occur whenever the IRC_A value is above T_a. It can be described as follows:

$$A = \begin{cases} 1, & \text{if } IRC_A > T_a \\ 0, & \text{Otherwise} \end{cases} \tag{6.6.2}$$

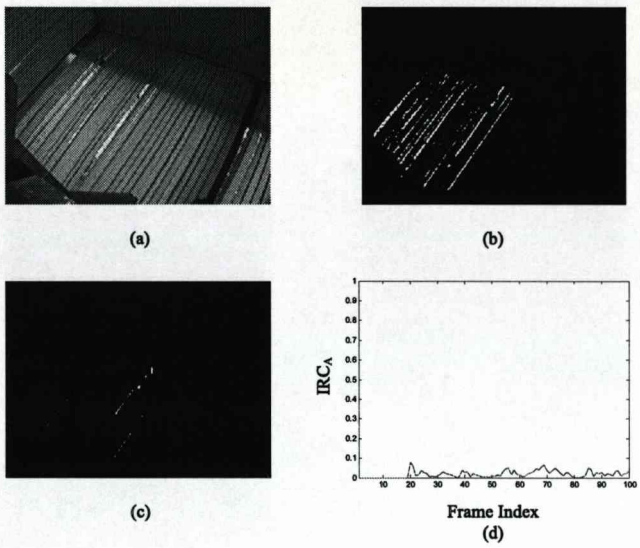


Figure 6.10: Results of IRC_A for the background without moving object. (a) Background image; (b) Result of optical flow; (c) Result of DBF method; (d) Result of IRC_A.

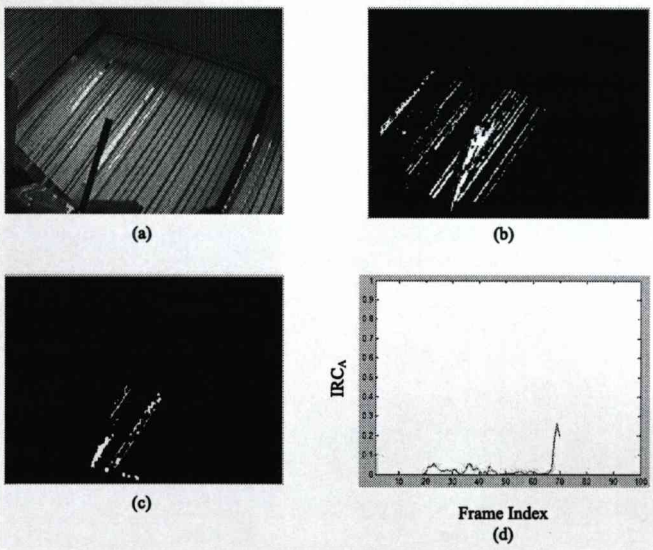


Figure 6.11: Results of IRC_A for the background without moving object. (a) Background image; (b) Result of optical flow; (c) Result of DBF method; (d) Result of IRC_A.

6.7 Experimental Results

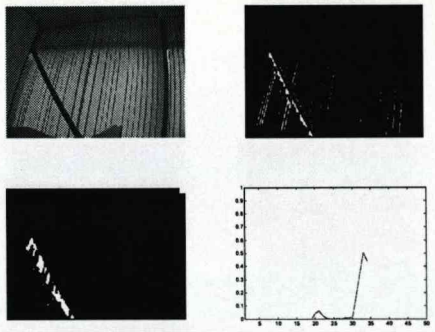
To test the sensitivity of the procedure to the speed of the moving object and the distances of the object to the camera, a serried of experiments has been conducted. In the experiments, the background scenario is kept the same - the vibrated strip curtains caused by winds. Five different speeds of the moving object (very slow, slow, normal, fast and very fast) and six different distances between camera and the moving object (20cm, 40cm, 60cm, 100cm, 150cm and 200cm) are chosen and tested. The results are shown in Fig.6.13, Fig.6.14, Fig.6.15, Fig.6.16, Fig.6.17 and Fig.6.18, respectively, and the figure numbering for the whole set of testing results is listed in Fig.6.12.

Speed

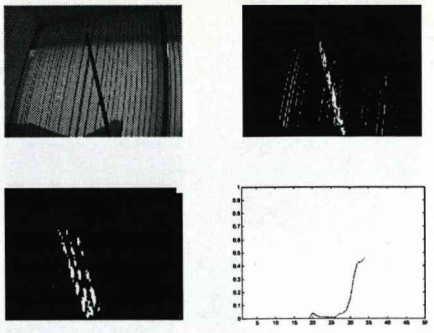
Results	Very Slow	Slow	Normal	Very Fast	Fast
20cm	Figure 6.13(a)	Figure 6.13(b)	Figure 6.13(c)	Figure 6.13(d)	Figure 6.13(e)
40cm	Figure 6.14(a)	Figure 6.14(b)	Figure 6.14(c)	Figure 6.14(d)	Figure 6.14(e)
60cm	Figure 6.15(a)	Figure 6.15(b)	Figure 6.15(c)	Figure 6.15(d)	Figure 6.15(e)
100cm	Figure 6.16(a)	Figure 6.16(b)	Figure 6.16(c)	Figure 6.16(d)	Figure 6.16(e)
150cm	Figure 6.17(a)	Figure 6.17(b)	Figure 6.17(c)	Figure 6.17(d)	Figure 6.17(e)
200cm	Figure 6.18(a)	Figure 6.18(b)	Figure 6.18(c)	Figure 6.18(d)	Figure 6.18(e)

Distance

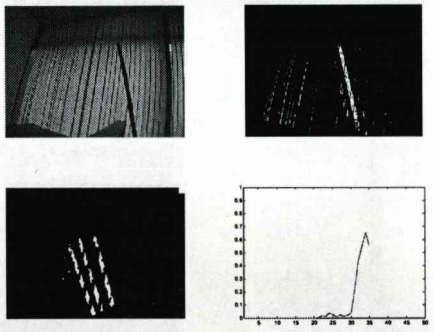
Figure 6.12: Testing results for the procedure.



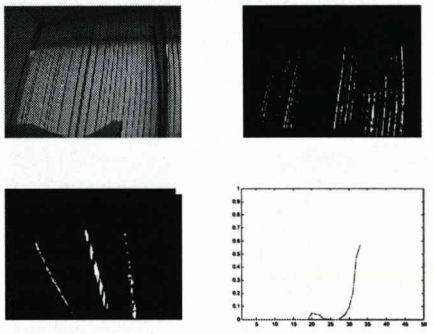
(a) Speed: very slow; Distance: 20cm



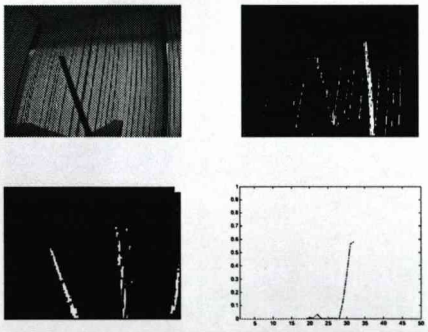
(b) Speed: slow; Distance: 20cm



(c) Speed: normal; Distance: 20cm

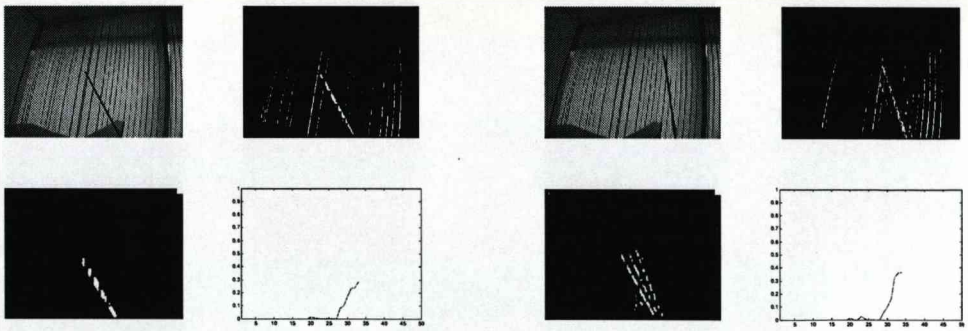


(d) Speed: fast; Distance: 20cm



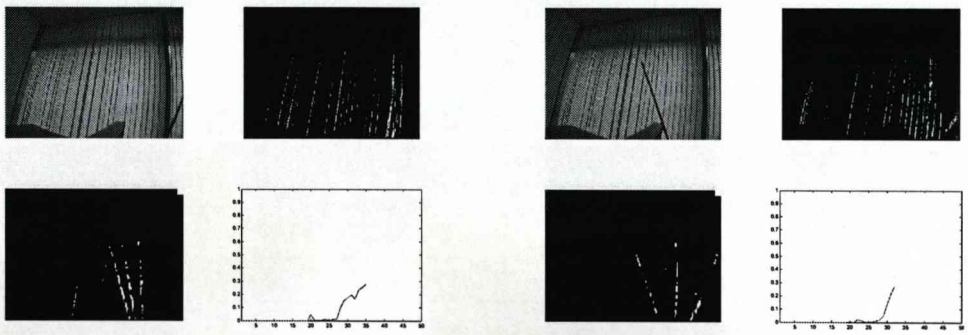
(e) Speed: very fast; Distance: 20cm

Figure 6.13: Experimental results under different speeds, distance is 20cm.



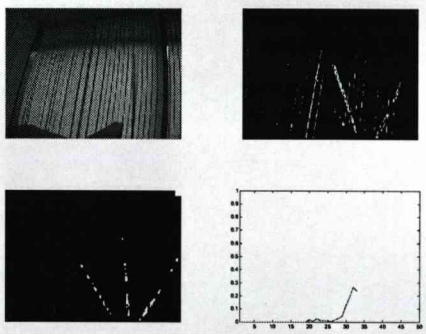
(a) Speed: very slow; Distance: 40cm

(b) Speed: slow; Distance: 40cm



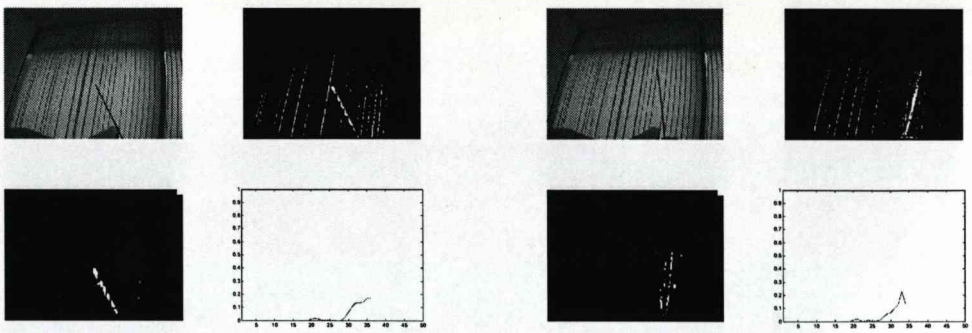
(c) Speed: normal; Distance: 40cm

(d) Speed: fast; Distance: 40cm



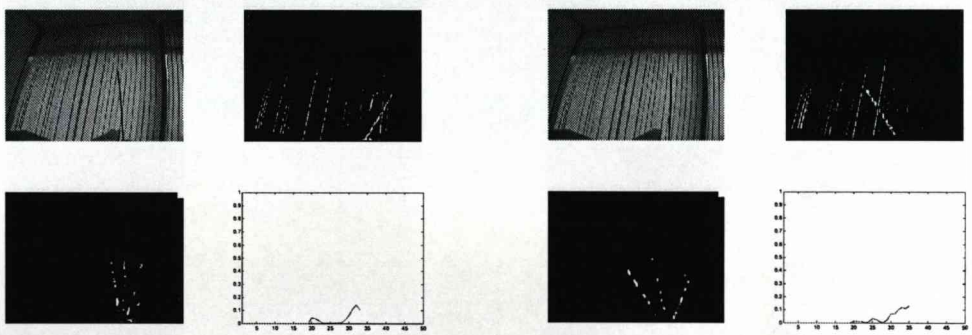
(e) Speed: very fast; Distance: 40cm

Figure 6.14: Experimental results under different speeds, distance is 40cm.



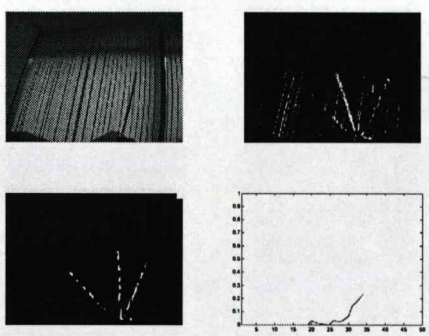
(a) Speed: very slow; Distance: 60cm

(b) Speed: slow; Distance: 60cm



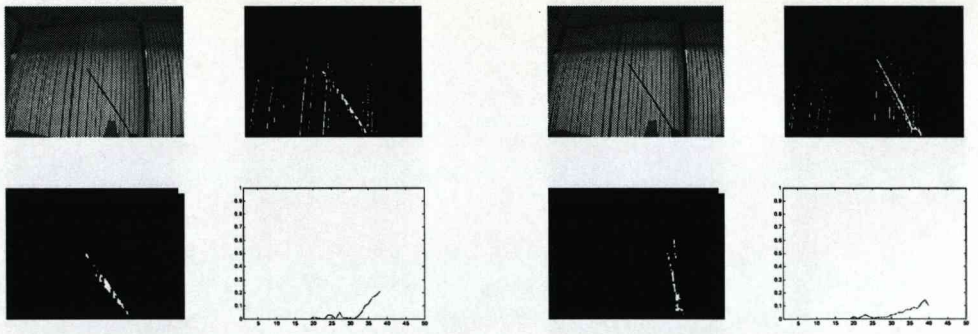
(c) Speed: normal; Distance: 60cm

(d) Speed: fast; Distance: 60cm



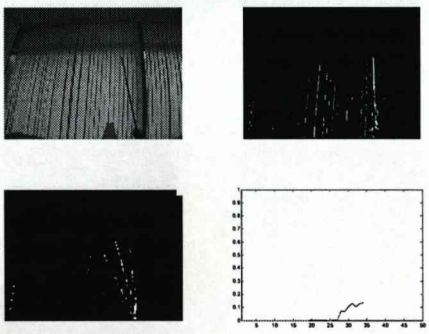
(e) Speed: very fast; Distance: 60cm

Figure 6.15: Experimental results under different speeds, distance is 60cm.

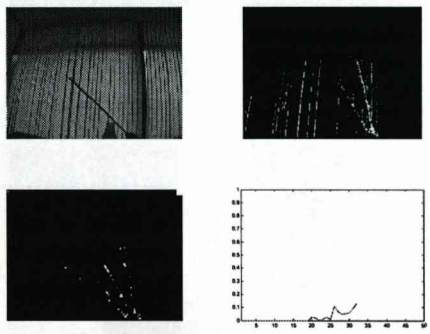


(a) Speed: very slow; Distance: 100cm

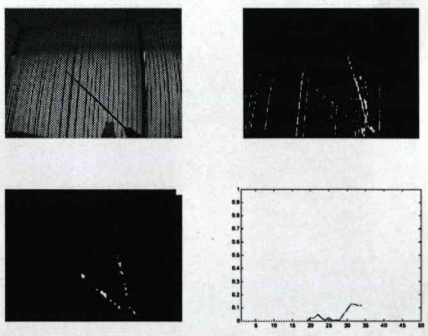
(b) Speed: slow; Distance: 100cm



(c) Speed: normal; Distance: 100cm

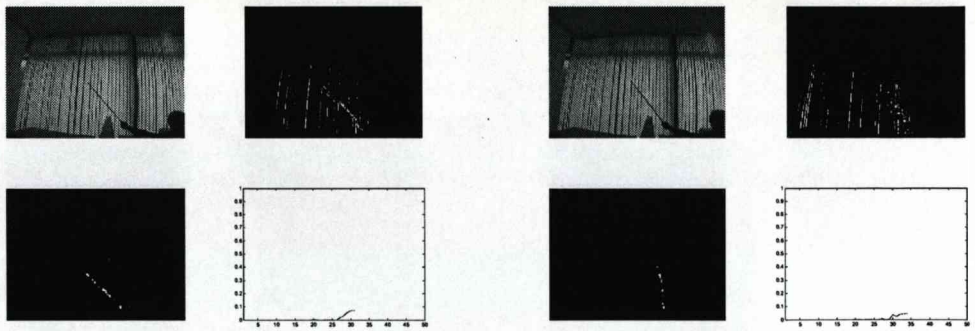


(d) Speed: fast; Distance: 100cm



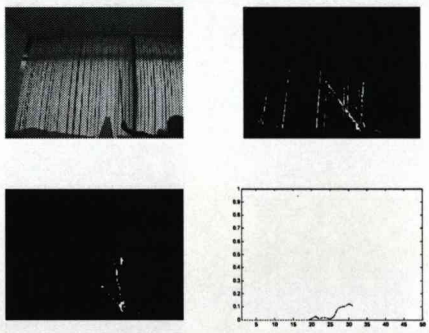
(e) Speed: very fast; Distance: 100cm

Figure 6.16: Experimental results under different speeds, distance is 100cm.

