Vessel Recognition In Color Doppler Ultrasound Imaging

Ashraf A Saad

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree: Department of Electrical Engineering

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

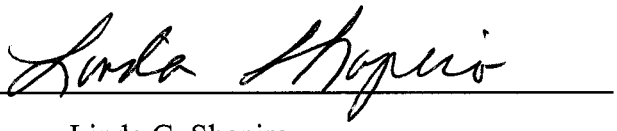
Ashraf A Saad

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

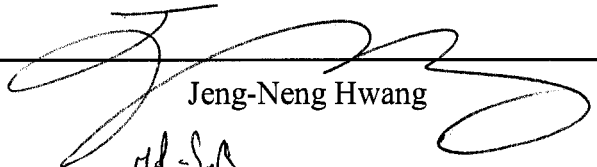
Chair of the Supervisory Committee:

Linda G. Shapiro

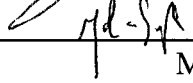
Reading Committee:



Linda G. Shapiro



Jeng-Neng Hwang



Maya Gupta

Date: 3/10/08

In presenting this dissertation in partial fulfillment of the requirements of the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature Ashraf

Date 3/10/2008

University of Washington

Abstract

Vessel Recognition in Color Doppler Ultrasound Imaging

Ashraf A Saad

Chair of the Supervisory Committee:
Professor Linda G. Shapiro
Electrical Engineering

Diagnostic ultrasound imaging is an important non-invasive tool for diagnosis and treatment decisions with a variety of clinical applications such as vascular disease, cardiology disease, and obstetrics. Color Doppler ultrasound imaging modality in particular offers complete 2-D cross sectional images depicting velocities of moving reflectors, such as blood flow. Color Doppler imaging is a crucial diagnostic tool for vascular disorders such as arterial atherosclerosis and deep vein thrombosis. Nowadays the vascular ultrasound exam is a tedious and time-consuming process that requires a high level of expertise. This is mainly due to the large number of manual controls offered by the ultrasound system to the clinical users to optimize the images for the best clinical outcome. Another reason is the lack of automation and quantification tools by the ultrasound system since the system utilizes very little high-level information from the acquired images. Moreover musculoskeletal injuries due to operating the ultrasound systems have become very common and a great concern for the medical community. Another concern is the standardization of the clinical exam outcome to minimize interobserver and intraobserver variability among diagnostic ultrasound practitioners. Introducing advanced automation and quantification capabilities to the ultrasound system will help with all these issues and streamline the clinical exam operation and outcome. In this thesis a complete vessel recognition system based on the analysis of the color Doppler ultrasound images is developed. The system includes the design and realization of many automated tasks including image acquisition, vessel segmentation, true velocity restoration, feature extraction, and classifier design. The involved tasks represent the building blocks for the automation and quantification applications that can be incorporated within the ultrasound system to achieve ease of use and more standardized clinical outcome. The thesis will also address a number of important clinical problems in the field of ultrasound vascular imaging and propose novel solutions that rely on the developed image understanding techniques.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

This thesis would never have been completed without the contribution and encouragement of many people. I wish to express my sincere appreciation to my advisor, Professor Linda Shapiro, for her guidance, encouragement, and support throughout my Ph.D. program. I would like to thank my advisory committee members, Professor Maya Gupta and Professor Jeng-Neng Hwang, for their valuable suggestions and support.

I thank Dr. Thanasis Loupas and Dr. Jeff Powers, from Philips Ultrasound, for their encouragement and help with revising my published papers and exchanging ideas. I thank my colleagues at the vision group and at Philips Ultrasound for their valuable discussions and suggestions.

I thank the department of electrical engineering at the University of Washington for supporting my work. I also thank Philips Ultrasound for partially sponsoring my Ph.D. study and for providing access to the ultrasound equipments.

I thank Dr. Orpheus Kolokythas and Dr. Manjiri Dighe from the University of Washington Radiology Department for their support and exchanging clinical ideas that can benefit from this research.

I thank David Hull, Jayne Armfield, Sue Blanchard, Mary C Flynn, and Steve Hill from Philips Ultrasound for their contribution and performing the clinical experiments included in this thesis.

Finally I would like to thank all my family for their encouragement, support, and patience during my Ph.D. study.

DEDICATION

To my parents Ali and Salwa, to my family Dalia, Ali, and Malak, and to my brothers Mohamed, Wael, Ahmed, and Yahia.

Chapter 1 Introduction

Ultrasound is a sound wave with frequency above the audible range that is greater than 20 kHz. In diagnostic ultrasound imaging, the frequency range is between 2 MHz and 10 MHz. Ultrasound imaging systems transmit a beam of ultrasound into the body and collect and analyze the returning echoes. By steering the beam to different directions and acquiring multiple lines, the system constructs a cross section image of the body.

The story of the development of diagnostic ultrasound [1] started with the discovery, made by the Italian biologist Lazzaro Spallanzani in 1794, that the ability of bats to navigate accurately in the dark was through echo reflection from high frequency inaudible sound waves. In 1826, the Swiss physicist Jean-Daniel Colladon successfully used an underwater bell to determine the speed of sound in the waters of Lake Geneva. The term SONAR refers to "Sound Navigation and Ranging." Diagnostic ultrasound systems can be regarded as a form of 'medical' SONAR. Lord Rayleigh in England published "the Theory of Sound" in 1877 that described a sound wave as a mathematical equation, forming the basis of further theoretical work in acoustics. The real breakthrough in the evolution of high frequency echo-sounding techniques came when the piezo-electric effect in certain crystals was discovered by the brothers Pierre and Jacques Curie in France in 1880. World Wars I and II inspired rapid developments and refinements in the naval and military SONAR by researchers in France and the United States.

The use of ultrasound waves in the medical field started initially with therapeutic applications rather than diagnostic ones, utilizing its heating and disruptive effects on animal tissues as early as 1920. The Austrian neurologist Karl Theo Dussik was generally regarded as the first physician to have employed ultrasound in medical diagnosis, which occurred in 1930. Dussik attempted to locate brain tumors and the cerebral ventricles by measuring the transmission of the ultrasound beam through the skull. In 1963, Physionics Engineering® Inc. in Colorado commercially launched the first hand-held, articulated-arm diagnostic ultrasound scanner in the United States. Aloka® produced Japan's first commercial medical ultrasound

scanner in 1960. The "First World Congress on Ultrasonic Diagnostics in Medicine" was held in Vienna in 1969. The innovation which soon completely changed the practice of ultrasound scanning was the advent of the real-time scanners, which were developed by Walter Krause and Richard Soldner and manufactured as the Vidoson® system by Siemens Medical Systems of Germany in 1965.

The Doppler principle was first described in 1842 by Christian Andreas Doppler in Austria. Medical applications of the Doppler ultrasound techniques were first introduced in 1955 by Shigeo Satomura and Yasuhara Nimura at the Institute of Scientific and Industrial Research in Osaka, Japan, for the study of cardiac valvular motion. In the United States, pioneer work in Doppler instrumentation started at the University of Washington in Seattle from 1958 onwards led by the pediatrician Robert Rushmer who attempted to characterize cardiovascular functions in animals. Rushmer recruited a number of engineers including Donald Baker for the cardiovascular instrumentation development program. The team pioneered transcutaneous continuous-wave flow measurements and spectral analysis in 1963, and the vascular surgeon D. Eugene Strandness undertook the clinical testing. Strandness published the first transcutaneous vascular spectral flow signals in 1964. ATL® (Advanced Technology Laboratories) was founded in 1969 near Seattle, Washington, by a small group of engineers developing marine electronic systems. Starting in 1973, technology developed at the University of Washington Center for Bioengineering was transferred to ATL in the development of systems for diagnostic sonography. The first duplex pulsed-Doppler scanner, the Mark1, was released in 1974 for cardiac investigation.

Further development led to 2-D color flow imaging. In 1975 Marco Brandestini and his team at the University of Washington obtained blood-flow images using a multi-gated pulsed Doppler system, where velocity waveforms and flow images were encoded in color and superimposed on gray scale 2-D anatomical images. They were able to demonstrate the value of color-flow imaging in the diagnosis of various cardiac defects. It was the work of Chihiro Kasai, Koroku Namekawa, Ryozo Omoto and co-workers in Tokyo, Japan, which was published in the English language in 1985, that led to the realization that real-time color-flow

imaging could be practically possible. Namekawa and Kasai were bio-engineers working for Aloka®, and Omoto was a cardiologist at the Saitama Medical School in Tokyo.

The Combison 330 system, from Kretztechnik®, which appeared in 1989, was the first commercial 3-D scanner in the market. In 1987, the Center for Emerging Cardiovascular Technologies at Duke University started a project to develop a real-time volumetric scanner for imaging the heart. In 1991 they produced a matrix array scanner that could image cardiac structures in real-time and 3-D. In 1994, Olaf von Ramm, Stephen Smith and their team produced an improved scanner that could provide good resolution down to 20 centimeters. The team developed state-of-the-art "medical ultrasound imaging" integrated circuits which were capable of processing signals from multiple real-time phased-array images, which became available commercially from Volumetric Medical Imaging, Inc. in 1997.

The advances in diagnostic ultrasound imaging continue to develop rapidly in both the technical and clinical aspects. On the technical side, new technologies, such as harmonic imaging, elastography, and high-resolution flow imaging have appeared and continue to grow in the 21st century. On the clinical side, diagnostic ultrasound imaging has become a primary diagnostic tool for many clinical applications including obstetrics [1], cardiology [2], and vascular disease [3]. Figure 1.1 shows a picture of one of the state of the art commercial ultrasound systems, the Philips iU22 ultrasound system [4].

The advances in diagnostic ultrasound mainly have focused on transducer design, ultrasound beam formation, signal processing, and visualization. So far, ultrasound systems use very little knowledge about the content of the images they acquire to optimize the acquisition and visualization processes. The main consequence of this fact is that machine manipulation is still a tedious and time-consuming task with too many controls to adjust. This requires experienced, highly-trained users (sonographers and physicians) to operate the system. Another consequence is the lack of standardized diagnostic results, since the quality of the acquired images depends greatly on the experience level of the sonographers. This makes it harder sometimes for the physicians (radiologists or cardiologists) to reach consistent diagnostic and treatment decisions. One such example of the ultrasound exam outcome

variability is the assessment of the stenosis degree of the carotid artery, which relies mainly on the estimation of the blood flow velocity from the spectral Doppler waveform. The medical community has attempted to propose a standard procedure to diagnose and assess the degree of stenosis using ultrasound Doppler techniques [5]. However it is still believed that the interobserver variability and human factor errors are the main source of disagreement in the clinical outcome for such an important clinical problem [6] [7].

Figure 1.1 An example of a diagnostic ultrasound system, the Philips iU22 system.

Another problem associated with the complexity of operating the ultrasound system is the increased number of musculoskeletal injuries among diagnostic ultrasound practitioners. Many studies attempted to identify the sources of causes of the work-related musculoskeletal injuries among ultrasound operators. Murphy and Russo [8] reported on the findings of the ultrasound project and survey sponsored by the Employee Health and Safety Services (EHS) at Healthcare Benefit Trust (HBT) in Canada to quantify the physical and psychosocial stressors and symptoms related to ultrasound tasks. They found the tasks reported to aggravate musculoskeletal symptoms the most were applying sustained pressure on the ultrasound transducer; abduction of the shoulder; sustained twisting of the neck and trunk; and repetitive twisting of the neck and trunk. Smith et al. [9] found a positive relationship between

musculoskeletal pain and frequency of scans (100 or more scans per month), and scan time (average scan time of 25 minutes or more per patient). Magnavita et al. [10] showed that discomfort from the transducer design was the best predictor of hand-wrist complaints. Mirk et al. [11] showed that muscular efforts such as gripping the transducer, applying sustained pressure and scanning with a flexed or hyperextended wrist were significantly correlated with increasing severity of symptoms in the hand, wrist and forearm area. All of these findings confirm the complexity of operating the ultrasound system and call for more automation and quantifications tasks that would help the medical community to reach the high quality diagnostic decisions easier and faster.

Based on the previous analysis, the main goal of our research is to "*Develop advanced image understanding techniques for the analysis and processing of ultrasound images in order to build an infrastructure for automation and quantification tasks that will improve the ease of use of the ultrasound system and help to standardize the outcome of the clinical exam.*" However due to the broad range of clinical applications served by ultrasound and the many different imaging modalities offered by ultrasound systems, we chose to focus on one important clinical application, *vascular imaging*, and to focus on one primary ultrasound imaging modality for that application, *color Doppler imaging*.

Ultrasound is a major screening and diagnosis tool for vascular diseases. Arterial diseases, such as stroke, are leading causes of morbidity and mortality in the United States, while venous diseases, such as vein thrombosis and pulmonary embolism, cause the incapacitation or death of thousands of patients each year [3].

Arteries and veins are 3-D tubular objects usually lying adjacent to each other. Arteries serve as conduits that carry oxygenated blood from the heart to the tissues, while veins carry deoxygenated blood back to the heart. The arterial blood flow is pulsatile due to heart beats and can be laminar or turbulent. Laminar flow is a steady flow with parabolic distribution across the vessel cross section, while turbulent flow happens with vessel narrowing or tortuous sites where blood moves in random directions. The venous blood flow is non-pulsatile but may be phasic due to respiration [3].

During a vascular exam, sonographers look at the anatomy and hemodynamics of arteries and veins of interest. The composite grayscale and color Doppler real-time imaging capability demonstrates both the anatomical and hemodynamic features of arteries and veins simultaneously. The main features of the ultrasound color Doppler images that are important to consider in any image understanding technique can be summarized as follows:

- The images are randomly-oriented cross sections of arteries or veins, since the ultrasound transducer can be oriented in any angle over the body site of interest. The most important orientation is the longitudinal cross section where the vessel appears in the grayscale image as two parallel echogenic (high contrast) lines that are separated by an anechoic (low contrast) lumen, while the color Doppler data occupies the vessel lumen. Transverse cross sections are also common where the vessel appears as a circle or an ellipse filled with color data.
- The pulsatility of the arterial blood flow is prominent in the color Doppler data; during the systolic phase the blood fills in the vessel lumen, while during the diastolic phase the blood filling is greatly minimized. The venous blood flow is almost non-pulsatile and steady.
- Due to the real-time nature of the ultrasound imaging modality, the vessel movement due to heart beating, respiration, and transducer movement is evident in image sequences.

To better understand the complexity of the ultrasound vascular exam, we decided to watch a number of exams at the University of Washington Radiology Department and at Philips Ultrasound Clinical Labs in Bothell, WA. We studied the ultrasound exam clinical workflow and control manipulation. Although the workflow varies among applications, diseases, and sonographers, it can be generalized as described below.

Sonographers typically start with grayscale imaging mode where they research the anatomy and locate the vessels of interest. To get the best image quality, they can manipulate a number of controls, such as scanning depth, beam focus, gain, and dynamic range. After getting the best grayscale image of the vessel of interest, sonographers switch to color mode where the color Doppler data is acquired within a region of interest called the color box, as

highlighted in Figure 1.2. In this figure, a screen shot of a typical ultrasound composite image of grayscale, color Doppler and spectral Doppler data is shown. The grayscale and color Doppler images are overlaid on top of each other and displayed concurrently in real-time while the spectral Doppler data scrolls with time.

Sonographers also manipulate a number of controls in the color mode to get the best color image quality. For example they may change the color box position and size to center it on the vessel's site of interest; or they may change the color box steering to increase the sensitivity of the color data by avoiding perpendicular angles as much as possible.

After adjusting the system to acquire the optimal grayscale and color images of the vessel of interest, sonographers switch to grayscale/color/spectral Doppler mode to acquire Doppler spectrum data for blood flow quantification purposes. In this mode, they usually adjust the spectral Doppler acquisition gate's position and size to align the gate, which is called the Doppler sample volume, with the vessel's site of interest from which to acquire spectral Doppler data. They may adjust the spectral Doppler line steering to increase the sensitivity and avoid perpendicular angles. They also adjust the Doppler angle line to specify the flow orientation angle with respect to the Doppler steering line to be used in the Doppler equation. All these controls are highlighted in Figure 1.2. Afterwards, sonographers capture spectral Doppler data from the vessel's site of interest for quantification purposes. The quantification usually includes measuring the highest velocities, mean velocities, and pressure gradients. The previous steps are repeated several times during the vascular exam for every vessel of interest and for every site of interest on each scanned vessel. For example, stenotic vessels must be investigated in multiple sites (pre-stenotic, stenotic, and post-stenotic sites) to construct a good hypothesis about the velocity profile and hemodynamics of the vessel.

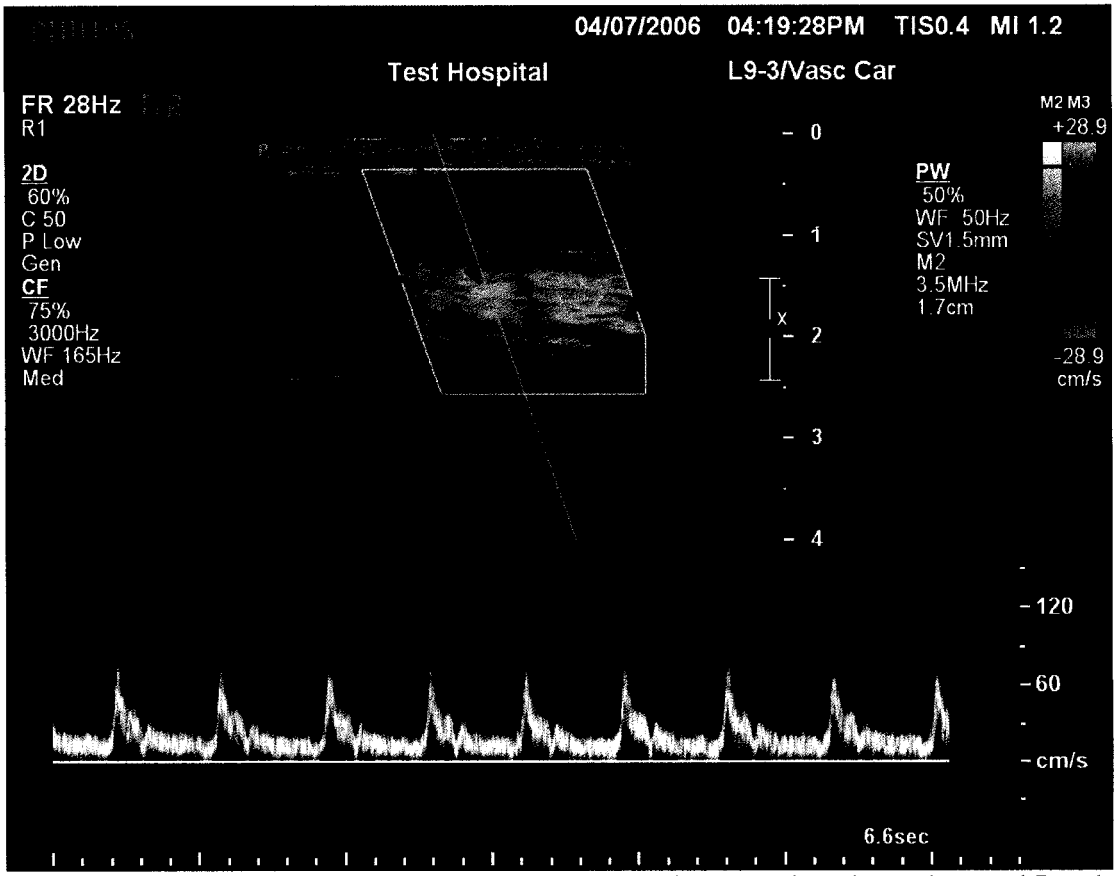


Figure 1.2 A screen capture of an ultrasound system composite grayscale, color, and spectral Doppler image of the carotid artery, showing all the graphical components that the user manipulates to optimize the image.

It is evident from this study that the manipulation of the current ultrasound systems during the vascular exam is a tedious and time-consuming task with too many controls to adjust. In addition, the sonographer's expertise level plays a major role in the quality of the exam outcome, which is used by the radiologists for diagnosis purposes. The majority of the time and effort is spent on manipulating the system controls to acquire high quality color Doppler images of different blood vessels and interrogate these vessels in order to assess the disease state of the patient. If we can add knowledge to the ultrasound system about the underlying vessels within the images, then it would be possible to automate many of the previously-mentioned manual operations and simplify the exam.

The simplest form of knowledge is to detect a vessel in the ultrasound image and use this information to optimize some system settings that are generic to all vessels, such as the acquisition depth, acquisition box size, and steering angle. However other settings, such as the ideal sampling rate and gain, vary from one vessel to another. Additionally it is common for more than one vessel to appear in the same image, and identifying generic vessels will not be sufficient to optimize the system according to the user's intended vessel. The ultimate form of knowledge in this case would be to recognize the specific vessels in the image and optimize the system based on the vessel of interest determined by some context information provided by the user, such as the exam type. For example, both the carotid artery and the jugular vein typically appear concurrently in the ultrasound image. If the system is capable of distinguishing the two vessels, then it will be straightforward to optimize the system around the carotid artery if an arterial exam type is selected, or around the jugular vein if a venous exam type is selected.

Based on this argument we refine our research goal to be: "*Develop advanced image understanding techniques to recognize specific blood vessels in color Doppler ultrasound images in order to build an infrastructure for automation and quantification tasks that would achieve ease of use of the ultrasound system and help to standardize the outcome of the vascular exam.*"

The problem of vessel recognition is formulated as follows: "*Given a number of color Doppler ultrasound frames that contain one or more vessels, recognize all the specific vessels that exist in those images.*" The steps involved in the vessel recognition system based on the analysis of the color Doppler ultrasound images are summarized in Figure 1.3. The first stage is to acquire the color Doppler ultrasound images from the ultrasound system along with the supporting acquisition information that is necessary for further processing. In Chapter 2, we will briefly discuss the fundamentals of the ultrasound imaging physics and acquisition process that are essential for guiding the development of the recognition system algorithms.

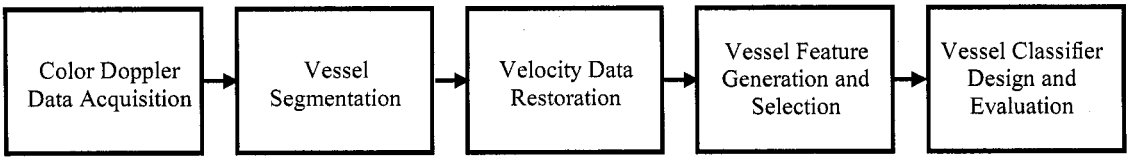


Figure 1.3 The vessel recognition system building blocks.

The second step of the vessel recognition system is the segmentation stage where the distinct vessels within the color Doppler ultrasound images are segmented. Although the vessels are almost self-segmented in color Doppler images since they are the only structure that appear in those images, there are still some situations where a segmentation algorithm is necessary, such as multiple vessels connected together or the presence of motion-induced noise. In Chapter 3, we will discuss the image acquisition and segmentation steps where a novel shape decomposition-based approach for vessel segmentation will be described. A novel clinical application of the segmentation results, which is the automation of the Doppler angle optimization that is crucial to the vessel stenosis grading [5] will also be described.

The third step of the vessel recognition system is a data preprocessing stage, where the sources of distortion in color Doppler images will be treated. Specifically the color Doppler aliasing problem, described in Chapter 2, that causes distortion of the true velocity values in color Doppler images will be handled. In Chapter 4, we will formulate the problem of color aliasing as a phase-wrapping problem and describe the theoretical foundations and literature review of the phase-unwrapping research field. A number of well-known phase unwrapping techniques will be applied to the color Doppler ultrasound images and their performance will be documented. In Chapter 5, a novel phase-unwrapping technique will be described that is customized for the color Doppler ultrasound aliasing problem. The performance of the new technique will be discussed in contrast to the performance of the existing phase-unwrapping techniques. Another novel clinical application that benefits from the velocity restoration results is the real-time display of the unwrapped velocities that should enable sonographers and physicians to locate the stenotic site easier and more accurately [3].

The fourth step of the vessel recognition system is the feature generation and selection step where discriminating features for the distinct vessels are extracted from the segmented

and preprocessed color Doppler ultrasound images. In Chapter 6, we will discuss several data reduction methods followed by different feature generation techniques applied to the color Doppler ultrasound data.

The fifth and final step of the vessel recognition system is the classifier design and evaluation. In Chapter 7, we will evaluate the performance of several classifiers with the vessel recognition problem, including several attribute selection and preprocessing methods, in order to determine the best design of the vessel recognition system. Finally in Chapter 8, we will discuss the conclusions and future work of this thesis.

Chapter 2 Ultrasound Imaging Physics and Acquisition

2.1. Introduction

In order to develop image analysis techniques for color Doppler ultrasound images, it is important to understand the physical properties and acquisition processes behind the formation of these images. This understanding should guide the development of robust image processing techniques for the recognition purpose of the blood vessels within color Doppler images.

In this chapter we will present a brief description of the physics, system design, and signal processing principals behind the formation of the ultrasound images. We will start with the basic physics behind sound waves production, propagation, and reflection. This will explain the interaction of the ultrasound beam with the body tissues it images.

We will then discuss ultrasound transducers, which are the part of the ultrasound device responsible for transmitting and receiving ultrasound energy into the human body. We will present the basic process of transmitting an ultrasound beam in a specific direction and receiving the reflected echoes from soft tissue boundaries.

Next we will discuss the three main image types produced by a diagnostic ultrasound system. The first image type is the grayscale anatomical image of the body organs, which is called the B-mode (or brightness) image. The second image type is the spectral Doppler display: a scrolling display of the blood velocity distribution versus time. The third type of images is the color Doppler image, which represents a 2-D map of the mean velocity of blood flow within human blood vessels. We are most interested in analyzing color Doppler images in this research; however it is equally important to understand the B-mode and spectral Doppler image types as well. Finally we will present the ultrasound image artifacts that will affect the image analysis techniques developed in this work.

2.2. Basic Ultrasound Physics

Sound is a mechanical energy transmitted by pressure waves in a material medium. "Mechanical" implies that the energy exists in the form of vibrations of the particles in the medium. "Waves" implies that the sound travels as a coordinated disturbance at a fixed speed through the medium. A wave carries energy through a medium without net movement of the particles. "Material medium" implies that sound waves require a medium, such as air, water, or body tissue, in which to travel. The production of sound requires a vibrating object, such as a person's voice box, a musical instrument, or an audio speaker system [13].

In diagnostic ultrasound the source of sound waves is a piezoelectric transducer, which vibrates in response to an electrical impulse. The *frequency* is the number of oscillations per second that the particles in the medium make as they vibrate about their resting position. The vibration frequency of the ultrasound transducer is between 1 MHz and 20 MHz , which is much higher than the audible frequency range, which is between 20 Hz and 20 KHz. The audible frequency range, while the diagnostic ultrasound frequency range. An ultrasound transducer is usually characterized by its frequency, which affects many aspects of the resulting image including the spatial resolution and penetration.

The acoustic *wavelength* is inversely proportional to the frequency. The wavelength is the distance between two peaks or valleys on the ultrasound wave. The relationship between the wave frequency and its wavelength is given by the following equation:

$$\lambda = \frac{c}{f} \qquad (1)$$

where λ is the wavelength, f is the frequency, and c is the speed of sound in soft tissue, which is estimated to be 1540 m/s in average. The wavelength for diagnostic ultrasound is typically 1 mm or less. The wavelength is important in ultrasound physics, since it affects the spatial resolution as well as the interaction with physical objects, such as transducer elements

or reflecting surfaces, when the object size is comparable to the wavelength of the ultrasound wave.

Reflection of the ultrasound beam occurs whenever the beam is incident on an interface formed by two tissues having different acoustic impedances, as shown in Figure 2.1. The *acoustic impedance* Z is equal to the product of the medium's density ρ and its speed of sound c as follows:

$$Z = \rho c \qquad (2)$$

The magnitude of the reflected wave depends on the acoustic impedance difference according to the following equation:

$$\frac{I_r}{I_i} = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \qquad (3)$$

where I_i is the incident intensity, I_r is the reflected intensity, Z_1 is the impedance on the proximal side of the tissue interface, and Z_2 is the impedance on the distal side of the interface. Interfaces where there is a large impedance difference, such as air/tissue or tissue/bone interface, reflect more of the incident beam. This explains the use of a coupling medium, called ultrasound gel, between ultrasound transducers and patients during the examination. It also explains why ultrasound is not used to image bones and lungs.

There are three types of reflectors that contribute to the B-mode ultrasound image. The first of these is the *specular reflector* that represents a large smooth interface with dimensions much larger than the ultrasound wavelength, such as muscle-fat interfaces. If the ultrasound beam is incident on a specular reflector with a nonperpendicular angle, part of the beam will be reflected with an angle equal to the incidence angle but with opposite direction, while the rest of the beam will be refracted with a change of direction while passing through

that interface, as shown in Figure 2.2. Thus the sensitivity of the ultrasound images increases as the angles become closer to perpendicular to large tissue interfaces.

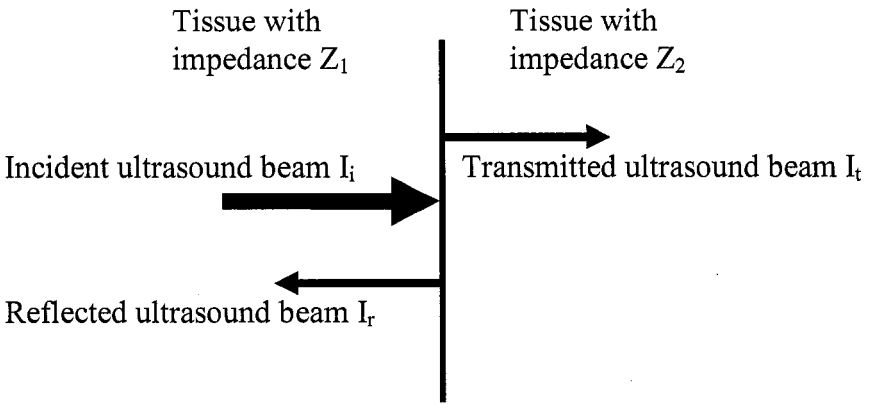


Figure 2.1 The reflection and transmission of ultrasound beams on smooth specular reflectors.

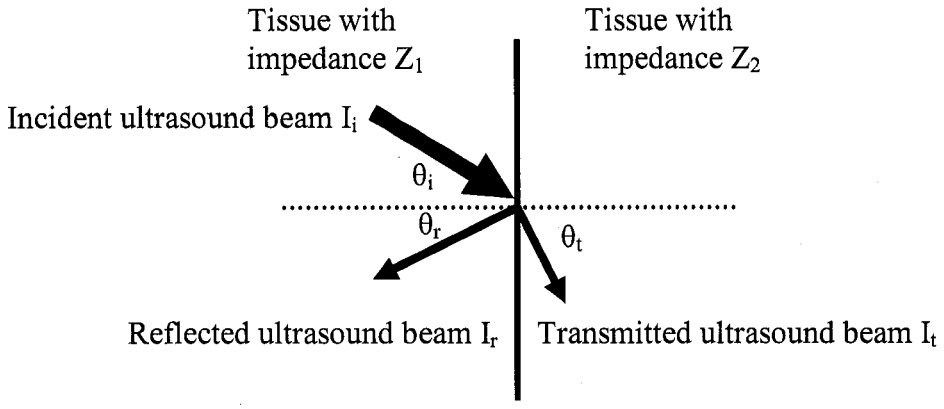


Figure 2.2 The reflection and transmission of ultrasound beams on smooth specular nonperpendicular reflectors.

The second type of reflector is called the *diffuse reflector* where the interface is not perfectly smooth and possesses a degree of roughness, such as heart chamber walls and vessel walls. With diffuse reflectors, the ultrasound beam is reflected into many directions with weaker amplitude than specular reflectors. However the image is not as sensitive as the

specular interface image to the incident angle of the beam. Figure 2.3 shows a grayscale B-mode image of the carotid artery and the jugular vein with the diffuse vessel wall reflectors.

The third type of reflector is called the *scattering reflector* where the size of the reflector is comparable or less than the wavelength of the ultrasound wave, such as the parenchyma of most organs. The reflected waves are scattered in all directions with weaker amplitude. Ultrasound scattering gives much of the diagnostic information seen in ultrasound imaging. The terms *hyperechoic* and *hypoechoic* are used frequently to describe structures in B-mode ultrasound images. Hyperechoic results from increasing the scattering level of the tissue, while hypoechoic results from decreasing the scattering level. For example liver tumors are usually characterized by hyperechoic regions compared to the normal liver tissue. . Figure 2.3 shows an example of a hyperechoic muscle tissue and a hypoechoic vessel lumen.

Figure 2.3 A grayscale B-mode ultrasound image of the carotid artery and jugular vein.

Red blood cells are called *Rayleigh scatterers,* since their dimensions are much less than the ultrasound wavelength. For example a typical ultrasound wavelength is 300 μm, which is much larger than the 8 μm diameter of a red blood cell. Scattering from Rayleigh scatterers increases with increasing the frequency.

The ultrasound beam faces attenuation as it travels through tissue. *Attenuation* implies reduced amplitude as a function of distance. The two main sources of attenuation are the

reflection and scatter of sound at tissue interfaces and the absorption of sound energy where it transforms to heat. However, the amount of heat in diagnostic ultrasound systems is too small to cause harm to the patient.

2.3. Ultrasound Transducers

Transducer refers to any device that is used to convert signals or energy from one form to another. Ultrasonic transducers convert electrical signal to acoustic energy and vice versa. They are used both as transmitters and receivers of ultrasound waves [13].

Transducers used in medical ultrasound employ the *piezoelectric effect* to generate sound waves and detect echo signals. The piezoelectric effect was discovered by Pierre and Jacques Curie in the 1880s. They found that when a force is applied perpendicular to the faces of a quartz crystal, an electrical charge results. Conversely, if an electrical signal is applied to the crystal, the crystal vibrates sending a sound wave into the medium. The piezoelectric effect and its inverse effect are shown in Figure 2.4.

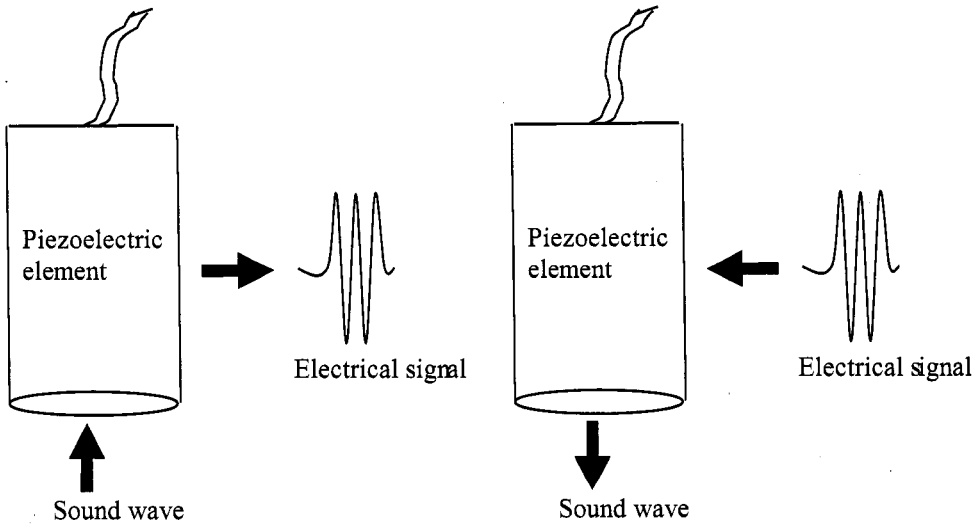


Figure 2.4 The piezoelectric effect (left) and its inverse (right).

Each transducer has a resonance frequency at which it is most efficient in converting electrical signals to acoustic energy and vice versa. The resonance frequency is determined mainly by the thickness of the piezoelectric element. In reality, ultrasound transducers are designed as broadband transducers with a range of operational frequencies that can be controlled by the ultrasound system circuitry itself. For example, the Philips vascular transducer L9-3 has a bandwidth of 3-9 MHz, which can be controlled by the system.

Most ultrasound transducers have a backing material that acts as a damping medium for the vibration. Damping the vibrations is necessary to produce pulses with very short durations, which achieve better axial resolution.

Most transducers also have impedance-matching layers to improve their sensitivity, that is, their ability to detect very low weak echoes. The acoustic impedance of some piezoelectric materials can be 20 times the impedance of the soft tissue. The impedance of the matching material is chosen in-between the impedances of the element and the tissue to reduce the reflections at the transducer-tissue interface.

In ultrasound imaging, ultrasound pulses are emitted by the transducer in a certain direction. The echo signals received from the reflectors along the beam path are displayed on the monitor. The ultrasound beam is swept across the region of interest, as shown in Figure 2.5. The image is constructed using echo arrival times to determine reflector depths and the beam axis position to determine lateral locations of reflectors.

The resolution of the resulting ultrasound image depends on many factors. There are two main types of resolutions: spatial resolution and temporal resolution. Spatial resolution includes axial resolution, lateral resolution, and elevation resolution. *Axial resolution* refers to the minimum reflector spacing along the axis of the ultrasound beam that results in separate echoes on the display. Axial resolution is determined by the pulse duration, which decreases as the ultrasound frequency increases. However increasing the ultrasound frequency increases the beam attenuation, which decreases the beam penetration. The choice of the ultrasound

frequency for a specific clinical application is always a tradeoff between the required resolution and the target depth.

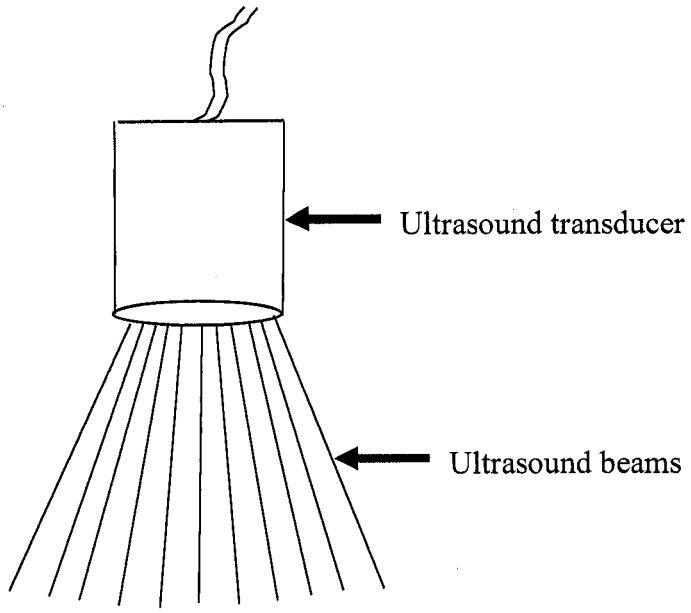


Figure 2.5 Ultrasound image construction using a sweeping beam technique.

Lateral resolution refers to the ability to distinguish between two closely spaced reflectors that are positioned perpendicular to the ultrasound beam axis. Lateral resolution depends on the transducer beam width. *Elevation resolution*, or slice thickness, dictates the thickness of the section of tissue that contributes to echoes visualized on the image. Elevation resolution depends on the beam size perpendicular to the image plane. Because of the beam slice thickness, the cross sectional ultrasound image is, in fact, constructed from a thick volume of tissue instead of a single plane. Figure 2.6 shows the three different spatial resolutions that characterize an ultrasound image.

The ultrasound beam is characterized by the *main beam,* which is aligned with the axis of the piezoelectric element, and by the *side lobes*, which represent off-axis energy. Side

lobes mainly degrade the lateral resolution if they are strong. To reduce the effect of the side lobes, a process called *element apodization* is applied to the signal that is applied to the piezoelectric elements. Instead of applying a uniform signal, a tapered signal is applied where the signal strength decreases as it gets closer to the element edges.

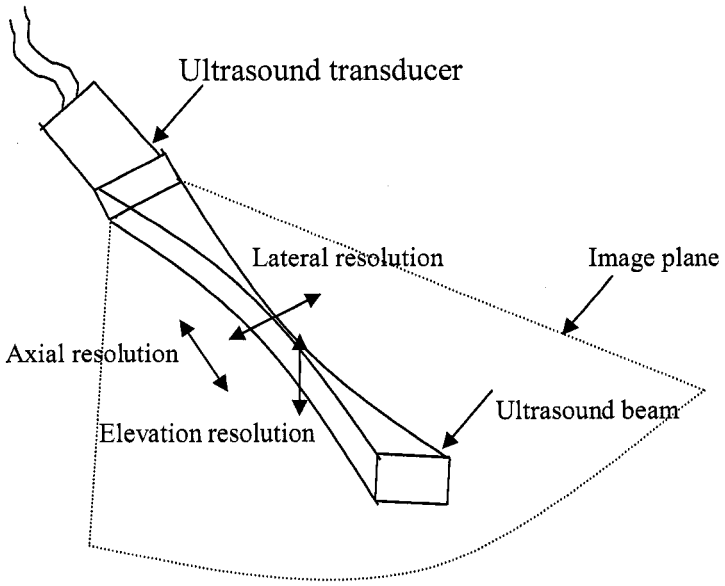


Figure 2.6 Ultrasound spatial resolution types.

Since the ultrasound beam tends to diverge as it travels away from the transducer, current ultrasound systems apply beam focusing techniques to enhance the lateral resolution. For single element transducers, an acoustic lens is used to achieve fixed focus for the ultrasound beam. For multiple element, or array, transducers, which are the most common transducers used by diagnostic ultrasound systems nowadays, the beam focusing is achieved by the electronic beam forming process [13].

An *array transducer* consists of a group of closely-spaced piezoelectric elements, each with its own electrical connection to the ultrasound system. An element with its pulser-receiver circuit is called a *channel*. The elements can be excited individually or in groups, called *apertures*, to construct ultrasound beams. The most common types of array transducers

are *linear array*, *curvilinear array*, and *phased array* transducers. Figure 2.7 shows the different types of ultrasound transducers.

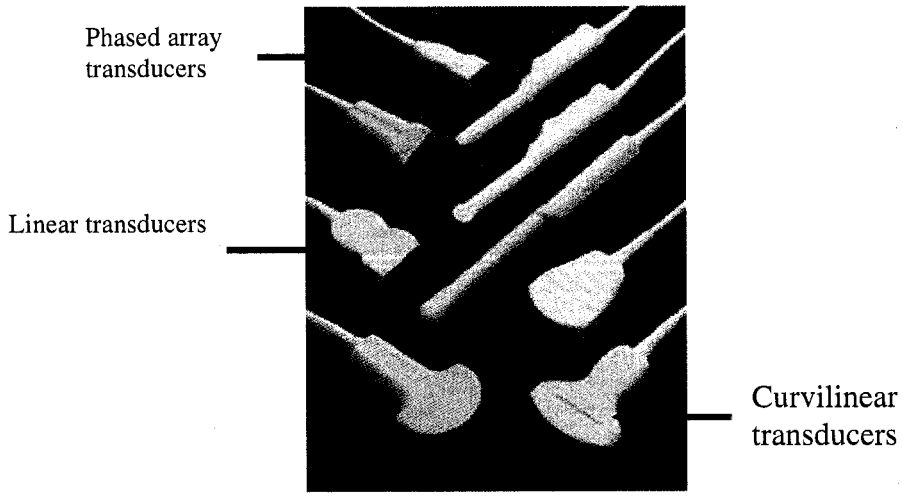


Figure 2.7 Ultrasound transducers.

In linear and curvilinear arrays, the elements are arranged side by side in a line or an arc shape. Individual beams are produced by exciting a group of elements, say 20 elements. The linear array produces a rectangular shaped image, while the curvilinear array produces a trapezoidal shaped image. Linear arrays are most suitable for vascular applications where wide field of views of shallow blood vessels are needed, while curvilinear arrays are most suitable for abdominal and obstetric applications where deeper organs are being imaged.

In phased array transducers, all elements are excited to produce a single beam. The form factor of phased array transducers are usually much smaller than that of linear arrays. Thus they are more suitable for cardiac applications, where the field of view is limited due to the rib bones and the lungs. Ultrasound energy cannot go through the rib bones, and the acoustic impedance of the lung air totally mismatches the soft tissue impedances.

One main advantage of array transducers is the *electronic beam steering* ability, where beams are swept across the image field of view electronically with no mechanical motion of the transducer elements. Another advantage is the *electronic transmit focusing* ability, in

which the ultrasound beam can be electronically focused at any depth, specified by the user. The beam focusing is achieved by introducing time delays to the excitation pulses applied to the individual elements that constitute one beam, as shown in Figure 2.8. In this case, the wavefronts emerging from the different elements converge toward the focal position. Modern ultrasound systems offer a way to the user to choose the focal position. An extension to the transmit focus feature is the *multiple transmit focus* ability where more than one focused beams are used simultaneously to construct the final beam that has a larger focused area. However the frame rate is greatly decreased due to the delay necessary for transmitting multiple beams for each direction.

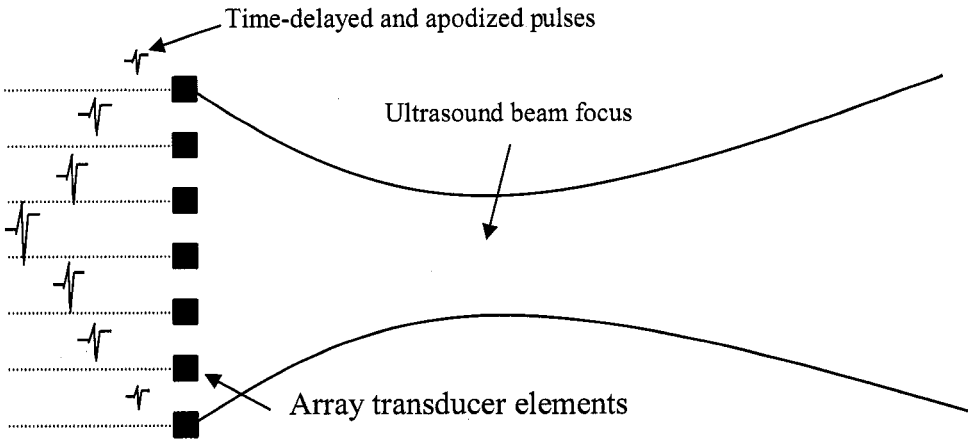


Figure 2.8 Array transducer electronic transmit focusing and apodization

The group of elements used for transmitting the ultrasound beam are also used to receive the returning echoes, which are amplified and combined together to produce a single signal for each reflector. The received signals are focused electronically via an operation called the *dynamic receive focus* process, which compensate for the differences between the arrival times of the signals of the same reflector to different elements. The dynamic receive focus process introduces delays in the received signal. These delays vary according to the depth of the reflector without any user control.

To minimize variations of beam width with depth, array transducers offer *dynamic aperture expansion*. The *aperture* refers to the size of the transducer surface that is used to produce and receive a single beam. By increasing the receiver aperture size with depth, the beam width is kept minimal across the whole image depth, which enhances the lateral resolution of the resulting image.

While that element apodization is used to minimize side lobe effect, array transducers apply a different apodization scheme, called *array apodization*. The amplitude of the signals applied to the transmit aperture elements decrease as we move to the aperture edge, as shown in Figure 2.8.

Recently multi-dimensional array transducers are offered where multiple rows of elements are used instead of a single row. 1.5D array transducers typically use 5-7 rows of elements that apply electronic focusing techniques to the elevation direction as well, which enhances the elevation resolution by reducing the slice thickness. True 2-D array transducers have just started to be commercially available in recent years, which offer real-time 3-D volumetric imaging.

2.4. Grayscale (B-mode) Ultrasound Imaging

Recalling the technique used by modern diagnostic ultrasound systems to construct B-mode ultrasound images, the technique is called *pulse-echo* ultrasound. In this technique, an ultrasound transducer transmits short-duration acoustic pulses in well-directed focused beams into the body. The pulse undergoes partial reflection at tissue interfaces. This results in echo signals that return to the transducer. The time delay of the echo is used to determine the distance of the corresponding tissue interface. The rate at which pulses are emitted is called the *pulse repetition frequency* or PRF [13].

The returning echoes go through amplification and compensation for attenuation losses. The amplification is necessary, since the echo amplitudes are typically too low to allow

visualization. Sound beam attenuation results in weakening the distal echoes more than the proximal ones to the transducer. A depth-dependent gain, called *time gain compensation* or TGC, is applied to the returning echoes to compensate for attenuation losses.

The *dynamic range* defines the amplitude range over which an electronic device responds effectively. The range of amplitudes of reflected echoes is typically higher than the dynamic range of the display. For that reason, ultrasound systems compress the dynamic range of the input signal to fit on the display. Usually a logarithmic compression technique is applied to boost the low amplitude signals, which represent scattered echoes from soft tissues that are very important to visualize and diagnose. Figure 2.9 shows an example of a B-mode ultrasound image of a 17 week fetus's face.

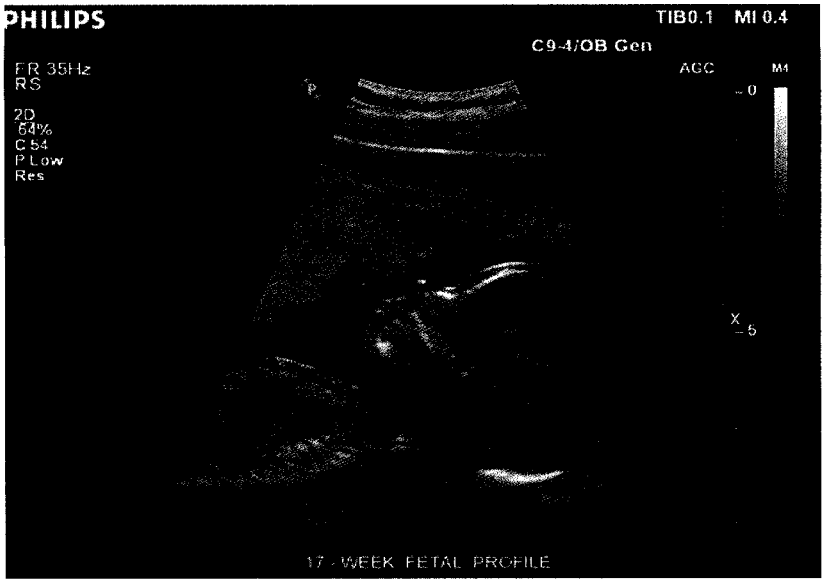


Figure 2.9 A B-mode ultrasound image of a fetus face.

2.5. Spectral Doppler Ultrasound Imaging

Doppler ultrasound is an important non-invasive technique for detecting and measuring the velocity of moving structures, particularly blood flow, within the human body. Doppler ultrasound was first introduced into medicine in the late fifties. The "*Doppler effect*"

was named after the Australian physicist Christian Andreas Doppler who discovered it in 1843 [12]. The Doppler effect states that when an observer is moving relative to a wave source (or vice versa), the measured frequency is different from the emitted frequency. The frequency difference is proportional to the velocity of the moving object and the sign of the difference depends on the direction of the movement relative to the stationary object.

2.5.1. Blood Flow

Before we review the ultrasound Doppler imaging techniques, we first give brief hints on the blood flow, which is the main application of the Doppler technique. The hemodynamics of blood flow in arteries are complex. However, useful approximations were made by several researchers, using fairly simple models, in attempts to understand the flow patterns and how they develop with disease state. For a complete review of blood flow models and characteristics, refer to the second chapter in the book written by Evans and McDicken [14]. The flow in the normal circulation is mostly laminar and pulsatile. In *laminar flow* the fluid particles move along smooth paths in layers with every layer sliding smoothly over its neighbor. Arterial blood flow is mainly *pulsatile* due to the heart beating mechanism. Venous blood flow is mainly *steady* especially for veins distal from the heart. Steady flow is that in which the velocity and pressure conditions are constant with respect to time.

The arterial pulsatile flow can be approximated as the sum of a steady flow component and a series of oscillatory components. The steady flow component results in a parabolic flow profile across the transverse section of the blood vessel, while the pulsatile components cause oscillating profiles which vary between parabolic and blunt. At each branch and bifurcation the velocity profiles get disturbed and require a finite distance to re-establish a steady flow. In general, blood flow profiles can be approximated as parabolic flows in main veins. In large vessels, such as the aorta, the flow may take on a more blunt profile. If there is a blockage or narrowing, such as a carotid stenosis, the flow becomes turbulent. Figure 2.10 shows the different approximated blood flow patterns involved in human blood vessels.

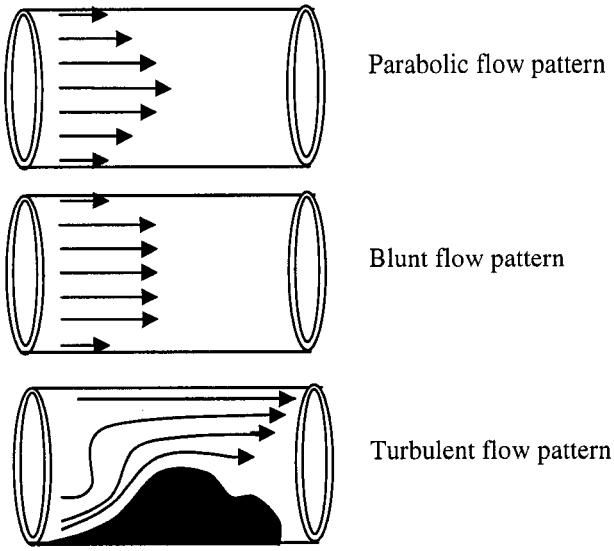


Figure 2.10 Blood flow patterns for typical, large and obstructed arteries veins.

2.5.2. The Doppler Equation

In Doppler ultrasound an ultrasound signal is transmitted into the body and the reflected echoes from the moving blood flow are collected and analyzed to detect the frequency shift and hence the velocity of the blood flow. The magnitude of the frequency shift is proportional to the velocity of the moving blood cells, while the sign of the shift defines the direction of the blood relative to the transmitted ultrasound beam. The Doppler equation that relates the frequency shift to the blood flow velocity is given as [14]:

$$\Delta f = \frac{2 f_o v \cos\theta}{c} \tag{4}$$

where Δf is the Doppler frequency shift, f_o is the transmitted ultrasound frequency, v is the moving blood cells velocity, c is the speed of sound, and θ is the angle between the transmitted beam and the blood flow path, which is called the *Doppler angle*. Figure 2.11

shows the Doppler effect with ultrasound while interrogating a blood vessel. Both the transmitted frequency and the shifted received frequency are shown as well as the Doppler angle between the ultrasound beam and the flow direction.

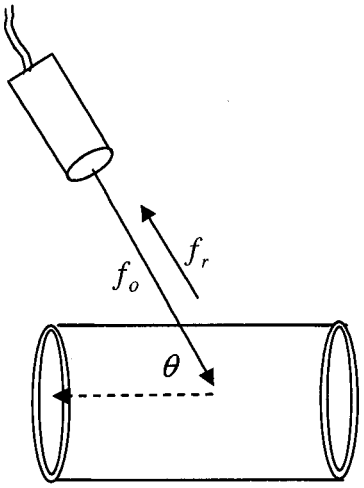


Figure 2.11 The Doppler effect with ultrasound.

It is clear from the Doppler equation that the Doppler frequency is directly proportional to the blood flow velocity, but it also depends on the transmitted ultrasound frequency. Another important factor that affects the velocity measurement is the Doppler angle. The Doppler frequency is proportional to the cosine of the Doppler angle. If the angle is less than 90°, the cosine term is positive and the blood flow is moving toward the transducer; if the angle is more than 90°, the cosine term is negative and the blood flow is moving away from the transducer. Thus the ultrasound system is capable of measuring the directional velocity of the blood flow with respect to the transducer direction. One other aspect of the Doppler angle is that the sensitivity of the velocity estimation degrades as the angle approaches 90°, as the cosine term approaches zero in this case.

Modern ultrasound systems offer a graphical control for the operator to specify the Doppler angle by aligning a graphical line with the vessel orientation in order to estimate the true velocity of the blood flow. If the operator makes a slight mistake in specifying the flow angle, the velocity estimation will be erroneous. The degree of error increases with increasing

the actual Doppler angle due to the effect of the cosine function. For example a 5° error with a 0° Doppler angle causes a velocity estimation error of 0.4%, while a 5° error with a 60° Doppler angle causes a velocity estimation error of 18.3%.

Another source of error with the velocity estimation is a phenomenon called *spectral broadening*. Spectral analysis is typically used in ultrasound systems to visualize the range of Doppler frequencies present in a Doppler signal. But even with a fixed speed blood flow, the returning Doppler signal contains a band of Doppler frequency shifts. This is due to the size of the ultrasound transducer, which causes the blood reflectors to see many Doppler angles instead of just one, as shown in Figure 2.12. The different Doppler angles create different frequency shifts, and hence different velocity estimations. Spectral broadening increases with increasing the Doppler angle.

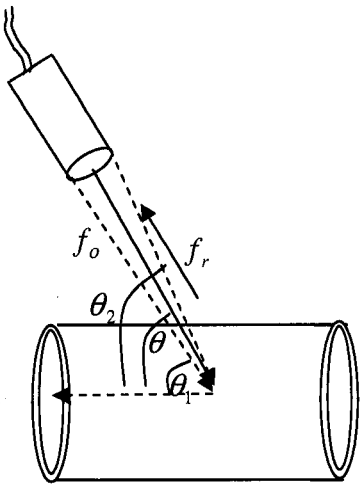


Figure 2.12 The Doppler spectral broadening phenomenon.

2.5.3. Spectral Analysis of the Doppler Data

Spectral analysis is the process by which a complex signal is broken down into simple frequency components. The most common way to perform spectral analysis of a signal is a process called Fourier analysis, or more specifically the fast Fourier transform (FFT) process. The FFT analyzer operates on small segments of the Doppler signal (1-5 ms) and produces a

histogram showing the relative amount of signal within each of several discrete frequency bins. It then operates on another segment to produce a continuous display, called the *scrolling Doppler spectrum display*. The scrolling Doppler display shows the distribution of Doppler frequencies, and hence the blood flow velocities, at each instant of time as a scrolling display. The brightness of a certain velocity is proportional to the number of scatterers moving with this velocity, as shown in Figure 2.13.

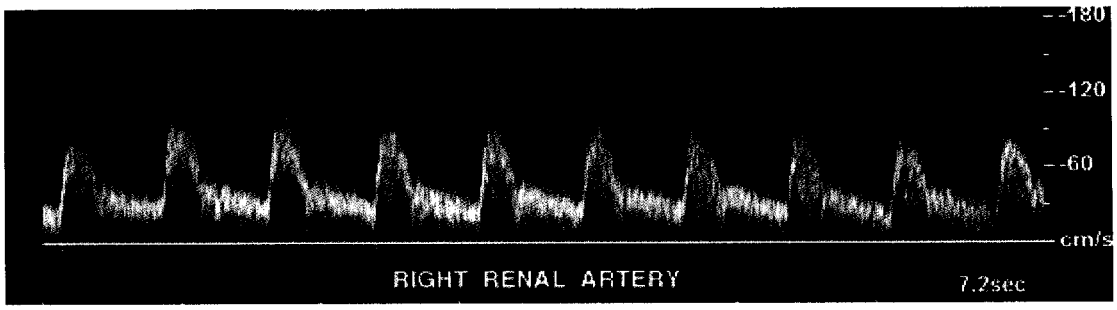


Figure 2.13 The scrolling Doppler spectrum display.

Doppler spectrum patterns from major vessels have been established. Deviations from normal patterns indicate pathological cases. Various parameters have been derived from the Doppler spectrum of a complete heart cycle, including systole and diastole phases, to quantify important clinical properties of the flow, such as peak systolic velocity (PSV), minimum diastolic velocity (MDV), mean velocity (MV), pulsatility index (PI), and resistivity index (RI). *Pulsatility index* (PI) measures the relative resistance to the flow of the vascular system, and it is defined as:

$$PI = \frac{PSV - MDV}{MV} \qquad (5)$$

Resistivity index also measures the flow resistance but without the need of calculating the mean velocity across a complete heart cycle. It is defined as:

$$RI = \frac{PSV - MDV}{MDV} \qquad (6)$$

2.5.4. Spectral Doppler Modes

There are two main types of spectral Doppler imaging modes usually offered by the ultrasound system, continuous-wave (CW) Doppler and pulsed-wave (PW) Doppler [14].

In continuous-wave (CW) Doppler mode an ultrasound transducer element is continuously excited using a single frequency sinusoidal signal. Another element of the transducer is continuously receiving the returning echoes.

The echo signal is demodulated by multiplying it with a reference signal derived from the transmitter. The resulting demodulated signal contains a mixture of both the sum and the difference of the reference signal frequency and the return signal frequency. The difference frequency is the Doppler frequency shift that we are after. The mixture signal is passed by a low pass filter to filter out the higher sum frequencies. Further filtering is applied to the signal to remove the high amplitude, low frequency signals originating from the slowly moving reflectors, such as vessel walls. This filter is usually called a *wall filter* referring to the vessel wall signal.

To detect the direction of the blood flow, a signal processing technique called *quadrature demodulation* is generally used. In this technique, the received signal is branched into two separate demodulator circuits in which the signal is mixed with a reference signal derived from the transmitter. The two reference signals have a phase shift of 90°. The demodulated output signals will be different in phase. Their relative phase difference depends on whether the received signal frequency is greater or less than the transmitted frequency. Hence the phase difference depends on whether the blood flow is moving toward or away from the transducer. The quadrature demodulation technique will be described in some detail in the following section where it is also used for processing color Doppler imaging. Figure 2.14 shows an example of a CW Doppler imaging for a case of aortic insufficiency. The top part of the image shows the composite B-mode and color Doppler image of the heart, while

the bottom part shows the scrolling CW Doppler display. The velocity below the *Doppler baseline* (zero velocity line) represents a regurgitant flow due to aortic insufficiency.

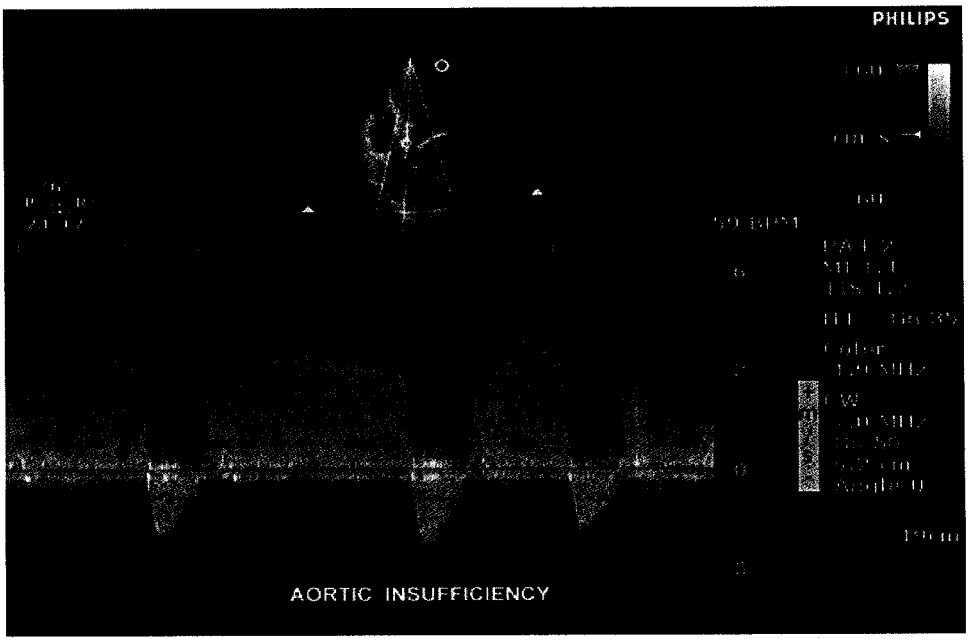


Figure 2.14 CW Doppler for Aortic insufficiency.

One main drawback of CW Doppler is that moving scatterers anywhere in the beam path will contribute to the Doppler signal. So if there is more than one vessel in the beam path, all vessels will contribute to the Doppler signal. The pulsed-wave (PW) Doppler technique resolved this problem by providing a way to select Doppler signals from specific depths. The region from which the signal is detected is called the Doppler *sample volume (SV)*. Ultrasound systems allow the precise selection of the depth and angle of the sample volume by offering the operator a graphical control to specify the position and size of the sample volume.

The PW Doppler technique was first introduced in the late sixties/early seventies by Wells [15] and Baker [16]. In PW Doppler mode sound pulses are produced by a transducer element at regular intervals. The returning echoes are demodulated the same way as CW Doppler. The signals from a desired depth, specified by the Doppler SV, are isolated. The echoes of all transmitted pulses are collected and analyzed. If reflectors within the range gate,

or the SV, are moving, their echoes will have slightly different phase for different transmitted pulses. A sample-and-hold unit retains the demodulator output between pulse-echo sequences. The filtered version of this retained signal is the Doppler signal.

Current ultrasound systems combine both pulse-echo grayscale mode and either CW or PW Doppler mode in a mode called *Duplex mode*. The grayscale anatomical image and the Doppler spectrum are displayed concurrently. The grayscale image is used to localize blood vessels and a graphical line, a *Doppler line*, is used to define the Doppler beam. In PW Doppler the SV graphics is plotted on the Doppler line, and the operator is able to move both the Doppler line and the SV to locate the site of interest within the vessel from which to acquire the Doppler signal from, as shown in Figure 2.15.

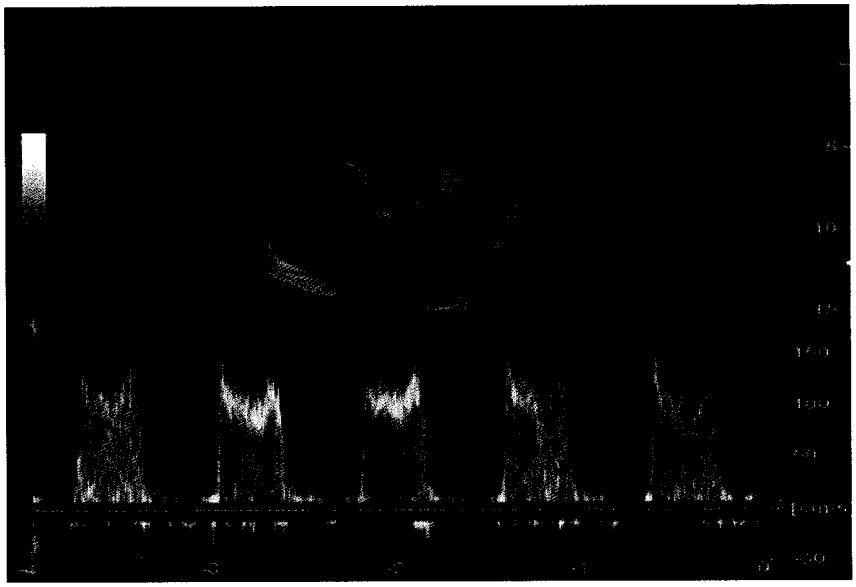


Figure 2.15 B-mode/PW Duplex mode.

2.5.5. Spectral Doppler Aliasing

With a PW Doppler instrument, the Doppler signal is sampled rather than recorded continuously as in CW Doppler mode. The sampling rate is called the *pulse repetition*

frequency or PRF. There is an upper limit on the PRF, which is controlled by the fact that the ultrasound pulse takes time to travel to the sample volume depth and the echoes take time to return back to the transducer face. The following pulse has to wait until all echoes from the previous pulse are collected. If the waiting time is insufficient, *range ambiguities* arise where the system will be uncertain of the actual depth that produced the returning echo. According to the sampling theorem, the sampling rate must be greater than or equal to at least twice the maximum frequency of the sampled signal to avoid aliasing. *Aliasing* is the production of lower-frequency components in the signal spectrum from those frequency components that exceed the Nyquist frequency, which is half the sampling frequency or PRF/2.

Aliasing is manifested on the Doppler spectrum display as the velocity wraps around, producing an apparent reversal of flow direction. So, very high velocity scatterers will appear as if they are moving in the reverse direction with a lower velocity rate. Figure 2.16 shows an example of PW Doppler aliasing where the high velocity components are rendered as low velocity ones.

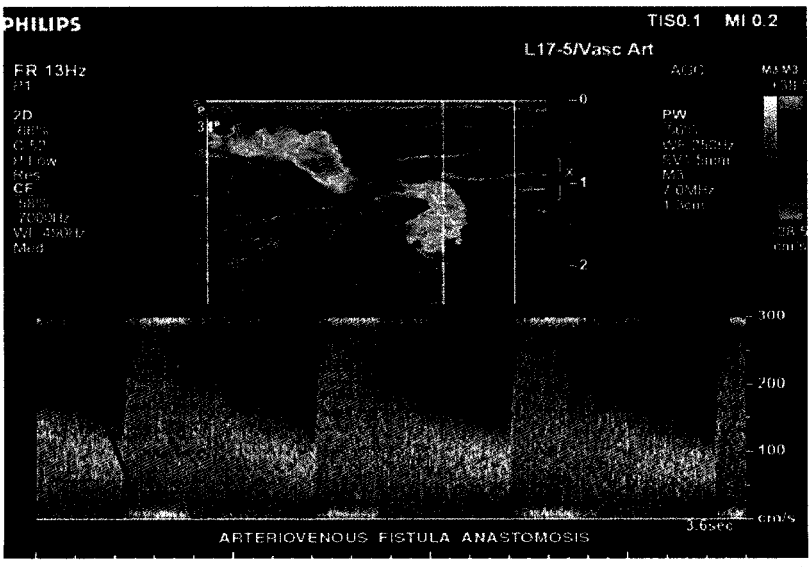


Figure 2.16 PW Doppler aliasing.

The ultrasound system usually offers a control for the operator to adjust the PRF to eliminate aliasing from the Doppler spectrum display. However the PRF can reach its

maximum without fully eliminating the aliasing effect due to pathological conditions that causes higher velocities, such as stenosis, or due to imaging a deeper vessel with high velocity profile.

The maximum detectable velocity with PW Doppler can be calculated as:

$$v_{\max} = \frac{c^2}{8 f_o d} \qquad (7)$$

where c is the speed of sound in soft tissue (1540 m/s), f_o is the transmitted ultrasound frequency, and d is the reflector depth or the sample volume depth.

2.6. Color Doppler Ultrasound Imaging

The spectral Doppler techniques presented in the previous section described the interrogation of a specific site within a blood vessel to track the velocity distribution with time. Color Doppler imaging techniques, on the other hand, offer a real-time two-dimensional cross-sectional image of the velocity distribution within the whole vessel or a number of vessels. A color Doppler image depicts in each pixel the mean Doppler shift frequency, and hence the mean velocity, detected in the direction along the ultrasound beam. This imaging technique was first introduced by Namekawa et al. in the early eighties [17], and it was first commercialized by Aloka Co. Ltd (Japan) in 1982 based on the signal processing developed and later reported by Kasai et al. in 1985 [18].

Combined with B-mode grayscale imaging, composite color/B-mode real-time imaging provides anatomical information along with physiological flow information, as shown in Figure 2.17. The composite B-mode/color Doppler real-time image can be combined with the scrolling spectral Doppler display to guide the placement of the PW sample volume across the stenosed areas of a vessel. In commercial ultrasound systems this mode is called

triplex mode, which is shown in Figure 2.18. The ultrasound system offers many controls to the user to optimize the color Doppler image: the color box size and position, the Doppler sample volume size and position, the Doppler steering angle, and the Doppler angle used for calculating the velocities using equation (4).

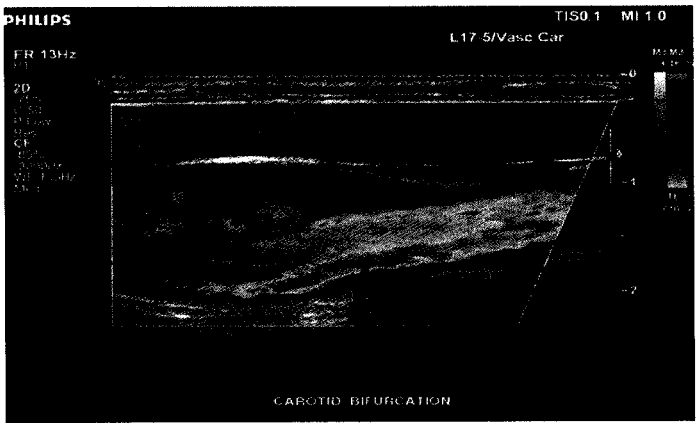


Figure 2.17 Real-time Color/B-mode imaging of a carotid bifurcation.

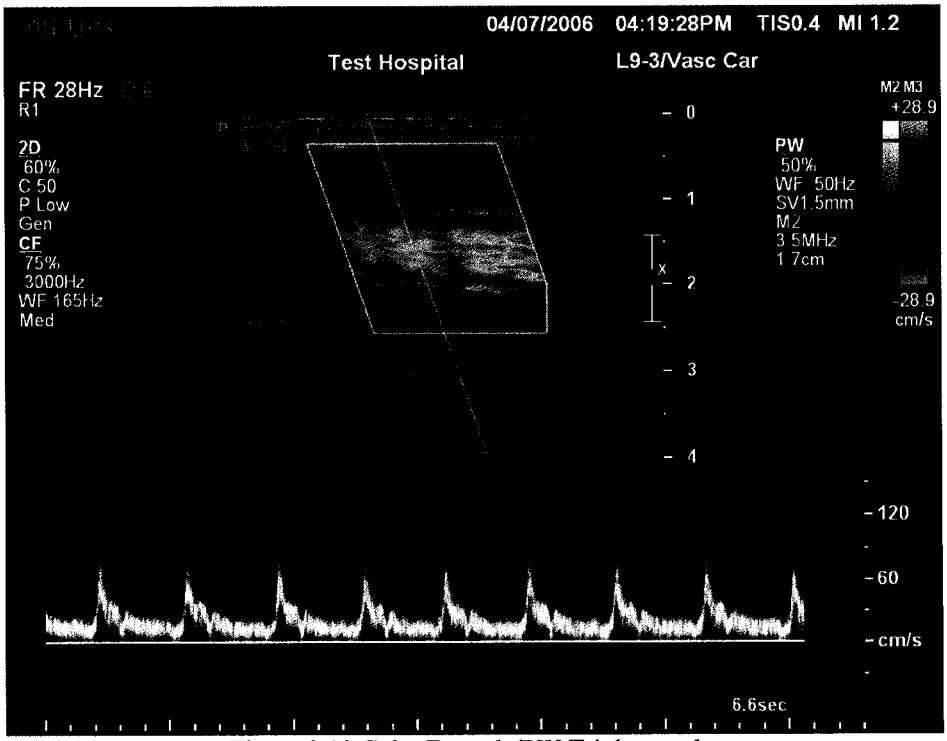


Figure 2.18 Color/B-mode/PW Triplex mode.

2.6.1. Color Doppler Imaging Signal Processing

In color Doppler imaging, ultrasound pulses are transmitted along beam lines covering the image field determined by the operator for color imaging. The color Doppler image field, which is called the *color box*, is usually smaller in size than the B-mode image due to the additional delay in color Doppler acquisition compared to B-mode imaging, as shown in Figure 2.17. The transmit pulses for color Doppler have longer duration than B-mode pulses and hence a narrower frequency bandwidth.

In color Doppler imaging, multiple pulse-echo sequences are used for each beam instead of just one pulse-echo sequence as in B-mode imaging. The sequence of pulse-echoes is called a *pulse packet*. Echoes returning from one sequence contain both stationary and moving reflector signals. The stationary reflector signals are mainly identical in all sequences. Only the moving reflector signals are different across different sequences and these signals are used to estimate the blood velocity. The more pulses used, or equivalently the longer the pulse packet, the better the estimation of the blood velocities, but at the same time, the slower the frame rate.

Figure 2.19 shows a block diagram of the main signal path blocks that are involved in the acquisition and display of color Doppler images. The beam former block is responsible for transmitting the ultrasound color Doppler pulses along narrow and well-directed beams, and receiving and amplifying the returning echoes that are modulated by the radio frequency (RF) signal.

The purpose of the demodulator is to remove the carrier radio frequency (RF) and filter the Doppler frequency shifts from the low frequency components caused by the stationary (or slowly moving) vessel walls. A secondary purpose of the demodulator unit is to detect the direction of the blood flow relative to the ultrasound beam.

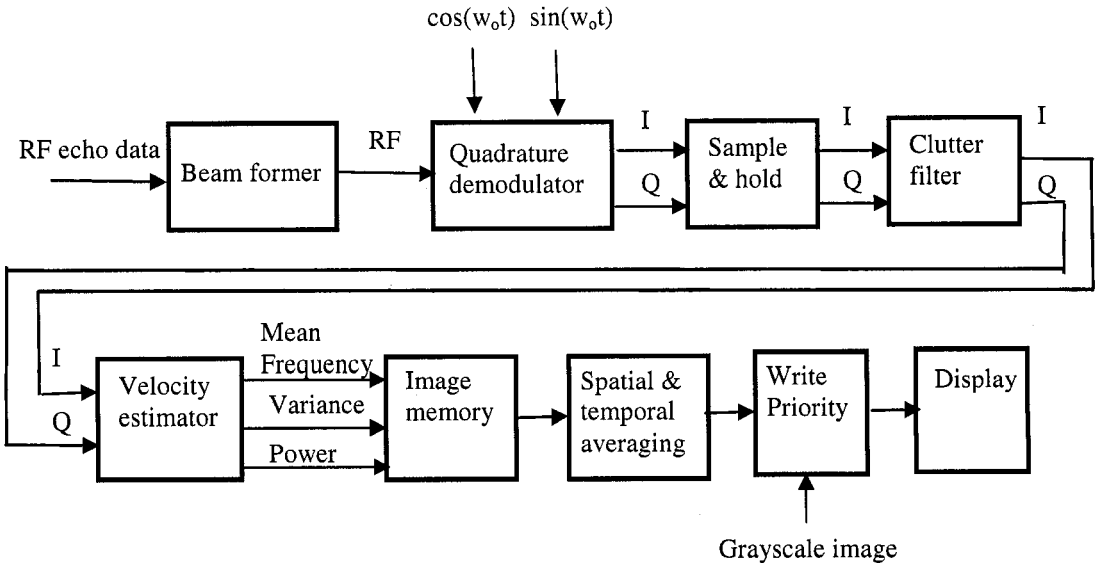


Figure 2.19 A block diagram of the signal path of the color Doppler imaging.

The most widely used demodulation technique for processing color Doppler imaging is the *quadrature demodulation* technique. In this technique the received RF Doppler signal can be simplified as a composite signal that contains three frequency components: the carrier frequency (RF) and the two signals resulting from motion towards and away from the transducer, as described by the following equation:

$$S(t) = A_o \cos(w_o t + \phi_o) + A_f \cos(w_o t + w_f t + \phi_f) + A_r \cos(w_o t + w_r t + \phi_r) \qquad (8)$$

where A, w, and ϕ refer to the amplitude, angular frequency, and phase of each signal. The subscripts o, f, and r denote carrier, forward, and reverse signals respectively. The signal is multiplied by the carrier frequency signal $\cos(w_o t)$ as well as a 90° phase shifted signal derived from the carrier one $\sin(w_o t)$. The output signals are then filtered to remove the DC component and the high frequency components resulting in the "in-phase I" and "quadrature Q" signal components, as follows:

$$I(t) = \frac{1}{2}\left[A_f \cos(w_f t + \phi_f) + A_r \cos(w_r t - \phi_r)\right] \quad (9)$$

$$Q(t) = \frac{1}{2}\left[A_f \cos(w_f t + \phi_f + \pi/2) + A_r \cos(w_r t - \phi_r - \pi/2)\right] \quad (10)$$

It is evident from these two equations that both the in-phase and quadrature signals contain only the forward or reverse Doppler signal. If the Doppler signal results from blood moving towards the transducer, i.e. only forward component, then the in-phase component will lag the quadrature one by 90°. If the Doppler signal was only reverse component, then the in-phase component will lead the quadrature component by 90°. This explains how color Doppler systems are able to detect blood flow direction and display it, as shown in Figure 2.20.

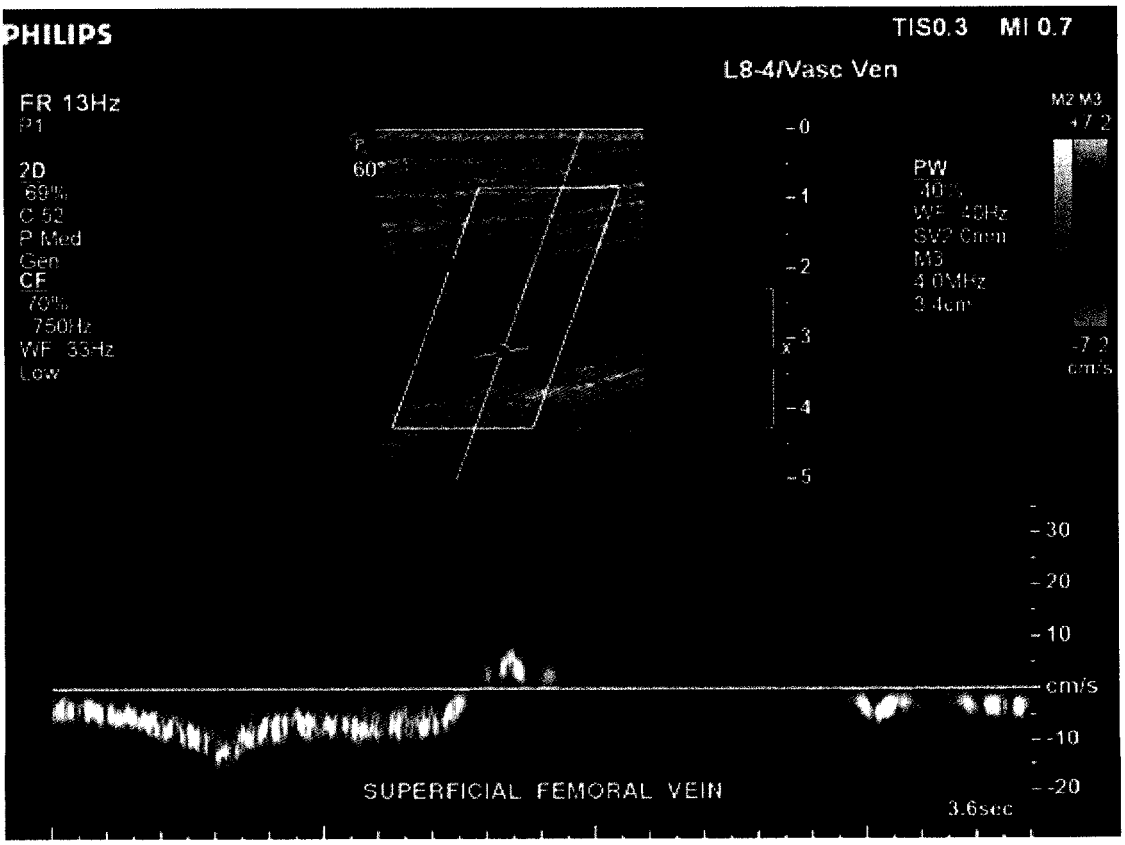


Figure 2.20 Flow direction detection. The superficial femoral artery flow is moving towards the transducer (red color), while the superficial femoral vein flow is moving away from it (blue color).

The sample-and-hold block is responsible for collecting all the signals of a single pulse packet for each position or sample volume along the ultrasound beam. Once the data of a single sample volume is complete, it is passed to the clutter filter module.

Echo cancellers, or *clutter filters*, are used to filter out the high-amplitude low-frequency signals from stationary, or nearly-stationary, objects. Unlike wall filters used in spectral Doppler analysis, clutter filters usually have a more complex design to deal with the high frame rate requirement for color Doppler image. The simplest form of clutter filters applies point-by-point subtraction between two echo sequences to eliminate the identical parts that correspond to stationary objects. Different clutter filter designs for color Doppler imaging, including FIR [19], IIR [20], and regression filters [21], have appeared in the literature.

Spectral analysis techniques are not suitable for color Doppler imaging due to the short time spurts over which echo signals are acquired for each position in the color image and also due to the computational complexity of the FFT process, which is not suitable for real-time imaging. Typically the mean velocity is estimated for each acquired position within the color Doppler image. There are many techniques in the literature used to derive the mean velocity from the Doppler sequences, the best known of which is the *phase shift autocorrelation* technique used in the very first color Doppler systems [17], [18]. The technique measures the change in the phase of the signals in each position from one transmit pulse to the next. The phase shift is correlated to the mean Doppler frequency shift, which is correlated to the mean velocity. The sign of the phase shift is related to the direction of the blood flow relative to the ultrasound beam.