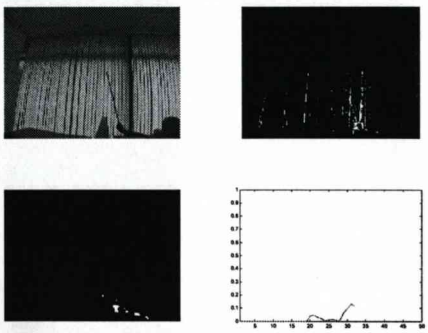


(a) Speed: very slow; Distance: 150cm

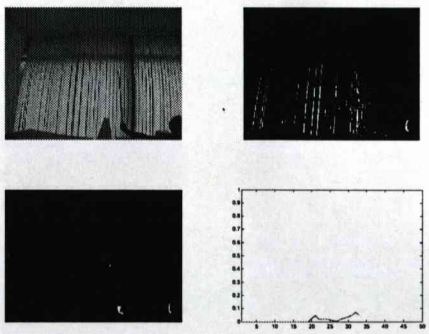
(b) Speed: slow; Distance: 150cm



(c) Speed: normal; Distance: 150cm

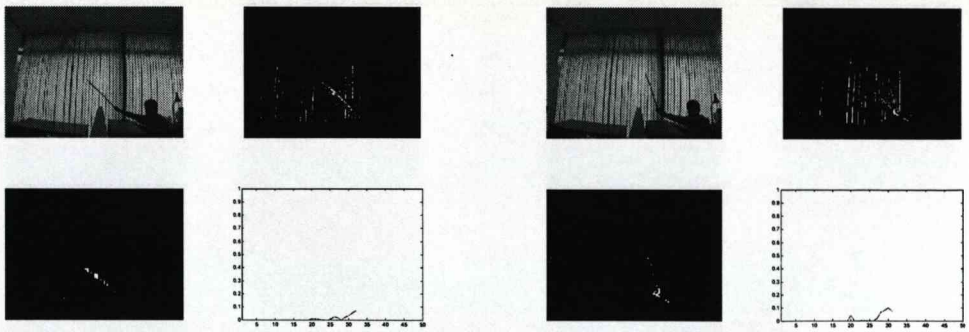


(d) Speed: fast; Distance: 150cm



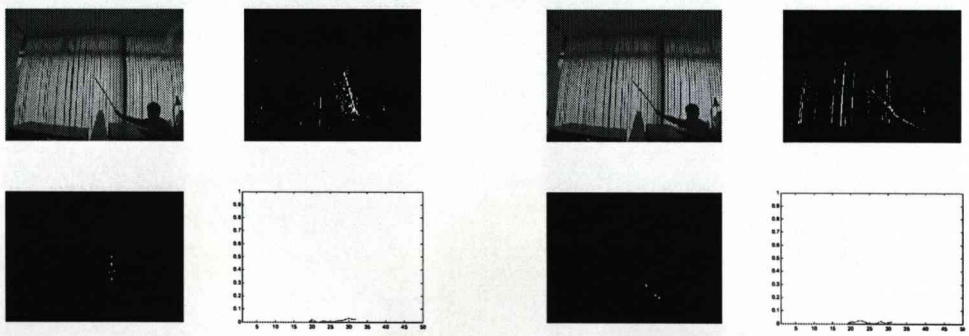
(e) Speed: very fast; Distance: 150cm

Figure 6.17: Experimental results under different speeds, distance is 150cm.



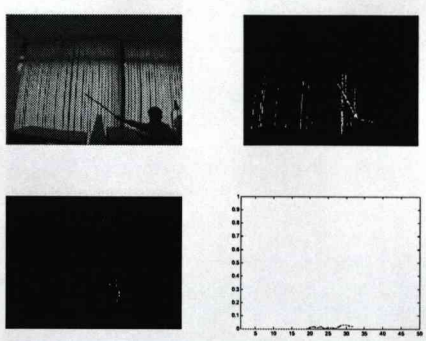
(a) Speed: very slow; Distance: 200cm

(b) Speed: slow; Distance: 200cm



(c) Speed: normal; Distance: 200cm

(d) Speed: fast; Distance: 200cm



(e) Speed: very fast; Distance: 200cm

Figure 6.18: Experimental results under different speeds, distance is 200cm.

6.8 Summary

In this chapter, a new procedure is proposed for motion detection combining temporal difference method, optical flow method, double background filtering method and morphological processing methods. The procedure integrates the advantages of all the methods and presents a fast and robust motion detection procedure. The temporal difference method is introduced first to detect the initial coarse motion object area. Then the optical flow method is applied based on the result of temporal difference method to calculate any possible movement pixel for each video frame. Because of the temporal difference method, the calculation demand for the optical flow is reduced greatly and the moving area is still detected accurately. Then, an improved motion detection procedure is proposed based on a double background filtering technique with morphological processing. The DBF method is used to obtain and keep a stable background image to cope with the appearance of the moving object and is used to eliminate the background interference and separate the foreground moving object from it. The morphological processing methods are used and combined with DBF to gain the better results. Finally, the calculation of the instantaneous rate of change for the moving object appearance is used for the motion detection.

The experimental results show that the procedure can detect the moving objects successfully under the complex background with different speeds of the moving object and different distances between the camera and moving object. But with the increasing of the speed of the moving object and the distance between the camera and moving object, the accuracy of the detection will decrease. With our experimental environment, the procedure works very well under different speeds of moving object when the distance between the camera and the moving object is within 150cm, but it may not work well when the distance between camera and moving object is over 150cm and the speed of moving object is faster than normal. The procedure may not work if the speed of moving object is too fast to be captured by the camera, or the distance between camera and moving object is too far away to make the object too small to be seen.

Chapter 7

Parallel Motion Detection Strategy Using Temporal Difference and Pyramidal Structure-Based Optical Flow

Real-time detection of moving object is very important for video surveillance. In this chapter, we present a novel motion detection procedure using parallel motion detection strategy which is consists of the temporal difference detection and pyramidal structure-based optical flow detection. Firstly, the temporal difference method is used for the whole image to detect coarse motion areas. Secondly, the image is decomposed into coarse levels based on Guassian pyramid construction and the optical flow method is performed independently at each resolution level of the image pyramidal structure for the motion area estimation. Finally, the moving object can be detected by combining the results of parallel motion detection strategy. Analysis is given as to the efficiency and accuracy of our new detection procedure. It shows that the procedure is much faster than conventional mono-resolution detection methods and it needs less computation demands than other methods for the real-time surveillance and it also preserves high accurate rate detection at the same time. Experimental results demonstrate good performance of the proposed procedure

in this chapter.

7.1 Introduction

Detection of moving objects in video streams is known to be a significant and difficult research problem. Several techniques for moving object detection have been studied for years, among them the three representative approaches are temporal difference, background subtraction and optical flow. Our recently research works have been focusing on how to integrate the advantages of these methods [26, 27].

In the case of motion detection and analysis, the vast amount of visual information is a challenge to the processing capabilities of most computers. However, it is commonly believed that in most cases only a small fraction of the available visual information is relevant to the visual activities of a motion detection system performing particular vision tasks. For an artificial visual detection system, it is expected to be able to selectively discard the redundancy visual information and intelligently keep the concentration of the detection system on the area where a target moving object appears in the images.

Previous work on multi-resolution motion detection has been reported. In [137, 138], multi-resolution visual energy change information represented in Laplacian and Gaussian image pyramids was used in motion detection and tracking. Particularly in [138], Burt implemented a coarse-to-fine multi-resolution target search scheme in a pipeline architecture that is based on specially purposed parallel processors, demonstrating an efficient solution to the parallel multi-resolution motion sensing problem. However, because a relatively complex technique of the integration of Gaussian and Laplacian image pyramids was used, Burt's method is not efficient in image representation. In addition, because of the specific parallel computing architecture employed, Burt's vision system is not easily applied in general application cases. Moreover, quantitative analysis of the efficiency improvement brought about by the utilization of the multi-resolution technique was not reported neither in these two works. Some research work on multi-resolution motion detection which

is based on pyramidal structure for motion detection and tracking has been reported to improve the efficiency of calculation [139, 140, 141, 142, 143, 144].

In this chapter, we present a parallel motion detection strategy which is based on the temporal difference detection and pyramidal structure-based optical flow detection. The main goal of this procedure is to separate the background and foreground effectively and detect the object in motion accurately. The parallel motion detection strategy is introduced to detect the motion area. It includes two steps: (1) Temporal difference method, which is used for the whole image to detect coarse motion areas. (2) Pyramidal structure-based optical flow, which is used independently at each resolution level of the image pyramidal structure for the motion area estimation. And, the moving object can be detected by combining the result of parallel motion detection strategy which is proposed not only improves the capability of detecting the object in motion, but also reduces the computation demands. Theoretical investigation together with the experimental results presented in this chapter show the superiority of our procedure over conventional methods.

The procedure is depicted in the flowchart of Fig.7.1.

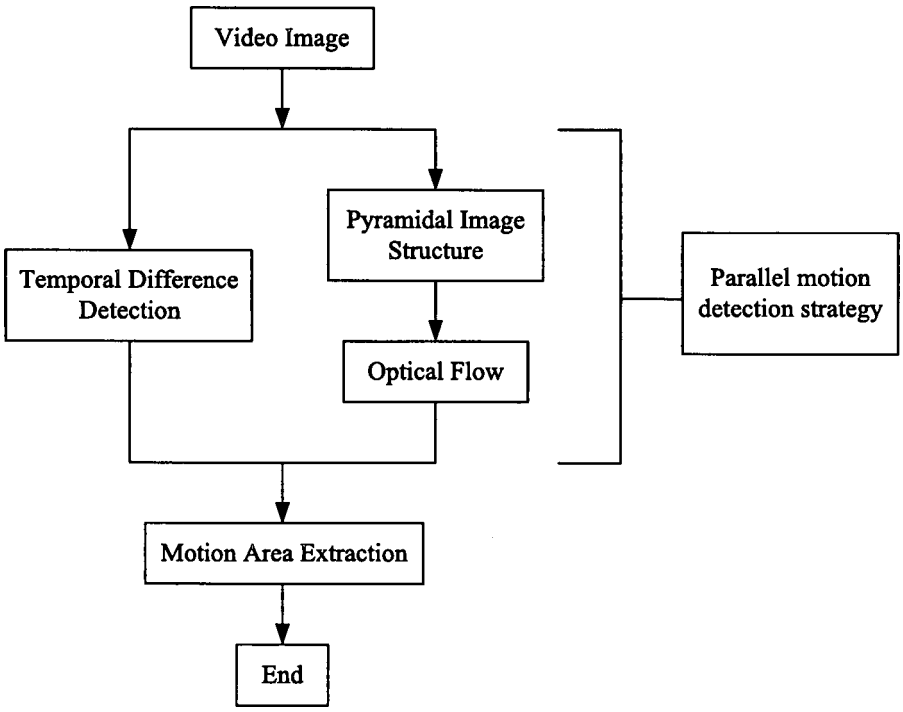


Figure 7.1: Flowchart of the motion detection procedure

As can be seen, the whole procedure is comprised of two steps: (1) Parallel motion detection strategy which includes two methods, temporal difference detection and pyramidal structure-based optical flow detection, is used to detect motion areas; (2) Motion area extraction, the motion area is extracted from the result of step (1).

7.2 Parallel motion detection strategy

7.2.1 Temporal difference detection and optical flow detection

Temporal difference detection and optical flow detection are two popular methods for detecting moving objects in video sequences. The temporal difference method is highly adaptive to dynamic environments, but generally does a poor job of extracting the complete shapes of certain types of moving objects. The method based on optical flow is complex, but it can detect the motion accurately even without knowing the background. The main idea in this paper is to integrate the advantages of these two methods with multi-resolution technique. So the parallel motion detection strategy is proposed. Both temporal difference detection and optical flow detection methods which are the same as the those introduced in Chapter 6 are integrated in the procedure.

7.2.2 Pyramidal structure-based optical flow detection

A multi-resolution (image pyramid) is a data structure within which an input image is represented at successively reduced resolutions. The motivation for computing a structure description is to transform the information in each image into a representation in which searching and matching are more efficient at a fixed computational cost. Many researchers have shown that the efficiency of the searching and matching process can be dramatically improved by performing the search at multiple resolutions. Many basic image operations can also be performed efficiently within pyramid structures [145]. Some pyramid-based "computational tools" have been proposed like Gaussian pyramid, recursive interpolation, Laplacian pyramid, and hierarchical discrete correlation.

Gaussian pyramid construction

The hierarchical image structure we used in this procedure is Gaussian pyramids [146]. The first step in Gaussian pyramid construction is to low-pass filter the original image g_0 to obtain image g_1. The g_1 is a "reduced" version of g_0 in that both resolution and sample density are decreased. In a similar way, g_2 is formed as a reduced version of g_1, and so on. Filtering is performed by a procedure equivalent to convolution with one of a family of local, symmetric weighting functions. An important member of this family resembles the Gaussian probability distribution, so the sequence of images $g_0, g_1, \ldots, g_n$ is called the Gaussian pyramid. A graphical representation of this process in one dimension is given in Fig.7.2.

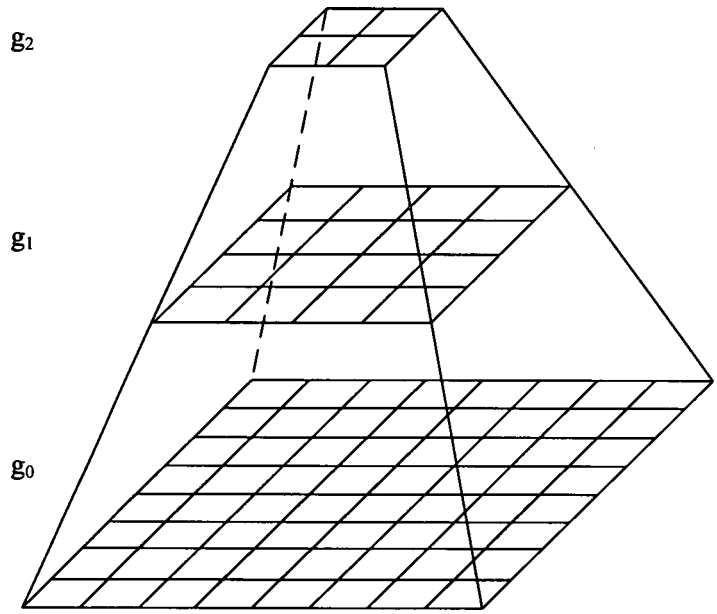


Figure 7.2: The illustration for the Guassian image pyramid.