To see how the mean frequency is estimated from the phase of the autocorrelation function, the mean frequency $\overline{w}$ of a Doppler power spectrum $P(w)$ can be defined as:

$$\overline{w} = \frac{\int_{-\infty}^{\infty} wP(w)dw}{\int_{-\infty}^{\infty} P(w)dw} \qquad (11)$$

The autocorrelation function $R(\tau)$ is defined as:

$$R(\tau) = \int_{-\infty}^{\infty} P(w)e^{jw\tau}dw \qquad (12)$$

The mean frequency can be written in terms of the autocorrelation function as:

$$\overline{w} = -j\frac{\dot{R}(0)}{R(0)} \qquad (13)$$

where $\dot{R}(0)$ is the first difference of the autocorrelation function at zero lag. If we represent the autocorrelation function by the magnitude and phase functions $\dot{R}(\tau) = |R(\tau)|e^{j\phi(\tau)}$, the mean frequency can be approximated as:

$$\overline{w} = \dot{\phi}(0) \approx \phi(T)/T \qquad (14)$$

where T is the time between subsequent ultrasound pulses. Thus the mean frequency is calculated from the phase of the autocorrelation function at lag $\tau = T$. The maximum likelihood estimate of the mean frequency from a number of samples in one pulse packet is given by:

$$\overline{w} = \frac{1}{T}\tan^{-1}\left\{\frac{\sum_{1}^{N} Q(i)I(i-1) - I(i)Q(i-1)}{\sum_{1}^{N} I(i)I(i-1) + Q(i)Q(i-1)}\right\} \qquad (15)$$

where N is the packet size and I_i and Q_i are the in phase and quadrature components, respectively, of the Doppler signal output from the clutter filter.

A very important conclusion from the previous derivation is that the color Doppler image is, in fact, a phase map of the autocorrelation function. This fact will be used in chapters 4 and 5 when we treat the color Doppler aliasing problem. The variance of frequency distribution and the signal power are also derived from the autocorrelation function of the Doppler signal and used for display in other variations of the color Doppler imaging. The variance is used to highlight areas of turbulent flow where there is a high variation in the velocity distribution, such as cardiac valve jets. The power is used in *power Doppler imaging* mode where it is useful to image slow velocities in small vessels, such as the kidney vessel tree. Figure 2.21 shows an example of a power Doppler image of the carotid bifurcation; the Doppler signal amplitude is visualized without the direction of the flow.

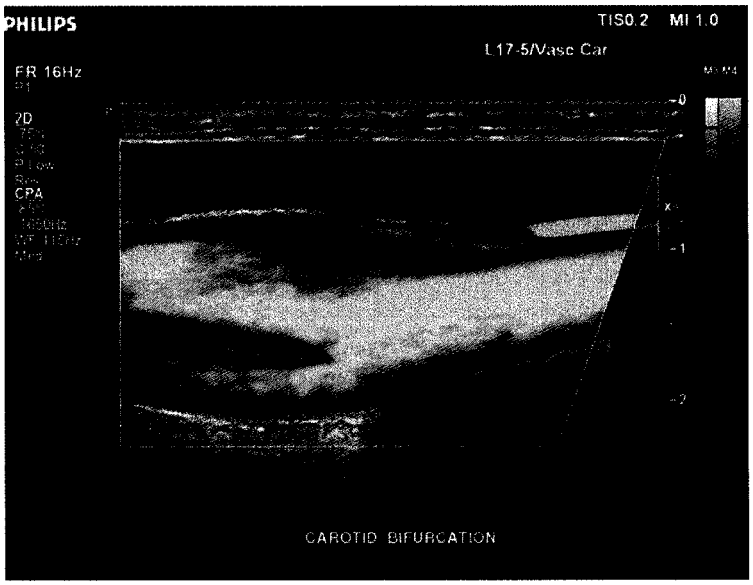


Figure 2.21 Power Doppler image of the carotid bifurcation.

Due to the limitations imposed by the frame rate and pulse packet size, the number of true color Doppler lines is usually low relative to the size of the color Doppler image. Spatial

interpolation techniques are applied to fill in between the true lines and create a pleasant looking image.

Color Doppler signal is stochastic in nature, which causes the estimated parameters to fluctuate randomly. For this reason, temporal filters are also applied to successive color Doppler frames to enhance the aesthetics of the image and minimize flashing [22]. The temporal filtering process is called the *persistence* process, and its degree is controlled by the operator.

The last processing block before display is the write priority process. This process chooses between the color Doppler velocity estimate and the underlying grayscale B-mode pixel. This is necessary to minimize color Doppler noise artifacts that do not represent true flow estimates. The write priority algorithms rely on one or more of several parameters including the Doppler signal power, the velocity magnitude, the Doppler variance, and the grayscale intensity [23].

In color Doppler imaging, a color map is used to display the mean velocity along with the direction of the blood flow. The hue of a colored image is associated with the wavelength of the light. The hue is the main attribute associated with the perceived color of a pixel. A red hue is typically used to represent flow moving toward the transducer, while a blue hue is used to represent flow moving away from the transducer. Saturation is a measure of color purity, i.e. the freedom from dilution with white light. Higher velocity pixels are shown with whiter colors in color Doppler imaging. Intensity represents the brightness level of the light, where maximum intensity is displayed as white, while minimum intensity is displayed as black. Intensity level can also be used to indicate the flow rate or velocity amplitude. Figure 2.22 shows a typical map used to display color Doppler images.

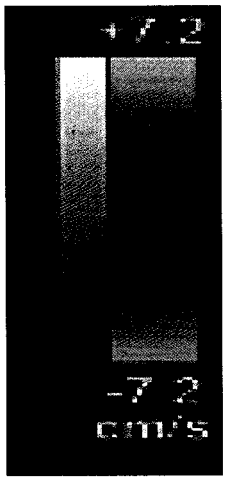


Figure 2.22 A typical color map for color Doppler image display.

Similar to the spectral PW Doppler imaging, color Doppler imaging also suffers from the aliasing problem. When aliasing is present, the color Doppler signal tends to "wrap-around", appearing as high velocities but in the opposite direction. The operator can overcome or reduce the aliasing problem by increasing the sampling rate, the color PRF. In many situations it may not be possible to completely remove the aliasing effect due to the high velocity within the vessel or the depth of the vessel. As shown previously, Figure 2.17 highlights a region of color aliasing where the higher velocities are rendered as reverse flow velocities.

2.7. Ultrasound Image Artifacts

In this section we will review a number of ultrasound image artifacts that will affect the algorithms developed in the rest of this thesis. Zagzebski [13] and Evans [14] describe many image artifacts in grayscale and color Doppler ultrasound images arising from well-known factors, such as attenuation, reflection, refraction, and beam width, related to the ultrasound physics and acquisition techniques.

Speckle noise is an artifact that manifests itself by showing organ interiors as a granular pattern rather than a homogeneous shade of gray. This is a result of scattered and reflected waves from tissue micro scatterers that interfere randomly, constructively or

destructively. This artifact will greatly affect the performance of any image processing technique that tries to detect vessels from ultrasound grayscale images, since the speckle noise introduces many false edges as shown in Figure 2.23. The edge detection image is obtained using the classical Canny edge detector [24].

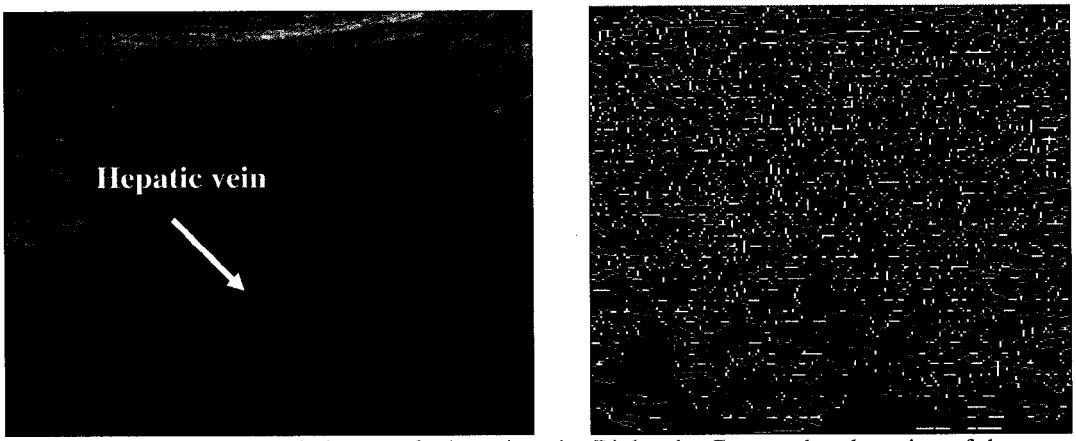


Figure 2.23 Left: a grayscale image of a hepatic vein. Right: the Canny edge detection of the grayscale image.

Insonation angle dependence is another artifact that manifests itself in ultrasound images as showing fuzzy and incomplete boundaries of the body organs. This is a result of the orientation dependence on the reflected echoes from specular reflectors. This artifact will affect the segmentation techniques used to identify organs in ultrasound images, since gradient information may be missing altogether in some parts of the image. Figure 2.23 shows how fuzzy the vessel boundaries can be in ultrasound grayscale images.

Currently ultrasound systems do not account for motion errors due to breathing and transducer/body organs motion. This may result in blurring in ultrasound images if the speed of motion is comparable to the acquisition rate. The effect is minimal, since the ultrasound acquisition rates are very high. However, any image processing technique that utilizes multiple frames should account for this vessel *motion artifact* by registering the consecutive frames before processing them.

Color aliasing, which was described in the previous section, corrupts the velocity values within the image due to the misinterpretation of the high velocity pixels. This artifact will affect algorithms that use color information for quantification purposes. For example different color regions may belong to the same vessel, while same color regions may belong to two different ones. The color aliasing problem will be treated in detail in Chapter 4 and Chapter 5.

As mentioned above, the velocity estimation depends on the angle between the ultrasound beam and the flow direction. If the vessel is long enough, then the estimated velocities show fluctuations and false reversal of direction in the ultrasound color Doppler image, while the true flow has uniform direction and velocity. This Doppler angle dependence artifact should also affect the segmentation techniques that rely on the color information. Figure 2.24 shows an example of the angle dependence artifact where the bottom vessel has one part in red and the other part in blue, indicating different flow directions, a false indication caused by this artifact.

Another color Doppler imaging artifact is called the *color bleeding* artifact in which two or more adjacent vessels appear as if they are connected. This typically happens when the ultrasound system fails to clearly identify the borders of the distinct vessels in the grayscale image due to weak returning echoes or non-perpendicular acquisition angles. In these cases the internal segmentation algorithm erroneously connects two or more distinct vessels with color Doppler data. Figure 2.24 shows an example of the color bleeding artifact which causes four distinct vessels to connect to each other in multiple points.

The *color flash* artifact is another motion artifact that manifests itself in color images as temporally random color blobs, since the ultrasound system erroneously interprets these motion vectors as moving reflectors. Figure 2.25 shows the flash artifact for an ultrasound image of the distal aorta. Any segmentation technique should try to distinguish true flow pixels from color flash pixels to successfully segment the vessels.

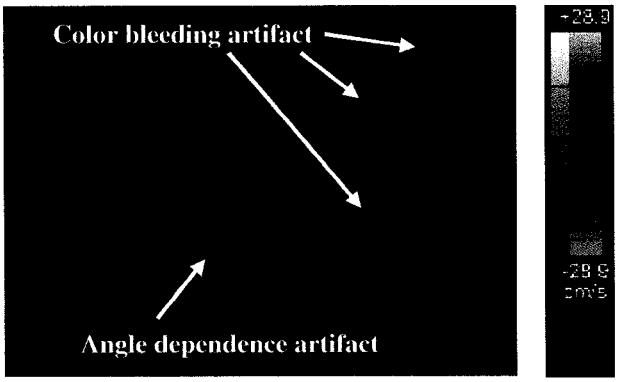


Figure 2.24 Doppler angle dependence artifact.

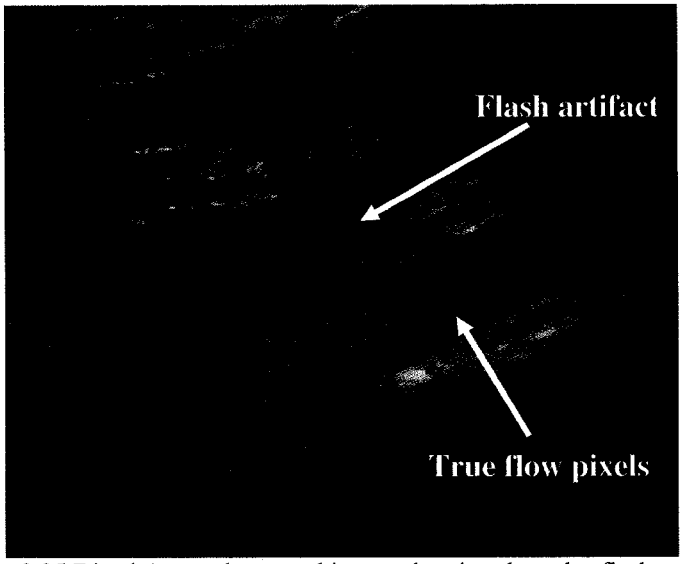


Figure 2.25 Distal Aorta ultrasound image showing the color flash artifact.

In summary, grayscale and color ultrasound images suffer from different artifacts, which impose unique challenges to the image processing tasks that attempt to analyze them. In the following chapter, we will develop a vessel segmentation technique to extract distinct vessels from color Doppler ultrasound images.

Chapter 3 Shape Decomposition Approach for Vessel Segmentation

3.1. Introduction

The first step of our vessel recognition system is to obtain a proper representation of the vessel that should be suitable for extracting discriminating features of that vessel. In this chapter we will discuss both the acquisition and vessel segmentation stages of the vessel recognition system as shown in Figure 3.1. According to the description of the ultrasound imaging fundamentals in the previous chapter, we have learned that grayscale ultrasound images carry information about vessel shape and geometry as blood vessels show up as hypoechoic longitudinal or circular structures, while color Doppler ultrasound images carry information about the blood hemodynamics as well as the vessel shape. The anatomical information provided by the grayscale images are used by the ultrasound system operators to guide finding the vessels of interest, however the blood hemodynamics, such as the peak velocity, the degree of pulsatility, and the type of phasicity, are the main source of diagnostic and discriminating information for the blood vessels. For that reason we have decided to rely on the color Doppler ultrasound imaging to recognize the underlying vessels. To retain the hemodynamic features of the vessel, a number of color Doppler frames that span at least one heart cycle should be considered; we name a set of consecutive color Doppler frames a *cineloop*.

Figure 3.1 The stages of the vessel recognition system. The acquisition and segmentation stages are the focus of the current chapter.

We use the Philips Ultrasound system iU22 [4] to capture ultrasound image cineloops. The iU22 system offers a research tool to transfer images from the device memory to an

external computer in XML-like format where the image pixels along with header information that contains image parameters are captured. The tool allows us to specify the number and type of frames to capture. We chose to capture 30 frames of both the grayscale and color Doppler images for each dataset. The format of the grayscale image is unsigned 8-bit pixels that represent grayscale values between 0 and 255, while the format of the color Doppler image is signed 8-bit pixels that form a color-coded representation of the directional mean velocity of each pixel.

The blood vessels are mostly self-segmented in color Doppler images since the other anatomical structures are relatively stationary and do not exhibit any Doppler echoes, which leaves the blood vessels alone in the color Doppler image. However there are many imaging artifacts, such as those described in section 2.7 including the color flash, bleeding, and aliasing artifacts, which will add difficulty to the segmentation and recognition tasks. Color bleeding artifact is the primary source of difficulty of the vessel segmentation task, which is the focus of this chapter. Color bleeding causes two or more distinct vessels to appear as if they are connected in certain points across the border of the vessels. This artifact happens only if the vessels are adjacent; however peripheral arteries and veins frequently run parallel to each other as the case with the carotid artery and jugular vein which run parallel to each other in the neck; the color bleeding artifact usually happens with these two vessels as shown in Figure 3.2.

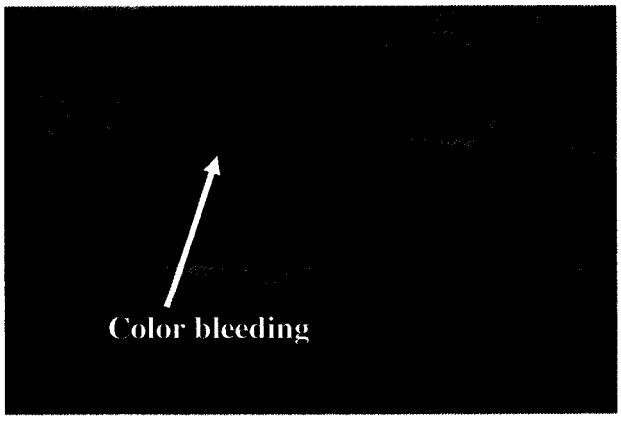


Figure 3.2 Color Doppler bleeding artifact with the jugular vein (top vessel) and the carotid artery (bottom vessel).

In this chapter we will address the vessel segmentation task from the color Doppler cineloops by accounting for the color bleeding artifact. In section 3.2 we will survey the literature of the vessel extraction and segmentation techniques in medical imaging. We found the majority of the techniques suitable for grayscale imaging, which is a common image type of different medical imaging modalities including X-ray, computed-tomography, and MRI. Since ultrasound grayscale imaging suffers from many unique artifacts, as described in section 2.7, we found the existing techniques unsuitable for the ultrasound vessel segmentation problem. A new segmentation technique based on well-known shape decomposition approaches was developed to work with color Doppler images and segment distinct vessels that are connected due to the color bleeding artifact; the details of the algorithm will be described in section 3.3. In section 3.4 we will discuss the first clinical application that could directly benefit from the vessel segmentation step, which is the Doppler angle automation application. Finally the conclusions and future work will be discussed in section 3.5.

3.2. Vessel Segmentation Literature Review

There are many vessel extraction and delineation techniques in the literature. A recent comprehensive survey of vessel extraction techniques was published by Kirbas in which several techniques are referenced and categorized [25]. Vessel segmentation is an important research problem in all medical image modalities including Magnetic Resonance Imaging (MRI), Magnetic Resonance Angiogram (MRA), X-Ray Angiography (XRA), Computed Tomography (CT), and Digital Subtracted Angiography (DSA). There are many applications for vessel segmentation, such as computer aided diagnosis, registration of patient images, quantification, and computer-based image retrieval systems.

In the following sub-sections, we present the most relevant vessel segmentation and recognition techniques categorized by the methodology behind the techniques.

3.2.1. Multi-Scale Approaches

In multi-scale approaches, the segmentation is done in different image resolutions. The main advantage of these techniques is increasing the robustness and processing speed. The multi-scale approach usually applies scale-space filtering to the original image where the image is convolved with smoothing kernels (usually Gaussian) of different sizes. The resulting images of different scales are processed concurrently followed by a fusion process to combine all the outcomes of different scales.

Armande et al. [26] employed a multi-scale approach to detect cerebral vessels network in Digital Subtraction Angiography (DSA) images. The centerlines of the vessels are modeled as thin nets, which are curvilinear structures with maximum gray levels in a given direction. The multi-scale approach helped to detect thin nets of different widths in the same image. They applied Differential Geometry concepts, such as curvatures of the image surface to characterize the thin nets. They had to prove the stability of the surface curvature in scale-space, which guarantees that moving from coarse to fine scale, new responses appear but existing ones never disappear. They also derived a relationship between the vessel width and the scale-space filter scale (the standard deviation in case of the Gaussian filters).

3.2.2. Skeleton-Based Approaches

The notion of skeleton was first introduced by Blum [27] as a region-based shape descriptor that summarizes the general form of objects. Many researchers have found that the skeleton of the blood vessels (or tubular anatomical objects in general) can summarize the relevant medical information needed for further quantification and registration techniques. Generally there are two approaches that use the skeleton of the blood vessels. One approach is to extract the boundary of the object by any segmentation technique and then use a thinning algorithm to produce an approximation to the skeleton in a topology-preserving manner. Most thinning algorithms are based on iteratively removing object border points that satisfy certain topological and geometrical constraints. The entire process is repeated until only the skeleton

approximation is left. The second approach is to detect the skeleton directly and then to locate the object boundary around the skeleton.

Sorantin et al. [28] applied a skeleton-based technique for the assessment of larengo-tracheal stenosis on 3-D spiral-CT images. First, they segmented the laryngo-tracheal tract (LTT) in 3-D datasets using fuzzy-connectedness concepts introduced by Udupa et al. [29] where a fuzzy object is defined to be a fuzzy connected component of pixels. Fuzzy connectedness is a fuzzy relation on the set of all pixels. It combines together the notion of fuzzy adjacency of pixels, which is independent of image information, and fuzzy affinity between pixels, which depends on image intensity values. Next, they computed the skeleton of the LTT using a thinning algorithm. Third, they extracted the medial axis, the centerline of the skeleton, from the skeleton using a pruning technique. Finally, they quantified the stenosis sites from the medial axis.

3.2.3. Ridge-Based Approaches

Ridge-based methods treat grayscale images as 3-D elevation maps by mapping the intensity into height. Ridge points are local peaks in the direction of maximal surface gradient, and can be obtained by tracing the intensity map from an arbitrary point, along the steepest ascent direction. The main advantage of ridges is that they are invariant to affine transformations. The ridges of the elevation maps approximate the skeleton of the tubular objects given that objects have nearly circular cross sections which have intensity extrema near their centers. This is a common feature for blood vessels in medical images as in CT and MRI imaging.

Aylward et al. [30] segmented tubular objects in medical images by extracting the ridges of the image surface. The Hessian matrix of the height function is used to define the ridge points as height extrema. Their technique starts with a seed point on the image near the vessel; the nearest ridge point is found using conjugate directions search with respect to the Hessian matrix. Once the ridge point is found, they apply a ridge traversing technique to find

all ridge points. Then the width of the vessel is found at every ridge point to complete the representation of the vessel. The main drawback of this technique is the fair amount of user interaction to specify seed points for the ridge traversing algorithm.

3.2.4. Region-Growing Approaches

Region-growing techniques typically start from a seed point then segment images by incrementally recruiting pixels to a region based on some predefined criteria. Two important segmentation criteria are value similarity and spatial proximity. The assumption is that pixels that are close to each other and have similar intensity values are likely to belong to the same object. The main disadvantage of region-growing techniques is that they often require user-supplied seed points. Due to the variations in image intensities and noise, region-growing can also result in holes and over segmentation. Thus, it requires post-processing of the segmentation result.

Schmitt et al. [31] determine contrast agent propagation in 3-D rotational X-ray vessel tree images. They combine thresholding with a region-growing technique to segment vessel tree in 3-D. The optimal threshold is determined experimentally. The region-growing algorithm is applied to the segmented 3-D binary vessel tree in order to cluster the tree voxels into segments of separate vessel branches, each of which will be assigned the same contrast agent propagation arrival time value for further analysis. Their technique is based on spatial proximity only, since the input data is a binary image. Starting with a seed point on the main vessel root, the algorithm adds generations of neighboring pixels to the current connected component. When a vessel bifurcation happens the algorithm splits into two new connected components. Eventually the vessel tree is segmented into separate branches.

3.2.5. Differential Geometry-Based Approaches

Differential geometry (DG)-based methods handle images as hyper-surfaces and extract features using the curvature and the crest lines of the surface. The crest points of the hyper-surface correspond to the center lines of the vessel structure. The 2-D and 3-D images are treated similarly, being modeled as 3-D and 4-D hyper-surfaces respectively. In DG, a 3-D surface is described by two principal curvatures and their corresponding orthogonal directions, called principal directions. These features are invariant under affine transformations and therefore well-suited to medical image registration.

The previously cited reference Armande et al. [26], who employed a multi-scale approach to detect cerebral vessels network in Digital Subtraction Angiography (DSA) images, used the differential properties of the image surface to characterize the thin vessel centerlines as the zero-crossings of the directional Derivative of the Maximum principal Curvature.

3.2.6. Matching Filters Approaches

Matching filters approaches convolve the image with multiple matched filters to extract objects of interest. In extracting vessel contours, filters with different orientations and sizes are designed to detect the vessels. However the convolution kernel size affects the computational load. Matching filters are usually followed with some other image processing operations, such as thresholding and connected component analysis to extract the final vessel contours.

Poli et al. [32] developed a real-time algorithm to enhance and segment blood vessels in coronary angiogram images based on a set of linear filters sensitive to the vessel's orientation and thickness. The filters are obtained as linear combinations of properly shifted

separable Gaussian filters. Vessel segmentation is performed via thresholding with hysteresis of the filtered image.

3.2.7. Mathematical Morphology Approaches

Morphology is the study of object forms or shapes. Morphological operators apply structuring elements to either binary or gray-scale images. Dilation and erosion are the two main operators. Dilation expands objects by a structuring element to fill holes or connect disjoint regions. Erosion shrinks objects by a structuring element to disconnect linked objects or remove noise ones. Closing (dilation followed by erosion) and opening (erosion followed by dilation) are two popular compound operations.

Eiho et al. [33] applied morphological operations to segment the coronary artery tree from the x-ray angiogram images. They first applied a gray-scale Top-Hat operator to enhance the image contrast, which is done by subtracting the image processed by the closing operator from the original image. Then they eroded the enhanced image to enlarge the vessel width (which has darker gray levels than the background) and decrease the noise. Finally they segmented the vessel using an adaptive thresholding technique and used the binary thinning operator to extract the vessel tree centerlines.

3.2.8. Model-Based Approaches

In model-based approaches, explicit models for vessels, which incorporate knowledge about the geometry, topology and morphology of the vessels, are developed. There are two main approaches to deformable models (DMs), parametric (or explicit) DMs and geometric (or implicit) ones. In addition, there are a number of parametric models including the generalized cylinders models.

Parametric DMs use parametric curves or surfaces that deform under the influence of internal and external forces and are attracted to the object boundary. These methods were originally introduced by Kass et al. [34]. The DM is parameterized by a contour with a number of control points that are connected together and each point has an associated energy that contributes to the total contour energy. The forces that act on the contour are either internal forces that maintain smoothness and connectivity degrees on the contour or external forces extracted from the image to attract the contour toward the object of interest. The contour evolves under these forces toward a resting position with a minimum total energy. The research community has been very active in enhancing the classical DMs by solving problems, such as parameter selection, initialization, reaching global minima, increasing the influence range of the external forces, preventing the internal forces from shrinking the contour, and accounting for topological changes, such as branching vessel trees [35].

Luo et al. [36] introduced a new DM that overcomes the problems associated with traditional snakes, such as contour initialization, internal parameter setting, and the limitations in the capture range of the external energy. The model internal energy maintains smoothness without any shrinking side effects on the contour, while the external energy combines both edge and region information. They tested the new DM on digital subtract angiography images for the arterial tree.

Geometric DMs use propagating interfaces under the curvature-dependent speed function that stop at object boundaries. These methods were introduced by Casselles et al. [37] and Malladi et al. [38]. The main advantages of the geometric DMs are the ease of handling sharp corners and topological changes, and natural extensions to higher dimensions. Malladi et al. [38] have shown results for segmenting the arterial tree from digital subtract angiogram images.

Parametric models define objects of interest parametrically. For tubular objects, the models used are either overlapping ellipsoids or generalized cylinders that consist of a space axis and a cross section function defined on that axis. The parameters of the model are

estimated from the image. More sophisticated models may be developed to approximate pathological irregular shapes and bifurcations.

O'Donnell et al. [39] introduced the extruded generalized cylinder (EGC) as an enhancement of the classical generalized cylinder (GC) models. The EGC model allows a non-planar spine and overcomes the problems caused by inflection points and spine torsion. They applied the ECG model to segment carotid artery images.

3.2.9. Tracking-Based Approaches

In tracking-based vessel segmentation techniques, a seed point on the vessel is specified after which local operators get applied to track the vessel. Some of these methods apply edge detection operators followed by sequential tracing, incorporating connectivity information about the vessel boundary. Other approaches are fuzzy clustering and graph representation. The ridge tracking technique developed by Aylward et al. [30] and presented in a previous section is an example of tracking methods.

3.2.10. Artificial Intelligent (AI) Approaches

AI techniques employ high-level knowledge about the vessel information to guide the segmentation. Sources of knowledge include the imaging acquisition technique and the anatomy or pathology features of the vessels.

Goldbaum et al. [40] described a system for automated diagnosis and image understanding of retinal images. They organized the objects of interest into five classes: curvilinear objects (blood vessels), bright blobs, darker blobs, the optic nerve, and the fovea. They used template matching techniques to segment each category and then extracted a set of features for each object, such as intensity mean, intensity standard deviation, and geometrical factors. The feature vectors were input to different classifiers to filter the false objects. For the

diagnosis part, they developed a Bayesian probabilistic expert system to classify the input images into one of 43 vascular diseases of the retina.

3.2.11. Summary of the Vessel Segmentation Literature Review

Most of the techniques reviewed above have been developed for MRI, CT, and X-ray images, which tend to have better image quality than ultrasound images. In addition, most of the techniques assume the presence of tubular elongated objects or thin vasculature tree. While this may be true for MRI and CT images, it does not always hold for ultrasound images, since the ultrasound scan plane constantly varies throughout an exam and can therefore intersect a vessel at any arbitrary plane. We can think of the ultrasound vascular image as a zoomed image over an arbitrary site and orientation of the vessel, which make it hard to assume any geometrical models of the vessel.

Many of the previous techniques need user interactions to either provide a seed point or to correct for the segmentation. Ultrasound segmentation techniques for automation purposes will need to be fully automated, since the user interaction would violate the main purpose of the automation. Computational speed of the algorithm is also an important issue for ultrasound applications, since ultrasound is a real-time modality. For automation purposes, the techniques need to be fast to save time for the sonographers.

While there are a few ultrasound-specific vessel segmentation techniques, all of them deal with grayscale images, not color Doppler ones, and mainly intravascular ultrasound images (IVUS), which have better quality than standard ultrasound images [41]. In the following section, we present a novel technique that we developed to segment vessels from ultrasound color Doppler images.

3.3. Shape Decomposition Approach for Color Doppler Image Segmentation

Due to the artifacts present in grayscale ultrasound images such as incomplete boundaries and speckle noise, we decided not to rely on the grayscale images to guide the vessel segmentation. Instead we rely solely on the color Doppler image to detect the distinct vessels and resolve the color bleeding artifact, which is the main source of problems with the vessel segmentation from color Doppler images. The following sections describe the segmentation approach in detail.

3.3.1. Image Preprocessing

A necessary preprocessing step is to obtain a representative image for a vessel from a cineloop that contains a number of frames that encompass both diastolic and systolic phases of the blood flow. We achieve this task by averaging all the frames in the captured cineloop to eliminate the effect of the flow pulsatility, followed by a threshold step to obtain a binary image of the color Doppler image. The threshold value chosen matches the threshold used by the ultrasound system to display the color pixels after eliminating the low velocity noise pixels. Then a hole-filling algorithm is applied to remove the gaps inside the object.

3.3.2. Vessel Shape Decomposition Technique

In case of imaging several adjacent vessels, the simple thresholding step described above results in a complex shaped binary object that consists of several vessels linked together due to the color bleeding artifact described in section 2.7. Figure 3.3 shows an example image of four adjacent vessels and its binarization. We will use this image throughout the description of our algorithm. The goal of our shape decomposition technique is to partition this complex object into distinct vessels.

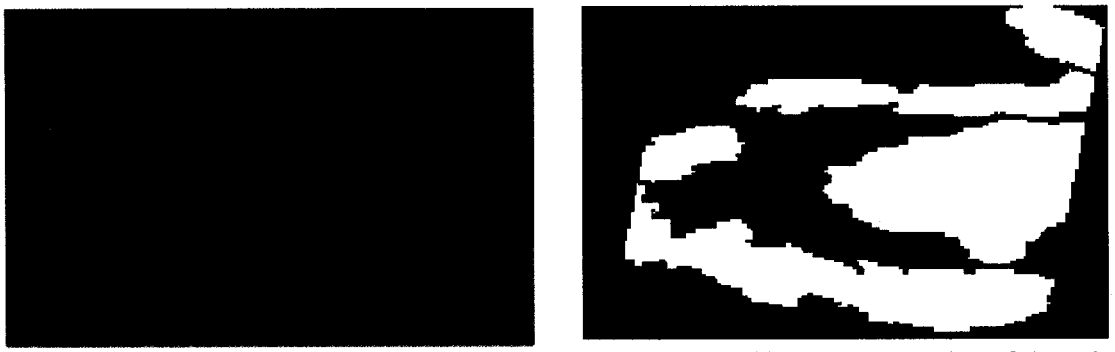


Figure 3.3 Left: Example image of four adjacent vessels. Right: A binary representation of the color Doppler image on the left. The red lines represent the correct partitioning.

In general, shape is an important visual feature that is used for object recognition, image database retrieval, image matching, and image analysis. There are many shape description and representation techniques in the literature [42]. Structural shape description is one of the major shape description methods, where a shape is described in terms of simpler shape parts and the relationships between these parts. Shape decomposition techniques can be classified into region-based versus boundary-based partitioning. Examples of region-based decomposition are overlapping disks, maximally convex parts, generalized cylinders, and superquadrics. Examples of boundary-based decomposition are high-curvature points, constant-curvature segments, and polygonal approximations. Many theories also exist in the literature to describe the correct and intuitive partitioning scheme for shapes based on psychophysical and ecological evidence. Siddiqi and Kimia [43] reviewed most of these techniques and theories while presenting a partitioning scheme based on two types of parts, *neck-based* parts and *limb-based* parts. Neck-based parts arise from narrowing in shape, while limb-based parts arise from a pair of negative curvature minima with co-circular tangents. Figure 3.4 shows examples of both partitioning schemes.

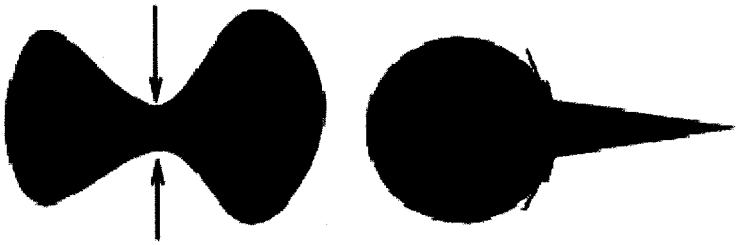


Figure 3.4 Left: An example of neck-based partitioning. Right: An example of limb-based partitioning.

Our technique shares some concepts with the partitioning scheme described in [43]. We also use the notion of a "part-line," which is defined as a line whose end points lie on the object boundary and is entirely embedded in the object, as a separator of parts. The technique is based on several observations we made after studying a large number of color Doppler images with adjacent vessels and observing a number of diagnostic ultrasound exams:

1. The correct part-lines that result in separating the linked vessels involve one or two negative curvature minima, as shown in Figure 3.3. We call the part-line that involves two curvature minima a *double-point part-line* and the part-line that involves a single curvature minimum, a *single-point part-line*.

2. Some negative curvature minima do not belong to any correct part-line due to the irregular and noisy nature of the input object boundary; the technique to be developed must not be misled by these artificial minima.

3. Distinct vessel segments are mainly convex and elongated, since longitudinal views are the most useful diagnostic views in ultrasound imaging.

4. When two adjacent vessels, which mostly run parallel to each other, are linked, the eccentricity of the resulting linked object will be less than that of at least one of the two original vessel objects.

Based on the previous observations, we describe our shape decomposition technique as follows.

3.3.2.1 Negative-Curvature Minima Detection

Based on the first observation above, we need to detect all negative-curvature minima. There are many algorithms in the literature to detect curvature maxima and minima. Li et al. [44] constructed the corner detection problem as a fuzzy classification problem based on

human perception of local graphic features, such as the curvature and the support region on both sides of the corner point. The advantages of their technique are robustness against noise and adaptability to corners of multiple scale features. Arrebola et al. [45] detected corner points via developing a curvature estimator that is robust to noise. They used the histogram of the chain code of two segments before and after each curve point as an estimator of the curve orientation. By calculating the correlation coefficient between these two histograms, the high curvature points are characterized by low correlation coefficients. A number of researchers, such as Hua et al. [46] and Quddus et al. [47] have used the wavelet transform to detect corners in multi scales where the discrete wavelet transform is applied to either the curve orientation profile or the curve parametric representation, and the modulus maxima is calculated for each wavelet level. The peaks of the modulus maxima at the higher scale suggest true corners that are tracked in lower scales to find their exact locations.

We chose to implement the algorithm described by Han in [48] which seemed suitable for the irregular and noisy nature of our contours and for its simplicity and fast execution time. The algorithm is described as follows. Let p_i be a point on the discrete curve. For a given value L, we draw a line segment connecting the two points p_i and p_{i+L}, as shown in Figure 3.5. We then calculate the signed distance from the line connecting p_i and p_{i+L} to points p_j, where $i < j < i+L$. If p_j lies on the left side of the vector $V_{p_i p_{i+L}}$, then its distance is positive, otherwise the distance is negative. The signed distance is calculated as follows:

$$d_j = \frac{(x_{i+L} - x_i) * (y_i - y_j) - (x_i - x_j) * (y_{i+L} - y_i)}{\sqrt{(x_{i+L} - x_i)^2 + (y_{i+L} - y_i)^2}} \qquad (16)$$

where d_j is the signed distance of point $p_j = (x_j, y_j)$ to the line connecting points $p_i = (x_i, y_i)$ and $p_{i+L} = (x_{i+L}, y_{i+L})$.

The points with the positive and negative maximum distances are found, as shown in Figure 3.5 as points p^+ and p^- respectively. An accumulator, $h(p)$, for all curve points (initialized with zeros) is updated for the positive and negative maximum distance points as $h(p^+) = h(p^+) + 1$ and $h(p^-) = h(p^-) + 1$. Then the line segment is shifted by one point on the curve to connect the two points p_{i+1} and p_{i+L+1} and the whole process is repeated again. After consuming all curve points, the points whose accumulators are local maxima are marked as convex points, while those whose accumulators are local minima are marked as concave points.

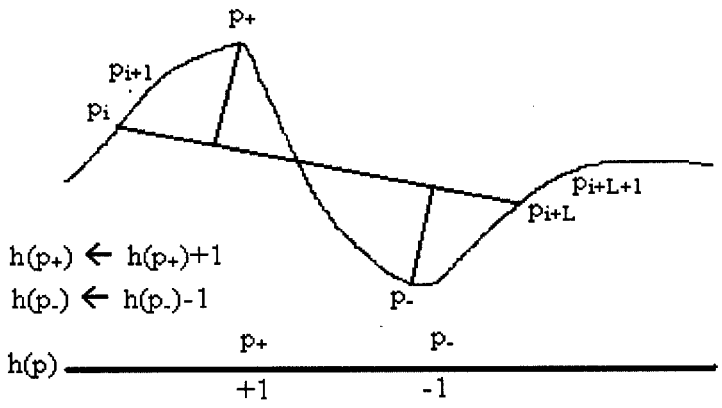


Figure 3.5 The concave points detection algorithm.

In our case, the parameters of the algorithm were set once empirically and then used for all test images. The main criterion for the purpose of segmentation is to find all true concave points, since missing a concave point would cause the decomposition to fail if this point contributes to the partitioning. While the algorithm allows for false concave points to be detected, they will be filtered out in the following steps of the algorithm. Figure 3.7a shows the detected concave points of our example image using the described algorithm.

3.3.2.2 Hierarchical Shape Decomposition Technique Using Part-Lines

After finding the concave points of the complex object of connected vessels, we start a hierarchical shape decomposition process by considering the partition lines associated with these points. The pseudo-code of the algorithm is shown in Figure 3.6, which starts with the complex binary object as an input and outputs the partitioned simple objects that represent distinct vessels. We will use our example image shown in Figure 3.3 to describe the algorithm steps in detail.

In step 3 of the algorithm, we detect all concave points as described in the previous section. Figure 3.7a shows the detected concave points of our example image of four connected vessels. In step 4, we detect all *double-point part-lines*; a double-point part-line connects a pair of concave points and lies completely inside the object. Figure 3.7b shows all double-point part-lines of the example image. In step 5, we filter the double-point part-lines list by retaining every part-line with an enclosing circle (with the part-line as its diameter) that mostly lies inside the object. Ideally this would require comparing the foreground pixels with all pixels within the circle, but we chose to implement a faster approximation by comparing the foreground perimeter pixels to all perimeter pixels; the results were still satisfactory. The threshold used for the enclosing circle criterion is set adaptively based on the original object eccentricity. This is due to the fact that as the original object gets longer, the partition line gets longer and the enclosing circle has less intersection with the object. Figure 3.7c shows an example of a part-line's enclosing circle that does not have sufficient foreground pixels; therefore its associated part-line is rejected. This step is a relaxation of the neck-based parts extraction algorithm described in [43], which mandates complete inscription of the enclosing circle. This deviation was necessary in our case, since our shapes have irregular noisy boundaries. Figure 3.7d shows the filtered double-point part lines. The retained part-lines represent the list of candidate partition lines.

```
1.  Shape Decomposition (object)
2.  {
3.     Detect concave points;
4.     Detect double-point part-lines;
5.     Filter double-point part-lines;
6.     Detect single-point part-lines;
7.     Filter single-point part-lines;
8.     Assign lists of concave points, double-point part-lines, and single-point part-lines
    to object
9.     Push object into stack;
10.    While (object stack is not empty)
11.    {
12.       Pop object and its lists from stack;
13.       if (double-point pat-line list is not empty)
14.       {
15.          Find best (maximum eccentricity) double-point part-line;
16.          if (best double-point part-line is found)
17.          {
18.             Partition object using best double-point part-line;
19.             Filter concave points and part-lines lists;
20.             Assign concave points and part-lines lists to the two new objects;
21.             Push the two new objects into stack;
22.          }
23.          else // best double-point part-line is not found
24.          {
25.             Push the object into stack;
26.          }
27.       }
28.       else if (concave points list is not empty)
29.       {
30.          Find best single-point part-line;
31.          if (best single-point part-line is found)
32.          {
33.             Partition object using best single-point part-line;
34.             Filter concave points and part-lines lists;
35.             Assign concave points and part-lines lists to the two new objects;
36.             Push the two new objects into stack;
37.          }
38.       }
39.    }
40. }
```

Figure 3.6 The shape decomposition algorithm.

In step 6, we detect all *single-point part-lines*, which are associated with a single concave point. We consider two vectors for every associated concave point. These vectors are aligned with the two curve segments that define the concave angle of that point, and they point away from the point and across the shape as shown in Figure 3.7e. Ideally the vector orientation should be calculated by fitting a straight-line with a number of boundary points before and after the concave point in order to better represent the curve orientation. But as an approximation we chose to define the vectors by a single boundary point at a fixed distance before and after the curvature minimum point. We then calculate the intersection of the two vectors with the object boundary. The two lines defined by the curvature minimum and the boundary intersection points represent candidate partition lines and are called single-point part-lines. In step 7, we filter the single-point part-lines list using the enclosing circle criterion described above for filtering the double-point part-lines. In steps 8 and 9, we assign the list of concave points, double-point part-lines, and single-point part-lines to the object and pushing it into a stack that holds binary objects and their associated lists.

In steps 10-12, the hierarchical decomposition starts by popping one object from the stack along with its lists. The algorithm considers the double-point part-lines first. In step 15, we utilize the previously described observation that when two adjacent vessels are linked, the eccentricity of the resulting linked object will be less than that of at least one of the two original vessel objects. We find the "best part-line" from the candidate list of double-point part-lines as the line which results in the maximum eccentricity part. In our example, we consider each part-line of the four lines shown in Figure 3.7d as a partition line. We then calculate the eccentricity of the two resulting parts, and compare those with the original object eccentricity. The best part-line is the one that is associated with the maximum eccentricity part, as highlighted in Figure 3.7d. In step 18, we partition the object into two objects using the best double-point part-line found. Figure 3.7e shows the best part-line chosen with the two resulting parts, where the red part has higher eccentricity than the original object.

In step 19, the concave points list is filtered by removing the two points of the best part-line, and the double-point part-line list is filtered by removing all part-lines that either intersect or share an end point with the best part-line. Figure 3.8a shows the remaining

concave points and part-lines after filtering the two lists. In step 20, we assign the remaining concave points and part-lines to the two new parts to which they belong, as shown in Figure 3.8b. In step 21, the two new parts are pushed into the object stack along with their list of concave points and part-lines.

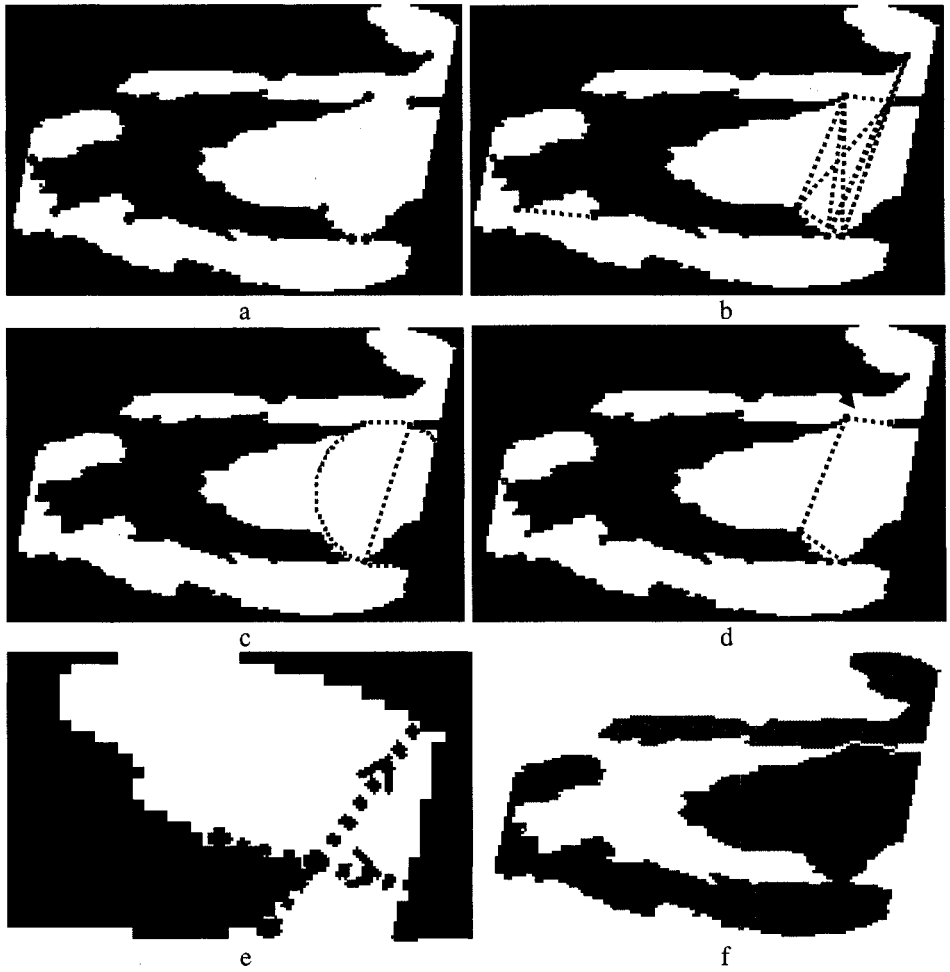


Figure 3.7 a: Negative-curvature minima. b: Double-point part-lines. c: A part-line's enclosing circle. d: Filtered double-point part-lines. e: Two single-point part-lines associated with a concave point. f: First partition with best part-line.

The process of finding a best double-point part-line is repeated for every object in the stack. If a best part-line is found, then it is used to partition the shape further into two new parts, until all double-point part-lines are consumed. A part-line is consumed by accepting it as a partition line, discarding it as a result of the filtering process of a former part-line, or

discarding it if it does not result in a part with a higher eccentricity than the initial object. Figure 3.8c shows another object with double-point part-lines and the best part-line found. Figure 3.8d shows the resulting two new parts that are pushed into the stack along with their associated lists of concave points and single-point part-lines. Figure 3.8e shows the object decomposition after consuming all double-point part-lines.

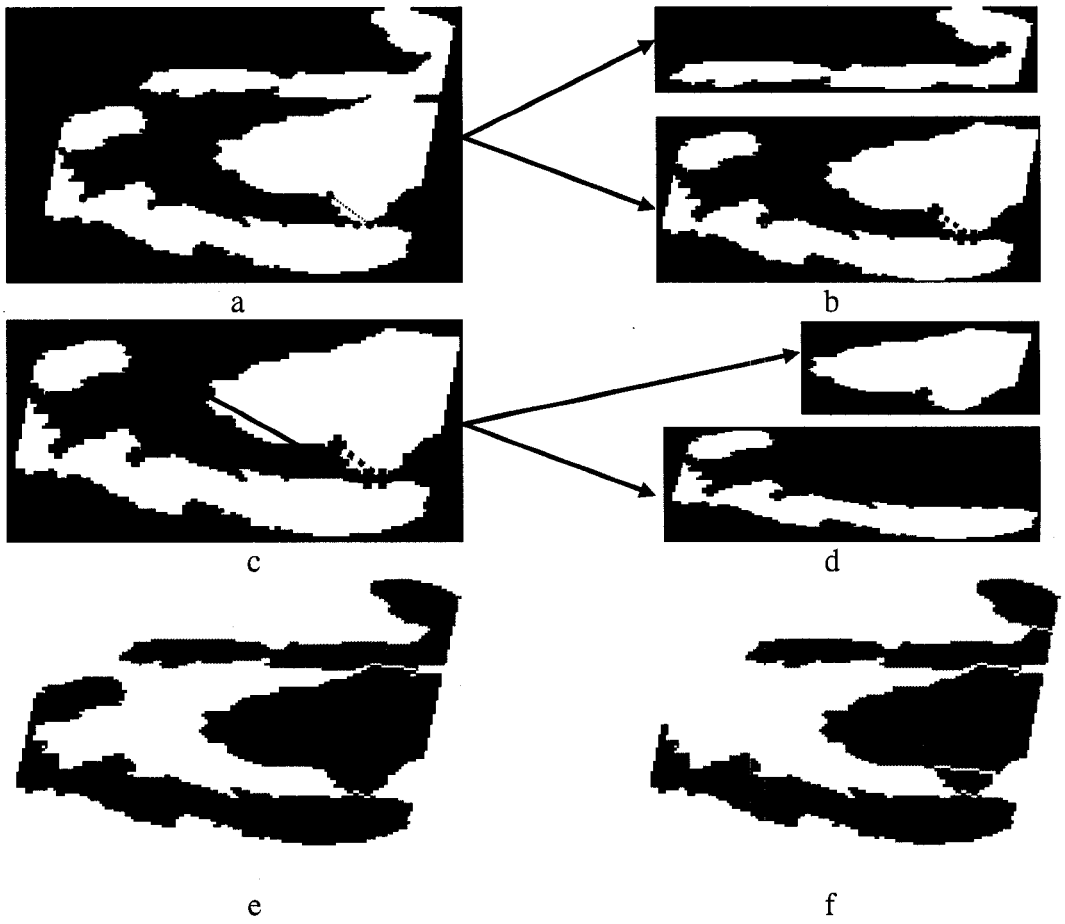


Figure 3.8 a: Filtered concave points and part-line lists. b: The two new parts with their associated concave points and double-point part-lines. c: A part with double-point part-lines. d: The two new parts with their associated concave points. e: Object partition after consuming all double-point part-lines. f: The final partition of the original object.

So far the original shape is partitioned into several parts using the double-point part-lines, each of which still has zero or more concave points associated with it. These points are either associated with true partition lines (single-point part-lines) or false points detected

because of the irregular shape boundary. The following steps describe how to find the remaining true partitions and discard the false ones.

Steps 28 to 36 involve the processing of the single-point part-lines, which is identical to the processing of the double-point part-lines. The single-point part-lines are considered for partitioning the shape further using the eccentricity criterion. The algorithm stops after consuming all part-lines. Figure 3.8f shows the final partition of the shape after consuming all single-point part-lines.

It is evident from the final partitioning that the four vessels are successfully segmented. One comment here is that the largest vessel is split into two parts (the light blue and yellow parts); this is due to the discontinuity of the color Doppler data inside the vessel in the original image. In future work we will try to develop an algorithm to merge parts that belong to the same vessel. Another comment is that there is an additional segment (dark blue) that represents the link between two vessels (light blue and red). This is expected and acceptable, since it should not affect future applications that rely on vessel recognition.

3.3.2.3 Vessel Segmentation Results

More results are presented here of the shape decomposition-based vessel segmentation technique. Figure 3.9 shows results of different frames of the same color Doppler cineloop of Figure 3.3. Despite the variations among the input images, the decomposition results were consistent and the four vessels were successfully segmented. This illustrates the robustness of the technique against minor changes of the input image. Figure 3.10 and Figure 3.11 show the results of processing other color images with multiple linked vessels. The technique successfully segmented the linked vessels in every image.

3.3.2.4 Discussion

In this section, we have presented a technique to segment ultrasound color Doppler images based on a shape decomposition approach. The underlying problem is very challenging for traditional image segmentation techniques, such as deformable models or region growing approaches. However, by developing a shape decomposition technique that is tailored to ultrasound-specific vessel appearance; we were able to successfully recognize distinct vessels. We have tested our technique on 60 images (5 frames for each dataset), and it successfully segmented the linked vessels in every image.

Figure 3.9 Partitions of different frames of the same color Doppler dataset.

The purpose of our shape decomposition technique differs from the one described in [43], since our goal was specific, to segment distinct vessels in color Doppler images where the output is well-defined, while in [43] the purpose was generic, to develop a shape decomposition technique for any visual form, where the output may be subjective. However, it is worth mentioning that our technique fulfills all the constraints of a good partitioning scheme listed in [43]: invariant to rigid transformations, robust to local transformations, stable to global transformations, and produces parts in a hierarchy.

Figure 3.10 Vessel segmentation results. Partitions of different color Doppler images with multiple linked vessels.

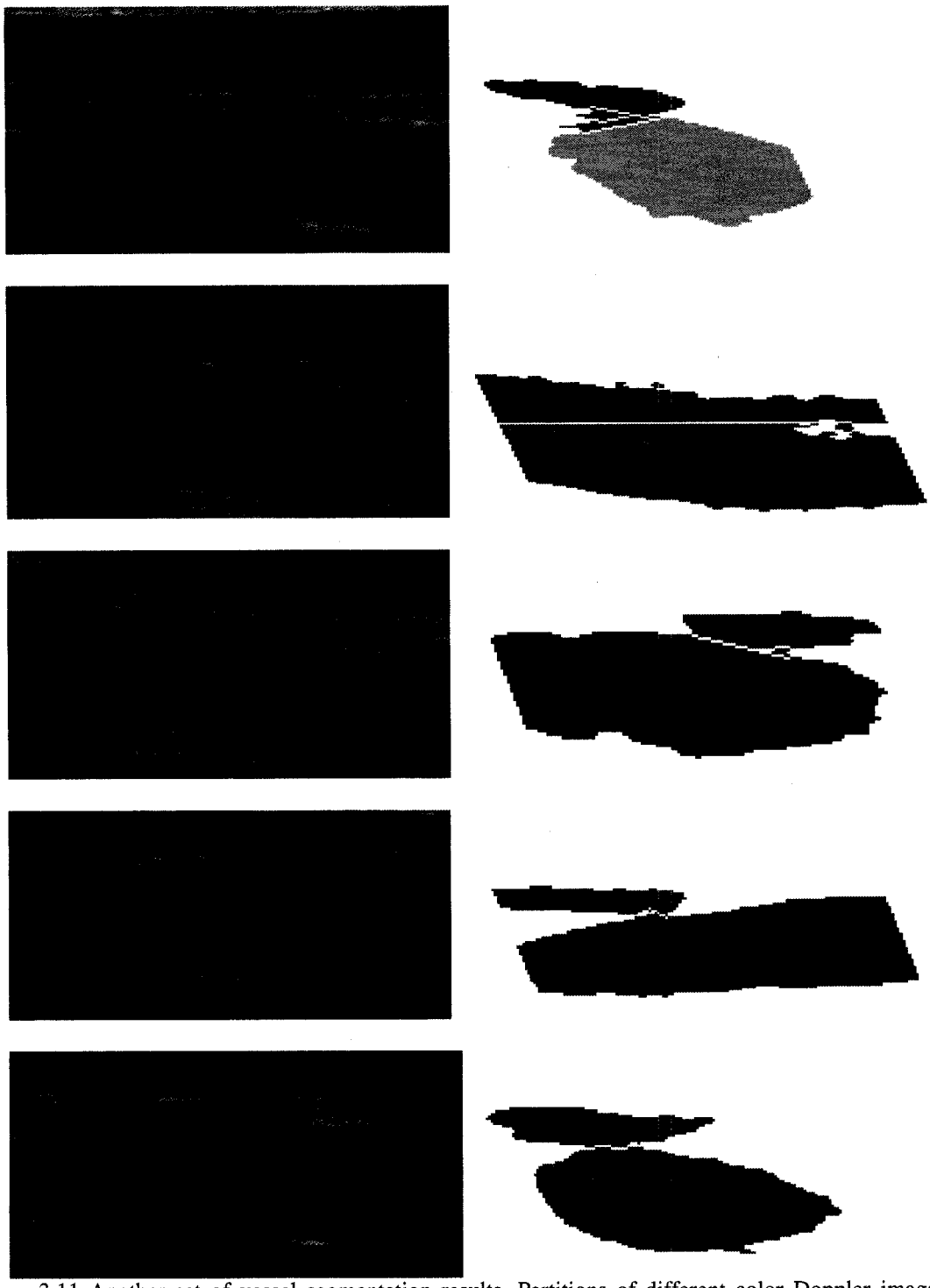


Figure 3.11 Another set of vessel segmentation results. Partitions of different color Doppler images with multiple linked vessels.

Our technique is invariant to rigid transformations, since the key aspects of the algorithm, which are negative-curvature minima, part-lines, and eccentricity (used in a relative sense to compare different objects), are all invariant to rigid transformations. Our technique inherits robustness and stability from the scheme described in [43]. It is evident in Figure 3.9 that local deformations do not affect parts that are remote to the change; also slight changes in the object boundary lead to slight changes in the parts as indicated by the partitioning of different frames of the same cineloop. Our technique is hierarchical in nature since every step starts with one object (two or more linked vessels) and ends up with exactly two objects.

We validated the segmentation results with expert sonographers from Philips Ultrasound Labs in Bothell, Washington. The raw color Doppler images were presented to different sonographers and they identified the distinct vessels that exist in each image. We then matched the manually identified vessels with the automatically segmented ones. In all cases, the automated segmentation matched the manual one, however the exact boundary of the vessels may differ. This will not cause a problem for our system since the purpose of the segmentation technique is not to identify the accurate vessel boundaries, but rather to recognize distinct vessels in the images. In the future, some automation applications may require accurate delineation of the vessel boundaries, which will require enhancing the segmentation technique presented here. The technique presented in this section was published in IEEE ICPR conference in 2006 [59].

3.4. Doppler Angle Automation Application

In this section we will describe one automation application that benefits from the newly-developed vessel segmentation technique. The application is to automate the estimation of the Doppler angle used to calculate the blood flow velocity as indicated in equation (4) above in section 2.5.2.

All current ultrasound systems offer a graphical user interface (Doppler angle line) which the sonographers can move anywhere over the image to locate the line center over the

vessel's site of interest. Then they can rotate a knob to align the Doppler line with the vessel axis, as indicated previously in Figure 1.2. The ultrasound system software internally uses the position of the Doppler angle line provided by the user to acquire Doppler spectrum signals from that position, while it uses the angle provided by the user to calculate the blood flow velocity. The two main drawbacks of this manual technique is the time consumed to manually position and align the Doppler angle line for each vessel interrogated during the exam, and the inaccuracy of the angle alignment that can happen due to complex vessel geometry or inexperience of the sonographer.

The blood flow velocity measurement in ultrasound plays a crucial rule in the diagnosis of vessel stenosis and plaque. For example a recent consensus paper [5] published by a panel of experts in vascular ultrasonography who represented the society of radiologists in ultrasound, stated that the peak systolic velocity measurement in Doppler ultrasound should be used along with the grayscale plaque findings to diagnose and grade the internal Carotid artery stenosis. Recently, Steinman et al. [6] and Lui et al. [7] stated that the error in the Doppler angle manual specification by the user is the most contributing factor to the error and variability in the velocity estimations.

For all the previous reasons, we have chosen the automation of the Doppler angle to be the first clinical application of the vessel segmentation task presented above. The following subsections will describe the application in detail.

3.4.1. Vessel Skeletonization

As a first step towards vessel quantification, we have chosen the skeleton representation as an efficient and compact representation of the segmented vessels. Maillet and Sharaiha [49] dedicated a chapter in their book to surveying the informal and mathematical definitions and algorithms that deal with object skeletonization. Informally a skeleton is a thin central structure that uniquely represents the object. Formally a perfect skeleton should be totally contained in the object, be one pixel wide, preserve the object

topology, and allow reconstruction of the object. However, some of these conditions may be relaxed depending on the application. One early mathematical model of a skeleton was introduced by Blum [27] as the locus of centers of maximal discs totally contained in the object. Another model was introduced by Montanari [50] using wave propagation with constant speed from each border point towards the inside of the object perpendicular to the border; the wavefront intersection points belong to the object skeleton.

In the literature, there are two main classes of techniques to extract binary object skeletons. The first class of techniques is called "skeletonization techniques" where a set of object pixels are characterized as eligible skeletal pixels, and then further processing is done to satisfy the skeleton conditions listed above. The second class of techniques is called "thinning techniques" which process the object iteratively until the skeleton is obtained. Skeletonization techniques use the fact that the value of the distance transform of the object pixels (which is the discrete distance from each object pixel to the nearest background pixel) indicates the radius of the largest disc contained in the object. Thus the skeleton pixels correspond to the local maxima in the distance transform map of the object. Further processing may be applied to add or remove pixels to the eligible skeleton pixels to form the final skeleton. Also a pruning process can be applied to remove the small branches of the skeleton to obtain the object center line. Thinning techniques usually use morphological approaches, such as morphological skeleton or thinning operators [49] to mimic the wave propagation model of the skeleton.

For our purpose of extracting the vessel skeletons, we used an optimized code from Philips Research Labs in France and Athens that implemented a recent method for object skeletonization introduced by Bitter et al. [51]. This graph-theoretic approach maps the object pixels into graph vertices and the pixel neighbor relations to graph edges. The technique integrates a modified version of Dijkstra's shortest path algorithm with the inverse of the distance map of the object (which acts as a weighting or penalty function of the object pixels), resulting in successful pruning of minor branches and robust overall performance. Figure 3.12 and Figure 3.13 show multiple examples of the vessel skeletonization results for different vessels.

3.4.2. Doppler Angle Estimation

The final step of our automation application is to calculate the Doppler angle for any arbitrary point, specified by the sonographer, within the vessel. For this purpose we use the skeleton points around the specified vessel site of interest to fit a least-squares line, whose orientation is taken as the estimate of the Doppler angle at this site.

We have experimented with an adaptive version of the classical weighted least-squares fitting method, developed by Philips Research Labs in Athens. The classical weighted least squares method minimizes the total square error given as [52]:

$$\varepsilon^2(a,b) = \sum_{i=1}^{N} \left(\frac{y_i - a - bx_i}{\sigma^2}\right)^2 \qquad (17)$$

where (x_i, y_i) are the skeleton points around the site of interest, a and b are the line parameters, and σ^2 is the variance of each skeleton point. The adaptive part came from using the square of the distance transform value of the skeleton point as the reciprocal of the variance to give less weight to outlier skeleton points closer to the object boundary. Figure 3.12 and Figure 3.13 show multiple examples of the Doppler angle estimation results for different vessels.

3.4.3. Clinical Evaluation

It was necessary to verify the accuracy of the automatic Doppler angle estimation against the manual angle setting currently done by sonographers and physicians. For that purpose we compared the automatically detected angles with the manual angles set by expert sonographers from Philips Ultrasound.

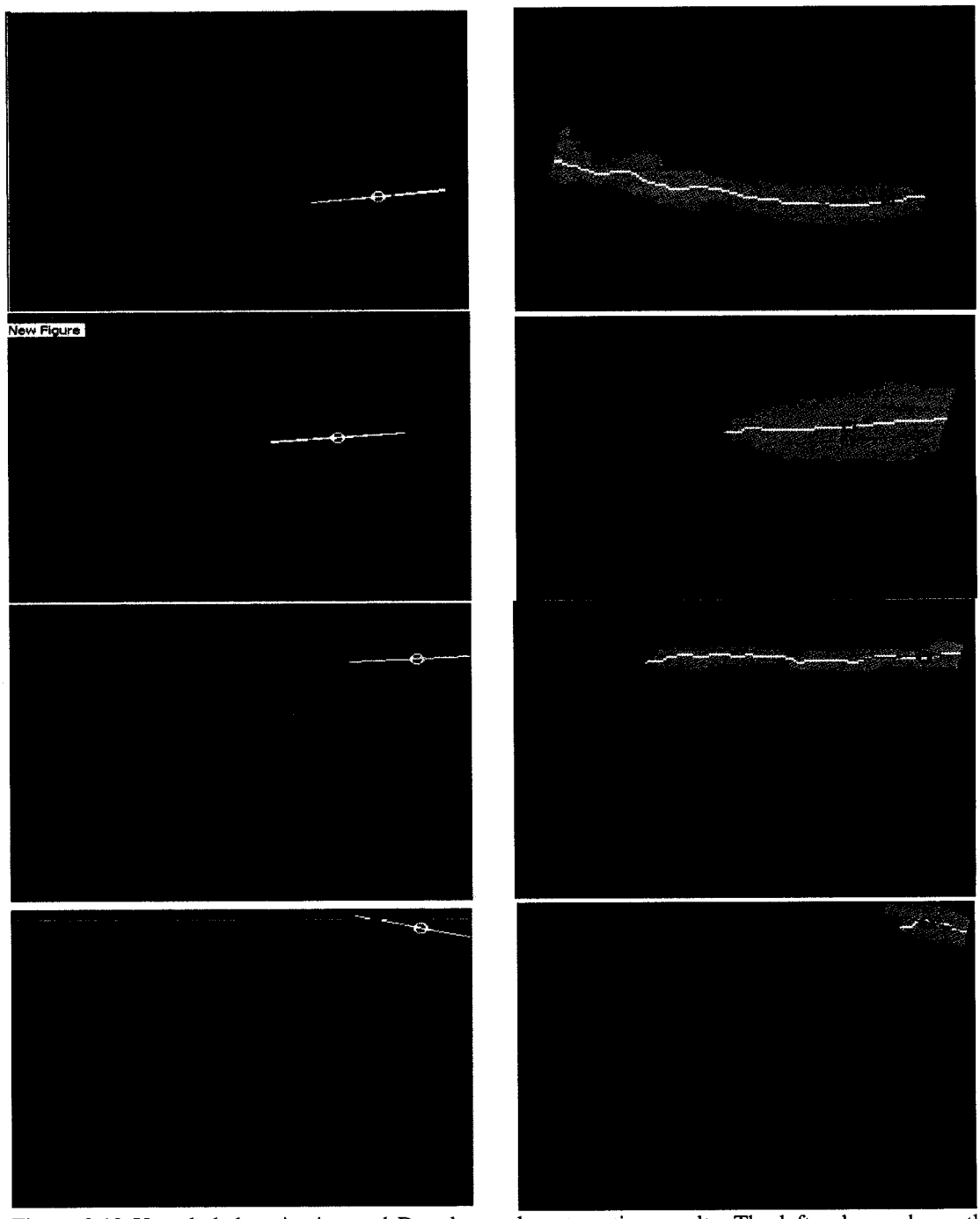


Figure 3.12 Vessel skeletonization and Doppler angle automation results. The left column shows the composite grayscale and color ultrasound image with the estimated angle as a white straight line. The right column shows the segmented image with the skeleton of the vessel of interest as a white curve, an arbitrary vessel site of interest as a red circle, the nearest skeleton point to the vessel site as a green rectangle, and the Doppler angle estimation as a red straight line.

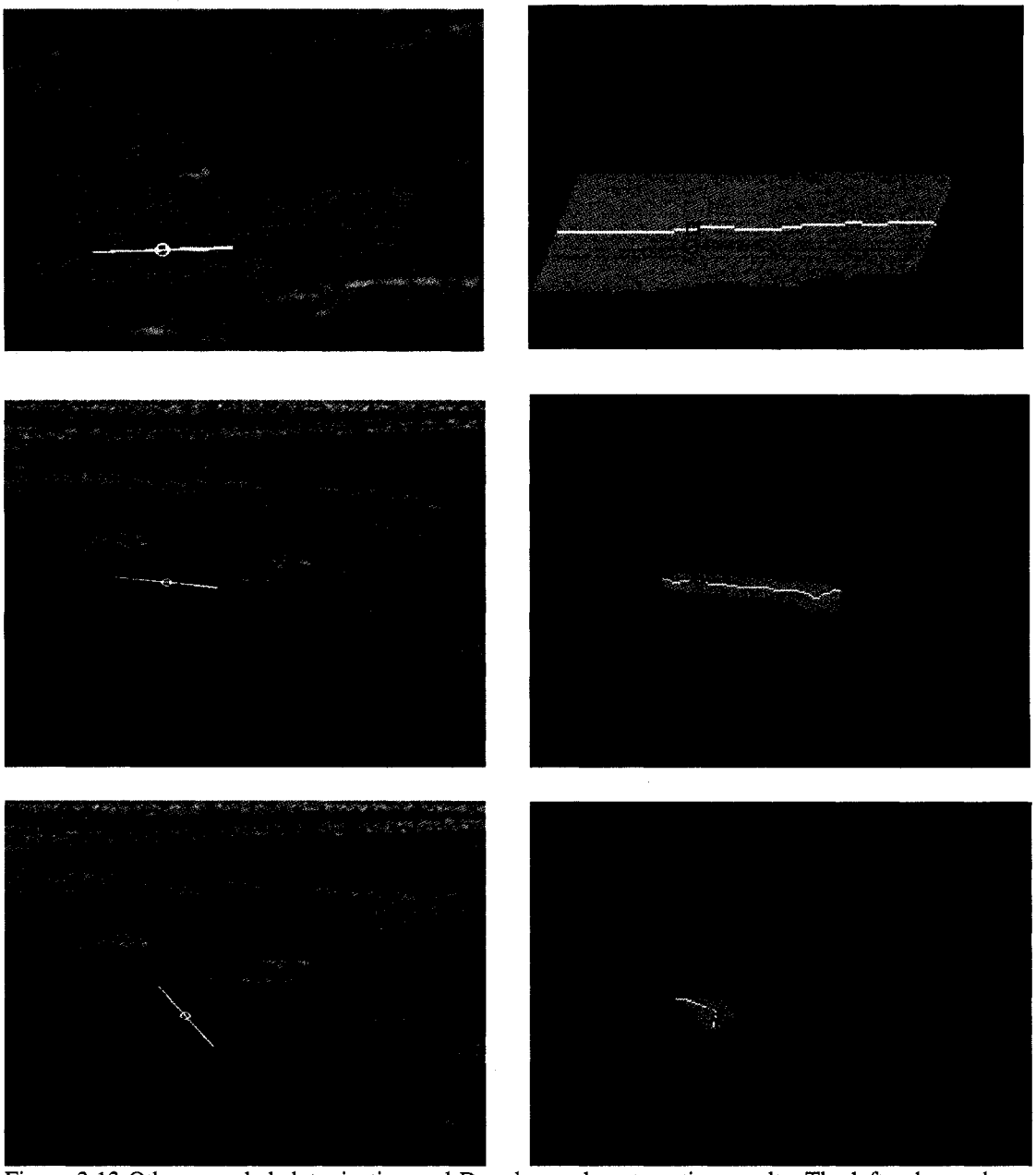


Figure 3.13 Other vessel skeletonization and Doppler angle automation results. The left column shows the composite grayscale and color ultrasound image with the estimated angle as a white straight line. The right column shows the segmented image with the skeleton of the vessel of interest as a white curve, an arbitrary vessel site of interest as a red circle, the nearest skeleton point to the vessel site as a green rectangle, and the Doppler angle estimation as a red straight line.

We conducted two types of experiments to quantify the accuracy of the angle estimation. In the first type of experiments, we used a Matlab© simulation to present color Doppler images to the sonographers and provide them with a simple user interface tools to draw the Doppler angle line manually to align it with the vessel orientation, according to their perception. Then we calculate the angle automatically using our technique for the same vessel site. Figure 3.14 shows a screenshot of the Matlab© simulation; the dashed yellow line is manually drawn by the sonographer to represent the vessel's orientation at this site. The solid white line is the automatic angle detected by the developed automation technique.

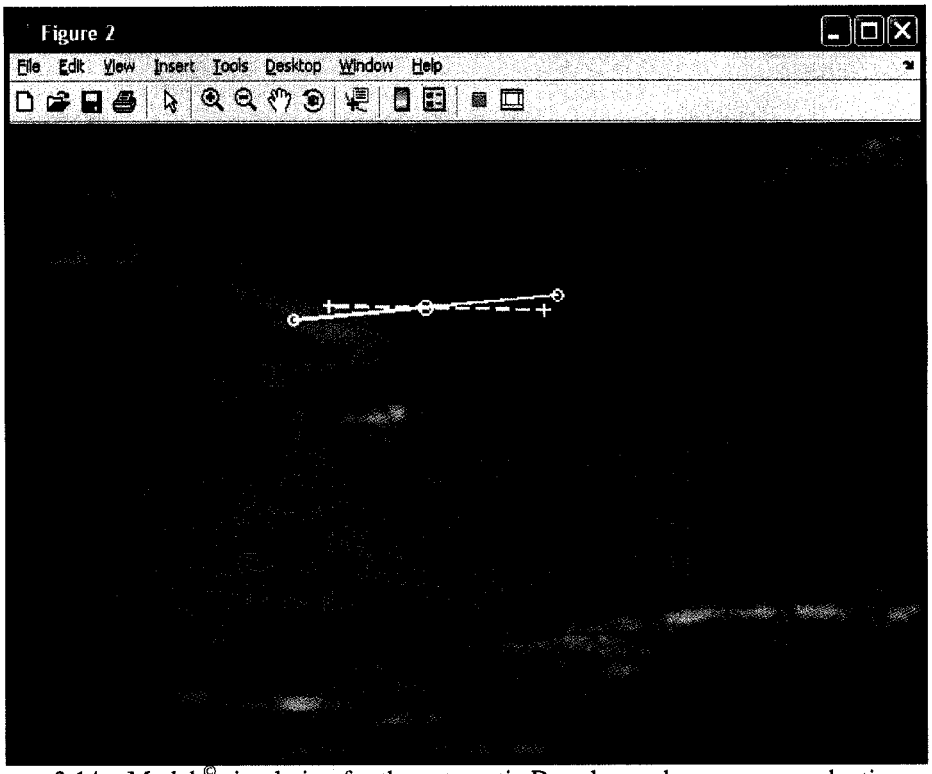


Figure 3.14 Matlab© simulation for the automatic Doppler angle accuracy evaluation.

In the second type of experiments, the participating sonographers used the developed real-time prototype of the Doppler angle automation technique on the Philips ultrasound system iU22 [4] to scan volunteered human models in a real clinical setup. The sonographers conducted normal vascular exams by interrogating different vessels. They used the user-interface knob provided by the system to set the manual Doppler angle line to align it with the

vessel orientation. Then they triggered the automation at the same vessel site to compare the results.

We collected data from both experiment types conducted by three different sonographers and analyzed the results as shown in Figure 3.15 and Figure 3.16 for the Matlab© simulation, and in Figure 3.17 and Figure 3.18 for the real-time prototype. Two display formats are presented: the x-y scatter plot, which is useful for appreciating the strength of the relation between the two types of measurements, and the Bland-Altman plot (difference of individual measurements vs. average of individual measurements), which is best for quantifying numerical agreement [58]. For both types of experiments, the automatic and manual angle results are very similar, exhibiting high correlation coefficient (0.97 for the Matlab© simulation and 0.99 for the real-time prototype), low mean of individual differences (-0.93 degrees for the Matlab© simulation and 1.66 degrees for the real-time prototype), plus reasonable standard deviation of the individual differences (5.4 degrees for the Matlab© simulation and 6.44 for the real-time prototype).

It is worth mentioning that the effect of the difference between the manual and automatic angles on the velocity estimation (which is the diagnostic criterion used to grade vessel stenosis) will be further minimized by the fact that the velocity relies on the cosine of the angle not the angle itself.

The sonographers who contributed to the clinical evaluation studies anticipate that such an application will streamline the ultrasound Doppler vascular exam and achieve more accurate and standardized outcomes. This should enhance the quality of the patient diagnosis and treatment decisions. The results of the automation application were published in the Society for Imaging Informatics in Medicine (SIIM) conference in 2008 [53].

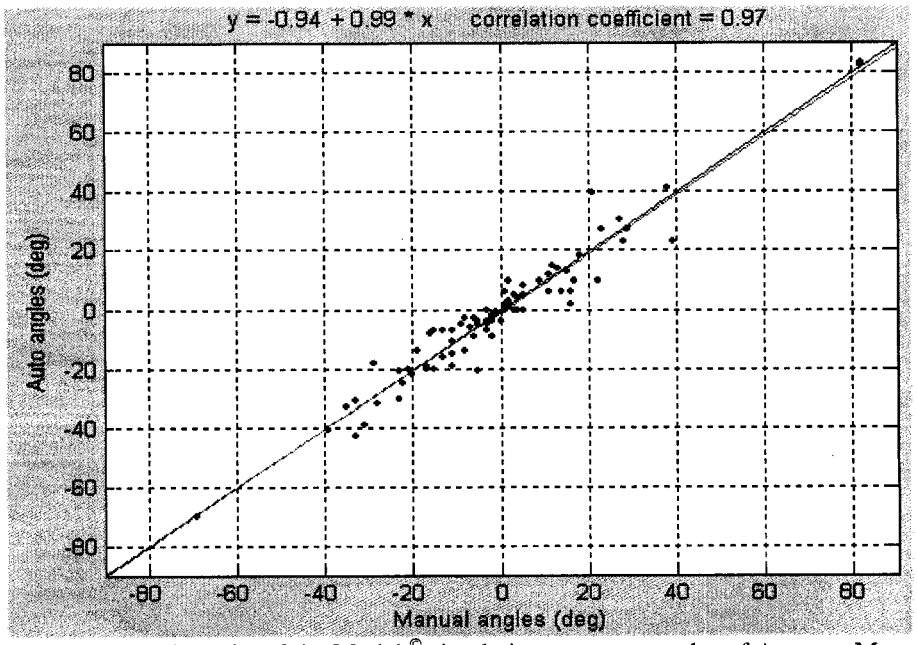


Figure 3.15 Experimental results of the Matlab© simulation. x-y scatter plot of Auto vs. Manual angles, plus identity line (blue), and least-squares-line (magenta).

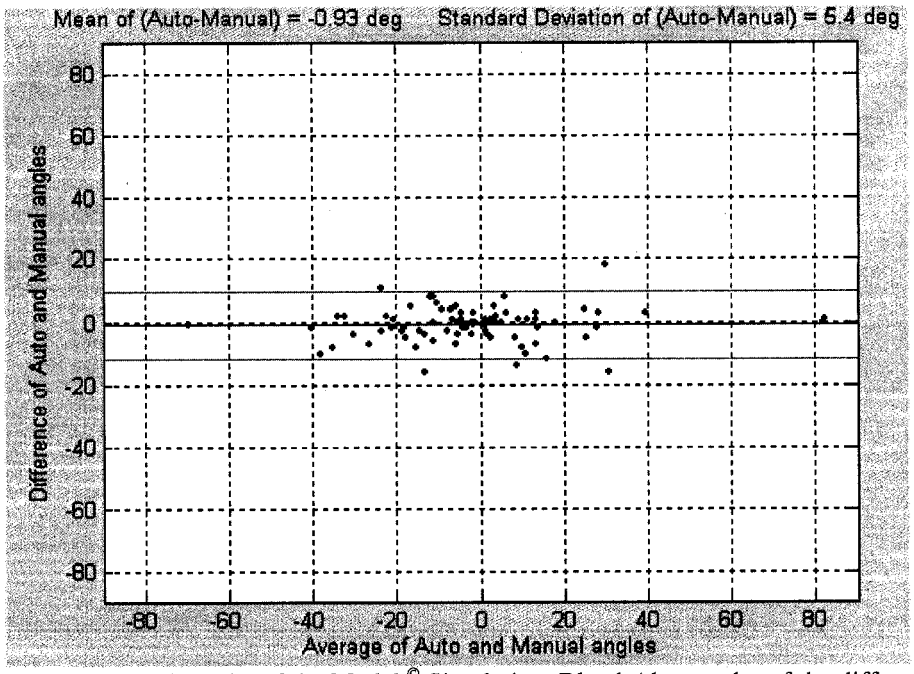


Figure 3.16 Experimental results of the Matlab© Simulation. Bland-Altman plot of the difference of the Auto and Manual angles vs. the average of the Auto and Manual angles, plus mean of differences (blue), and mean ± two standard deviations of differences (magenta).

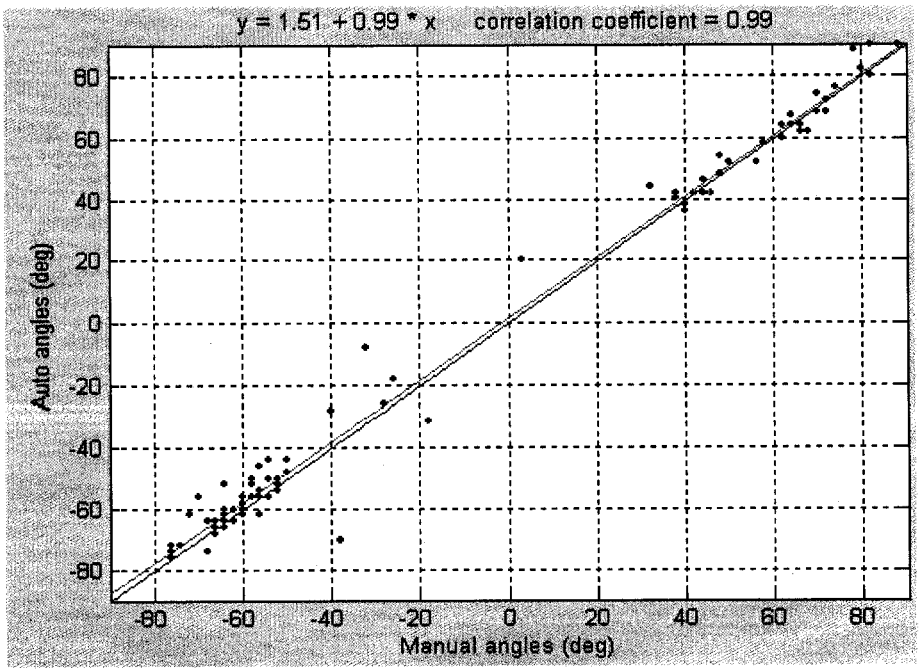


Figure 3.17 Experimental results of the real-time prototype. x-y scatter plot of Auto vs. Manual angles, plus identity line (blue), and least-squares-line (magenta).

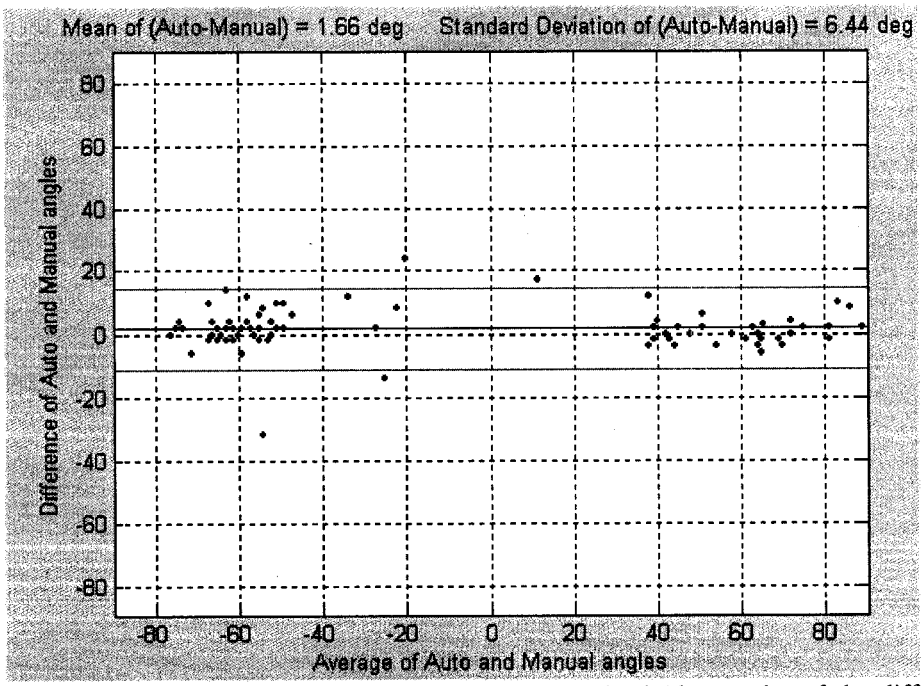


Figure 3.18 Experimental results of the real-time prototype. Bland-Altman plot of the difference of Auto and Manual angles vs. the average of Auto and Manual angles, plus mean of differences (blue), and mean ± two standard deviations of differences (magenta).

3.5. Conclusions and Future Work

So far we have dealt with a novel and challenging problem to develop a vessel segmentation technique for color Doppler ultrasound imaging as a first step towards a complete vessel recognition system. We have also built an automation clinical application on top of the segmentation step. The automation application targets an important clinical problem, which is setting the ultrasound Doppler angle. The angle setting greatly affects the blood flow velocity measurements, which in turn greatly affects the patient diagnosis for vascular diseases. In the following subsections we will suggest future enhancements for the segmentation and automation steps.

3.5.1. Vessel Segmentation Enhancements

We have developed a novel segmentation technique based on shape decomposition approaches to recognize distinct vessels in ultrasound color Doppler images. Although the developed technique showed promising results, it still fails to address challenging cases, such as vessel over-segmentation due to poor blood filling, noisy color images with flash artifacts, and complex geometrical vessels due to pathology.