The level-to-level averaging process is performed by the function REDUCE.

$$g_k = REDUCE(g_{k-1}) \tag{7.2.1}$$

and

$$g_k(i,j) = REDUCE(g_{k-1}) \sum_{p=-N}^{N} \sum_{q=-N}^{N} w(p,q) g_{k-1}(2i+p, 2j+q) \qquad (7.2.2)$$

where $w(p,q)$ is the weighting window function which consists of the Gaussian coefficient.

The Gaussian coefficient is widely used in the Gaussian filter due to its excellent features. One of the principal reasons for using the Gaussian as a smoothing filter is that it has the frequency response of low-pass feature which means that it can remove high spatial frequency components from an image. The Gaussian filter is a 2-D convolution used to eliminate noise in the form of:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2}\left(\frac{x^2+y^2}{\sigma^2}\right)\right\} \qquad (7.2.3)$$

where σ is called a scale parameter and the Gaussian distribution with $\sigma = 1$ is shown in Fig.7.3.

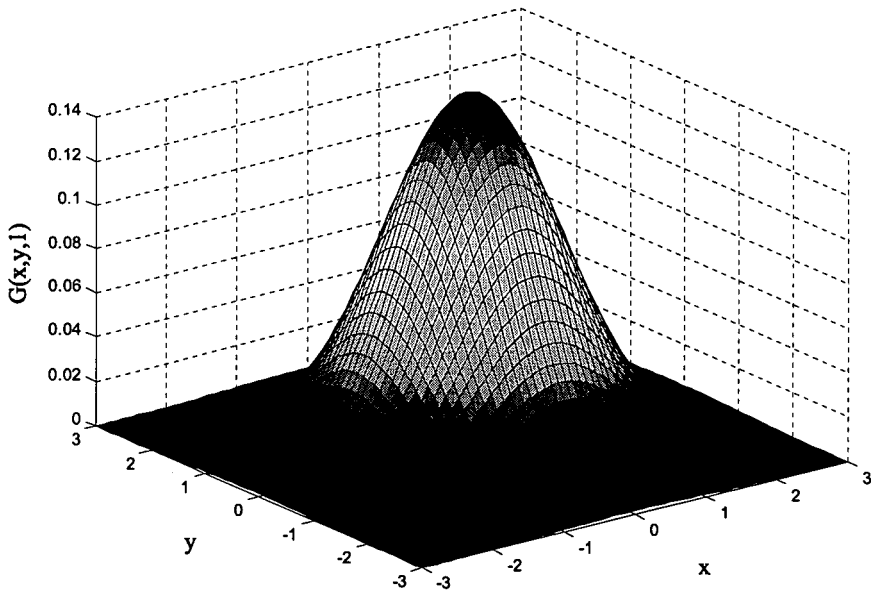


Figure 7.3: Gaussian distribution with $\sigma = 1$.

From Fig.7.3, we can see that the Gaussian filter outputs a weighted average of each pixel's neighborhood, with the average weighted more towards the value of the central pixels. With the increasing of σ, the signal will become more smooth, but the edge may become more blurring. In our experiment, σ is set as 1 and the noise can be eliminated effectively. Compared to the same size mean filter's uniformly weighted average, the Gaussian filter provides gentler smoothing and preserves edges better.

The idea of Gaussian smoothing is to use this 2-D distribution as a point-spread function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. We selected the 5×5 Gaussian coefficient pattern in this method because it provides adequate filtering at low computational cost. Table 7.1 shows a size of 5×5 decimal-valued convolution kernel that approximates a Gaussian filter with $\sigma = 1$.

Table 7.1: Gaussian convolution kernel with $\sigma = 1$

0.002915	0.013064	0.021539	0.013064	0.002915
0.013064	0.05855	0.096532	0.05855	0.013064
0.002915	0.096532	**0.15915**	0.096532	0.002915
0.013064	0.05855	0.096532	0.05855	0.013064
0.002915	0.013064	0.021539	0.013064	0.002915

Image up-sampled among resolution levels

Because the image pyramids are generated in accordance with the rule of dyadic resolution reduction, one pixel in an image layer with a lower resolution in the image pyramid corresponds to 2×2 pixels in its beneath image layer with a higher resolution, as shown in Fig.7.4.

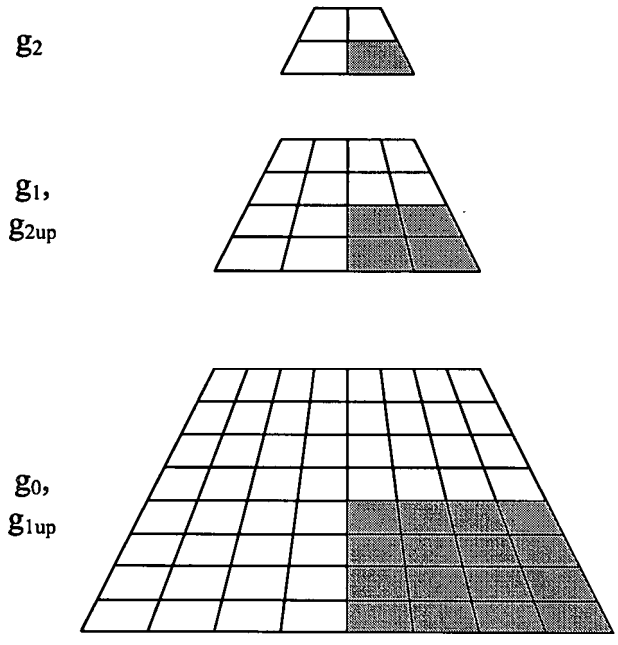


Figure 7.4: The relationship between the pixels in different resolution levels.

Sequentially, the projection of a pixel (i, j) in the k-th resolution level onto the m-th level $(m < k)$ is a square image area consisting of $2^{k-m} \times 2^{k-m}$ pixels. The coordinates of the pixel of the projected image area u, v is related to i, j with the following equation:

$$\begin{aligned} u &= 2^{k-m} i \\ v &= 2^{k-m} j \end{aligned} \tag{7.2.4}$$

Thus, the information up-sampled among different image layers for pixel correspondence, search path selection, and result exchanges should obey the relationship defined in (7.2.1).

Coarse-to-fine optical flow estimation

In order to reduce the computational complexity, a Gaussian pyramid is built to provide a range of coarse to fine views of the image. The motion information is first derived at the lowest resolution level, that is, at the top of the pyramid. Once these motion vectors have been estimated, they are utilized as the prediction motion vectors at the next higher resolution level where these predictions are refined. This process is repeated until the full resolution level is reached. The optical flow detection is used to do the motion estimation in this pyramidal motion estimation technique and the computational load is reduced because of the smaller image size at the lower levels of resolution, and because of the utilization of the motion information that has already been obtained at a given resolution in predicting the motion information at a higher resolution.

Therefore, a hierarchical coarse-to-fine optical flow estimation is introduced in our procedure, as is illustrated in Fig.7.5. It consists of two parts: (1) pyramid construction with image down-sample; (2) optical flow calculation with image up-sample.

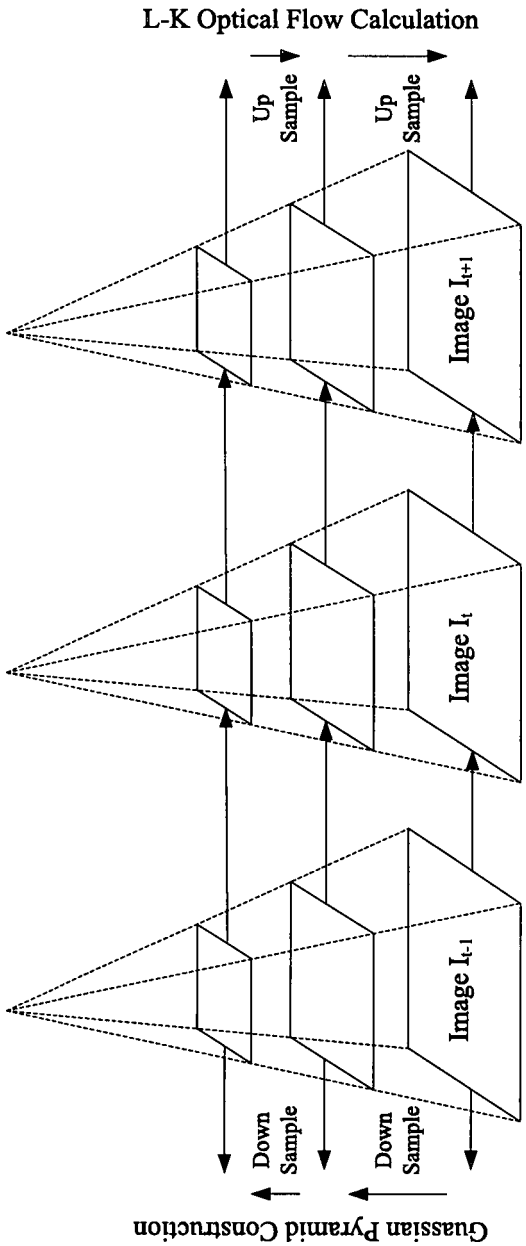


Figure 7.5: Coarse-to-fine optical flow estimation

Morphological processing

Mathematical morphology operator, dilation, which is introduced in Chapter 6 is used to expand the motion areas detected by pyramid-structure based optical flow estimation to make them more visible.

Motion area extraction

With the coarse-to-fine method described above, rough optical flow fields can be obtained. In order to acquire accurate boundaries of moving objects, a method which is proposed in [147] is introduced by using the result of parallel motion detection strategy. The motion area I_{m} can be obtained from the combination results of temporal detection I_{T} and pyramidal structure-based optical flow detection I_{OFd}. It can be denoted as the equation below:

$$I_{\mathrm{m}} = (I_{\mathrm{T}}(x, y, t))\&(I_{\mathrm{OFd}}(x, y, t)) \tag{7.2.5}$$

Summary

The pyramidal structure-based optical flow detection process is outlined as below: (1) Construct the Gaussian pyramids of three successive images; (2) Compute the optical flow at the coarsest resolution level; (3) Map the optical flow pixels obtained in the last step onto the next resolution level; (4) Compute the optical flow based on the result of (3) at the current resolution level; (5) Check if the detection process has reached at the finest resolution level. If so, then stop the detection process and output the detection results; otherwise, go back to Step (3).

The method also possesses two advantages over the mono-resolution target-search methods: (1) An iterative detection process is performed in the multi-resolution domain to increase the accuracy of the location of the target; (2) The search at each resolution level (with the possible exception of the coarsest one) can be restricted within one or more particular areas to reduce the total computational cost of the search process.

7.3 Experimental results

In this section, the effectiveness of the proposed procedure for motion detection is demonstrated for a simulation environment whose background is a vibrated column curtain caused by winds. The procedure runs using MATLAB program. The size of the video image is 320 × 240 pixels and the sample rate is 25 frames per second.

Fig.7.6 illustrates the procedure on a video sequence in which a moving object is moving under a complex background. Fig.7.6(a) shows original video image. Fig.7.6(b) shows the result of video image after using temporal difference detection. Fig.7.6(c) shows the result of the first level of Guassian pyramid construction, g_1. Fig.7.6(d) shows the result of the second level of Guassian pyramid construction, g_2. Fig.7.6(e) shows the result of optical flow detection based on g_2. Fig.7.6(f) shows the result of upsample for g_2, g_{2up}. Fig.7.6(g) shows the result of optical flow detection based on g_{2up}. Fig.7.6(h) shows the result of upsample for g_1, g_{1up}. Fig.7.6(i) shows the result of morphological processing for g_{1up}. Fig.7.6(j) show the result of motion detection for a moving object.

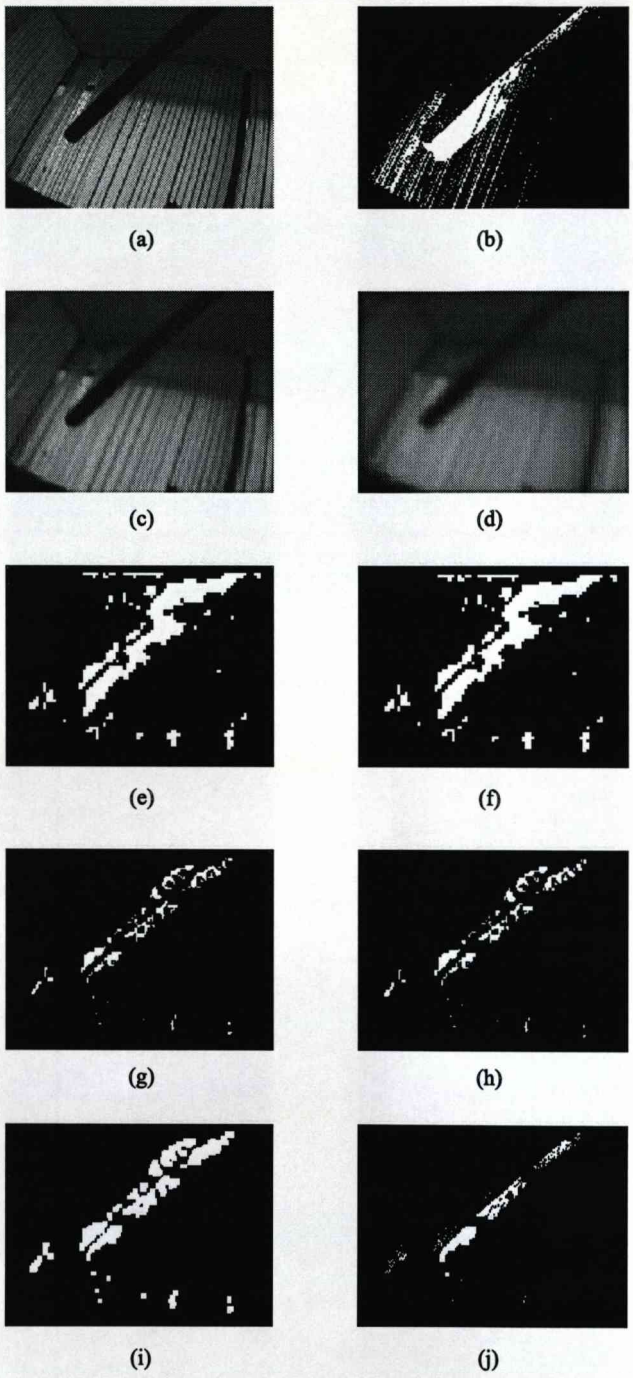


Figure 7.6: Experimental results.

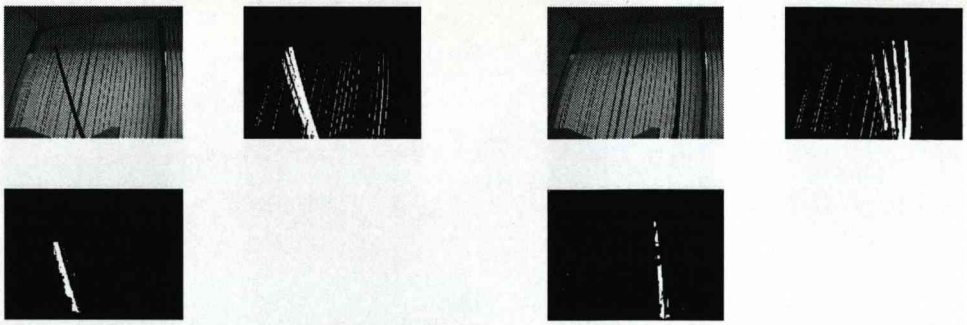
In this chapter, the same experiments are organized as described in Chapter 6 to test the effectiveness of the procedure. Five different speeds of the moving object (very slow, slow, normal, fast and very fast) and six different distances between camera and moving object (20cm, 40cm, 60cm, 100cm, 150cm and 200cm) are chosen. The results are shown in Fig.7.8, Fig.7.9, Fig.7.10, Fig.7.11, Fig.7.12 and Fig.7.13, respectively, and the figure numbering for the whole set of testing results is listed in Fig.7.7.

Speed

Results	Very Slow	Slow	Normal	Very Fast	Fast
20cm	Figure 7.8(a)	Figure 7.8(b)	Figure 7.8(c)	Figure 7.8(d)	Figure 7.8(e)
40cm	Figure 7.9(a)	Figure 7.9(b)	Figure 7.9(c)	Figure 7.9(d)	Figure 7.9(e)
60cm	Figure 7.10(a)	Figure 7.10(b)	Figure 7.10(c)	Figure 7.10(d)	Figure 7.10(e)
100cm	Figure 7.11(a)	Figure 7.11(b)	Figure 7.11(c)	Figure 7.11(d)	Figure 7.11(e)
150cm	Figure 7.12(a)	Figure 7.12(b)	Figure 7.12(c)	Figure 7.12(d)	Figure 7.12(e)
200cm	Figure 7.13(a)	Figure 7.13(b)	Figure 7.13(c)	Figure 7.13(d)	Figure 7.13(e)

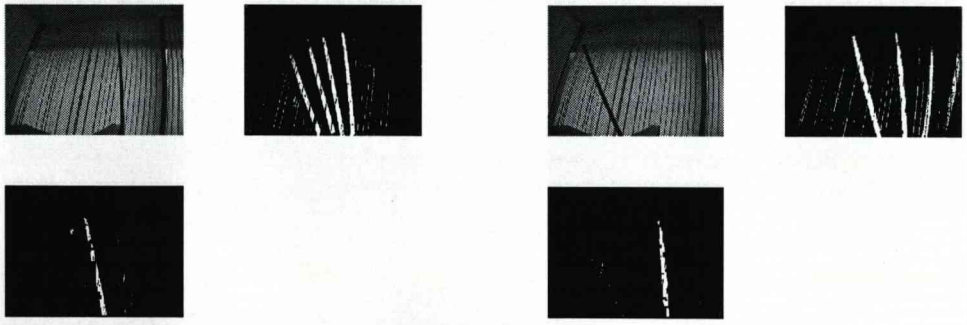
Distance

Figure 7.7: Testing results for the procedure.



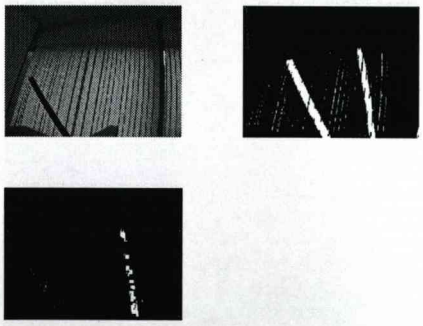
(a) Speed: very slow; Distance: 20cm

(b) Speed: slow; Distance: 20cm



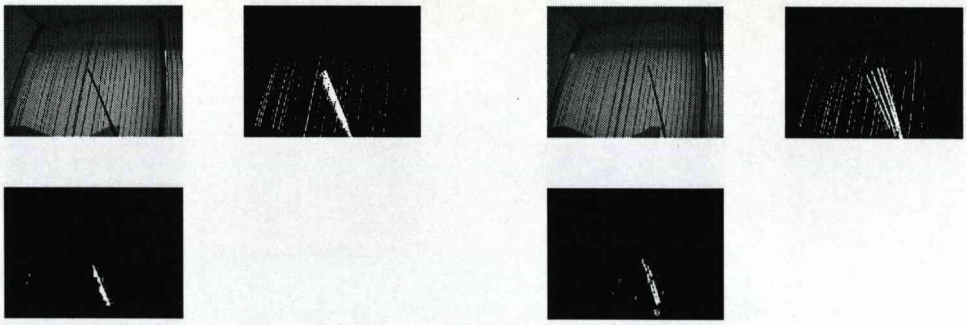
(c) Speed: normal; Distance: 20cm

(d) Speed: fast; Distance: 20cm



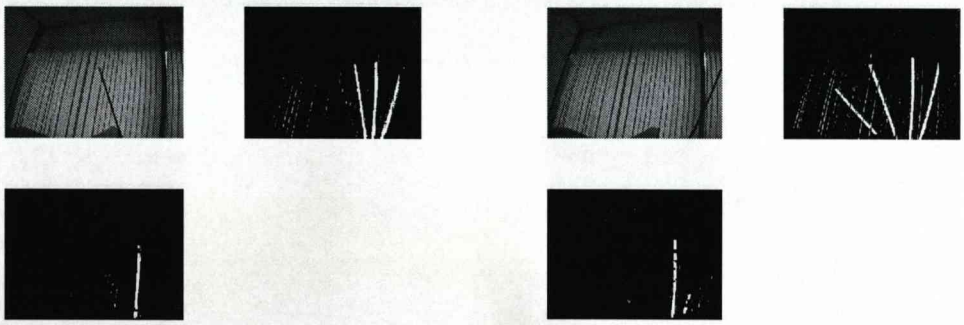
(e) Speed: very fast; Distance: 20cm

Figure 7.8: Experimental results under different speeds, distance is 20cm.



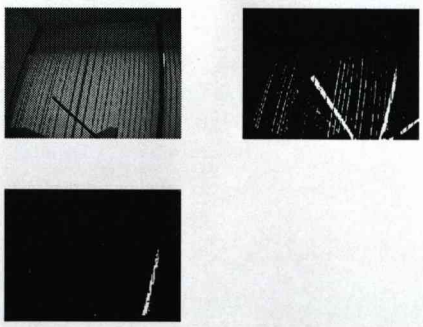
(a) Speed: very slow; Distance: 40cm

(b) Speed: slow; Distance: 40cm



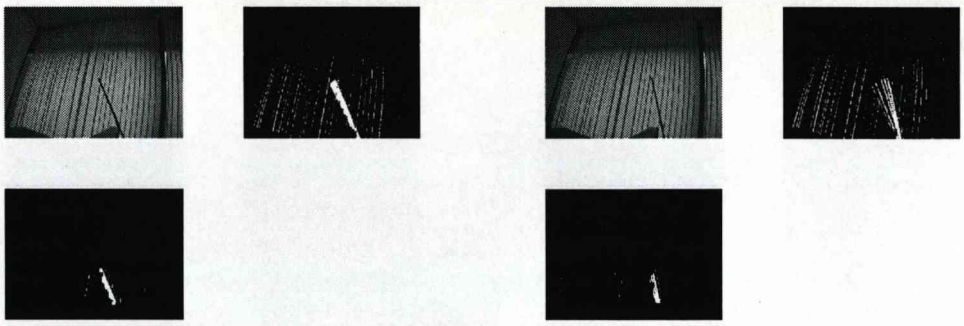
(c) Speed: normal; Distance: 40cm

(d) Speed: fast; Distance: 40cm



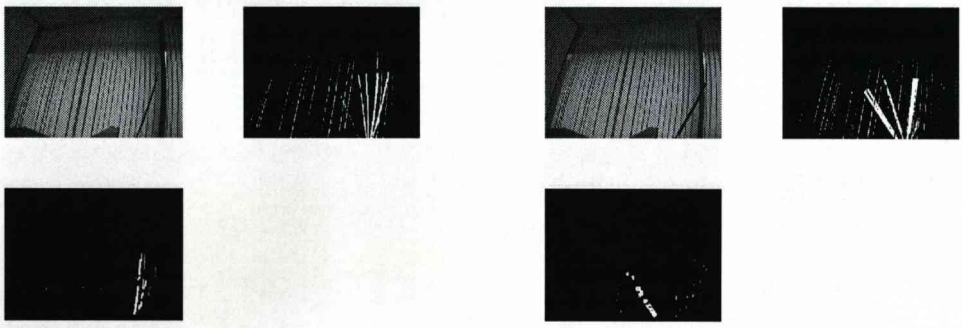
(e) Speed: very fast; Distance: 40cm

Figure 7.9: Experimental results under different speeds, distance is 40cm.



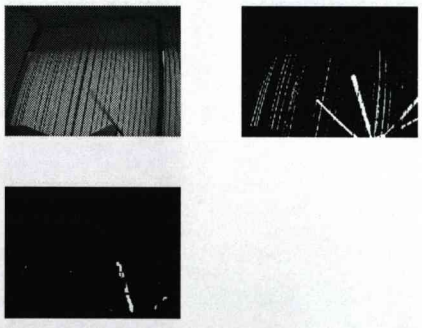
(a) Speed: very slow; Distance: 60cm

(b) Speed: slow; Distance: 60cm



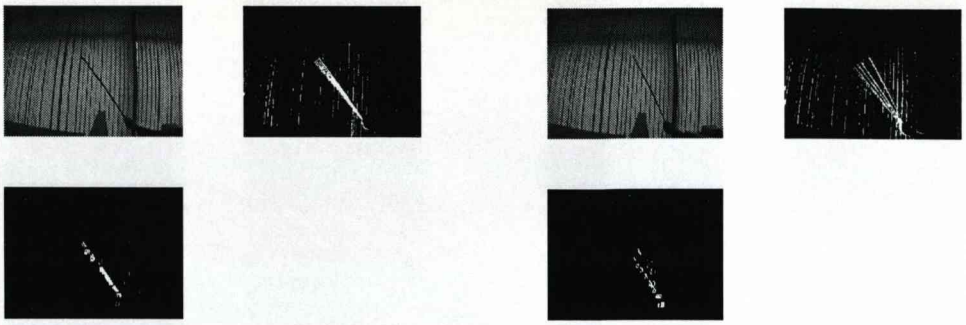
(c) Speed: normal; Distance: 60cm

(d) Speed: fast; Distance: 60cm



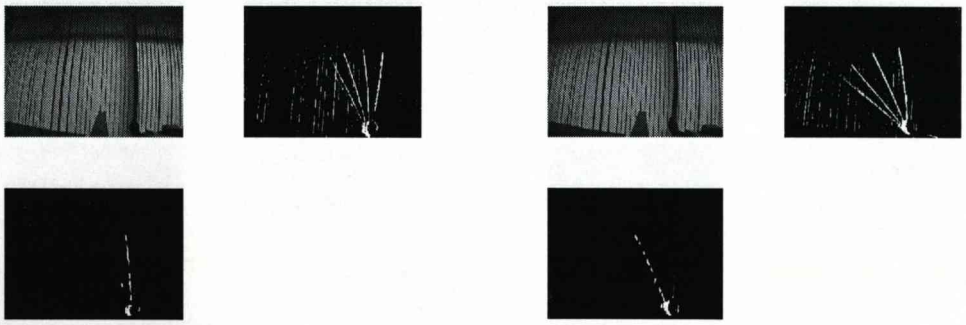
(e) Speed: very fast; Distance: 60cm

Figure 7.10: Experimental results under different speeds, distance is 60cm.



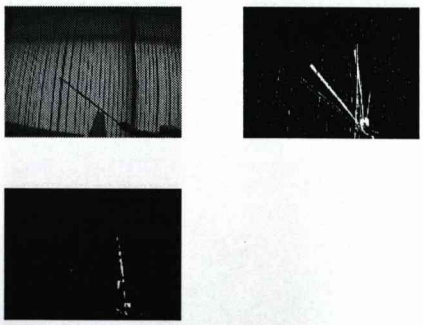
(a) Speed: very slow; Distance: 100cm

(b) Speed: slow; Distance: 100cm



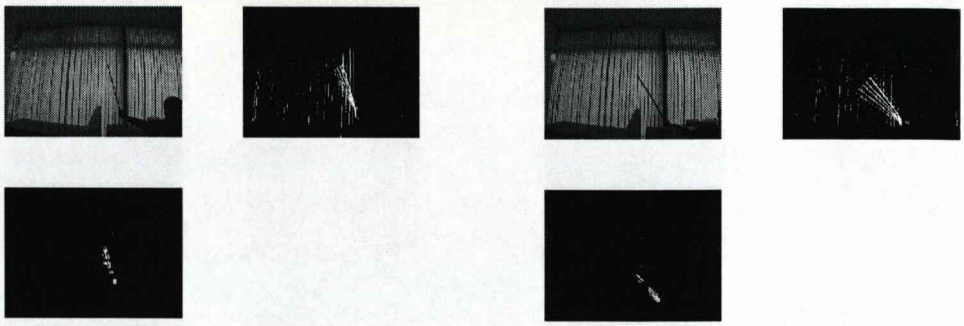
(c) Speed: normal; Distance: 100cm

(d) Speed: fast; Distance: 100cm



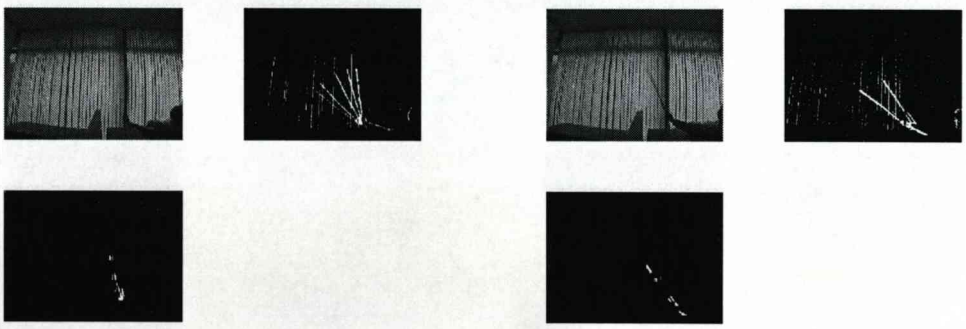
(e) Speed: very fast; Distance: 100cm

Figure 7.11: Experimental results under different speeds, distance is 100cm.



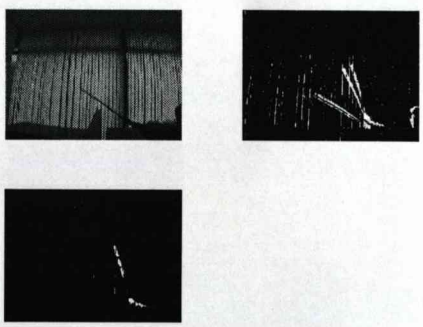
(a) Speed: very slow; Distance: 150cm

(b) Speed: slow; Distance: 150cm



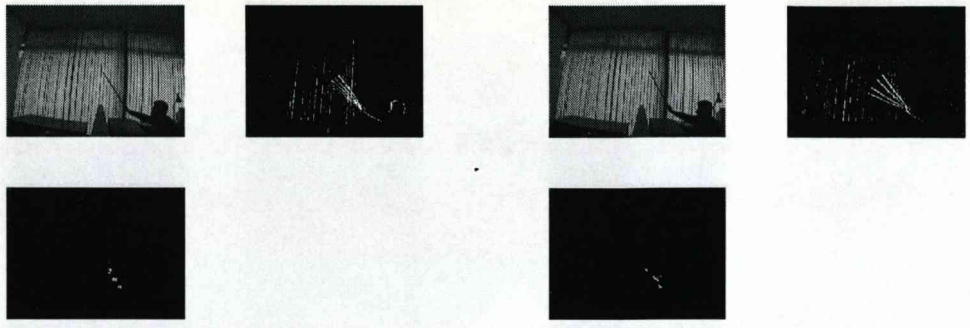
(c) Speed: normal; Distance: 150cm

(d) Speed: fast; Distance: 150cm



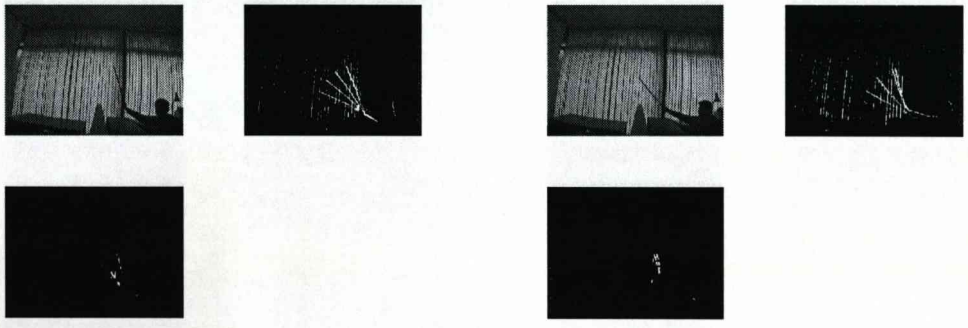
(e) Speed: very fast; Distance: 150cm

Figure 7.12: Experimental results under different speeds, distance is 150cm.