One enhancement would be to apply multiscale curvature-based shape evolution techniques to simplify the representation of the noisy vessel contours in ultrasound images. There are many multi-scale shape representation techniques in the literature based on Gaussian [53], morphological [55], and wavelets [56] based techniques. The multiscale approach should enhance the robustness of the segmentation technique by enhancing the critical point detection step and it should also enhance the accuracy of the angle estimation, since the skeletons will be smoother.

Another enhancement would be combining grayscale and color Doppler information to guide the segmentation. Through studying many ultrasound vessel image sequences, we found that none of the two image types is ideal to represent the real vessel anatomy alone due

to the inherent artifacts in each image type. We believe that through combining the useful information from both image types, we should be able to develop robust vessel segmentation techniques for ultrasound imaging.

The use of temporal information to guide the segmentation is also a promising direction, since ultrasound imaging is a real-time modality. By using the correlation between image sequences, we should be able to track the vessel in every frame. We have already used the temporal information briefly by averaging a number of frames to remove the pulsatility artifact from the vessel of interest. However more advanced techniques can be developed to deal with motion artifacts (due to patient breathing or transducer movement) through the introduction of motion estimation techniques. Also scene change detection techniques can be applied to filter outlier frames (due to movement) before averaging.

3.5.2. Doppler Angle Automation Enhancement and Clinical Validation

There are still opportunities to enhance the Doppler angle automation application that was presented earlier. The angle estimation technique using the least squares line fitting of the skeleton points gave reasonable results in most cases. However the estimated angle is not accurate if the vessel is curved or tortuous, since the line fit may be considering points that do not contribute to the vessel orientation at this specific site. One enhancement could be to adaptively select a variable number of skeleton points to contribute to the line fittings based on minimizing the fitting error. Another enhancement could be to fit a higher-order polynomial or a spline to the adaptively selected skeleton points. Finally, an alternative approach would be to approximate the Doppler angle as the Discrete Tangent Orientation of the skeleton curve. There are many techniques in the literature to estimate the discrete tangent orientation [57] and we plan to evaluate them in the context of automatic Doppler angle estimation.

Chapter 4 Phase Unwrapping Foundations and Literature Review

4.1. Introduction

Velocity aliasing is a very common artifact in color Doppler ultrasound imaging. In this chapter and the following one we will address the color Doppler aliasing problem to restore the true velocity values from the aliased ones as a preprocessing step of our vessel recognition system, as shown in Figure 4.1. The aliasing artifact occurs when sampling frequency (or pulse repetition frequency 'PRF') used to acquire color Doppler images is not high enough to unambiguously sample the highest blood flow velocity within the imaged vessels. The artifact manifests itself as high velocity pixels that appear to have reverse flow velocities, as shown in Figure 4.2. Current ultrasound systems usually offer a manual control for the PRF to enable sonographers to increase the sampling rate when velocity aliasing occurs. Sonographers usually prefer to use a sampling rate that is just adequate for acquiring the blood flow pattern of the vessel of interest without aliasing. Setting the PRF too high will decrease the sensitivity of color Doppler images. In many situations the PRF reaches its maximum setting without eliminating the aliasing problem completely. This may happen with pathological cases where the vessel of interest is stenosed and the blood flow within the vessel has very high velocity. This could also happen with deep vessels for which the PRF is limited by the round trip of the ultrasound pulse. Sonographers are used to the velocity aliasing artifact and sometimes use it as a diagnostic tool to indicate stenosis or narrowing in the vessel when the aliasing degree increases.

Figure 4.1 The stages of the vessel recognition system. The velocity restoration stage is the focus of the current and next chapters.

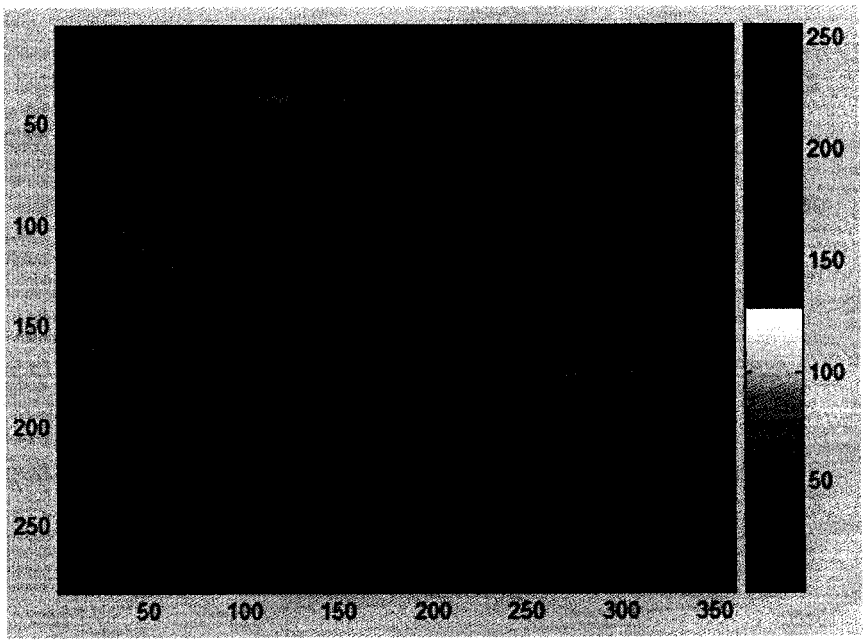


Figure 4.2 A color Doppler ultrasound image of the proximal common carotid artery. The unaliased velocity pixels are shown in blue while the aliased velocity pixels are shown in red.

Solving the velocity aliasing problem is a key to the vessel recognition goal, which is the thrust of this work. The goal of the solution would be to restore the original velocity values from the aliased ones. Then it would be feasible to characterize the vessel from the unaliased velocity loops by deriving estimates for differentiating features such as maximum velocity, pulsatility, and flow direction.

In this chapter we will handle the velocity aliasing problem for color Doppler ultrasound images. This is, by itself, a novel research topic, since there are no prior publications for treating the color Doppler ultrasound aliasing problem. In section 2.6, we described the theory behind color Doppler ultrasound image acquisition to highlight the fact that color Doppler images are phase images. Hence the velocity aliasing problem can be formulated as a phase unwrapping problem. Phase unwrapping is a well-known research topic in synthetic aperture radar (SAR) interferometry, optical interferometry, adaptive optics, and to some extent in magnetic resonance imaging (MRI). In section 4.2, we will explain the well-established theory of phase unwrapping. In section 4.3, we will review the most important techniques for solving the phase wrapping problem. In section 4.4, we will evaluate the

performance of some classical phase unwrapping techniques with color Doppler ultrasound images and comment on their shortcomings. Finally in section 4.5 we will summarize our findings and conclusions regarding the performance of existing phase unwrapping techniques with color Doppler ultrasound data.

4.2. Phase Unwrapping Theory

In this section we will review important theoretical foundations of the phase unwrapping problem that were treated in depth by Ghiglia and Pritt in their book on phase unwrapping [60].

We start with recalling basic definitions from the signal processing field that are essential to the phase unwrapping problem. A *signal* is any time-varying (or space-varying) quantity. The simplest example of a signal is a periodic sinusoidal signal defined as:

$$s(t) = A\sin(2\pi t/T) = A\sin(2\pi ft) = A\sin(\omega t) \qquad (18)$$

where A is the signal magnitude, T is the signal period that defines the elapsed time of one cycle of the periodic sinusoidal signal (measured in seconds), and f is the signal frequency that defines how many signal cycles per second (measured in Hz). Figure 4.3a shows an example of different sinusoidal signals with different frequencies. ω is the signal angular frequency that defines the rotation rate of the signal if we plot the instantaneous angle on a circle as shown in Figure 4.3b (measured in radians/sec).

Figure 4.3a represents a time domain plot that shows how the signal changes over time. Another useful plot is the frequency domain plot which shows how much of the signal lies within each given frequency band over a range of frequencies.

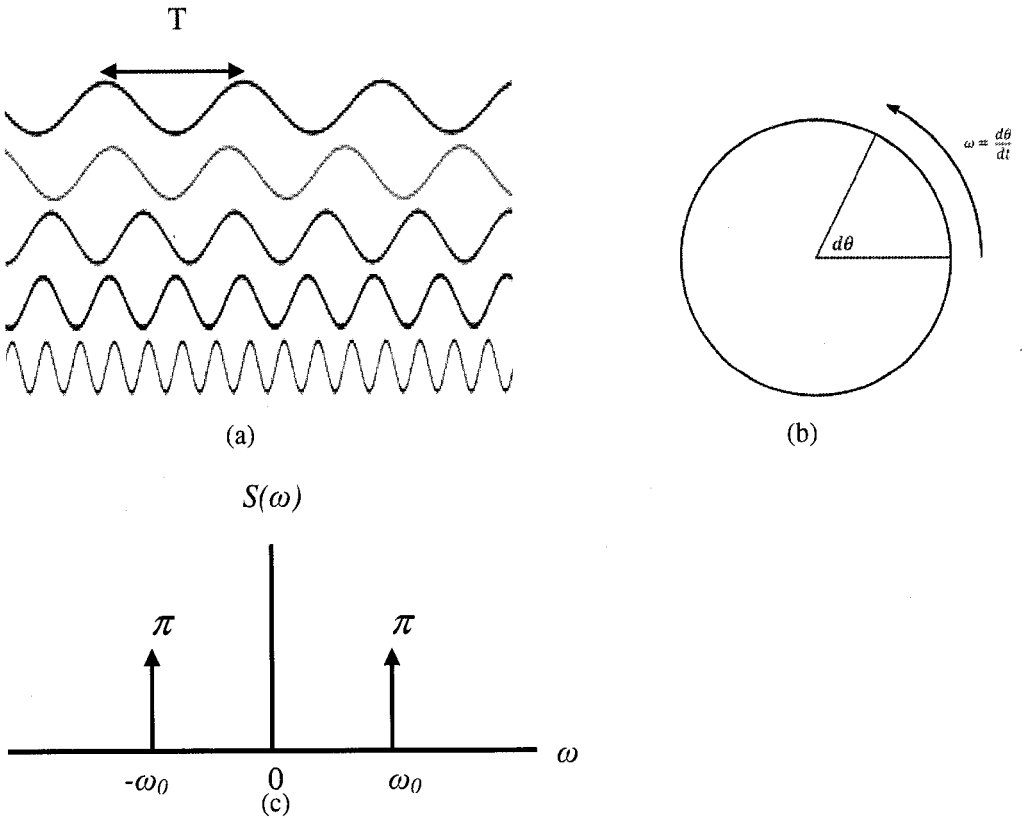


Figure 4.3 (a) Sinusoidal signals of various frequencies. The bottom signals have higher frequencies than those above. (b) Angular frequency is a measure of the rotation angle rate. (c) The Fourier transform of a sinusoidal signal.

The frequency analysis of a signal is achieved using the well-known mathematical operation, the forward Fourier transform defined as:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt \qquad (19)$$

The original signal can be reconstructed back from its Fourier transform components via the inverse Fourier transform defined as:

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega \qquad (20)$$

The Fourier transform simply states that the signal $s(t)$ can be decomposed into an infinite number of complex sinusoidal signals $e^{j\omega t}$ weighted by the Fourier coefficients $S(\omega)$. The forward transform calculates the coefficients, while the inverse transform reconstructs the signal back from its coefficients. The Fourier transform of a simple sinusoidal signal, as defined in equation (18), is zero everywhere except at $+\omega_0$ and $-\omega_0$, where there is an impulse; this is shown in Figure 4.3c. The Fourier transform of a sinusoidal signal indicates that the whole energy of the signal is concentrated at the single frequency that characterizes the signal.

The Fourier transform is a complex function with real and imaginary parts and can be expressed as:

$$S(\omega) = R(\omega) + jI(\omega) \qquad (21)$$

where $R(\omega)$ and $I(\omega)$ are the real part and imaginary part respectively. Usually the transform is expressed in terms of magnitude and phase spectrums, defined as:

$$|S(\omega)| = \sqrt{R(\omega)^2 + I(\omega)^2} \qquad \varphi(\omega) = \tan^{-1}(I(\omega)/R(\omega)) \qquad (22)$$

The magnitude spectrum defines the strength of each complex exponential comprising the Fourier decomposition of the signal. The phase spectrum of a signal defines the relative displacement of each complex exponential comprising the Fourier decomposition of that signal.

To recognize the importance of phase, we recall an experiment that Oppenheim and Lim conducted in their classical paper [77]. The experiment is simply swapping the phase spectrum of two different images and reconstructing the images back using the correct magnitude and the wrong phase. Figure 4.4 shows two images, one for Albert Einstein's and

the other for Mona Lisa. The figure also shows both the magnitude and phase spectrums of both images, calculated using the Fast Fourier Transform (FFT). The zero (dc) frequency is centered within the magnitude and phase spectrums. The magnitude spectrums show the distribution of the magnitudes of the spatial frequency components of the original images, while the phase spectrums appear random and not connected to the original images. Figure 4.5 shows the two new reconstructed images after swapping the phase information. The image in Figure 4.5a is reconstructed from the Mona Lisa's magnitude spectrum and Einstein's phase spectrum using the Inverse Fourier Transform (IFFT), while the image in Figure 4.5b is reconstructed from Einstein's magnitude spectrum and Mona Lisa's phase spectrum. It is obvious that the recognition of each image is due to more to the content of the phase spectrum rather than the magnitude spectrum. The incorrect magnitude spectrum only causes fuzziness in the reconstructed image. This small experiment emphasizes the importance of phase information that defines the relative position of each complex exponential comprising the Fourier decomposition.

One important fact, derived from the previous discussion, is that the phase must be extracted from the actual signal by some mathematical operation, such as the FFT. The calculated phase always lie between $+\pi$ and $-\pi$, as indicated by equation (22) above. The calculated phase is called the *principal phase* or *wrapped phase* value because the original phase is wrapped into the interval $(-\pi, \pi]$. The wrapping process is a nonlinear process defined as:

$$\psi(\omega) = \varphi(\omega) + 2\pi k(\omega) = W(\varphi(\omega)) \qquad (23)$$

$\psi(\omega)$ is the wrapped phase, $\varphi(\omega)$ is the true unwrapped phase, and $k(\omega)$ is an integer function that forces ψ to be within the interval $(-\pi, \pi]$. W is a wrapping operator that maps the true phase into the interval $(-\pi, \pi]$. Figure 4.6 shows an example of a linear true phase function and how it relates to the wrapped phase. In many applications $\varphi(\omega)$ relates to some physical quantity. In color Doppler ultrasound imaging it represents the velocity of the blood flow within a vessel. The phase unwrapping problem is concerned with obtaining an estimate

of the true phase values $\varphi(\omega)$ from the wrapped phase values $\psi(\omega)$. In reality, we work with digitized phase values where the independent variable ω is replaced by the discrete independent variable n.

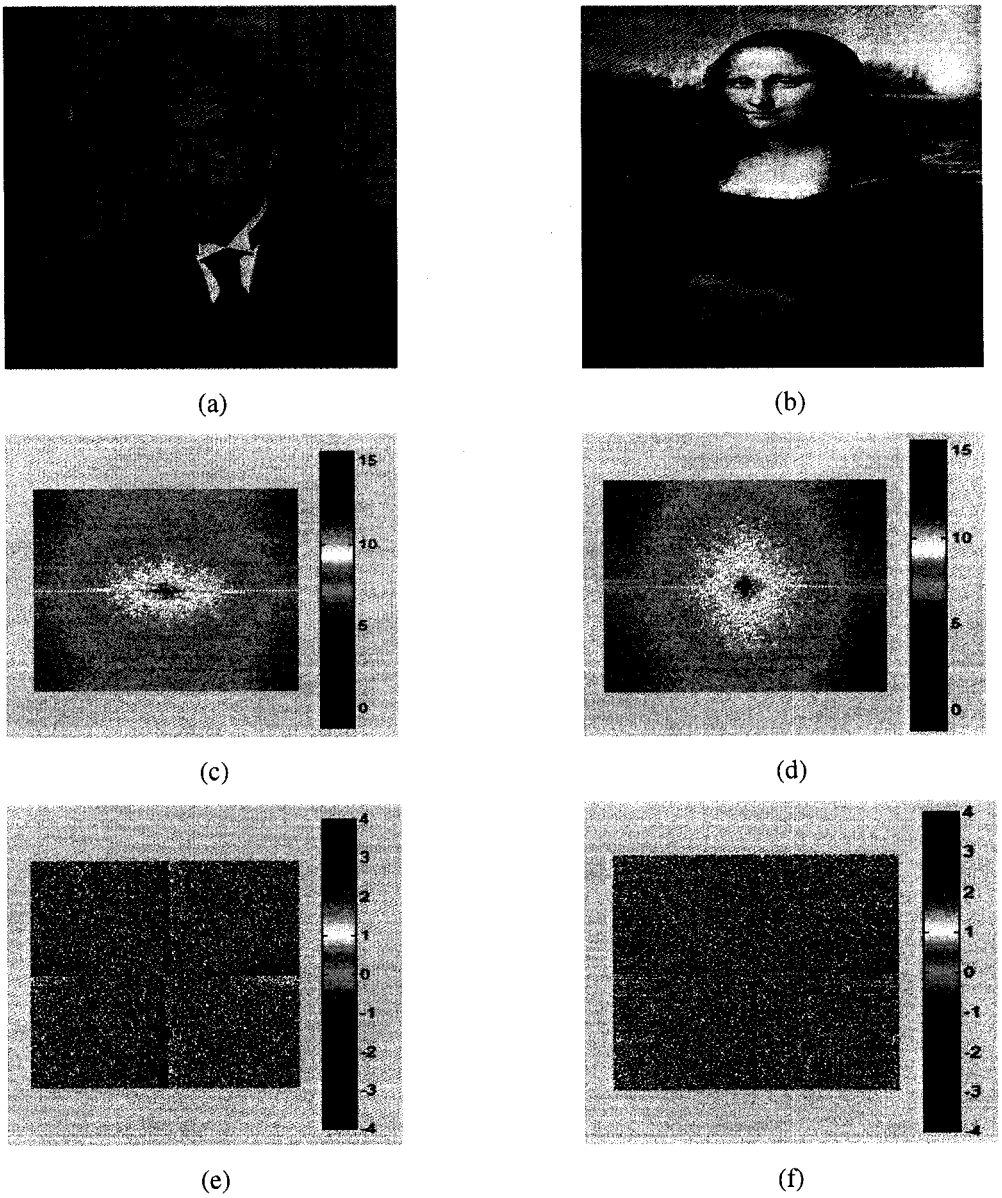
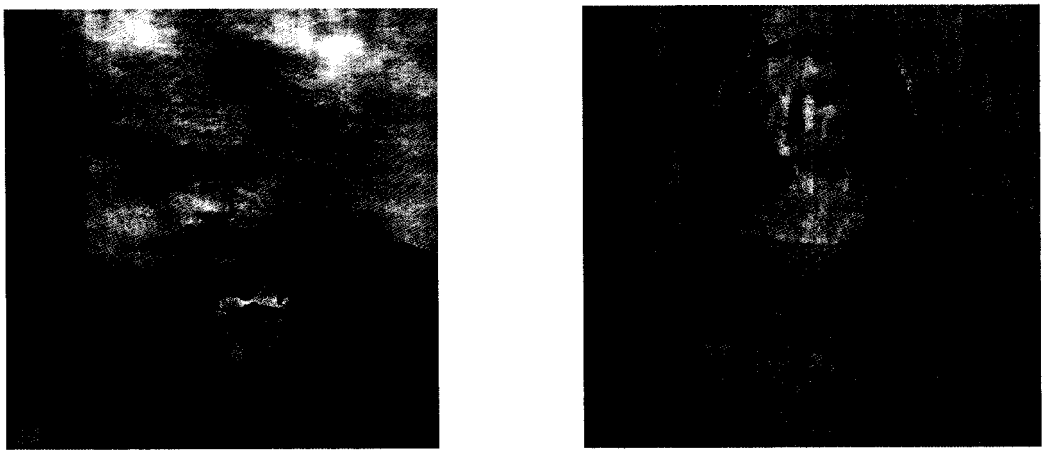


Figure 4.4 (a) Albert Einstein's image. (c) Log magnitude of its Fourier transform. (e) Phase of its Fourier transform. (b) Mona Lisa's image. (d) Log magnitude of its Fourier transform. (f) Phase of its Fourier transform.



(a) (b)

Figure 4.5 (a) Image reconstructed from Albert Einstein's phase spectrum and Mona Lisa's magnitude spectrum. (b) Image reconstructed from Mona Lisa's phase spectrum and Albert Einstein's magnitude spectrum.

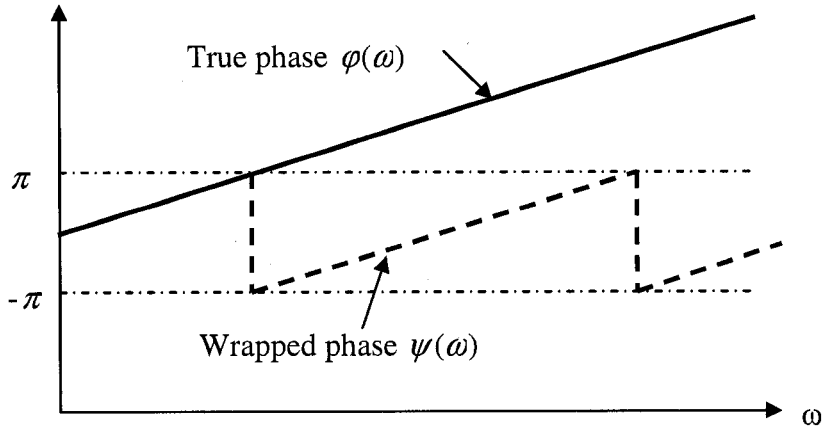


Figure 4.6 The relationship between the true phase $\varphi(\omega)$ and the wrapped phase $\psi(\omega)$.

In a classical phase unwrapping reference [61], Itoh analyzed the 1-D phase unwrapping problem and showed that the unwrapping phase can be obtained by integrating (summing) the wrapped phase differences, provided that the absolute true phase differences are less than π. The unwrapped phase values can be calculated recursively as follows:

$$\varphi(m) = \varphi(0) + \sum_{n=0}^{m-1} W(\nabla(\psi(n)) \qquad (24)$$

where ∇ is a difference (or gradient) operator defined as:

$$\nabla(\psi(n)) = \psi(n+1) - \psi(n) \qquad (25)$$

Itoh's phase unwrapping algorithm for 1-D phase signal can be summarized in the following pseudo-code:

1- Given 1-D wrapped phase data $\psi(n)$ for N samples, compute the phase gradients (differences):

$\nabla_n = \psi(n+1) - \psi(n)$ for n=0,..., N-2.

2- Compute the wrapped phase differences:

$W(\nabla_n) = \tan^{-1}(\sin(\nabla_n)/\cos(\nabla_n))$ for n=0,..., N-2.

3- Initialize the first unwrapped phase value: $\varphi(0) = \psi(0)$.

4- Unwrap the phase values by summing the wrapped phase differences:

$\varphi(n) = \varphi(n-1) + W(\nabla_{n-1})$ for n=0,..., N-1.

Figure 4.7 Pseudo-code for Itoh's 1-D phase unwrapping algorithm

The condition set by Itoh's method on successive true phase differences (to be less than π) can be expressed as follows.

$$-\pi < \nabla(\varphi(n)) < \pi \qquad (26)$$

This condition can be easily violated if the original complex signal, from which the phase is derived, is under-sampled (sampling rate is below Nyquist sampling rate); in which case the resulting true phase will be aliased and Itoh's unwrapping method will fail to recover the original true phase values. Another source of error in the sampled phase values is the noise which could be additive or multiplicative to the original phase values and will affect the quality of the recovered true phase values. Ideally Itoh's phase unwrapping method will succeed if the true phase data is continuous and noise free (with no more than π jumps). In real applications, either or both of these conditions can easily be violated where phase discontinuity and/or noise can exist in the measured phase data, which imposes a challenge to the unwrapping task. Phase unwrapping literature mainly deals with the phase discontinuity and noise problems to develop robust unwrapping techniques.

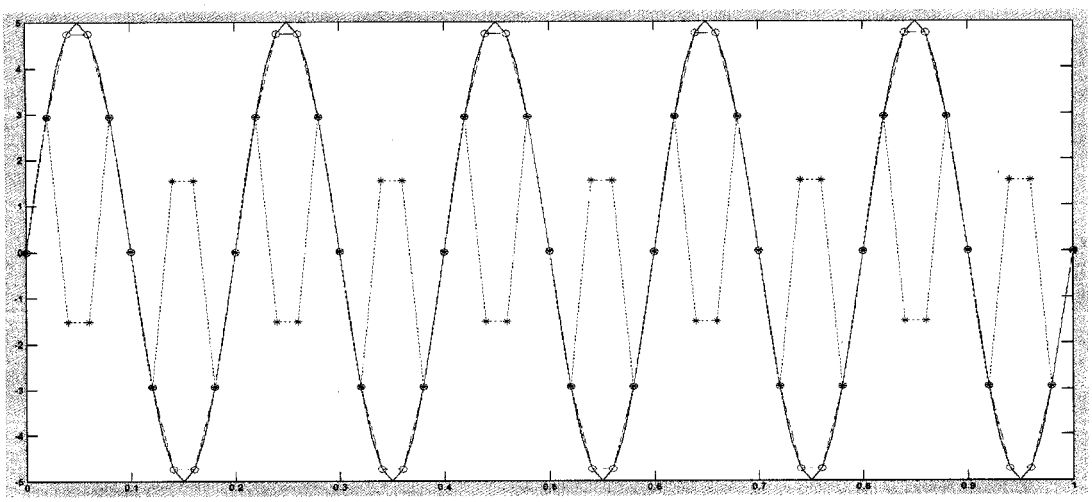
Figure 4.8 shows an example of Itoh's phase unwrapping method for a sinusoidal phase function. Figure 4.8a shows the case where the original phase signal is adequately sampled (blue asterisks) with a high enough sampling rate to prevent aliasing. In this case, the unwrapped results (red circles) match exactly the original true phase signal. Figure 4.8b shows the case where the sampling rate is not high enough to prevent aliasing and thus the unwrapped results do not match the original true phase signal. By examining the differences between the true phase samples in the two cases, it is evident that Itoh's method succeeds when the differences are less than π.

Itoh's 1-D phase unwrapping concept, that unwrapped phase can be obtained by integration of the phase gradient, can be extended to N-dimensional signals. Assuming the phase gradients are known along with the phase at some initial point r_0, the phase at some other point r can be obtained from the following path integral:

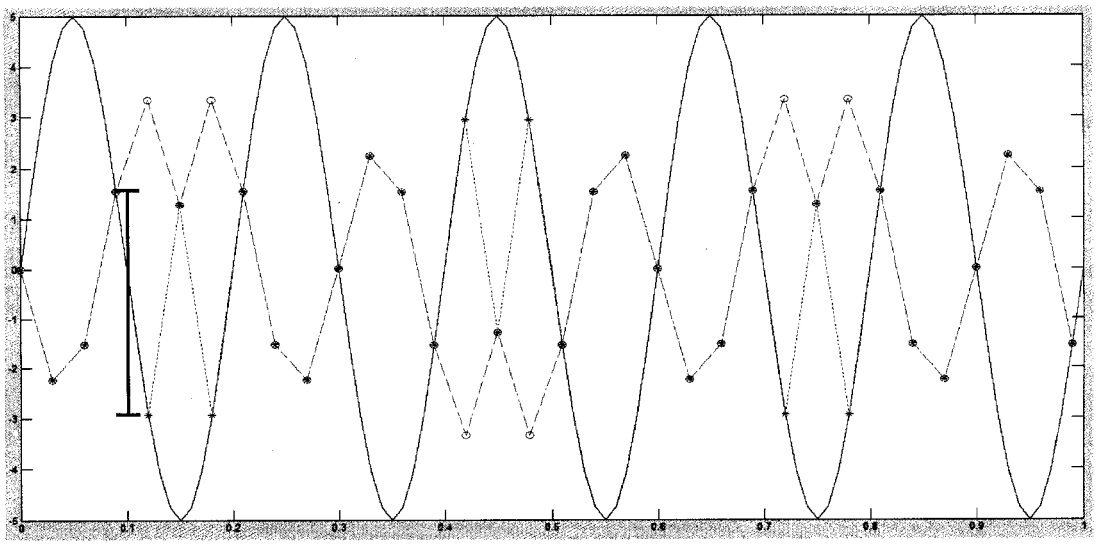
$$\varphi(r) = \int_C \nabla\varphi.dr + \varphi(r_0) \qquad (27)$$

where $\nabla\varphi$ is the phase gradient, $\varphi(r_0)$ is the true phase of an initial point r_0, and C is any path in the N-dimensional space connecting points r_0 and r. We have seen previously that aliasing or noise can destroy our ability to correctly unwrap the 1-D phase data by introducing discontinuity in the phase data (greater than π jumps). In two or more dimensions, aliasing or noise can make the unwrapped phase dependent on the integration path, since there will be many paths connecting the two points r_0 and r.

The notion of phase discontinuity in the context of 2-D phase unwrapping was first discovered by Ghiglia et al. in [62]. The authors found that the unwrapped phase depended on the followed path for the unwrapping. Furthermore, they have discovered that the phase discontinuities (or inconsistencies, as they called it initially) are restricted to certain isolated points or regions. Phase inconsistencies were detected by integrating wrapped phase differences around each (2×2) sample path in the 2-D phase map. Inconsistencies were flagged where those closed-path integrals yielded non-zero values. There was a sign associated with each inconsistency, which was also called a charge. Later, Goldstein et al. called these phase discontinuities "residues" in their paper [63] to make the analogy with the residues encountered in complex variable contour integration theory. The authors determined that each residue sign (or charge) must be balanced by imposing barriers (or branch cuts) that must not be crossed during the unwrapping process.



(a)



(b)

Figure 4.8 Itoh's 1-D phase unwrapping example. (a) A sinusoidal phase function with adequate sampling. Wrapped phase samples $\psi(n)$ are designated by blue asterisks. The unwrapped results are designated by red circles and they match the true phase exactly. (b) A sinusoidal phase function with insufficient sampling. Wrapped phase samples $\psi(n)$ are designated by blue asterisks. The unwrapped results are designated by red circles and they do not match the true phase. The differences between the true phase samples are greater than π. The distance between the 4^{th} and 5^{th} samples is shown as an example.

In [60], Ghiglia and Pritt elaborated on the analogy between the phase residues or discontinuities (jumps greater than π) and complex variable function residues. The residues of a complex function $f(z)$, where z is a complex variable $z = x + jy$, are the coefficients of the $(z - z_i)^{-1}$ terms of the Laurent series expansion of $f(z)$. For example $f(z) = \frac{1}{z(z-1)} = \frac{1}{z-1} - \frac{1}{z}$ has two residues +1 and -1 associated with the two poles at 1 and 0 respectively. The residue theorem for functions of a complex variable states that the closed contour integral is equivalent to the algebraic problem of evaluating the sum of the enclosed residues, as follows:

$$\oint f(z)\,dz = 2\pi j \sum_i r_i \qquad (28)$$

where r_i represents a residue that is enclosed by the closed contour.

For our function, $f(z) = \frac{1}{z(z-1)} = \frac{1}{z-1} - \frac{1}{z}$, the closed contour integral will be equal to zero if the contour does not enclose any of the two poles; the integral will be non-zero if it encloses one pole; and the integral will be zero again if it encloses the two poles since their associated residues (+1 and -1) will cancel one another.

By analogy, Ghiglia and Pritt developed a phase unwrapping theorem that states that the closed path integral of the phase derivatives is a function of the sum of the enclosed residue charges [60]:

$$\oint \nabla \varphi(r)\,dr = 2\pi\left(\sum sign\,(residue\,)\right) \qquad (29)$$

In other words, enclosing balanced residues in a region will result in a line integral around that region that is equal to zero for any simple path chosen. Thus, consistent phase unwrapping is

possible if, and only if, all integration paths do not encircle unbalanced residue charges. Once the residues are balanced by connecting opposite sign residues with branch cuts, the unwrapping can proceed along any path that does not cross branch cuts. This residue balancing concept constitutes the basis for a main class of phase unwrapping algorithm, which is the path-following unwrapping methodology. The pseudo-code for the residue detection for a 2-D phase image of size M×N can be summarized as follows:

1- Given 2-D wrapped phase data $\psi(m,n)$ for M columns by N rows phase image, compute the horizontal and vertical phase gradients (differences):

$\nabla^{x}_{m,n} = \psi(m+1,n) - \psi(m,n)$ and $\nabla^{y}_{m,n} = \psi(m,n+1) - \psi(m,n)$ for m=0,...,M-2 and n=0,..., N-2.

2- Compute the wrapped horizontal and vertical phase gradients:

$W(\nabla^{x}_{m,n}) = \tan^{-1}(\sin(\nabla^{x}_{m,n})/\cos(\nabla^{x}_{m,n}))$ and

$W(\nabla^{y}_{m,n}) = \tan^{-1}(\sin(\nabla^{y}_{m,n})/\cos(\nabla^{y}_{m,n}))$ for m=0,..., M-2 and n=0,..., N-2.

3- Compute the wrapped phase gradient integral for each 2×2 square in the phase image:

$S_{m,n} = W(\nabla^{x}_{m,n}) + W(\nabla^{y}_{m+1,n}) + (-W(\nabla^{x}_{m,n+1})) + (-W(\nabla^{y}_{m,n}))$ for m=0:M-2 and n=0:N-2

4- Test the integrals for phase residues (discontinuities):

if $S_{m,n} > 0$ then pixel (m,n) has a positive charge residue

else if $S_{m,n} < 0$ then pixel (m,n) has a negative charge residue

else pixel (m,n) has no residue.

Figure 4.9 Pseudo-code for the residue detection algorithm.

Figure 4.10 shows an example of the phase residues detection algorithm for color Doppler ultrasound images. Figure 4.10a shows a zoomed color Doppler image. Figure

4.10b shows the resultant residue image where positive residues are shown in red while negative residues are shown in blue. Figure 4.10c zooms into a single negative residue, showing the corresponding 2×2 square used to detect the residue; the original phase values (mapped between -128:127) are shown. Figure 4.10d shows a scaled version of the 2×2 square where the phase values are mapped to the interval $[-\pi:\pi)$; this step also shows the scheme of calculating the phase gradient in a clockwise fashion. Figure 4.10e shows the difference operators as defined in equation (25). Figure 4.10f shows the wrapping operator, as defined in equation (23), applied to the phase gradient values. For example, $\nabla_2 = 1.3\pi$ is outside the principal phase range $[-\pi:\pi)$; the wrapping operator subtracts -2π from ∇_2 to bring it within the principal phase range ($W(\nabla_2) = -0.7\pi$). Finally Figure 4.10g shows the integral (summation) of all wrapped differences, which is a non-zero negative value indicating a negative residue in the upper-left location of the 2×2 square under investigation.

To clarify the previous point, we will show two examples of phase unwrapping using Itoh's classical method [60]. We extended Itoh's 1-D algorithm to handle 2-D phase images by unwrapping along the first row of the image using the upper most left pixel as initialization then unwrapping along each column by using pixels in the first unwrapped row as initialization. Figure 4.11 shows two examples of wrapped color Doppler images. Figure 4.11a shows a case of minor wrapping where only a few pixels are wrapped. Figure 4.11c shows a case of severe wrapping with many more wrapped pixels. Applying the residue detection algorithm, as described as pseudo-code above, on the two wrapped images resulted in no detected residues for the simply wrapped image and a few residues for the severely wrapped image. The detected residues of Figure 4.11c image are shown in Figure 4.11e.

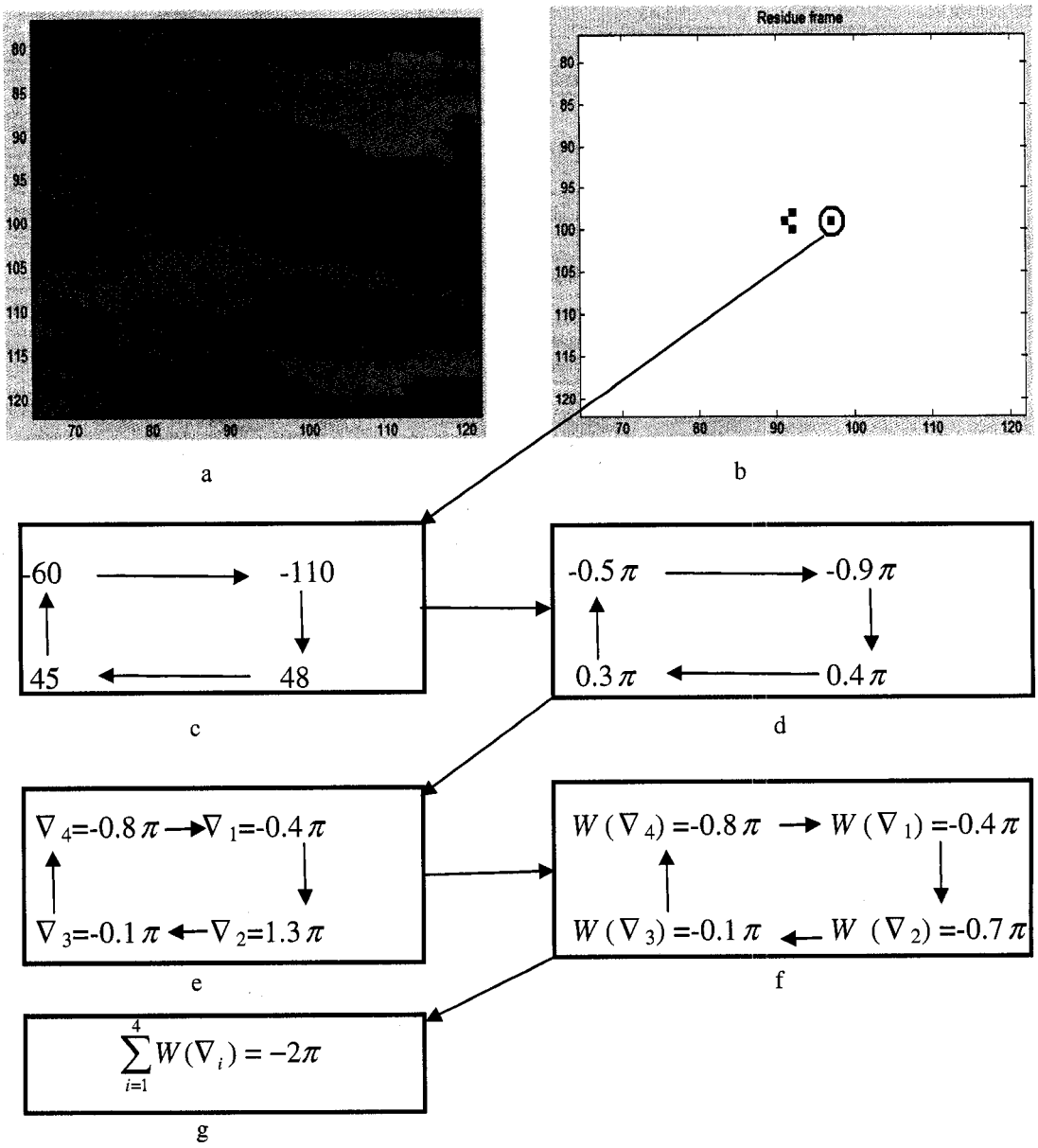


Figure 4.10 An example of phase residue detection. (a) A zoomed color Doppler image. (b) The corresponding detected residues. Positive residues are in red, while negative residues are in blue. (c) The original signed byte phase data of a 2×2 square that corresponds to a negative residue. (d) The same 2×2 square scaled to [$-\pi : \pi$). (e) The difference operators applied to the 2×2 square. (f) The wrapping operator applied to the difference operators. (g) The sum of the wrapped difference values is a non-zero negative value, which indicates a negative residue.

Applying Itoh's phase unwrapping algorithm on both images resulted in the two unwrapped images displayed in Figure 4.11b and d respectively. The unwrapping result shown in Figure 4.11b was successful as all wrapped (red) pixels are transformed into unwrapped (light blue) pixels with high phase values. It is worth noting that the used color map covers the phase range of $[-\pi : \pi)$ or [-128:127). All the unwrapped pixels outside this range are clipped in Figure 4.11b. Figure 4.11d shows unwrapping errors as vertical streaks extending from the residues in the direction of the integration paths. This simple experiment proved that Itoh's phase unwrapping algorithm cannot handle phase data with regions of discontinuity, as manifested by the residues. In the next section, we will review the most relevant techniques developed in the literature to handle such cases with phase discontinuity.

4.3. Phase Unwrapping Literature Review

As described in the previous section, phase unwrapping is path dependent when phase residues (or discontinuities) exist in the phase image. Residues can be easily detected by calculating the integral of the wrapped phase derivatives for loops of 2×2 squares. A non-zero value indicates a residue.

In the literature, there are two main classes of phase unwrapping techniques, most of which were developed for SAR images and a few of which were developed for MRI images. The first class of techniques is called "*path-following*" methods which use localized pixel-by-pixel operations to unwrap the phase. The main idea behind these methods is to guide the integration of the phase gradients so as to avoid any error propagation by avoiding crossing the boundaries between areas of phase discontinuity. The unwrapping errors will be introduced when the integration path crosses two successive pixels with a true phase difference greater than π. The error will propagate along the rest of the integration path and it will affect all the following pixels. This explains the streaks in Figure 4.11d above. The other class of unwrapping techniques is called "*minimum normalization*" methods which minimize a global measure of the differences between the gradients of the wrapped input phase and those of the unwrapped solution.

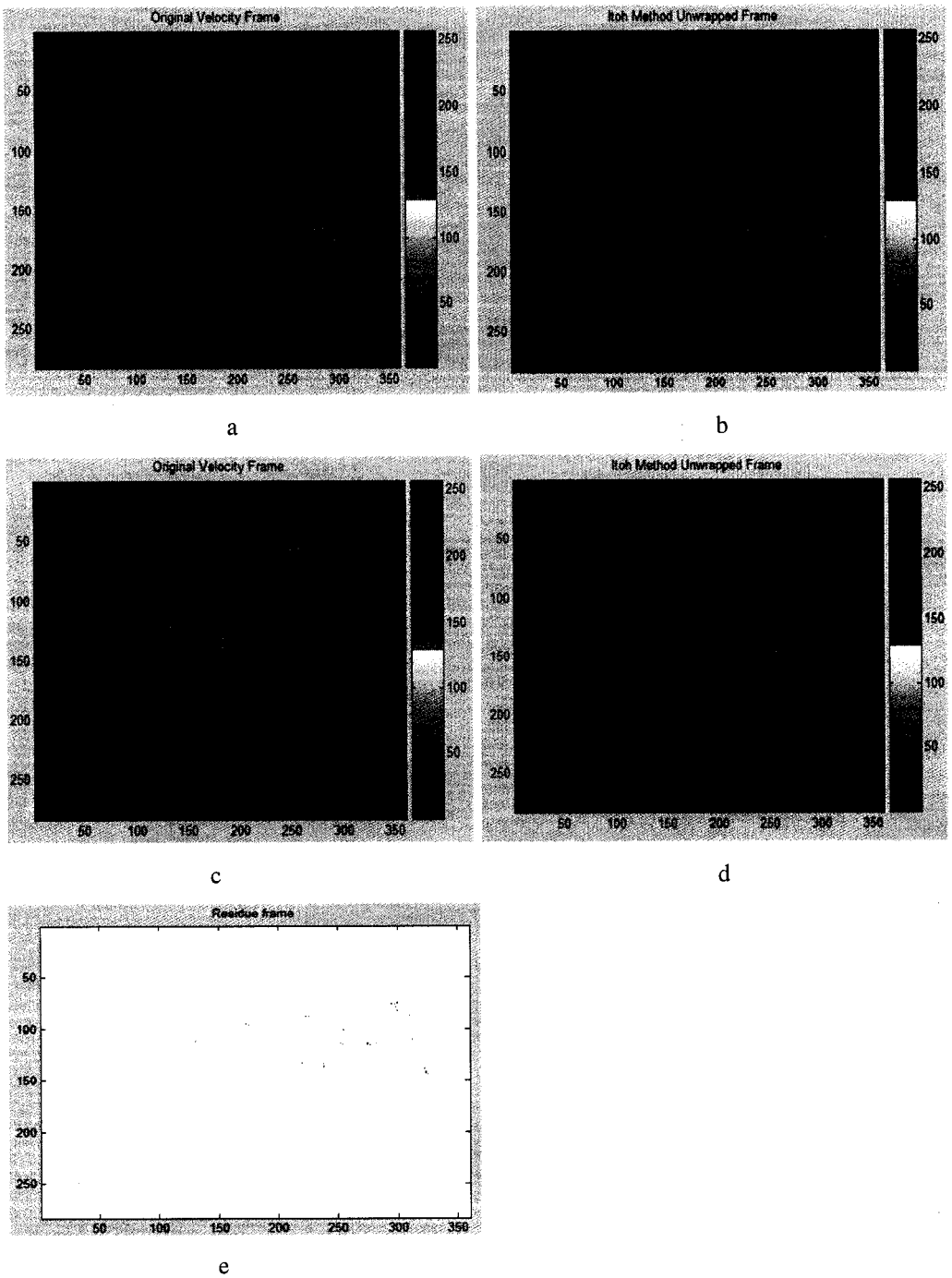


Figure 4.11 Phase unwrapping with Itoh's classical method. (a) An aliased color Doppler image with no residues. (b) Successful unwrapping result of (a) using Itoh's method. (c) An aliased color Doppler image with residues. (d) Unsuccessful unwrapping result of (c) using Itoh's method. (e) Residues detected from (c).

4.3.1. Path-following Unwrapping Methods

The main idea behind path-following methods is to place cut-lines (branch cuts) that connect opposite-sign phase residues to avoid path-dependent unwrapping results. The phase derivative integral avoids crossing these branch cuts. Instead it proceeds around the branch cuts until the whole phase image is covered. This guarantees that any closed path integral is zero. Thus the resulting unwrapped phase map is consistent independent of the starting pixel or the followed path, according to the residues theorem in equation (29) above. Figure 4.12 shows a demonstration of the path-following concept. Figure 4.12a shows a simulation of a phase image with detected positive and negative residues. Figure 4.12 b and c show how branch cuts can connect each pair of opposite-sign residues to balance the residues using two different configurations. Now the phase unwrapping can proceed using the integration of the wrapped phase derivatives, using Itoh's method, as indicated by the recursive mechanism in equation (24) above.

The criterion to place branch cuts is not obvious and is the core research topic behind all path-following methods. One obvious criterion is to avoid creating closed regions (islands) with branch cuts (as shown in Figure 4.12b), since the phase information cannot propagate across branch cuts into these isolated regions. Thus phase unwrapping may be partly or completely erroneous within these regions. Another obvious criterion is the length of the branch cuts, which should be minimized as much as possible to avoid disconnecting large regions within the phase image (as shown in Figure 4.12c).

One class of path-following methods uses only the location of the residues and their signs to develop a criterion to connect the branch cuts. The classical path-following method was developed by Goldstein et al. [63]. The algorithm is fast and produces short branch cuts. It uses a nearest-neighbor heuristic to find a configuration of the branch cuts, the sum of whose lengths is nearly minimal. The algorithm scans the phase image searching for a residue. When a residue is found, a small neighborhood (3×3) around that residue is searched for other residues. If another residue is found then it is connected to the residue in the center of the 3×3

neighborhood, forming a branch cut. If the two connected residues have opposite signs then the branch cut is marked as balanced and the search continues for a new residue. If the two residues have the same sign then the search for other residues continues with larger neighborhood windows (5×5, 7×7 ...etc) until the branch cut is balanced or the image border is reached. The resultant branch cuts connect either a pair or clumps of pairs of opposite-sign residues which guarantees consistent unwrapping results.

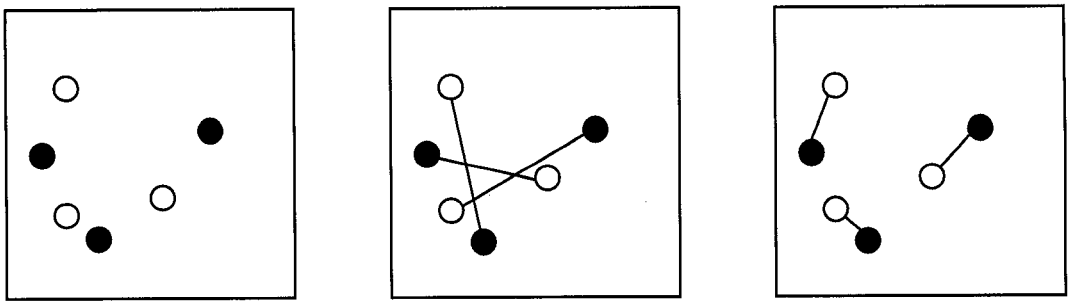


Figure 4.12 Path following phase unwrapping concept. (a) Phase image with positive residues (white circles) and negative ones (black circles). (b) Non-optimal configuration of balancing the residues using branch cuts. (c) Optimal configuration of balancing the residues using branch cuts.

Other path-following techniques restrict the branch cut configuration so that each branch cut connects only a single pair of opposite-sign residues (dipoles), in contrast to Goldstein's algorithm which may result in branch cuts that connect several pairs of opposite-sign residues. Cusack et al. [64] investigated several techniques for finding the optimal configurations of the dipole cuts, including two nearest-neighbor algorithms, a stable-marriage approach, and a simulated annealing approach. Buckland et al. [65] used a graph-theoretic approach called "the Hungarian algorithm" to find the configuration of dipole cuts that minimizes the sum of the length of cuts.

Another class of path-following algorithms tries to extract more information than the residues alone from the phase image, in order to guide the unwrapping process. Before we describe some of these algorithms we will first define concepts such as "*quality maps*" and "*quality masks*" that are extracted from the phase image and used to guide the unwrapping. *Quality maps* are arrays of values that define the quality or goodness of the phase data. Quality

maps can be derived from the original signal from which the phase data was derived. For example, in Synthetic Aperture Radar (SAR) imaging, the magnitude of the complex signal or some kind of correlation coefficients from that signal is used to construct a quality map. Pixels with high magnitude or high correlation coefficient are marked as high quality phase values, and vice versa. Other quality maps can be derived directly from the phase data. Ghiglia and Pritt [60] reviewed the most common quality maps used in the phase unwrapping literature. We review some of them here.

The pseudo-correlation quality map calculates the quality value at each pixel from the wrapped phase values ψ of a $k \times k$ neighborhood of that pixel as follows:

$$q_{m,n} = \frac{\sqrt{(\sum_i \sum_j \cos(\psi_{i,j}))^2 + (\sum_i \sum_j \sin(\psi_{i,j}))^2}}{k^2} \tag{30}$$

The pseudo-correlation calculation represents a vector sum of a small neighborhood of phase values. Highly correlated phase areas will have large vector sum values, while random phase areas, mainly noise regions, will have small correlation values. One drawback of the pseudo-correlation quality map is that it under estimates the quality values within highly varying phase regions, which may not be necessarily noise regions. The phase derivative variance quality map is defined as follows:

$$q_{m,n} = -\frac{\sqrt{\sum_i \sum_j ((\nabla^x_{i,j} - \overline{\nabla}^x_{m,n})^2 + (\nabla^y_{i,j} - \overline{\nabla}^y_{m,n})^2)}}{k^2} \tag{31}$$

The map is defined in a $k \times k$ neighborhood of each phase pixel. $\nabla^x_{i,j}$ and $\nabla^y_{i,j}$ are the phase horizontal and vertical partial derivatives (wrapped phase differences). $\overline{\nabla}^x_{m,n}$ and $\overline{\nabla}^y_{m,n}$ are the averages of the partial derivatives in the $k \times k$ window. The phase derivative variance

must be negated to reflect a quality, not a penalty, map. The max phase gradient quality map is defined as follows:

$$q_{m,n} = -\max(\max(|\nabla^{x}_{i,j}|), \max(|\nabla^{y}_{i,j}|)) \qquad (32)$$

The idea again is that noisy image regions will show high phase derivative values.

Quality masks are binarized versions of the quality maps. A threshold value is chosen either empirically or automatically to binarize the quality map. The quality mask is then used to mask out those phase pixels that are below the quality threshold. The unwrapping algorithm that uses a quality mask simply ignores those masked out phase pixels during the unwrapping and focuses only on the phase pixels with valid quality values.

Quality-guided path-following techniques usually do not detect residues nor construct branch cuts explicitly. Instead they use the quality map to guide the unwrapping. Bone was the first researcher to adopt a quality mask concept [66]. He used the second derivative of the phase data followed by a thresholding step to obtain a mask of high quality phase data to unwrap. Lim et al. [67] extended the technique by using a variable threshold that increases as the unwrapping progresses.

Another class of path-following techniques mixes branch cut detection algorithms and quality-guided techniques. The main idea is to use low-quality path to guide the placement of the branch cuts. Flynn [68] proposed a region-growing technique that starts with the residues and grows mask-cuts through low-quality map values. The resultant masks were thick and had to be thinned using morphological operators.

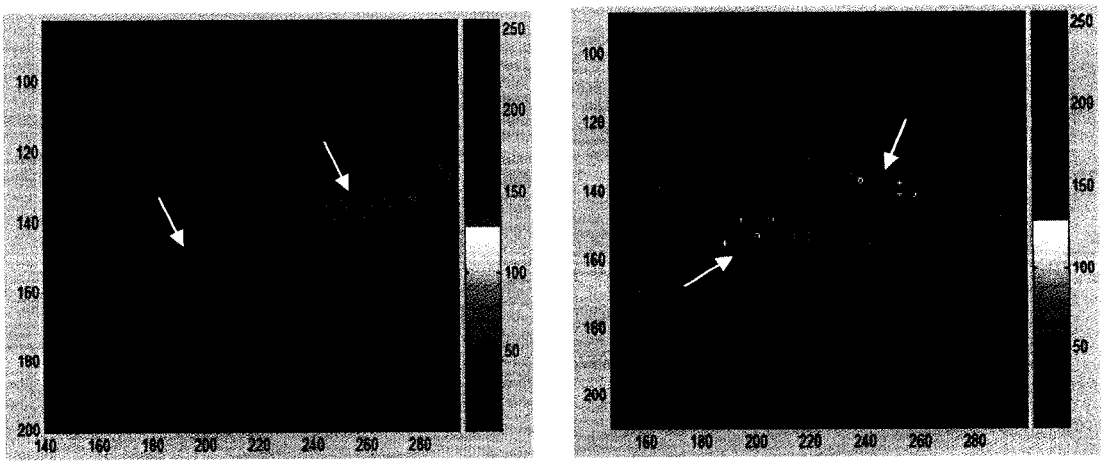
Quality-guided algorithms work well when good quality maps exist or can be derived from the phase data. One concern with some quality-guided algorithms that do not explicitly detect residues is that there is no guarantee that the integration path will avoid enclosing unbalanced residues. In this case error propagation will occur.

Another class of path-following algorithms relies on the detection of fringelines. *Fringelines* are borderlines between two adjacent pixels where phase wrapping occurs since it denotes a jump of 2π in the wrapped phase data. Figure 4.13 shows two examples of fringelines in color Doppler images. The first example shows a color Doppler image with simple aliasing where the true phase is continuous everywhere and no residues exist. The fringelines are the contours that separate the positive (blue) from the negative velocity data. In the case of simple aliasing, the fringelines tend to be closed contours that enclose regions of wrapped phase, as shown in Figure 4.13a. If the fringeline intersects with the image boundary (or the vessel wall in case of ultrasound) then we can think of it as closed by the image or vessel boundary. The second example shows a color Doppler image with severe aliasing where phase discontinuity exists and is indicated by the detected residues. In this case, the fringelines tend to be open contours that start and end at different residues, as shown in Figure 4.13b.

Phase unwrapping is then viewed as the process of assigning an integer multiple of 2π to each region delineated by a fringeline to minimize the discontinuities across the fringelines. The idea looks simple but in reality fringeline detection may be challenging due to noisy data or the existence of residues which will make the fringelines open-ended and it becomes hard to delineate wrapped phase regions.

Despite these problems with the fringeline detection approach for phase unwrapping, Flynn developed a successful and robust algorithm based on this idea [69]. His algorithm finds the unwrapped surface that is congruent to the wrapped phase and whose discontinuities are minimal in some sense. The algorithm utilizes a tree-growing approach that traces paths of discontinuities in the intermediate unwrapped surface, detects the paths that form loops, and adds multiples of 2π to the phase values enclosed by the loops to reduce the number of discontinuities. It performs this process iteratively until no more loops are detected. The process is guaranteed to converge to an optimal solution. The discontinuity is defined as a pair of adjoining pixels whose difference exceeds π in magnitude. The algorithm is considered a path-following technique in the sense that it follows the closed discontinuity (fringe) lines

within the image to unwrap the enclosed region at once. Flynn's algorithm can also use quality map values as weight factors to restrict discontinuities to low quality regions.



(a) (b)

Figure 4.13 Fringelines represent boundaries between wrapped pixels where a phase jump of 2π occurs. (a) A zoomed color Doppler image with simple wrapping with no residues. The closed contours between the positive (blue) and negative (red) pixels represent fringelines. (b) A zoomed color Doppler image with severe aliasing where residues exist. The fringelines become open contours.

All the previously reviewed algorithms are considered path-following techniques since they all generate phase unwrapping paths in the image by means of localized, pixel-by-pixel operations, such as nearest neighbor search, region growing, or fringeline tracking. However they differ in the way they generate the integration path and the way they avoid the crossing of phase discontinuity regions to prevent error propagation.

4.3.2. Minimum-Norm Unwrapping Methods

Another class of algorithms for phase unwrapping applies mathematical techniques to find the unwrapped solution that fits the given phase data on a global, rather than local, basis. Such algorithms are called "*minimum-norm*" methods. Specifically, these techniques seek the unwrapped phase whose local derivatives match the measured derivatives with "minimal error." Minimal error can be defined in different ways; the most common one is the least

squares error. Least squares error notion is formally a minimum norm in the L^2 sense. For phase unwrapping, that means the sum of the squared differences between the gradients of the solution and those of the wrapped phase data is minimized. The minimum norm techniques can also be considered surface fitting techniques where the solution surface has a gradient that matches the original wrapped gradients.

To set the stage for the L^p-norm phase unwrapping technique, we introduce the concept for a simple line-fitting problem where we are given a number of data points and would like to fit a line that best describes all these points. The best fit will be the line that minimizes the error defined as:

$$\varepsilon^p = \sum | y_i - (ax_i + b) |^p \qquad (33)$$

where (x_i, y_i) are the given data points, and a, b are the line slope and vertical axis intersection respectively. If $(p = 2)$ then this is the common least squares error case; if $(p = 1)$ then this is the minimum absolute error case.

For phase unwrapping, the minimum L^p-norm solution must minimize the error:

$$\varepsilon^p = \sum\sum | \phi_{i+1,j} - \phi_{i,j} - \nabla^x_{i,j} |^p + | \phi_{i,j+1} - \phi_{i,j} - \nabla^y_{i,j} |^p \qquad (34)$$

where $\nabla^x_{i,j}$ and $\nabla^y_{i,j}$ are the wrapped phase horizontal and vertical derivatives respectively. Weights can be added to the derivative errors to solve a weighted minimum L^p-norm problem. Weights can be derived from quality maps or masks to ignore regions of noise-corrupted phase. In [60], Ghiglia and Pritt derived the mathematical equations for solving the minimum L^p-norm problem and showed that the solution is a nonlinear PDE given by:

$$\frac{\partial}{\partial x}[U(x,y)(\phi_x - \psi_x)] + \frac{\partial}{\partial y}[V(x,y)(\phi_y - \psi_y)] = 0 \qquad (35)$$

where ϕ_x and ϕ_y are the solution's unwrapped phase derivatives. ψ_x and ψ_y are the input wrapped phase derivatives. $U(x,y)$ and $V(x,y)$ are weight factors dependent on both the input and output phase derivatives and are defined as:

$$V(x,y) = |\phi_y - \psi_y|^{p-2} \qquad U(x,y) = |\phi_x - \psi_x|^{p-2} \tag{36}$$

For the special case of the least squares error, where $p = 2$, $U(x,y)$ and $V(x,y)$ are simply unity functions, and the PDE solution equation is simplified to be Poisson's equation, given by:

$$\nabla^2\phi = \rho = \psi_{xx} + \psi_{yy} \tag{37}$$

where ψ_{xx} and ψ_{yy} are the second derivatives of the input phase data, ρ is the sum of second derivatives, and ∇^2 is the Laplacian operator. The classical solution of Poisson's equation is the Gauss-Seidel relaxation method defined by:

$$\phi_{i,j} = \frac{(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) - \rho_{i,j}}{4} \tag{38}$$

This process is iterated until convergence, which can be slow. Other transform-based methods, such as the FFT or DCT, are more efficient for solving Poisson's equation [70], [71].

The transform-based solutions require periodic functions. Thus the sum of input second derivatives function ρ of size $(M \times N)$, defined in equation (37), is reflected about its last row and column to construct the periodic function $\tilde{\rho}$ of size $(2M \times 2N)$, as follows:

$$\tilde{\rho}_{i,j} = \begin{cases} \rho_{i,j} & 0 \le i \le M, 0 \le j \le N \\ \rho_{2M-i,j} & M < i < 2M, 0 \le j \le N \\ \rho_{i,2N-j} & 0 \le i \le M, N < j < 2N \\ \rho_{2M-i,2N-j} & M < i < 2M, N < j < 2N \end{cases} \tag{39}$$

The resulting FFT solution of Poisson's equation is given by:

$$\Phi_{m,n} = \frac{P_{m,n}}{2\cos(\pi m / M) + 2\cos(\pi n / N) - 4} \tag{40}$$

where $\Phi_{m,n}$ and $P_{m,n}$ are the 2-D Fourier transforms, of size $(2M \times 2N)$, of the periodic unwrapped solution $\tilde{\phi}_{i,j}$ and the periodic input derivatives sum function $\tilde{\rho}_{i,j}$. The final unwrapped solution $\phi_{i,j}$ is obtained from the inverse Fourier transform of $\Phi_{m,n}$, restricted to the first quadrant interval $(M \times N)$. DCT methods do not require mirror reflection of the data and thus are slightly more efficient algorithms in terms of memory utilization.

The multigrid algorithm is another efficient technique to rapidly solve PDE equations on large grids [71]. The idea behind the multigrid technique is to construct a pyramid of decreasing resolution grids, on which the classical Gauss-Seidel relaxation algorithm can be applied. The convergence rate of the multigrid technique is much faster than Gauss-Seidel relaxation, which works only on the highest resolution grid.

All unweighted least-squares phase unwrapping approaches give the same result with the same input image. But they are all not suitable for phase data with discontinuity (residues) since they integrate through the residues. The problem is manifested by the tendency of the unweighted least-squares techniques to underestimate the local, average phase slope when residues are present [60].

Weighted least-squares phase unwrapping techniques, on the other hand, use weights to avoid integrating through the residues. Weights can be provided by quality maps or masks to deemphasize regions of noise, measurement errors, aliasing, or phase discontinuity. The weighted least-squares problem can be described as the minimization of the following error:

$$\varepsilon^2 = \sum\sum U_{i,j} \mid \phi_{i+1,j} - \phi_{i,j} - \Delta^x_{i,j} \mid^2 + V_{i,j} \mid \phi_{i,j+1} - \phi_{i,j} - \Delta^y_{i,j} \mid^2 \qquad (41)$$

where $U_{i,j}$ and $V_{i,j}$ are functions of the user-provided weight array $w_{i,j}$, defined as follows:

$$V_{i,j} = \min(w^2_{i,j+1}, w^2_{i,j}) \qquad \text{and} \qquad U_{i,j} = \min(w^2_{i+1,j}, w^2_{i,j}) \qquad (42)$$

A matrix description was derived in [60] to formulate the weighted least-squares problem as a set of linear equations, which was used in many solutions. The weighted-least squares problem cannot be solved using transform methods, such as FFT or DCT, but many iterative solutions have been proposed in the literature to solve it. *Preconditioned Conjugate Gradient (PCG)* is one such method [70]. Conjugate gradient methods iteratively solve a set of sparse linear equations, such as Poisson's equation. The problem is formulated as a minimization problem and the conjugate gradient approach converges faster than other classical minimization techniques. However the convergence rate depends on the condition number of the underlying matrix. The closer the matrix is to the identity matrix, the faster the convergence rate will be. The PCG method applies a preconditioning step that transforms the underlying matrix to one near the identity matrix. An important post-processing operation is needed for all least-squares unwrapping algorithms. Unlike path-following methods, least-squares algorithms do not produce an unwrapped surface that is congruent to the wrapped phase. That is re-wrapping the unwrapped surface does not result in a match to the original wrapped input phase. The post-processing step is called a "congruence operation" and is described in detail in [73].

Another solution to the weighted least-squares phase unwrapping problem is the weighted multigrid algorithm [71]. Similar to the unweighted multigrid technique, the weighted multigrid algorithm uses a pyramid of decreasing resolution grids, while the phase weights are restricted to the coarser grids. The convergence of this technique is much faster than the PCG algorithm.

Ghiglia and Romero proposed a generalization of the weighted least-squares phase unwrapping [72]. They proposed an algorithm that iterates on the PCG algorithm to find a solution whose gradients agree with those of the wrapped phase data in the sense of a minimum L^p norm, The most useful value of the parameter (p) appears to be (p=0), which causes the gradients of the solution to differ from those of the wrapped phase data in as few places as possible. Because the algorithm generates its own data-dependent weights (see equations (35) and (36)), it works with or without a quality map. However the algorithm requires intensive computation due to its double iterative approach (it iterates on PCG, which in turn iterates on FFT or DCT unweighted methods).

4.3.3. Unwrapping Methods Comparison

Comparative analysis papers have appeared in the literature to evaluate a number of phase unwrapping methods in terms of execution time, memory requirements, and success rate. Mark Pritt evaluated eight phase unwrapping algorithms [74], four of which are path-following methods and the other four of which are minimum-norm methods. He tested the algorithms on eight example images that represent a wide spectrum of phase unwrapping problems. He concluded that the FFT/DCT method is the fastest and requires minimum memory; Goldstein's algorithm comes next with fast execution and low memory requirement as well. The L^p-norm method is the slowest and requires the largest memory. In terms of results evaluation, Flynn's technique and the L^p-norm method were the most successful algorithms, while the FFT/DCT method was the least successful one.

Ghiglia and Pritt evaluated the same eight algorithms in their book [60] that Pritt evaluated in his paper, but with more thorough analysis. They recommended starting with Goldstein's algorithm for a new phase unwrapping problem, and leaving Flynn's and L^p-norm algorithms as last resorts because of their heavy computational demand. In case there is a good quality map available, they recommended trying the quality-guided, PCG, mask cut, or weighted multigrid algorithm. The authors also showed some links between the two main classes for phase unwrapping, path-following and min. norm, which could explain obtaining similar results on some data.

4.3.4. True Phase Cutline Detection

The path-following methods presented earlier attempt to construct branch cuts that balance opposite sign residues to avoid encircling unbalanced number of residues by the phase derivatives integration paths. However all these methods utilize little information from the phase image to guide the placement of the branch cuts, such as the location of the residues or the quality maps derived from the phase data. These approaches seek "*consistent*" unwrapping results, but not necessarily "*correct*" results. Consistent results imply reaching the same unwrapped values no matter what unwrapping path is used. Correct results, on the other hand, would be harder to evaluate since they will depend on the problem domain.

Recently, some publications tried to achieve "correct" results for specific phase unwrapping domains, such as MRI. Chavez et al. [75] presented a thorough analysis of the phase unwrapping problem for MRI phase maps and proposed a new path-following technique to obtain correct unwrapping results. Before describing their technique, we will introduce the core concept behind it, which is the detection of the true discontinuity borderlines. In [76], Kramer and Loffeld described an algorithm to detect the lines that separate regions of discontinuous phase, which they called cutlines. The authors outlined their terminology by defining some important concepts in phase unwrapping problems. *Fringelines* are curves where jumps of $\pm 2\pi$ occur in the wrapped phase data. These are easy to see in the phase image as abrupt transition from the maximum positive to maximum negative phase or vice

versa. *Cutlines* are borderlines between adjacent pixels where the modulus of the true phase variation gets larger than π. The left image in Figure 4.14 highlights the imaginary fringelines and cutlines of a color Doppler image. These are the true unwrapped phase discontinuity curves that get lost due to the wrapping effect in the wrapped phase image. The only detectable traces of the cutlines in the wrapped phase image are the residues representing the endpoints of the cutlines. Fringelines tend to be closed if there are no residues present. When residues exist then fringelines become open, or in fact closed by the hidden cutlines. Residues represent intersection points between the visible fringelines and the hidden cutlines. Branch cuts can be thought of as approximations of the true cutlines, which may prevent error propagation and achieve consistent phase unwrapping results but will not guarantee correct results if they do not exactly match the true cutlines.

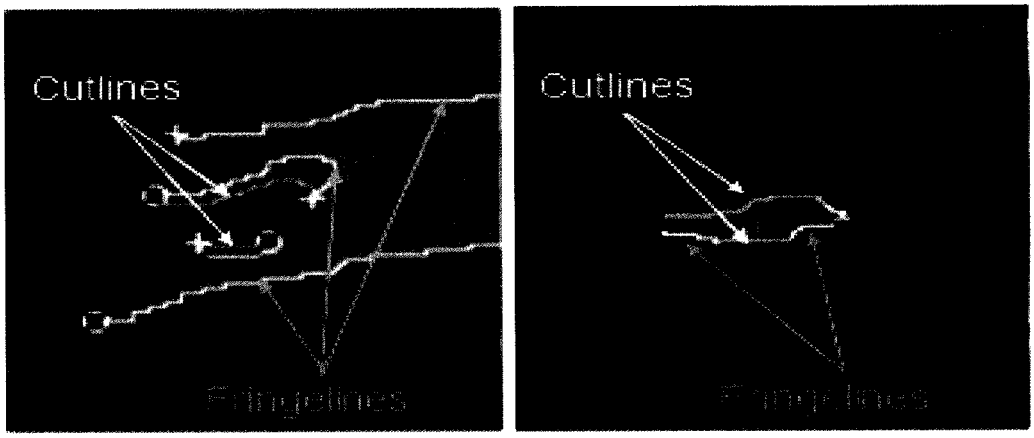


Figure 4.14 Left: fringelines and cutlines highlighted on a color Doppler ultrasound image. Right: the shifted fringelines and the same cutlines highlighted on the same color Doppler image that has been phase-shifted.

Kramer and Loffeld proposed a method for detecting large portions of the true cutlines, but not necessarily the complete connected lines. Their idea was based on the fact that shifting the wrapped phase data by a certain angle will not alter the position of the cutlines (including their endpoints or residues) since they represent phase discontinuity in the true phase, which will be maintained irrespective of any phase shift. However the fringelines will change with the phase shift since they represent phase discontinuity in the wrapped phase data. The idea is explained on the right image of Figure 4.14, where a phase shift equal to π is

applied to the color Doppler image on the left; the new fringelines and cutlines are highlighted on the phase shifted image. It is clear that the fringelines have shifted due to phase-shifting the image itself, while the cutlines are still the same. Starting with the endpoint (residue), which belongs to both the fringeline and the cutline, the fringeline will follow the cutline before leaving it then meeting with the cutline again on the closing end. If it is possible to identify these common points between the fringelines and cutlines in every phase shifted image then it would be possible to detect many portions of the hidden cutline. Their algorithm was based on this simple idea and started with detecting the residues, and then generating N phase shifted images from the original wrapped phase image. In every phase shifted image, the fringeline associated with every residue is detected. The corresponding cutline associated with each residue is calculated as the union of the intersection of each two successive fringelines in two phase shifted images. More precisely, let $A_{n,i}$ be the set of all points belonging to the fringeline i in the n^{th} shifted image. The corresponding cutline is found as the union of intersections of fringes as follows:

$$cutline_i = \bigcup_{n=0}^{N-1} [A_{n,i} \bigcap A_{n+1,i}] \tag{43}$$

The fringeline tracking procedure described by Kramer and Loffeld deserves some explanation. It starts with detecting the residue points similar to Goldstein's method in [63], where each 2×2 square in the phase image is checked according to the following formula:

$$S(x,y) = W(\phi_2 - \phi_1) + W(\phi_3 - \phi_2) + W(\phi_4 - \phi_3) + W(\phi_1 - \phi_4) \tag{44}$$

The wrapping operator W is defined in equation (23) above. The wrapped phase values ϕ_i lie in the range $[-\pi, \pi)$ and hence the phase differences lie in the range $[-2\pi, 2\pi)$. Thus the wrapping operator in case of a fringeline (a $\pm 2\pi$ jump in the wrapped phase) that exists between the two pixels (ϕ_i, ϕ_j) can be replaced by an addition or subtraction of 2π to bring the difference back into the wrapping range $[-\pi, \pi)$, as shown below:

$$W(\phi_i - \phi_j) = \phi_i - \phi_j - 2\pi\, sign(\phi_i - \phi_j) = \phi_i - \phi_j + F(\phi_i - \phi_j) \tag{45}$$

where $F(\phi_i - \phi_j)$ is an operator defined as follows:

$$F(\phi_i - \phi_j) = \begin{cases} -2\pi\, sign(\phi_i - \phi_i) & \text{if there is a fringeline between } \phi_i \text{ and } \phi_j \\ 0 & \text{if there is no fringeline between } \phi_i \text{ and } \phi_j \end{cases} \tag{46}$$

Substituting equation (45) and (46) into (44) gives:

$$S(x, y) = F(\phi_2 - \phi_1) + F(\phi_3 - \phi_2) + F(\phi_4 - \phi_3) + F(\phi_1 - \phi_4) \tag{47}$$

In this case, if the 2×2 square does not intersect with any fringeline, then $S(x, y)$ will be equal to zero since all terms in equation (47) will be equal to zero. If the fringeline starts or ends at the center of the 2×2 square under consideration, then $S(x, y)$ will be equal to $\pm\pi$, as shown in Figure 4.15a, since three terms will be equal to zero and only one term where the fringeline passes will be non-zero. If the fringeline passes through the 2×2 square, as shown in Figure 4.15b, then it has to enter the square between two pixels and exit from another direction. The two fringeline segments intersecting with the square will balance each other and $S(x, y)$ will again be equal to zero. This explains why we are only able to identify the open fringeline endpoints (residues), which also belong to the invisible cutline, while the rest of the cutline is not easily detectable.

To detect the rest of the fringeline, let's assume we know the entry point of the fringeline into the 2×2 square, e.g. between pixels (ϕ_2, ϕ_3) as shown in Figure 4.15b. Since the square does not contain any endpoints, we know that $S(x, y) = 0$. Since we know the entry point of the fringeline, $S(x, y)$ becomes:

$$S(x,y) = W(\phi_5 - \phi_2) + W(\phi_6 - \phi_5) + W(\phi_3 - \phi_6) + \phi_2 - \phi_3 - 2\pi\, sign(\phi_2 - \phi_3) \qquad (48)$$

Let

$$\tilde{S}(x,y) = S(x,y) + 2\pi\, sign(\phi_2 - \phi_3) = F(\phi_5 - \phi_2) + F(\phi_6 - \phi_5) + F(\phi_3 - \phi_6) \qquad (49)$$

Now one of the three terms in equation (49) has a non-zero value, since the fringeline has to exit the 2×2 square from another point (it is $F(\phi_3 - \phi_6)$ in Figure 4.15b), and thus non-zero value for $\tilde{S}(x,y)$ should indicate the extension of the fringeline. Similar formulas can be derived for the other three horizontal and vertical 2×2 squares adjacent to the residue's 2×2 square, as shown in squares in Figure 4.15c. Checking the four adjacent squares should reveal the extension of the fringeline with the non-zero value for $\tilde{S}(x,y)$. This process is repeated iteratively to track the fringeline until its other endpoint.

One drawback of Kramer's cutline detection method is that the detected cutlines are incomplete, and not sufficient to unwrap the phase data without propagating unwrapping errors. The authors suggested using the magnitude data for SAR images to guide the filling of the gaps of the cutlines. Another drawback is that the number of phase shifted images N is critical. Choosing very small N results in very sparse cutline pieces, while choosing very large N results in too many false cutlines pieces. False cutlines may result due to the limitation of the image spatial resolution. The fringelines of different phase shifted images may intersect due to quantization errors, even if the intersections do not belong to the true cutlines. One solution to mitigate this problem is to zoom the phase shifted images before tracking the fringelines to provide sufficient resolution and avoid false intersections, but this will increase the processing time.

The authors showed their results on only one simulation image of a phase ramp with two different noise levels. It was not clear how well this technique would do with real phase

images. However the idea was novel in the sense that it tried to detect the true discontinuity lines to achieve correct, in addition to consistent, phase unwrapping results.

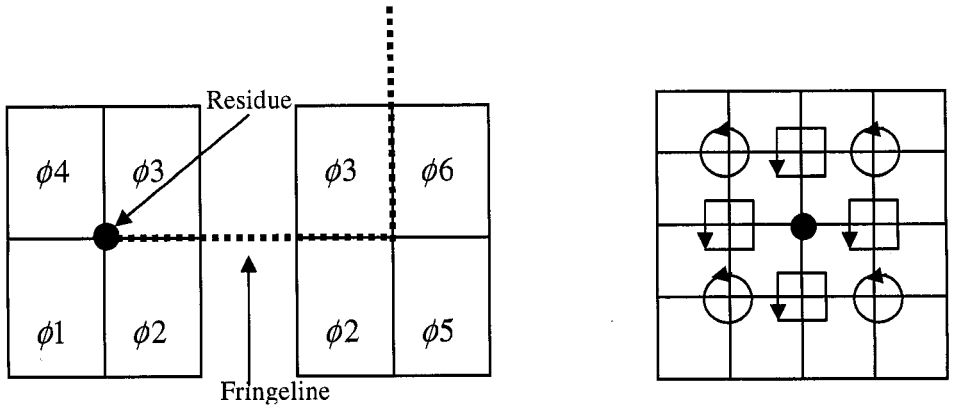


Figure 4.15 Kramer's fringeline tracking method. (a) A 2×2 square with a residue in the center. (b) The neighboring 2×2 square where the fringeline extends. (c) The 4 horizontal and vertical neighboring 2×2 squares are used by Kramer et al. to check for the fringeline extension. The 8 neighboring squares, including the diagonal ones, are used by Chavez et al. to detect fringelines.

Chavez et al. [75] proposed a similar idea to Kramer's cutline detection to unwrap magnetic resonance phase images. They proposed using a score map generated by tracking the phase-shifted fringelines which connect the dipoles (pairs of opposite sign residues). This score map is then eroded (thinned) so as to obtain the branch cuts used to unwrap the data. Their approach to track fringelines is different than the one proposed in [76].

Their fringeline tracking method relies on an interesting concept called the "*dipole annihilation*" process. The dipole annihilation process states that removal of the residues (i.e. replacement of the 2×2 squares pixels, associated with the residues, by zeros) alone is not sufficient to guarantee a streak-free phase unwrapping result. In fact, the removal of a residue's 2×2 square propagates the residue to one of the eight adjacent 2×2 squares. Consecutive removal of the newly generated residues creates adjacent residues along a fringeline path connecting a positive and negative residue until they superimpose and cancel

out. The resulting fringeline will be 2 pixels in thickness since it results from the consecutive removal of the residue's 2×2 squares.

The authors applied their fringeline detection process to several phase shifted images and relied on the same idea used by Kramer et al.: those parts of the fringeline connecting a given dipole (opposite sign residue pair) that do not vary much with phase offset are part of the cutline. This is due to the fact that the smaller the phase gradient is in a region, the more the phase offset will cause the fringeline to vary in that region. But instead of intersecting successive fringelines, they superimposed all fringelines and used the resultant image as a score map where the true cutlines will show up as ridges in this score map. The authors used a region-growing technique to track the ridges in the score map image followed by a thinning technique to prune the cutlines. Another enhancement described in [75] was the way they generated the phase-shifted fringelines. Instead of shifting the original phase image globally, they used a non-zero value for the annihilation process that represented the complement of the phase shift. For example, to detect a fringeline associated with a $\pi/2$ phase shifted image, the removed residues are replaced by a $-\pi/2$ value instead of zeros. The resulting fringeline is identical to the one produced by a global phase shift applied to the image and a zero value used for the annihilation process, but in a much shorter time due to saving the global image phase shift step.

The proposed method was shown to work successfully on one simulated image and three MRI images, which do not represent sufficient data to evaluate the algorithm. Also the computation complexity of the algorithm increases as the number of phase shifts increases. The number of phase shifts is by itself a critical factor for the performance of the technique, just like Kramer's method.

4.4. Previous Phase Unwrapping Techniques Performance with Ultrasound

In this section, we will evaluate some of the previously reviewed phase unwrapping techniques on color Doppler ultrasound images. The source code for these techniques was developed by Ghiglia and Pritt and accompanied their book [60]. However the code was designed for the UNIX operating system and we ported the code to the Microsoft Windows XP operating system. We integrated the algorithms with Matlab© scripts that handle input/output, data preprocessing, display, and statistical analysis routines. We have chosen several algorithms to evaluate with ultrasound images to assess strengths and weaknesses for a broad range of techniques, the evaluated algorithms are:

1. Goldstein's classical path-following algorithm [63]
2. Flynn's mask cut algorithm [68]
3. Flynn's minimum discontinuity algorithm [69]
4. Ghiglia's DCT algorithm [70]
5. Ghiglia's Preconditioned Conjugate Gradient (PCG) [70]

In addition, we also evaluated the most recent true cutline detection techniques developed for MRI images by Chavez et al. in [75] and the original cutline detection method of Kramer and Loffeld in [76]. Before we present the evaluation results we will describe the ultrasound data acquisition protocol and image types.

4.4.1. Color Doppler Ultrasound Image Acquisition Protocol

We used the Philips premium ultrasound equipment, the iU22 [4] for ultrasound data acquisition. The velocity data represents Doppler phase shift data, and thus it can be thought of as a phase map. Ultrasound systems usually display tissue images using a grayscale map where bright regions represent interfaces, such as vessel walls, between different tissue types as shown in Figure 4.2. Doppler blood flow images are usually displayed using a pseudo-color map that uses shades of red for positive Doppler values and shades of blue for negative values

(or vice versa, blue for positive and red for negative as shown in Figure 4.2). A zero velocity is indicated by a black color.

A number of signal path controls that are available for the ultrasound system users had to be fixed during all data acquisition experiments to guarantee consistent and valid data for the unwrapping algorithms. The most important control was the "*color persistence*" control, which determines how much temporal averaging is applied to color frames before display. Persistence is an aesthetics tool that enables the ultrasound operator to assess the pulsatility of the blood flow. We have verified experimentally that the "color persistence" corrupts the residues in the phase data, which makes the phase unwrapping task more difficult. Turning the "color persistence" off maintains the location and polarity of the phase residues. Another important parameter is the "*wall filter,*" which represents a high-pass filter that rejects the low-frequency high-amplitude vessel wall signal; it is also called the "*clutter filter.*" We used the minimum setting of the wall filter to minimize filtering out the low frequency wrapped phase data. Other less-important parameters had to be fixed as well to minimize data post-processing and preserve the raw phase information. The spatial "*smoothing*" was set to "low" to minimize spatial averaging of color Doppler data. The *Doppler steering angle* was also fixed during each experiment to a value that gave maximum Doppler sensitivity (by avoiding perpendicular Doppler angles).

Two types of data were used for the phase unwrapping evaluation, *in vitro* data and *in vivo* data. The *in vitro* data was obtained using an ultrasound flow phantom. A typical flow phantom consists of a pump and a tube that is immersed in a tank filled with tissue mimicking material. The tissue mimicking material has a speed of sound similar to the typical human body speed of sound. The pump circulates a blood-mimicking material inside the tube to simulate the blood flow. The phantom is programmable to simulate different Doppler waveforms, such as constant, saw-tooth, or sinusoidal waveforms, or even physiological waveforms such as carotid or femoral waveforms. The *in vivo* data represents blood flow of human arteries and veins. A skilled sonographer helped with some experiments by scanning a human model, and the datasets were captured according to the previously mentioned protocol. The rest of the *in vivo* datasets were self scanning of the first author for easily accessed

vessels, such as the carotid artery and the jugular vein. All experiments followed the same acquisition protocol described above.

The scale parameter, also called the PRF or pulse repetition frequency, is allowed to change throughout the data acquisition experiments. The PRF parameter controls the sampling rate of color Doppler images. If the PRF is high enough, then phase wrapping will not occur. Gradually lowering the PRF will introduce phase wrapping, or color Doppler aliasing, whose degree will increase with decreasing the PRF. During each experiment, the sampling rate started with a high setting that prevented wrapping altogether, and then the rate was gradually decreased to allow phase wrapping to increase. Very low sampling rates were avoided, since they introduce very noisy and corrupted phase images, and do not represent clinically useful cases.