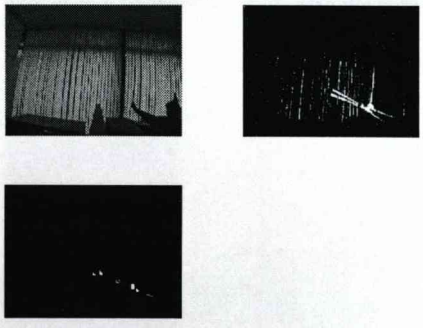
(a) Speed: very slow; Distance: 200cm

(b) Speed: slow; Distance: 200cm



(c) Speed: normal; Distance: 200cm

(d) Speed: fast; Distance: 200cm



(e) Speed: very fast; Distance: 200cm

Figure 7.13: Experimental results under different speeds, distance is 200cm.

7.4 Conclusions

In this chapter, a new procedure based on parallel motion detection strategy which includes temporal difference detection and pyramidal structure-based optical flow detection is proposed. The procedure integrates the advantages of two motion detection methods, temporal difference and optical flow, and multi-resolution technique and presents a fast and robust motion detection procedure. The parallel motion detection strategy is the main part of the procedure, in which, on the one hand, the temporal difference method is used to initially detect any possible motion areas in the video image, on the other hand, the optical flow method combined with Gaussian pyramidal structure is used to do the motion areas estimation. Finally, the motion area is extracted from combination results of temporal difference detection and pyramidal structure-based optical flow detection.

The experimental results showed that the procedure can distinguish the moving object successfully from the complex background with different speeds of the moving object and different distances between the camera and moving object. But with the increase of the speed of the moving object and the distance between the camera and moving object, the accuracy of the detection will decrease. Under our experimental environment, the procedure works very well under different speeds of moving object when the distance between camera and moving object is within 150cm, but it may not work well when the distance between camera and moving object is over 150cm and the speed of moving object is faster than normal. Of course, the procedure may not work if the speed of moving object is too fast to be captured by the camera, or the distance between camera and moving object is too far away to make the object too small to be seen.

Chapter 8

Conclusions and Future Work

This chapter concludes the thesis. It summarizes the major achievements of the presented research work in the field of new procedure development for digital signal processing. In this chapter, possible directions for further investigations are also indicated.

8.1 Conclusion

At the beginning of the thesis, the background, motivations, objectives and significance of this research work are all presented. According to the different practical situations and requirements, this research work has been focused on the development of new digital signal processing procedures with the applications to speech, electromyography and image processing.

Chaos theory has been introduced including the theoretical background and mathematical definitions. The chaos theory is a scientific theory describing erratic behavior in certain nonlinear dynamical systems. In mathematics and physics, chaos theory describes the behavior of certain nonlinear dynamical systems that under certain conditions exhibit a phenomenon known as chaos. On the other word, chaos theory is kind of method which is used to find out the invariable characteristics from the random signal for control and predict. Based on chaos theory, a new noise reduction procedure has been developed. Compared to the traditional methods, my research work provides a new way

to process the signal for noise reduction. The procedure consists of three main steps: phase space reconstruction, phase space projection and transforming back to the time domain. The main idea of the procedure is to reconstruct the attractor of the contaminated signal in the phase space for identification of noise contributions. The simulation results have shown that the experimental noisy signal can be cleaned by the new procedure and the SNR value increased obviously after the procedure using. It is also demonstrated that this chaos theory generalised technique can be used as an alternative and effective means of the digital signal processing.

The main problem with the application to filtering the noise from the corrupted speech signals is that some useful information is missing from the distorted signal. Sometimes, recovering the missing information is more difficult than cleaning the contamination from a signal. Based on chaos theory, a procedure of phase space trajectory reconstruction for speech signal restoration has been developed. The procedure is based on the theory of nonlinear dynamic system analysis, in which the signal is transformed (projected) into a high dimension phase space and the space interpolation method is used to reconstruct the new trajectory back into the time domain for the signal restoration. The simulation results have shown that, by applying our method to do the signal restoration, the result of noise reduction and the signal quality have been improved satisfactorily.

Uterine electromyography (EMG) signal is a complex nonlinear signal which can be used in clinical for early diagnosis of preterm labour. But the uterine EMG signal is difficult to differentiate uterine contractions which lead to preterm birth. A procedure of EMG signal processing has been developed for signal classification by using chaos theory generalised techniques and other signal processing methods. There are three main steps in the procedure, signal pre-processing, signal feature extraction and signal classification. In this procedure, the main idea is that the difference between term labour signal and preterm labour signal can be the contraction patterns. So the analysis work is focus on the contraction patterns of the signal, try to extract some features to do the classification work. The classification accuracy of artificial

neural network (ANN) is 64.1% which is an encouraging result. Due to lack of the organized data, the work needs to continue before any conclusions can be drawn.

In the image processing, two procedures have been developed for moving object detection under the noisy environment. The first one is based on accumulative optical flow and double background filtering. The second one is based on parallel motion detection strategy using temporal difference and pyramidal structure-based optical flow. Both procedures integrate the advantages of different motion detection methods and use different strategies to achieve better performance. The two procedures, both of them have high capability of anti-interference and preserves high accurate rate detection at the same time. They also needs less computation demands than other traditional methods for the real-time video surveillance. The results show that both of procedures can separate the background interference and foreground moving object very well, and detect motion area successfully from the complex background under different speeds of the moving object and different distances between the camera and moving object.

8.2 Suggestions for future work

With an understanding presented in this thesis, we know that the procedure development is very important to the digital signal processing and the following aspects would be worthy of further investigation:

- With some constraints, in new noise reduction procedure, the sinusoidal signal is used as the experimental signal to test the effectiveness of the idea for noise reduction which is based on chaos theory. More other types of signal, especially the random signals, are to be included into the framework. A more general framework without any constraint is also desirable in the future although more effort is certainly needed.

- Time delay and embedding dimension are two parameters that influence the phase space reconstruction. In our procedures, the time delay and

embedding dimension which are used for signals are 1 and 3, respectively. The combination of different time delay and different embedding dimension can construct different phase spaces. The optimal choosing for both of them can easily and correctly explore the dynamical features of signal in phase space and to find out the better solution.

- In the procedure of speech signal processing, besides the speech signal itself, the characteristic of human voice should be considered as a factor and combined with our method to do the processing.

- In the procedure of EMG signal processing, different feature extraction methods and classification methods are to be investigated. Of course, the more accurate and better quality data are needed.

- In the procedures of image processing, the procedure complexity should be taken into account because of the real-time application. Simplifying the computation of the procedure and keeping the high detection accuracy should be investigated. The procedure also needs to be developed to suit for variety of different noisy environment.

Appendix A

TSTOOL Software Package

Introduction

TSTOOL is a software package for nonlinear time series analysis. It is implemented mainly in Matlab, with some time-critical parts written in C/C++ (as mex-functions).

Objectives

1. Implement existing algorithms for nonlinear time-series analysis
2. Develop new methods for specific data analysis problems
3. Create an expandable platform for signal processing

Function

TSTOOL can be used for computing:

- Time-delay reconstruction
- Lyapunov exponent(s)
- Fractal dimensions

- Mutual information
- Surrogate data tests
- Nearest neighbor statistics
- Return times
- Poincare sections
- Nonlinear prediction

How to install TSTOOL

TSTOOL is based on Matlab, so it will run on any platform for which Matlab is available. Some parts of TSTOOL are coded in C++ and require therefore an ANSI C++ compliant compiler (e.g. gcc 2.95) to be compiled.

The Web Linking

TSTOOL Home Page:
http://www.physik3.gwdg.de/tstool/index.html

Appendix B

Video Processing Development Program (VCAPG2 for MATLAB)

Introduction

VCAPG2 is DirectShow based video capture DLL that support various types of video capture cards. VCAPG2 is a new version of MATLAB resident video capture program, which employs Microsoft's DirectShow libraries rather than the older Video for Windows (VfW) library. In order to run VCAPG2, you will need to install the DirectX runtime libraries. The vcapg2 requires a video capture device, such as a USB camera to be installed and working properly. This code has been developed and tested with DirectX 8.1 and 9.0 with MATLAB6.1/6.5.

Features

Features are:

1. Up to 6 video capture cards can grab images at one command.
2. Supported video capture devices are including IEEE1394 based DV CAM,

USB Camera as well as old VFW based Camera.

3. VCAPG2 can run up to almost twice as fast as VfW based program.

How to install VCAPG2

Just copy vcapg2.dll to your MATLAB current directory. You can check MATLAB current directory by usiing 'pwd' command. After that you just type 'vcapg2' will appear preview window.

The Web Linking

VCAPG2 Home Page:
http://www.ikko.k.hosei.ac.jp/ matlab/matkatuyo/vcapg2.htm

Appendix C

EMG Data for Signal Calssification

All EMG data in our experiment, term labor signals and pre-term labor signals, are plotted in Fig.C.1, Fig.C.2, Fig.C.3 and Fig.C.4 respectively.

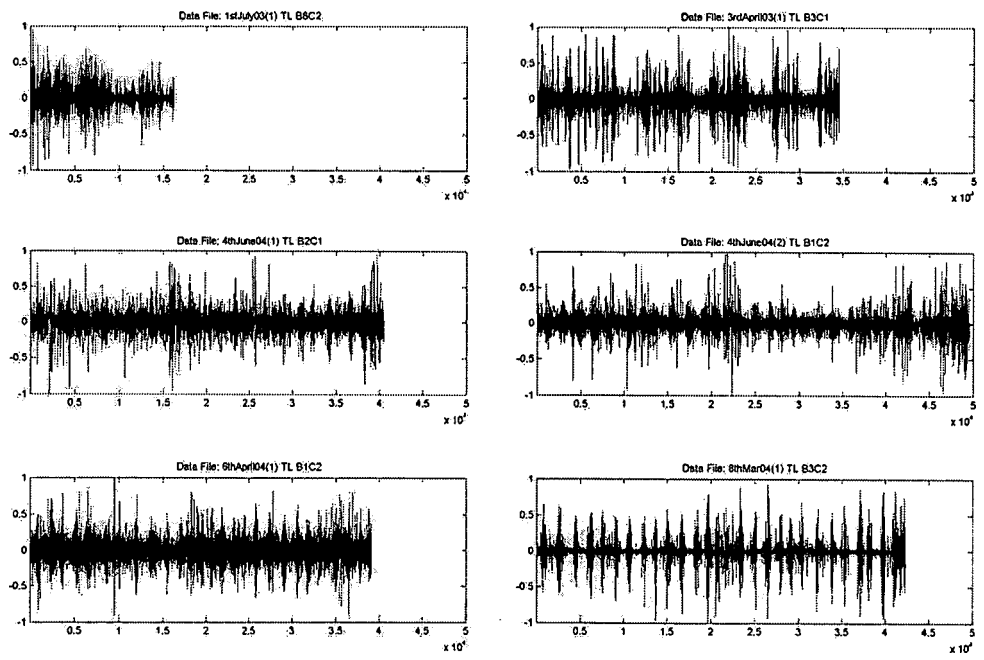


Figure C.1: Term labor Data.

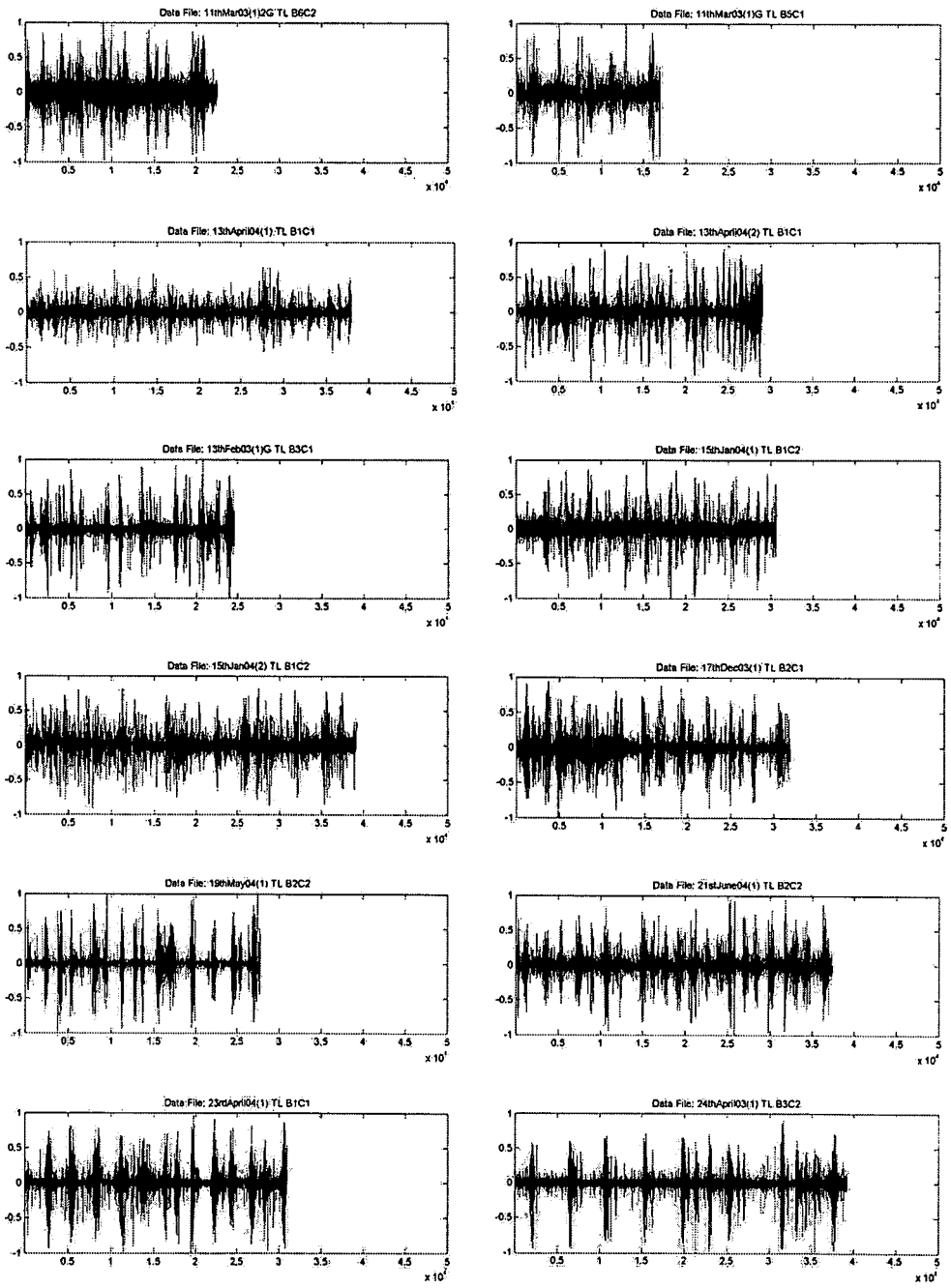


Figure C.2: Term labor Data.