Table 4.1 summarizes the *in vitro* datasets used to evaluate the different phase unwrapping algorithms. Different simulation waveforms (carotid, constant, and femoral waveforms with optimal Doppler steering angle, and femoral waveform with non-optimal steering angle) were captured. The "*Nyquist velocity*" column specifies the velocity beyond which phase wrapping occurs; it is an indication of the sampling rate. The maximum velocity column specifies the approximate maximum color Doppler velocity of the blood-mimicking flow under consideration. It indicates how much wrapping should occur when compared to the Nyquist velocity. It is important to note that the maximum velocity is not the true maximum velocity of the blood-mimicking flow inside the phantom tube, but it represents the color Doppler estimation of the projection of this true maximum velocity in the direction of the ultrasound system transducer. For example, the first dataset of the carotid waveform has a Nyquist velocity of 96.3 cm/s, while the maximum velocity output from the phantom is 67.3 cm/s. In this case, the phase data will not be wrapped since the maximum Doppler velocity is less than the Nyquist velocity. The fourth row of the carotid waveform table shows a maximum velocity of 67.4 cm/s, which is higher than the Nyquist velocity of 57.8 cm/s; phase wrapping will occur in this case.

Table 4.1 *In vitro* color Doppler ultrasound datasets using a flow phantom. (a) Carotid waveform. (b) Constant waveform. (c) Femoral waveform. (d) Femoral waveform with non-optimal Doppler steering.

Carotid Case #	Nyquist Velocity (cm/s)	Maximum Velocity (cm/s)
1	96.3	67.4
2	77.0	67.4
3	67.4	67.4
4	57.8	67.4
5	48.1	67.4
6	38.5	67.4
7	33.7	67.4
8	28.9	67.4
9	24.1	67.4
10	19.3	67.4

(a)

Constant Case #	Nyquist Velocity (cm/s)	Maximum Velocity (cm/s)
1	96.3	67.4
2	48.1	67.4
3	38.5	67.4
4	28.9	67.4

(b)

Femoral Case #	Nyquist Velocity (cm/s)	Maximum Velocity (cm/s)
1	96.3	67.4
2	48.1	67.4
3	38.5	67.4
4	28.9	67.4
5	19.3	67.4
6	12.0	67.4

(c)

Femoral Case #	Nyquist Velocity (cm/s)	Maximum Velocity (cm/s)
1	48.1	38.5
2	38.5	38.5
3	28.9	38.5
4	19.3	38.5
5	12.0	38.5

(d)

The following set of figures give examples of the captured cineloops for different sampling rates. Figure 4.16 shows a cineloop of 30 frames of a simulated carotid artery waveform with adequate sampling rate (Nyquist velocity=77.0 cm/s) compared to the maximum velocity present (67.4 cm/s), which occurs during the systole phase. The carotid flow is *pulsatile*, i.e. it fluctuates with the heart systole and diastole phases. This is evident

from the figure as the systole frames show maximum blood filling of the vessel lumen (e.g. frames 4 and 24), while diastole frames show poor filling (e.g. frames 8 and 18). The carotid flow is also *uniphasic*, i.e. the blood moves in one forward direction during the whole heart cycle and never moves backward. This is evident from the figure as all the frames show blue color indicating the blood is flowing toward the transducer. Figure 4.17 shows the same carotid simulation flow but with lower sampling rate (Nyquist velocity=48.1 cm/s), which causes Doppler aliasing or phase wrapping to occur, as evident in frames (2-6). Figure 4.18 shows a constant simulated flow with adequate sampling rate (Nyquist velocity=96.3 cm/s). It is evident that the constant flow is non-pulsatile and uniphasic. Figure 4.19 shows a constant simulated flow with inadequate sampling rate (Nyquist velocity=28.9 cm/s), which causes phase wrapping in all frames.

Figure 4.20 shows a femoral simulated flow with adequate sampling rate (Nyquist velocity=96.3 cm/s) where no aliasing is present. The femoral flow is pulsatile and *triphasic*, where the blood flows in both the forward and reverse directions during the systole and diastole phases respectively. This is evident in the data as some frames are blue and others are red. Figure 4.21 shows inadequate sampling rate (Nyquist velocity=19.3 cm/s) for the femoral waveform; phase wrapping is evident in both directions.

Figure 4.22 also shows femoral waveform with adequate sampling but with non-optimal steering angle (almost perpendicular Doppler angle). Non-optimal steering causes confusion in the blood direction interpretation by the ultrasound system, which manifests itself by mixed red and blue color blobs within the same frame. Also the mean velocity is under estimated since the cosine of the Doppler angle is almost zero for perpendicular angles, which explains the dimmer color Doppler images with non-optimal Doppler steering. Figure 4.23 shows inadequate sampling rate with the non-optimal steering case for the femoral flow. Although this case is not common, since sonographers usually avoid non-optimal steering angles, it still represents the most challenging case for ultrasound phase unwrapping.

Figure 4.24, Figure 4.25, Figure 4.26, and Figure 4.27 show *in vivo* datasets for a human carotid artery and jugular vein, for both unaliased and aliased cases. It is evident from

these figures that the *in vivo* data is more complicated than the *in vitro* phantom data because of the more complex human body and the additional artifacts that exist, such as the body motion artifact. In the following sections we will demonstrate the behavior of some classical phase unwrapping techniques with color Doppler ultrasound images. The pros and cons of each algorithm will be mentioned.

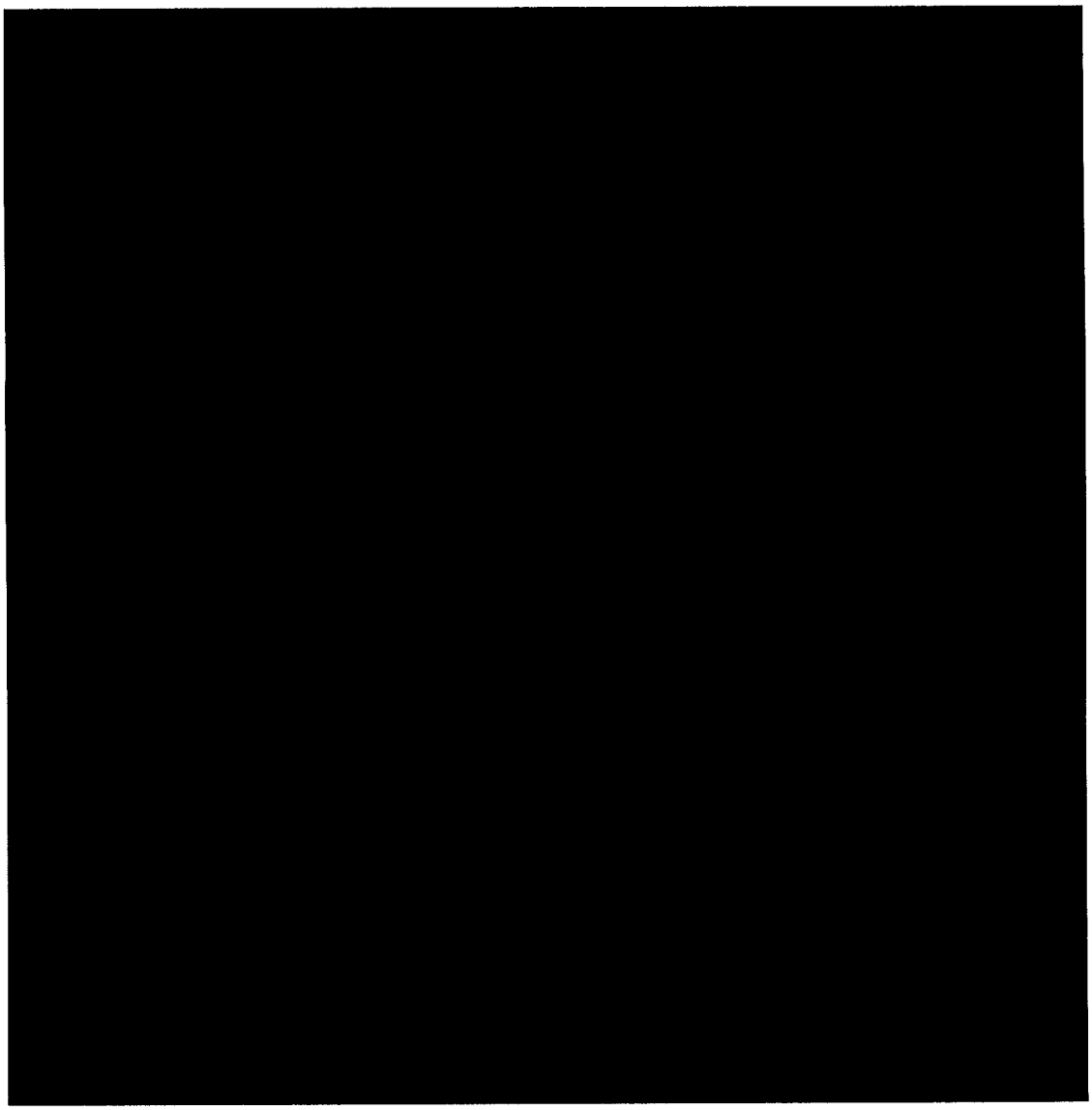


Figure 4.16 Simulated carotid waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 77.0 cm/s, while the maximum velocity is 67.4 cm/s; phase wrapping is absent.

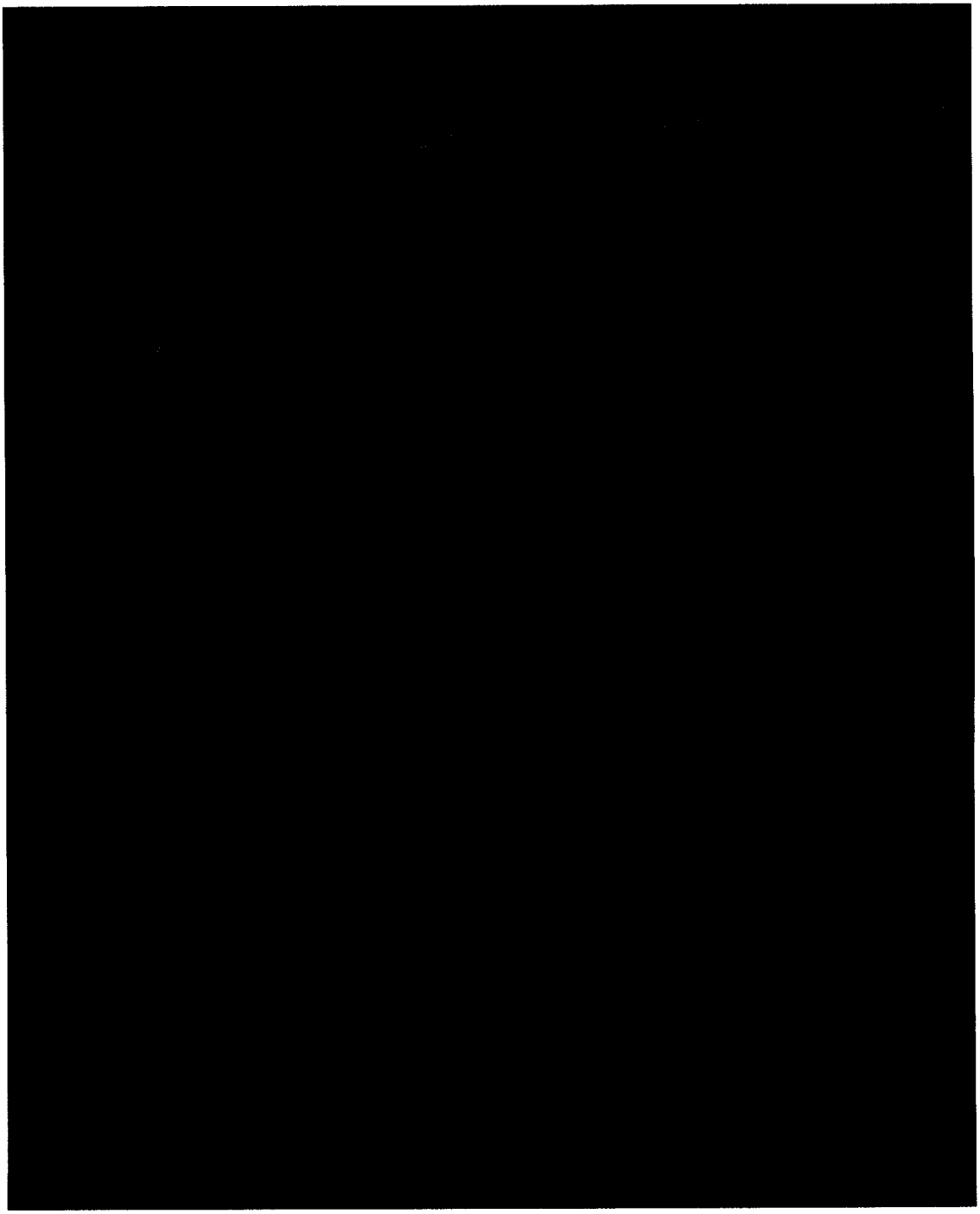


Figure 4.17 Simulated carotid waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 48.1 cm/s, while the maximum velocity is 67.4 cm/s; phase wrapping is present in some systole frames.

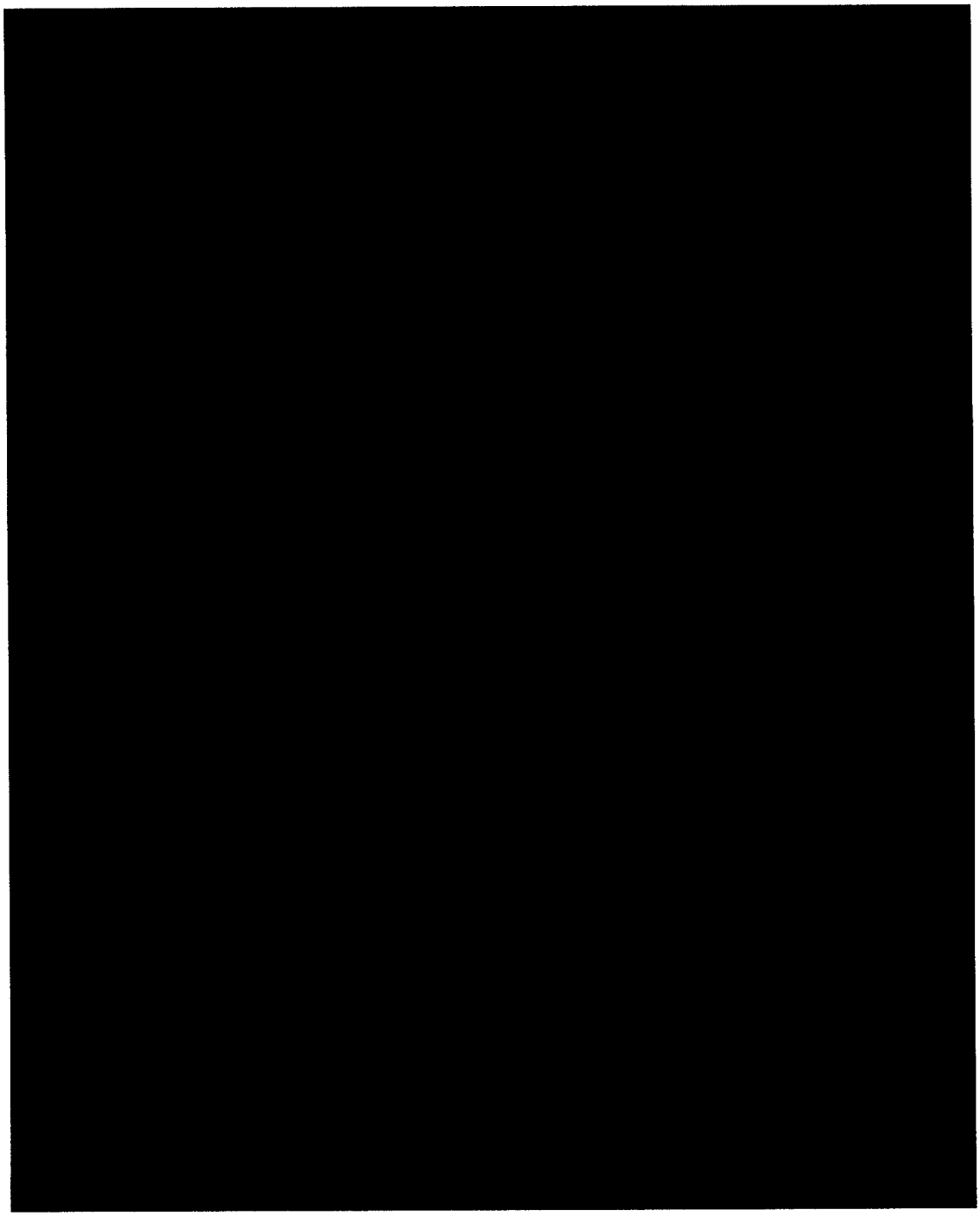


Figure 4.18 Simulated constant waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 96.3 cm/s, while the maximum velocity is 67.4 cm/s; phase wrapping is absent; constant flow is non-pulsatile and uniphasic.

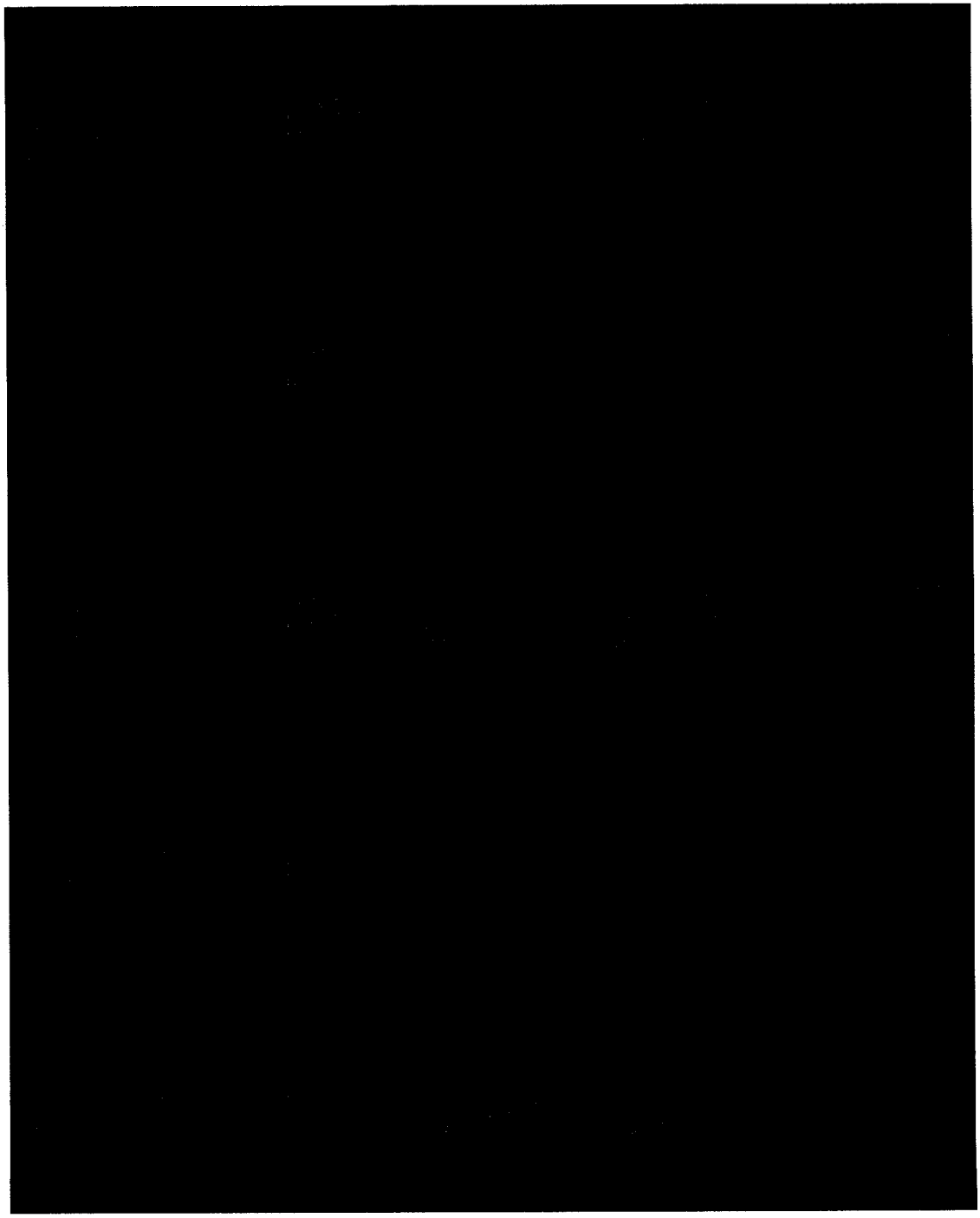


Figure 4.19 Simulated Constant waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 28.9 cm/s, while the maximum velocity is 67.4 cm/s; phase wrapping is present in all frames since constant flow is non-pulsatile.

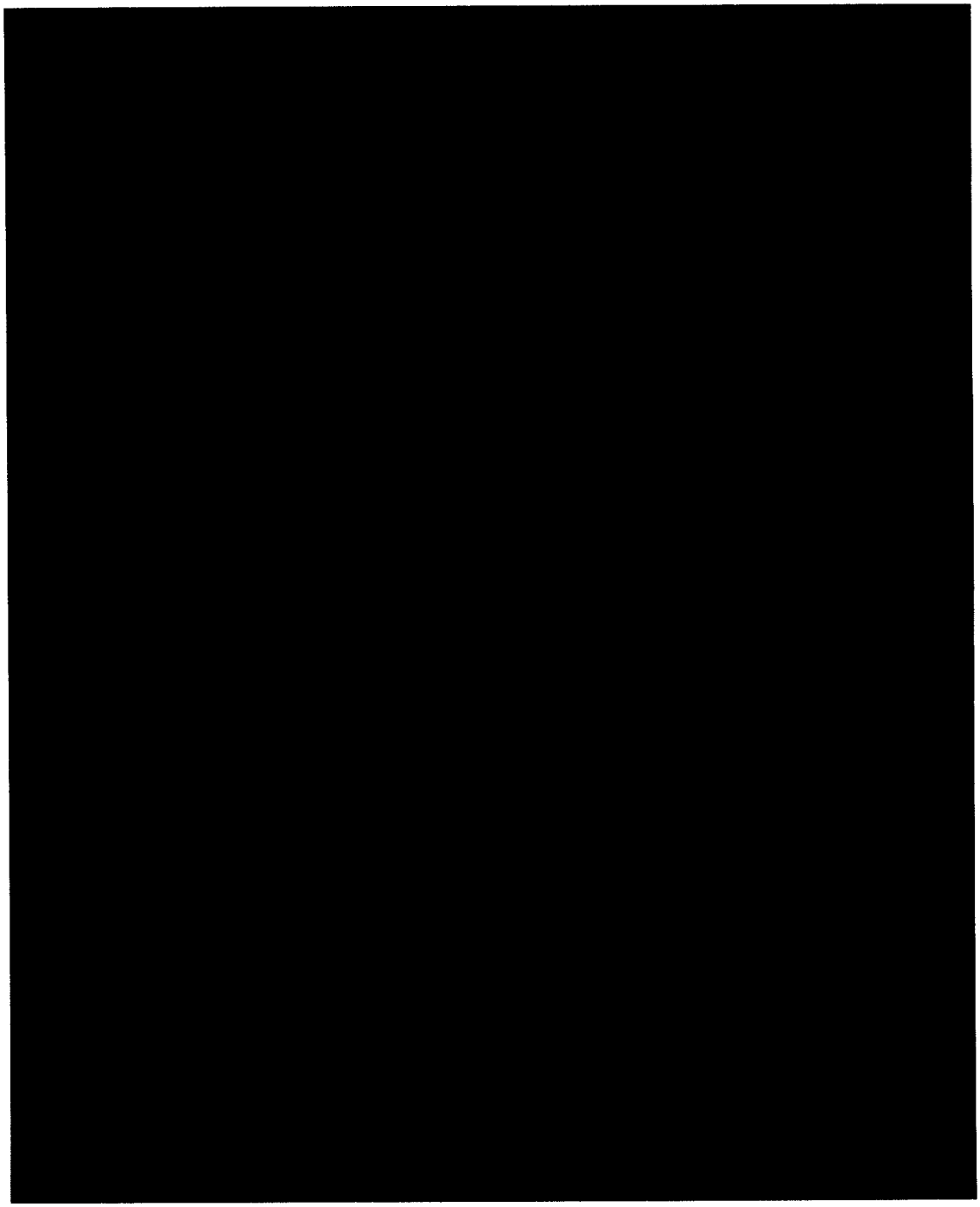


Figure 4.20 Simulated femoral waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 96.3 cm/s, while the maximum velocity is 67.4 cm/s; phase wrapping is absent. Femoral flow is pulsatile and triphasic.

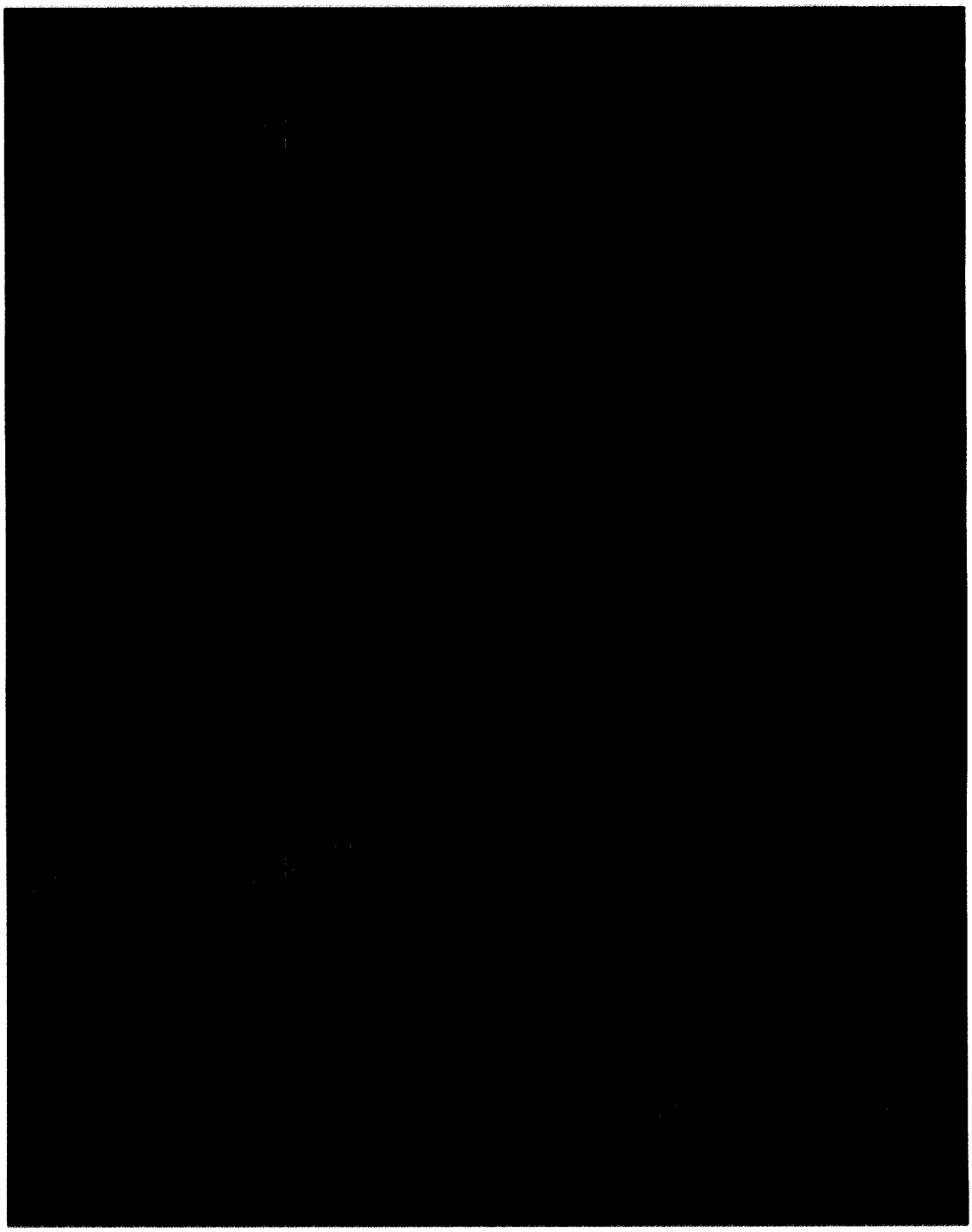


Figure 4.21 Simulated femoral waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 19.3 cm/s, while the maximum velocity is 67.4 cm/s; phase wrapping is present in some systole frames in both directions since femoral flow is triphasic.

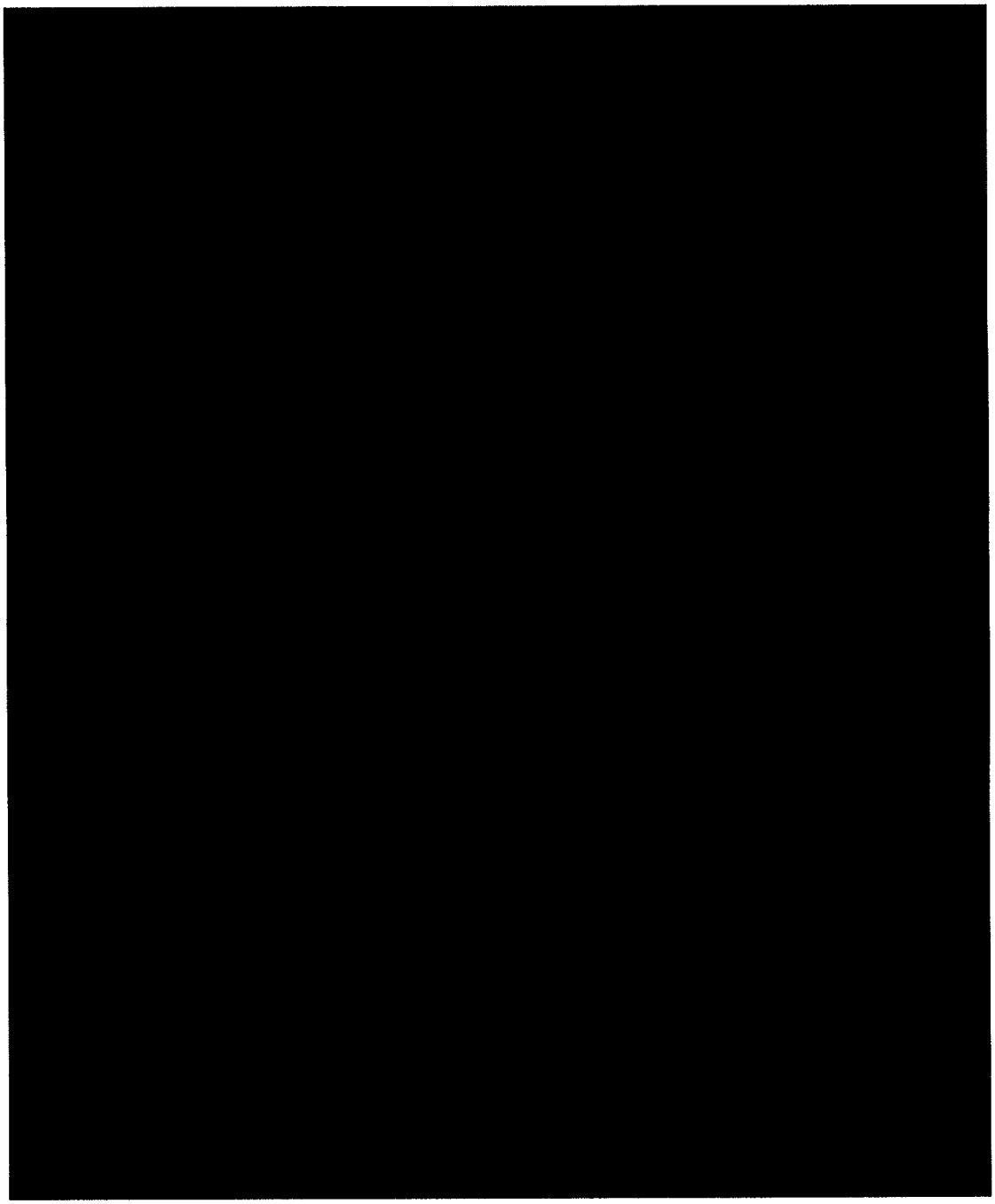


Figure 4.22 Simulated femoral waveform from a flow phantom with non-optimal Doppler steering. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 48.1 cm/s, while the maximum velocity is 38.5 cm/s; phase wrapping is absent. Non-optimal steering causes both forward and reverse flow to coexist in some frames.

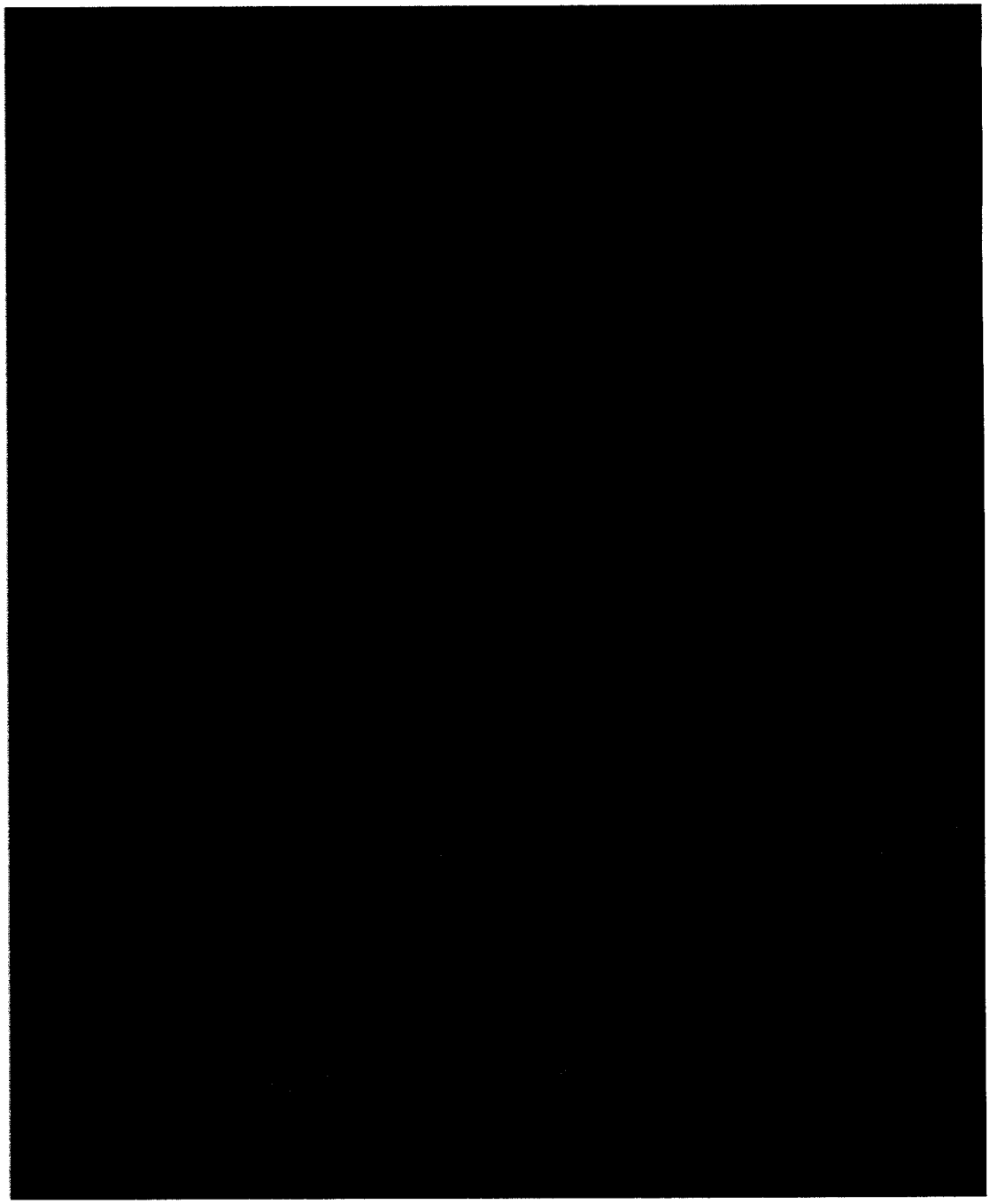


Figure 4.23 Simulated femoral waveform from a flow phantom with non-optimal Doppler steering. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 19.3 cm/s, while the maximum velocity is 38.5 cm/s; phase wrapping is present in some systole frames in both directions.

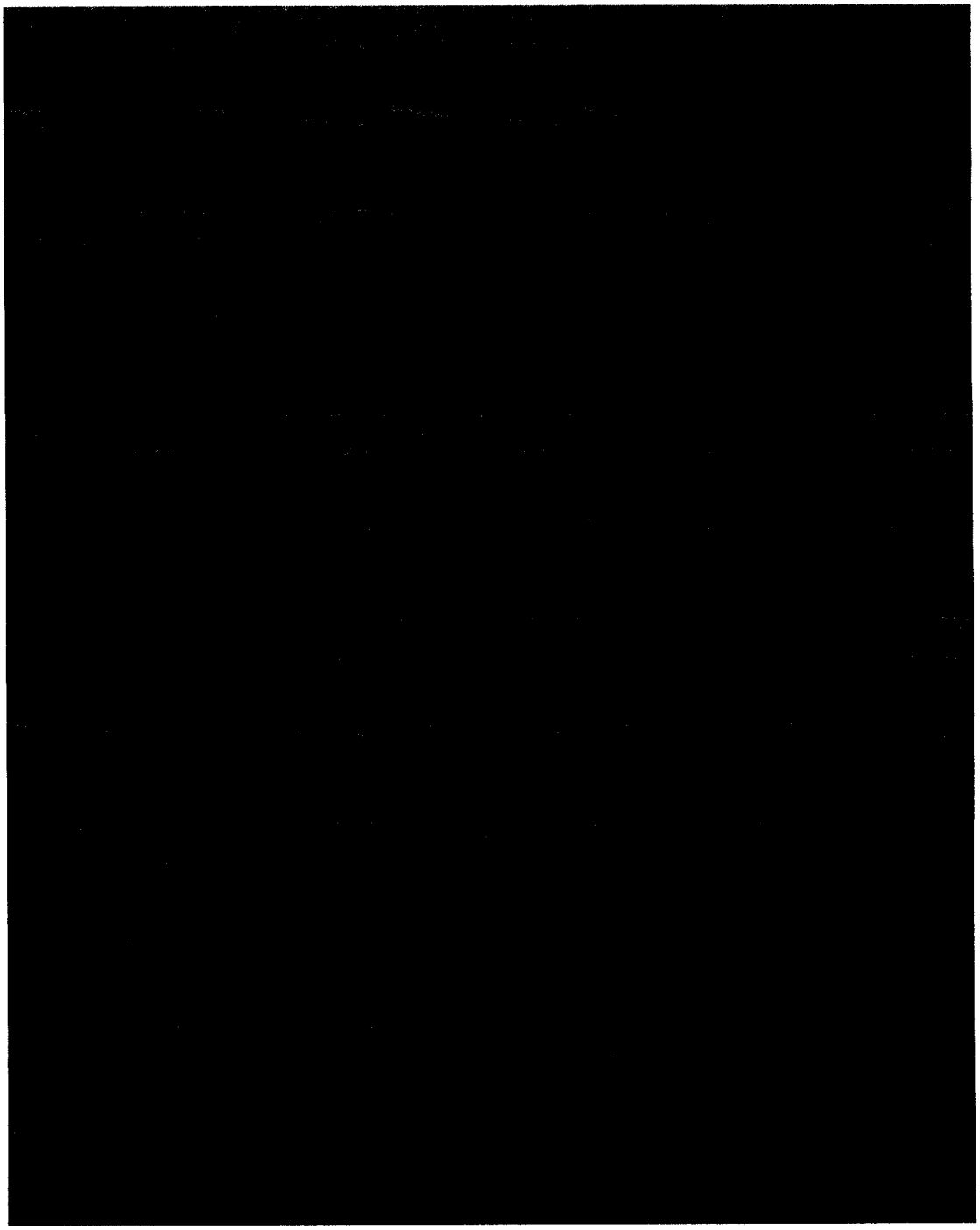


Figure 4.24 *In vivo* Common carotid Artery (CCA). A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 38.5 cm/s, which was adequate to avoid phase wrapping.

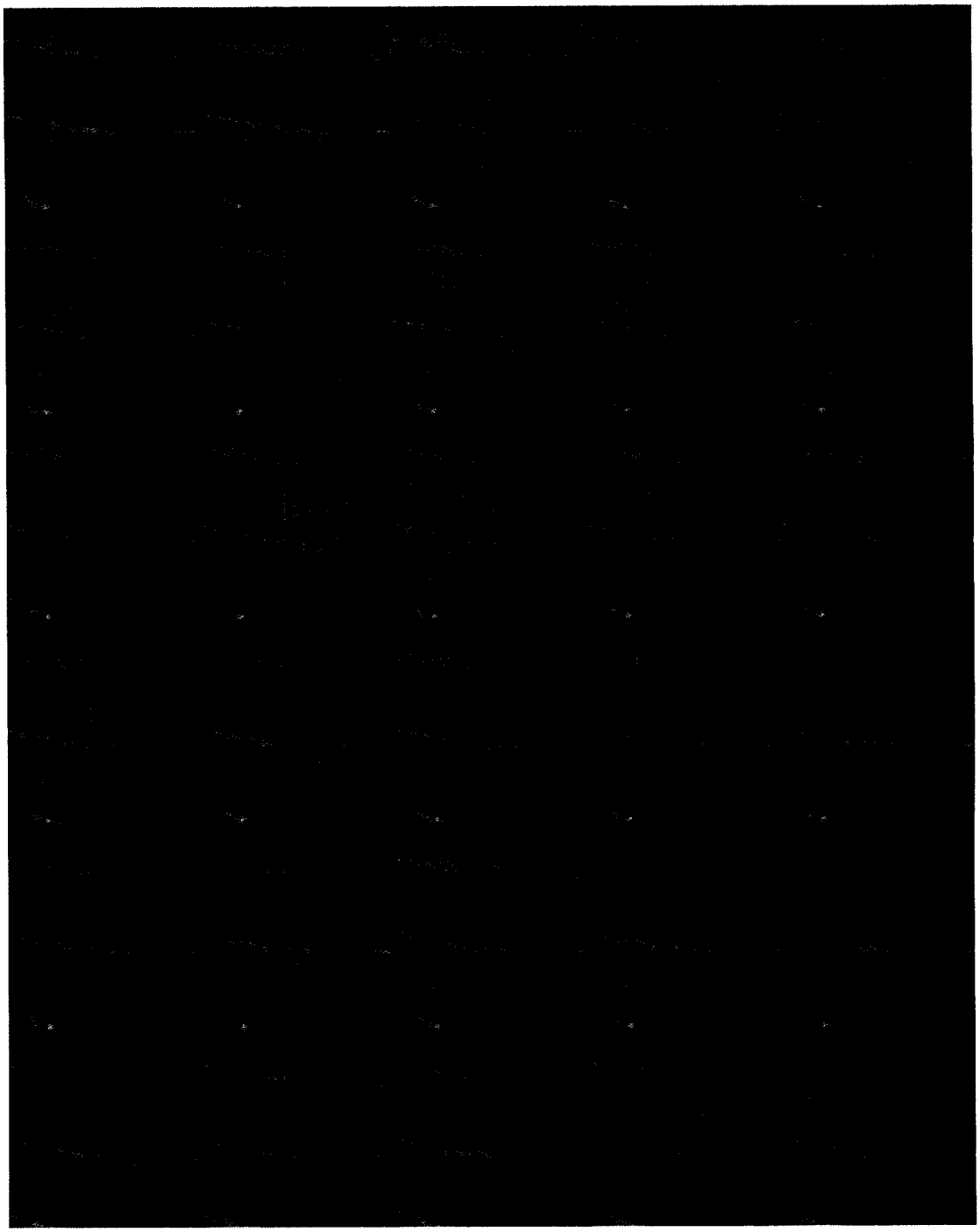


Figure 4.25 *In vivo* Common carotid Artery (CCA). A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 12 cm/s, which was inadequate and caused phase wrapping.

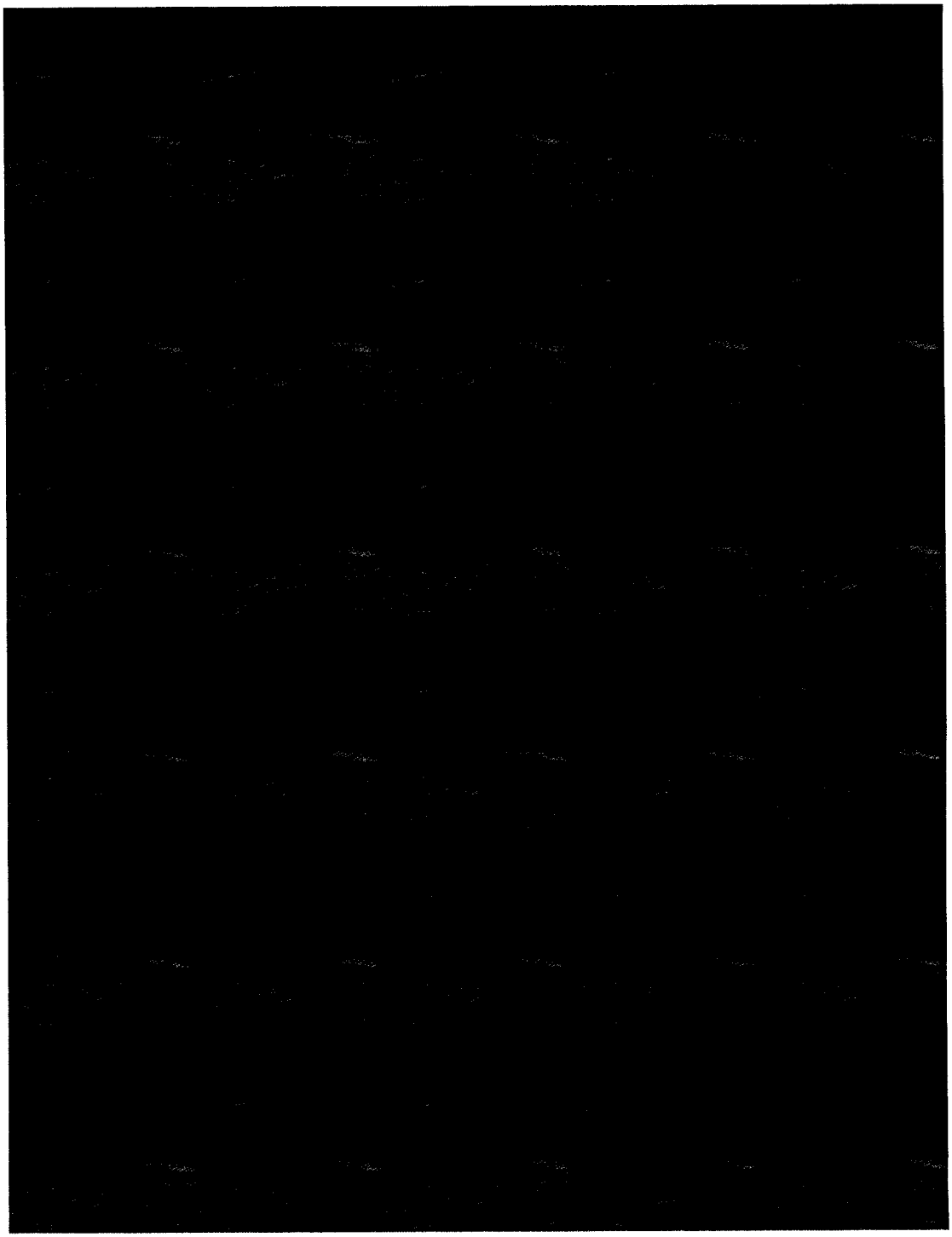


Figure 4.26 *In vivo* Jugular Vein (JV). A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 12 cm/s, which was adequate to avoid phase wrapping. The Jugular vein is pulsatile and uniphasic.

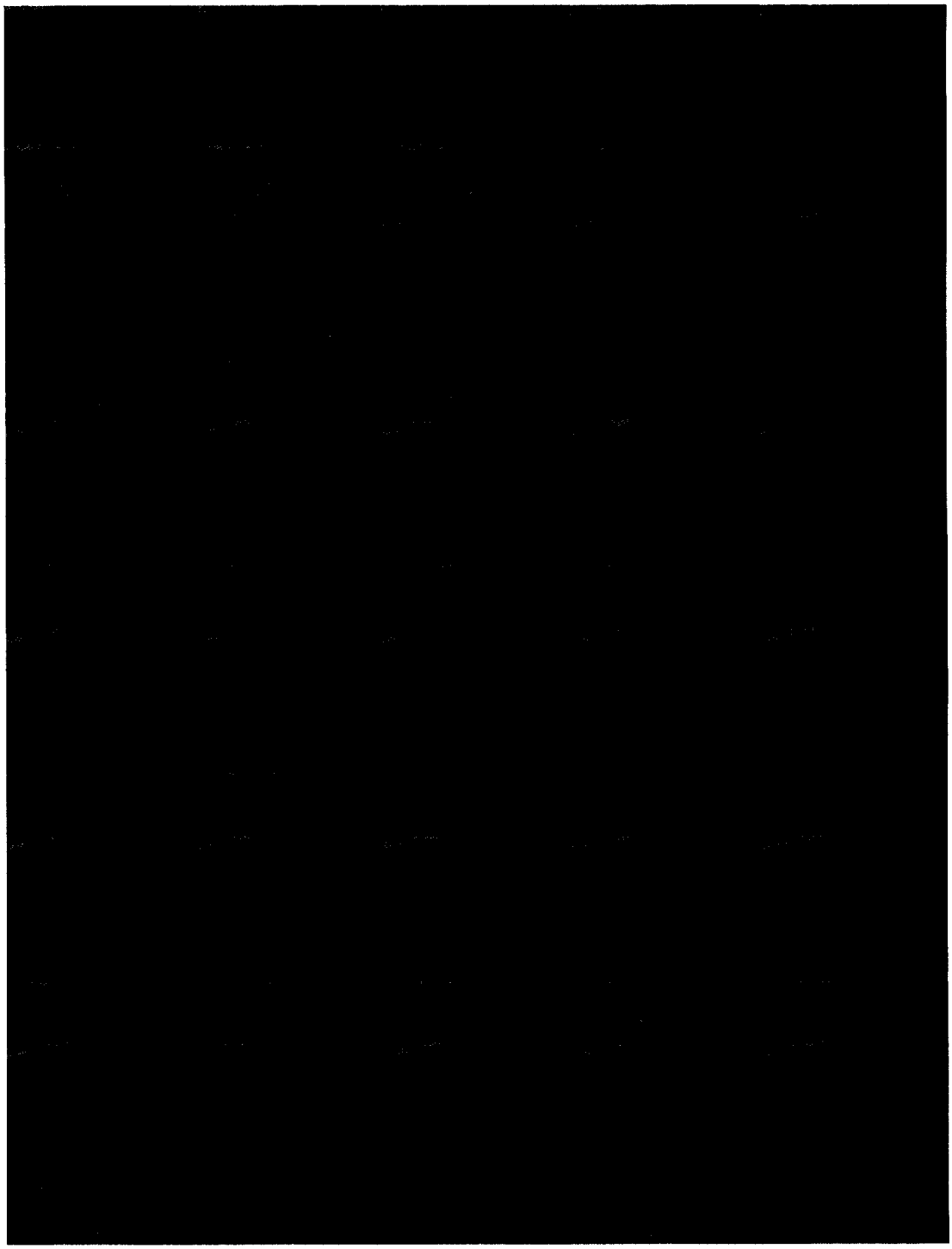


Figure 4.27 *In vivo* Jugular Vein (JV). A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. The Nyquist velocity is 4.8 cm/s, which was inadequate and caused phase wrapping.

4.4.2. Goldstein's Path-following Algorithm [63] Results

As described previously, Goldstein's path-following algorithm relies on a heuristic nearest-neighbor search approach to connect residues in a balanced configuration to prevent unwrapping errors. In this section we will evaluate the performance of the algorithm with ultrasound data and highlight the main strengths and weaknesses of the technique. Due to the excessive datasets we have, we will only show a few interesting cases that deserve attention, specifically the failed cases. We will conclude with an overall assessment of the technique.

Figure 4.28a shows a Doppler phantom carotid flow with the detected residues; positive residues are indicated with plus signs (+) while the negative ones are indicated with circle signs (o). One interesting observation is that, aside from the boundary residues, all residues occur in pairs (one positive and one negative) when two regions of discontinuous phase are connected. Specifically, all red pixels in the phase image are aliased while the blue islands inside the red blobs have a mix of unaliased light blue pixels and severely aliased dark blue pixels.

Figure 4.28b shows the detected branch cuts by Goldstein's algorithm. It is clear that the branch cuts are compact and have minimal length. Also the cuts combine an arbitrary number of balanced residues, not necessarily a single pair of them. Figure 4.28c shows the unwrapped phase image displayed with the same color map used for the original wrapped phase image but the unwrapped values are down-scaled to fit within the range of colors available. Specifically in this case the unwrapped values lie in the range $[-2\pi, 2\pi)$ and hence all values are divided by 2 to fit the color map which maps phase values into the range $[-\pi, \pi)$. Because of this scaling operation, the unwrapped phase image will look darker than the original wrapped image. However the aliased pixels will look brighter than the unaliased ones in the unwrapped phase image.

By carefully studying the unwrapping results, it is evident that the algorithm did a fairly good job in unwrapping the phase data, except for a small region (in dark red marked by the green solid arrow in Figure 4.28c) that was not correctly unwrapped and mistakenly shown as reverse flow for a uniphasic carotid waveform, which should not have any reverse flow. The explanation of this error is that the detected branch cuts missed a connection between a pair of opposite sign residues (marked by the two green arrows in Figure 4.28a), which left two discontinuous phase regions connected.

Figure 4.29 shows another example of the flow phantom carotid waveform with the original phase, detected branch cuts, and unwrapped phase. The algorithm did a good job of successfully unwrapping the image where all aliased red pixels were successfully unwrapped. However a small region that is originally severely wrapped (a dark blue area marked by a green arrow in Figure 4.29a) is mistakenly left intact in the unwrapped phase (a dark blue area marked by a green arrow in Figure 4.29c). This is due to the non-optimal branch cut configuration detected in Figure 4.29b in that region, where the algorithm preferred nearest opposite residues over a more logical and optimal configuration that would result in correct unwrapping.

Figure 4.30 is a third example of the flow phantom carotid waveform. The unwrapping error in this case is more severe and affected a larger region (red blob in Figure 4.30c). This is again due to a missing branch cut between two opposite residues (marked by green arrows in Figure 4.30a and b) where the phase unwrapping error propagated to a larger region between two discontinuous phase regions.

Figure 4.31 shows an example of the flow phantom femoral waveform. In this case, there are not that many residues and the detected branch cuts were optimal, as shown in Figure 4.31b. However the unwrapped phase was almost completely wrong with overestimated phase for the majority of the pixels, shown as very light blue pixels in Figure 4.31c. The root cause of this unsuccessful result was due to the fact that the initial pixel used to unwrap the whole image was in fact an aliased pixel (the red pixel marked by the green arrow in Figure 4.31a). This case revealed an important and always forgotten step in all phase unwrapping techniques

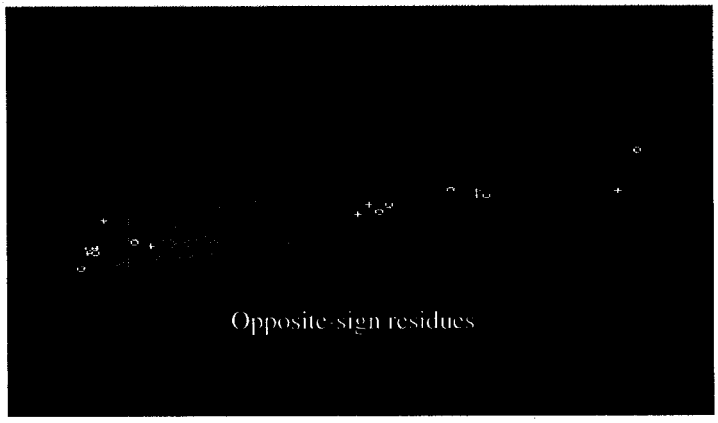
in the literature; they all assume the true phase of the initial pixel to lie between $\pm\pi$. If this is not true then an error of at least 2π will propagate to all pixels in the image, as happened in this case.

Figure 4.32 shows another example of the flow phantom femoral waveform with many residues. In this case the detected branch cuts had many errors and the resulting unwrapped image had errors in most pixels, as shown in Figure 4.32c.

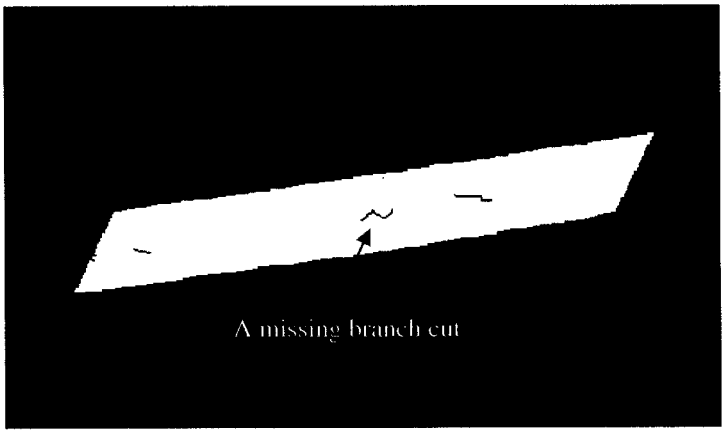
Figure 4.33 shows an example of the flow phantom femoral waveform with non-optimal steering. The algorithm was confused by the original adjacent red and blue blobs due to the non-optimal steering. The unwrapping results had many errors as well.

Figure 4.34 shows an example of the *in vivo* carotid waveform. The residue count is much higher than the *in vitro* cases, perhaps due to noise. The unwrapping results were fairly good except for small underestimated phase regions (red blobs in Figure 4.34c) and overestimated phase regions (very light blue phase regions in Figure 4.34c). The reasonable result could be due to the fact that the correct residue pairs are close to each other and the algorithm does a good job in connecting close residues.

Overall, the performance of Goldstein's algorithm was not satisfactory for ultrasound data since many errors due to random branch cut connections are made, making the results of the unwrapping unpredictable. The algorithm is, however, very simple and fast.



(a)

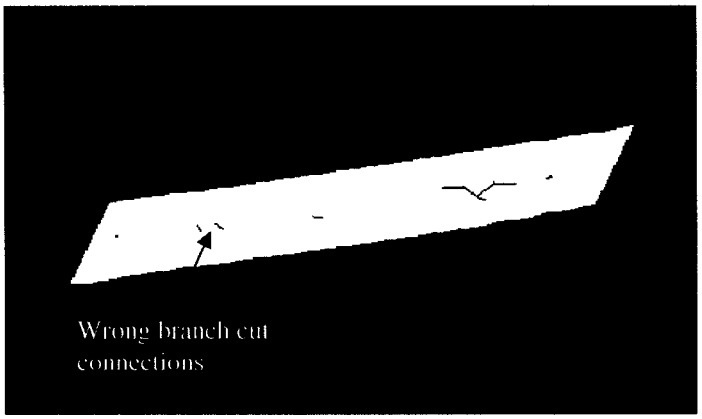


(b)

(c)

Figure 4.28 Goldstein's phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.

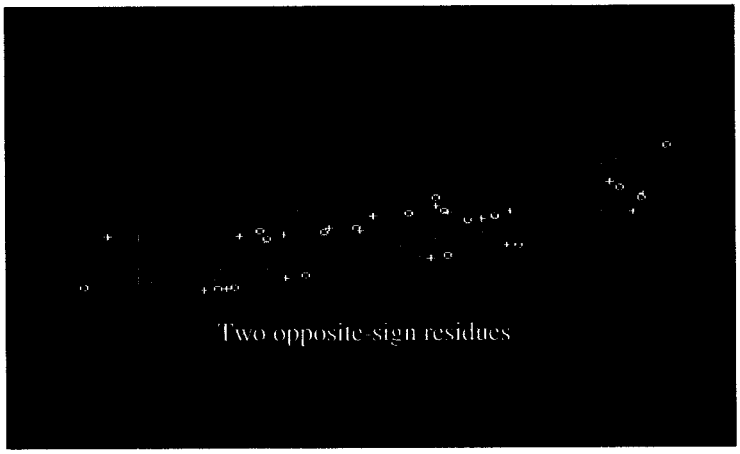
(a)



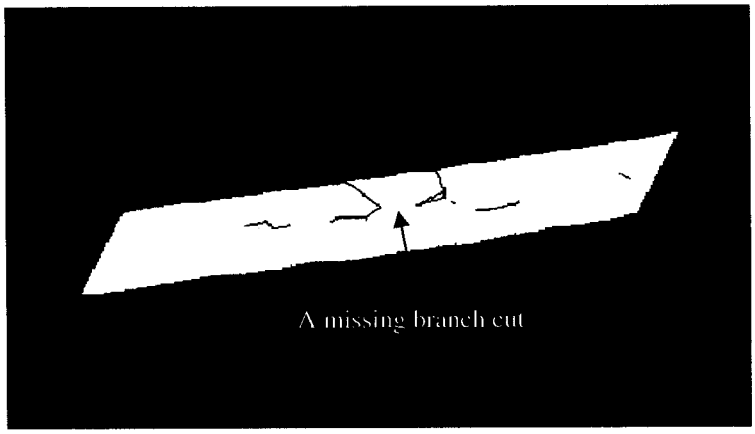
(b)

(c)

Figure 4.29 Goldstein's phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.



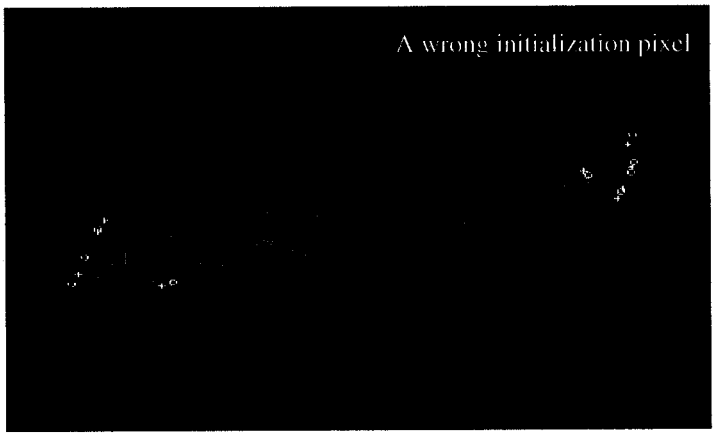
(a)



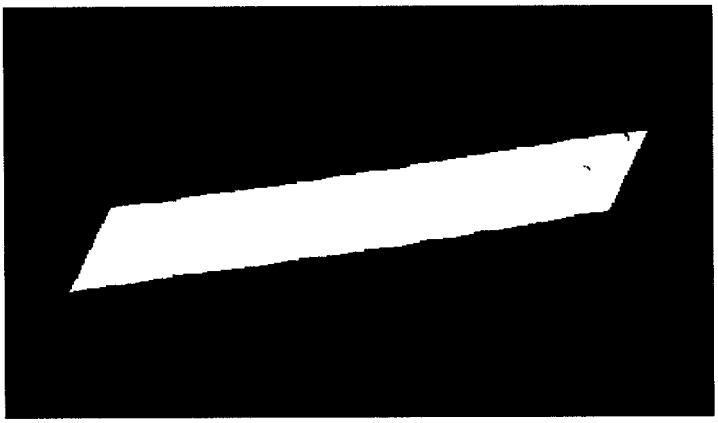
(b)

(c)

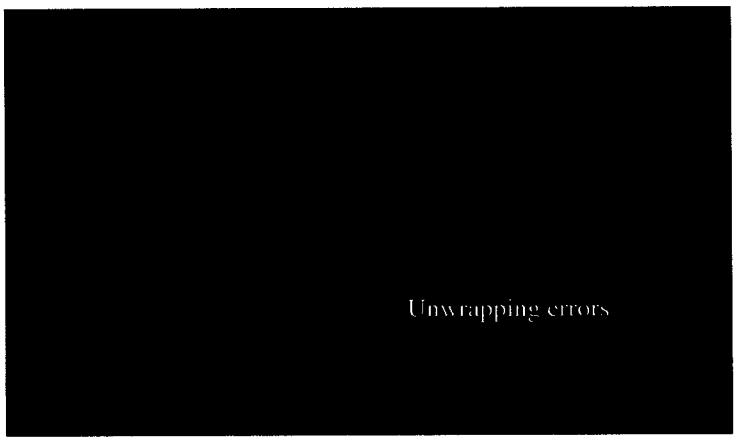
Figure 4.30 Goldstein's phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.



(a)

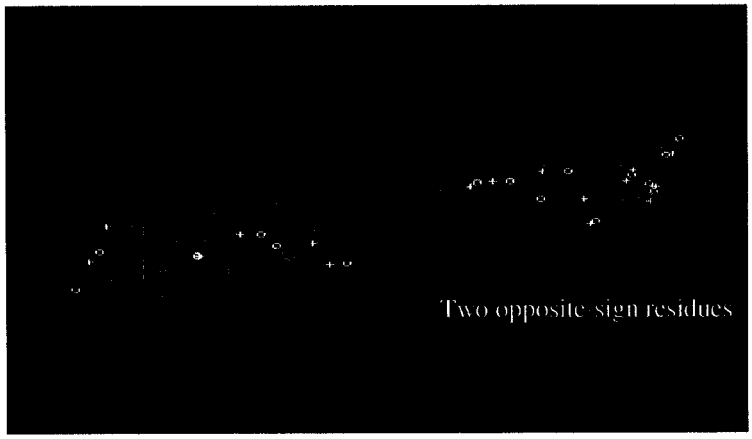


(b)

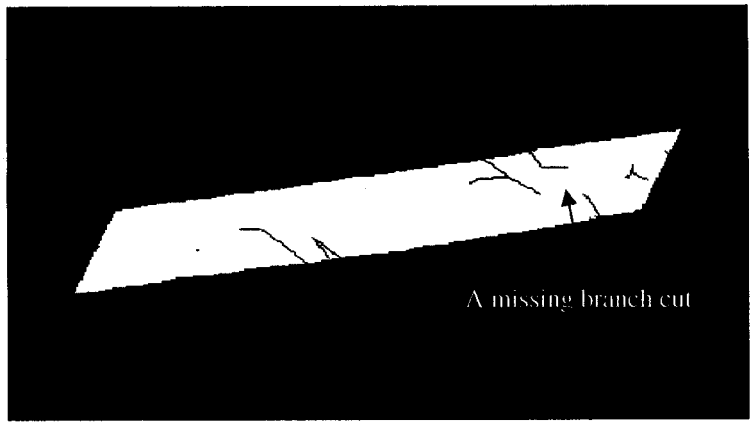


(c)

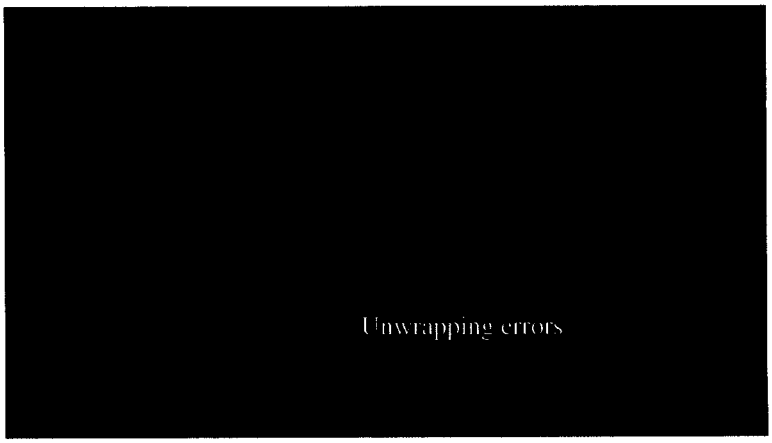
Figure 4.31 Goldstein's phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.



(a)

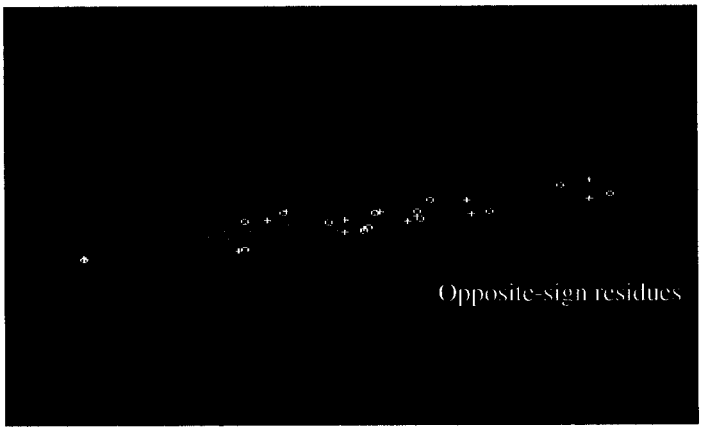


(b)



(c)

Figure 4.32 Goldstein's phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.

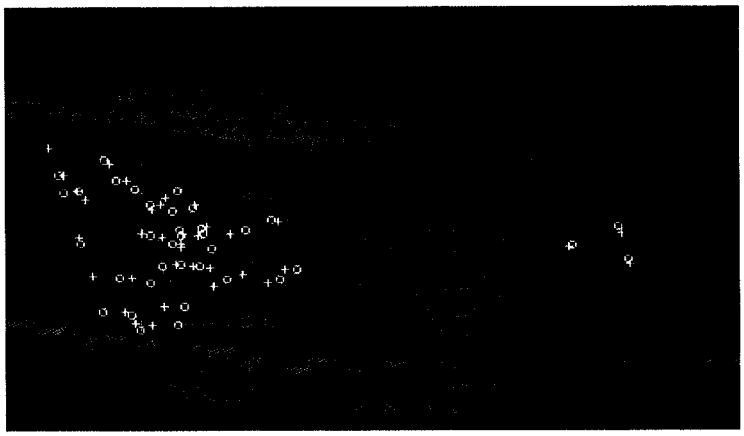


(a)

(b)

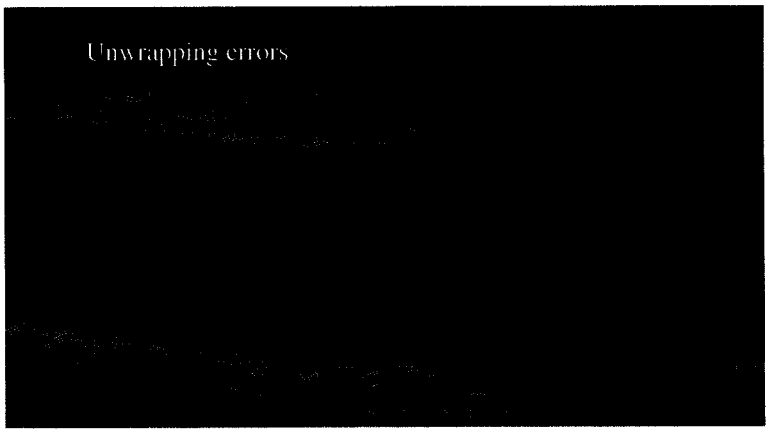
(c)

Figure 4.33 Goldstein's phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.



(a)

(b)



(c)

Figure 4.34 Goldstein's phase unwrapping for *in vivo* carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.

4.4.3. Flynn's Mask Cut Algorithm [68] Results

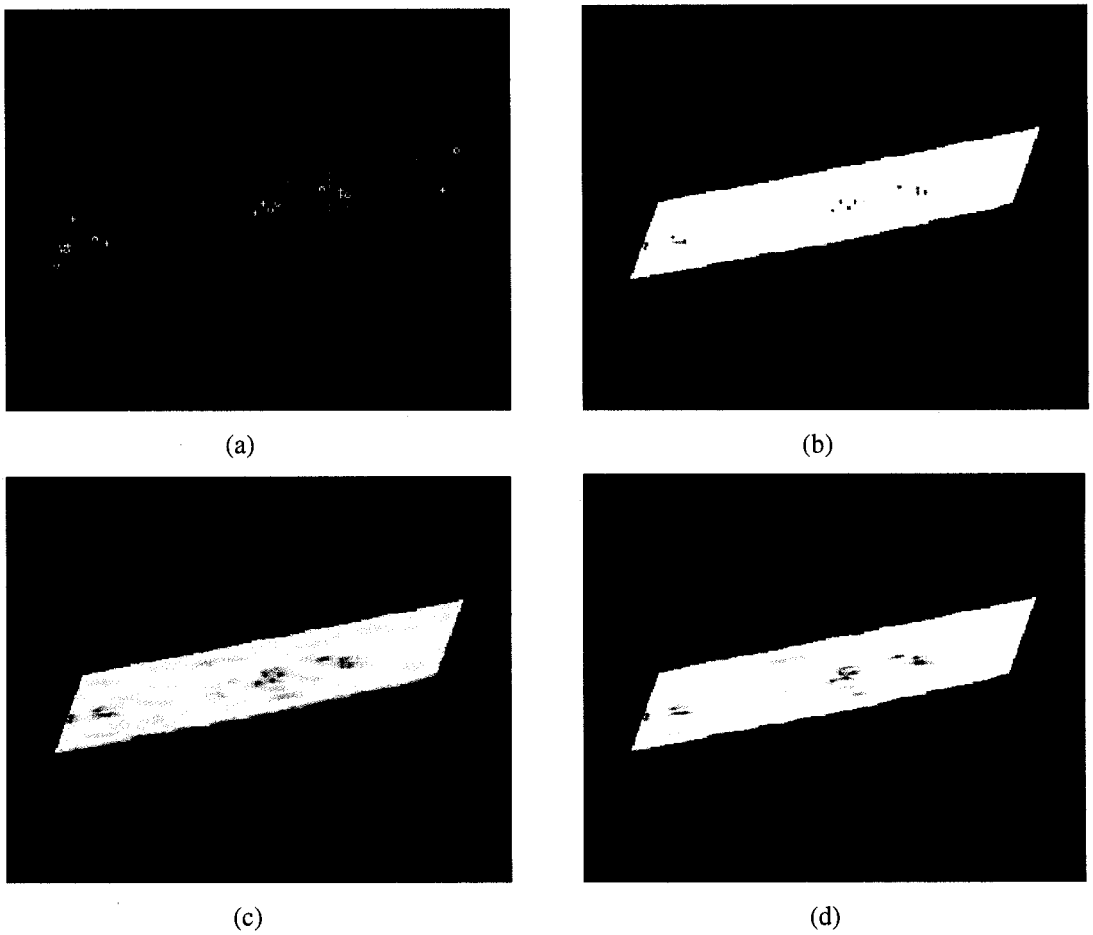
As explained previously, Flynn's mask cut algorithm uses a quality map to derive the detection of branch cuts by following paths of minimal quality. We have tried three quality maps with a few cases to decide the most suitable one to use with ultrasound images. The first was the minimum variance quality map as defined in equation (31) above and shown in Figure 4.35b below. The variance map has low values on and near the residues but otherwise the map shows high values everywhere else, which makes it less useful to guide branch cut placement. The second map was the minimum gradient quality map as defined in equation (32) above and shown in Figure 4.35c below. Similarly, this map has low quality values near the residues but it also has low values in more areas of the image. However the low value areas are not restricted to the neighborhood of the residues, and they get very crowded near the residues. This makes branch cut detection unpredictable but more accurate than with the minimum variance maps. The third map was the maximum pseudo-correlation defined in equation (30) above and shown in Figure 4.35d below. This was the most useful map for ultrasound data, since the low quality areas are highly correlated with the optimal paths between residues. We chose to use this map to evaluate Flynn's mask cut algorithm.

Figure 4.36 shows the results of the mask cut algorithm with a flow phantom carotid waveform. The unwrapping results are better than those of Goldstein's algorithm shown in Figure 4.28. This is due to the more accurate branch cuts as shown in Figure 4.36b. However it is evident that the branch cuts generated by the mask cut algorithm are in general longer than those generated by Goldstein's algorithm.

Figure 4.37 shows the results of a flow phantom femoral example. Despite the good unwrapping results, small regions were not successfully unwrapped due to missing branch cuts, as indicated by the arrows in Figure 4.37b and c. Figure 4.38 shows an example of the flow phantom femoral waveform with non-optimal steering angle. Again the unwrapping results look reasonable except for a small region near the boundary where the phase was overestimated (shown as light blue area in the unwrapped image). The cause was a missing

branch cut between two opposite residues (marked by the green arrows) where there was no low quality path between the two residues. Figure 4.39 and Figure 4.40 show similar cases with more severe unwrapping errors due to missing branch cuts.

In summary, Flynn's mask cut algorithm gave better results than Goldstein's algorithm. This is mainly due to better detected branch cuts that rely on more than just the position of the residues, by utilizing the information provided by a good quality map. However, the derived quality maps do not always agree with the optimal branch cut placement that isolates regions of discontinuous phase to prevent unwrapping errors.

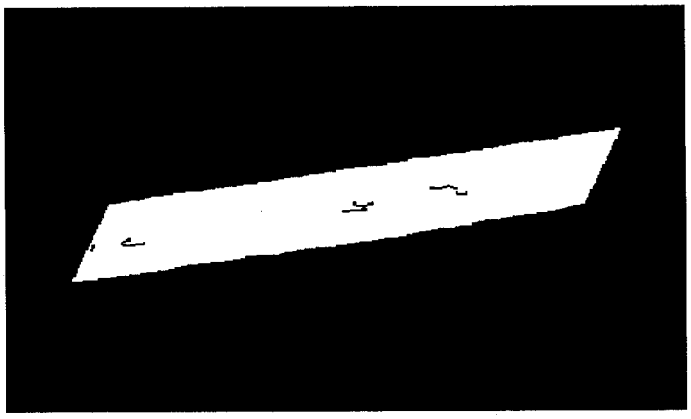


(a) (b)

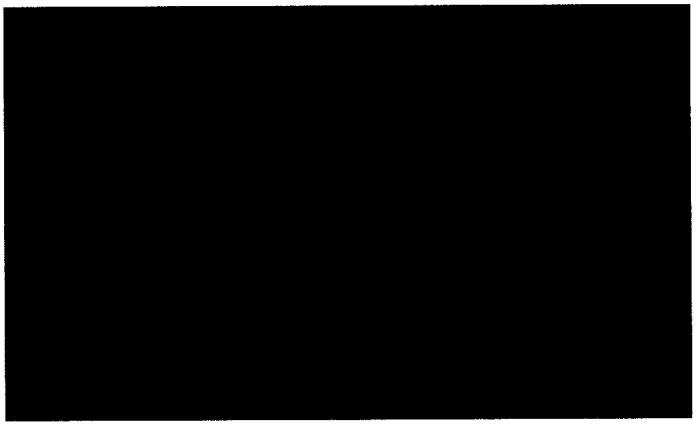
(c) (d)

Figure 4.35 Different quality maps for the Mask cut algorithm. (a) A flow phantom carotid waveform. (b) Minimum variance quality map. (c) Minimum gradient quality map. (d) Maximum pseudo-correlation quality map.

(a)

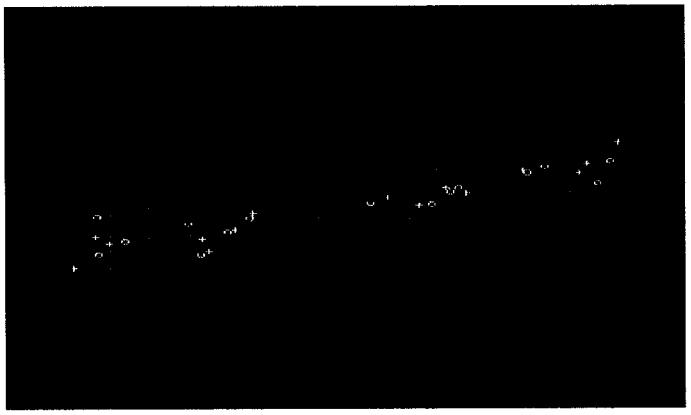


(b)

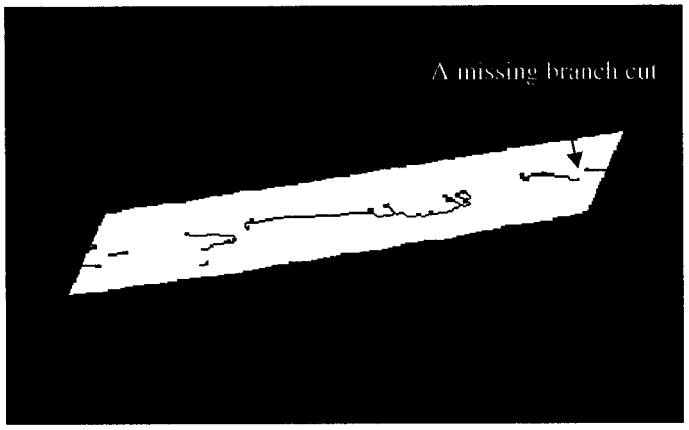


(c)

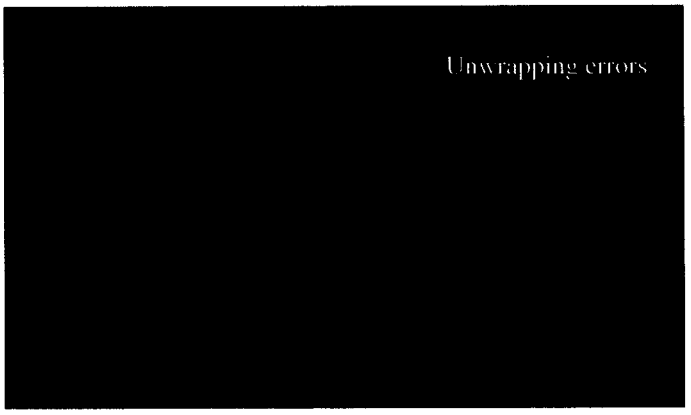
Figure 4.36 Flynn's mask cut phase unwrapping for a phantom carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.



(a)



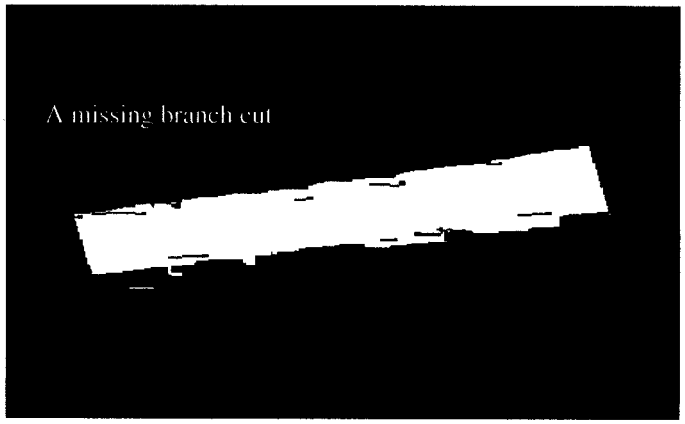
(b)



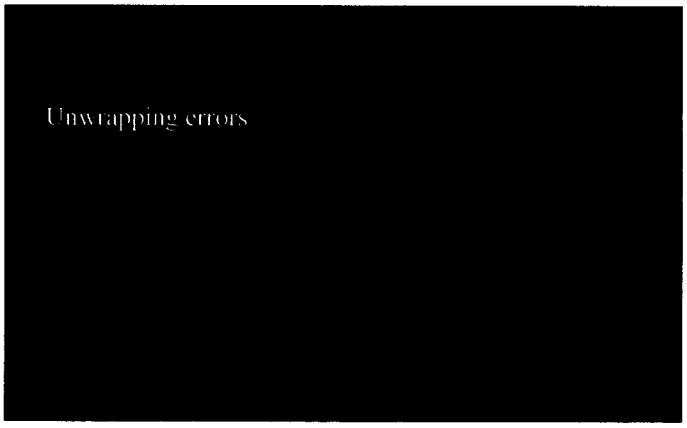
(c)

Figure 4.37 Flynn's mask cut phase unwrapping for a phantom femoral flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.

(a)



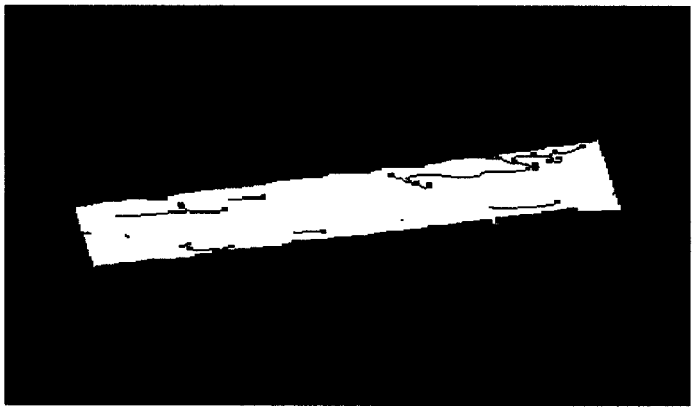
(b)



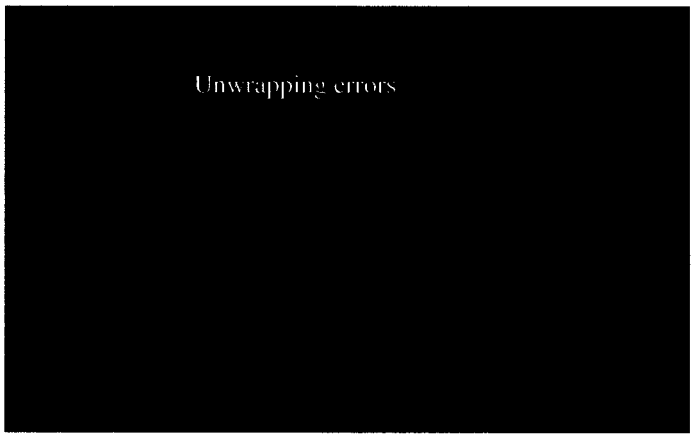
(c)

Figure 4.38 Flynn's mask cut phase unwrapping for a phantom femoral flow with non-optimal steering. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.

(a)

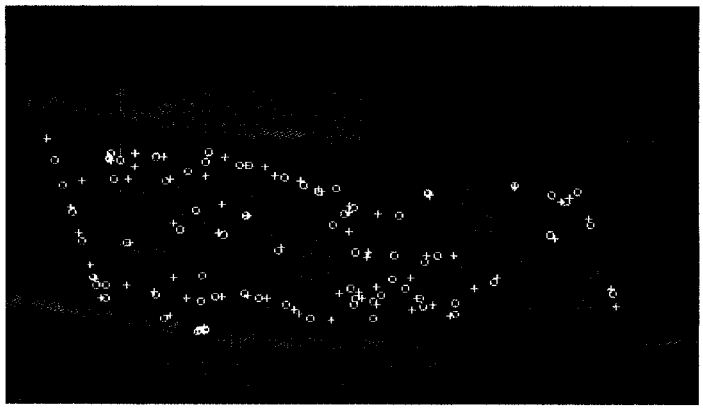


(b)



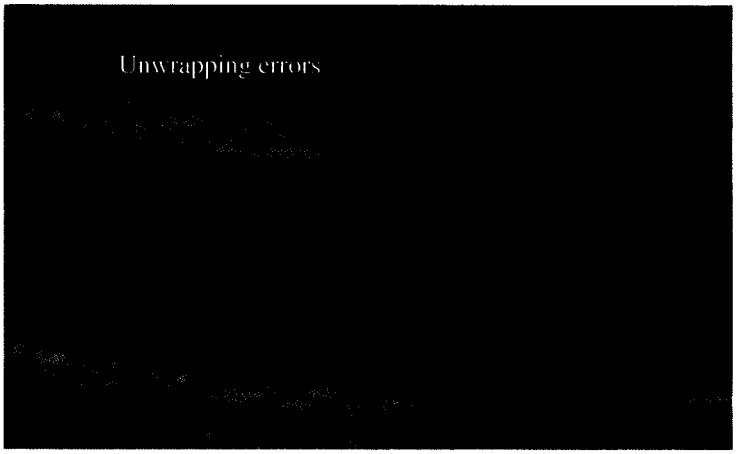
(c)

Figure 4.39 Flynn's mask cut phase unwrapping for a phantom femoral flow with non-optimal steering. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.



(a)

(b)



(c)

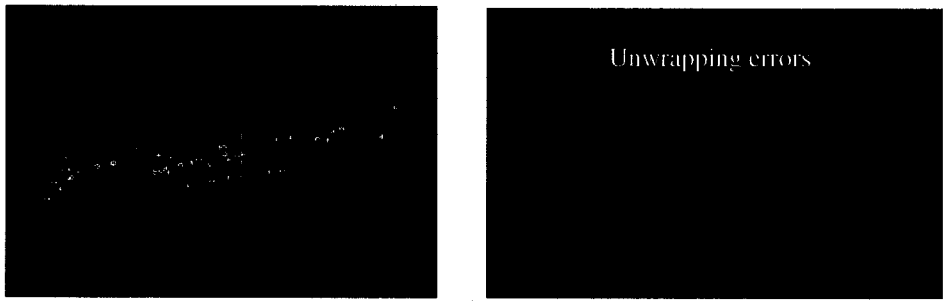
Figure 4.40 Flynn's mask cut phase unwrapping for an *in vivo* carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Detected branch cuts. (c) Unwrapped phase image with down-scaled color map.

4.4.4. Flynn's Minimum Discontinuity Algorithm [69] Results

As explained before, Flynn's minimum discontinuity algorithm relies on detecting fringelines and compensating for the 2π jumps across them. This algorithm can be used with or without a quality map and we have tried both but we did not see a huge performance difference, so we chose not to use a quality map with this technique.

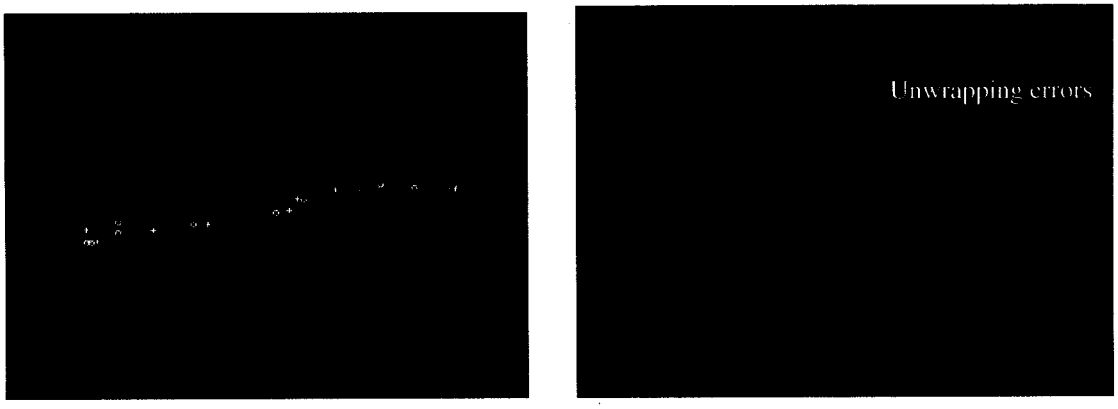
Figure 4.41 shows an example of a reasonably unwrapped result of a flow phantom carotid waveform. However some small regions were not successfully unwrapped (as marked by the arrows). Figure 4.42 shows another less successful example of a phantom femoral flow. A large region has unwrapping error as indicated by the arrow in Figure 4.42b. Figure 4.43 shows an example of the *in vivo* carotid waveform where the unwrapped image was much better than both Goldstein's branch cut and Flynn's mask cut algorithms where the unwrapping error is in a much smaller area.

One main drawback of Flynn's minimum discontinuity algorithm is its computational complexity, which makes the execution time much slower than the other algorithms. This will not be suitable for ultrasound applications which need to be instantaneous, if not real time, due to the real-time nature of the ultrasound imaging. However this algorithm is the most reliable among all path-following techniques.



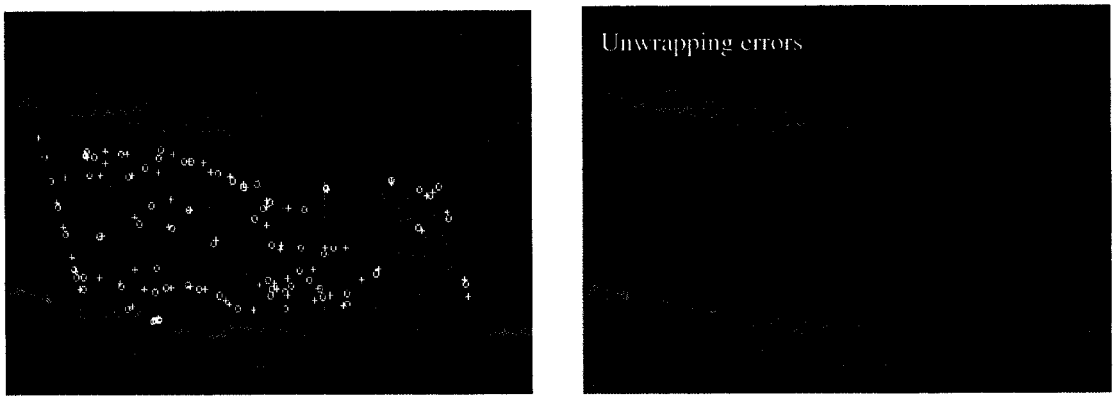
(a) (b)

Figure 4.41 Flynn's minimum discontinuity phase unwrapping for a flow phantom carotid waveform. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Unwrapped phase image with down-scaled color map.



(a) (b)

Figure 4.42 Flynn's minimum discontinuity phase unwrapping for a flow phantom femoral waveform with non-optimal steering. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Unwrapped phase image with down-scaled color map.



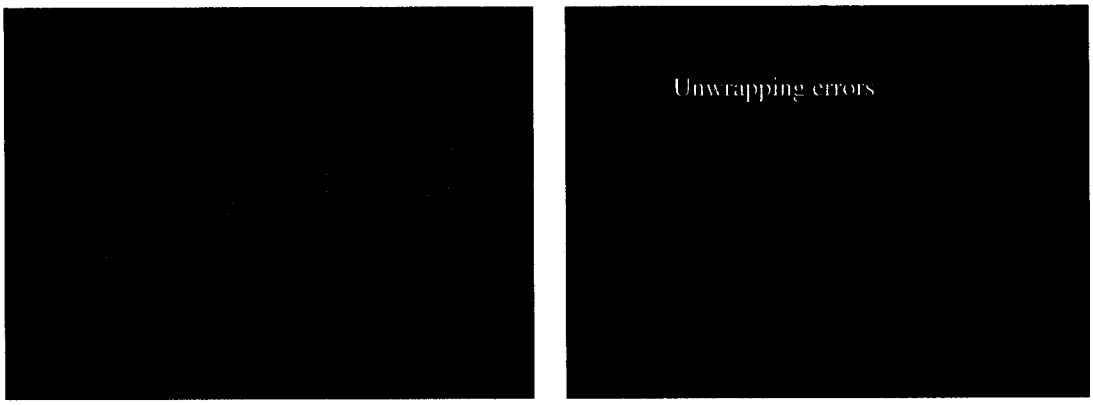
(a) (b)

Figure 4.43 Flynn's minimum discontinuity phase unwrapping for an *in vivo* carotid flow. (a) Original wrapped phase image with detected residues ("+" for positive residues and "o" for negative ones). (b) Unwrapped phase image with down-scaled color map.

4.4.5. Ghiglia's DCT Algorithm [70] Results

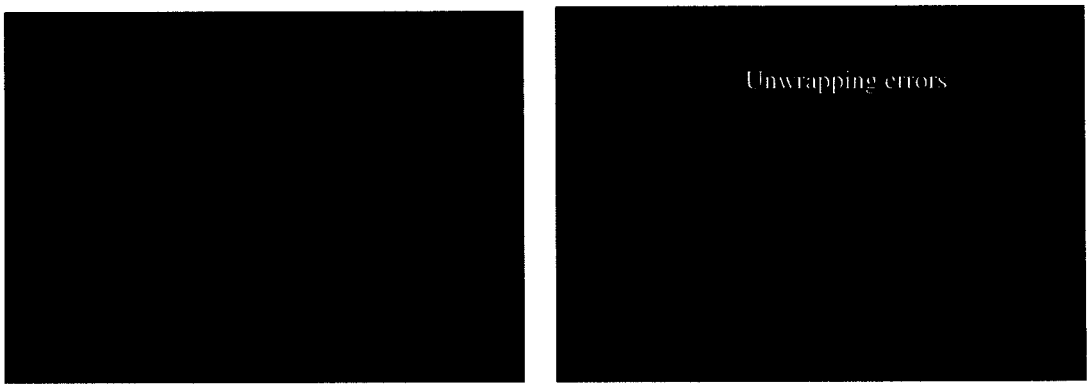
Ghiglia's DCT algorithm is a direct transform method to solve the unwrapping Poisson's equation as a global minimization problem, as described before. This algorithm represents an unweighted minimum least squares unwrapping method.

We had to pad the original color Doppler images with zeros to have a size of multiple of 2 in both dimensions as required by the Discrete Cosine Transform. Despite the increased size, the execution time of the algorithm was still fast compared to other techniques. However the results showed much more unwrapping errors relative to all previous techniques, as shown in Figure 4.44, Figure 4.45, and Figure 4.46 below. This is mainly due to the fact that the algorithm ignores the sources of phase discontinuity (the residues) and seeks a global optimal solution. In this case the errors are not restricted to areas closer to the residues but the errors can show up anywhere in the image.



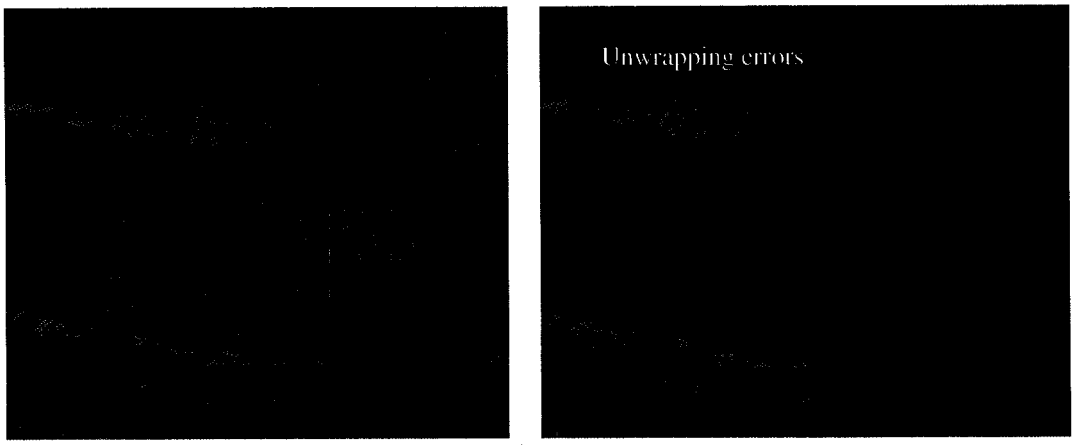
(a) (b)

Figure 4.44 Ghiglia's DCT phase unwrapping for a flow phantom carotid waveform. (a) Original wrapped phase image. (b) Unwrapped phase image with down-scaled color map.



(a) (b)

Figure 4.45 Ghiglia's DCT phase unwrapping for a flow phantom femoral waveform. (a) Original wrapped phase image. (b) Unwrapped phase image with down-scaled color map.

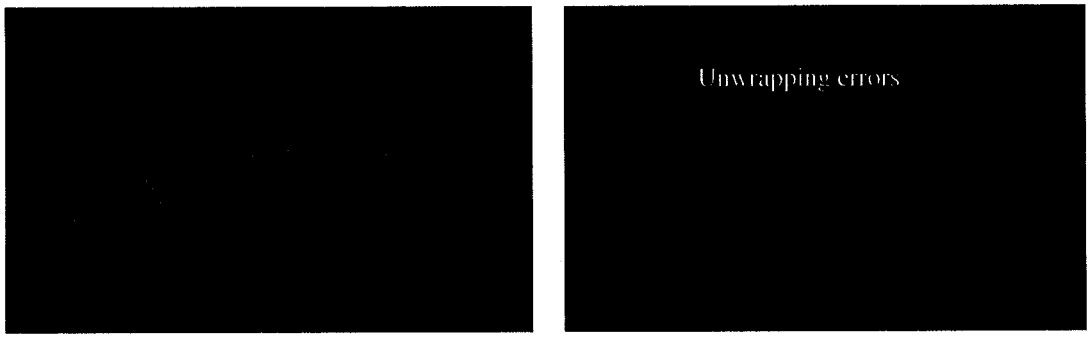


(a) (b)

Figure 4.46 Ghiglia's DCT phase unwrapping for an *in vivo* carotid waveform. (a) Original wrapped phase image. (b) Unwrapped phase image with down-scaled color map.

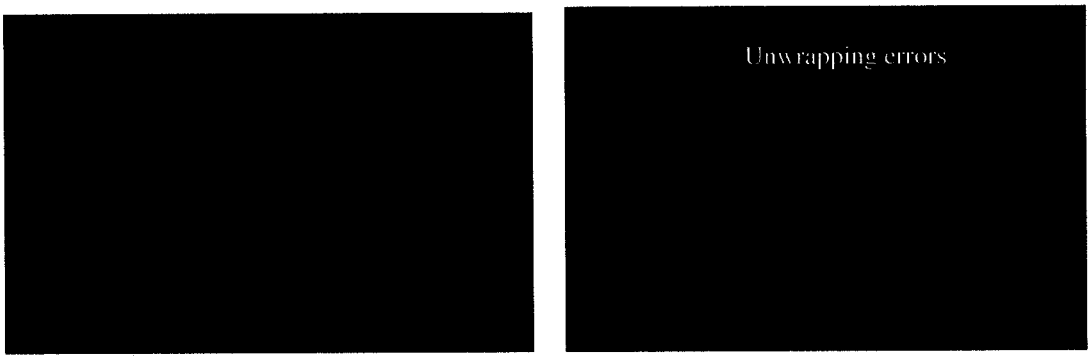
4.4.6. Ghiglia's Preconditioned Conjugate Gradient (PCG) Algorithm [70] Results

The PCG algorithm is an example of a weighted least squares solution that applies an iterative approach. The results were no better than the DCT based approach for ultrasound data, where many unwrapping errors showed up in the unwrapped images, as shown in Figure 4.47, Figure 4.48, and Figure 4.49 below. In addition, the algorithm is extremely slow and will not be suitable for real-time or even interactive ultrasound applications.



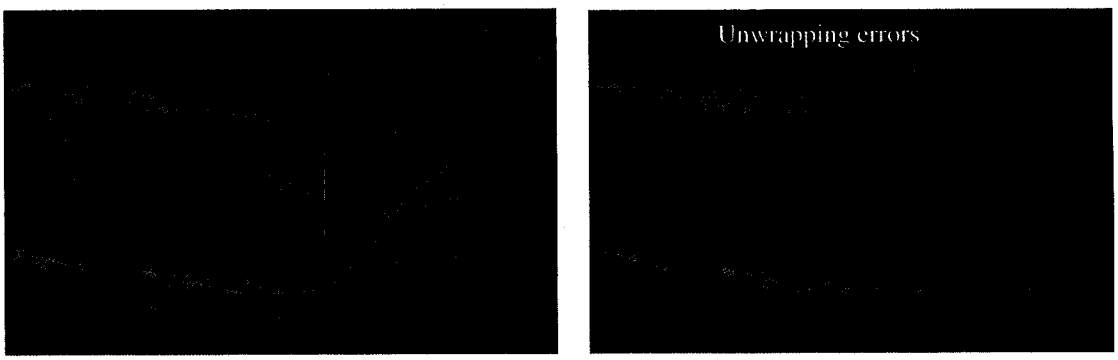
(a) (b)

Figure 4.47 Ghiglia's PCG phase unwrapping for a flow phantom carotid waveform. (a) Original wrapped phase image. (b) Unwrapped phase image with down-scaled color map.



(a) (b)

Figure 4.48 Ghiglia's PCG phase unwrapping for a flow phantom femoral waveform. (a) Original wrapped phase image. (b) Unwrapped phase image with down-scaled color map.



(a) (b)

Figure 4.49 Ghiglia's PCG phase unwrapping for an *in vivo* carotid waveform. (a) Original wrapped phase image. (b) Unwrapped phase image with down-scaled color map.

4.4.7. Kramer's Cutline Detection Algorithm [76] Results

Kramer's cutline detection algorithm does not explicitly unwrap the phase image, but it tries to detect portions of the true cutlines that separate regions of phase discontinuity. We implemented the algorithm as described in section 4.3.4 where the cutline associated with each residue is calculated as the union of the intersection of each two successive fringelines in two phase shifted images, see equation (43). To enhance the performance, we chose not to shift the whole phase image with multiple phase shifts; instead we applied the *phase annihilation*

process described by Chavez et al. in [75] to detect the fringelines. The value used for the annihilation process matched the complement of the sought phase shift.

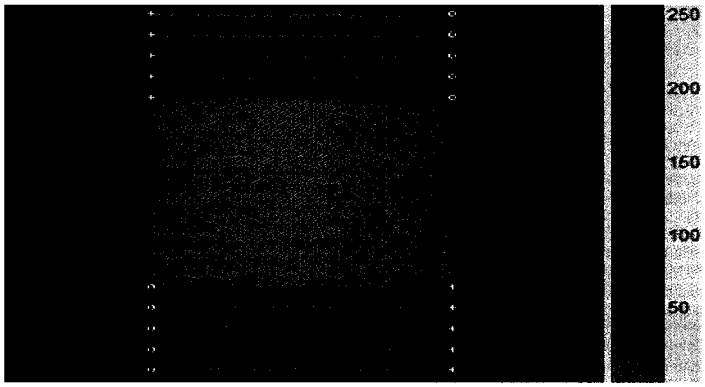
To test our implementation, we first tried the algorithm on some simulation images. We experimented with different numbers of phase shifted images and we found that using 10 phase shifted images is a good compromise. The first image is very similar to the one used in the original paper [76] where it represents a phase ramp, as shown in Figure 4.50. The unwrapped image of a phase ramp that runs vertically is shown in Figure 4.50a. In this image, the phase discontinuity lines are the solid vertical lines that separate the continuous phase region from the black background regions on its right and left. Figure 4.50b shows the phase ramp image after it is wrapped within the range $(-\pi, \pi]$ along with the detected residues. Figure 4.50c shows the detected cutlines by Kramer's algorithm; the result looks similar to the original paper's result. It is clear that the detected cutlines separate the phase ramp region from the background. However, the detected cutlines are disconnected, which is expected from the algorithm as described by the authors.

We tried another simulation image of a spiral shear phase image where two continuous phase regions of a spiral shape are adjacent, as shown in Figure 4.51a. A shear is a line of sharp discontinuity. This image was used by Ghiglia and Pritt in [60] to evaluate several phase unwrapping techniques. The true cutlines or discontinuity lines in this case are the boundary spiral lines between the two regions. Figure 4.51b shows the cutlines detected by Kramer's algorithm. It is evident again that the cutlines are disconnected and will not be sufficient to unwrap the image successfully without introducing errors. However the detected pieces match exactly the boundary between the two spiral phase regions.

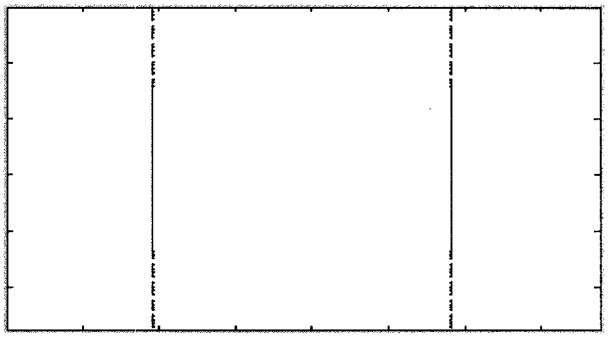
After we assured the correctness of our implementation and evaluated the algorithm with simulation images with known results, we switched to our color Doppler ultrasound datasets. Figure 4.52 a and c show color Doppler images for a simulated carotid waveform from a flow phantom; Figure 4.52b and d show the detected cutlines. It is evident that the detected cutlines do not perfectly separate regions of phase discontinuity and they do not

connect closer residues that should be connected to separate regions of phase discontinuity, as indicated by the arrows.

(a)

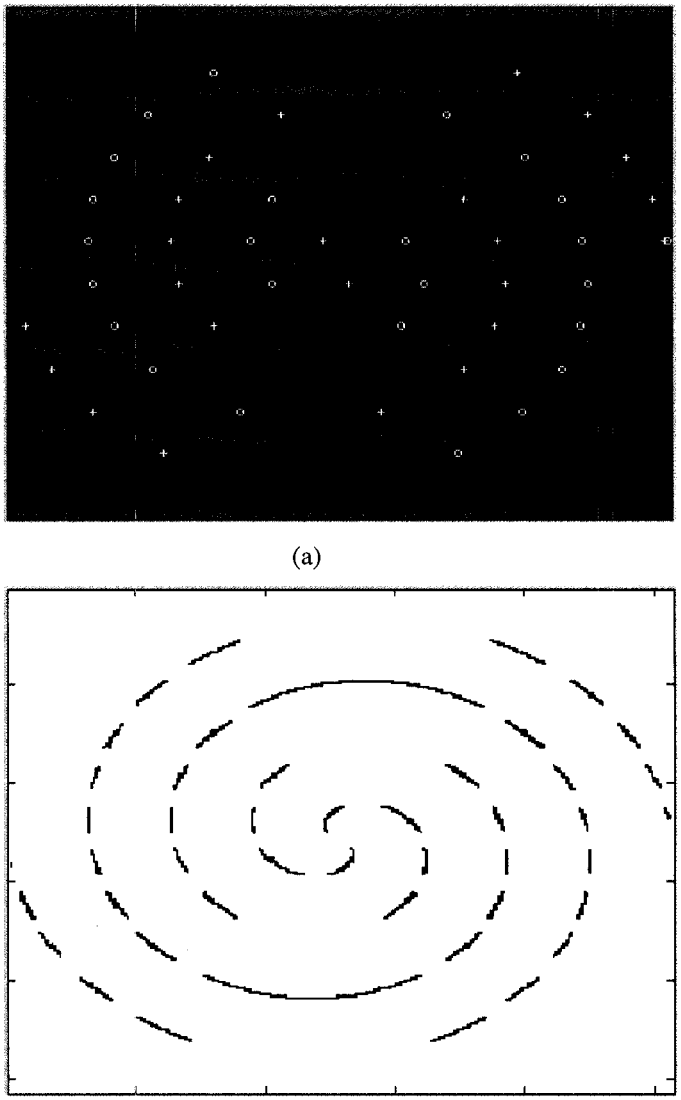


(b)



(c)

Figure 4.50 A simulation image of a phase ramp for Kramer's cutline detection algorithm evaluation. (a) An unwrapped phase ramp image. (b) The wrapped phase ramp image with the detected residues. (c) The detected cutlines by Kramer's method.



(a)

(b)

Figure 4.51 A simulation image of a spiral shear image for Kramer's cutline detection algorithm. (a) A wrapped phase spiral shear image. (b) The detected cutlines by Kramer's method.

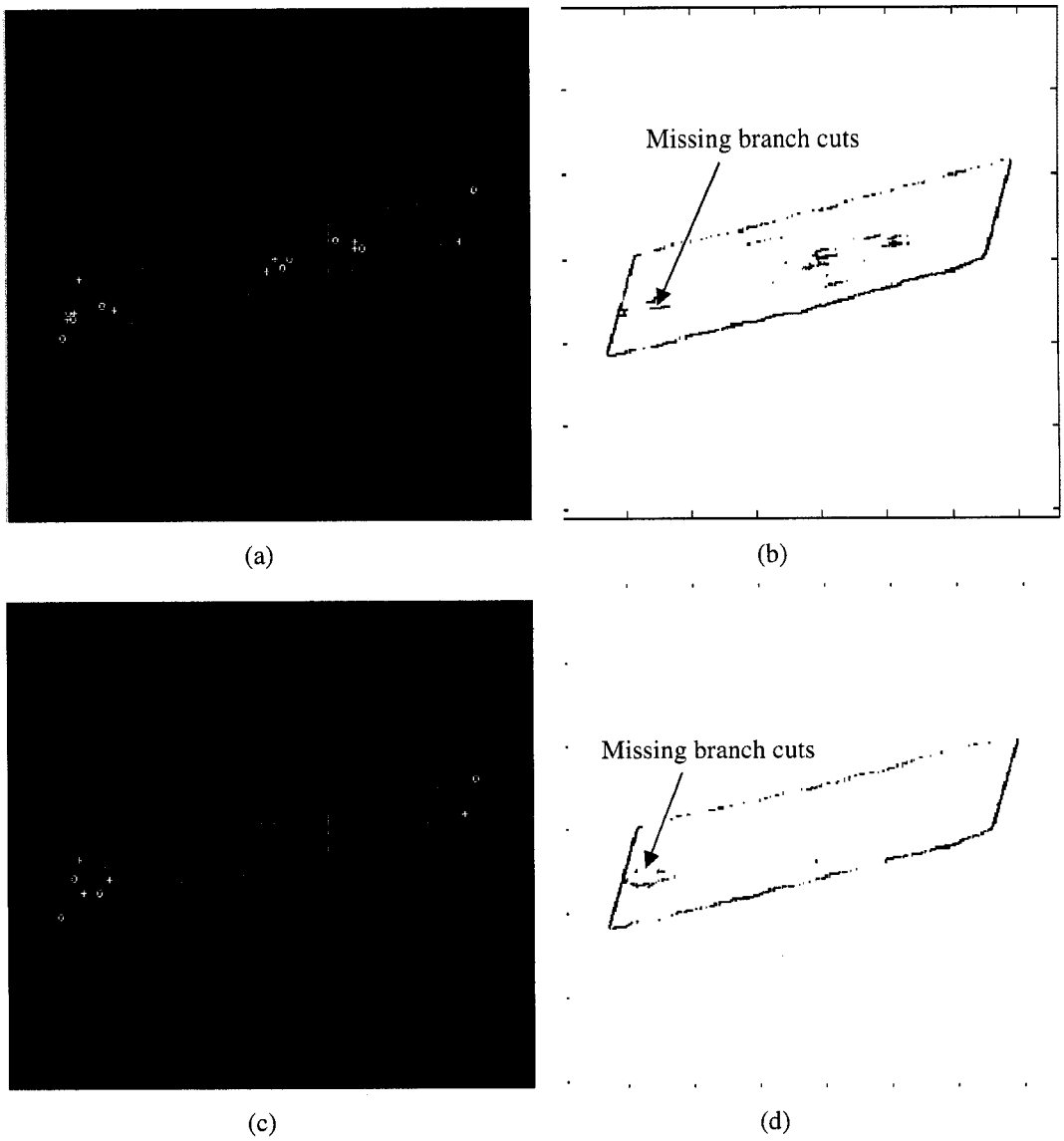


Figure 4.52 (a) & (c) color Doppler images for a carotid waveform simulation. (b) & (d) The detected cutlines by Kramer's method.

4.4.8. Chavez's MRI Phase Unwrapping Algorithm [75] Results

Chavez's MRI phase unwrapping technique uses a similar idea to Kramer's cutline detection algorithm. The algorithm also tries to detect true cutlines that separate regions of phase discontinuity by intersecting fringelines produced by different phase shifts. The phase shifted fringeline intersections are superimposed to construct a score map for candidate cutline paths. The ridges of the score map are tracked to find the final cutlines. Our implementation for the algorithm stops at constructing the score map and we have not implemented the ridge tracking part. This is because it was easy to analyze the score map and predict the unwrapping results from it. We experimented with different numbers of phase shifted images, and we found that using 10 phase shifted images is a good compromise.

Figure 4.53 shows the same phase ramp simulation image used to evaluate Kramer's cutline algorithm. Figure 4.53a shows the wrapped image with all detected residues. Figure 4.53b shows the score map that is constructed from the superposition of phase shifted fringelines. Figure 4.53c and d zoom in the phase and score map images to highlight the findings. It is evident from the score map that the brightest lines coincide with the sought vertical cutlines that should separate the phase ramp from the surrounding background. However, the lines are not continuous where there are gaps between them with low scoring values.

Figure 4.54a shows the spiral shear wrapped phase image with the detected residues. Figure 4.54b shows the constructed score map. Similar to the previous example, the highest score lines are aligned with the sought cutline between the two discontinuous phase regions. However there are gaps between the lines that may cause unwrapping errors.

We tried the algorithm on ultrasound images. Figure 4.55 shows an example of a flow phantom carotid image with the detected score map. The figure also shows zoomed images for both the wrapped phase and score map images. From the score map image, it is clear that the connectivity of the cutlines is much better than the simulation images. This is probably due to

the different nature of the discontinuity lines in ultrasound images versus those of the simulation images; however gaps may still exist between line pieces. By carefully examining the ridges of the score map, which would map to the detected cutlines, we notice that they do not separate the discontinuous regions (severely aliased or dark blue region versus unaliased or light blue region). Instead they follow the line of zero phase that separates positive and negative phase regions, as indicated by the arrows. This will result in unwrapping errors for the severely wrapped regions. Figure 4.56 shows another example of a flow phantom carotid image where the quality of the score map is worse than the previous example, and the cutlines still miss the boundary between the discontinuous phase regions.

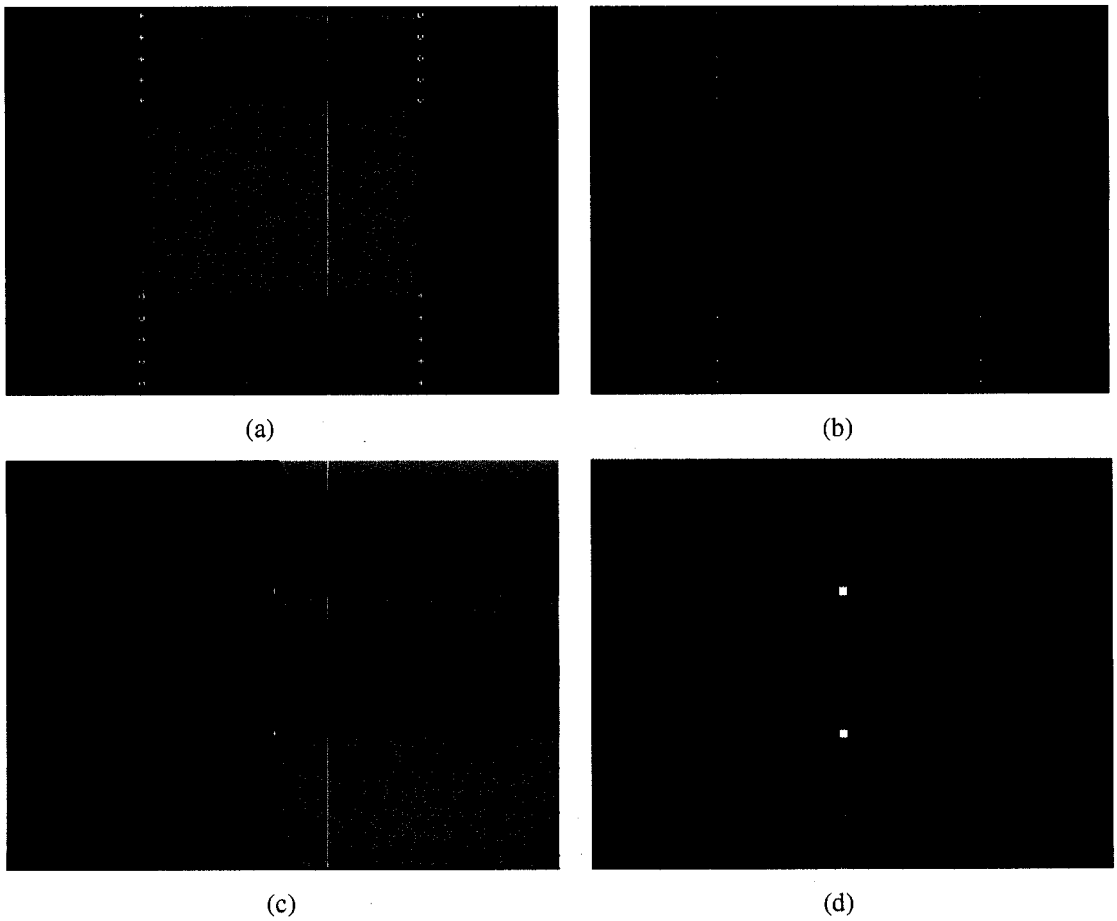
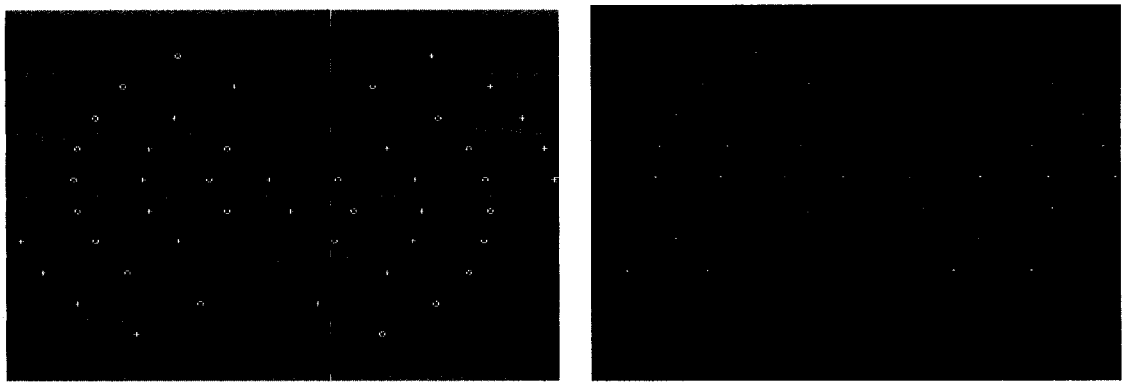
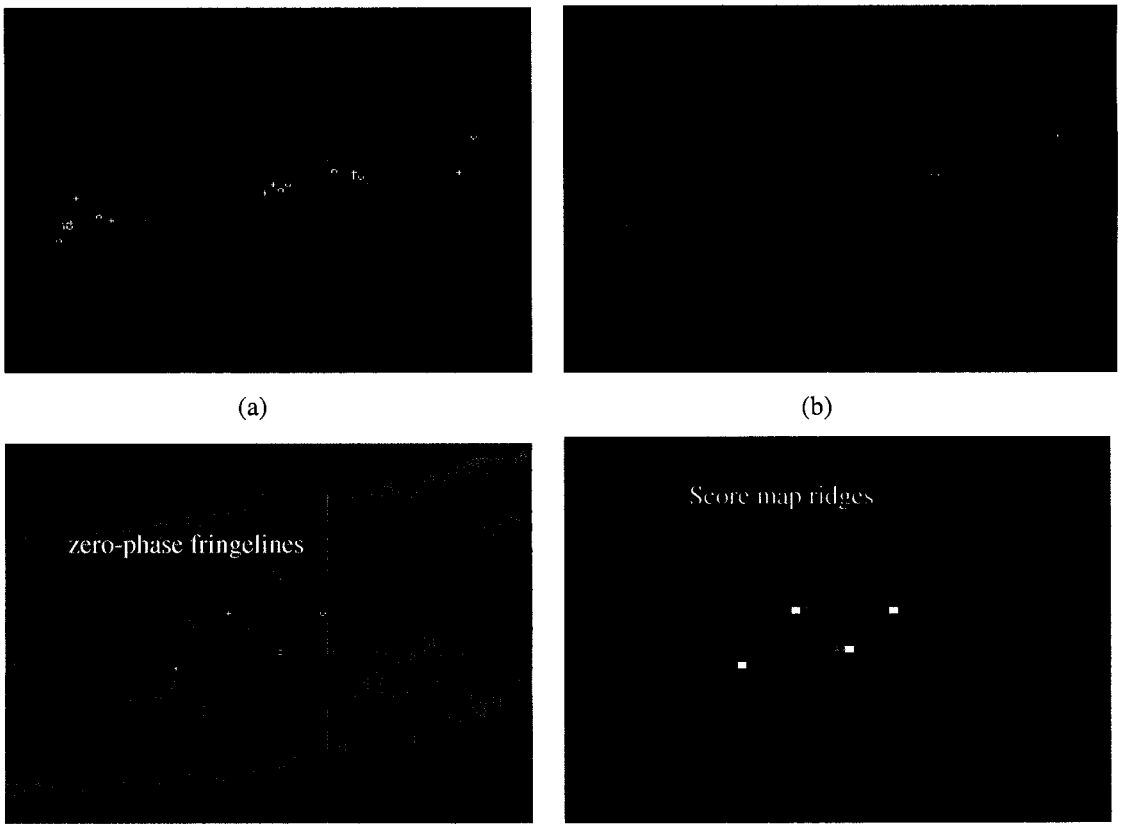


Figure 4.53 A simulation image of a phase ramp for Chavez's MRI unwrapping algorithm evaluation. (a) A wrapped phase ramp image with the detected residues. (b) The score map constructed from the phase shifted fringeline superposition. (c) & (d) zoomed images of (a) and (b).



(a) (b)

Figure 4.54 A simulation image of a phase ramp for Chavez's MRI unwrapping algorithm evaluation. (a) A wrapped phase spiral shear image with the detected residues. (b) The score map constructed from the phase shifted fringeline superposition.



(a) (b)

(c) (d)

Figure 4.55 A flow phantom carotid image for Chavez's MRI unwrapping algorithm evaluation. (a) The wrapped phase image with the detected residues. (b) The score map constructed from the phase shifted fringeline superposition. (c) & (d) zoomed images for (a) and (b).

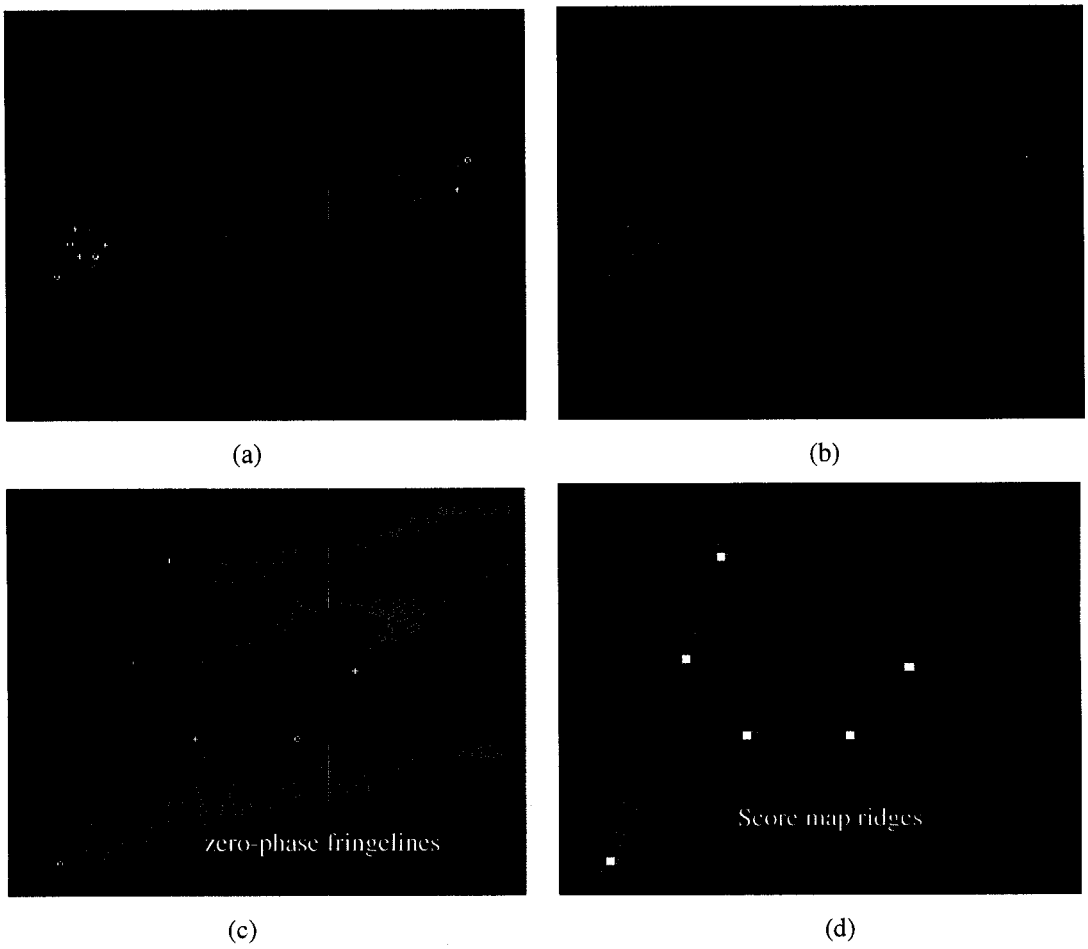


Figure 4.56 A flow phantom carotid image for Chavez's MRI unwrapping algorithm evaluation. (a) The wrapped phase image with the detected residues. (b) The score map constructed from the phase shifted fringeline superposition. (c) & (d) zoomed images for (a) and (b).

4.5. Conclusions

So far we have evaluated the performance of different phase unwrapping techniques from the literature on color Doppler ultrasound images. All of the techniques did well with simple phase-wrapping problems. However, none of them was able to successfully unwrap all the cases that we tried.

Goldstein's branch cut algorithm gave random and unpredictable results, since it relies solely on the residues location, and the detected branch cuts are random. There is a great

chance that the resulting branch cuts will not completely separate regions of phase discontinuity, and unwrapping errors will propagate across these regions. Flynn's mask-cut algorithm did a better job by using the pseudo-correlation quality map to guide the placement of the branch cuts. However the low quality regions did not always match the regions of phase discontinuity and allowed wrapping errors to occur. Flynn's minimum-discontinuity algorithm was the most robust path-following technique. However it showed some minor errors, especially near the vessel boundaries. The slow execution time of the algorithm will prevent it from being useful for ultrasound applications that require interactive if not real-time processing.

The minimum norm methods were, in general, less successful than the path-following methods. This is due to the fact that the minimum norm methods apply a global minimization technique, rather than a local search technique, to unwrap the images. Global errors tend to spread across the whole image instead of being local to the residues regions. This is especially true for the unweighted least-squares techniques like Ghiglia's DCT algorithm that does not take into account the effect of residues. Ghiglia's PCG algorithm, that applies a weighted least-squares technique, utilizes a quality map that takes into consideration the effect of the residues. The PCG results are more robust than those of the DCT algorithm; however wrapping errors near the vessel boundaries occurred. Again the slow execution time of this double iterative approach will prevent it from being useful for ultrasound applications.

The true cutline detection and unwrapping methods indicated a promising direction to seek the accurate lines of phase discontinuity. However the brief results we obtained were not satisfactory enough for ultrasound images. The methods suffered from disconnected cutlines that will require post processing steps to connect in a logical way. In addition, the detected cutlines did not exactly match the discontinuity lines between different phase regions; this will result in unwrapping errors.

The conclusion from the previous analysis was that we need to develop our own phase-unwrapping technique that works best for the color Doppler ultrasound aliasing problem. In the following chapter, we will derive a thorough analysis of the phase-wrapping

problem for ultrasound images, taking into consideration the physics and signal processing of the color Doppler technology. We will then describe our novel phase-unwrapping technique for ultrasound images and present the results on many datasets, including those used to evaluate previous techniques.

Chapter 5 Fringeline Tracking Approach for Color Doppler Phase Unwrapping

5.1. Introduction

In the previous chapter, we introduced the problem of phase unwrapping for color Doppler images. The theory behind phase unwrapping was described and the most successful phase unwrapping techniques were reviewed. The results of some well-known phase unwrapping techniques on color Doppler ultrasound images were presented and analyzed. The results highlighted the shortcomings of all the techniques we tried, which motivate the development of a new technique that is customized to the color Doppler phase unwrapping problem.

In this chapter we will describe a new technique for unwrapping color Doppler ultrasound images. In section 5.2, we will formulate the problem of color Doppler ultrasound phase unwrapping by highlighting the unique aspects that exist in ultrasound images in contrast to the other imaging modalities, such as SAR and MRI images, that suffer from phase wrapping problems. In section 5.3, we will describe the core idea behind our technique for phase unwrapping that relies on the fringeline tracking concept that is used previously in [75] and [76] to detect true phase discontinuity cutlines. The detailed steps of our new algorithm will be described in section 5.4. The results of the new technique will be given in section 5.5, while the validation of the results will be discussed in sections 5.6 and 5.7. Finally future ideas and extensions to the current technique will be discussed in section 5.8.

5.2. Ultrasound Phase Unwrapping Problem Statement

In this section we will try to analyze the color Doppler ultrasound wrapping problem in some detail. Our goal is to highlight the important aspects of the problem in order to guide the development of a solution that is customized to the ultrasound data.

We start our discussion by introducing a simulation method that introduces artificial wrapping to an original color Doppler ultrasound image that is not wrapped, i.e. all velocity values within this image lie in the range $(-\pi, \pi]$, as shown in Figure 5.1a. The artificial wrapping is achieved by multiplying the original unwrapped velocity pixels by a factor that is greater than one and then wrapping the resultant velocity values in the range $(-\pi, \pi]$, as indicated by the equation below:

$$V_w = \tan^{-1}(\sin(nV_u)/\cos(nV_u)) \tag{50}$$

where V_u is the original unwrapped velocity that lies in the range $(-\pi, \pi]$, V_w is the simulated wrapped velocity that results from multiplying the original velocity by a factor n that is greater than one. Figure 5.1b-f show velocity wrapping simulations for $n = 2,3,4,5$, and 6 respectively. It is clear from the figures that the wrapping degree increases as the velocity values increase. Another important observation is that the residues (indicated by + and o signs) are confined to the boundary areas of the vessel, where the color Doppler data touches the grayscale areas surrounding the vessel. In contrast, the residues are completely absent from the lumen of the vessel, even when regions of wrapped phase data are touching. In particular, the fringelines that mark boundaries of 2π phase jumps are closed contours. The lack of residues in these simulated images (except at the boundaries which are usually ignored by most phase unwrapping techniques) simplifies the task of phase unwrapping.

To better understand the phase wrapping problem for color Doppler ultrasound data, we conducted another experiment using a flow phantom. The experiment is similar to the one we previously used for testing existing phase unwrapping techniques and described in section 4.4.1. The main idea is to sweep the sampling rate, or the pulse repetition frequency (PRF), during the acquisition of color Doppler cineloops of a flow phantom carotid waveform. Some color Doppler ultrasound parameters are controlled during the experiment to minimize phase data distortion.

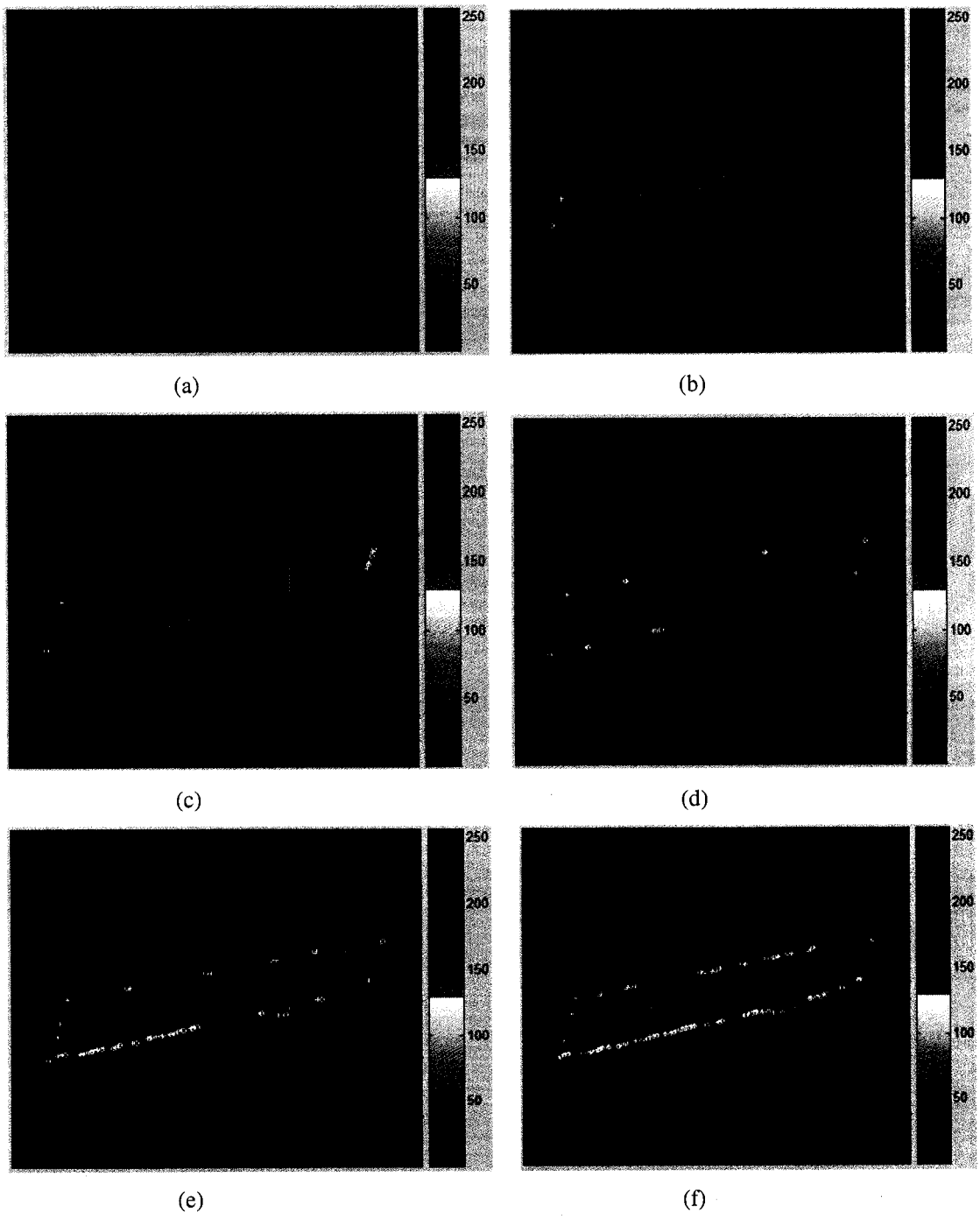


Figure 5.1 Color Doppler ultrasound phase wrapping simulation. (a) Original unwrapped color Doppler image. (b) A wrapping simulation with velocity values multiplied by 2. (c) A wrapping simulation with velocity values multiplied by 3. (d) A wrapping simulation with velocity values multiplied by 4. (e) A wrapping simulation with velocity values multiplied by 5. (f) A wrapping simulation with velocity values multiplied by 6.

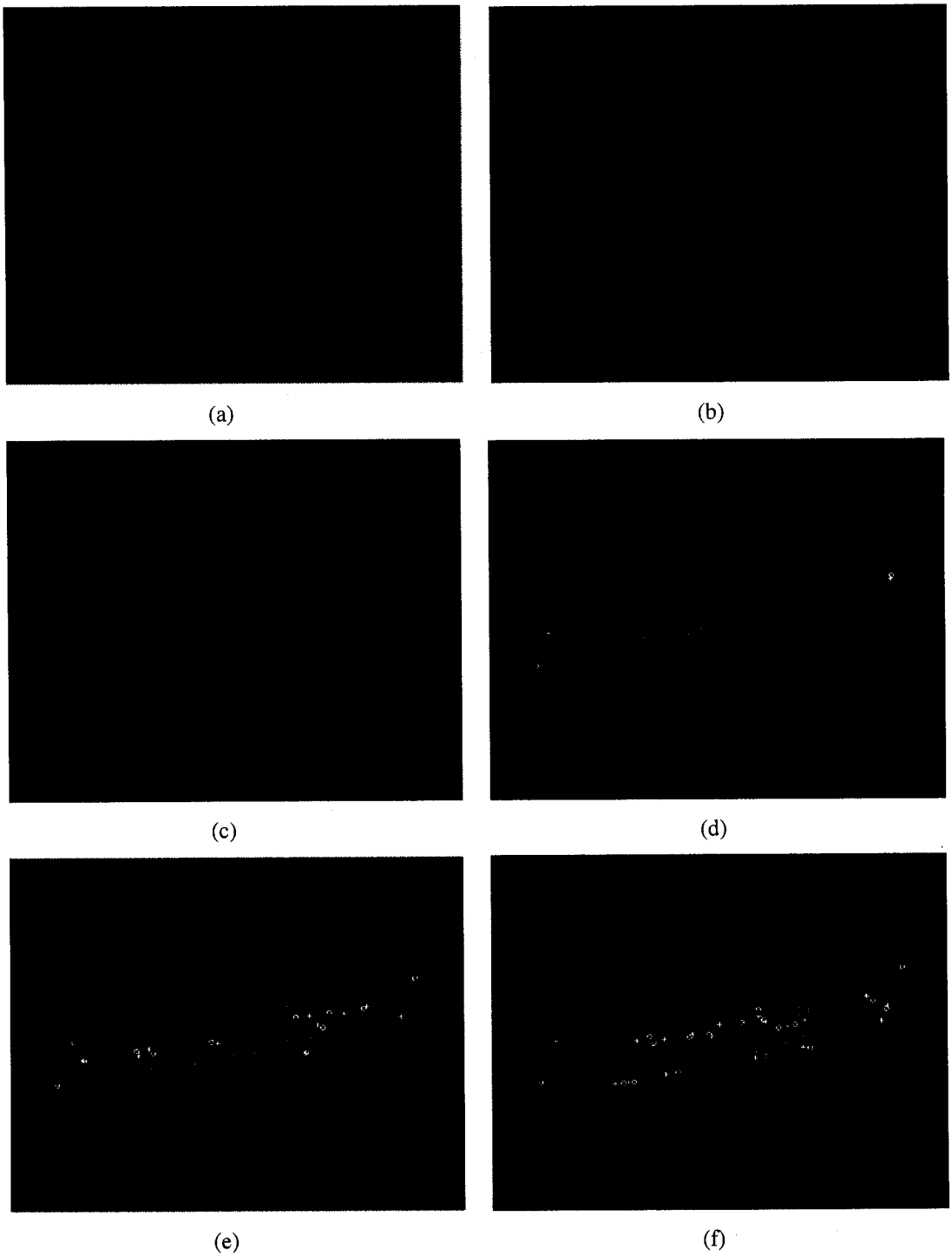
The color persistence is turned off and the wall filter is set to its minimum value. Starting with the highest possible PRF, there will be no wrapping present, since the sampling rate is adequate for the maximum velocity present within the simulated carotid vessel. Gradually lowering the PRF causes phase wrapping to increase. Since the velocity reaches its maximum during the systole phase within the carotid artery, only the systolic frames of the acquired cineloops are shown in Figure 5.2. The figure shows six different systolic frames with different sampling rates, starting with the highest rate. The phase wrapping increases as the sampling rate decreases. When the wrapping degree is less than π, as shown in Figure 5.2c and d, the residues are confined to the vessel boundaries, and the fringelines tend to be closed. If the wrapping degree exceeds π, as shown in Figure 5.2e and f, the residues appear in the vessel lumen, in contrary to the wrapping simulation example shown in Figure 5.1 above. The main difference between the simulated wrapping in Figure 5.1 and the true wrapping in Figure 5.2 is that the true high velocities have gone through the signal processing operations within the ultrasound system, including the non-linear high pass filter, the wall filter, which rejects the low frequency wall motion signal. This non-linear processing is responsible for creating adjacent regions of discontinuous phase (unwrapped pixels in light blue and wrapped ones in dark blue). When these discontinuous regions touch each other, residues appear within the vessel lumen.

The previous observation is essential to understand the nature of the phase wrapping problem for color Doppler ultrasound images. The source of the residues is primarily the adjacency of discontinuous phase regions instead of noise, although thermal and digital noise may exist in ultrasound images and create residues as well. But the noise contribution to the residues is minor compared to the contribution of the phase discontinuity created by the non-linear processing in the ultrasound system. As a result, the true cutlines that should separate the phase discontinuity regions will not have to be short in length, which is a characteristic of noise-induced residues. On the contrary, the cutlines may become long, depending on the size of the discontinuous regions. This explains why most of the existing path-following techniques did not perform well with ultrasound images when we tested them in the previous chapter.

Another important conclusion that we derived from the analysis of different *in vivo* scanning of healthy human vessels is that the useful range for phase wrapping that we should consider is maximally the three-time folding of the original unwrapped data, i.e. the original unwrapped data lies in the range of $(-3\pi, 3\pi]$. Practically very low sampling rates are not used by the ultrasound system clinicians since the non-linear processing of such a high degree of aliasing totally distorts the images and makes them clinically unusable. A typical ultrasound system is capable of sampling at a rate that is equivalent to a Nyquist velocity of 130 cm/s, for a shallow artery such as the carotid. Three-times wrapping would mean that the mean velocity should not exceed 130*3=390 cm/s. This is a very safe assumption since the worst pathological case of the carotid stenosis will not exceed such a high mean velocity value. Typically a peak systolic velocity above 125 cm/s indicates a narrowing or stenosis in the vessel [78]. The velocity estimated in color Doppler images represents the mean velocity, which is typically half the value of the peak velocity for an ideal laminar steady flow with parabolic velocity cross-sectional profile [14].

To summarize the findings of our analysis of the color Doppler ultrasound wrapping problem, here are the most important aspects that affect the solution:

- The main source of phase residues is the adjacency of discontinuous phase regions that result from the non-linear processing of the acquired wrapped data. Thus the true cutlines are not necessarily short in length.
- The clinically useful range for phase wrapping for ultrasound data is the three-fold wrapping order or equivalently the true unwrapped phase lies in the range $(-3\pi, 3\pi]$.



(a) (b)

(c) (d)

(e) (f)

Figure 5.2 The systolic frame of a flow phantom Carotid waveform cineloops with sweeping sampling rates (PRFs). (a) The highest PRF. (f) the lowest PRF.

5.3. The Fringeline Tracking Core Concept

As we explained in the previous chapter, the main source of complexity for all phase unwrapping problems is the existence of residues (or phase inconsistencies), and color Doppler ultrasound wrapping problem is no exception to this fact. If the phase unwrapping process using the integral of the phase gradients ignores these residues, then unwrapping phase errors will occur and propagate along the remaining path of the phase unwrapping process. If the residues are to be balanced by connecting opposite sign residues with branch cuts that prevent the unwrapping process from crossing these boundaries, the resulting unwrapped image is guaranteed to have consistent values, no matter what the initial point and the path integral were.

This was the main idea behind all path-following techniques that we explained in the previous chapter. However the ad-hoc connections between the opposite residues do not guarantee the correctness of the unwrapping results. A few researchers tried to detect the true phase discontinuity lines (cutlines) to achieve correct results [75], [76]. However these techniques did not work well with ultrasound data either, due to the different nature of the ultrasound phase images.

The ultimate goal of our research is to be able to recognize blood vessels from color Doppler images through the extraction of quantifiable discriminating features from the ultrasound data. Hence the accuracy of the unwrapping results is a primary requirement for the successful quantification of the color Doppler data. For this reason, we decided to focus on the true cutline detection concept and find a way to get it to work for our ultrasound problem.

One important observation from the analysis of the wrapping problem for color Doppler ultrasound data that we mentioned in the previous section is that the practically useful range for phase wrapping is $(-3\pi, 3\pi]$. This results in the fact that the pixels of the color Doppler image with original positive velocities can be classified into three categories:

- Unwrapped pixels: the phase data lies in the range $(0, \pi]$.

- Moderately-wrapped pixels: the wrapped phase data lies in the range $(-\pi, 0]$.
- Severely-wrapped pixels: the wrapped phase data lies in the range $(0, \pi]$.

Figure 5.3 shows the color map used for displaying color Doppler images along with the phase range of the three pixel categories. Figure 5.4 marks the three categories on a zoomed color Doppler image. As described in section 4.3.4 previously, the boundary lines that separate the unwrapped pixels from the moderately-wrapped pixels are called fringelines, and the boundary lines that separate the unwrapped pixels from the severely-wrapped pixels are called cutlines. The fringelines can be detected from the image, but the cutlines are difficult to detect. The complexity of detecting the cutlines emerges from the fact that the two categories of pixels share the same phase range $(0, \pi]$ in the wrapped phase image. However one important observation is that the unwrapped pixels tend to cluster around π, while the severely-wrapped pixels tend to cluster around 0, as shown in the Figure 5.4. This is a generic behavior with color Doppler ultrasound images, due to the continuity of the phase values from unwrapped to moderately-wrapped to severely-wrapped regions. The only source of discontinuity is the adjacency of the unwrapped and severely-wrapped regions.

Based on the previous observation, the logical threshold that can separate the two clustered regions (one around π and one around 0) is the $\pi/2$ threshold. But how do we find this threshold line?

The answer lies in the phase-shifted fringeline detection concept used by Chavez [75] and by Kramer [76]. If we introduce a phase shift of $\pi/2$ to the original color Doppler image and track the fringelines with Kramer's method, or equivalently use the residue annihilation process described by Chavez et al. to track the $\pi/2$ shift fringeline, then the resulting fringeline will represent a separation line between the two clusters, one around π and the other around 0. In other words, the hidden cutline in the original image that separates the unwrapped from the severely-wrapped phase regions is transformed into a fringeline in the phase-shifted image, and hence it becomes easy to track. Figure 5.5 shows the fringelines associated with the $\pi/2$ phase shift detected by the residue annihilation process. The regions

of phase discontinuity (unwrapped and severely-wrapped regions) are clearly separated by the fringelines.

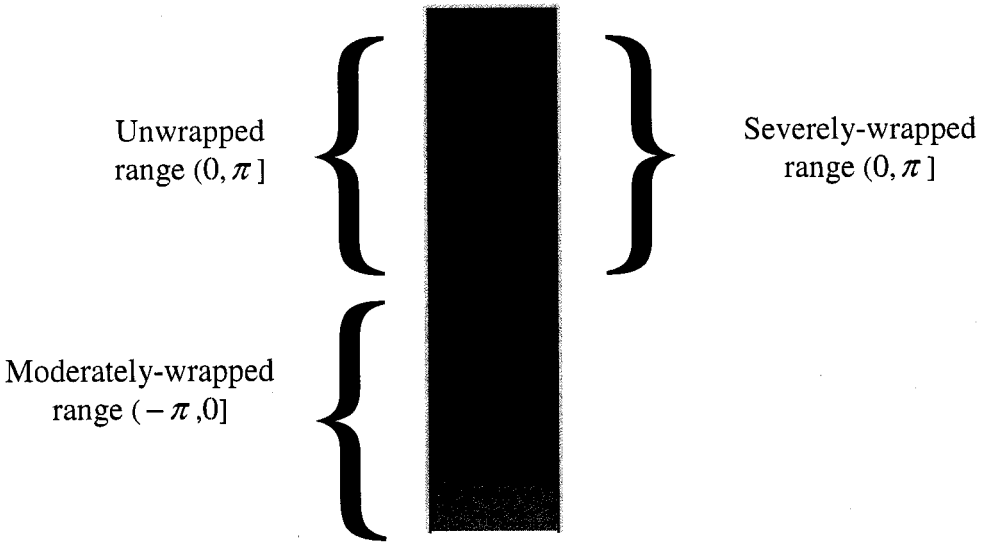


Figure 5.3 Phase range for the three pixel categories that exist in the color Doppler ultrasound images.

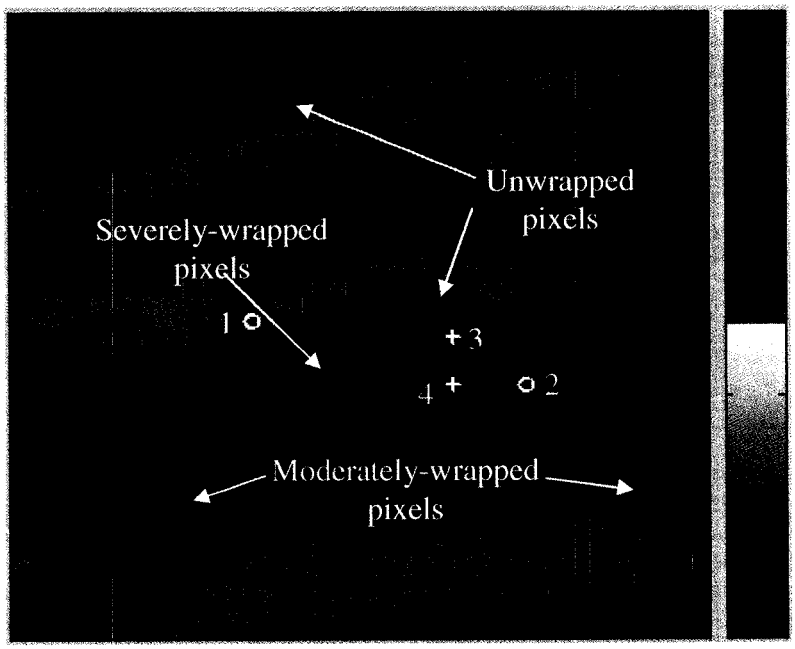


Figure 5.4 The three pixel categories indicated on a zoomed color Doppler image. The four residues are marked by numbers.

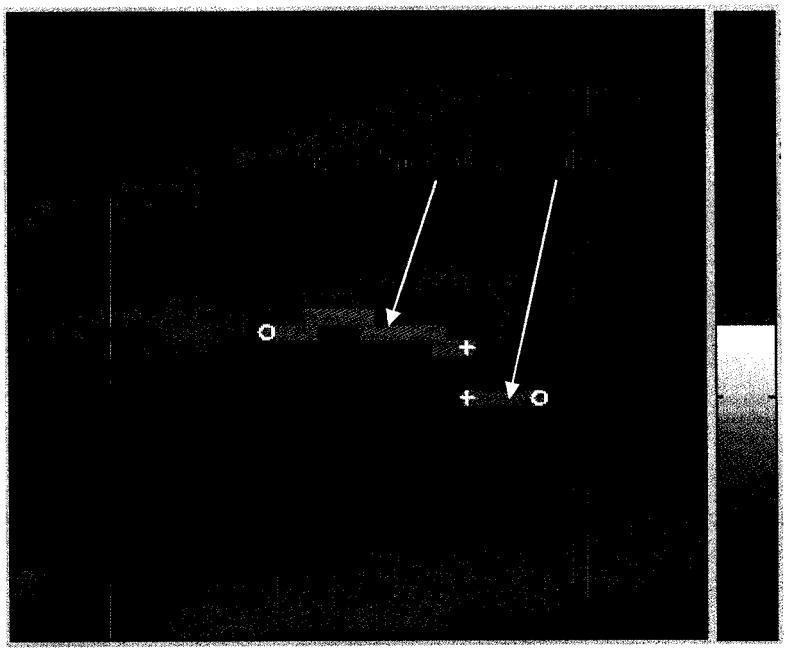


Figure 5.5 Fringelines of $\pi/2$ phase shift. The fringelines separate the unwrapped phase regions from the severely-wrapped regions.

The idea of using the fringelines associated with phase shifts of $\pi/2$ (or $-\pi/2$ for the wrapped negative-phase images) as an approximation to the hidden cutlines that separate the regions of phase discontinuity is the core concept behind our novel phase unwrapping technique. The fringelines themselves can be used as branch cuts, or they can be used as guidelines for the pairing of opposite sign residues. For example, the image in Figure 5.4 has two negative (1 & 2) and two positive (3 & 4) residues; the detected fringelines in Figure 5.5 can be used to guide the pairing of residues (1 & 3) and (2 & 4). In that case, the final branch cuts will be single-pixel straight lines, instead of the two-pixel curved lines of the original fringelines (produced by the annihilation process).

The idea behind the new unwrapping technique is simple; however, there are many open questions that need to be answered to guarantee a robust unwrapping algorithm:

- Which phase shift value, $\pi/2$ or $-\pi/2$, should be used? We will see later that this is not an obvious choice due to the many different waveforms and wrapping situations with ultrasound data.

- How robust is the fringeline tracking procedure? We will show how complicated the wrapping cases can get and what type of artifacts affect the robustness of the tracking procedure.
- How should the initial pixel to use for the unwrapping process be chosen? The initialization step is a very critical step that affects the results of the unwrapping process. We found that this step was often overlooked in the literature and the assumption used in most phase unwrapping techniques is that the initial unwrapped pixel was always given.

In the following section we give the details of the newly developed phase unwrapping algorithm for color Doppler ultrasound images.

5.4. A New Fringeline-Tracking Approach for Phase Unwrapping

The new algorithm has the following steps:

- *Flow direction detection*: this step is essential for determining the blood flow direction from the color Doppler cineloop data. The blood flow direction will be used later by many steps of the unwrapping algorithm.
- *Phase mask construction*: during this step, a mask is constructed to encompass the vessel lumen area only, excluding the tissue regions that do not have color Doppler data. Also a hole-filling algorithm is applied to the phase data to eliminate small gaps within the vessel lumen.
- *Residue detection*: in this step the residues within the phase image are detected.
- *Dipole detection*: this is the core step of the algorithm where the fringelines of the $\pi/2$ phase shift are tracked for each residue. The detected fringelines are then used to guide the pairing of opposite sign residues to construct residue dipoles. This process is repeated for a phase shift of $-\pi/2$.
- *Cutline construction and phase unwrapping*: the dipoles detected in the previous step are used to construct cutlines within the phase image. Then the phase data is unwrapped while avoiding crossing the cutlines. Two unwrapped images are constructed for the two phase shifts used: $\pi/2$ and $-\pi/2$.

- *Unwrapped image selection*: in this step heuristic rules are applied to choose one of the two unwrapped phase images for the two used phase shifts: $\pi/2$ and $-\pi/2$.

5.4.1. Flow Direction Detection

The first step in our algorithm is to detect the original blood flow direction from the captured cineloop. The flow direction will be used in many steps of the algorithm. The first aim of detecting the blood flow direction is to find an unwrapped pixel to be used for the initialization of the path-following unwrapping algorithm. The initial pixel to use for the unwrapping process has been always overlooked in the literature although it is absolutely critical to the success of any phase-unwrapping technique. The second aim of detecting the blood flow direction is to guide the selection of the proper phase shift ($\pi/2$ or $-\pi/2$) to use for the cutline detection, as we will describe later.

Determination of the flow direction is not an obvious task due to many factors. The phase wrapping problem is, by itself, the main challenge to this task where the phase data is distorted due to the wrapping process. Wrapped pixels will appear as if the blood flow is going into the reverse direction. The complexity of the flow patterns of the human blood vessels adds another dimension of difficulty to this task. The degree of pulsatility varies among arteries and veins; arteries are usually more pulsatile than veins. Even the pulsatility degree varies among arteries (or veins) depending on the proximity of the vessel to the heart and the size of the vessel. In section 4.4.1 we presented *in vitro* and *in vivo* examples of color Doppler ultrasound cineloops of different vessels with different pulsatility patterns. We have seen examples of the uniphasic Carotid waveform, the triphasic femoral waveform, and the steady venous flow. Our approach to solving this problem is to develop an image analysis algorithm that takes into consideration the temporal and spatial features of the blood flow that manifest themselves in color Doppler images.

The temporal feature that we use is the pulsatility feature that exists in all peripheral arteries and some major veins which are closer to the heart, such as the jugular vein. For a

pulsatile vessel, the blood velocity starts with the lowest values at the diastole phase and ramps up to its highest values during the systole phase. If a phase wrapping occurs due to low sampling rates, the distortion in the flow direction will mainly occur in the systole frames, and maybe in some adjacent frames if the aliasing is severe. However by moving away from the systole frame toward the diastole frame, the phase wrapping will diminish and the flow direction is reserved. See Figure 4.17, Figure 4.21, Figure 4.25, and Figure 4.27 for examples of pulsatile vessels with phase wrapping. We used this physiological property in our algorithm by considering a number of adjacent frames around the frame of interest to decide the flow direction for that frame.

The pulsatility feature alone will not work for steady flow veins, such as the phantom-simulated steady flow vein shown in Figure 4.19. When phase wrapping occurs with these veins, the wrapping shows up in all frames. However there is another physiological property that we can rely on to detect the flow direction: the cross-sectional velocity profile. The velocity profile can be modeled by a parabolic function across the vessel diameter for a steady laminar flow [14]. Due to the vessel wall friction, the blood velocity is almost zero at the vessel wall and starts to increase as we move inward until it reaches its maximum value at the vessel centerline, as indicated by the following equation and shown in a schematic diagram in Figure 5.6.

$$V(r) = (1 - r^2 / R^2) V_{\max} \qquad (51)$$

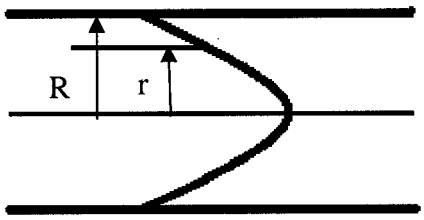


Figure 5.6 A schematic diagram of the parabolic velocity profile across the vessel cross section.

When phase wrapping occurs in steady flow vessels, the wrapped pixels tend to center around the vessel centerline, while the regions near the vessel wall will not have much phase wrapping; this is evident in Figure 4.19. We incorporated this important spatial feature in our flow direction detection algorithm by giving more weight to the pixels near the vessel wall to determine the flow direction, since they tend to be less corrupted by the phase wrapping process. The following pseudo-code summarizes the steps involved in the flow direction detection algorithm:

```
For each frame (i) in a color Doppler cineloop of N frames
{
  For each frame in the adjacent neighborhood of frame i (i-n to i+n)
  {
    1- Calculate a binary mask of all phase pixels.
    2- Apply a hole filling algorithm to fill in small gaps in the phase mask.
    3- Calculate the distance transform of the phase mask.
    4- Calculate the maximum value of the distance transform map.
    5- Calculate the inverse distance transform by subtracting the distance transform values
       from the maximum value.
    6- Calculate the weighted sum of all positive and all negative phase pixels. The inverse
       distance transform values are used as weights.
    7- Accumulate the weighted velocity sum for all frames in the neighborhood.
  }
  If (accumulated weighted positive phase sum > accumulated weighted negative phase sum)
     flow direction of frame (i) is positive.
  Else
     flow direction of frame (i) is negative.
}
```

Figure 5.7 Pseudo-code for the flow direction detection algorithm.

The flow direction needs to be calculated for each frame in the acquired phase cineloop. This is especially important for vessels with triphasic flow where the flow direction changes during one heart cycle, as shown in Figure 4.21 for the femoral flow. The first step of

the algorithm is to consider a neighborhood of each frame to utilize the temporal aspect of the pulsatility property as explained above. The number of adjacent frames is critical for the triphasic flow, since we do not want to mix frames with different flow direction. Ideally it should a function of the heart rate or cycle but we found that considering two frames before and two frames after the current frame is good enough for all the data sets we have.

The next step is to construct a binary mask for each frame in the neighborhood. The binary mask has true values for all non-zero pixels in the color Doppler velocity frame, and has false values outside. Figure 5.8 shows a cineloop of 30 color Doppler frames of a flow phantom simulation of a carotid waveform. Phase wrapping occurs in the systolic frames (frames 3-7). Figure 5.9a shows one systole frame from the cineloop; the detected residues are also shown. Figure 5.9b shows the binary flow mask; a careful look at this mask indicates some gaps within the vessel lumen. These gaps will affect the quality of the distance transform that we will use later. So it is essential to apply a hole filling technique to close the gaps. We used a binary morphological operation, the closing operation, to fuse the small gaps within the vessel lumen. Figure 5.9c shows the binary mask after applying the closing operation to fuse the gaps.

The next step is to construct a weighting mask that favors the pixels near the vessel boundary to benefit from the spatial property of the velocity profile and give more weight to the unwrapped pixels near the vessel boundary. For that purpose we use a classical image analysis technique, called the distance transform. For each white pixel in the flow mask, the distance transform assigns a number that is the distance between that pixel and the nearest black, or boundary pixel of the mask. We used the built-in MATLAB© [79] distance transform function, which implements the algorithm described in [80]. Figure 5.9d shows the distance transform of the binary flow mask. The distance transform gives more weight to the inner pixels closer to the centerline of the mask; instead we are interested in giving more weight to the border pixels. To achieve this goal, we simply invert the distance transform by subtracting the transform values from the maximum value within the distance transform. Figure 5.9e shows the inverse distance transform where the border pixels have more weight than the inner ones.

The inverse distance transform is used as a weight mask for the original phase image shown in Figure 5.9a. Two weighted sums are calculated, one for the positive phase pixels (the blue pixels in Figure 5.9a) and one for the negative ones (the red pixels in Figure 5.9a). The positive and negative weighted sums are accumulated across all neighboring five frames. Then the decision about the flow direction is simply made based on comparing the two accumulated weighted sums; the larger sum dictates the flow direction for the current frame. To give a numeric example, the frame in Figure 5.9 has a positive accumulated weighted sum of 9.6×10^5 and a negative accumulated weighted sum of 3.2×10^5. Hence the flow direction for this frame will be positive (i.e. blue). We see in this example how robust the algorithm is where the five neighboring frames were wrapped (frames 3-7) with many distorted pixels. However the temporal and spatial heuristics used by the algorithm favored the correct flow direction of this frame.

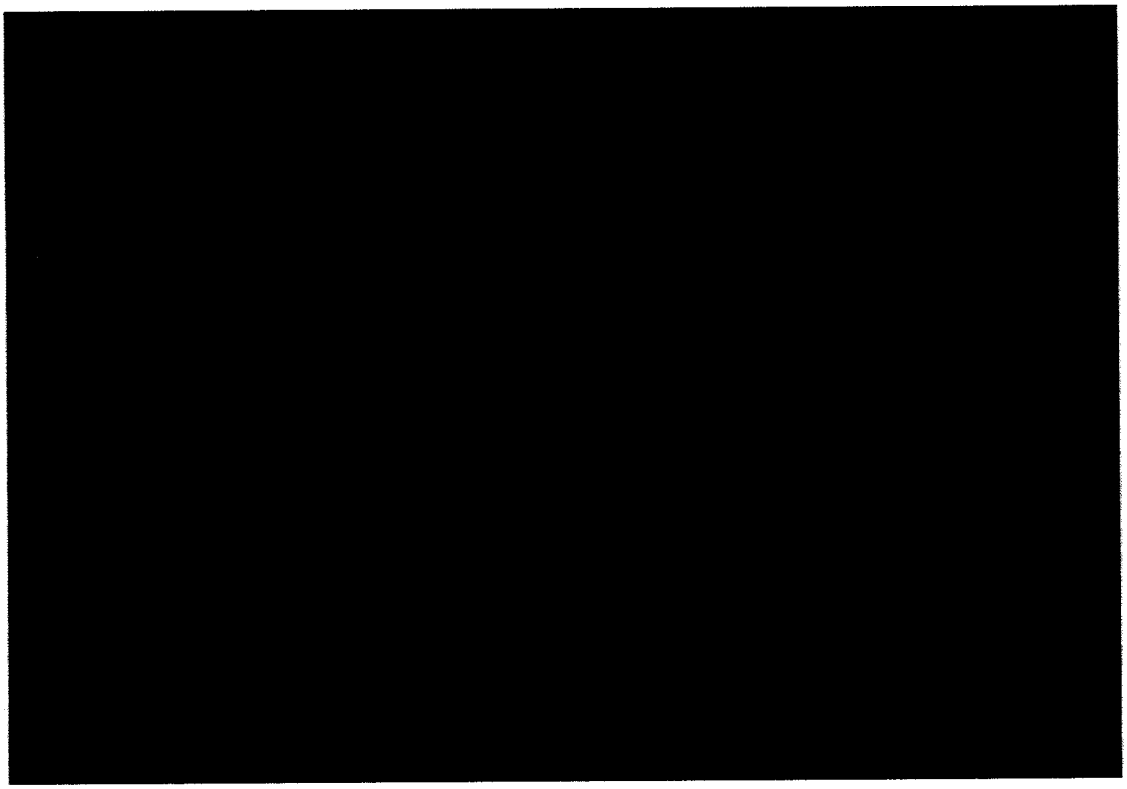
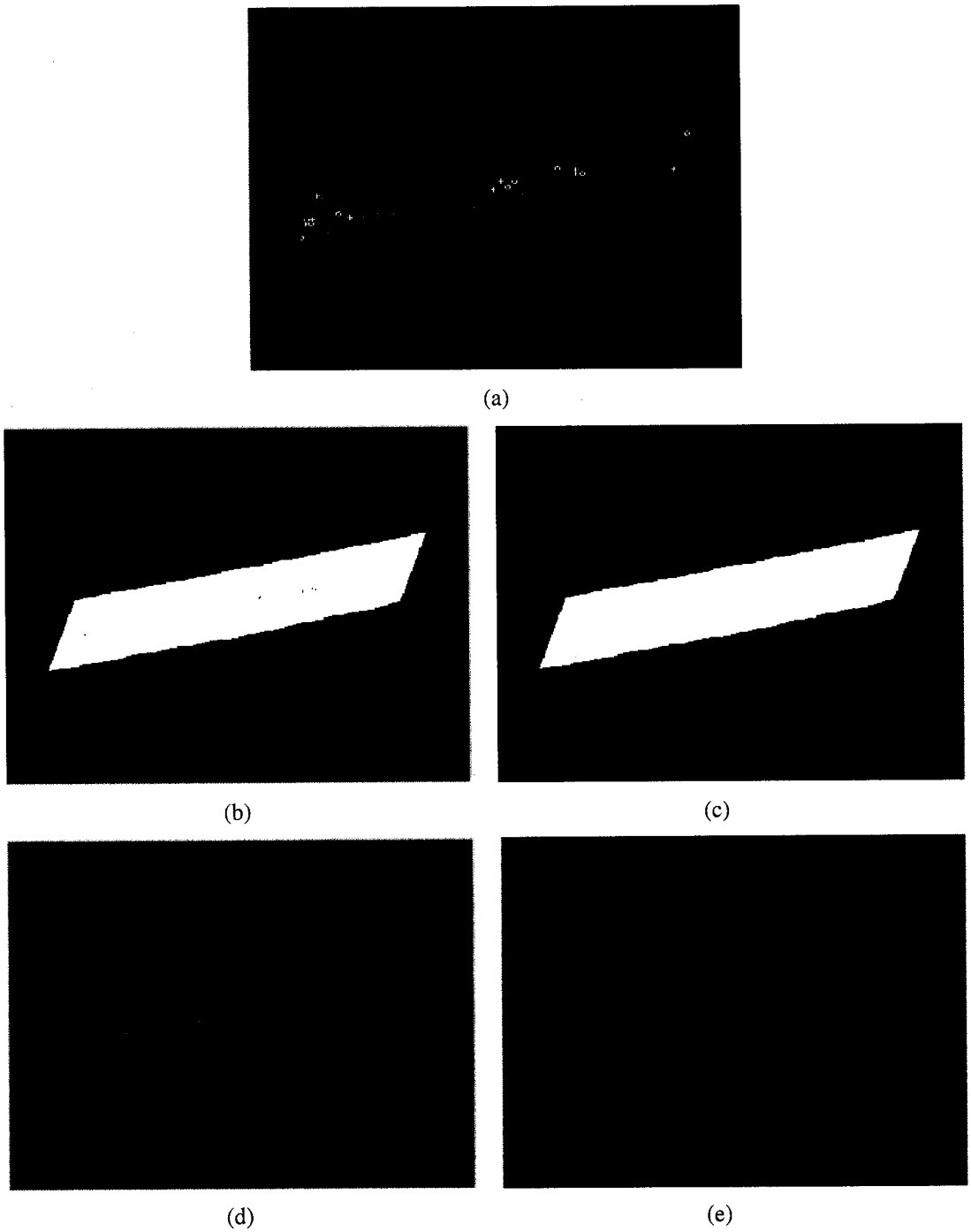


Figure 5.8 A simulated carotid waveform from a flow phantom. A cineloop of 30 color Doppler frames is overlaid on top of 30 grayscale frames and displayed row-wise. Phase wrapping is present in the systole frames (frames 3 to 7).