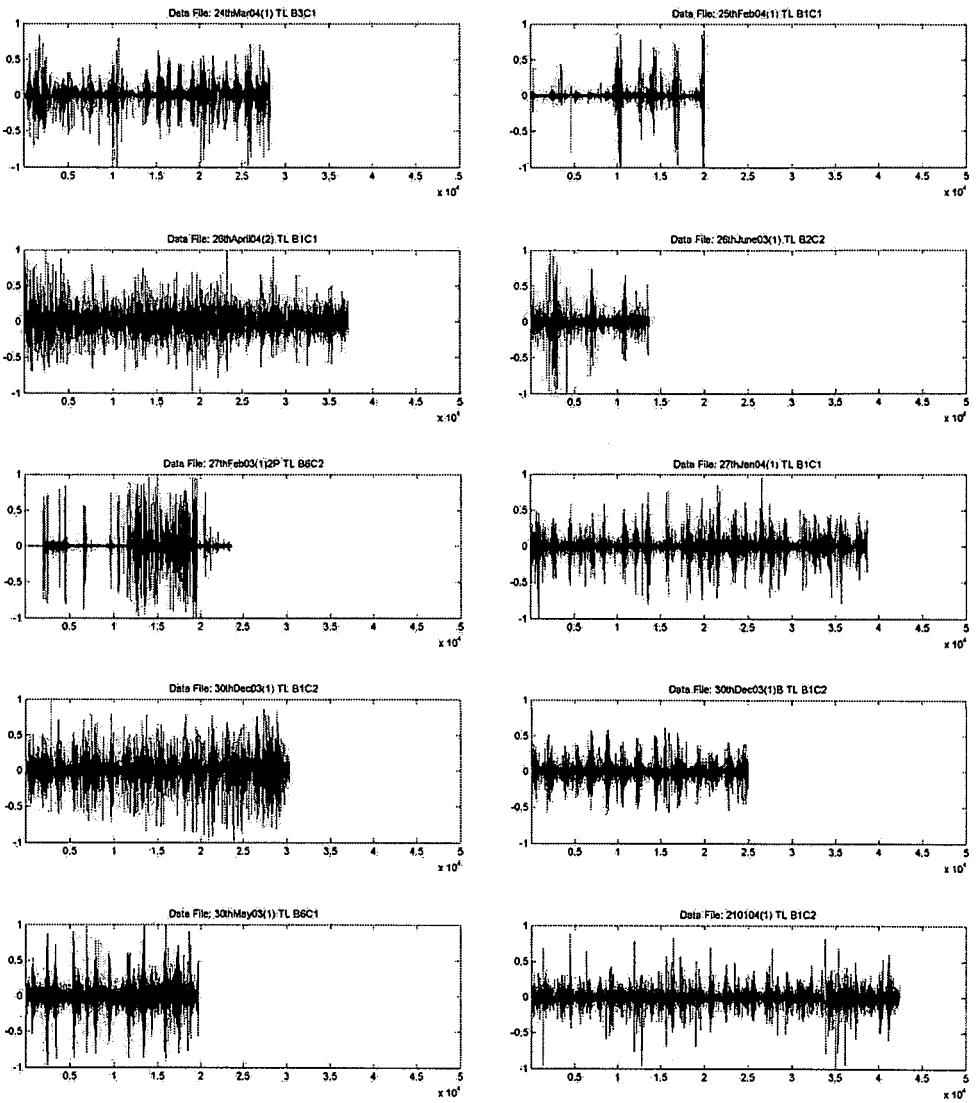


Figure C.3: Term labor Data.

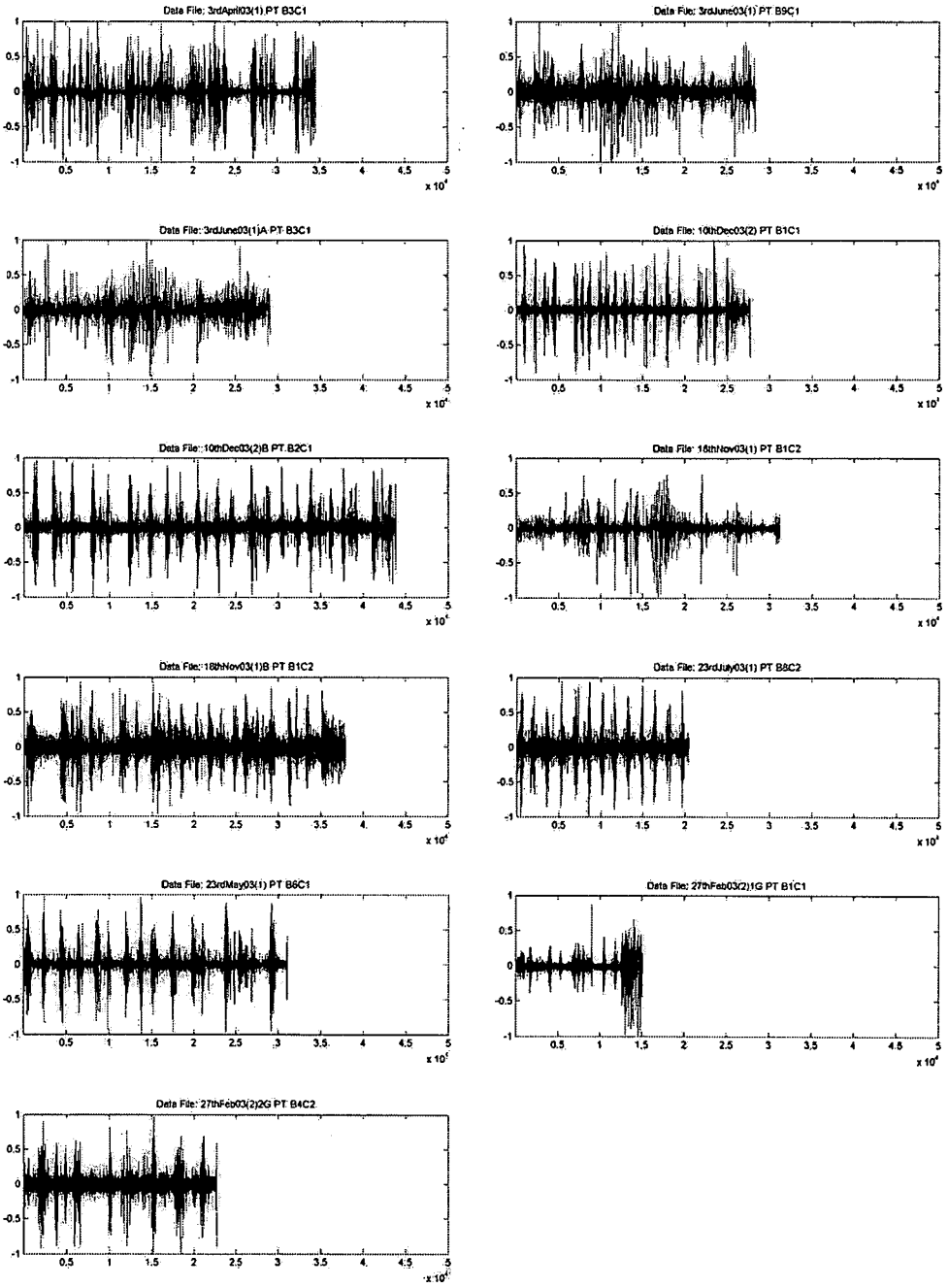


Figure C.4: Pre-term labor Data.

Appendix D

Publication Award

The paper below has been selected as "The Best Paper Award of The 2007 International Conference of Signal and Image Engineering".

N. Lu, J. Wang, Q.H. Wu and L. Yang, Motion Detection Based On Accumulative Optical Flow and Double Background Filtering, *Proceedings of World Congress on Engineering 2007*, London, UK, Vol.1, pp.602-607, 2-4 July, 2007.

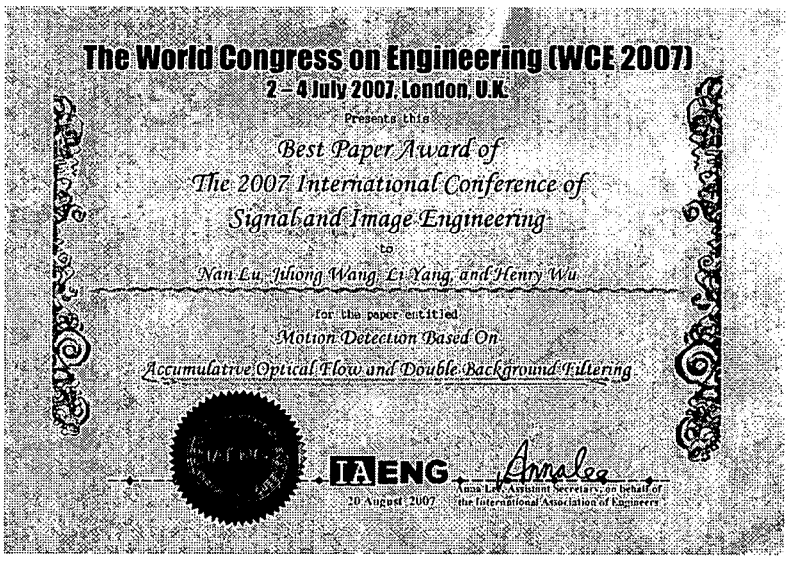


Figure D.1: The Best Paper Award.

Appendix E

Matlab Codes

E.1 Singular Value Decomposition Algorithm

```
function [u,s,v] = svdsim(a,tol)

% SVDSIM  simple SVD program
%
% A simple program that demonstrates how to use the QR
% decomposition to perform the SVD of a matrix. A may be
% rectangular and complex.
%
% usage: [U,S,V]= SVDSIM(A)
%     or       S = SVDSIM(A)
%
% with A = U*S*V' , S>=0 , U'*U = Iu   , and V'*V = Iv
%
% The idea is to use the QR decomposition on A to gradually
% "pull" U out from the left and then use QR on A transposed to
% "pull" V out from the right. This process makes A lower
% triangular and then upper triangular alternately. Eventually,
% A becomes both upper and lower triangular at the same time,
% (i.e. Diagonal) with the singular values on the diagonal.
```

```
% Matlab's own SVD routine should always be the first choice to
% use, but this routine provides a simple "algorithmic
% alternative" depending on the users' needs.

% Paul Godfrey
% October 23, 2006

if ~exist('tol','var')
   tol=eps*1024;
end

% reserve space in advance
sizea=size(a); loopmax=100*max(sizea); loopcount=0;

% or use Bidiag(A) to initialize U, S, and V
u=eye(sizea(1)); s=a'; v=eye(sizea(2));

Err=realmax; while Err>tol & loopcount<loopmax ;
%   log10([Err tol loopcount loopmax]); pause
    [q,s]=qr(s'); u=u*q;
    [q,s]=qr(s'); v=v*q;

% exit when we get "close"
    e=triu(s,1);
    E=norm(e(:));
    F=norm(diag(s));
    if F==0, F=1;end
    Err=E/F;
    loopcount=loopcount+1;
end
% [Err/tol loopcount/loopmax]
```

```
% fix the signs in S
ss=diag(s); s=zeros(sizea); for n=1:length(ss)
    ssn=ss(n);
    s(n,n)=abs(ssn);
    if ssn<0
       u(:,n)=-u(:,n);
    end
end

if nargout<=1
   u=diag(s);
end

return
```

E.2 MMSE log-STSA Algorithm

```
function output=MMSESTSA85(signal,fs,IS)

% output=MMSESTSA85(signal,fs,IS)
% Short time Spectral Amplitude Minimum Mean Square Error Method
% for Denoising noisy speech. based on log-Spectral MMSE Ephraim
% et al (1985) paper under the same title. signal is the input
% noisy speech, fs is its sampling frequency and IS (which is
% optional) is the initial silence estimate (The default value
% is 0.25 which means that it is assumed that the first 0.25
% seconds of the signal is speech inactive period and may be
% used for initial noise parameter estimation). The output is
% the restored estimate of clean speech.
% Author: Esfandiar Zavarehei
% Created: Jan-04
```

```
% Last Modified: 24-01-05

if (nargin<3 | isstruct(IS))
    IS=.25; %Initial Silence or Noise Only part in seconds
end
W=fix(.025*fs); %Window length is 25 ms
SP=.4; %Shift percentage is 40% (10ms) %Overlap-Add method
       %works good with this value(.4)
wnd=hamming(W);

% IGNORE FROM HERE ..............................
if (nargin>=3 & isstruct(IS))
        %This option is for compatibility with another programme
    W=IS.windowsize
    SP=IS.shiftsize/W;
    %nfft=IS.nfft;
    wnd=IS.window;
    if isfield(IS,'IS')
        IS=IS.IS;
    else
        IS=.25;
    end
end
% ....................................UP TO HERE

pre_emph=0; signal=filter([1 -pre_emph],1,signal);

NIS=fix((IS*fs-W)/(SP*W) +1);
                            %number of initial silence segments

y=segment(signal,W,SP,wnd);
          %This function chops the signal into frames Y=fft(y);
```

```
YPhase=angle(Y(1:fix(end/2)+1,:)); %Noisy Speech Phase
Y=abs(Y(1:fix(end/2)+1,:)); %Specrogram
numberOfFrames=size(Y,2); FreqResol=size(Y,1);

N=mean(Y(:,1:NIS)')'; %initial Noise Power Spectrum mean
LambdaD=mean((Y(:,1:NIS)').^2)';
                          %initial Noise Power Spectrum variance
alpha=.99;
% used in smoothing xi (For Deciesion Directed method for
% estimation of A Priori SNR)

NoiseCounter=0;
NoiseLength=9;
              %This is a smoothing factor for the noise updating

G=ones(size(N)); %Initial Gain used in calculation of the new xi
Gamma=G;

X=zeros(size(Y)); %Initialize X (memory allocation)

h=waitbar(0,'Wait...');

for i=1:numberOfFrames
    %%%%%%%%%%%%%%%VAD and Noise Estimation START
    if i<=NIS % If initial silence ignore VAD
        SpeechFlag=0;
        NoiseCounter=100;
    else % Else Do VAD
        [NoiseFlag, SpeechFlag, NoiseCounter, Dist]=vad(Y(:,i),N,
        NoiseCounter); %Magnitude Spectrum Distance VAD
    end
```

```
    if SpeechFlag==0 %If not Speech Update Noise Parameters
        N=(NoiseLength*N+Y(:,i))/(NoiseLength+1);
                                    %Update and smooth noise mean
        LambdaD=(NoiseLength*LambdaD+(Y(:,i).^2))
                ./(1+NoiseLength);
                                %Update and smooth noise variance
    end
    %%%%%%%%%%%%%%%%%%%VAD and Noise Estimation END

    gammaNew=(Y(:,i).^2)./LambdaD; %A postiriori SNR
    xi=alpha*(G.^2).*Gamma+(1-alpha).*max(gammaNew-1,0);
                    %Decision Directed Method for A Priori SNR

    Gamma=gammaNew;
    nu=Gamma.*xi./(1+xi);
                        %A Function used in Calculation of Gain

    G= (xi./(1+xi)).*exp(.5*expint(nu));
                                %Log spectral MMSE [Ephraim 1985]
    X(:,i)=G.*Y(:,i); %Obtain the new Cleaned value

    waitbar(i/numberOfFrames,h,
            num2str(fix(100*i/numberOfFrames)));
end

close(h);
output=OverlapAdd2(X,YPhase,W,SP*W);
                                %Overlap-add Synthesis of speech
output=filter(1,[1 -pre_emph],output);
                                %Undo the effect of Pre-emphasis

function
```

```
ReconstructedSignal=OverlapAdd2(XNEW,yphase,windowLen,ShiftLen);

% Y=OverlapAdd(X,A,W,S);
% Y is the signal reconstructed signal from its spectrogram. X
% is a matrix with each column being the fft of a segment of
% signal. A is the phase angle of the spectrum which should
% have the same dimension as X. if it is not given the phase
% angle of X is used which in the case of real values is zero
% (assuming that its the magnitude). W is the window length of
% time domain segments if not given the length is assumed to be
% twice as long as fft window length. S is the shift length of
% the segmentation process for example in the case of non
% overlapping signals it is equal to W and in the case of %50
% overlap is equal to W/2. if not givven W/2 is used. Y is the
% reconstructed time domain signal.
% Sep-04
% Esfandiar Zavarehei

if nargin<2
    yphase=angle(XNEW);
end if nargin<3
    windowLen=size(XNEW,1)*2;
end if nargin<4
    ShiftLen=windowLen/2;
end if fix(ShiftLen)~=ShiftLen
    ShiftLen=fix(ShiftLen);
    disp('The shift length have to be an integer as it is the
         number of samples.')
    disp(['shift length is fixed to ' num2str(ShiftLen)])
end

[FreqRes FrameNum]=size(XNEW);
```

```
Spec=XNEW.*exp(j*yphase);

if mod(windowLen,2) %if FreqResol is odd
    Spec=[Spec;flipud(conj(Spec(2:end,:)))];
else
    Spec=[Spec;flipud(conj(Spec(2:end-1,:)))];
end sig=zeros((FrameNum-1)*ShiftLen+windowLen,1); weight=sig;

for i=1:FrameNum
    start=(i-1)*ShiftLen+1;
    spec=Spec(:,i);
    sig(start:start+windowLen-1)=sig(start:start+windowLen-1)
                                 +real(ifft(spec,windowLen));
end ReconstructedSignal=sig;

function Seg=segment(signal,W,SP,Window)

% SEGMENT chops a signal to overlapping windowed segments
% A= SEGMENT(X,W,SP,WIN) returns a matrix which its columns are
% segmented and windowed frames of the input one dimentional
% signal, X. W is the number of samples per window, default
% value W=256. SP is the shift percentage, default value SP=0.4.
% WIN is the window that is multiplied by each segment and its
% length should be W. the default window is hamming window.
% 06-Sep-04
% Esfandiar Zavarehei

if nargin<3
    SP=.4;
end if nargin<2
    W=256;
```

```
end if nargin<4
    Window=hamming(W);
end
Window=Window(:); %make it a column vector

L=length(signal); SP=fix(W.*SP);
N=fix((L-W)/SP +1); %number of segments

Index=(repmat(1:W,N,1)+repmat((0:(N-1))'*SP,1,W))';
hw=repmat(Window,1,N); Seg=signal(Index).*hw;

function [NoiseFlag, SpeechFlag, NoiseCounter,
Dist]=vad(signal,noise,NoiseCounter,NoiseMargin,Hangover)

% [NOISEFLAG, SPEECHFLAG, NOISECOUNTER, DIST]=vad(SIGNAL,NOISE,
                                NOISECOUNTER,NOISEMARGIN,HANGOVER)
% Spectral Distance Voice Activity Detector SIGNAL is the the
% current frames magnitude spectrum which is to labeld as noise
% or speech, NOISE is noise magnitude spectrum template
% (estimation), NOISECOUNTER is the number of imediate previous
% noise frames, NOISEMARGIN (default 3)is the spectral distance
% threshold. HANGOVER (default 8)is the number of noise segments
% after which the SPEECHFLAG is reset (goes to zero). NOISEFLAG
% is set to one if the the segment is labeld as noise
% NOISECOUNTER returns the number of previous noise segments,
% this value is reset (to zero) whenever a speech segment is
% detected. DIST is the spectral distance.
% Saeed Vaseghi
% edited by Esfandiar Zavarehei
% Sep-04

if nargin<4
```

```
    NoiseMargin=3;
end if nargin<5
    Hangover=8;
end if nargin<3
    NoiseCounter=0;
end

FreqResol=length(signal);

SpectralDist= 20*(log10(signal)-log10(noise));
SpectralDist(find(SpectralDist<0))=0;

Dist=mean(SpectralDist); if (Dist < NoiseMargin)
    NoiseFlag=1;
    NoiseCounter=NoiseCounter+1;
else
    NoiseFlag=0;
    NoiseCounter=0;
end

% Detect noise only periods and attenuate the signal
if (NoiseCounter > Hangover)
    SpeechFlag=0;
else
    SpeechFlag=1;
end
```

E.3 Lucas-Kanade Optical Flow Algorithm

```
function [u, v] = LucasKanade(im1, im2, windowSize);

% LucasKanade  lucas kanade algorithm, without pyramids
```

```
                                                    (only 1 level);

% REVISION: NaN vals are replaced by zeros

[fx, fy, ft] = ComputeDerivatives(im1, im2);

u = zeros(size(im1)); v = zeros(size(im2));

halfWindow = floor(windowSize/2); for i =
halfWindow+1:size(fx,1)-halfWindow
   for j = halfWindow+1:size(fx,2)-halfWindow
      curFx = fx(i-halfWindow:i+halfWindow,
                                        j-halfWindow:j+halfWindow);
      curFy = fy(i-halfWindow:i+halfWindow,
                                        j-halfWindow:j+halfWindow);
      curFt = ft(i-halfWindow:i+halfWindow,
                                        j-halfWindow:j+halfWindow);

      curFx = curFx';
      curFy = curFy';
      curFt = curFt';

      curFx = curFx(:);
      curFy = curFy(:);
      curFt = -curFt(:);

      A = [curFx curFy];

      U = pinv(A'*A)*A'*curFt;

      u(i,j)=U(1);
      v(i,j)=U(2);
```

```
   end;
end;

u(isnan(u))=0; v(isnan(v))=0;

% u=u(2:size(u,1), 2:size(u,2));
% v=v(2:size(v,1), 2:size(v,2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [fx, fy, ft] = ComputeDerivatives(im1, im2);
% ComputeDerivatives Compute horizontal, vertical and time
% derivative between two gray-level images.

if (size(im1,1) ~= size(im2,1)) | (size(im1,2) ~= size(im2,2))
   error('input images are not the same size');
end;

if (size(im1,3)~=1) | (size(im2,3)~=1)
   error('method only works for gray-level images');
end;

fx = conv2(im1,0.25*[-1 1; -1 1])+conv2(im2, 0.25*[-1 1; -1 1]);

fy = conv2(im1,0.25*[-1 -1; 1 1])+conv2(im2, 0.25*[-1 -1; 1 1]);

ft = conv2(im1,0.25*ones(2))+conv2(im2, -0.25*ones(2));

% make same size as input
fx=fx(1:size(fx,1)-1, 1:size(fx,2)-1); fy=fy(1:size(fy,1)-1,
1:size(fy,2)-1); ft=ft(1:size(ft,1)-1, 1:size(ft,2)-1);
```

Bibliography

[1] D. Stranneby and W. Walker. *Digital Signal Processing and Applications.* Dag Stranneby and William Walker, ISBN: 0-7506-6344-8, 2004. Second edition.

[2] C.E. Shannon. *Claude E. Shannon: Collected Papers.* John Wiley & Sons, ISBN: 0-780-30434-9, 1993.

[3] B. A. Shenoi. *Introduction To Digital Signal Processing and Filter Design.* John Wiley & Sons, ISBN: 0-471-65442-6, 2006.

[4] W. Hurewicz. *Collected Works of Witold Hurewicz.* American Mathematical Society, ISBN: 0-821-80011-6, 1995.

[5] J.F. Kaiser, *Digital filters.* Wiley, New York, 1966.

[6] J.W. Cooley and J.W. Turkey. An Algorithm for Machine Calculation of Complex Fourier Series, *Mathematics of Computation* 19(90):297-301, 1965.

[7] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME - Journal of Basic Engineering* 82:35-45, 1960.

[8] S. Haykin and B. Widrow. *LeastMeanSquare Adaptive Filters.* John Wiley & Sons, ISBN: 0-471-21570-8, 2002.

[9] S. White. *Digital Signal Processing: A Filtering Approach.* Thomson Delmar Learning, ISBN: 0-766-81531-5, 2000.

[10] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Dover, ISBN: 0-486-43938-0, 2005.

[11] S. Haykin and L. Bamber. *Adaptive Filter Theory.* Prentice Hall, ISBN: 0-130-90126-1, 2001. Fourth edition.

[12] G.R. Arce. *Nonlinear Signal Processing: A Statistical Approach.* John Wiley & Sons, ISBN: 0-471-67624-1, 2005.

[13] I.Pitas and A.N. Venetsanopoulos. *Nonlinear digital filters: Principles and Applications.* Springer, ISBN: 0-792-39049-0, 1990.

[14] J. Serra. Introduction to mathematical morphology. *Computer vision, graphics, and image processing*, 35(3):283-305, Sep., 1986.

[15] J. Goutsias, L. Vincent and D.S. Bloomberg. *Mathematical morphology and its applications to image and signal processing*, Springer, ISBN: 0-792-37862-8, 2000.

[16] I. Grattan-Guinness. Joseph Fourier and the revolution in mathematical physics. *Journal of the Institute of Mathematics and its Applications*, 5:230-253, 1969.

[17] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematical Computation*, 19:297-301, 1965.

[18] S.G. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, ISBN: 0-124-66605-1, 1998.

[19] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674-693, 1989.

[20] J.W. Gibbs. *The Collected Works of J. Willard Gibbs.* Yale University Press, 1948.

[21] H.D.I. Abarbanel. *Analysis of Observed Chaotic Data.* Springer, ISBN: 0-387-94523-7, 1996.

[22] N. Lu, J.H. Wang and Q.H. Wu. Speech signal restoration using phase space reconstruction, *XVIII International Conference on Systems Engineering*, pages 155-160, Sep., 2006. Coventry, UK. (Published)

[23] N. Lu, J.H. Wang and Q.H. Wu. Signal restoration using phase space reconstruction method, *IEEE Transaction On Signal Processing.* (Submitted)

[24] N. Lu and J.H. Wang. Uterine electromyography features extraction for classification of term and pre-term signals, *Proceedings of The 2nd International Conference on Complex Systems and Applications*, Jun. 8-10, 2007. Jinan, China. (Published)

[25] N. Lu, J.H. Wang and Q.H. Wu. Uterine electromyography signal feature extraction and classification, *International Journal of Modelling, Identification and Control.* (Prepare to submit)

[26] N. Lu, J.H. Wang, Q.H. Wu and L. Yang. Motion detection based on accumulative optical flow and double background filtering, *Proceedings of World Congress on Engineering 2007*, 1:602-607, Jul. 2-4, 2007. London, UK. (Published, The Best Paper Reward)

[27] N. Lu, J.H. Wang, Q.H. Wu and L. Yang. An improved motion detection method for real-time surveillance, *IAENG International Journal of Computer Science*, 35(1):119-128, Mar., 2008. (Published)

[28] N. Lu, J.H. Wang and Q.H. Wu, Motion detection using temporal difference and pyramidal structure-based optical flow. *Proceedings of American Control Conference 2008*, Seattle, Washington, USA, 11-13 June, 2008. (Submitted)

[29] N. Lu, J.H. Wang and Q.H. Wu, Bidirectional Motion Detection Strategy Using Temporal Difference and Pyramidal Structure-Based Optical Flow, *International Journal of Automation and Control.* (Submitted)

[30] N. Lu, J. Wang and Q.H. Wu, Motion detection using optical flow and inter-frame block-based histogram correlation, *International Conference on Modelling, Identification and Control 2008*, Shanghai, China, 29 June - 2 July, 2008. (Submitted)

[31] N. Lu, J. Wang, Q.H. Wu and J.L. Wei, Bidirectional motion detection strategy for real-time surveillance. *UKACC Control Conference 2008*, Manchester, UK, 2-4 September, 2008. (Submitted)

[32] K.T. Alligood, T.D. Sauer and James A. Yorke. *Chaos: An Introduction to Dynamical Systems.* Springer-Verlag New York Inc., ISBN: 0-387-94677-2, 2000.

[33] V.G. Maz'ia, T.O. Shaposhnikova and V. Maz'ya. *Jacques Hadamard: A Universal Mathematician.* American Mathematical Society, ISBN: 0-821-81923-2, 2000.

[34] J.B. Green. *Poincare and the Three Body Problem.* American Mathematical Society, ISBN: 0-821-80367-0, 1996.

[35] J. Coates and S. Helgason. *Ergodic Theory and Dynamical Systems.* Birkhauser Verlag, ISBN: 3-764-33096-1, 1982.

[36] G.D. Birkhoff and G.C. Rota. *Ordinary Differential Equations.* John Wiley & Sons Inc, ISBN: 0-471-07411-0, 1978.

[37] U. Frisch. *Turbulence: The Legacy of A. N. Kolmogorov.* Cambridge University Press, ISBN: 0-521-45713-0, 1995.

[38] M.L. Cartwright. *Integral Functions.* Cambridge University Press, ISBN: 0-521-04586-0, 1956.

[39] G.H. Hardy and J.E. Littlewood. *Inequalities.* Cambridge University Press, ISBN: 0-521-35880-9, 1988.

[40] M.W. Hirsch, S. Smale and R. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos.* Academic Press, ISBN: 0-123-49703-5, 2003.

[41] M. Ausloos and M. Dirickx. *The Logistic Map and the Route to Chaos: From the Beginnings to Modern Applications.* Springer-Verlag Berlin and Heidelberg GmbH & Co. K, ISBN: 3-540-28366-8, 2005.

[42] E. Lorenz. *The Essence of Chaos.* Routledge, ISBN: 0-295-97514-8, 1998.

[43] R. Abraham and Y. Ueda. *The Chaos Avant-garde: Memories of the Early Days of Chaos Theory.* World Scientific Publishing, ISBN: 9-810-24404-5, 2001.

[44] Y. Ueda. *The Road to Chaos.* Aerial Press, ISBN: 0-942-34414-6, 1992.

[45] E. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130-141, 1963.

[46] O.E. Rössler. An equation for continuous chaos. *physics letter A*, 57(5):397-398, 1976.

[47] E. Ott. *Chaos in Dynamical Systems.* Cambridge University Press, ISBN: 0-521-01084-5, 2002.

[48] F. Takens. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence*, 898:366C381, 1981.

[49] J.C. Sprott. *Chaos and Time-Series Analysis.* Oxford University Press, ISBN: 0-19-850839-5, Oxford, U.K., 2003.

[50] G.P. King, R. Jones and D.S. Broomhead. Phase portraits from a time series: A singular system approach. *Nuclear Physics B Proceedings Supplements*, 2:379-390, 1987.

[51] A.M. Fraser and H. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134-1140, Feb., 1986.

[52] M.B. Kennel, R. Brown and H.D.I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical con-

struction. *Physical Review A - Atomic, Molecular, and Optical Physics*, 45(6):3403-3411, March 1992.

[53] L.Y. Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*, 110(1):43-50, December 1997.

[54] T. Buzug and G. Pfister. Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors. *Physical Review A*, 45(10):7073-7084, May 1992.

[55] B. Mandelbrot. *The Fractal Geometry of Nature*. W.H.Freeman & Co Ltd., ISBN: 0-716-71186-9, 1982.

[56] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189-208, 1983.

[57] P. Grassberger and I. Procaccia. Characterization of strange attractors. *Physical Review Letters*, 5(5):346-349, 1983.

[58] A. Wolf, J.B. Swift, H.L. Swinney and J.A. Vastano. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16:285-317, 1985.

[59] M.T. Rosenstein, J.J. Collins and C.J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65:117-134, 1993.

[60] A.G. Darbyshire. Calculating Lyapunov exponents from a time series. *IEE Colloquium on Exploiting Chaos in Signal Processing*, 2:1-6, June 1994.

[61] M. Banbrook, G. Ushaw and S. McLaughlin. How to extract Lyapunov exponents from short and noisy time series. *IEEE Transactions on Signal Processing*, 45(5):1378-1382, May 1997.

[62] M.G. Signorini and S. Cerutti. Lyapunov exponents calculated from heart rate variability time series. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 16(1):119-120, 1994.

[63] M.T. Johnson and R.J. Povinelli. Generalized phase space projection for nonlinear noise reduction. *Physica D: Nonlinear Phenomena*, 201(3):306-317, Feb., 2005.

[64] E.J. Kostelich and T. Schreiber. Noise reduction in chaotic time-series data: A survey of common methods. *Physical Review E*, 48(3):1752-1763, Sep., 1993.

[65] S.M. Hammel. A noise reduction method for chaotic systems. *Physics Letters A*, 148(8):421-428, 1990.

[66] T. Sauer. A noise reduction method for signals from nonlinear systems. *Physica D: Nonlinear Phenomena*, 58(1):193-201, 1992.

[67] J.I. Salisbury. Noise reduction using chaotic theory. *Ocean Technologies and Opportunities in the Pacific for the 90's' - Proceedings*, 1565-1569, Oct., 1991.