(a)

(b)

(c)

(d)

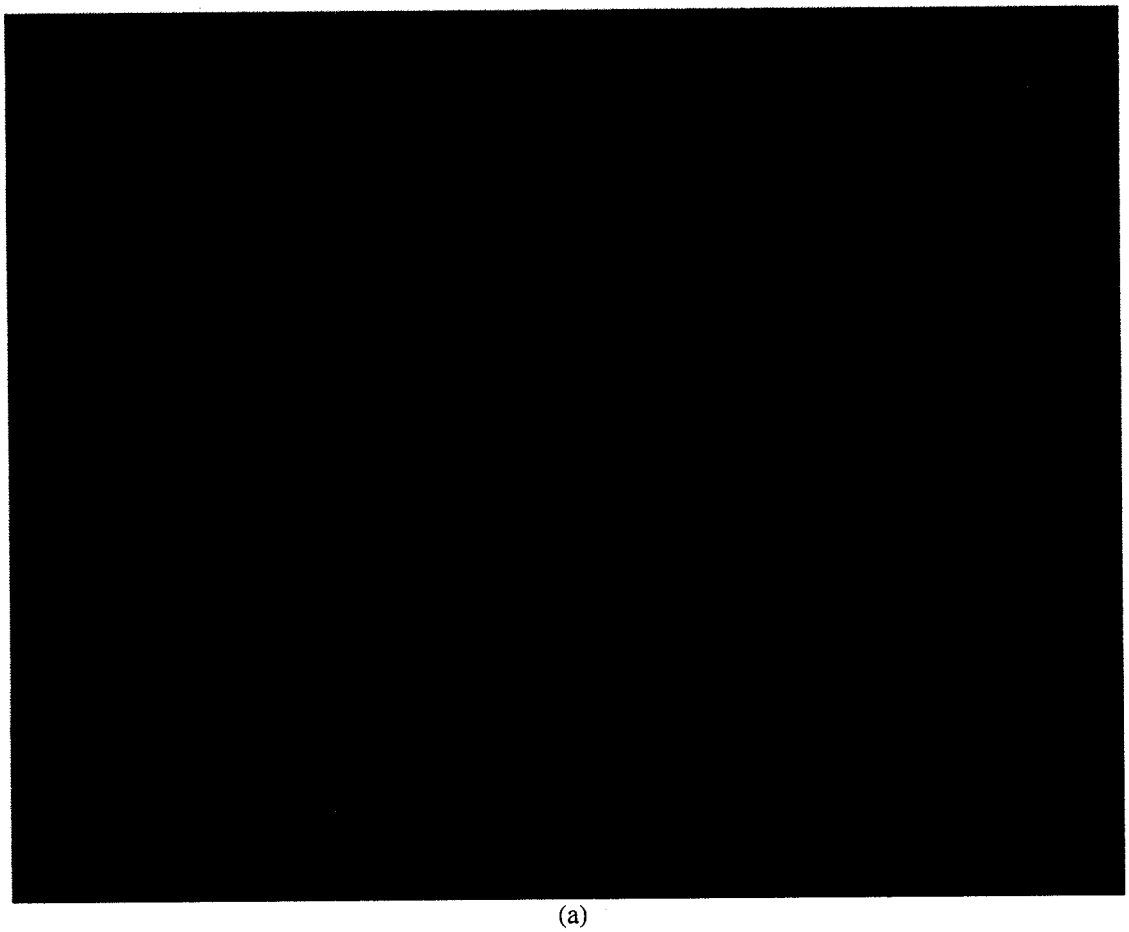
(e)

Figure 5.9 The steps of the flow direction detection algorithm. (a) The original frame of interest. (b) The binary flow mask. (c) The binary flow mask with gap filling. (d) The distance transform of the flow mask. (e) The inverse distance transform.

We tried our flow direction detection algorithm on all *in vitro* and *in vivo* datasets we have and the success rate was above 99%. Figure 5.10a shows an example for a cineloop of a simulated uniphasic carotid waveform from a flow phantom. Figure 5.10b shows the results of the flow direction detection algorithm in a tabular form, where a (+) sign indicates a positive direction or flow towards the transducer and a (-) sign indicates a negative direction or flow away from the transducer. We notice specifically the correct estimated flow direction for the frames with severe phase wrapping (frames 5 to 9). Figure 5.11 shows another successful example of a simulated steady venous flow with severe phase wrapping in all frames; the flow direction was successfully estimated for all frames. Figure 5.12 shows an example of a simulated triphasic femoral waveform, which is the most challenging since the flow direction changes during one heart cycle. Despite the difficulty, the algorithm was able to detect the flow direction successfully for all frames.

For the few failed cases from our datasets, there were two root causes of the failure. The first one is the boundary conditions where the first or last frame is misestimated. This is due to the fact that there are not enough neighboring frames before or after the first or last frame respectively. This problem can be easily overcome by capturing a few frames before and after the heart cycle of interest to use as padding frames for this algorithm. The other cause of failure is the change of direction with the triphasic flow. The transitional frame that occurred during the reversal of the flow from positive to negative (or vice versa) is prone to erroneous direction estimation. A simple median or averaging filter can be applied to the consecutive frame directions to remove these sudden outlier directions. Figure 5.13 shows an example of a failed case of a simulated triphasic femoral waveform. The fourth frame failed because it was a transitional frame from positive to negative direction flow. However the transitional frame usually contains low velocity content that is not wrapped, thus the erroneous flow direction should not affect the phase-unwrapping algorithm.

Figure 5.14 and Figure 5.15 shows successful cases for the *in vivo* carotid artery and jugular vein respectively. The flow direction in all frames with wrapped phase data is correctly estimated.

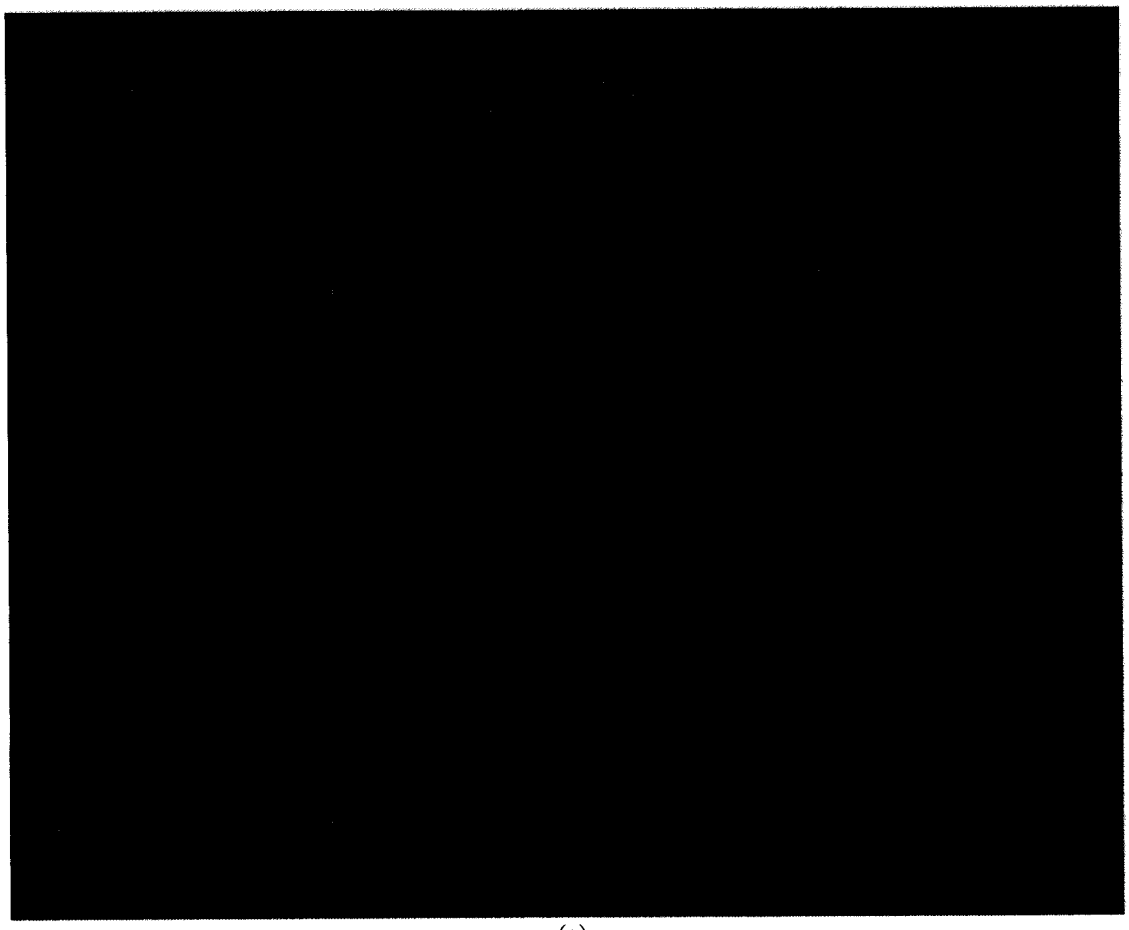


(a)

+	+	+	+	+
+	+	+	+	+
+	+	+	+	+
+	+	+	+	+
+	+	+	+	+

(b)

Figure 5.10 Flow direction estimation results for a simulated uniphasic carotid waveform. (a) A color Doppler cineloop of 30 frames. (b) The estimated flow direction for each frame. The flow direction of all frames is correctly estimated.

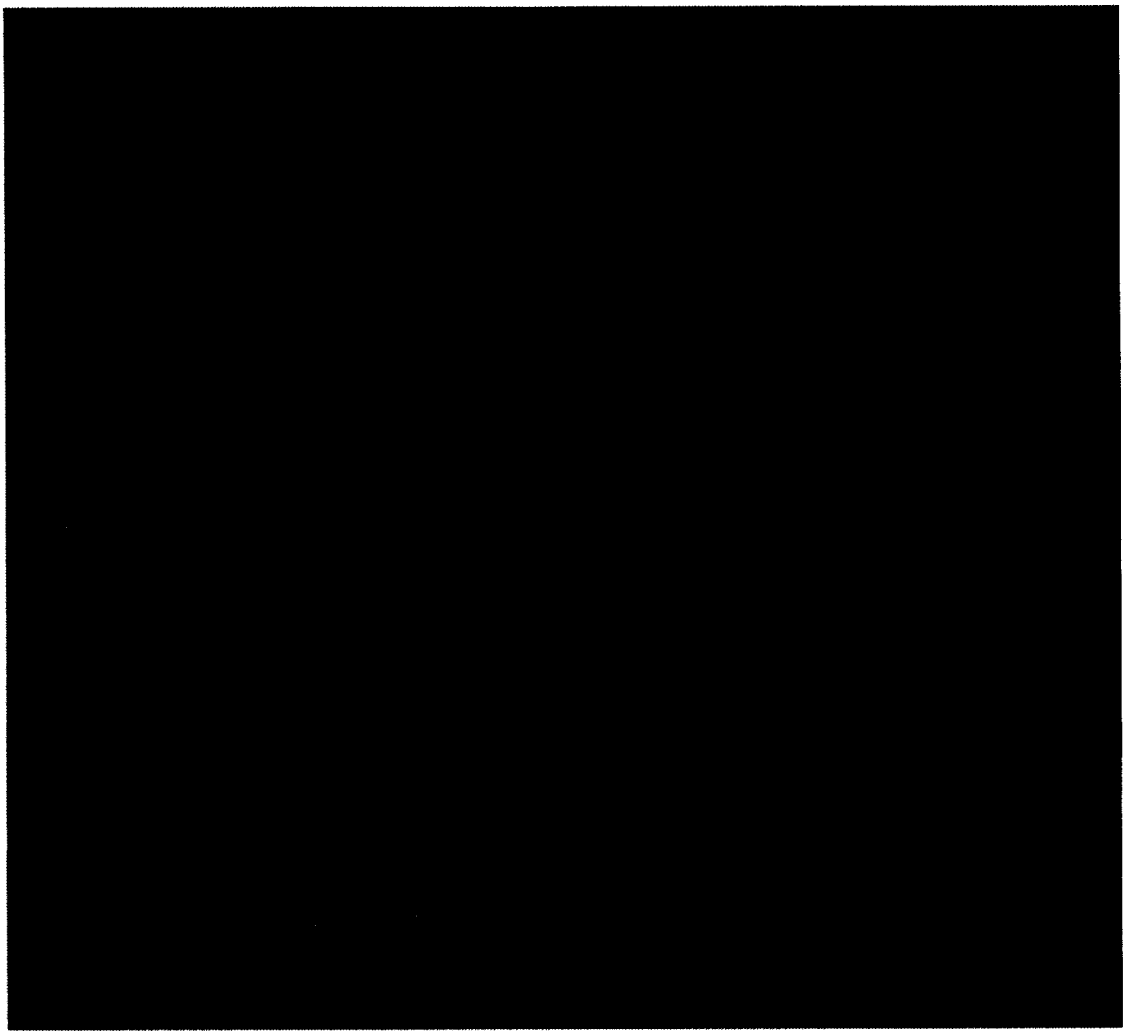


(a)

+	+	+	+	+
+	+	+	+	+
+	+	+	+	+
+	+	+	+	+
+	+	+	+	+

(b)

Figure 5.11 Flow direction estimation results for a simulated steady venous waveform. (a) A color Doppler cineloop of 30 frames. (b) The estimated flow direction for each frame. The flow direction of all frames is correctly estimated.

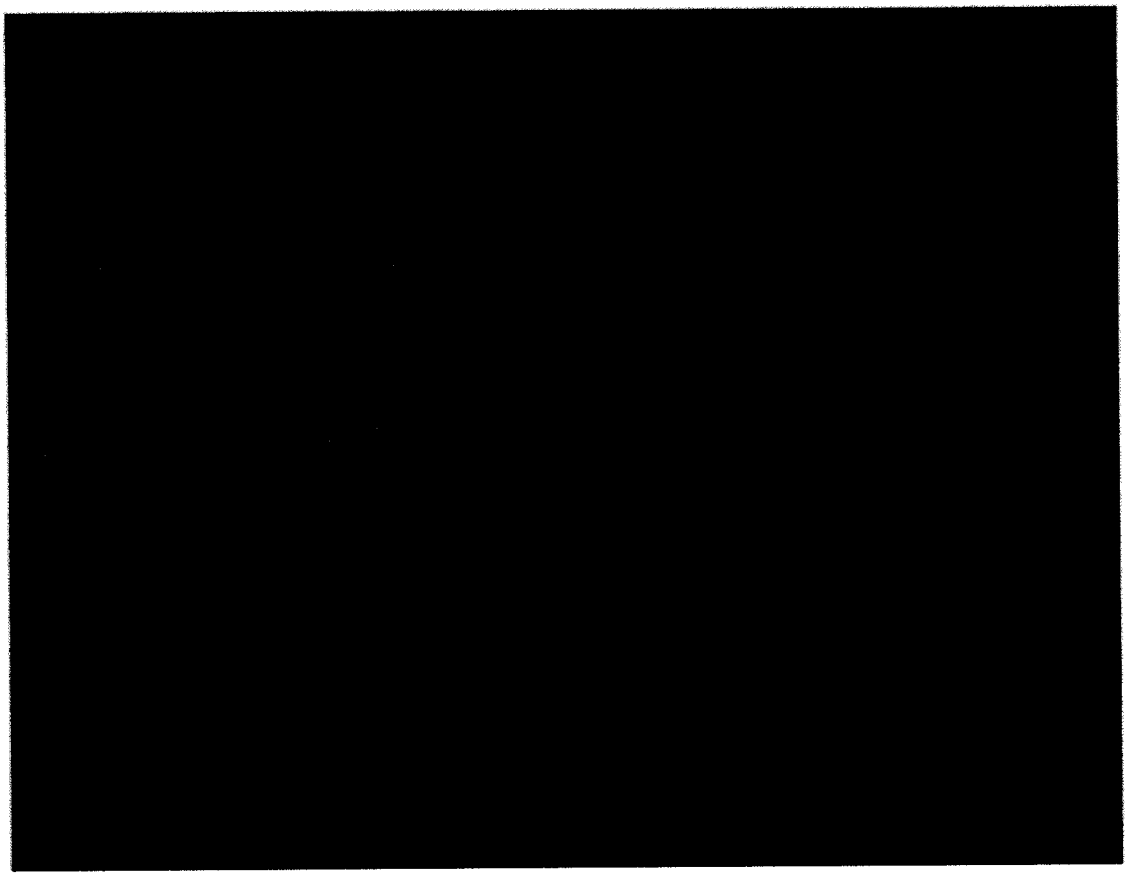


(a)

+	+	+	+	+
-	-	-	-	-
-	-	-	-	-
+	+	+	+	+
+	+	+	+	+

(b)

Figure 5.12 Flow direction estimation results for a simulated triphasic femoral waveform with non-optimal Doppler steering angle. (a) A color Doppler cineloop of 30 frames. (b) The estimated flow direction for each frame. The flow direction of all frames is correctly estimated.

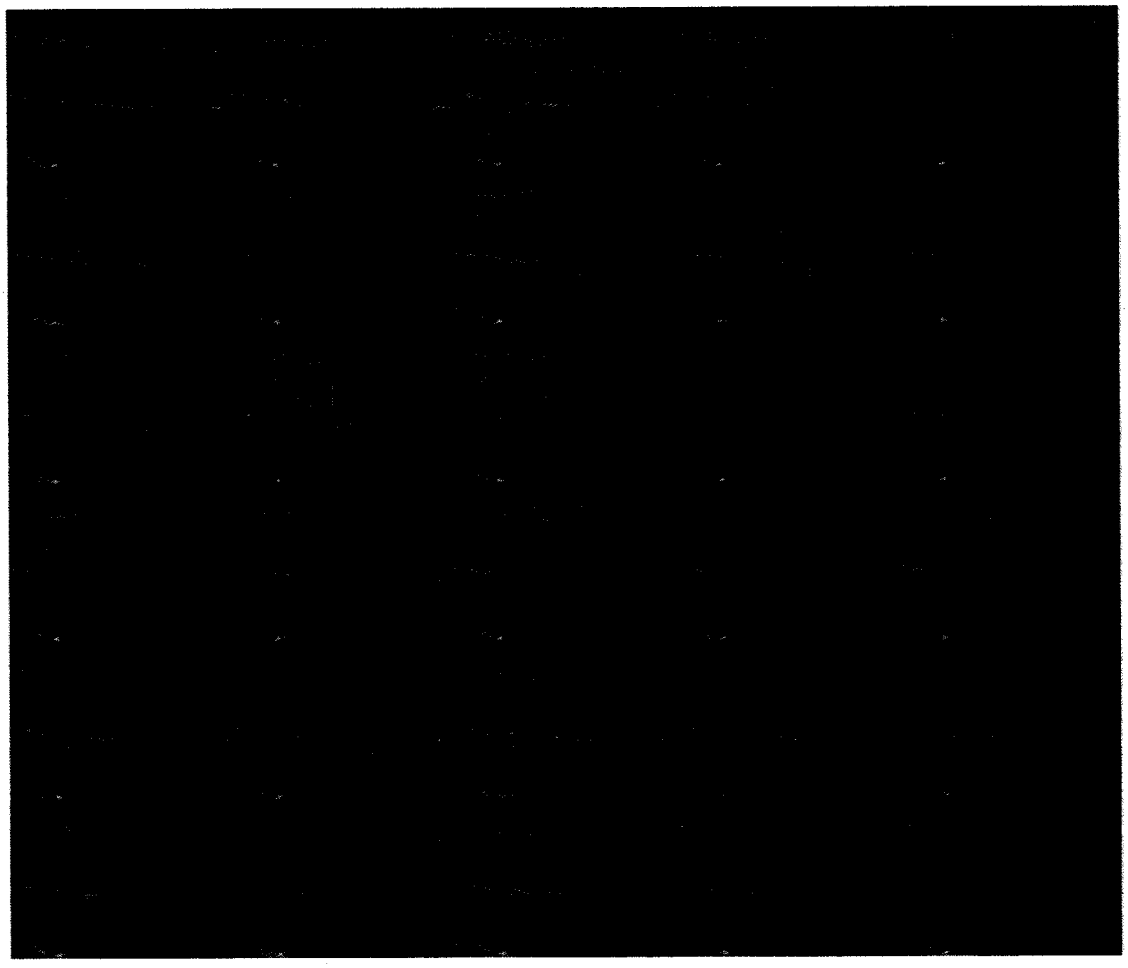


(a)

+	+	+	+	-
-	-	-	-	-
-	-	-	+	+
+	+	+	+	+
+	+	+	+	+

(b)

Figure 5.13 Flow direction estimation results for a simulated triphasic femoral waveform. (a) A color Doppler cineloop of 30 frames. (b) The estimated flow direction for each frame. The incorrectly estimated frame is marked by a red circle.

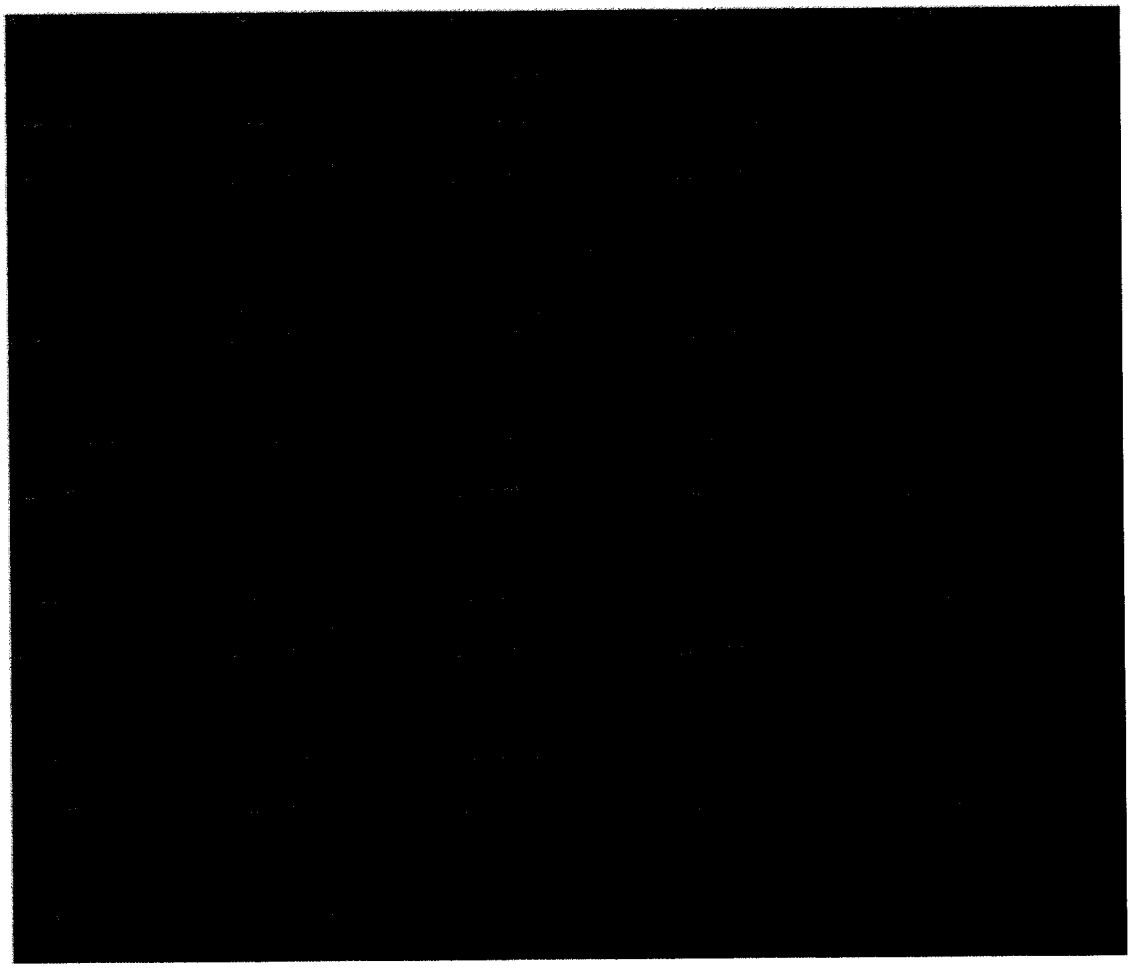


(a)

+	+	+	+	+
+	+	+	+	+
+	+	+	+	+
+	+	+	+	+
+	+	+	+	+

(b)

Figure 5.14 Flow direction estimation results for an *in vivo* uniphasic carotid waveform. (a) A color Doppler cineloop of 30 frames. (b) The estimated flow direction for each frame. The flow direction of all frames is correctly estimated.



(a)

-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

(b)

Figure 5.15 Flow direction estimation results for an *in vivo* uniphasic jugular vein waveform. (a) A color Doppler cineloop of 30 frames. (b) The estimated flow direction for each frame. The flow direction of all frames is correctly estimated.

5.4.2. Phase Mask Construction

The next step after estimating the flow direction is to construct a binary mask for the color Doppler image. There are many purposes for this mask. The first one is to use the mask to exclude the residues that lie on the border of the vessel to simplify the residue pairing process. The second purpose of the mask is to connect some residues that lie closer to the vessel border to that border, and exclude them from the residue pairing step. The last purpose of the mask is to exclude the background pixels during the phase unwrapping process, and thus speeding up the execution time by only focusing on the true phase data within the vessel lumen, instead of the whole image.

There are three main steps in the construction of the binary mask, two of which we have used previously in the flow direction detection step. The first step is to threshold the phase image to construct a binary mask for the non-zero phase values. All background pixels with zero phase value will be masked out this way. However, some small real phase regions with zero phase value will also be masked out. These regions occur with severe wrapping cases with more than π wrapping, where the wrapped phase value crosses the zero phase value, as shown in Figure 5.9. The binary mask in Figure 5.9b excludes all background pixels but also has gaps within the vessel lumen that coincide with the regions of severe phase wrapping in Figure 5.9a.

To include the phase regions of these gaps in the phase unwrapping process, we apply a hole-filling algorithm. The hole-filling algorithm we use is a binary morphological closing operation with a 3×3 square structure element. Figure 5.9b shows the processed mask with the closing operation where the small gaps within the vessel lumen are fused. It is important to mention that the gaps could be larger in size than 3×3 and they may even represent true anatomical structure for tortuous vessels. In this case, the gaps will not be completely fused by the morphological closing operation. The effect of this problem will be the inability to unwrap the gap regions that are masked out; this should not have a major effect on our ultimate goal of quantifying the color Doppler data. We experimented with a more aggressive hole-filling

algorithm that is adaptive to the gap size but the side effect of fusing true anatomical gaps, not caused by the wrapping process, had more negative impact on the quality of the unwrapping results.

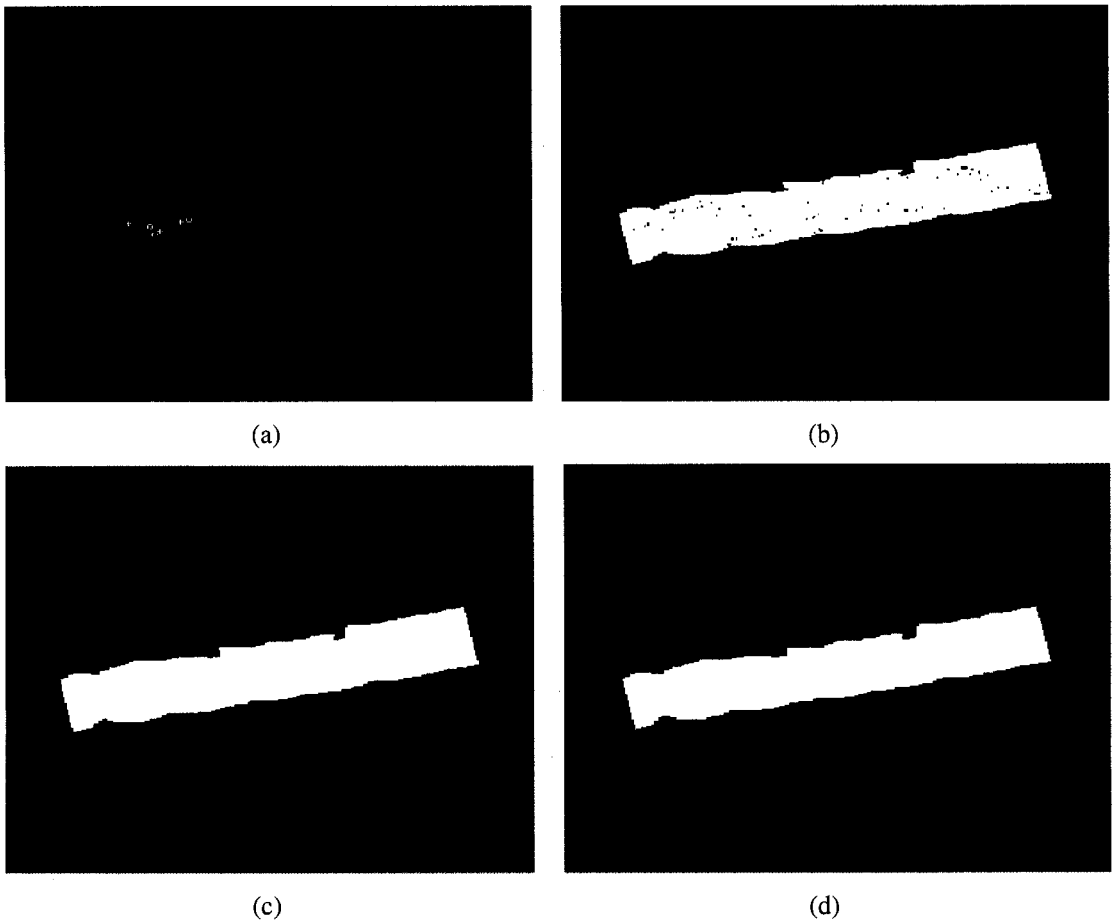
The final step of the mask construction process is to shrink the border of the binary mask, or in other words, fatten the background. The reason for this step is to exclude the residues that are very close to the vessel border during the cutline detection step. This is an important simplification step, since the vessel border discontinuity usually causes many residues that do not have a logical counterpart within the vessel lumen. Thus these residues complicate the dipole detection step. To achieve this step, we apply a morphological dilation process to the inverse of the binary mask or use the "fatten mask" operation described in [60], which is very similar to an inverse dilation operation.

To give some examples of the binary mask construction process, Figure 5.16 shows the mask associated with a simulated femoral flow with phase wrapping and a non-optimal Doppler steering angle, which causes flow direction ambiguity that manifests itself in the opposite flow direction blobs, as shown in Figure 5.16a. Figure 5.16b shows the raw binary mask of the vessel lumen area; we can easily see many small gaps within the vessel lumen. The gaps in this case are mainly caused by the adjacency of the opposite flow direction regions due to the non-optimal Doppler angle ambiguity. Figure 5.16c shows the processed mask after the gap-filling operation; all the gaps inside the vessel lumen are fused. Figure 5.16d shows the mask after the shrinking operation of one pixel wide layer from the vessel boundary.

Figure 5.17a shows another example of an *in vivo* carotid flow with severe wrapping. Figure 5.17b shows the raw binary mask with many gaps caused by the severe phase wrapping. Figure 5.17c shows the gap-filled mask, and Figure 5.17d shows the shrunk mask.

5.4.3. Residue Detection

The next step after constructing the binary mask is to detect the residues inside the phase image. We apply the classical residue detection algorithm that scans the phase image and checks for phase discontinuity in every 2×2 square within the image. The algorithm was previously described in section 4.2 and summarized as pseudo-code in Figure 4.9. The detected residues are restricted to the vessel lumen area defined by the binary mask. If the top left pixel of the 2×2 square under consideration is masked out, the square is skipped for residue detection.



(a) (b)

(c) (d)

Figure 5.16 Binary phase mask construction. (a) A color Doppler image of a simulated femoral flow. (b) The raw binary mask with many gaps within the vessel lumen. (c) The mask with hole-filling operation. (d) The shrunk mask with a one pixel layer removed from the mask boundary.

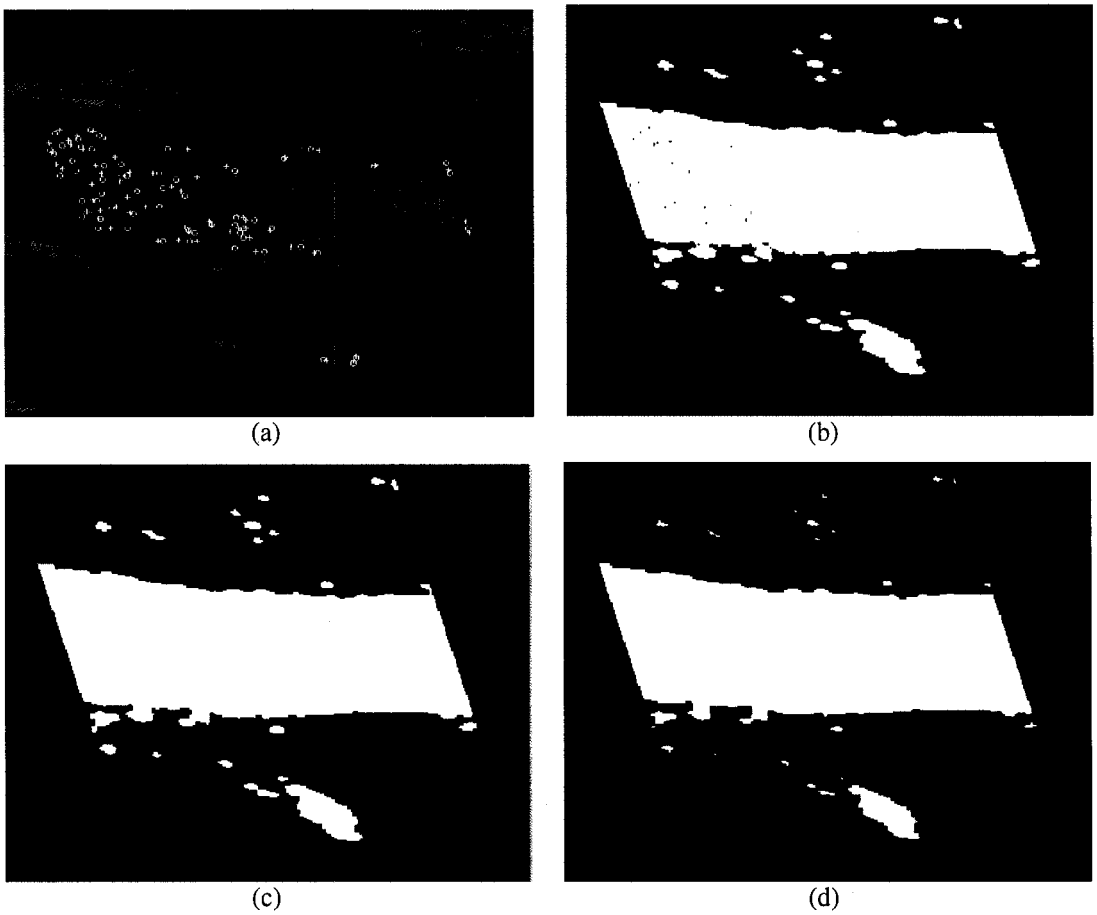


Figure 5.17 Binary phase mask construction. (a) A color Doppler image of an *in vivo* carotid flow. (b) The raw binary mask with many gaps within the vessel lumen. (c) The mask with hole-filling operation. (d) The shrunk mask with a one pixel layer removed from the mask boundary.

5.4.4. Dipole Detection

The dipole detection is the core step of the new phase unwrapping technique. The main idea, as described in section 5.3, is to use the fringeline tracking process to guide the pairing of the opposite sign residues to construct cutlines. The detected cutlines should be very close to the true phase discontinuity lines that separate the unwrapped phase regions from the severely-wrapped ones. Despite the simplicity of the main idea, there are some challenges that can affect the correctness and robustness of the detected cutlines. Our algorithm will have to take these challenges into consideration to guarantee accurate and robust results.

The first problem is how to decide which phase shift to use for the fringeline tracking process. As explained previously, a phase shift of either $\pi/2$ or $-\pi/2$ will result in the correct detection of the sought cutlines. Figure 5.18 shows different phase wrapping cases, which requires different optimal phase shifts to use. Figure 5.18a shows a simulated carotid waveform with a positive flow direction (blue). The moderately-wrapped pixels are shown in red giving a false impression of a reverse flow, while the severely-wrapped are shown in dark blue. The optimal phase shift to use for the fringeline tracking to separate the unwrapped pixels from the severely-wrapped ones in Figure 5.18a is $\pi/2$. Figure 5.18b shows another example of a simulated constant venous flow with a positive flow direction. The optimal phase shift to separate the unwrapped (light blue) from the severely-wrapped (dark blue) regions in Figure 5.18b is $\pi/2$. Figure 5.18c shows a simulated femoral waveform with a negative flow direction (red). The optimal phase shift to separate the unwrapped (light red) from the severely-wrapped (dark red) regions in Figure 5.18c is $-\pi/2$. Figure 5.18d shows another frame of the same femoral waveform, but with a positive flow direction (blue) after the flow has reversed its direction during the same heart cycle of this triphasic waveform. The optimal phase shift to separate the unwrapped (light blue) from the severely-wrapped (dark blue) regions in Figure 5.18d is $\pi/2$. Figure 5.18e shows a simulated femoral waveform with a negative flow direction (red) and a non-optimal Doppler steering angle. The non-optimal steering results in a flow direction ambiguity and the coexistence of both forward and reverse flow directions. In this case there are only unwrapped pixels (dark red and dark blue) and moderately-wrapped pixels (light blue). The optimal phase shift to separate the unwrapped positive (dark blue) from the moderately-wrapped (light blue) regions in Figure 5.18e is $\pi/2$. Figure 5.18f shows another frame of the same femoral waveform with the non-optimal Doppler steering angle but with a positive flow direction (blue) after the flow has reversed its direction during the same heart cycle of this triphasic waveform. In this case, the optimal phase shift to separate the unwrapped (dark red) from the moderately-wrapped (light red) regions is $-\pi/2$. We see with these examples how difficult it is to predict the optimal phase shift to use due to the many different potential scenarios.

Our approach is to use both phase shifts to produce two different cutline sets and two different unwrapping solutions. Then a post-processing step will apply different criteria to select one of the two produced solutions, we will discuss the solution selection step in the next section.

Another problem with the fringeline tracking process is the occasional failure of a certain residue's fringeline to track the corresponding opposite sign residue. Normally the annihilation process starts with a specific residue and sequentially shifts the residue to a neighboring 2×2 square by replacing the original 2×2 residue square with zeros. Eventually the process ends when the shifted residue meets with an opposite sign residue, and the two residues balance each other. In this case the fringeline extends between two opposite sign residues. However due to distortion in the phase image, the fringeline could lose its track and stray away from the other opposite sign residue. The distortion could be caused by noise or by the proximity of nearby residues, which may confuse the tracking process. Chavez et al. [75] suggested scaling up the original image to distance the residues from each other and minimize the tracking distortion. However this idea will increase the time of the fringeline tracking process, most often unnecessarily. We chose to account for the tracking distortion within the logic of the dipole detection, or the opposite residue pairing, process itself.

Another problem with the dipole detection process is the proximity of some residues to the vessel border. Figure 5.19 shows two examples of *in vivo* proximal carotid flow with many residues along the vessel border. For those residues, the opposite-sign residue is most probably absent from the image. Another problem with the residues closer to the vessel border is the low velocity values that exist near the vessel border due to the friction caused by the vessel walls. As explained previously, the cross sectional velocity profile is typically a parabolic function that starts with low velocity near the vessel wall and increases as we get closer to the vessel center where the velocity reaches its maximum value. The small velocity layer near the vessel wall causes the fringelines of the border residues to deviate from the corresponding opposite sign residue. Our algorithm has a special handling procedure for the border residues.

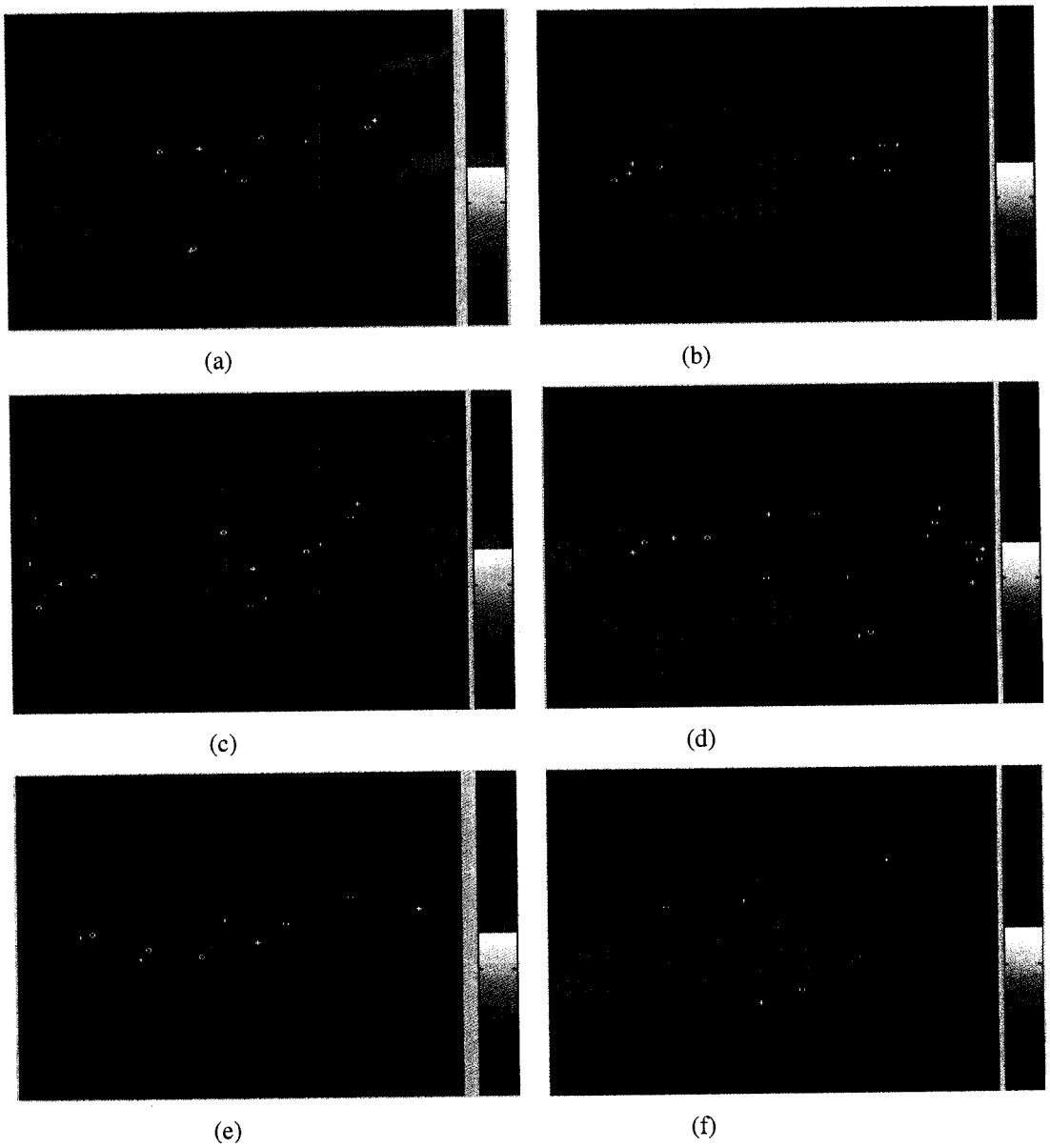


Figure 5.18 Different phase-wrapping cases with different optimal phase shifts for the unwrapping. (a) A simulated carotid waveform with $\pi/2$ optimal phase shift. (b) A simulated constant venous waveform with $\pi/2$ optimal phase shift. (c) A simulated femoral waveform with $-\pi/2$ optimal phase shift. (d) A simulated femoral waveform with $\pi/2$ optimal phase shift. (e) A simulated femoral waveform with non-optimal Doppler steering angle with $\pi/2$ optimal phase shift. (f) A simulated femoral waveform with non-optimal Doppler steering angle with $-\pi/2$ optimal phase shift.

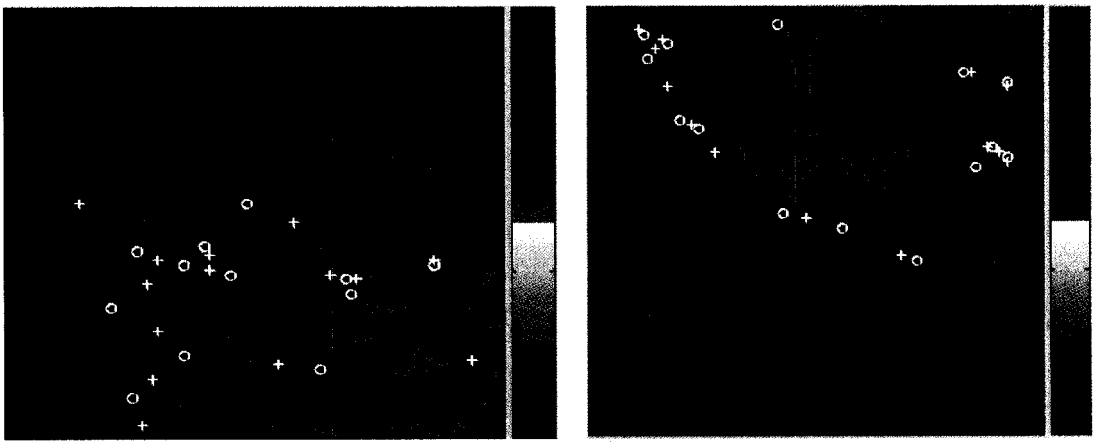


Figure 5.19 *In vivo* proximal carotid cases with many border residues.

The last problem is the possibility of finding some orphan residues, or residues which do not have corresponding opposite-sign residue to form a valid dipole. This could happen because of a missing residue outside the vessel lumen, or because of noise distortion. Figure 5.20 shows an example of an orphan residue in an *in vivo* proximal carotid flow. The missing residue that would pair with this orphan residue is outside the vessel lumen boundary, as indicated by arrows in the figure. Our algorithm will account for those orphan residues and guarantee the balance of all residues with cutlines to prevent phase unwrapping errors.

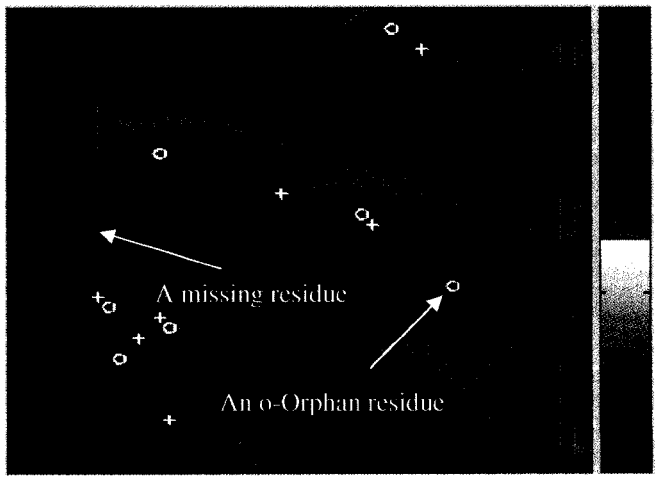


Figure 5.20 An *in vivo* proximal carotid case with an orphan residue.

The following pseudo-code, which will be followed by a detailed description of each step involved, summarizes the dipole detection algorithm.

1- Detect residues that are closer to the vessel border.
2- Detect residues that touch the vessel border.
3- Track fringelines of phase shift $\pi/2$ (or $-\pi/2$) and save the destination pixel for each residue.
4- Track fringelines of phase shift π and save the destination pixel for each residue.
5- Find all valid dipoles using phase shift of $\pi/2$ (or $-\pi/2$) for inner residues.
6- Find all valid dipoles using phase shift of π for the residues closer to the vessel border.
7- Connect the remaining residues to their nearest destinations.
8- Connect residues closer to the vessel border to the border itself.

Figure 5.21 Pseudo-code for the dipole detection algorithm

The first step of the dipole detection algorithm is to detect residues that are closer to the vessel boundary; we call these residues border-proximal residues. This is necessary for the special handling of the border residues within the residue pairing step. To detect these border-proximal residues, we calculate the Euclidean distance between each residue and the vessel boundary using the binary mask constructed in a previous step. Then we compare these distances with a predefined threshold that defines the proximity to the vessel border. The value of this threshold was derived empirically. The exact value of the threshold is not so critical but ideally it should be a function of the vessel diameter. Figure 5.22 shows examples of some border-proximal residues with an *in vivo* carotid flow.

The second step of the algorithm is to detect residues that touch the vessel border, which we call border-touching residues. This class of residues will also have special handling later, which is different than those residues which are closer to the border but not touching it. To detect the touching residues we simply check the four horizontal and vertical neighboring pixels to each residue and if at least one of them is a background pixel, with zero mask value, then that residue is flagged as a touching residue. Figure 5.22 shows examples of some border-touching residues with an *in vivo* carotid flow.

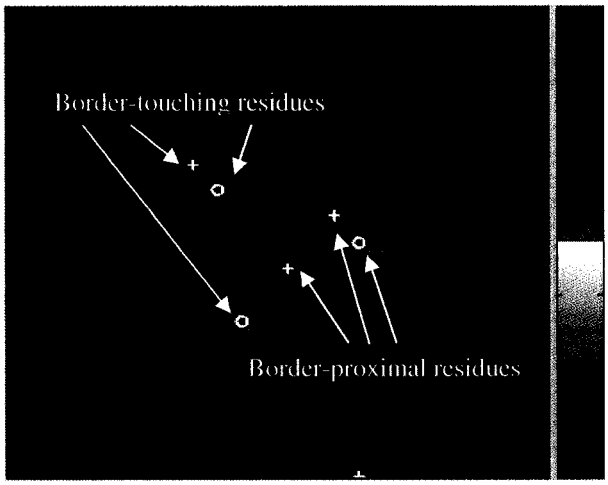


Figure 5.22 Examples of border-proximal and border-touching residues with an *in vivo* carotid flow.

The third step is to track the fringeline of each residue that is associated with a phase shift of $\pi/2$ or $-\pi/2$. As we explained before, we apply the dipole detection algorithm for both potential correct phase shifts and choose one solution later. We use the phase annihilation process that is described by Chavez et al. in [75] to track the fringelines. Sequentially replacing the 2×2 square of a residue by the negative of the intended phase shift value results in shifting the residue to one of the eight neighboring 2×2 squares. The residue shifting process continues along the fringeline associated with the used phase shift until it stops. The annihilation process stops at an opposite sign residue, unless it is distracted by the noise or by other nearby residues along the fringeline path. We save the destination pixel of the annihilation process for each residue; we call this pixel the primary fringeline destination.

The fourth step is to track the fringeline of each residue that is associated with a phase shift of π, similar to the previous step. We also save the destination pixels for all residues, which we call the secondary fringeline destination. This secondary phase shift will be used for the border-proximal residues, which otherwise will fail to connect to a reasonable opposite sign residue with the primary fringeline destination.

The fifth step is to detect valid dipoles, or residue pairs, for the inner residues that are away from the vessel border. The dipole detection process checks each residue against each opposite sign residue; a dipole is found when the following three conditions are true:

- The Euclidean distance between the two opposite sign residues is less than a threshold. This limits the number of residue combinations to check for unreasonable cutline lengths.
- At least one of the two opposite sign residues is neither a border-proximal residue nor a border-touching residue.
- At least one of the following is true:
    - The primary fringeline destination of either residue matches the second residue. This is the normal matching condition where the two residues track each other, or at least one of them tracks the other.
    - The primary fringeline destination of both residues match and the Euclidean distance between each residue and its destination is smaller than a threshold. This is necessary to cover cases of noise or influence of other nearby residues, which distract the fringeline tracking process. In this case the two residues may not reach each other but they both reach the same destination. The distance threshold here prevents the pairing of two distant residues.
    - One of the two residues is a border-touching residue and it matches the primary fringeline destination of the other residue. This is necessary for the inner residues that have their logical opposite sign residue lying on the vessel border.

This step detects dipoles of all inner residues while accounting for any distortions due to noise or interfering residues. Figure 5.23 shows several different examples for the inner residues dipole detection with different optimal phase shifts; the dipole residue pairs are connected with straight lines. It is evident from the figure that the resulting straight lines separate regions of discontinuous phase. More specifically, they separate unwrapped phase regions from severely-wrapped phase regions in Figure 5.23a, b, and d, while they separate unwrapped phase regions from moderately-wrapped phase regions in Figure 5.23c. However the straight line approximation of the true cutlines is not accurate in all cases (as is highlighted by the arrows), since the true cutlines are usually curved lines that result from the arbitrary adjacency

of two discontinuous phase regions. This will result in some inaccuracy in the final unwrapping results for those few pixels that are misclassified by the rough placement of the cutlines. However seeking accurate delineation of the true curved cutlines will impact the reliability of the algorithm due to the complexity of the ultrasound wrapping problem.

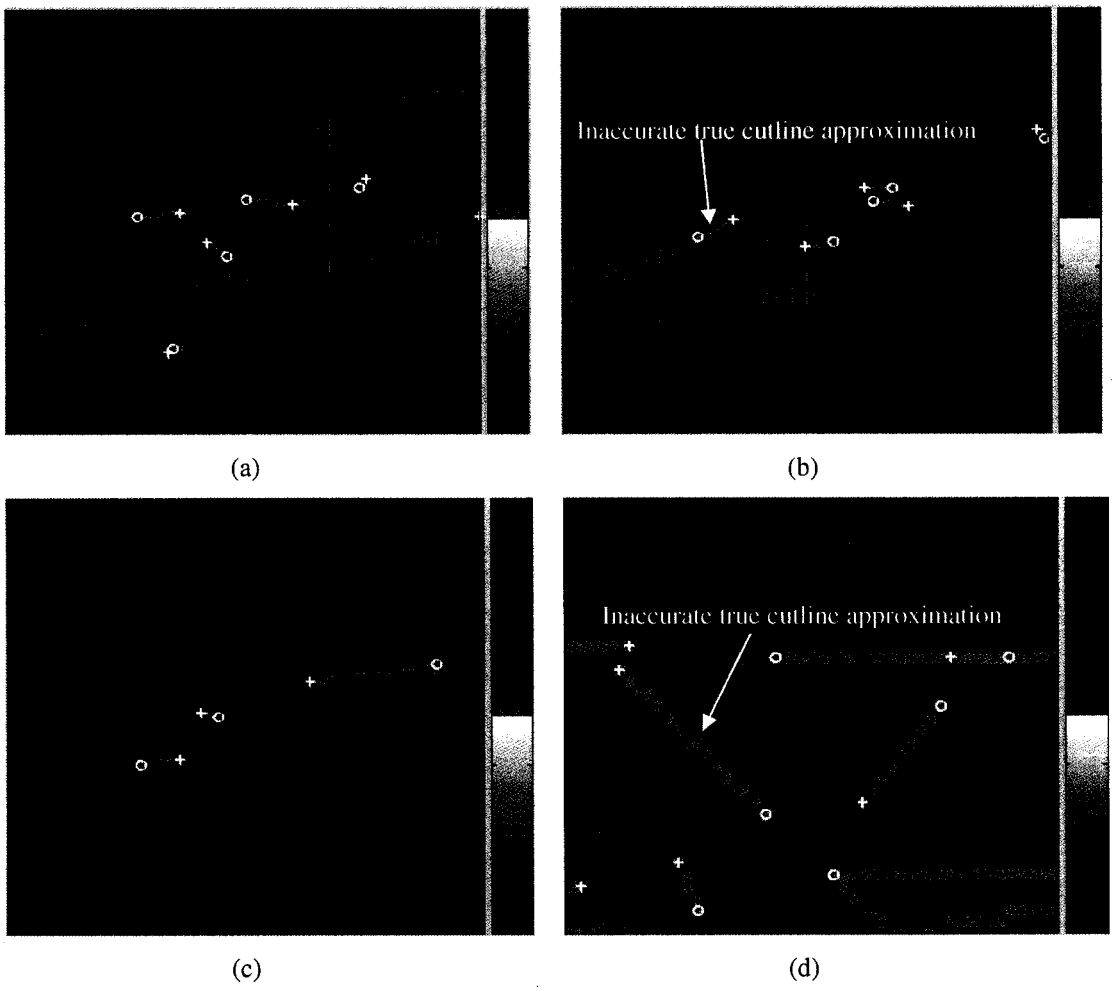


Figure 5.23 Dipole detection step. (a) Detected dipoles of a $\pi/2$ phase shift for a simulated carotid flow. (b) Detected dipoles of a $-\pi/2$ phase shift for a simulated femoral flow. (c) Detected dipoles of a $\pi/2$ phase shift for a simulated femoral flow with non-optimal Doppler steering. (d) Detected dipoles of a $\pi/2$ phase shift for an *in vivo* carotid flow.

The sixth step is to find dipoles for all border-proximal residues, not the border-touching ones. Neither $\pi/2$ nor $-\pi/2$ phase shift fringelines works for the border residues

due to the distortion caused by the low velocity pixels near the vessel wall. Figure 5.24 shows two examples of border-proximal residues. If there is a way to connect the two residues in each case with a cutline that would separate the wrapped region (red pixels) from the vessel boundary low phase region (dark blue) then this would prevent any phase unwrapping errors. These lines can be detected using a π phase shift fringeline tracking, as shown in Figure 5.24b and d. This is the reason we detected the π phase shift fringeline for all residues in a previous step and saved their destinations, which we called secondary fringeline destinations.

All border-proximal residues are checked against all opposite sign residues. If the distance between the two residues is smaller than a threshold and the secondary fringeline destination of either residue matches the other residue then a dipole of the two residues is found.

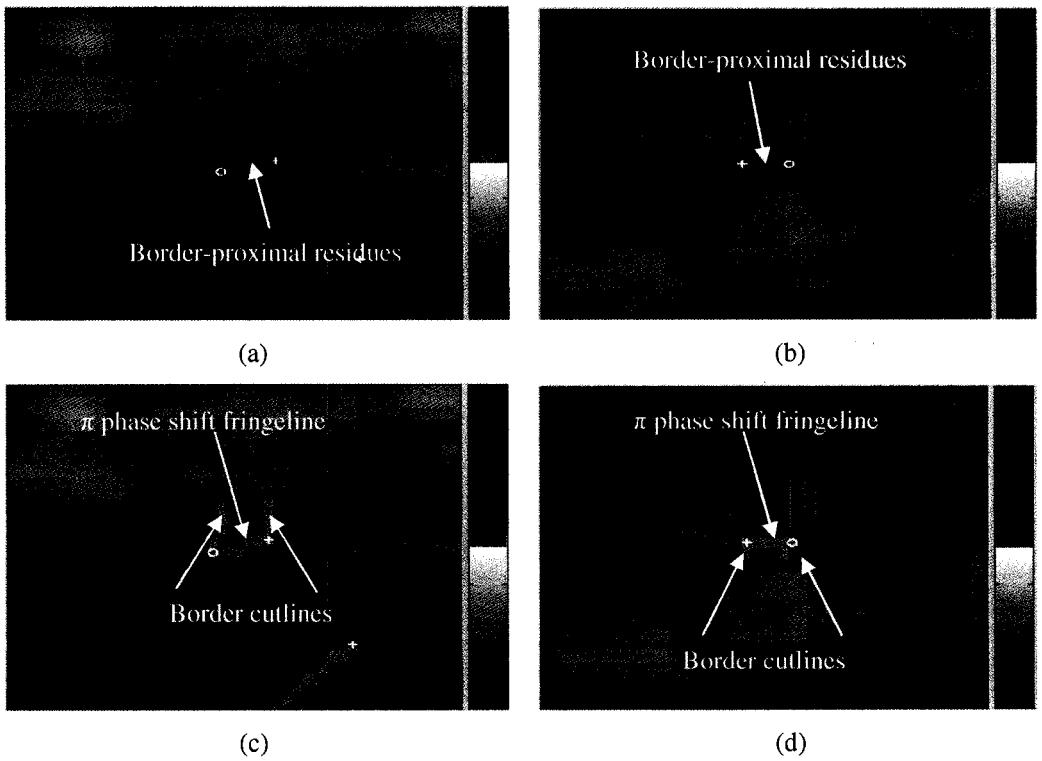


Figure 5.24 Examples of border-proximal residues, π phase shift fringelines, and border cutlines.

The seventh step is to find dipoles of the remaining unbalanced residues, which did not find matching residues using either the primary or the secondary fringeline destination in the previous steps. An unbalanced residue is paired with either the nearest opposite sign residue or the secondary fringeline destination, whichever is nearest. This step guarantees the connection of any orphan residues to a logical destination to prevent any unwrapping errors.

The last step is to connect the border-proximal residues to the vessel border, using what we call border cutlines. This step may seem redundant to step six where we connected the border-proximal residues to their secondary fringeline destinations. However we found it is important to connect these residues also to the vessel border to block the phase gradient integration during the unwrapping step from going through the vessel border. This blocking promotes the propagation of the phase unwrapping from the vessel center, which is more reliable, to the border areas, which is less reliable and error-prone. Figure 5.24b and d show examples of borer-proximal residues and their connections using border cutlines to the vessel border.

It is worth noting that the matching operation used in several steps within the dipole detection process is implemented as an intersection between the two 2×2 squares defined by the two involved residues, which lie in the top-left corner of these squares. This is necessary, because the annihilation process stops within a 2×2 square of an opposite sign residue, but not necessarily at the residue itself. The dipole-detection algorithm guarantees the pairing of all residues within the vessel lumen in a logical way that is consistent with the true cutlines between the discontinuous phase regions.

5.4.5. Cutline Construction and Phase Unwrapping

After the dipole detection process, the next step is to construct cutlines and use them to unwrap the phase while avoiding crossing these cutlines. The cutline construction process involves connecting the pair of residues in each dipole with a straight line. We use an

algorithm that creates an 8-connected discrete line between the two residue pixels. Figure 5.23 and Figure 5.24 show examples of the straight line approximations of the true cutlines.

The phase unwrapping step is a modified version of the classical flood-fill phase gradient integration process that is described in detail in [60]. The modification is related to the initialization step of the unwrapping process, where the classical method uses an arbitrary pixel to start the unwrapping, or it uses a high quality pixel if a phase quality map is provided. In our case, it is important to start with a pixel that is initially unwrapped. Otherwise the final solution will be biased by a phase shift of $n\pi$ in all pixels, depending on how much wrapping is inherent in the initial pixel, where $n > 1$ and represents the degree of wrapping of the initial pixel.

The initial pixel for the unwrapping in our modified algorithm has to fulfill two conditions. The first condition is that the pixel has to match the flow direction detected in the first step of the algorithm. The matching flow direction excludes the possibility that this pixel is moderately wrapped since the moderately wrapped pixels will have reverse flow direction. However the matching does not exclude the possibility that the pixel is severely wrapped, as the severely wrapped pixels share the same phase range (0, π] with the unwrapped pixels. The only differentiating property between the severely wrapped and the unwrapped regions is the clustering of the severely wrapped pixels below $\pi/2$ and the unwrapped pixels above $\pi/2$. Thus the second condition that increases the possibility for the initial pixel to be initially unwrapped is to test the average phase value of the surrounding 3×3 square of each pixel against the π value. If the average value is greater than the 90th percentile of π then the center pixel is most probably within an unwrapped phase region.

The phase image is scanned horizontally and the first pixel within the vessel lumen that fulfills both conditions is used as an initial pixel for the unwrapping process. The initial pixel is used to unwrap all pixels within the same connected region. It is possible that the constructed cutlines could create disconnected regions within the image by isolating certain regions within intersecting cutlines and the vessel border. In this case, each region is

unwrapped separately and requires an initialization pixel to seed the unwrapping algorithm. The described conditions for finding the initialization pixel are checked for each disconnected region in the image. If there is no pixel that fulfills both conditions for a certain region then the second condition is waived and the first pixel that matches the flow direction is chosen. This could happen for an isolated region with a few unwrapped pixels or no unwrapped pixels at all.

Figure 5.25 a, c, and e show some examples of wrapped phase images. Figure 5.25 b, d, f show the unwrapping results. Both the moderately-wrapped and severely-wrapped regions are successfully unwrapped in these cases.

5.4.6. Unwrapped Image Selection

The last step of our phase unwrapping algorithm is to select one of the two unwrapping solutions produced by the two different phase shifts $\pi/2$ and $-\pi/2$. In this step we apply three different criteria that attempt to validate the two solutions.

The first criterion is related directly to our initial assumption that we are solving the phase wrapping problem for the $(-3\pi,3\pi]$ range, which is clinically acceptable for the peripheral vascular applications in which we are interested. For this criterion, we count the number of unwrapped pixels that have phase value beyond the valid range $(-3\pi,3\pi]$. Then we use a threshold to validate the solution if the number of invalid unwrapped pixels exceeds this threshold. The threshold is important to allow some tolerance for the cases where the solution is valid except for very few pixels where the phase values are outside the valid range. This could happen because we approximate the true cutlines with straight lines connecting the detected dipoles. In this case, some of the unwrapped pixels may be classified as severely wrapped pixels and their unwrapped value will be over-estimated.

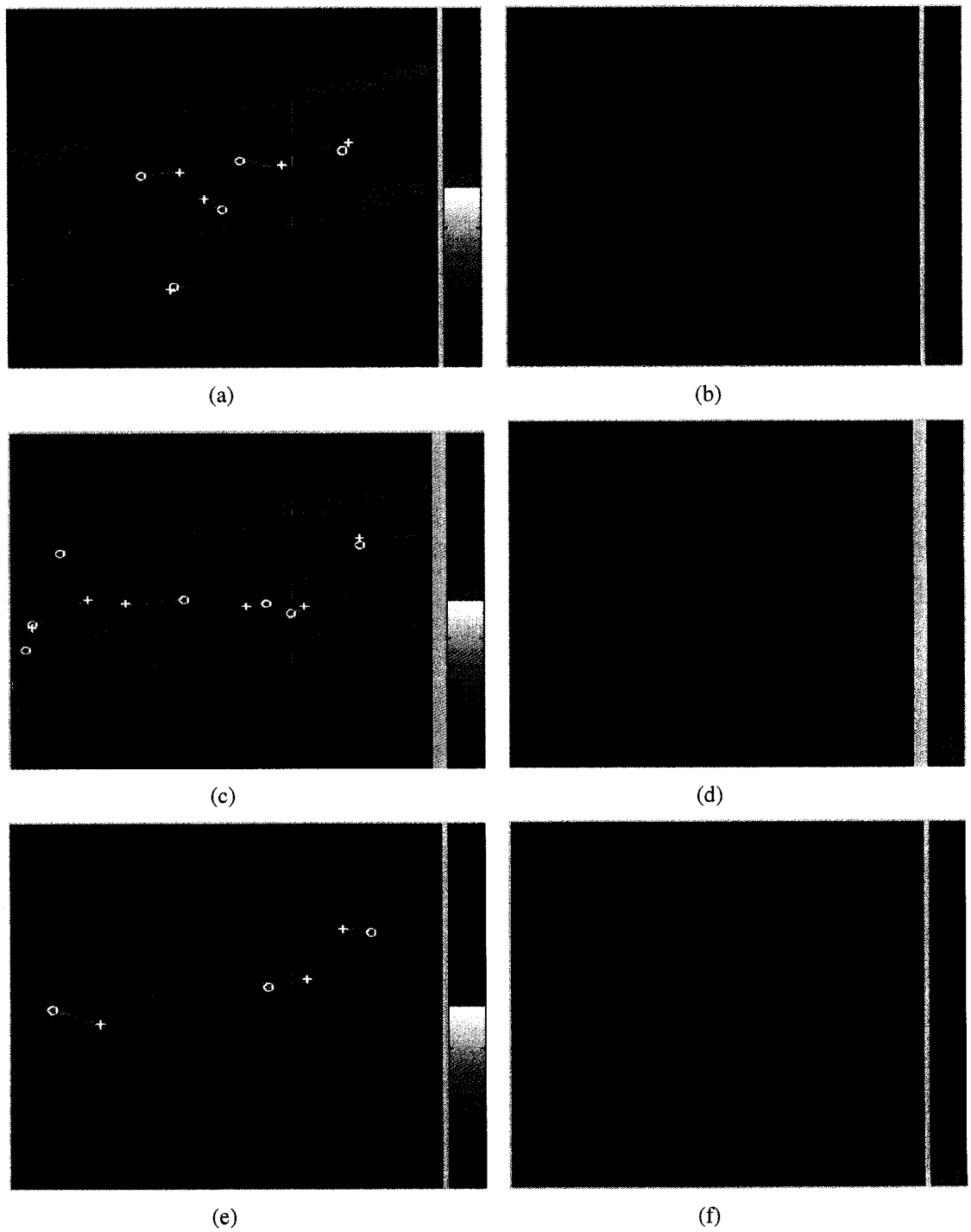


Figure 5.25 Phase unwrapping results. (a) A wrapped simulated carotid flow. (b) The unwrapped image. (c) A wrapped simulated femoral flow. (d) The unwrapped image. (e) A wrapped simulated femoral flow with non-optimal Doppler steering. (f) The unwrapped image.

If one phase shift solution is valid, but the other solution is not, then we accept the valid solution and reject the invalid one. If both solutions are invalid then we accept the solution with the smaller number of invalid pixels. In this case our algorithm failed to unwrap the phase data without introducing errors for both phase shifts and we choose the least erroneous solution. If both solutions are valid then we have to rely on a different criterion.

Figure 5.26 shows an example of a simulated femoral flow with the two sets of fringelines associated with the two phase shifts $\pi/2$ and $-\pi/2$. The two unwrapped phase images are shown in Figure 5.26e and f. In this case the $\pi/2$ phase shift image has a large region with phase values higher than 3π, which is the valid range for our problem. The other solution has all pixels within the valid range $(-3\pi,3\pi]$ and is selected as the unwrapping solution for the wrapped input image.

The second criterion to validate the two unwrapping solutions is the total length of the detected cutlines. If the total cutline length is not realistic, then it indicates the invalidity of that solution. However we require that the length criterion be prominent enough to be used, since too close total cutline length may not indicate the invalidity of one of the solutions. Specifically with ultrasound data where the residues are not due to noise, but due to adjacency of discontinuous phase regions, the true cutlines are not necessarily short.

Figure 5.27 shows an example of using the total cutline length criterion for a simulated carotid artery flow. In this case the two unwrapped solutions shown in Figure 5.27 e and f both have valid phase range pixels. Hence in this case the invalid phase range criterion cannot favor one solution over the other. However the total Euclidean length of all cutlines is 253 pixels for the $-\pi/2$ solution and is 79 for the $\pi/2$ solution. In this case one solution has a total cutline length which is greater than three times the other solution, and we consider this criterion prominent enough to favor the $\pi/2$ over the other one.

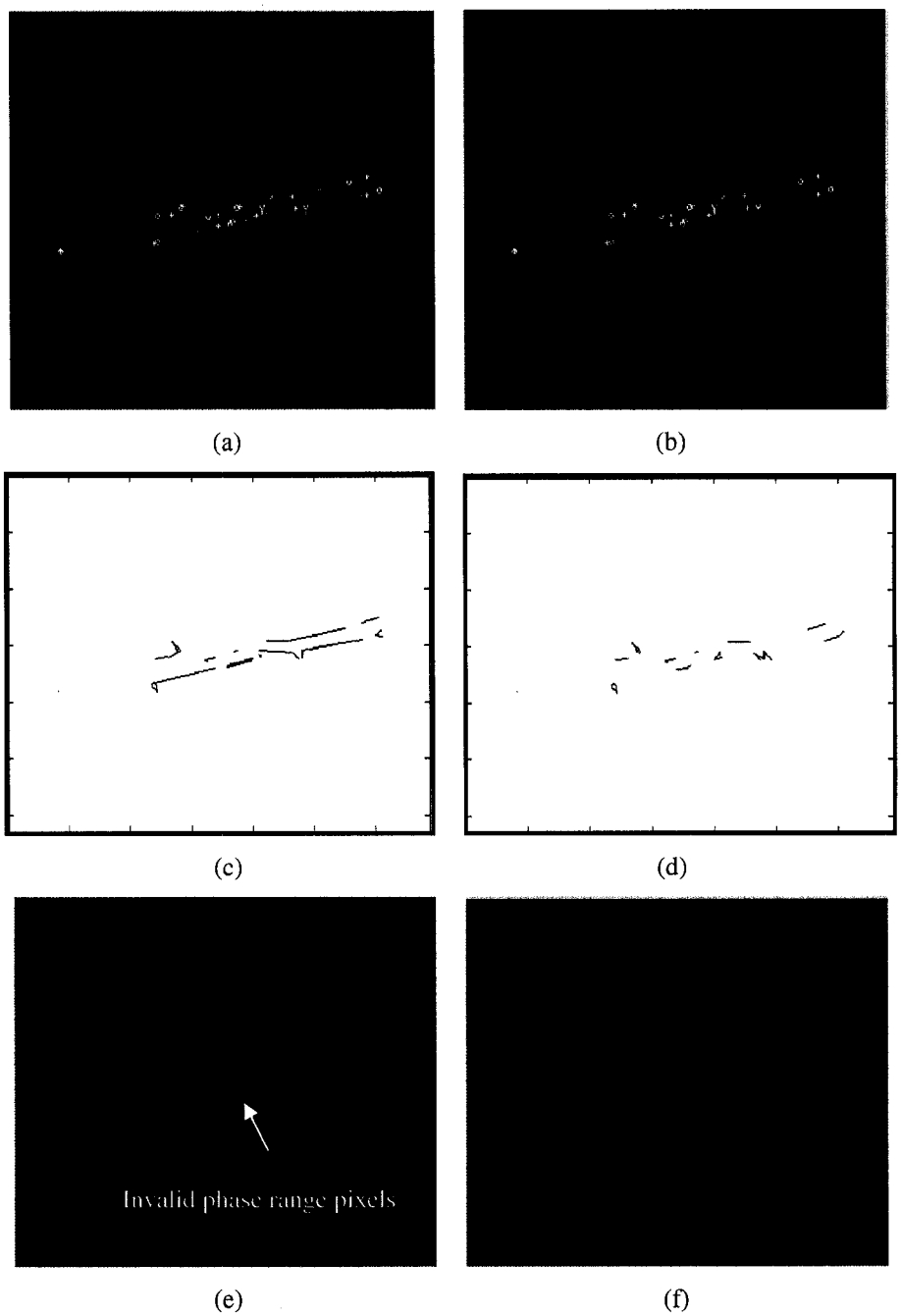


Figure 5.26 Invalid unwrapped phase range selection criterion. (a) A simulated femoral flow with $\pi/2$ phase shift fringelines. (b) The same input image with $-\pi/2$ phase shift fringelines. (c) Detected cutlines for $\pi/2$ phase shift. (d) Detected cutlines for $-\pi/2$ phase shift. (e) Unwrapped phase image with $\pi/2$ phase shift. (e) Unwrapped phase image with $-\pi/2$ phase shift.

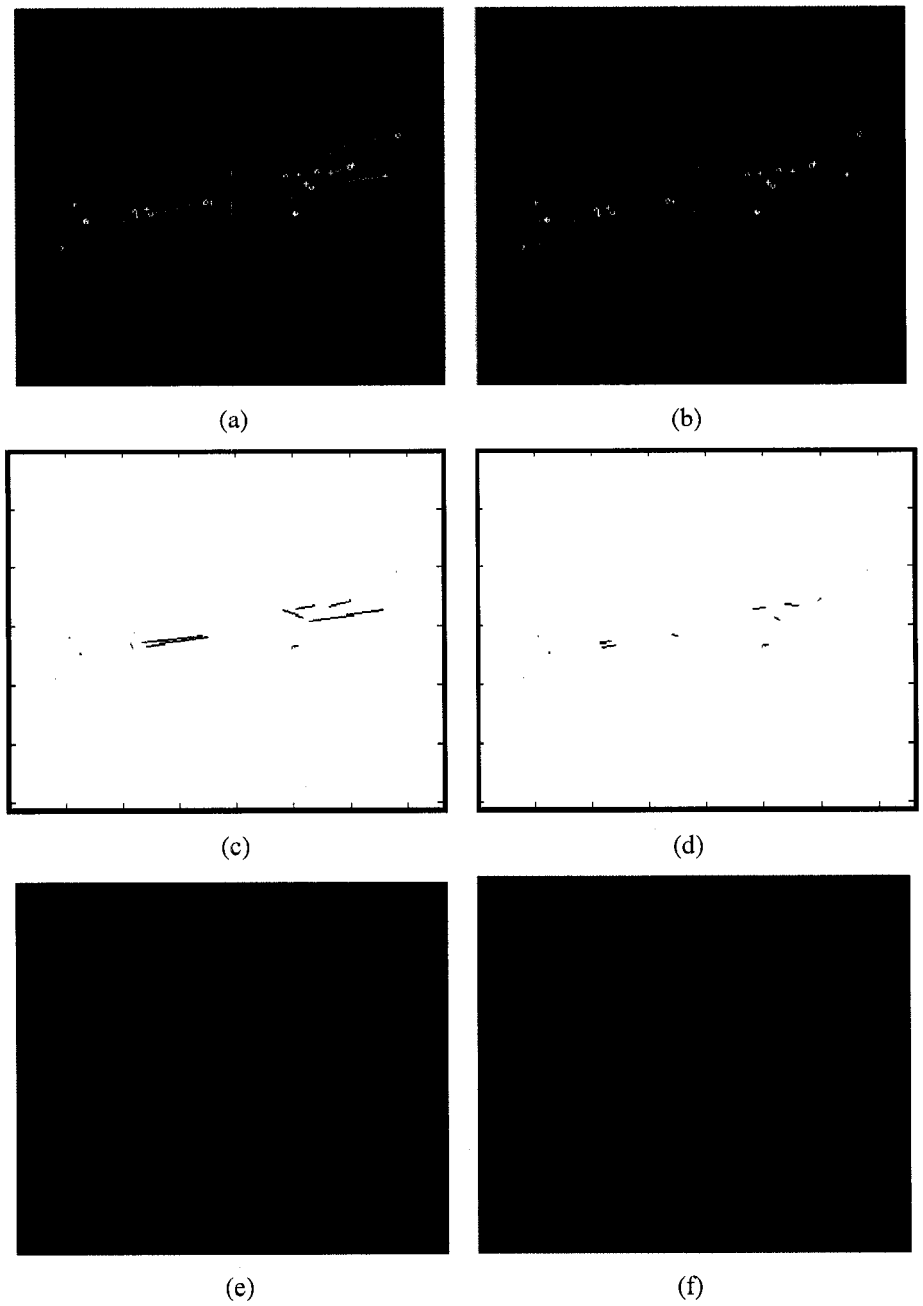


Figure 5.27 Total cutline length selection criterion. (a) A simulated carotid flow with $-\pi/2$ phase shift fringelines. (b) The same input image with $\pi/2$ phase shift fringelines. (c) Detected cutlines for the $-\pi/2$ phase shift with length=253 pixels. (d) Detected cutlines for the $\pi/2$ phase shift with length=79. (e) Unwrapped phase image for the $-\pi/2$ phase shift. (e) Unwrapped phase image with $\pi/2$ phase shift.

If the total length of the cutlines is comparable between the two solutions then we rely on the third and last criterion, which is a discontinuity measure, to validate one of the two solutions. The idea here is that the valid unwrapping solution should produce a phase image with a homogeneous phase map. This is a similar idea to the ones used in other phase unwrapping techniques, such as Flynn's minimum discontinuity technique [69], and all minimum discontinuity solutions [70], [72]. The sought solution is always the one with a minimum phase discontinuity.

In our case, we developed a metric for the phase discontinuity as the number of boundary pixels between the positive and negative phase regions. The idea here is that the final unwrapped image should reveal a homogeneous phase map with one correct flow direction. This is true even for the triphasic waveforms, which change flow direction across different frames but not within the same frame. Even with the non-optimal Doppler steering angle and the coexistence of different flow direction regions, the phase discontinuity metric captures the additional discontinuity on top of the original one already inherent in the frame. The solution with the smaller discontinuity measure is chosen as the valid solution.

Figure 5.28 shows an example of a solution selection based on the phase discontinuity metric. The two unwrapped solutions shown in Figure 5.28e and f both have valid phase range pixels, and the two cutline maps shown in Figure 5.28c and d have comparable cutline lengths of 336 and 285 for the $-\pi/2$ and $\pi/2$ solutions respectively. However the discontinuity measure is 194 for the $-\pi/2$ solution due to the misclassified pixels (marked by the arrow), while it is 50 for the $\pi/2$ solution. Hence the $\pi/2$ solution in Figure 5.28f is selected as the final unwrapping solution.

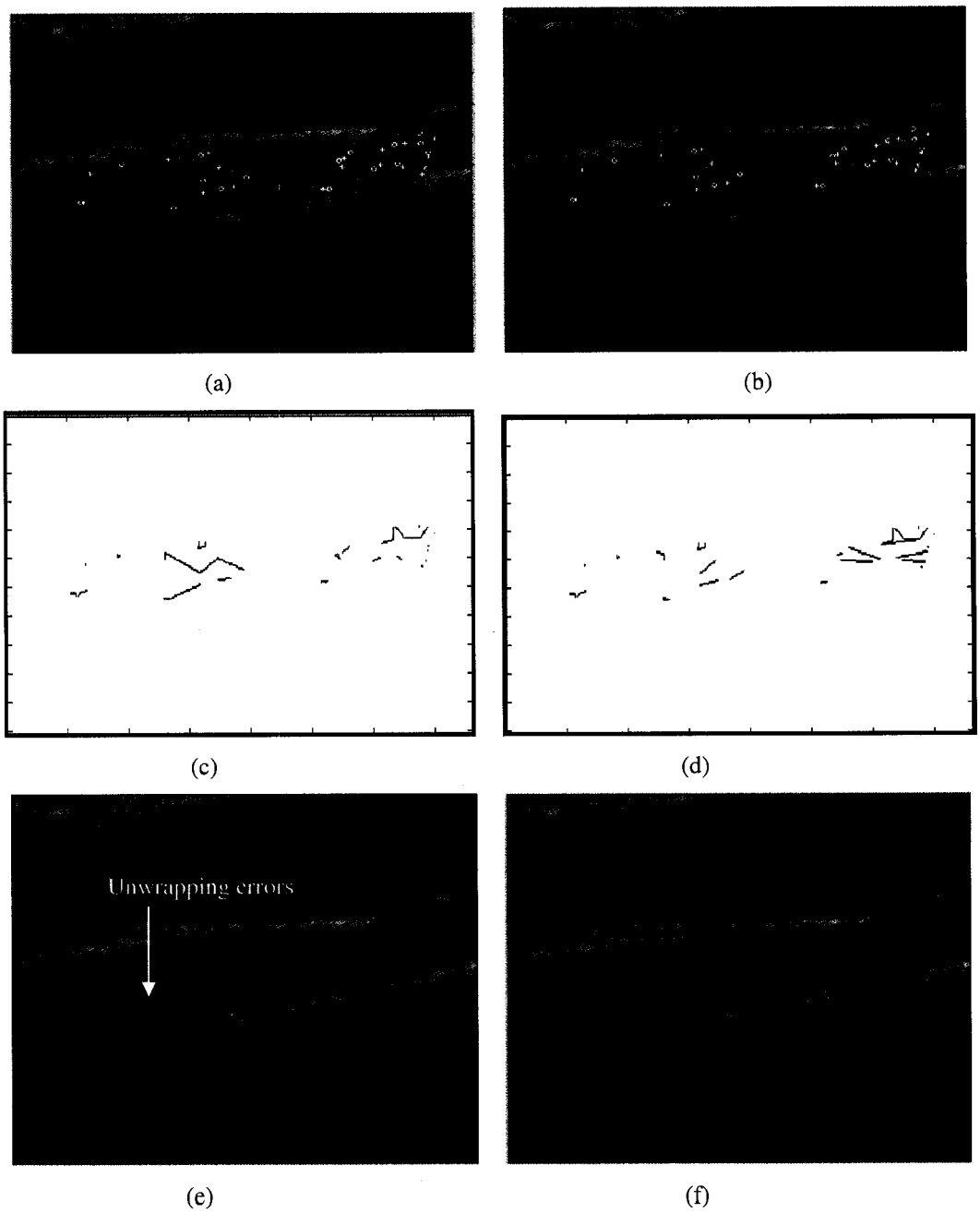


Figure 5.28 Phase discontinuity selection criterion. (a) An *in vivo* carotid flow with $-\pi/2$ phase shift fringelines. (a) The same carotid flow with $\pi/2$ phase shift fringelines. (c) Detected cutlines for the $-\pi/2$ phase shift with length=336 pixels. (d) Detected cutlines for the $\pi/2$ phase shift with length=285. (e) Unwrapped phase image for the $-\pi/2$ phase shift with phase discontinuity measure=194. (f) Unwrapped phase image with $\pi/2$ phase shift with phase discontinuity measure=50.

5.5. Unwrapping Results

In this section we will present the results of our phase unwrapping technique with all the datasets we acquired. We will start with the *in vitro* flow phantom data then we will show the results with the *in vivo* data.

5.5.1. *In Vitro* Flow Phantom Data Results

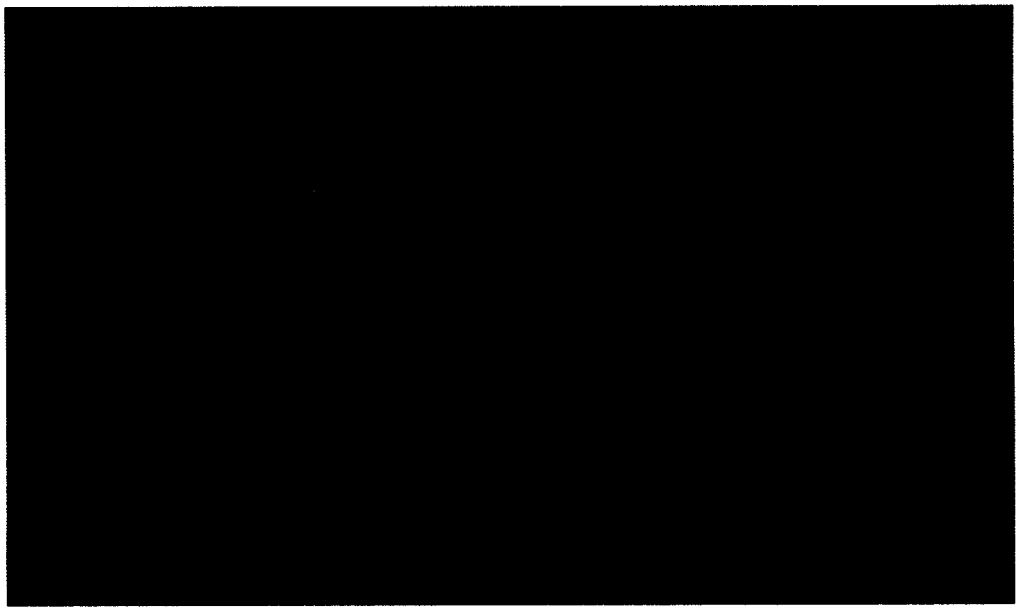
We tried our algorithm on all flow phantom datasets that we acquired previously according to a specific protocol that was described in detail in section 4.4.1. The flow phantom simulates different flow patterns that represent major blood vessels, such as the carotid and femoral arteries and the steady venous flow. Each dataset consists of 30 frames that encompass at least one heart cycle.

Due to the excessive number of frames we have, we will not be able to show the detailed results of all frames. Instead we will show the input wrapped phase images and the output unwrapped phase images as a mosaic image of all frames. We will then focus on a few frames of each dataset that represent severely-wrapped cases, which are the most challenging cases. We will not give the details of the originally unwrapped frames or those with moderate wrapping where there are only border residues, since these cases are straight-forward for any phase-unwrapping algorithm. For each presented case we will show three images, the original wrapped image with the detected residues and cutlines superimposed on top of it, the residues and cutlines map that is shows the dipole connections more clearly, and finally the unwrapped phase image that represents the solution of that case. To be able to use the same color map to display both the original wrapped phase image and the unwrapped phase image, the unwrapped phase image is scaled down to map its phase values within the principal phase range $(-\pi, \pi]$ before its display using the same color map used for the original phase image. Hence, the unwrapped phase image will be shown darker than it should be due to the down scaling process.

We will start with the simulated carotid flow cases. Figure 5.29 shows the wrapped input and unwrapped output images of the first cineloop for a simulated carotid flow. Figure 5.30 shows a severely-wrapped case. The detected dipoles are correct as shown in Figure 5.30b. The areas of severe wrapping were unwrapped successfully as highlighted by the arrows in Figure 5.30c. Figure 5.31 shows another example with a larger region of severe wrapping. The detected dipoles and unwrapping result look correct where the severely-wrapped regions are correctly unwrapped and highlighted by the arrows.

Figure 5.32 shows another simulated carotid flow example with severe phase wrapping. Figure 5.33 shows an example of an unsuccessful case where a region with severe wrapping was underestimated and the final unwrapped values are incorrect as highlighted by the arrow in Figure 5.33c. This happened because of a missing cutline that was supposed to connect an inner positive residue with a boundary negative one. The fringeline tracking process was off by a few pixels from the correct boundary residue and the positive residue ended up connecting to the nearest opposite sign residue, which did not prevent unwrapping errors.

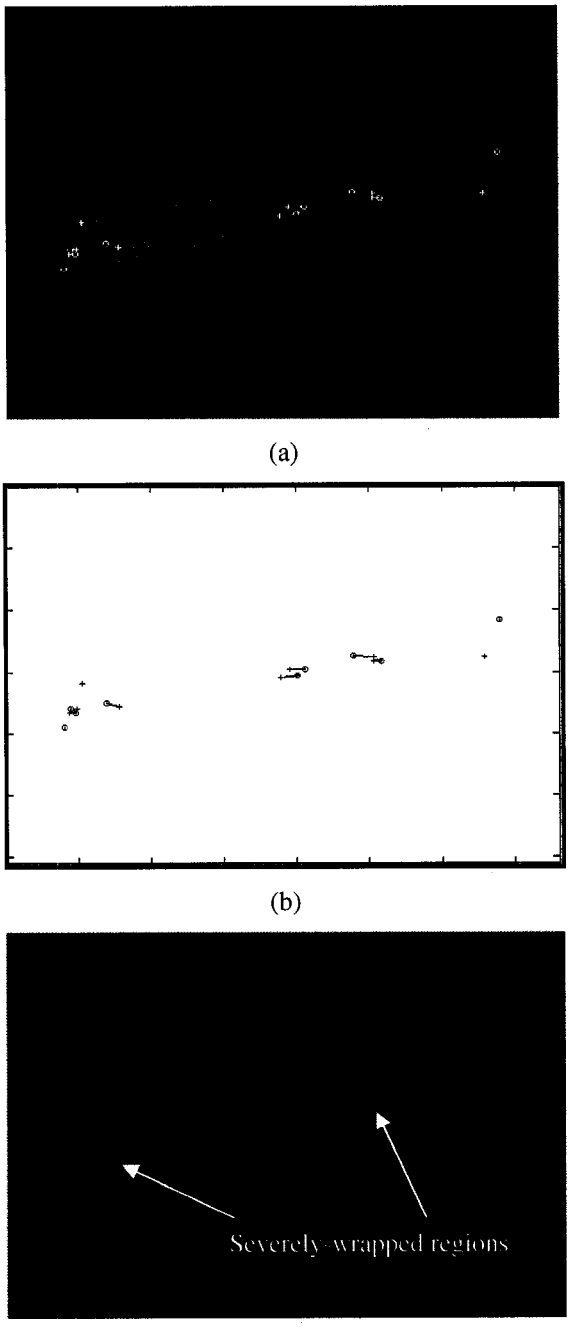
Figure 5.34 and Figure 5.35 show examples with more wrapping degrees where there are many residues in each case. The unwrapped phase images look very reasonable where some areas with severe phase wrapping are correctly unwrapped, as highlighted by the arrows. With these extreme wrapping cases, it is very difficult even for clinical experts to predict how the correct unwrapped image should look. At least the solution should not have wrapping errors in many pixels, and some regions of sever wrapping should be correctly unwrapped to preserve the true velocity information. Omitting some severe aliasing regions may slightly underestimate the true velocity, but it should not greatly affect the quantification of the color Doppler data or the following recognition tasks.



(a)

(b)

Figure 5.29 Simulated carotid flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images



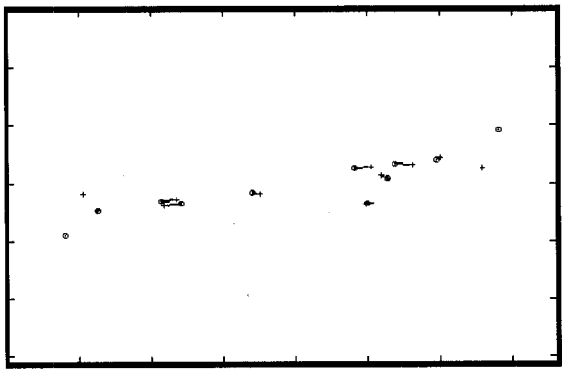
(a)

(b)

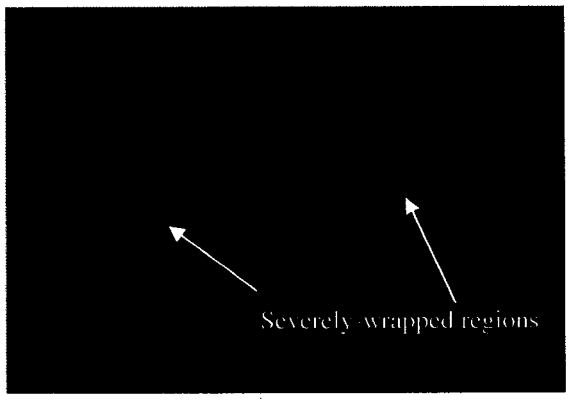
(c)

Figure 5.30 Simulated carotid unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

(a)

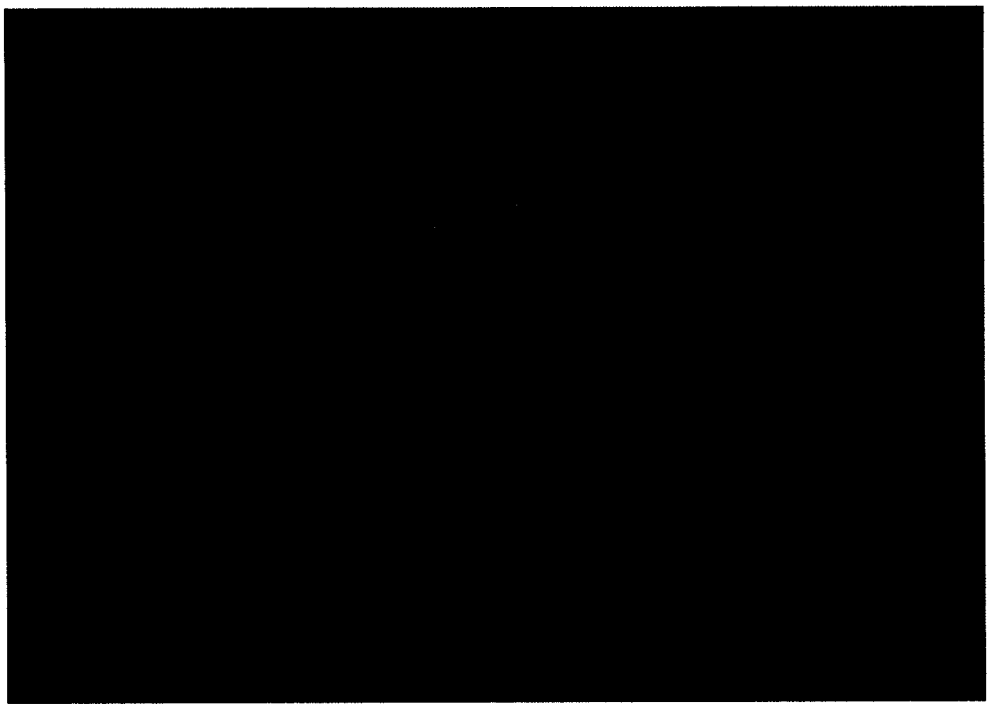


(b)



(c)

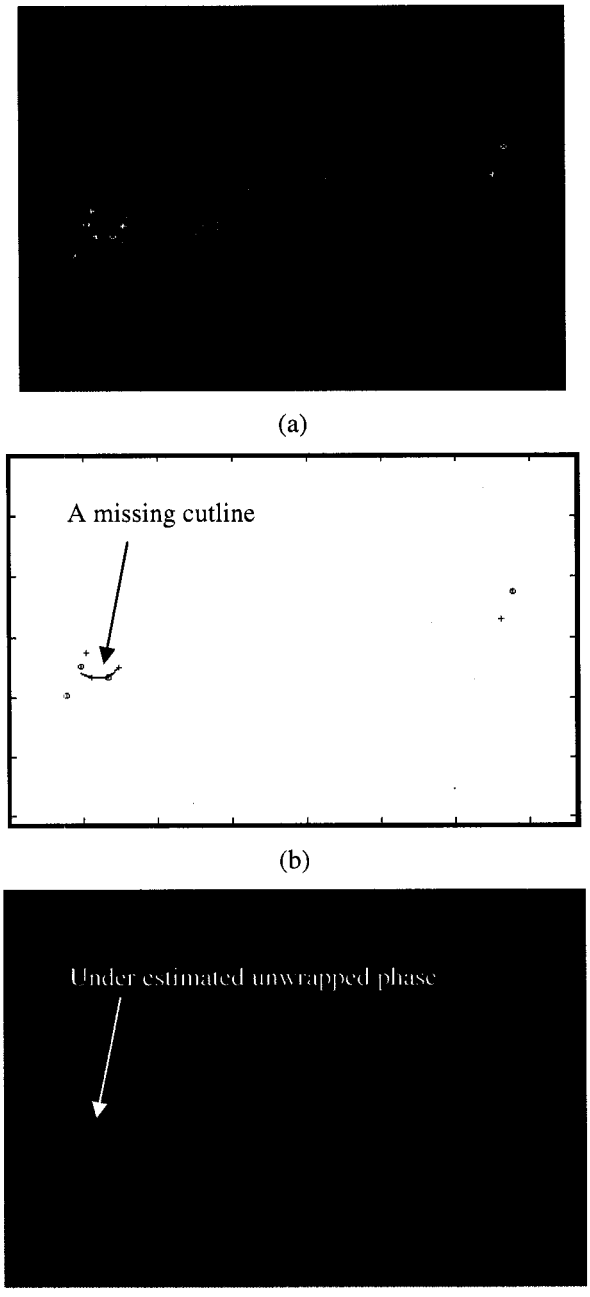
Figure 5.31 Simulated carotid unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.



(a)

(b)

Figure 5.32 Simulated carotid flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images



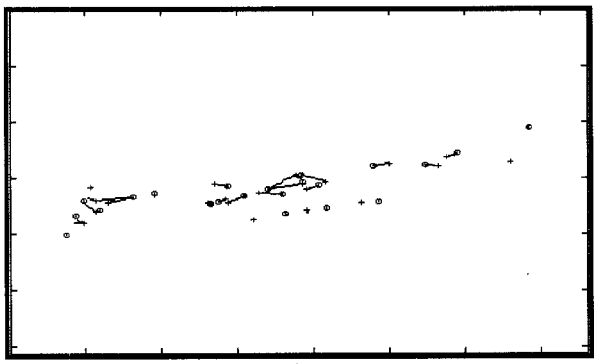
(a)

(b)

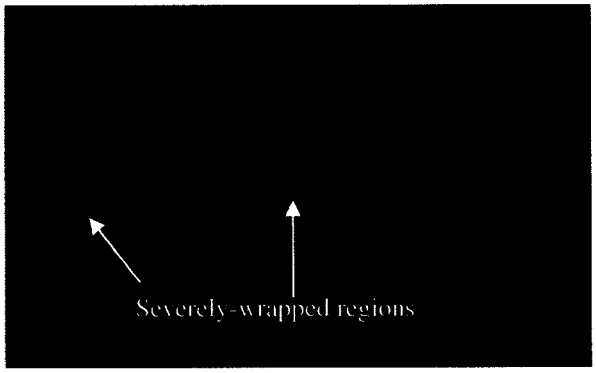
(c)

Figure 5.33 Simulated carotid unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

(a)



(b)



(c)

Figure 5.34 Simulated carotid unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

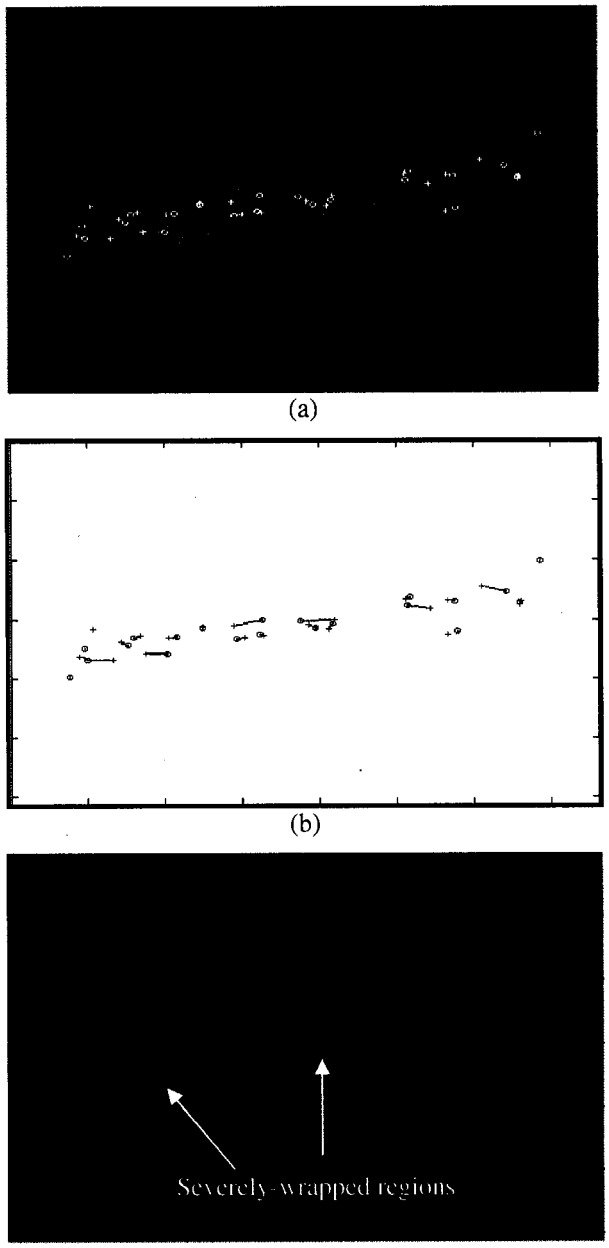


Figure 5.35 Simulated carotid unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

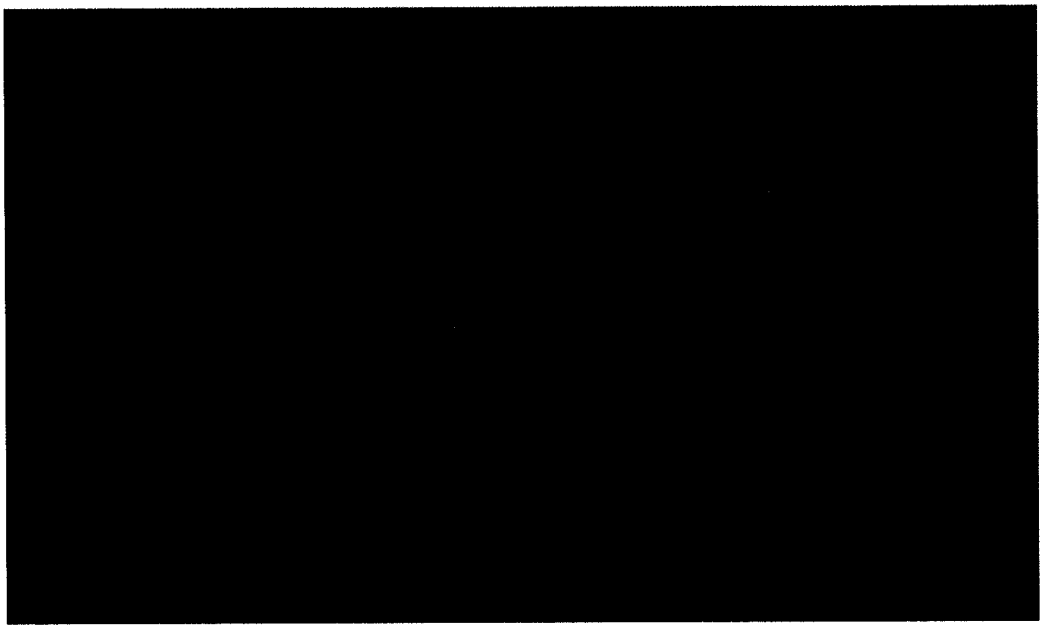
The new unwrapping algorithm was tried on simulated constant venous flow generated by a flow phantom. Figure 5.36 shows a cineloop of severely-wrapped venous flow. The unwrapped images are shown in Figure 5.36b and they all look correct. Figure 5.37 shows a case with severe wrapping regions, which were correctly unwrapped as highlighted by the arrows on the unwrapped image. Figure 5.38 shows another successful example of a severely-wrapped case.

The algorithm was also tried on simulated triphasic femoral carotid flow generated by a flow phantom. Figure 5.39 shows a cineloop of severely-wrapped simulated femoral flow. The unwrapped images are shown in Figure 5.39b, and they all look correct despite the reversal flow pattern of a triphasic waveform. Figure 5.40 shows a case with severe wrapping regions, which were correctly unwrapped as highlighted by the arrows on the unwrapped image. Figure 5.38 shows another successful example of a severely-wrapped case.

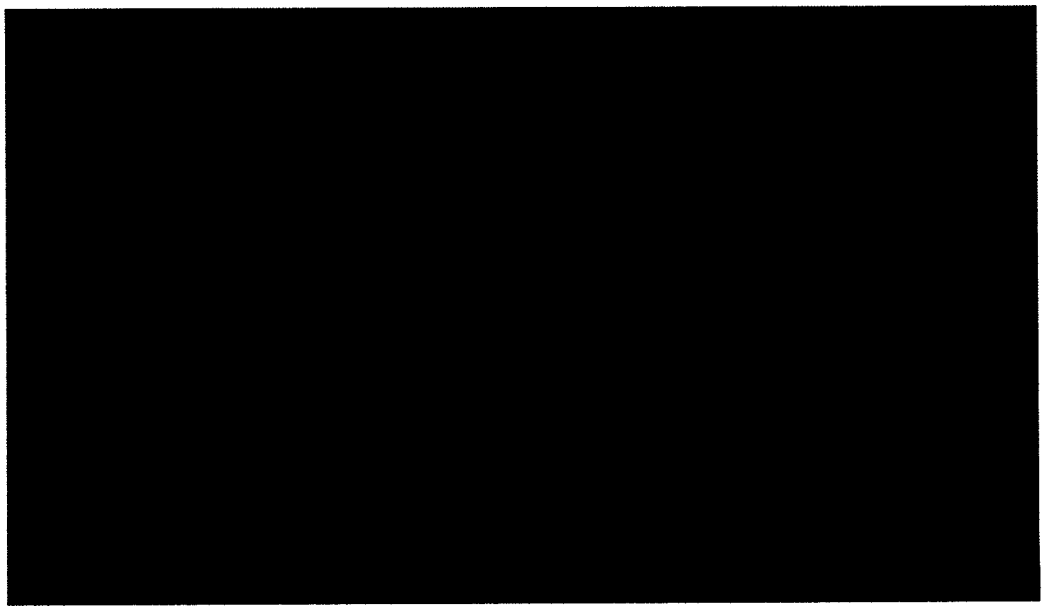
The last *in vitro* case on which the algorithm was tested on was the simulated femoral flow with a non-optimal Doppler steering angle. Although this case may not be clinically relevant, it represents the most challenging condition that would face any phase unwrapping algorithm for color Doppler ultrasound data. This is because the almost perpendicular Doppler steering angle causes flow direction ambiguity, which results in adjacent opposite flow regions, as shown in Figure 5.42a. The unwrapped cineloop is shown in Figure 5.42b; we notice the correctness of all the frames at least in terms of absence of major unwrapping errors.

Figure 5.43 shows a case with moderately-wrapped regions. In this case the optimal cutlines should separate the regions of moderate wrapping from the original unwrapped regions. The detected cutlines and the resultant unwrapped image are shown in Figure 5.43b and c respectively. Figure 5.44 shows another successfully unwrapped image but with different phase shift. Figure 5.45 shows a third example which was successfully unwrapped except for a small boundary region where the unwrapped phase was over-estimated, as highlighted by the arrow. This is due to the selection of the less optimal phase shift ($\pi/2$)

instead of the more optimal one $(-\pi/2)$. However the error is confined to a small region which is attached to the vessel boundary.

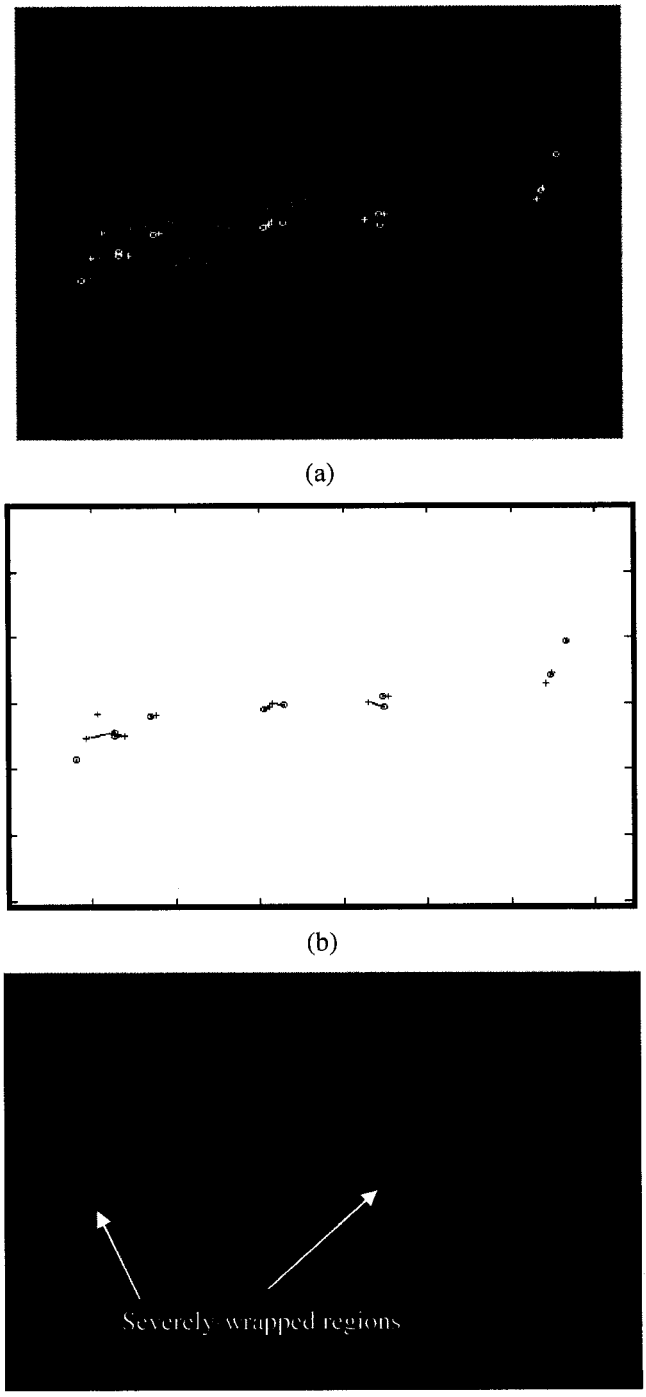


(a)



(b)

Figure 5.36 Simulated venous flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.



(a)

(b)

(c)

Figure 5.37 Simulated venous unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

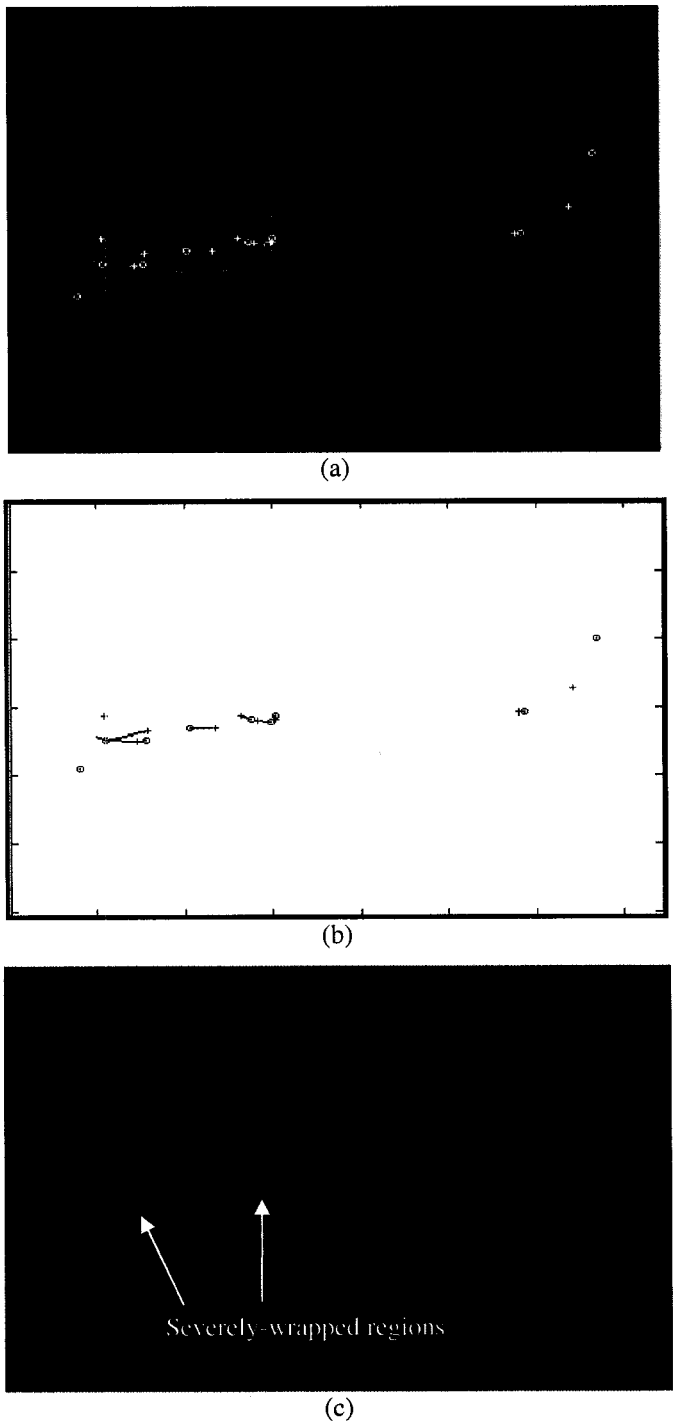
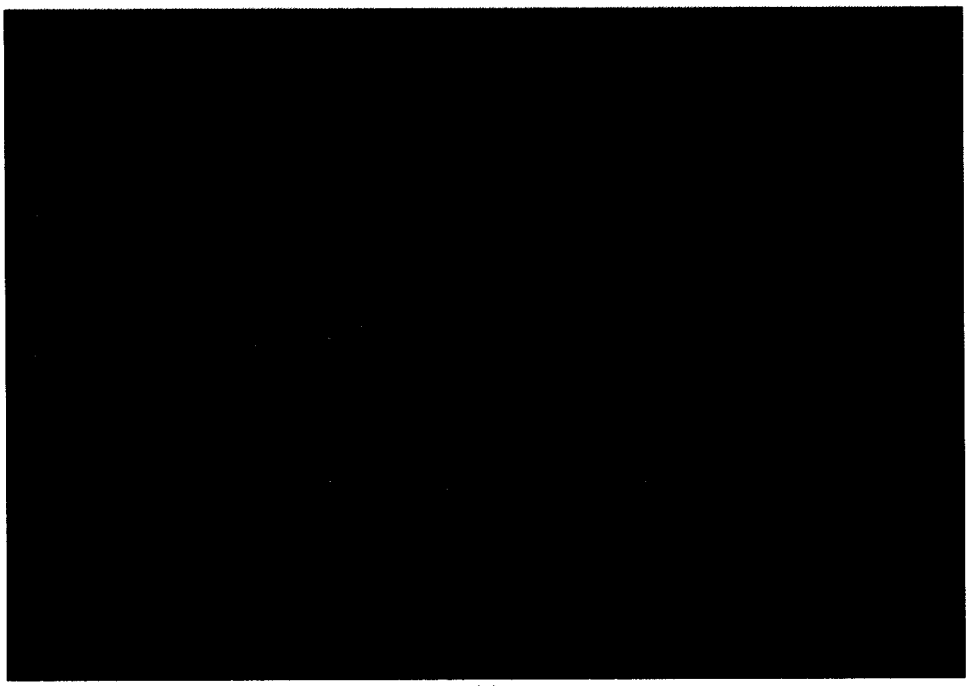
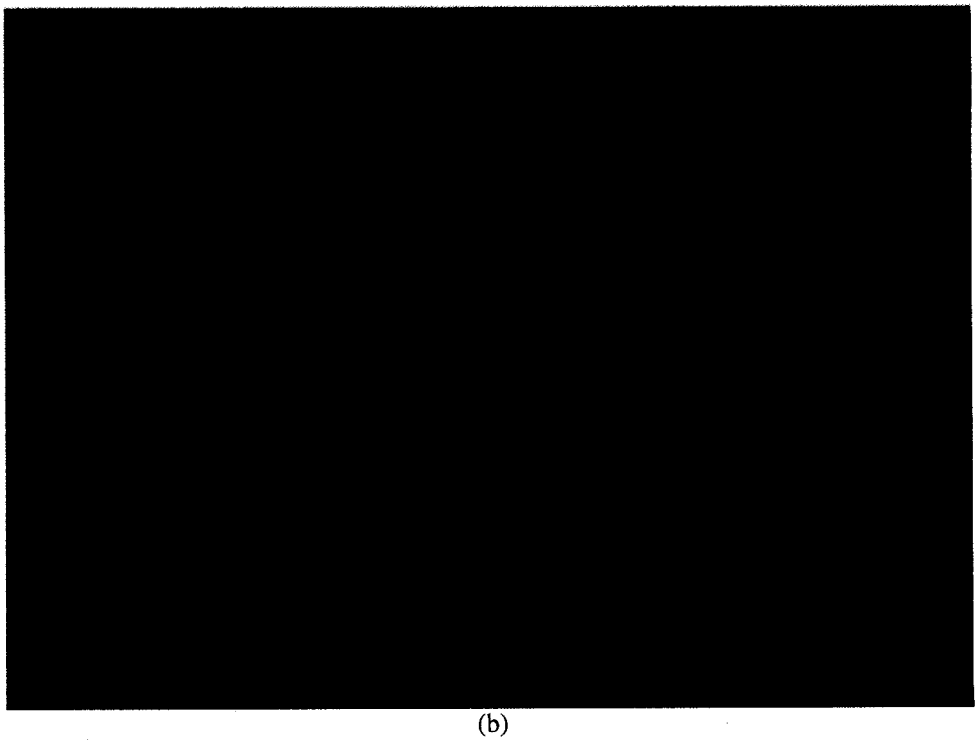


Figure 5.38 Simulated venous unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

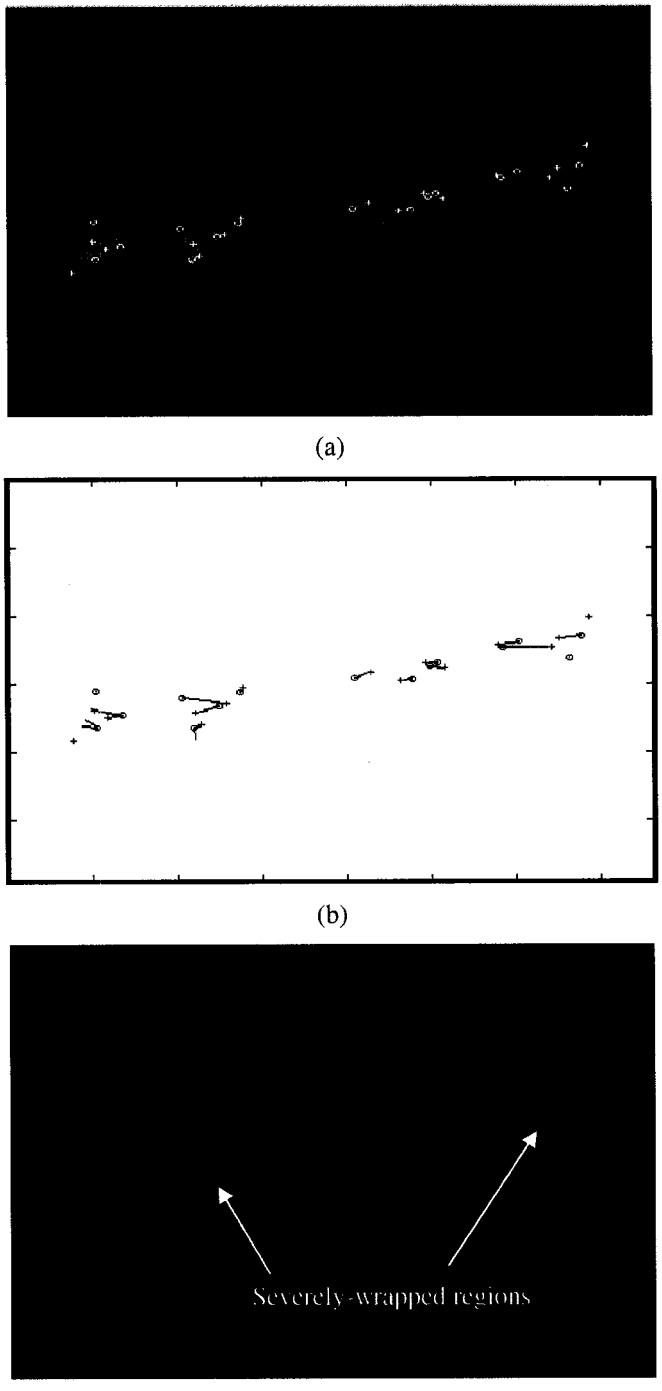


(a)



(b)

Figure 5.39 Simulated femoral flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.



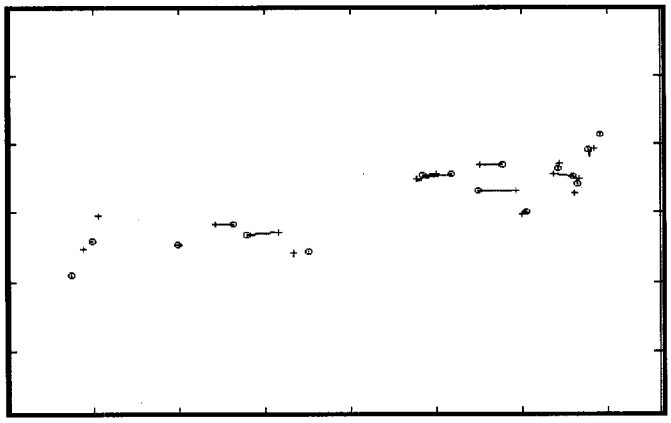
(a)

(b)

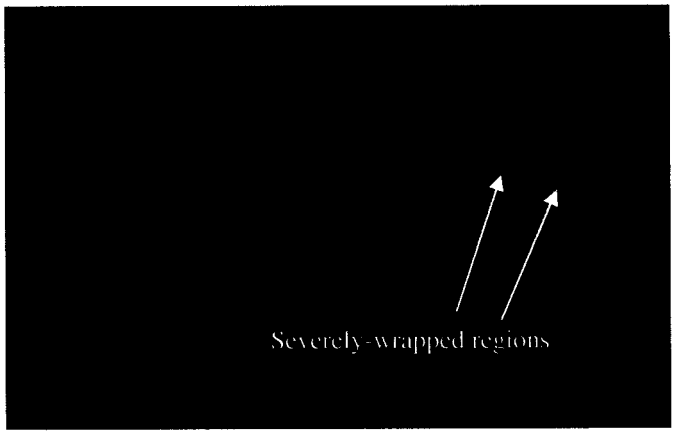
(c)

Figure 5.40 Simulated venous unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $-\pi/2$ phase shift. (c) Unwrapped phase image.

(a)

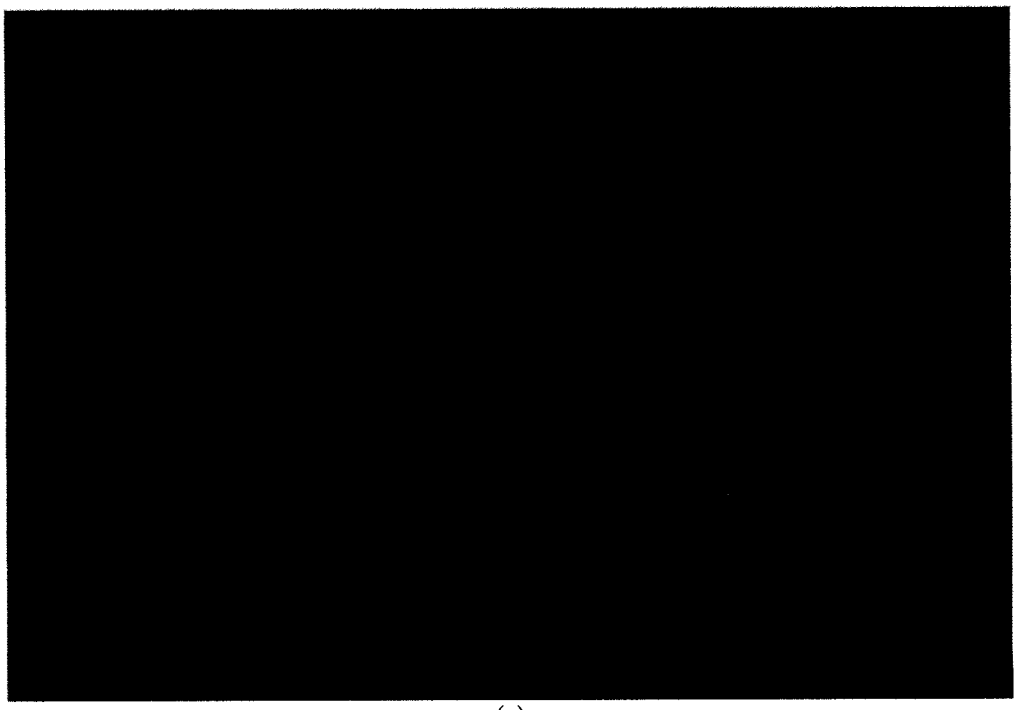


(b)

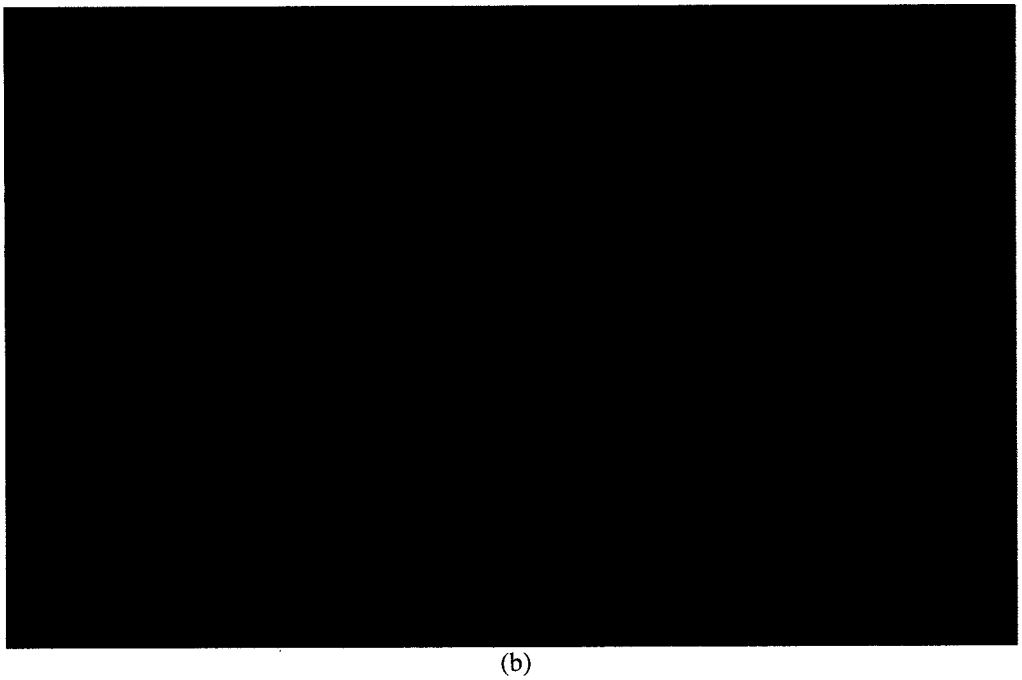


(c)

Figure 5.41 Simulated femoral unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

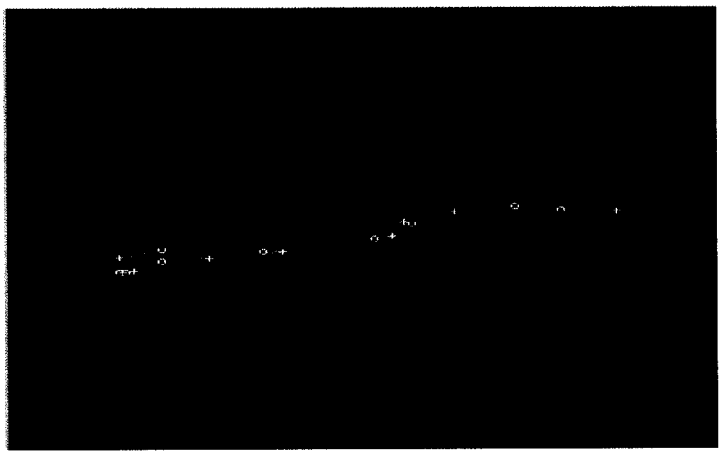


(a)

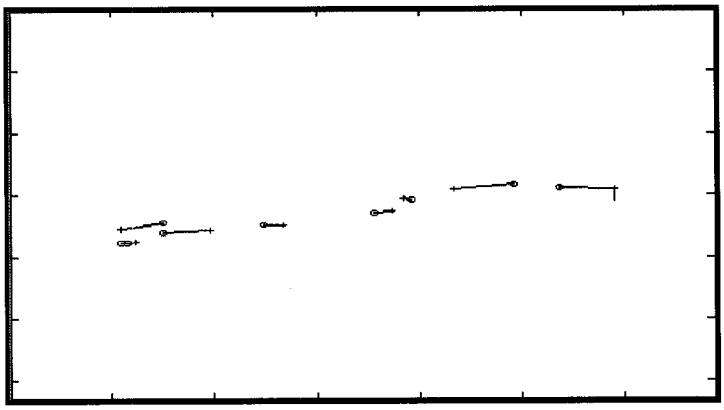


(b)

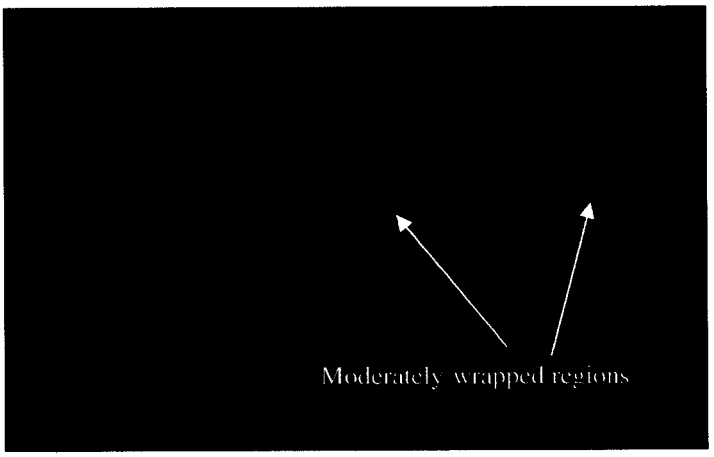
Figure 5.42 Simulated femoral flow with non-optimal Doppler steering angle cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.



(a)

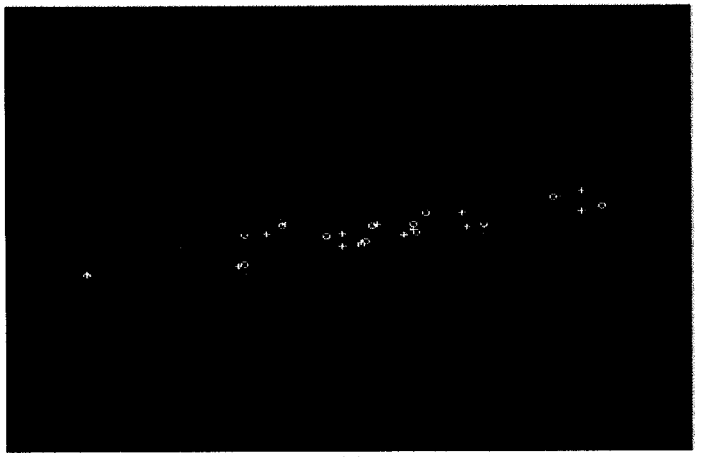


(b)

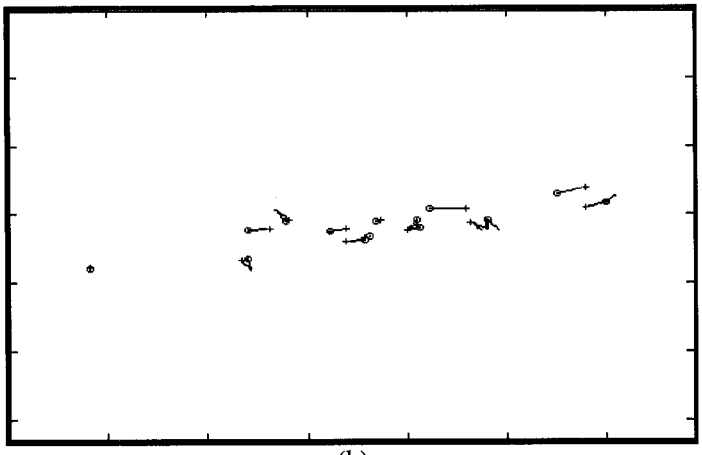


(c)

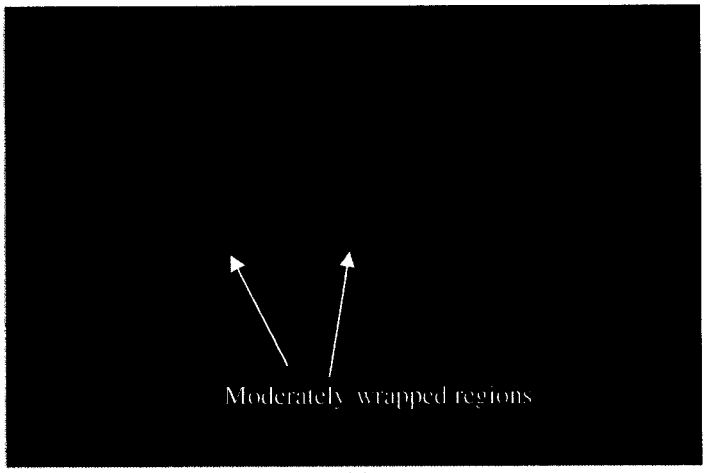
Figure 5.43 Simulated femoral with non-optimal steering unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi / 2$ phase shift. (c) Unwrapped phase image.



(a)



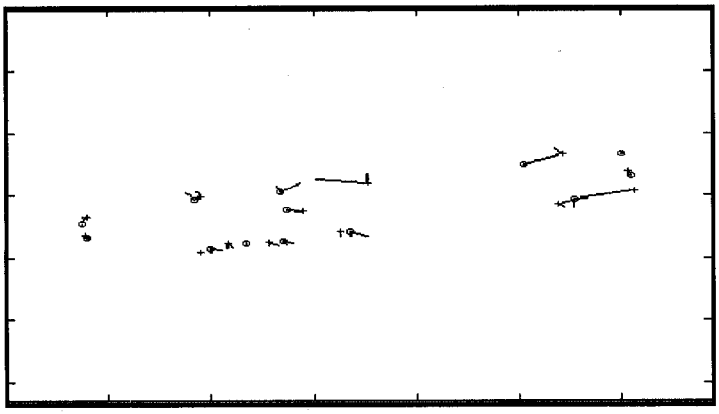
(b)



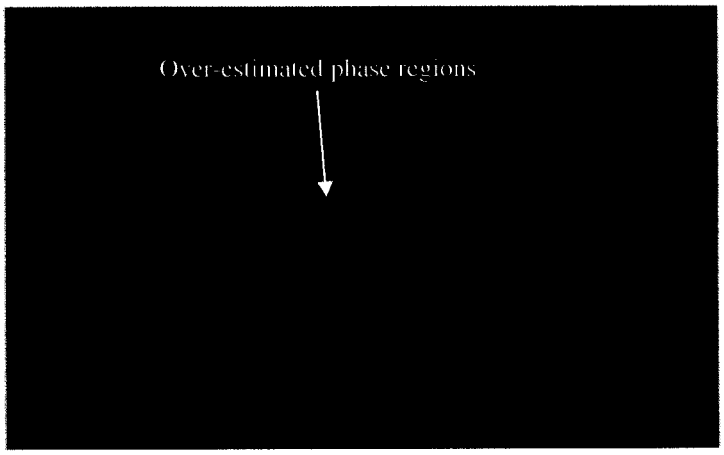
(c)

Figure 5.44 Simulated femoral with non-optimal steering unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $-\pi/2$ phase shift. (c) Unwrapped phase image.

(a)



(b)



(c)

Figure 5.45 Simulated femoral with non-optimal steering unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi / 2$ phase shift. (c) Unwrapped phase image.

5.5.2. *In Vivo* Flow Phantom Data Results

In this section, we will present the results of our phase unwrapping technique on *in vivo* datasets. The datasets were acquired according to the same acquisition protocol used for acquiring the *in vitro* phantom data in terms of control parameters and sample rate sweeping. The acquisition of the datasets was performed by certified sonographers, David Hull and Steve Hill, who work for Philips Ultrasound, Bothell, Washington. Human models were recruited to be scanned using the Philips high end iU22 ultrasound system [4]. The main peripheral arteries were scanned such as the proximal and mid carotid and the jugular vein.

Figure 5.46 shows a 30 cineloop of a proximal carotid artery flow with severe wrapping. The successfully unwrapped phase images are shown in Figure 5.46b. One severely-wrapped frame is shown in Figure 5.47a. It is clear that the *in vivo* data has more complex wrapping patterns than the *in vitro* ones. This is mainly due to the complex nature of the blood flow within the elastic human vessels, which can only be partially simulated using flow phantoms and simple hemodynamic models. The unwrapped phase result shown in Figure 5.47b is successful except for a small region that was erroneously unwrapped and highlighted by the arrow.

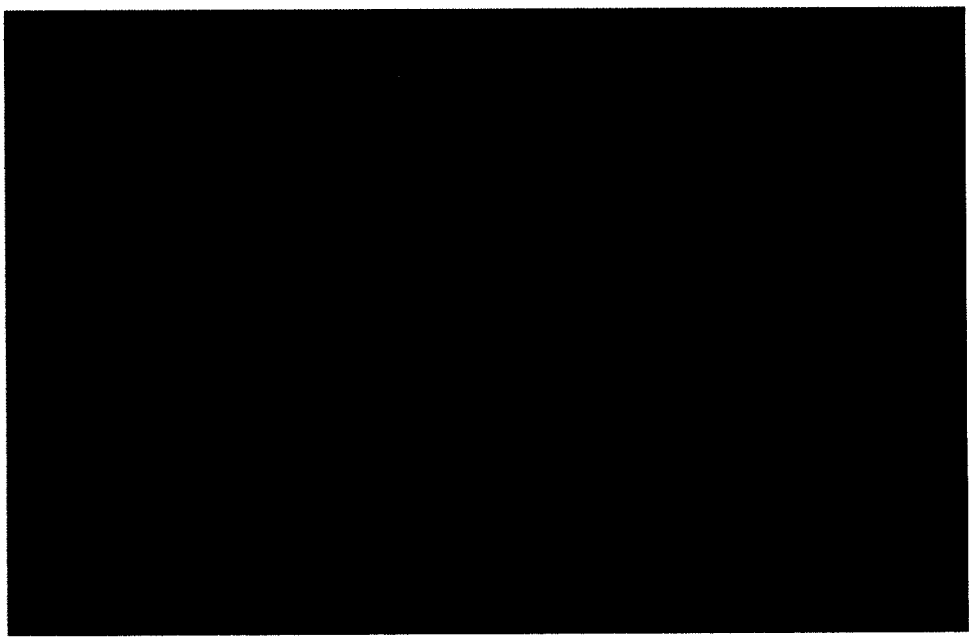
Figure 5.48 shows a more complex case of a mid carotid artery with severe wrapping. The complexity is increased due to the low flow sensitivity, which results in less color Doppler filling within the vessel lumen. The unwrapped phase images are shown in Figure 5.48b. Figure 5.49 shows one severely-wrapped frame of the mid carotid cineloop. The unwrapped phase image has a small region which was over-estimated, as highlighted by the arrow. Figure 5.50 shows another severely-wrapped image that ended with a small region with under-estimated unwrapped phase.

Figure 5.51 shows another cineloop of a severely-wrapped common carotid artery. The unwrapped frames in Figure 5.51b are successful. Figure 5.52 and Figure 5.53 show two frames with severe wrapping and many detected residues. The areas of severe wrapping are

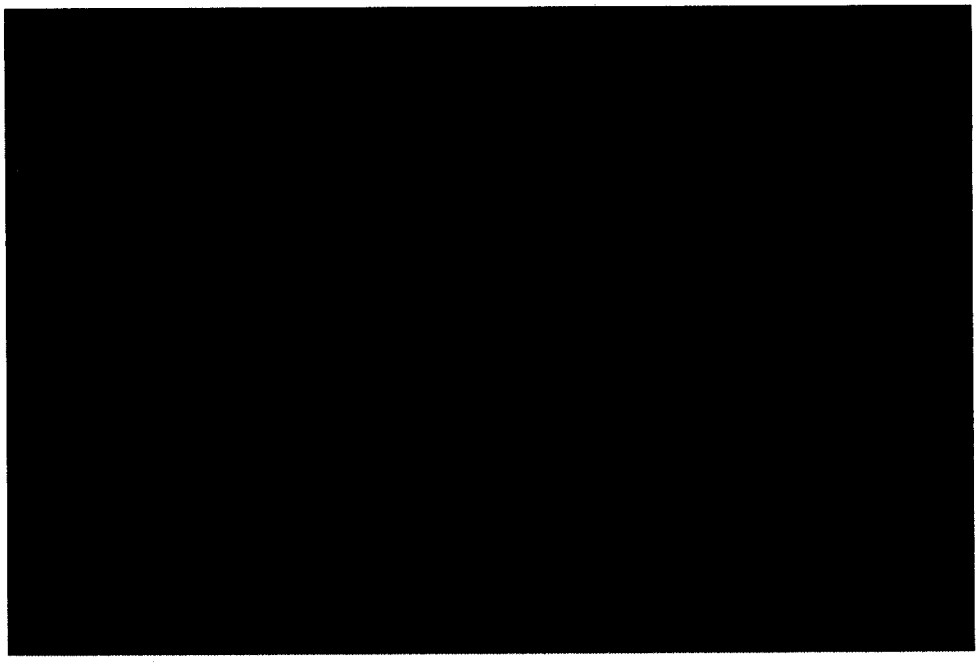
unwrapped correctly as shown in Figure 5.52b and Figure 5.53b and highlighted by the arrows. One important note here is that with the *in vivo* data, it is extremely hard, even for the clinical experts, to accurately identify the correct unwrapping results. This is due to the high degree of distortion of the wrapped image. However it is possible, with accurate inspection of the wrapped image, to identify some regions of severe wrapping and anticipate the unwrapping results.

Figure 5.54 shows a less successful result with a cineloop of a jugular vein where all the frames all severely-wrapped. The unwrapping results shown in Figure 5.54b indicate major errors in three frames (frame numbers 12, 14, and 24) and minor errors in other frames (including frames 2, 10, and 22). The errors were mainly caused by the fact that the vessel has a small cross section, which increased the number of border residues. The border residues are the most problematic ones for our algorithm despite the special handling steps for the border residues that are embedded in the algorithm.

Figure 5.55 shows one of the failing cases where there is a region of over-estimated unwrapped phase. Both phase shift solutions in this case failed and introduced invalid solutions with phase values outside the allowed phase range $(-3\pi,3\pi]$. The solution with the smaller number of invalid phase pixels (with $-\pi/2$ phase shift) was chosen although it was not the logical phase shift for the solution. Figure 5.56 shows another failed case where there is a large region of erroneous unwrapped phase, as highlighted by the arrow. The error was caused by the misplacement of cutlines which allowed for unwrapping errors to propagate to a large region of the vessel. In this case, the phase discontinuity criterion was used, and it picked the wrong phase shift solution ($-\pi/2$), because the logical phase shift introduced a solution with more discontinuity pixels. Figure 5.57 shows one of the successful examples for the jugular vein case. The unwrapping result shown in Figure 5.57b highlights a region of severe wrapping that was successfully unwrapped.

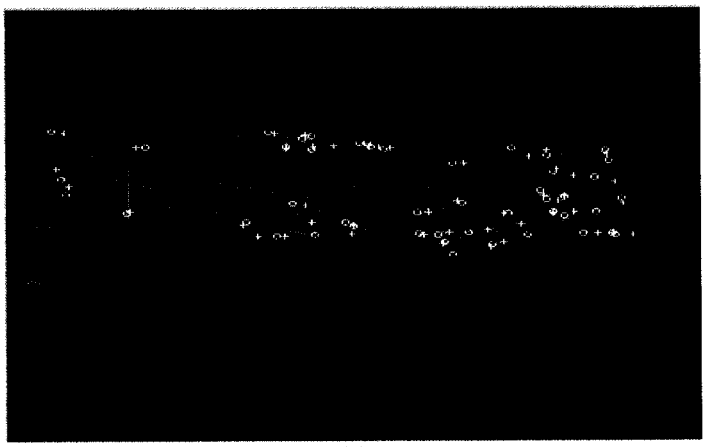


(a)

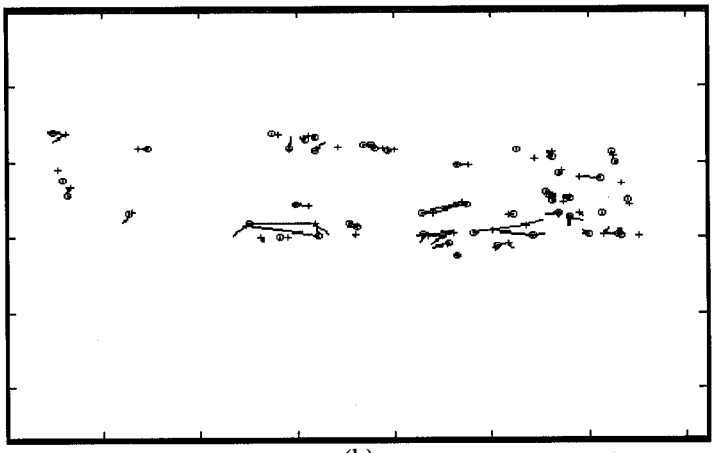


(b)

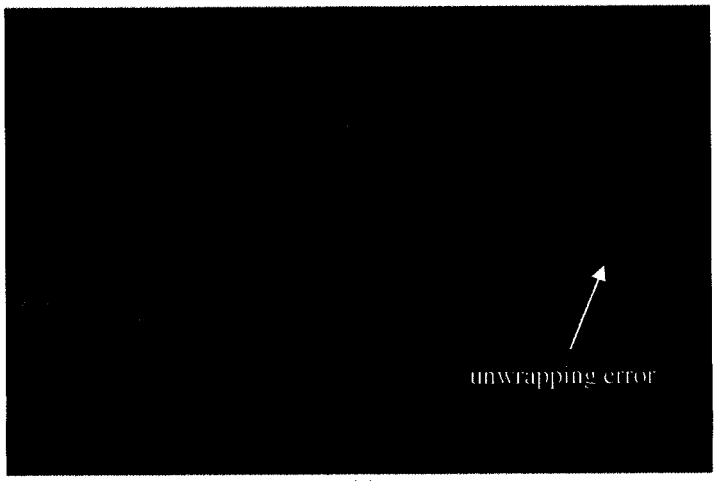
Figure 5.46 *In vivo* proximal common carotid artery flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.



(a)

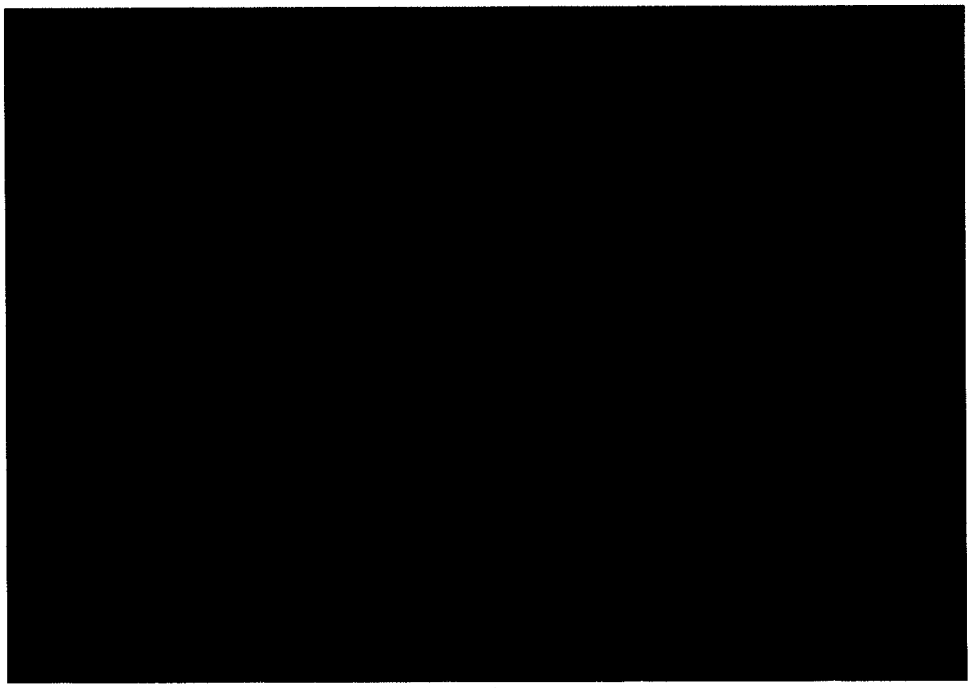


(b)

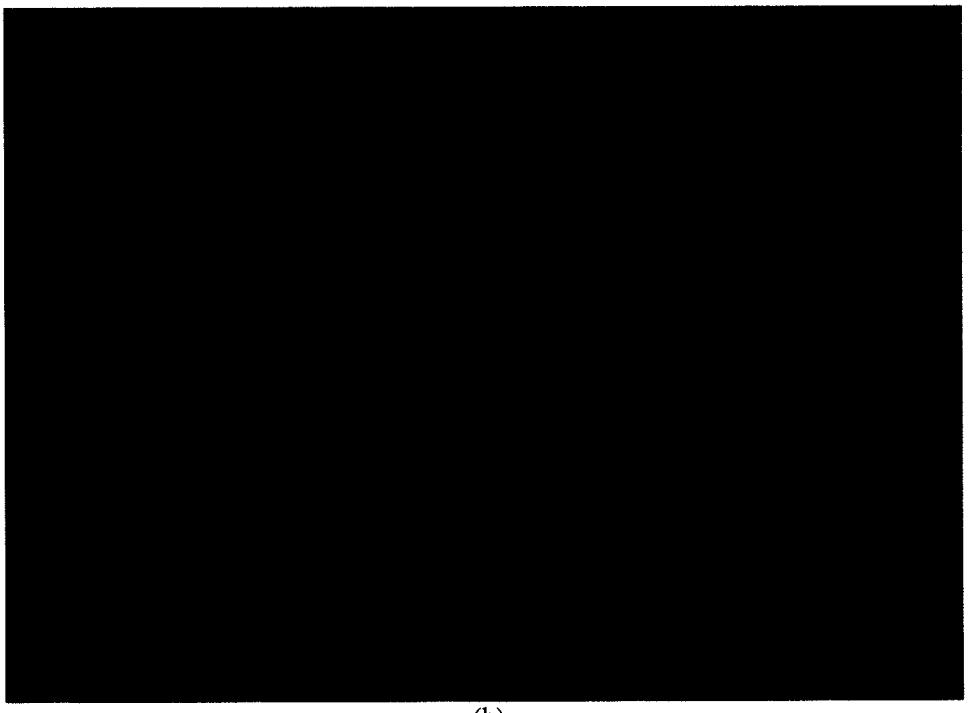


(c)

Figure 5.47 *In vivo* proximal common carotid artery unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

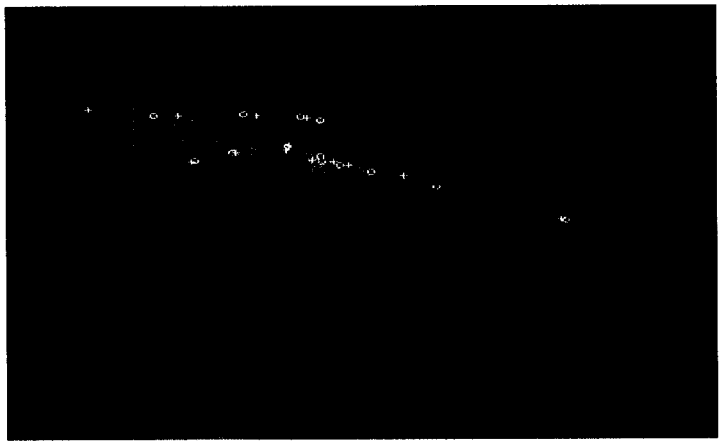


(a)

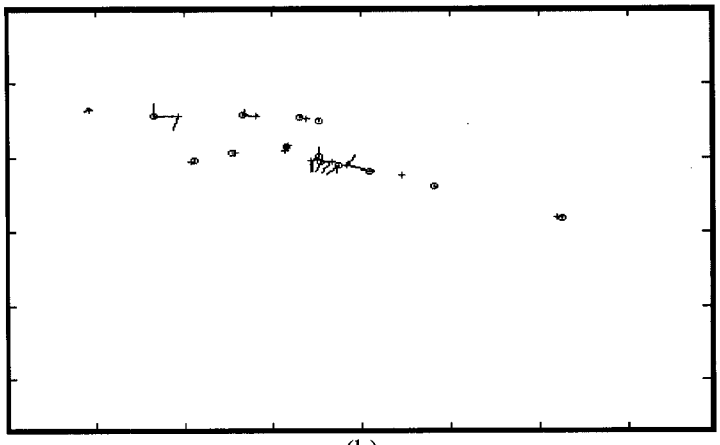


(b)

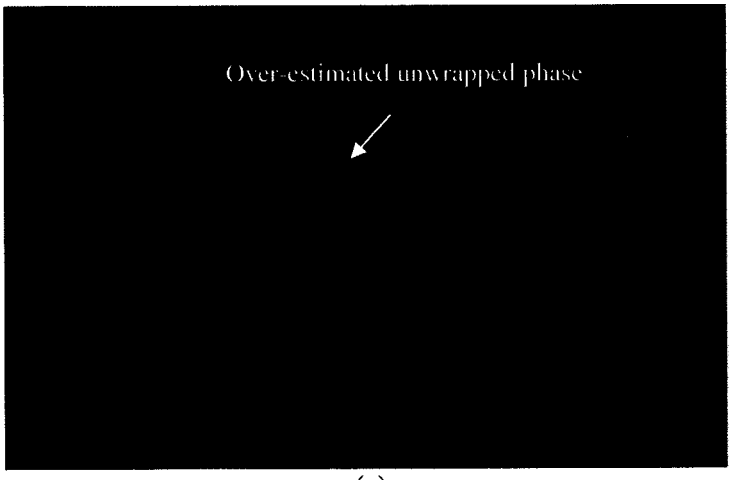
Figure 5.48 *In vivo* mid common carotid artery flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.



(a)



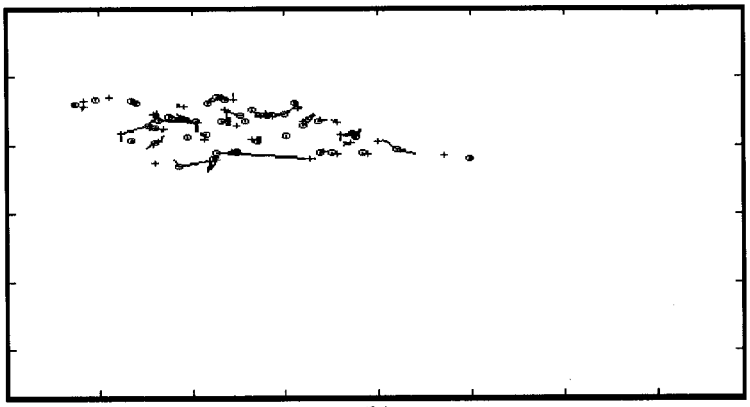
(b)



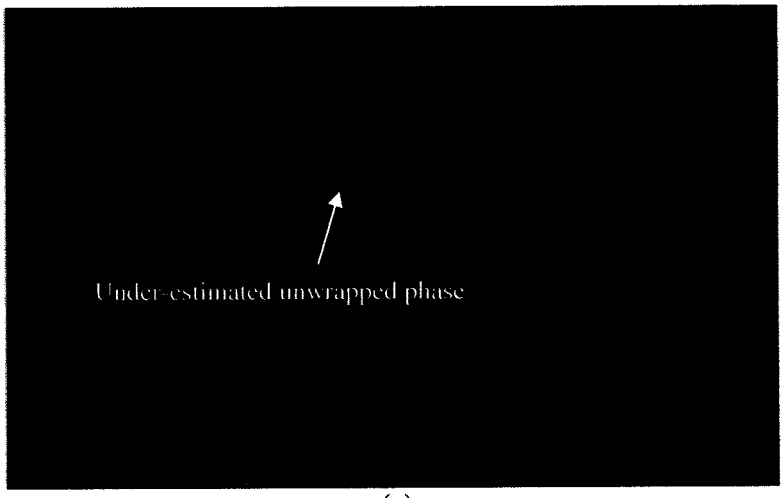
(c)

Figure 5.49 *In vivo* mid common carotid artery unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

(a)

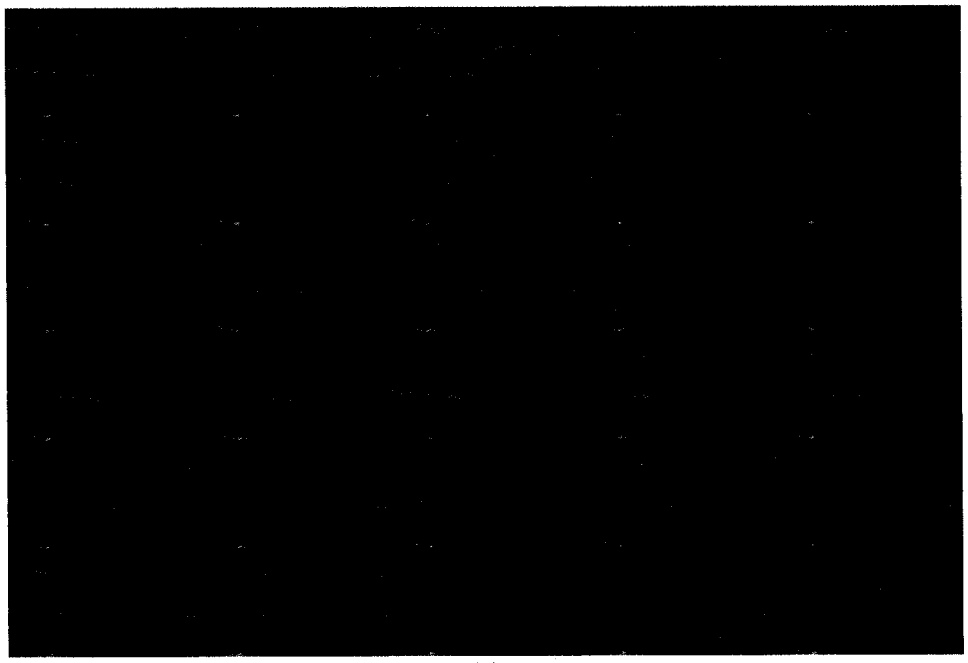


(b)

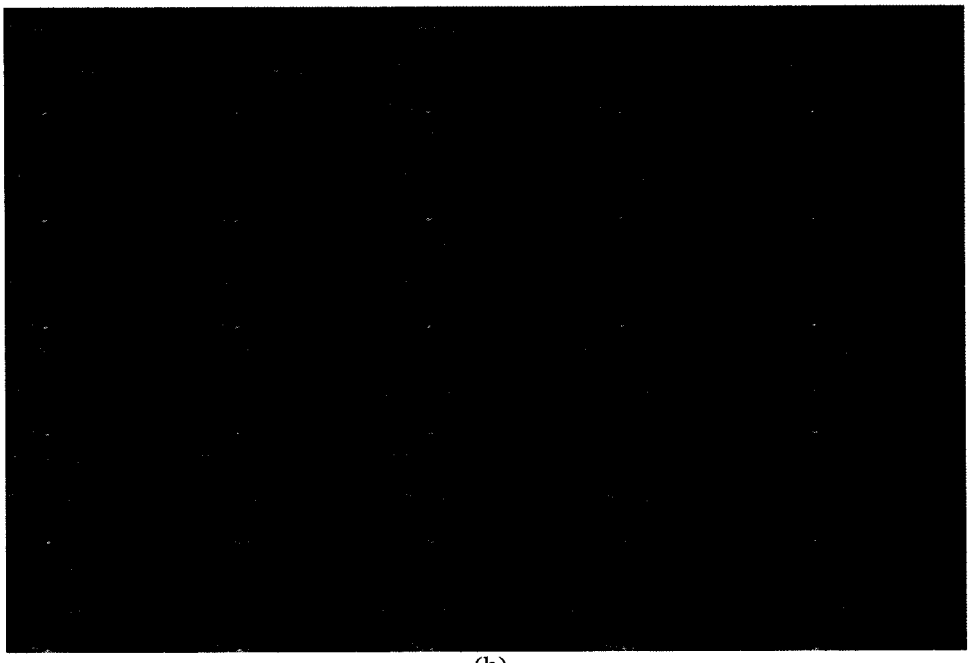


(c)

Figure 5.50 *In vivo* mid common carotid artery unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.



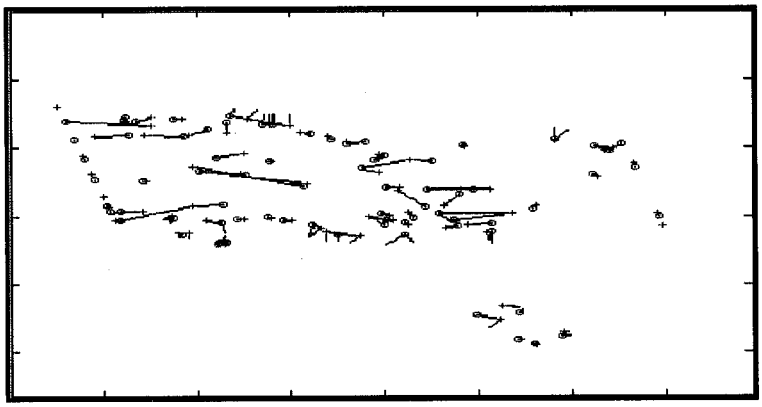
(a)



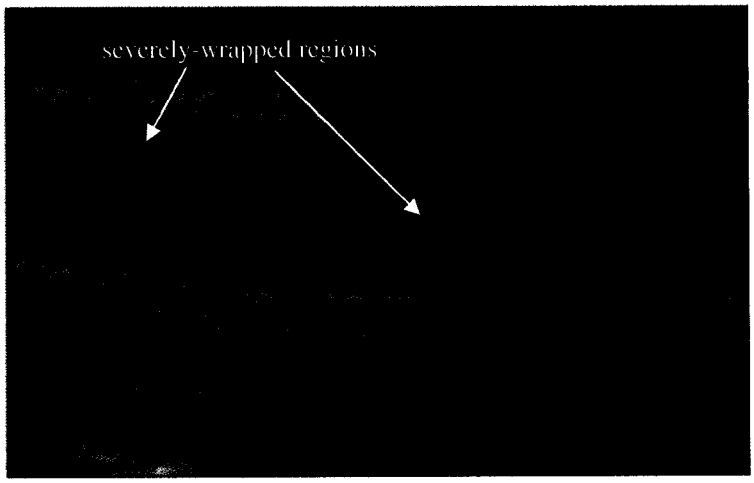
(b)

Figure 5.51 *In vivo* common carotid artery flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.