[68] K. Urbanowicz, J.A. Holyst, T. Stemler and H. Benner. Noise reduction in chaotic time series by a local projection with nonlinear constraints. *Acta Physica Polonica, Series B*, 35(9):2175-2197, Sep., 2004.

[69] R. Cawley and G.H. Hsu. Local-geometric-projection method for noise reduction in chaotic maps and flows. *Physical Review A*, 46(6):3057-3082, Sep., 1992.

[70] Z.K. Yang, L. Zheng and H. Jing. A new noise reduction method for chaotic signals: improved local-projection algorithm. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 3:1389-1392, Oct., 2002.

[71] E. Ott. *Chaos in Dynamical Systems*, Cambridge University Press, ISBN: 0-521-01084-5, Aug., 2002.

[72] A.I. Mees, P.E. Rapp and L.S. Jennings. Singular-value decomposition and embedding dimension. *Physical Review A - Atomic, Molecular, and Optical Physics*, 36(1):340-346, Jul., 1987.

[73] P.K. Sadasivan, D.N. Dutt. SVD based technique for noise reduction in electroencephalographic signals. *Signal Processing*, 55(2):179-189, Dec., 1996.

[74] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(6):1109-1121, Jul., 1984.

[75] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 133(2):443-445, Jul., 1985.

[76] R.J. McAulay and M.L. Malpass. Speech enhancement using a soft-decision noise suppression filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(2):137-145, Jul., 1980.

[77] C.H. You, S.N. Koh and S. Rahardja. Masking-based β-order MMSE speech enhancement. *Speech Communication*, 48(1):57-70, Jan., 2006.

[78] X.M. Gao, S.J. Ovaska and Z.O. Hartimo. Speech signal restoration using an optimal neural network structure. *IEEE International Conference on Neural Network*, 4:1841-1846, Jun., 1996.

[79] S.J. Godsill and P.J.W. Rayner. A Bayesian approach to the restoration of degraded audio signals. *IEEE Transactions on Speech and Audio Processing*, 3(4):267-278, Jul., 1995.

[80] J. Porter and S. Boll. Optimal estimators for spectral restoration of noisy speech. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 9:53-56, Mar., 1984.

[81] M.T. Johnson, A.C. Lindgren, R.J. Povinelli and X.L. Yuan. Performance of nonlinear speech enhancement using phase space reconstruction. *IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings*, 1:920-923, Apr., 2003.

[82] R. Hegger, H. Kantz and L. Matassini. Noise reduction for human speech signals by local projections in embedding spaces. *IEEE Transactions on Circuits and System I: Fundamental Theory and Applications*, 48(12):1454-1461, Dec., 2001.

[83] I. Kokkinos and P. Maragos. Nonlinear speech analysis using models for chaotic systems. *IEEE Transactions on Speech and Audio Processing*, 13(6):1098-1109, Nov., 2005.

[84] R. Hegger, H. Kantz and L. Matassini. Denoising Human Speech Signals Using Chaoslike Features. *Physical Review Letters*, 84(14):3197-3200, Apr., 2000.

[85] V. Pitsikalis and P. Maragos. Speech analysis and feature extraction using chaotic models. *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1:533-536, 2003.

[86] B.D. Bojanov, H.A. Hakopian and B. Sahakian. *Spline functions and multivariate interpolations*, Kluwer Academic Publishers, ISBN: 0-792-32229-0, Mar., 1993.

[87] D. Staudenmann, J.R. Potvin, I. Kingma and D.F. Stegeman. Effects of EMG processing on biomechanical models of muscle joint systems: Sensitivity of trunk muscle moments, spinal forces, and stability. *Journal of Biomechanics*, 40(4):900-909, 2007.

[88] W. Wang, A.D. Stefano and R. Allen. A simulation model of the surface EMG signal for analysis of muscle activity during the gait cycle. *Computers in Biology and Medicine*, 36(6):601-618, Jun., 2006.

[89] H. Choi, J.H. Jeong, S.H. Hwang, H.C. Choi and W.H. Cho. Feature evaluation and pattern recognition of lower limb muscle EMG during postural balance control. *Key Engineering Materials*, 326:867-870, 2006.

[90] X. Guo, P. Yang, L.L. Chen, X.T. Wang and L.F. Li. Study of the control mechanism of robot-prosthesis based-on the EMG processed. *Proceedings of the World Congress on Intelligent Control and Automation*, 2:9490-9493, 2006.

[91] Y.L. Meng, B.Z. Liu and Y.P. Liu. A comprehensive nonlinear analysis of electromyogram. *Engineering in Medicine and Biology Society, Proceedings of the 23rd Annual International Conference of the IEEE*, 2(2):1078-1081, 2001.

[92] G.J. Small, N.B. Jones, J.C. Fothergill and A.P. Mocroft. Chaos as a possible model of electromyographic activity. *International Conference on Simulation 1998*, pages 27-34, Oct., 1998. York, UK.

[93] P. Padmanabhan and S. Puthusserypady. Nonlinear analysis of EMG signals: a chaotic approach. *Annual International Conference of the IEEE Engineering in Medicine and Biology*, 1(1):608-611, Sep., 2004.

[94] H.L. Liang, H. Wang and Z.Y. Lin. Reduction of electrocardiogram interference from diaphragmatic electromyogram by nonlinear filtering. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 1:17-18, 2002.

[95] R.E. Garfield, K. Chwalisz, L. Shi, G. Olson and G.R. Saade. Instrumentation for the diagnosis of term and preterm labour. *Journal of perinatal medicine*, 26(6):413-436, 1998.

[96] W.L. Maner, R.E. Garfield, H. Maul, G. Olson and G. Saade. Predicting term and preterm delivery with transabdominal uterine electromyography. *Obstetrics and Gynecology*, 101(6):1254-1260, Jun., 2003.

[97] N. Agarwal, A. Suneja, S. Arora, O.P. Tandon and S. Sircar. Role of uterine artery velocimetry using color-flow Doppler and electromyography of uterus in prediction of preterm labor. *Obstetrics and Gynecology*, 30(6):402-408, Dec., 2004.

[98] R.E. Garfield, H. Maul, W. Maner, C. Fittkow, G. Olson, L. Shi and G.R. Saade. Uterine electromyography and light-induced fluorescence in the management of term and preterm labor. *Journal of the Society for Gynecologic Investigation*, 9(5):265-275, Oct., 2002.

[99] M. Doret, R. Bukowski, M. Longo, H. Maul, W.L. Maner, R.E. Garfield, and G.R. Saade. Uterine electromyography characteristics for early diagnosis of mifepristone-induced preterm labor. *Obstetrics and Gynecology*, 105:822-830, 2005.

[100] M. Sabry-Rizk, W. Zgallai, E.R. Carson, P. Hardiman, A. MacLean and K.T.V. Grattan. Nonlinear dynamic tools for characterizing abdominal electromyographic signals before and during labour. *Transactions of the Institute of Measurement and Control*, 22(3):243-270, Sep., 2000.

[101] B. Azzerboni, M. Carpentieri, F. La Foresta and F.C. Morabito. Neural-ICA and wavelet transform for artifacts removal in surface EMG. *IEEE International Conference on Neural Networks - Conference Proceedings*, 4:3223-3228, 2004.

[102] W.D. Zhou and J. Gotman. Removal of EMG and ECG artifacts from EEG based on wavelet transform and ICA. *Engineering in Medicine and Biology Society*, 1:392-395, Sep., 2004.

[103] T. Ehtiati and W. Kinsner. Multifractal characterization of electromyogram signals. *Canadian Conference on Electrical and Computer Engineering*, 2:792-796, 1999.

[104] T. Ehtiati, W. Kinsner and Z.K. Moussavi. Multifractal characterization of the electromyogram signals in presence of fatigue. *Canadian Conference on Electrical and Computer Engineering*, 2:866-869, 1998.

[105] M. Bodruzzaman, S. Devgan and S. Kari. Chaotic classification of electromyographic (EMG) signals via correlation dimension measurement. *Conference Proceedings-IEEE SOUTHEASTCON*, 1:95-98, 1992.

[106] X. Hu, Z.Z. Wang and X.M. Ren. Classification of forearm action surface EMG signals based on fractal dimension. *Journal of Southeast University (English Edition)*, 21(3):324-329, Sep., 2005.

[107] X. Hu, Z.Z. Wang and X.M. Ren. Classification of surface EMG signal with fractal dimension. *Journal of Zhejiang University: Science*, 6B(8):844-848, Aug., 2005.

[108] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189-208, Oct., 1983.

[109] D.Q. Li, W. Pedrycz and N.J. Pizzi. Fuzzy wavelet packet based feature extraction method and its application to biomedical signal classification. *IEEE Transactions on Biomedical Engineering*, 52(6):1132-1139, Jun., 2005.

[110] X. Hu, Z.Z. Wang and X.M. Ren. Classification of surface EMG signal using relative wavelet packet energy. *Computer Methods and Programs in Biomedicine*, 79(3):189-195, Sep., 2005.

[111] S.P. Lee, J.S. Kim and S.H. Park. Enhanced feature extraction algorithm for EMG pattern classification. *IEEE Transactions on Rehabilitation Engineering*, 4(4):439-443, Dec., 1996.

[112] C.I. Christodoulou and C.S. Pattichis. Unsupervised pattern recognition for the classification of EMG signals. *IEEE Transactions on Biomedical Engineering*, 46(2):169-178, February 1999.

[113] J.D. Costa and R.E. Gander. Classification of muscle contraction levels with an artificial neural network. *Engineering in Medicine and Biology Society, Proceedings of the 15th Annual International Conference of the IEEE*, 15(1):259-260, 1993.

[114] A. Asres, H. Dou, Z.Y. Zhou, Y.L. Zhang and S.C. Zhu. Combination of AR and neural network technique for EMG pattern identification. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 4:1464-1465, 1996.

[115] B. Thompson, P. Picton, N.B. Jones. A comparison of neural network and traditional signal processingtechniques in the classification of EMG signals. *IEE Colloquium on Artificial Intelligence Methods for Biomedical Data Processing*, 100:1-5, April 1996.

[116] Y.L. Tian and A. Hampapur, Robust salient motion detection with complex background for real-time video surveillance, *IEEE Computer Society Workshop on Motion and Video Computing*, Jan. 5-6, 2005. Breckenridge, Colorado.

[117] Z.L. Gan, P. Zhang, X.C. Zhu. New motion detection algorithm and its implementation based on video surveillance, *Journal of Nanjing Institute of Posts and Telecommunications*, 24(1):46, Mar., 2004.

[118] C. Bahlmann, Y. Zhu, Y. Ramesh, M. Pellkofer and T. Koehler. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information, *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 255-260, 2005.

[119] A. Elnagar and A. Basu. Motion detection using background constraints, *Pattern Recognition*, 28(10):1537-1554, Oct., 1995.

[120] J.W. Wu and M. Trivedi. Performance characterization for gaussian mixture model based motion detection algorithms, *Proceedings of International Conference on Image Processing*, 1:1097-1100, 2005.

[121] P.R.R. Hasanzadeh, A. Shahmirzaie and A.H. Rezaie. Motion detection using differential histogram equalization, *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, pages 186-190, 2005.

[122] D.X. Zhou and H. Zhang. Modified GMM background modeling and optical flow for detection of moving objects, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 3:2224-2229, 2005.

[123] G. Jing, C.E. Siong and D. Rajan. Foreground motion detection by difference-based spatial temporal entropy image, *IEEE Conference Proceedings: Analog and Digital Techniques in Electrical Engineering*, pages 379-382, 2004.

[124] M. Hu and Z.P. Chen. Spatial-temporal difference method for detecting small moving targets in visible image background clutter, *International Conference on Signal Processing Proceedings*, 2, 2007.

[125] A. Kuh and D. Mandic. Sequential detection using least squares temporal difference methods, *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 5:701-704, 2006.

[126] J. Lin, J.H. Xu, W. Cong, L.L. Zhou and H. Yu. Research on real-time detection of moving target using gradient optical flow, *IEEE International Conference on Mechatronics and Automation*, pages 1796-1801, 2005.

[127] H. Ishiyama, T. Okatani and K. Dequchi. High-speed and high-precision optical flow detection for real-time motion segmentation, *Proceedings of the SICE Annual Conference*, pages 751-754, 2004.

[128] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774-780, Aug., 2000.

[129] J. Barron, D. Fleet and S. Beauchemin. Performance of optical flow techniques, *International Journal of Computer Vision*, 12(1):42-77, 1994.

[130] H. Liu, T. Hong, M. Herman and R. Chellappa. Accuracy vs. efficiency trade-offs in optical flow algorithms, *The Proceeding of Europe Conference Of Computer Vision*, 1996.

[131] B. Galvin, B. McCane, K. Novins, D. Mason, and S.Mills. Recovering motion fields: an evaluation of eight optical flow algorithms, *The Proceeding Of the 9th British Machine Vision Conference*, 1:195-204, Sep., 1998.

[132] Wikipedia, the free encyclopedia. *Lucas Kanade method.* Available: http:\\en.wikipedia.org/wiki/Lucas_Kanade_method, 2007.

[133] J. Lopez, M. Markel, N. Siddiqi, G. Gebert and J. Evers. Performance of passive ranging from image flow, *IEEE International Conference on Image Processing*, 1:929-932, Sep. 2003.

[134] H.J. Elias, O.U. Carlos and S. Jesus, G. Gebert and J. Evers. Detected motion classification with a double-background and a neighborhood-based difference, *Pattern Recognition Letters*, 24(12):2079-2092, Aug., 2003.

[135] J. Zheng, B. Li, B. Zhou and W. Li. Fast motion detection based on accumulative optical flow and double background model, *The Proceeding of International Conference in Computer Science, Computational Intelligence and Security*, 3802:291-296, 2005.

[136] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*, Prentice-Hall, ISBN: 0-130-94650-8, Dec., 2002, Second Edition.

[137] E.H. Anderson, P.J. Burt and G.S. Wal. Change detection and tracking using pyramid transformatin techniques, *Proceedings of SPIE-Intelligent Robots and Computer Vision, SPIE Press*, 579:72-78, 1985.

[138] P.J. Burt. Smart sensing within a pyramid vision machine, *Proceedings of the IEEE*, 76(8):1006-1015, Aug., 1988.

[139] J.Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm, *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, 2000.

[140] M. George and T. Tjahjadi. Multi-resolution optical flow estimation using adaptive shifting, *IEEE International Conference on Image Processing*, 3:717-721, Oct., 1999.

[141] B.W. Jeon, G.I. Lee, S.H. Lee and R.H. Park. Coarse-to-Fine frame interpolation for frame rate up conversion using pyramid structure, *IEEE Transactions on Consumer Electronics*, 49(3):499-508, Aug., 2003.

[142] S.W. Song, M.Y. Liao and J.M. Qin. Multiresolution image motion detection and displacement estimation, *Machine Vision and Applications*, 3(1):17-20, 1990.

[143] J.W. Zan, M.O. Ahmad and M.N.S. Swamy. Neighbourhood-blocks motion vector estimation technique using pyramidal data structure, *IEE Proceedings: Vision, Image and Signal Processing*, 149(3):140-151, Jun., 2002.

[144] J.W. Zan, M.O. Ahmad and M.N.S. Swamy. An efficient pyramid for motion estimation, *Midwest Symposium on Circuits and Systems*, 3:268-271, Aug., 2002.

[145] C.A. Seqall, W. Chen and S.T. Acton. Video tracking using morphological pyramids, *Journal of Electronic Imaging*, 8(2):176-184, 1999.

[146] P.J. Burt and E.H. Adelson. The Laplacian Pyramid as a Compact Image Code, *IEEE Transactions On Communications*, 31(4):532-540, Apr., 1983.

[147] Y.S. Choi, Z.J. Piao, T.H. Kim and C.B. Park. Salient motion information detection technique using weighted subtraction image and motion vector, *International Conference on Hybrid Information Technology*, 1:263-269, 2006.