(a)



(b)



(c)

Figure 5.52 *In vivo* common carotid artery unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

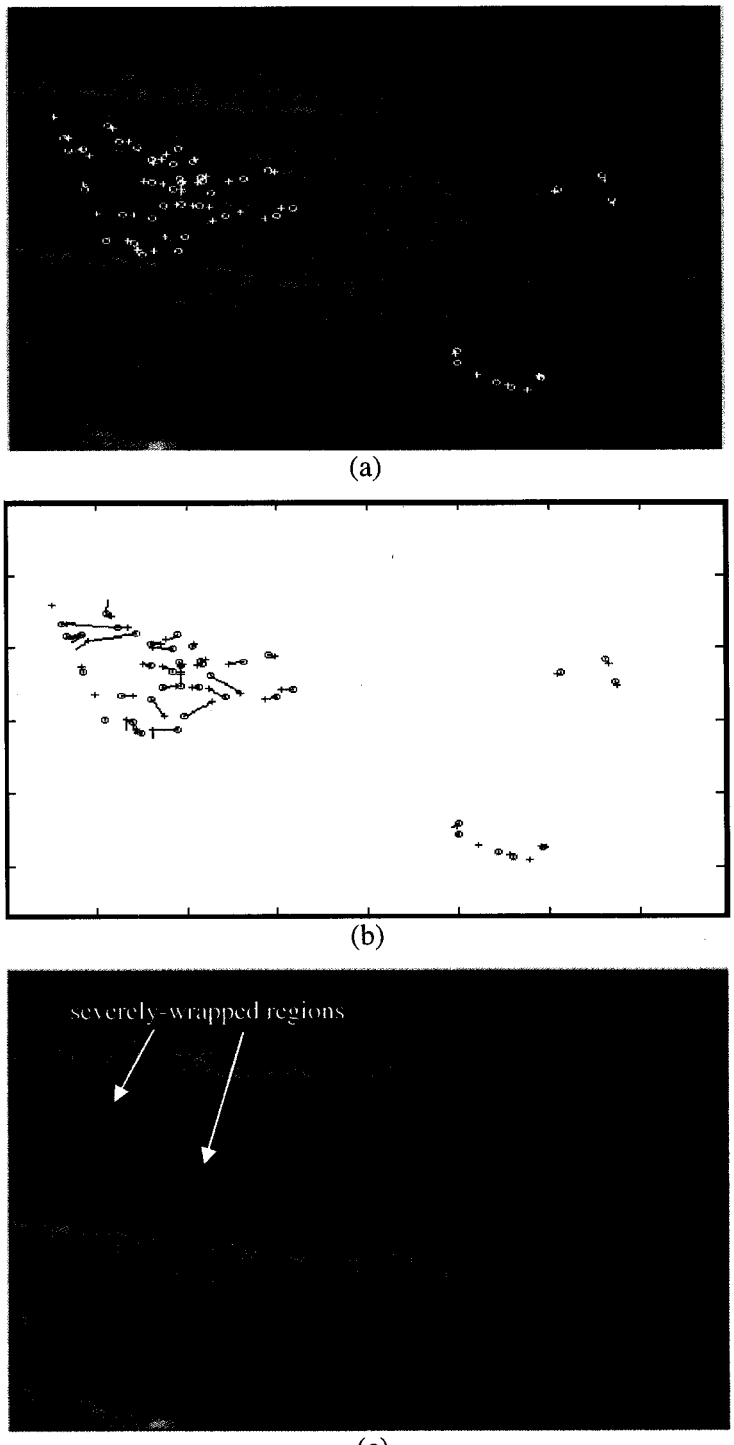
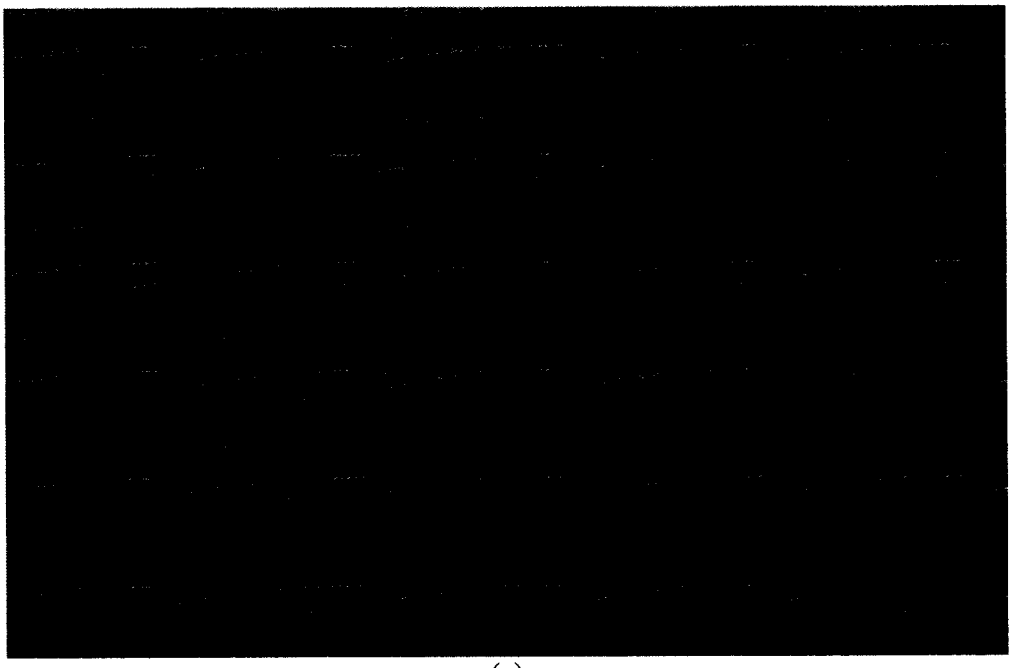
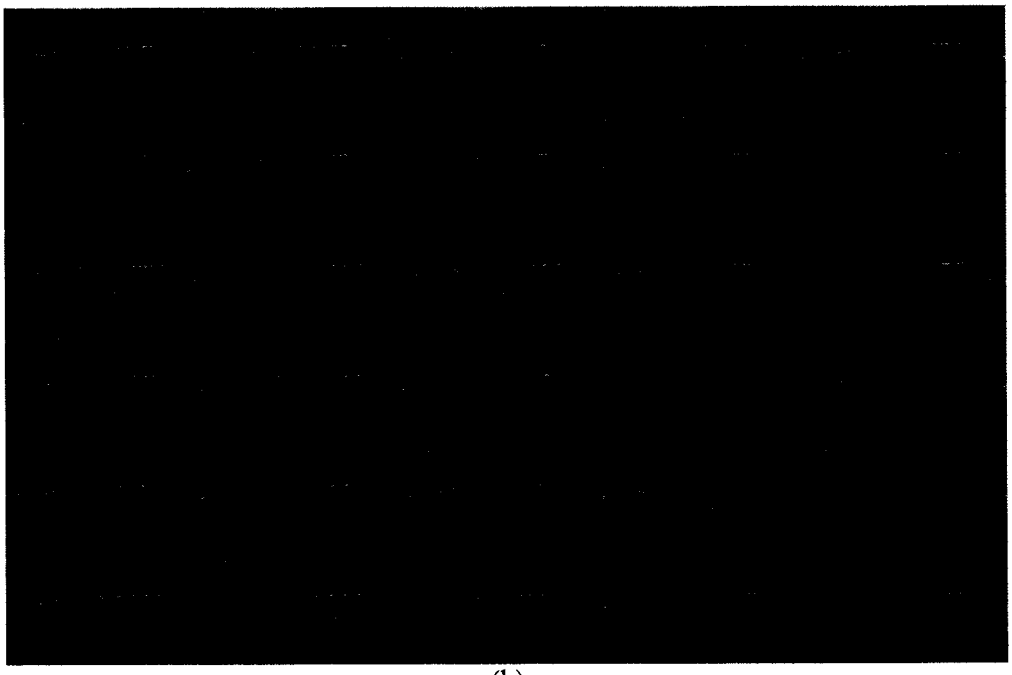


Figure 5.53 *In vivo* common carotid artery unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi / 2$ phase shift. (c) Unwrapped phase image.



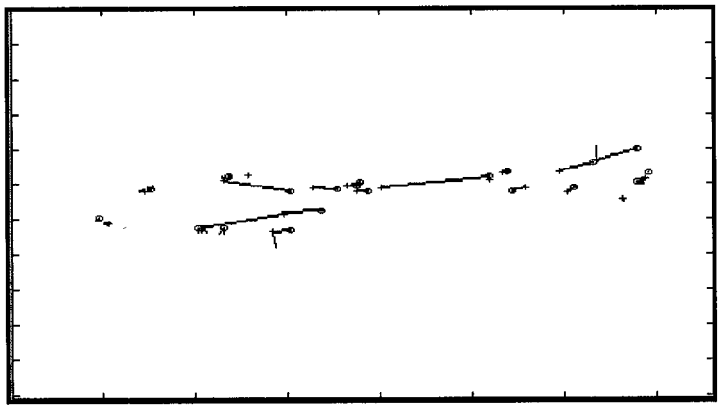
(a)



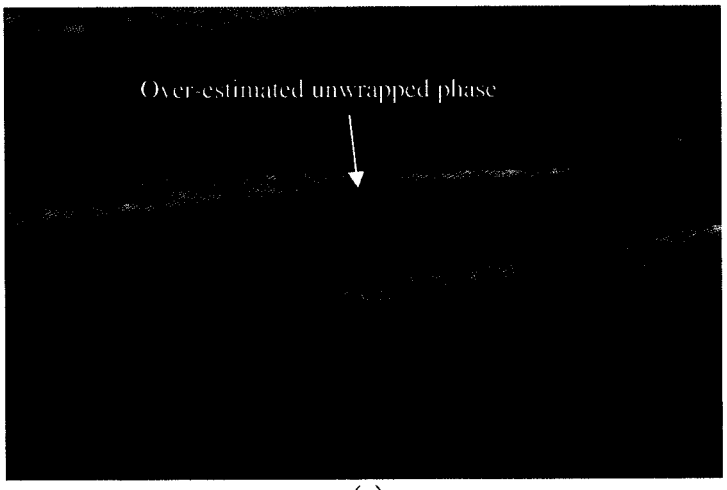
(b)

Figure 5.54 *In vivo* jugular vein flow cineloop. (a) Input wrapped phase images. (b) Output unwrapped phase images.

(a)



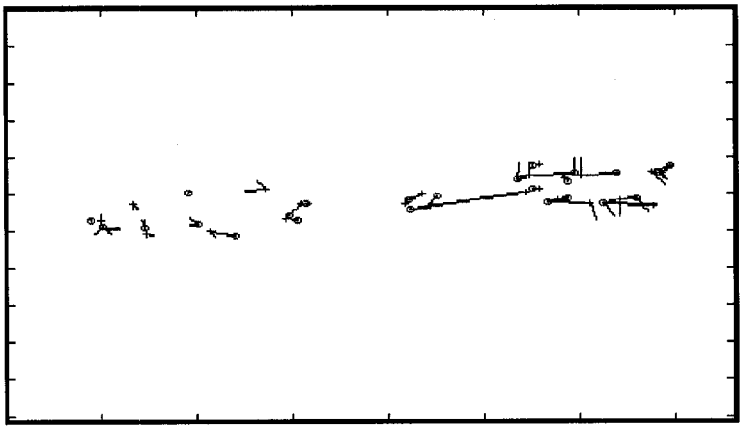
(b)



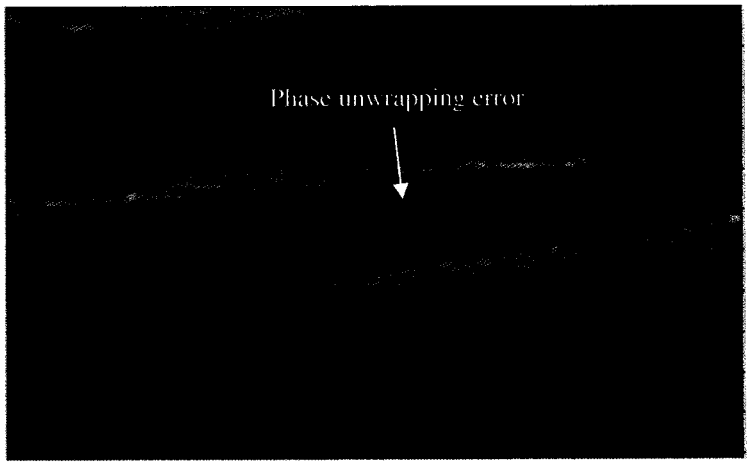
(c)

Figure 5.55 *In vivo* jugular vein unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $-\pi/2$ phase shift. (c) Unwrapped phase image.

(a)

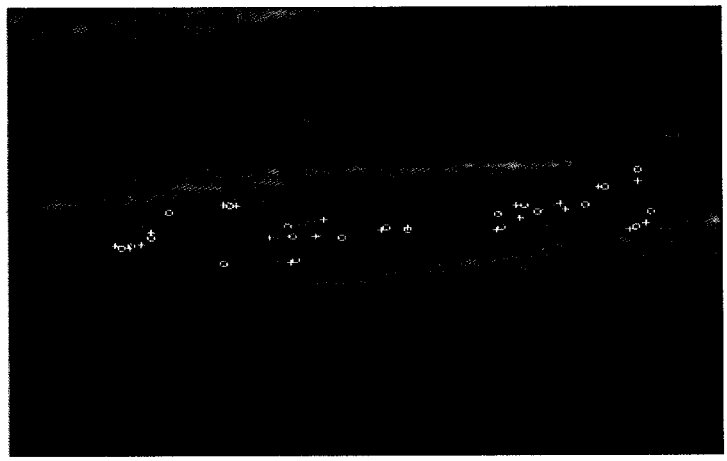


(b)

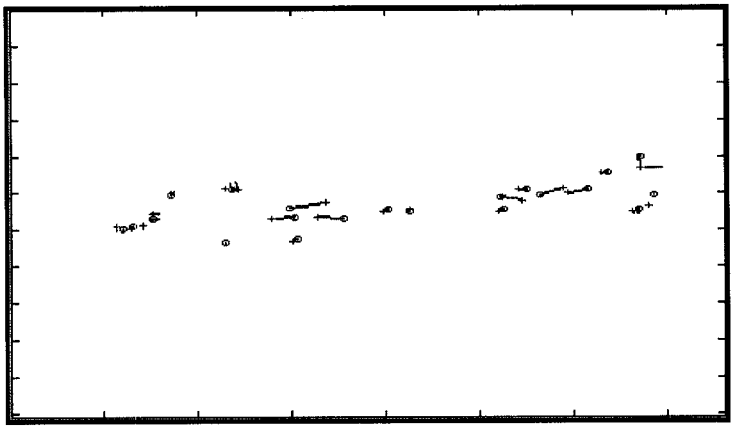


(c)

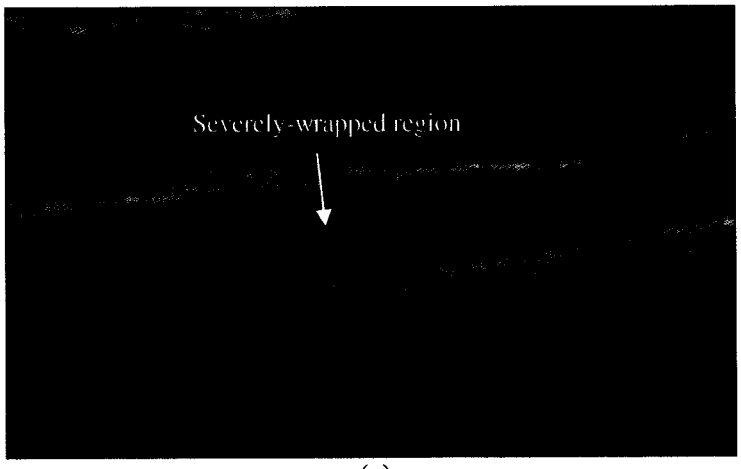
Figure 5.56 *In vivo* jugular vein unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $-\pi/2$ phase shift. (c) Unwrapped phase image.



(a)



(b)



(c)

Figure 5.57 *In vivo* jugular vein unwrapping results. (a) Wrapped phase image. (b) Detected cutlines for $\pi/2$ phase shift. (c) Unwrapped phase image.

5.5.3. Results Comparison with Existing Phase Unwrapping Techniques

To conclude the evaluation of our phase unwrapping algorithm, we show here some results of other phase unwrapping techniques with the same acquired cineloops used to evaluate our algorithm. In the previous chapter, we have presented the results of five well-known phase unwrapping techniques with color Doppler ultrasound data and we concluded that none of them was acceptable in this case. Here, we contrast the performance of our algorithm against some of those other techniques.

Figure 5.58 shows the same cineloop of the common carotid artery that is shown in Figure 5.51 previously. The unwrapping results of Goldstein's algorithm shown in Figure 5.58b indicate the failure of three severely-wrapped frames, as highlighted by the arrows. In contrast, the results of our phase unwrapping technique were successful, as shown in Figure 5.51b.

Figure 5.59 shows the same cineloop of the jugular vein that is shown in Figure 5.54 previously. The unwrapping results of Goldstein's algorithm shown in Figure 5.59b indicate the failure of many severely-wrapped frames, as highlighted by the arrows. In contrast, the results of our phase unwrapping technique had less number of failed unwrapping frames, as shown in Figure 5.54b.

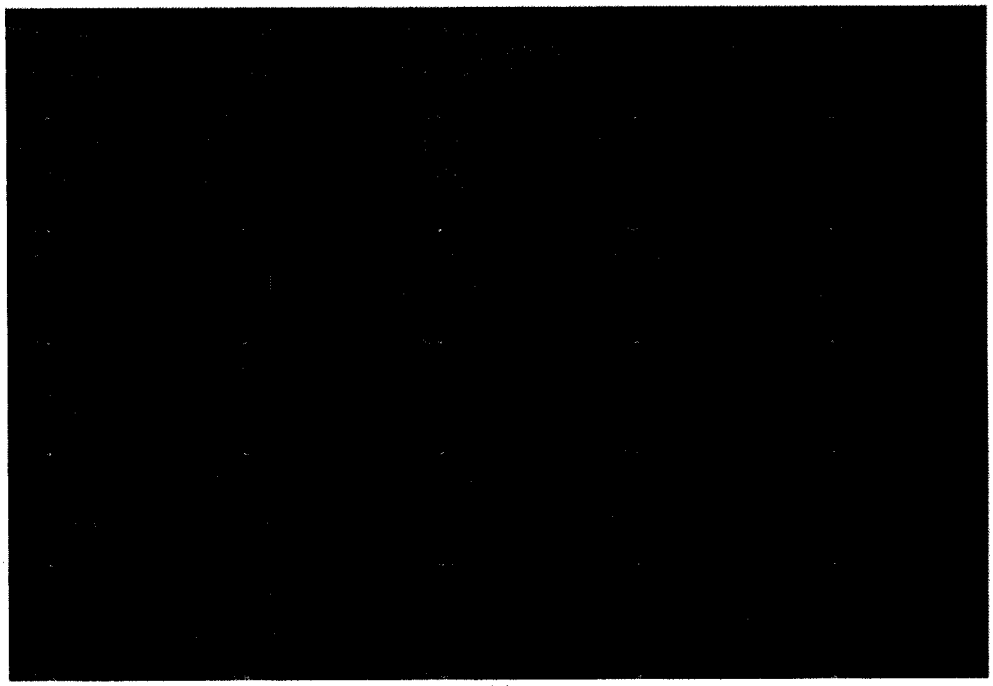
Figure 5.60 shows a cineloop of a jugular vein flow with moderate-wrapping that we did not present before with our algorithm because it was considerably easy case with complete success. The unwrapping results of Flynn's mask cut algorithm shown in Figure 5.60b indicate the failure of all wrapped frames, as highlighted by the arrows. The failure of the mask cut algorithm was mainly due to the failure of the initialization step to pick an originally unwrapped pixel. Our algorithm handles the initialization step gracefully, as explained in section 5.4.1.

Figure 5.61 shows the same cineloop of the *in vitro* triphasic femoral flow that is shown in Figure 5.39 previously. The unwrapping results of Flynn's mask cut algorithm shown in Figure 5.61b indicate the failure of many moderately-wrapped frames, as only some of them are highlighted by the arrows. Mostly all negative phase frames (frames 4 to 12) have completely wrong unwrapping values. In contrast, the results of our phase unwrapping technique were completely successful, as shown in Figure 5.39b.

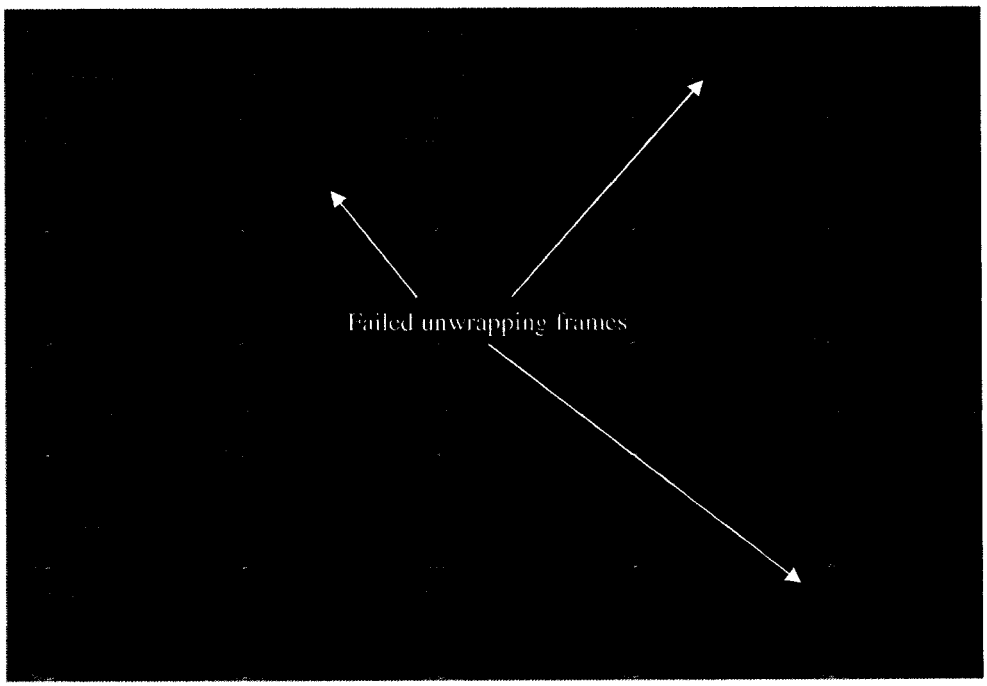
5.5.4. Phase Unwrapping Results Statistical Analysis

In this section we will try to quantify the performance of our phase unwrapping algorithm versus the existing algorithms by deriving some statistics from both the *in vitro* flow phantom and *in vivo* human blood vessel unwrapping results. Overall the *in vitro* data included 16 cineloops with a total of 180 aliased color Doppler frames, 52 of which are severely aliased while the rest are moderately aliased. The *in vivo* data consisted of 12 cineloops with a total of 79 aliased frames, 14 of which are severely aliased.

All the aliased frames were unwrapped using all five phase unwrapping techniques described in Chapter 4. The unwrapping results were qualitatively assessed to decide if a frame was successfully unwrapped or not. The correct unwrapping result was determined from the heuristics inherent in the acquired data including the vessel type, flow direction, sampling rate, and highest velocity from a similar acquisition with high enough sampling rate to avoid aliasing. A color Doppler frame was considered successfully unwrapped if roughly 95% of its area was correctly unwrapped and agreed with the correct phase unwrapping result. A 5% tolerance factor was allowed since it should not affect the forthcoming recognition tasks.

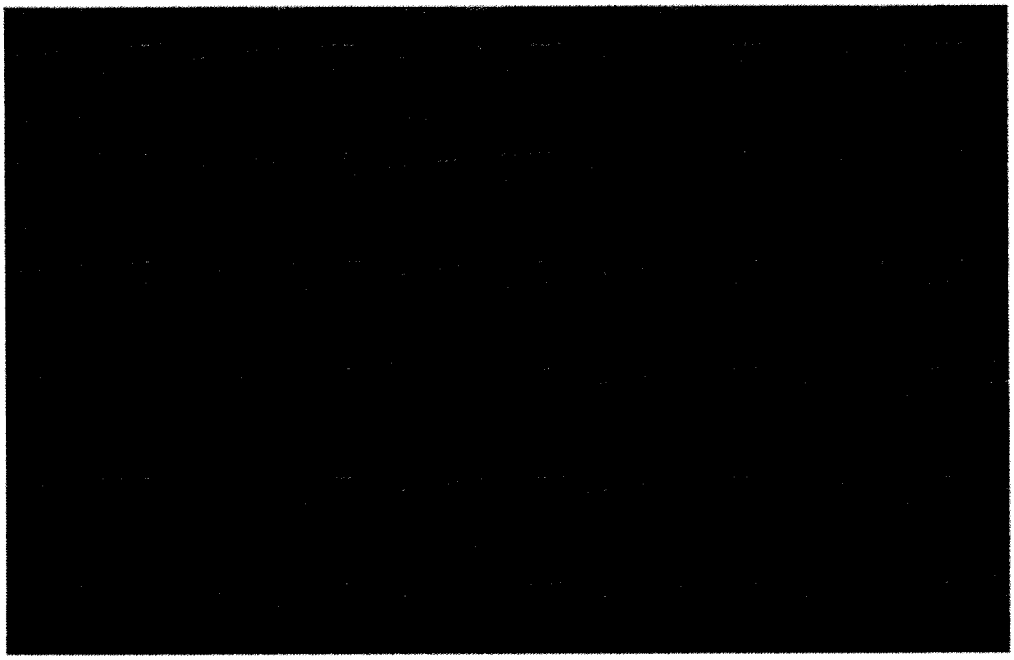


(a)

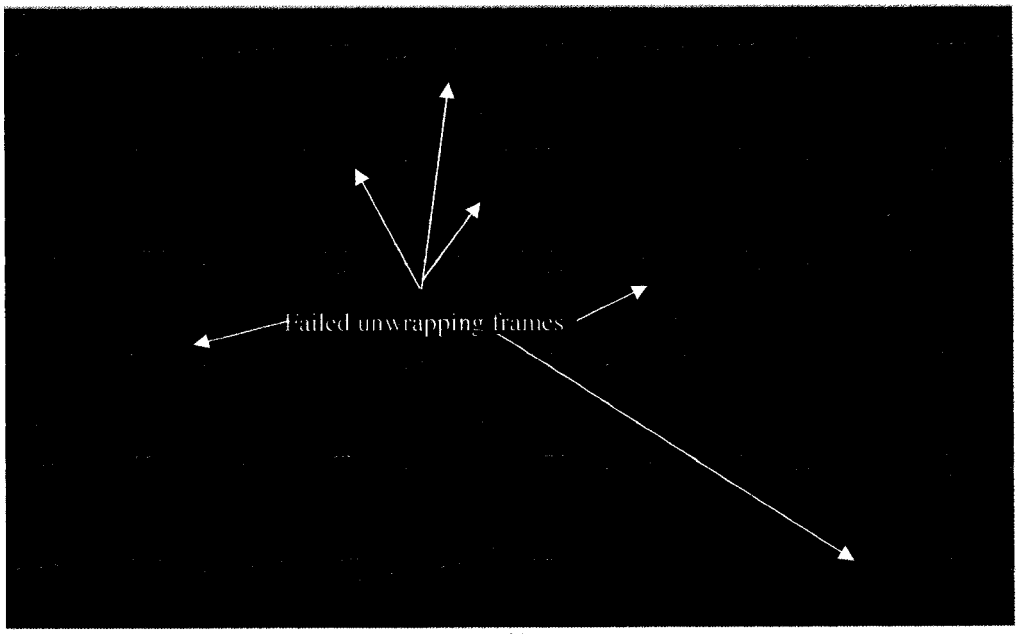


(b)

Figure 5.58 Goldstein phase unwrapping algorithm performance with *in vivo* common carotid artery flow. (a) Input wrapped phase images. (b) Output unwrapped phase images.

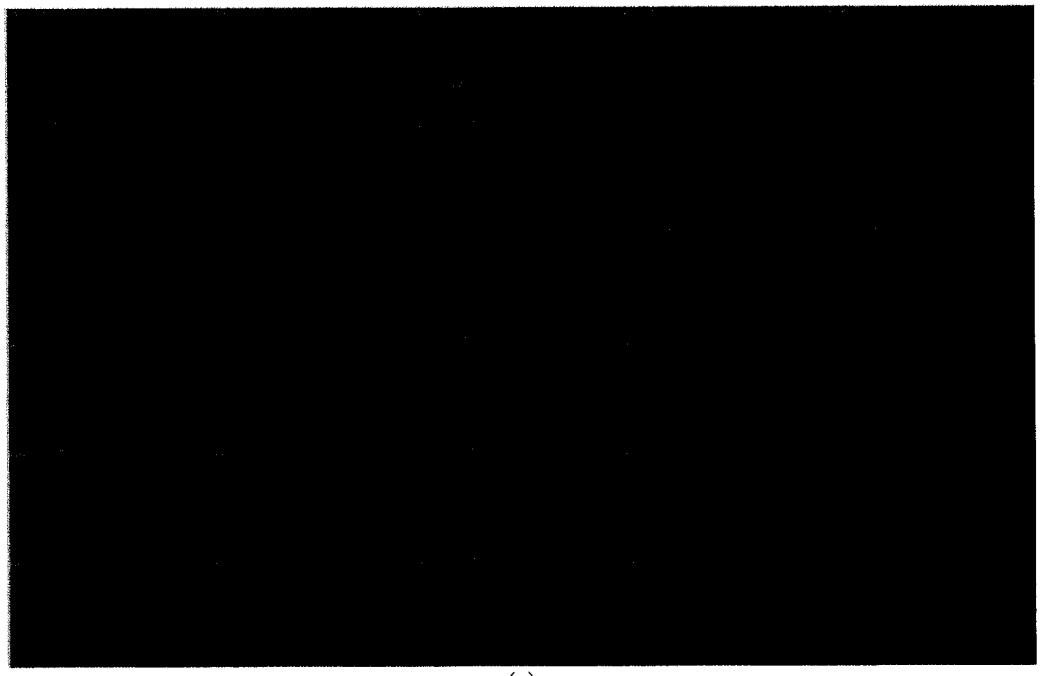


(a)

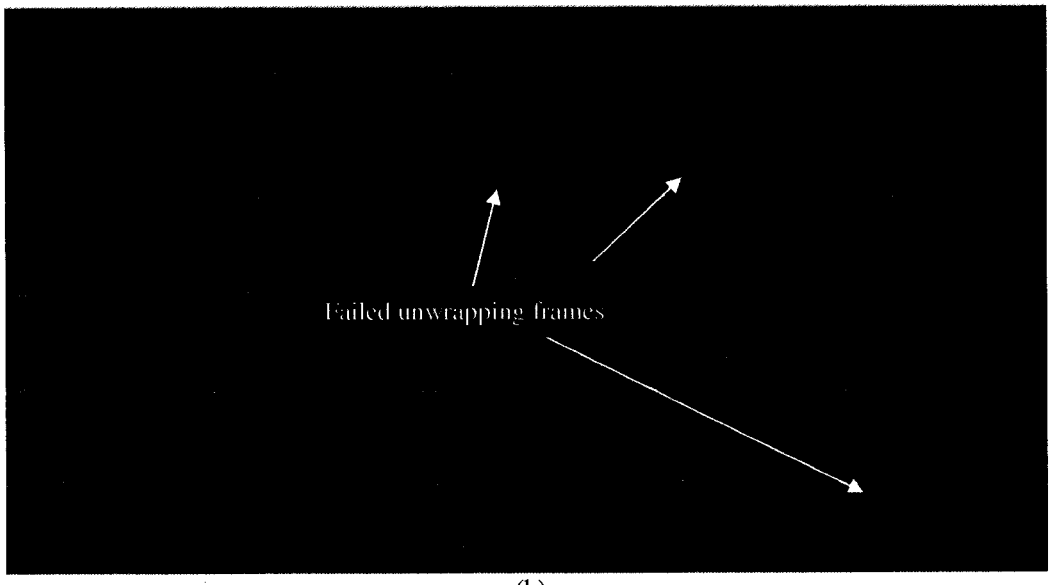


(b)

Figure 5.59 Goldstein phase unwrapping algorithm performance with *in vivo* jugular vein flow. (a) Input wrapped phase images. (b) Output unwrapped phase images.

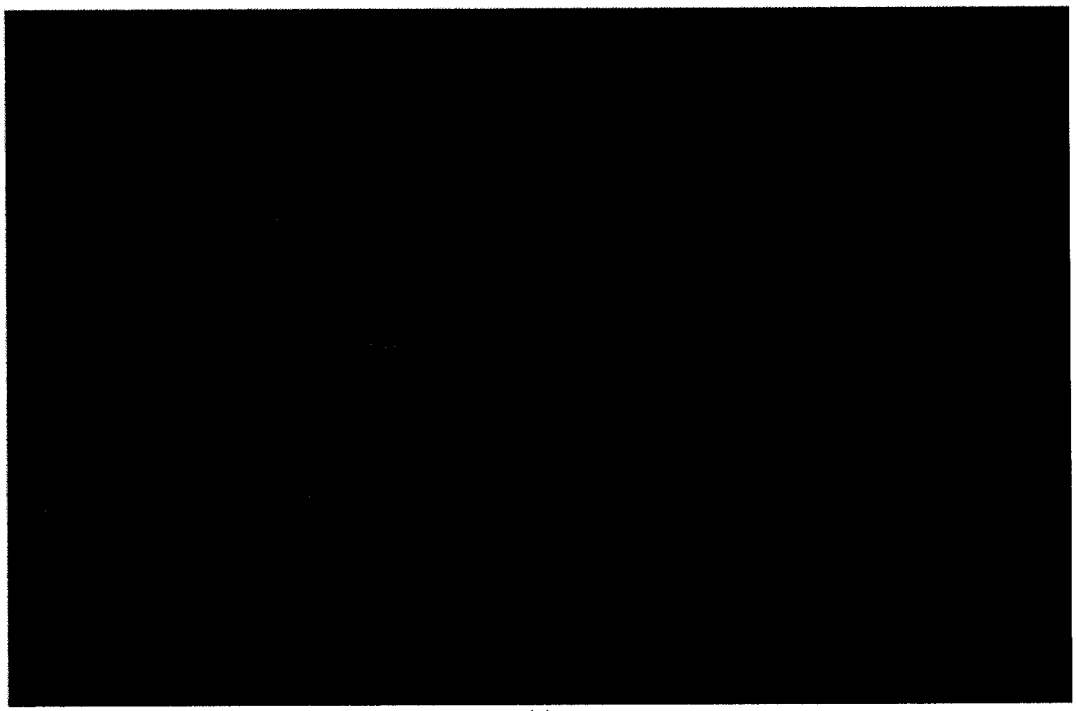


(a)

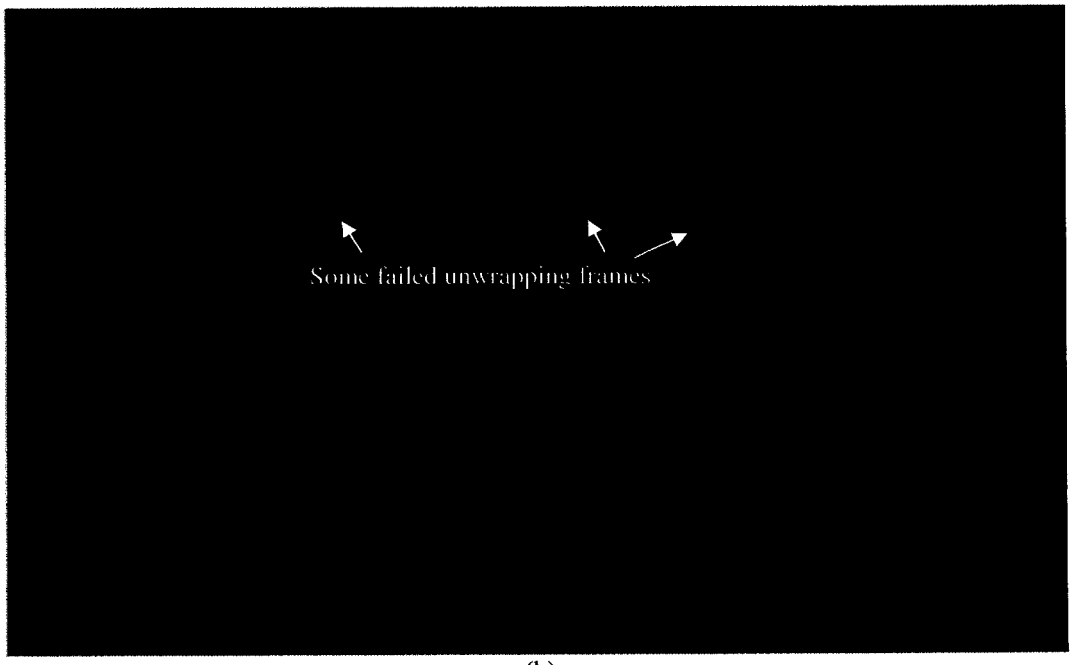


(b)

Figure 5.60 Mask cut phase unwrapping algorithm performance with *in vivo* jugular vein flow. (a) Input wrapped phase images. (b) Output unwrapped phase images.



(a)



(b)

Figure 5.61 Mask cut phase unwrapping algorithm performance with *in vitro* femoral flow. (a) Input wrapped phase images. (b) Output unwrapped phase images.

The unwrapping success rates are listed in the tables below. Table 5.1 lists the number of aliased frames of the *in vitro* data and the number of severely-aliased ones. The numbers of successfully-unwrapped frames as well as the success rates are listed for all the tested algorithms. Table 5.2 lists the unwrapping results for the *in vivo* data, while Table 5.3 combines both *in vitro* and *in vivo* data to give the overall performance of the different algorithms. The following conclusions can be drawn from the derived statistics:

- Our newly developed fringeline-tracking based phase unwrapping technique gave the best success rate among all tested algorithms for all data types (*in vitro*, *in vivo*, and both) and for all aliasing conditions (severely-aliased and moderately aliased).
- The second best algorithm was Flynn's minimum discontinuity algorithm [69]. However the algorithm execution time is very slow compared to all other algorithms, except the PCG algorithm, which is also extremely slow. The performances of both Goldstein's branchcut algorithm [63] and Ghiglia's DCT algorithm [70] were comparable; Ghiglia's Maskcut [68] algorithm performed slightly better than these two algorithms. The least successful algorithm was the PCG algorithm [70], which performs a global minimization.
- In general, the performance of the path-following techniques (our fringeline tracking, Branchcut, Maskcut, and Min. Discontinuity algorithms) is better than those of the global minimization type (DCT and PCG). The global-minimization algorithms introduced many unwrapping errors in the vessel boundary regions, similar to the one shown in Figure 4.49 and in areas where there were no residues at all
- In general, the performance of all algorithms degraded with the *in vivo* data in comparison with their performance with the *in vitro* one. This is expected due to the increased complexity of the *in vivo* data with more residues and added noise from body motion and tissue interaction.
- Many errors introduced by the existing path-following algorithms in the moderately-aliased frames (which do not contain any residues at all or contain only boundary residues) were due to initialization pixels or to wrongly-connected boundary residues. Our algorithm was carefully designed to handle initialization and boundary conditions.

Table 5.1 *In vitro* data phase unwrapping success rates for the different algorithms. The numbers of successfully-unwrapped frames as well as the success rates are listed for all tested algorithms.

	#Images	New Fringeline-tracking	Goldstein's Branch-cut	Flynn's Mask-cut	Flynn's Min. Disc.	Ghiglia's DCT	Ghiglia's PCG
All aliased	180	179	147	158	168	155	139
Success rate %		99%	82%	88%	93%	86%	77%
Severely-aliased	52	52	40	47	52	48	34
Success rate %		100%	77%	90%	100%	92%	65%

Table 5.2 *In vivo* data phase unwrapping success rates for the different algorithms. The numbers of successfully-unwrapped frames as well as the success rates are listed for all tested algorithms.

	#Images	New Fringeline-tracking	Goldstein's Branch-cut	Flynn's Mask-cut	Flynn's Min. Disc.	Ghiglia's DCT	Ghiglia's PCG
All aliased	79	74	65	69	70	59	54
Success rate %		94%	82%	87%	89%	74%	68%
Severely-aliased	14	10	5	7	6	6	4
Success rate %		71%	36%	50%	43%	43%	29%

Table 5.3 All *in vitro* and *in vivo* data phase unwrapping success rates for the different algorithms. The numbers of successfully-unwrapped frames as well as the success rates are listed for all tested algorithms.

	#Images	New Fringeline-tracking	Goldstein's Branch-cut	Flynn's Mask-cut	Flynn's Min. Disc.	Ghiglia's DCT	Ghiglia's PCG
All aliased	259	253	212	227	238	214	193
Success rate %		98%	82%	88%	92%	83%	75%
Severely-aliased	66	62	45	54	58	54	38
Success rate %		94%	68%	82%	88%	82%	58%

5.6. Unwrapping Results Qualitative Validation

So far we have described our novel phase unwrapping technique for color Doppler ultrasound imaging and we have shown the results with many *in vitro* and *in vivo* data. Our comments about the success or failure of the results were all qualitative in nature. We highlighted regions of moderate and severe wrapping that are correctly unwrapped. We also highlighted regions of failed unwrapping including over-estimated, under-estimated, and wrong flow direction regions. The important question that comes to mind is how to validate the unwrapping results? And if there is a better way than the qualitative assessment to measure the successes of any phase unwrapping technique?

Some researchers in the phase unwrapping field tried to answer these questions, either for their own algorithms or for comparing different unwrapping techniques. One way of validating the results is to start with a known artificial unwrapped phase map and apply the phase wrapping operator to that map then try any phase unwrapping technique on the wrapped map and compare the result with the original known unwrapped map. Simulation phase maps were used by many researchers in their published work related to the phase unwrapping problem [60], [74], [75], [76]. The problem with simulated images is that they usually do not reflect the challenges and unique features that exist in real world phase maps.

Another approach for validating the unwrapping results is to use the heuristics provided by the problem domain to anticipate what the correct or optimal unwrapping results should look like. Ghiglia and Pritt used the domain information provided by the SAR imaging acquisition and signal processing to judge the quality of the unwrapping results for SAR images [60], [72], [74]. Chavez et al. [75] utilized the information provided by the MRI physics and signal processing to judge the quality of the unwrapping results. The main problem with this approach is that it is sometimes hard to predict how the unwrapping solution should look, due to the complexity of the input data and the high degree of distortion caused by the wrapping operation.

A third approach for validating the unwrapping results is to try to quantify the unwrapping results. Ghiglia and Pritt, in their phase unwrapping book [60], suggested the use of the L^p-norm error measure, defined in equation (34) above, for evaluating the quality of the unwrapped solution. Their argument was that any phase unwrapping technique tries, directly or indirectly, to minimize some sort of error measure and it is logical to use the same criterion to validate the results afterwards. For example, the authors proved that Goldstein's algorithm approximately minimizes the L^0-norm while Flynn's minimum discontinuity minimizes the L^1-norm, and finally all minimum norm algorithms explicitly minimize a certain L^p-norm, such as the least squares algorithms that minimize the L^2-norm error. The authors described how to generate discontinuity maps that measure the differences between the phase gradients in both the input wrapped and output unwrapped phase images. The discontinuity maps nearly matched the branch cut maps used by some algorithms, which was expected since the branch cuts separate regions of discontinuous phase. The authors developed an evaluation method to judge the quality of the unwrapping results. Their methods relied on combining discontinuity maps with quality maps, if one is used by a certain algorithm. The combined map was qualitatively inspected for isolated, or nearly isolated, phase regions. The authors claimed that these isolated regions most likely have phase unwrapping errors.

The problem with the quantitative evaluation method presented in [60] is that it assumed a nearly homogeneous unwrapped phase map where the discontinuity regions should be minimal. This may be true with SAR images with mainly noise-induced residues, but color Doppler ultrasound imaging is different. We have argued before in section 5.2 that the non-linear signal processing of the ultrasound data is the main cause of phase discontinuity and the creation of residues. The correct solution is not necessarily optimal in the L^p-norm sense.

Our qualitative approach for evaluating the unwrapping results of our algorithm is very similar to the second evaluation approach described above, which is to use the heuristics provided by the problem domain to qualitatively assess the unwrapping results. The first step to achieve this goal was to fully understand the nature of the phase wrapping problem with color Doppler ultrasound images, which we covered in section 5.2 above.

The second step was to develop a controlled experiment to acquire the *in vitro* and *in vivo* datasets to try the algorithm with. The acquisition protocol described in section 4.4.1 achieved this goal by turning off some user controls that would otherwise corrupt the phase images and complicate the phase wrapping problem. The other aspect of the acquisition protocol was to sweep the sampling rate with the same input data, to generate a spectrum of phase wrapping degrees, starting from the totally unwrapped phase, to the moderately-wrapped phase, and finally to the severely-wrapped phase. Knowing the degree of wrapping up front helped with the anticipation of the unwrapping results.

The third step to use the problem domain heuristics to evaluate the results was to utilize the clinical information of the acquired data to anticipate the unwrapping results. Examples of such clinical information are the anatomy of the acquired *in vitro* or *in vivo* vessels, such as the carotid artery, the femoral artery or the jugular vein. The anatomy gives us an idea about the flow pattern to expect, such as the flow direction or the phasicity of the flow. Such information helped with the interpretation of the acquired frames and the resulting unwrapped ones, as explained before in the unwrapping result sections.

The last step for the qualitative evaluation was to consult clinical experts to interpret the unwrapping results. David Hull and Steve Hill, from Philips Ultrasound, thankfully reviewed some of the unwrapping results and commented on their validity. One important note here is that in some complex wrapping cases, specially the *in vivo* data, it is even hard to clinical experts to anticipate how the unwrapping results should look like? In these cases, only some phase regions can be evaluated but others may not be.

5.7. Unwrapping Results Quantitative Validation

The second approach for validating the unwrapping results is quantitative in nature. The idea is to utilize all the datasets of the same vessel with the sweeping sampling rate to estimate the maximum velocity. The sweeping sampling rate covers unaliased, moderately-

aliased, and severely-aliased cases. If the unwrapping results are successful then the estimated maximum velocity should be almost the same.

To reliably estimate the maximum velocity from the unwrapped frames, we excluded the boundary pixels since unwrapping errors tend to be around the vessel boundary and the true maximum velocity tends to occur closer to the vessel axis where the vessel wall fractions are minimal. The distance transform of the vessel was used to exclude the boundary pixels where 20% of the vessel pixels with the lowest transform values were excluded. Three different maximum velocity estimators were tried. The first maximum velocity estimator is the highest or peak velocity of all the velocity data; the second estimator is the median velocity of the highest 10% velocity pixels; while the third estimator is the trimmed mean of the highest 10% velocity pixels. The outer 10% pixels on each side are excluded from the trimmed mean. The following set of figures give some examples of the estimated maximum velocity for different datasets.

Figure 5.62 shows the maximum velocity three estimators for an *in vitro* carotid dataset. The horizontal axis represents the decreasing Nyquist velocity which correlates to the sampling rate, while the vertical axis represents the estimated maximum velocity from the unwrapped cineloop of the corresponding sampling rate. The mean and double standard deviation lines are also shown in the figure. Figure 5.63 shows another example of an *in vivo* carotid dataset, while Figure 5.64 shows a third example of an *in vivo* jugular vein dataset. The following conclusions can be drawn from these figures:

- The estimated maximum velocity is within an acceptable limit for the three estimator types, which indicates the success of our phase unwrapping algorithm to recover the true unaliased velocities from the distorted aliased ones.
- Both the median and trimmed mean estimators are more robust than the peak estimator due to their ability to exclude outlier overestimated or underestimated velocities. The trade off of these two estimators is the slight underestimation of the maximum velocity when compared with the unaliased peak estimator maximum velocities.

- The estimated maximum velocity overshooting or undershooting with the severely-aliased cases are mainly due to the inaccuracy of the detected cutlines which may misclassify some pixels as unaliased or aliased.

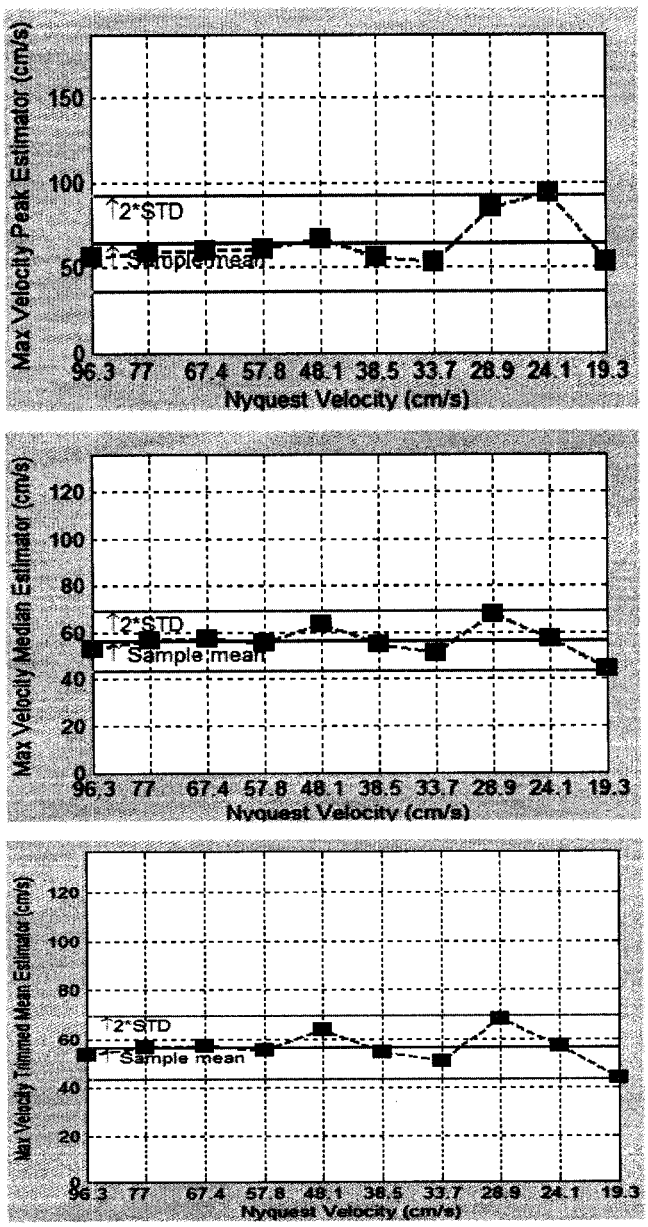


Figure 5.62 *In vitro* carotid phase unwrapping results quantitative validation. Maximum velocity estimation of all sweeping sampling rate cineloops. Top: max. velocity peak estimator, middle: max. velocity median estimator, bottom: max. velocity trimmed mean estimator.

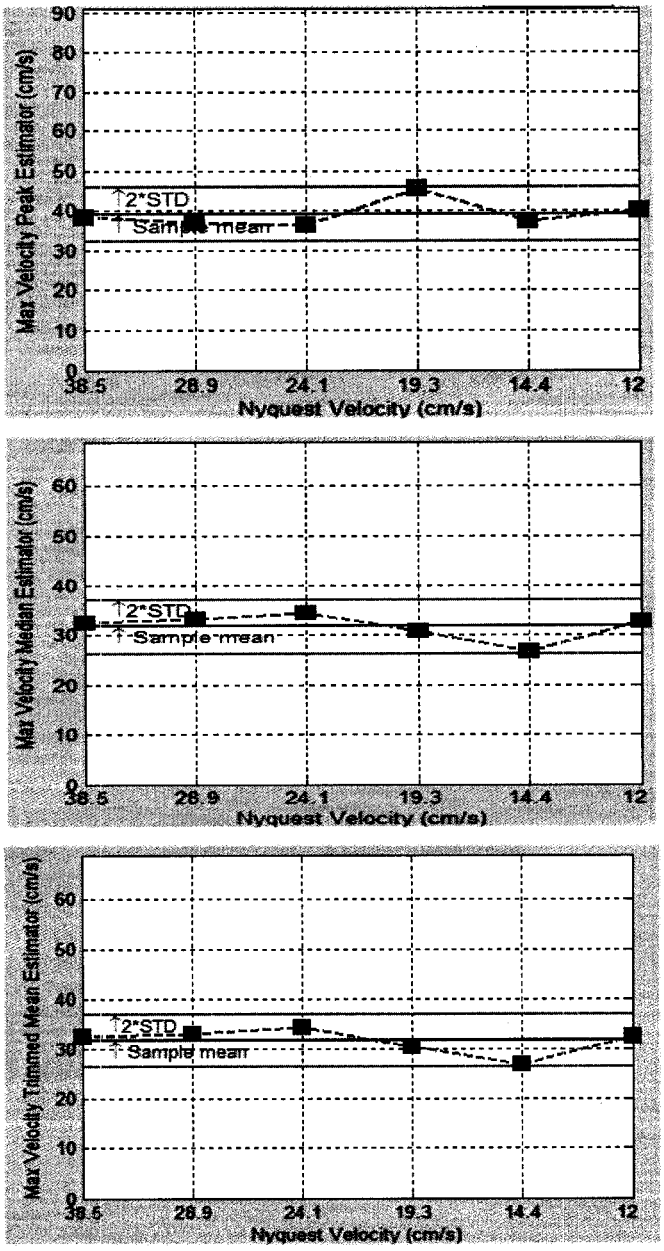


Figure 5.63 *In vivo* carotid phase unwrapping results quantitative validation. Maximum velocity estimation of all sweeping sampling rate cineloops. Top: max. velocity peak estimator, middle: max. velocity median estimator, bottom: max. velocity trimmed mean estimator.

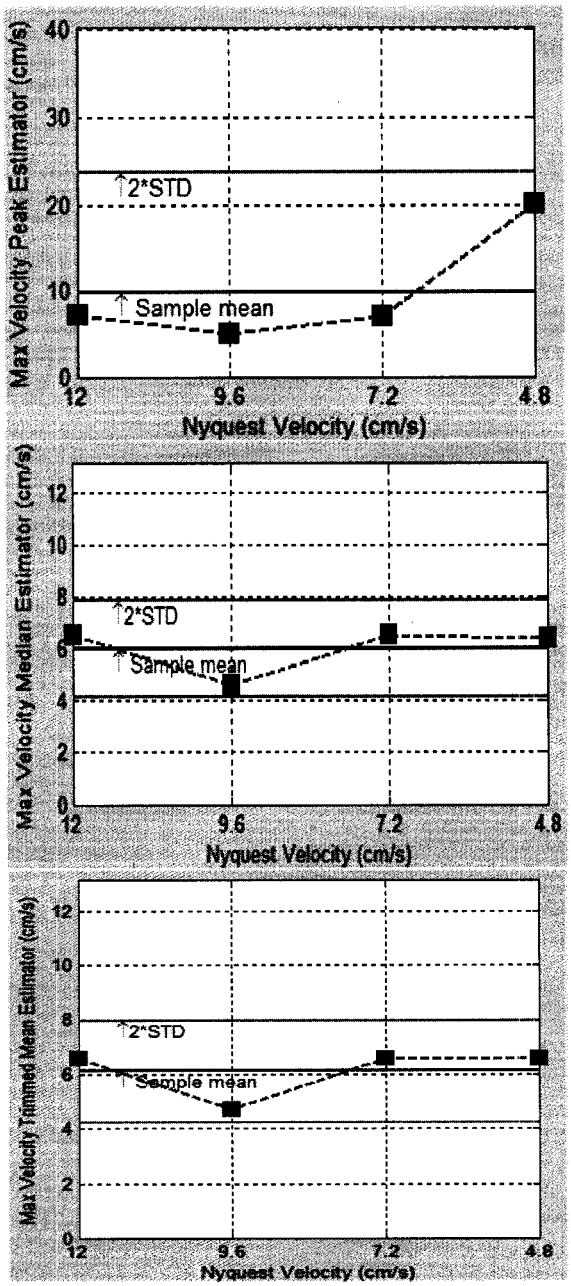


Figure 5.64 *In vivo* jugular vein phase unwrapping results quantitative validation. Maximum velocity estimation of all sweeping sampling rate cineloops. Top: max. velocity peak estimator, middle: max. velocity median estimator, bottom: max. velocity trimmed mean estimator.

5.8. Conclusions and Future Work

In this chapter we have introduced our novel phase unwrapping technique for color Doppler ultrasound phase imaging. The wrapping problem was formulated and analyzed. The core idea behind the algorithm, based on the phase-shifted fringeline tracking, was explained and the details of the algorithm were described. The unwrapping results were presented with both *in vitro* and *in vivo* data. Finally the results validation approach was described. The results of the new phase unwrapping algorithm were published in the SPIE Medical Imaging conference in 2008 [81].

The main advantage of our unwrapping algorithm is the accurate placement of the cutlines, which results in accurate unwrapping for the severely-wrapped regions. In the next chapter we will see how the quantification of the color Doppler data will benefit from this accurate unwrapping by providing good representation of the underlying blood vessel.

Although we have not done comprehensive numerical evaluation of the execution time of our algorithm, we can comfortably expect it to be suitable for interactive, if not real-time, ultrasound applications. This is due to the simplicity of the algorithm that is based on phase-shifted fringeline tracking, which is a linear operation of the number of detected residues. Our technique requires fringeline tracking for only two phase shift values $-\pi/2$ and $\pi/2$. This makes it much faster than the other fringeline tracking based approaches [75] [76], which require tracking fringelines of many phase shifted images, typically much more than two.

The last step of the algorithm, to select one of the two potential solutions, introduced a number of criteria to measure the validity of an unwrapping solution. The discontinuity criterion was the most interesting one, since it utilized the clinical fact that the blood flow will have a dominant direction at a certain instant of time for mainly laminar flow. Turbulent flow may have more complicated flow patterns but it is outside the scope of our work.

The initialization step that is part of our algorithm and described in section 5.4.1 greatly enhanced the performance of the algorithm. This step was often omitted by the existing phase unwrapping techniques in the literature. This may be due to the nature of the images in which there are a small number of wrapped phase pixels, as the case with the MRI images in [16]. In this case, the probability that the initial pixel used for the unwrapping process is wrongly aliased is minimal, and hence the initialization step is not that critical. However this is absolutely not the case with color Doppler ultrasound images in which the whole frame may be wrapped except for very few pixels. The constant venous cineloop in Figure 5.36 is a good example of such a case. This also explains the struggle of some classical phase unwrapping algorithms, even with simple wrapping cases, with ultrasound data; many cases failed due to errors in the initialization step. Our initialization algorithm that detected the original flow direction successfully incorporated both temporal and spatial aspects of the ultrasound data to reliably detect the flow direction. One aspect alone was not sufficient to cover the variability inherent in the different flow types imaged by ultrasound color Doppler.

Despite the very good performance of our phase unwrapping technique with ultrasound data, there is still a room for many enhancements that could increase the success rate even more. One of these enhancements is the notion of *adaptive unwrapping* where the unwrapping process can sense the introduction of phase unwrapping errors and can prevent the propagation of such errors and try to recover from that region and start over in another region. For example, the failed cases of the traditional phase unwrapping techniques presented in section 5.5.3 showed large portions of the images with wrong unwrapping values. This is mainly due to the propagation of the errors, as soon as they get generated, to the neighboring regions. One can develop such an adaptive unwrapping process that senses the generation of errors and try to recover from that state. This would be a completely new methodology that could be incorporated in any existing phase unwrapping technique and greatly enhance its performance.

Other enhancements may include better handling of the border residues, developing a probabilistic model for the dipole detection step (instead of the greedy technique that we

applied), and finally developing a better post-processing technique that selects one of the two potential unwrapping solutions, which could be a probabilistic model as well.

In Chapter 3 we discussed one potential clinical application that relies on the vessel segmentation, which is the Doppler angle automation. Here we will briefly discuss another potential clinical application that can rely on the phase unwrapping results. The idea is to develop a new display mode for the unwrapped color Doppler velocities that will allow easier interpretation of the blood flow. Currently the aliasing degree is qualitatively used to assess the degree of stenosis and turbulence in the vessel [3]. The accurate identification of the highest velocity spot within the vessel is a critical task during the Doppler exam. With the current technology, it is extremely hard to identify the highest stenotic spot from the aliased color Doppler image, instead clinicians rely on the spectral Doppler mode where they navigate around for the highest stenotic velocity using the Doppler sample volume; this process is very time-consuming [78].

Our proposal is to plot the unwrapped velocities in a 3-D rendering presentation where it will be easier to interpret the true velocities and identify regions of high velocity. Figure 5.65, Figure 5.66, and Figure 5.67 show snapshots of the 3-D rendering of the unwrapped velocities of the *in vitro* carotid, constant, and femoral waveforms respectively. The phasicity and pulsatility of the underlying waveforms are clear and the areas of highest velocity can easily be identified on these 3-D renderings, much easier than from the aliased cineloops or the spectral Doppler navigation modes. Although the idea looks promising, real-time implementation and clinical trials are crucial to test the validity of this new display concept of color Doppler images.

In the next chapter we will discuss the feature extraction step that is based on the phase unwrapping results presented in this chapter.

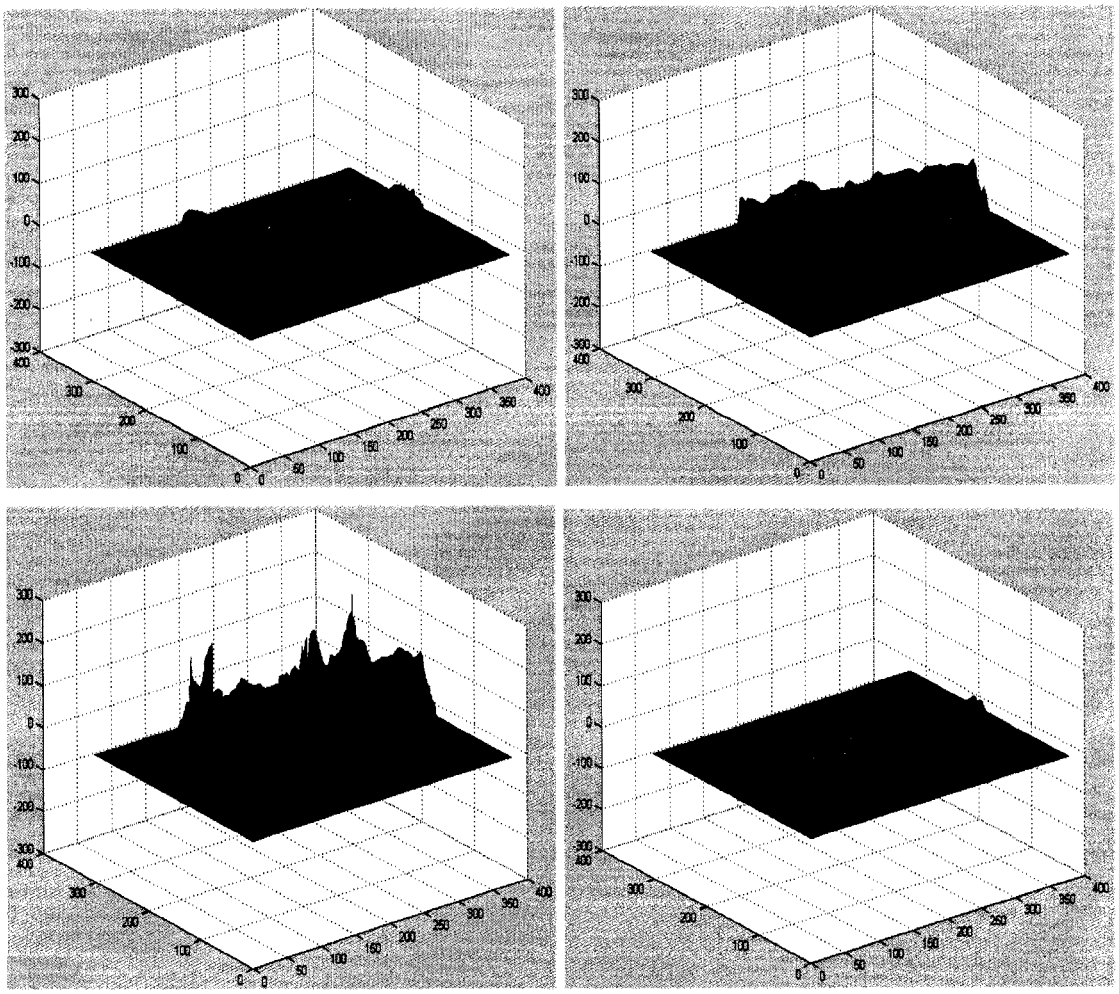


Figure 5.65 Screen shots of the 3-D rendering of the unwrapped velocities of an *in vitro* carotid waveform.

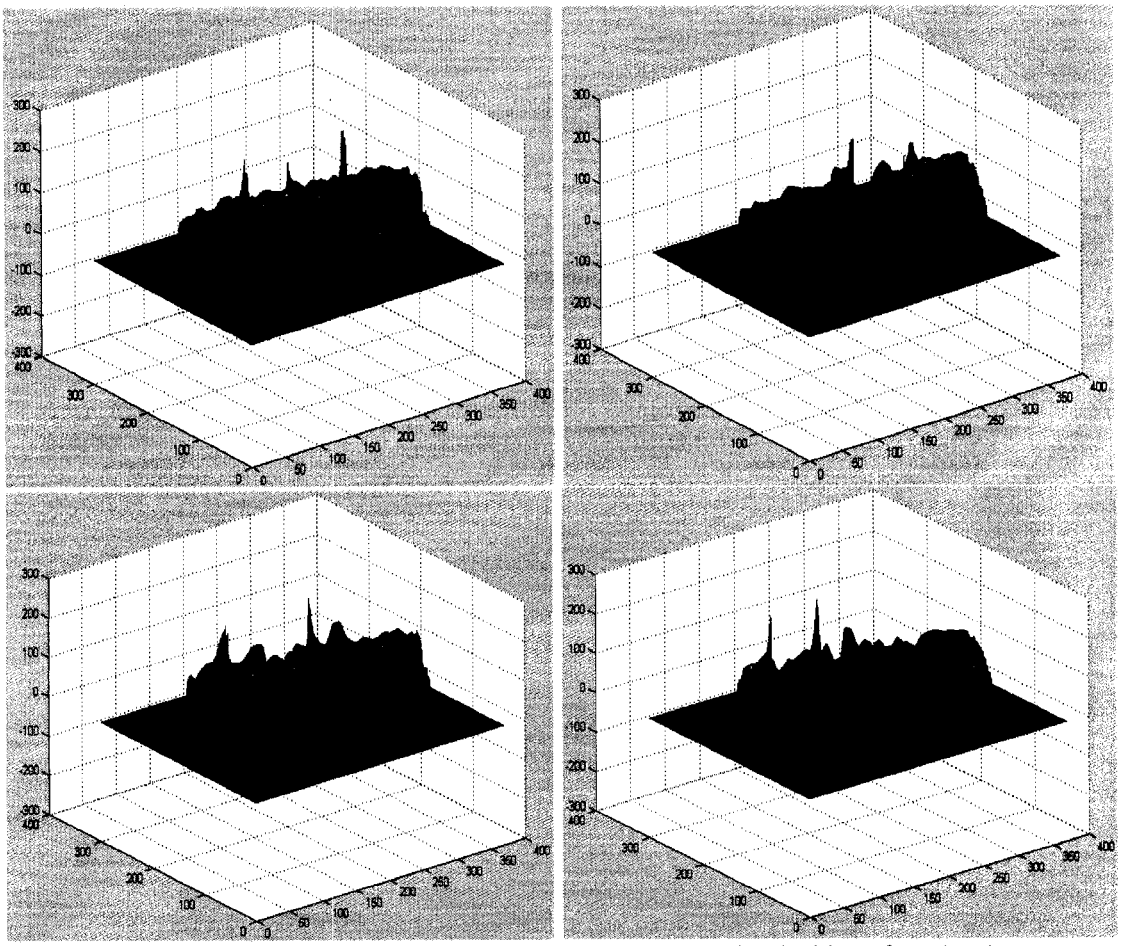


Figure 5.66 Screen shots of the 3-D rendering of the unwrapped velocities of an *in vitro* constant waveform.

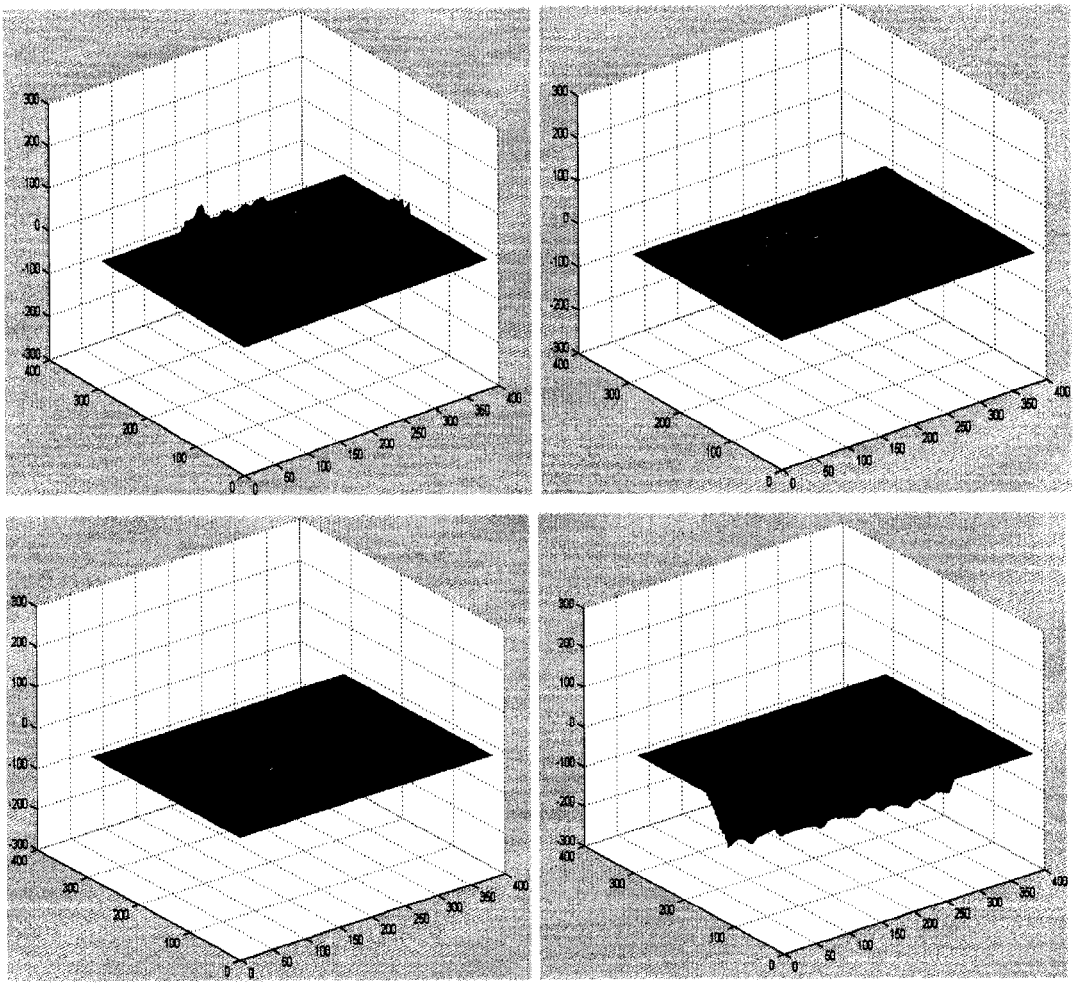


Figure 5.67 Screen shots of the 3-D rendering of the unwrapped velocities of an *in vitro* femoral waveform.

Chapter 6 Vessel Feature Generation and Selection

6.1. Introduction

In the previous chapter we discussed the newly developed phase unwrapping technique for color Doppler ultrasound images. The unwrapped images represent the undistorted raw data that maintain the spatial and temporal features of the underlying vessels. However, the data (cineloops of many color Doppler frames) is too huge in size to be used directly as an input for a vessel recognition system.

In this chapter we will discuss data reduction and feature extraction methods that prepare the color Doppler data for the recognition system, as shown in Figure 6.1. The sources of variation in ultrasound color Doppler data will be discussed in section 6.2. A data acquisition protocol using a Doppler flow phantom was developed to generate input data to the recognition system, and will be described in section 6.3. A histogram-based data reduction method was developed to retain the most-relevant information in the unwrapped color Doppler cineloops, and will be discussed in sections 6.4, 6.5, and 6.6. Data preprocessing methods were developed to achieve invariance against several transformations that represent expected variations of the imaged vessels; these methods will be described in section 6.7. Transform-based feature extraction methods were developed to reduce the dimensionality of the original data by packing the useful information to a smaller number of attributes, which will be described in section 6.8. The conclusions and future enhancements will be discussed in section 6.9.

Figure 6.1 The stages of the vessel recognition system. The feature generation and selection stage is the focus of this chapter.

6.2. Sources of Variation in Color Doppler Ultrasound Data

Feature invariance is a very important aspect of recognition problems, such as character, face, and ear recognition tasks. In each problem domain certain expected variations are defined and accounted for within the recognition system. For example, in character recognition, invariance to translation, rotation, scaling, and starting point selection are important. With ultrasound color Doppler images, it is important to identify the expected variations in the input data in order to account for these variations during the data acquisition, data preprocessing, feature extraction, and classifier design steps of the recognition system.

The following factors are identified as unique and important variations that need to be accounted for by the recognition system:

- Acquisition duration
- Acquisition starting point
- Doppler steering angle
- Mirroring
- Aliasing degree

Invariance to the acquisition duration implies that the recognition system should be independent of the duration of the color Doppler cineloop that represents the vessel. The only requirement for the acquisition duration is that it must be long enough to encompass at least one heart cycle to maintain the temporal pulsatility feature of the vessel. However, including more than one heart cycle should not affect the results of the recognition system. Also the number of cycles should not necessarily be an integer to simplify the acquisition process to be asynchronous with heart beating in order to alleviate the need for complicated heart beat-gating techniques during data acquisition. Gating acquisition techniques involve other tools, such as ECG instrument or Doppler peak velocity detection algorithms, to synchronize the acquisition with the systole phase to insure acquiring complete heart cycles.

Invariance to the acquisition starting point implies that the recognition system should be independent of where the acquisition starts within the heart cycle. Since the acquisition process is simplified to be asynchronous with heart beating, the first frame can lie anywhere within the heart cycle. This feature is similar to the starting point invariance for shape recognition problems, such as character recognition tasks.

Doppler steering angle is another feature that causes variation in the input data. Typical ultrasound systems offer a control to the user to select the optimal Doppler angle which achieves the best sensitivity of the interrogated vessel. The Doppler angle will be a function of the vessel of interest, the ultrasound transducer position and orientation, and the user preference. Changing the Doppler steering angle will change the amplitude of the depicted velocities in the color Doppler image according to the Doppler equation. We can think of the Doppler steering angle as a nonlinear scale change of the Doppler velocities. The nonlinearity comes from the fact that the human vessels can have some tortuosity, which causes the effective Doppler angle to be different for different sections of the vessel. Thus changing the steering angle will have different effects on different sections of the vessel.

Mirroring of the color Doppler data implies that the velocities in color Doppler images will be inverted in sign, i.e. positive velocities will be negative and vice versa. Mirroring is another source of variation in the input data. Typically the velocities moving toward the transducer are displayed in red, while the velocities moving away from transducer are displayed in blue. The majority of ultrasound operators prefer to display arteries in red while displaying veins in blue. For that reason most ultrasound systems offer a control to the user to invert the displayed velocities. Some shape recognition systems, such as character recognition, have addressed the mirroring problem.

The last source of variation in the input data is the degree of aliasing, which distorts the velocities in the color Doppler image. The last two chapters were dedicated to the aliasing problem, and a new unwrapping algorithm was developed to recover the true velocities from the aliased ones. The aliasing problem cannot always be avoided due to user preference or physics limitations. Thus the recognition system needs to account for it.

All the previously mentioned sources of variation will be accounted for during the data acquisition and feature extraction steps, which will be discussed in the following sections.

6.3. Data Acquisition Protocol

For the purpose of the vessel recognition task, the following criteria were considered for the data acquisition phase:

- To have access to different vessels with different waveforms that represent human vessels those are typically scanned with ultrasound.
- To have control over some attributes of the generated data such as peak velocity and pulsatility duration so that controlled variations can be reproduced to test the recognition system.
- To be able to generate a large number of cases for training and test purposes.

For all the previously mentioned reasons, a Doppler flow phantom was used for the data acquisition task for the recognition system. Five different waveforms that are typically found in human vessels are used as a case study for the vessel recognition task. These waveforms are: carotid, constant, femoral, sinusoidal, and square. Figure 6.2 shows examples of each waveform in duplex color Doppler mode. The top half of each image shows the composite live grayscale and color Doppler image, while the bottom half shows the scrolling spectral PW Doppler display of many cycles. The spectral Doppler data is acquired only from the position of the Doppler sample volume gate indicated by the line graphics on the color image.

The spectral Doppler display reveals the commonalities and differences of the different waveforms. For example, both the carotid and square waveforms are uniphasic; the velocities are either all positive or all negative, indicating that the blood is moving in one direction during the whole cycle. On the other hand the femoral waveform is triphasic where the blood changes direction twice during one cycle (the velocities change sign twice), while the sinusoidal waveform is biphasic where the blood changes direction once during one cycle

(the velocities change sign once). Another observation is that the constant waveform is not pulsatile, which mimics some real veins, while the other four waveforms are pulsatile.

During the initial data acquisition experiment, which was meant to provide training data for the feature extraction and recognition system, the peak velocity was held constant for each waveform. However the different waveforms had different peak velocities. The velocity ranged from 30 cm/s to 230 cm/s, as measured on the spectral Doppler display. The acquisition duration was maximized throughout the whole acquisition process. The number of acquired frames was set to 60 frames. However, due to memory constraints, the ultrasound system was not always able to capture the maximum number of frames. The goal behind maximizing the acquisition duration is to allow flexibility to experiment with different acquisition durations during the feature extraction and classification phases.

Two system settings are allowed to change during the data acquisition process: the color Doppler sampling rate or PRF and the Doppler steering angle. The PRF was allowed to sweep through a large range to generate different degrees of aliasing, starting with high sampling rates that generate unaliased datasets with low blood filling. The PRF would then be lowered incrementally to generate unaliased datasets with good blood filling, moderately aliased datasets, and severely-aliased datasets. Figure 6.3 shows examples of some carotid waveform color Doppler datasets with different sampling rates (indicated as a Nyquist velocity value which correlates with the PRF setting) and the same Doppler steering angle. It is evident from the figure that the aliasing degree increases as the PRF decreases. Also the color Doppler filling enhances as the PRF decreases; for example there are many frames with only a few color Doppler pixels in Figure 6.3a due to the higher PRF used.

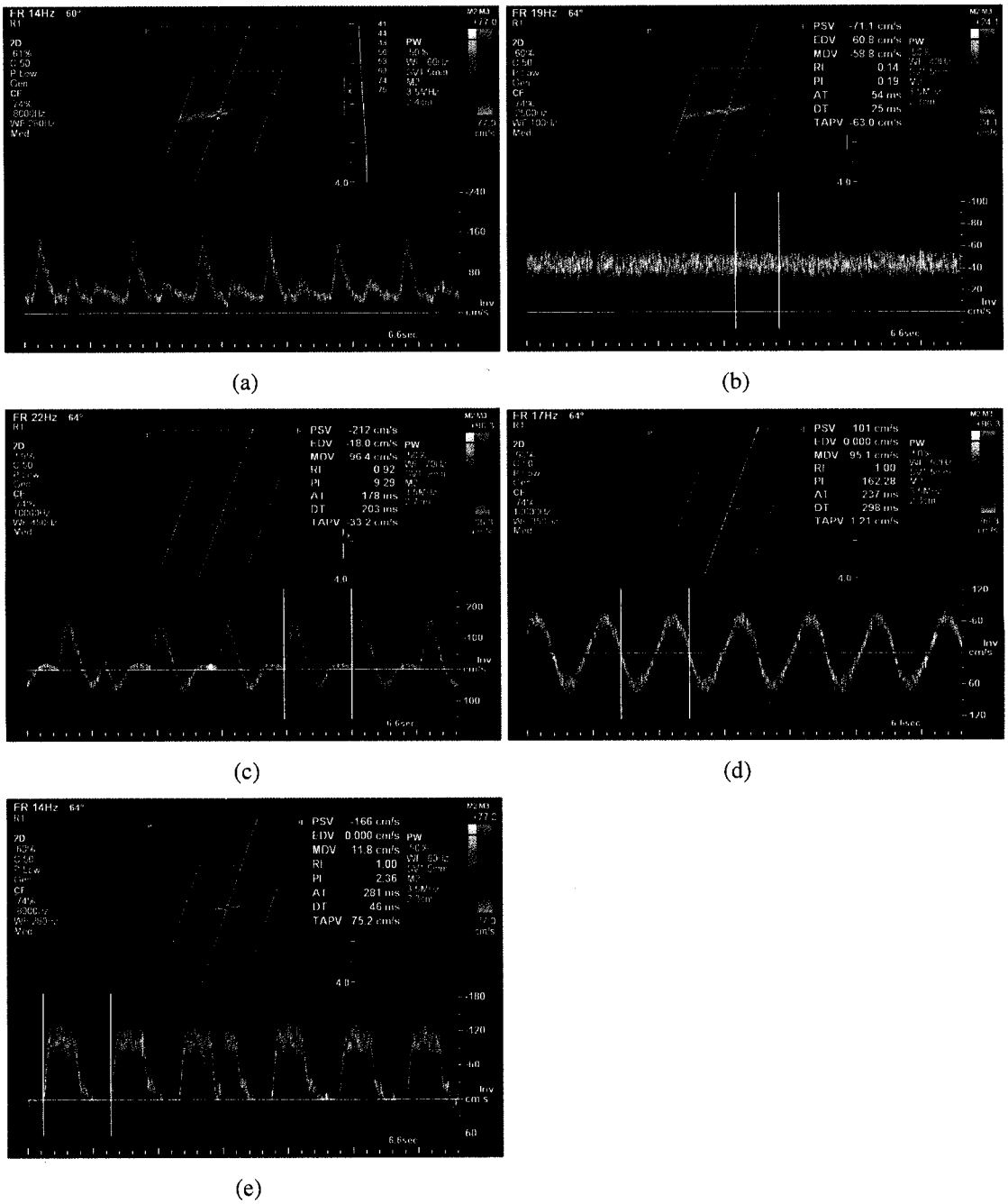


Figure 6.2 Spectral Doppler display for different Doppler phantom waveforms. (a) Carotid waveform. (b) Constant waveform. (c) Femoral waveform. (d) Sinusoidal waveform. (e) Square waveform.

For each PRF, the Doppler steering angle was allowed to sweep its full range in discrete increments, covering 20°, 16°, 12°, 8°, 4°, and 0° angles. The effect of lowering the steering angle is to lower the projected velocities on the color Doppler image. Figure 6.4 shows examples of some carotid waveform color Doppler datasets (less than 60 frames are shown) with different Doppler steering angles and the same PRF. It can be seen from the figure that the velocity values decreases as the Doppler steering angle decreases, since the Doppler beams become more and more perpendicular to the vessel. The total number of datasets for the training purpose was equal to 150 datasets for 5 waveforms, 5 PRF settings, and 6 steering angles. Each dataset contains 60 color Doppler frames, which adds up to 9000 images in total.

6.4. The Color Doppler Virtual Spectrogram (VS)

The data that was captured in the previous section represents the raw data which contains all the information about the underlying vessel. However some datasets are distorted by varying degrees of aliasing. The logical next step was to pass all datasets by the newly developed phase unwrapping algorithm that was described in the previous chapter. The resultant unwrapped datasets represent undistorted representation of the underlying vessels. Figure 6.5 and Figure 6.6 show the unwrapping results of the corresponding datasets shown previously in Figure 6.3 and Figure 6.4. Most of the unwrapped frames are correct except for a few frames that show some unwrapping errors, as indicated by the reverse flow direction regions (blue color blobs). This contradicts the fact that the carotid flow is unilateral and the flow stays in one direction during the whole heart cycle. Such errors will need to be accounted for during the feature extraction and recognition system design phases.

Two approaches were considered for feature extraction from color Doppler data. The first approach is the direct extraction of robust and discriminating spatial and temporal features from the unwrapped color Doppler datasets. Examples of such features are the peak velocity, the pulsatility measure, and the flow phasicity. In the previous chapter, we developed a robust estimator of the peak velocity and used it as a quantitative tool to assess the results of

our phase unwrapping technique. We could develop similar approaches for estimating other important features directly from the unwrapped color Doppler cineloops.

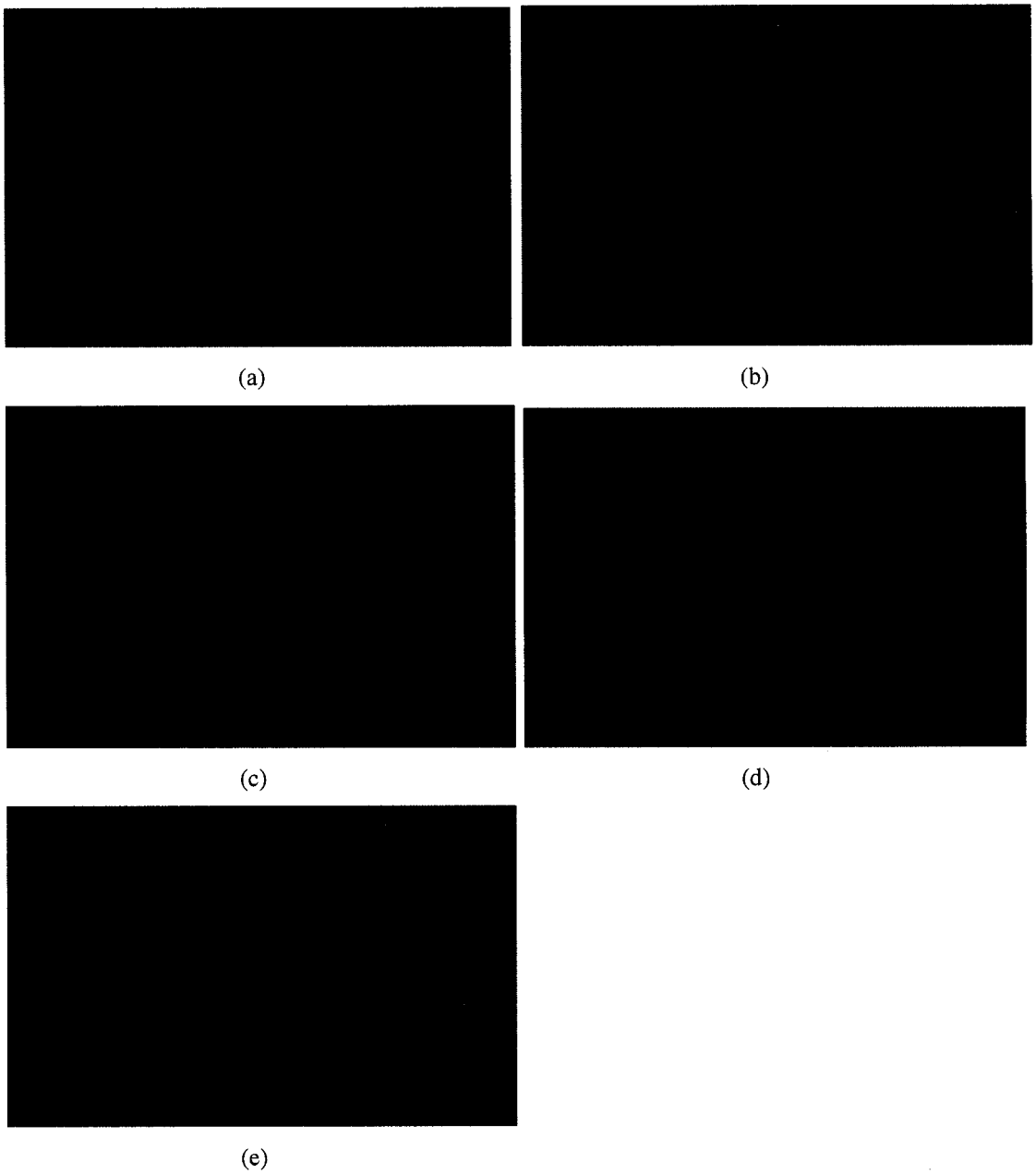


Figure 6.3 Several color Doppler datasets for the carotid waveform with different sampling rates and the same Doppler steering angle. The aliasing degree increases as the sampling rate decreases. (a) Nyquist velocity= 86.6 cm/s. (b) Nyquist velocity= 67.4 cm/s. (c) Nyquist velocity= 48.1 cm/s. (d) Nyquist velocity= 38.5 cm/s. (e) Nyquist velocity= 33.7 cm/s.

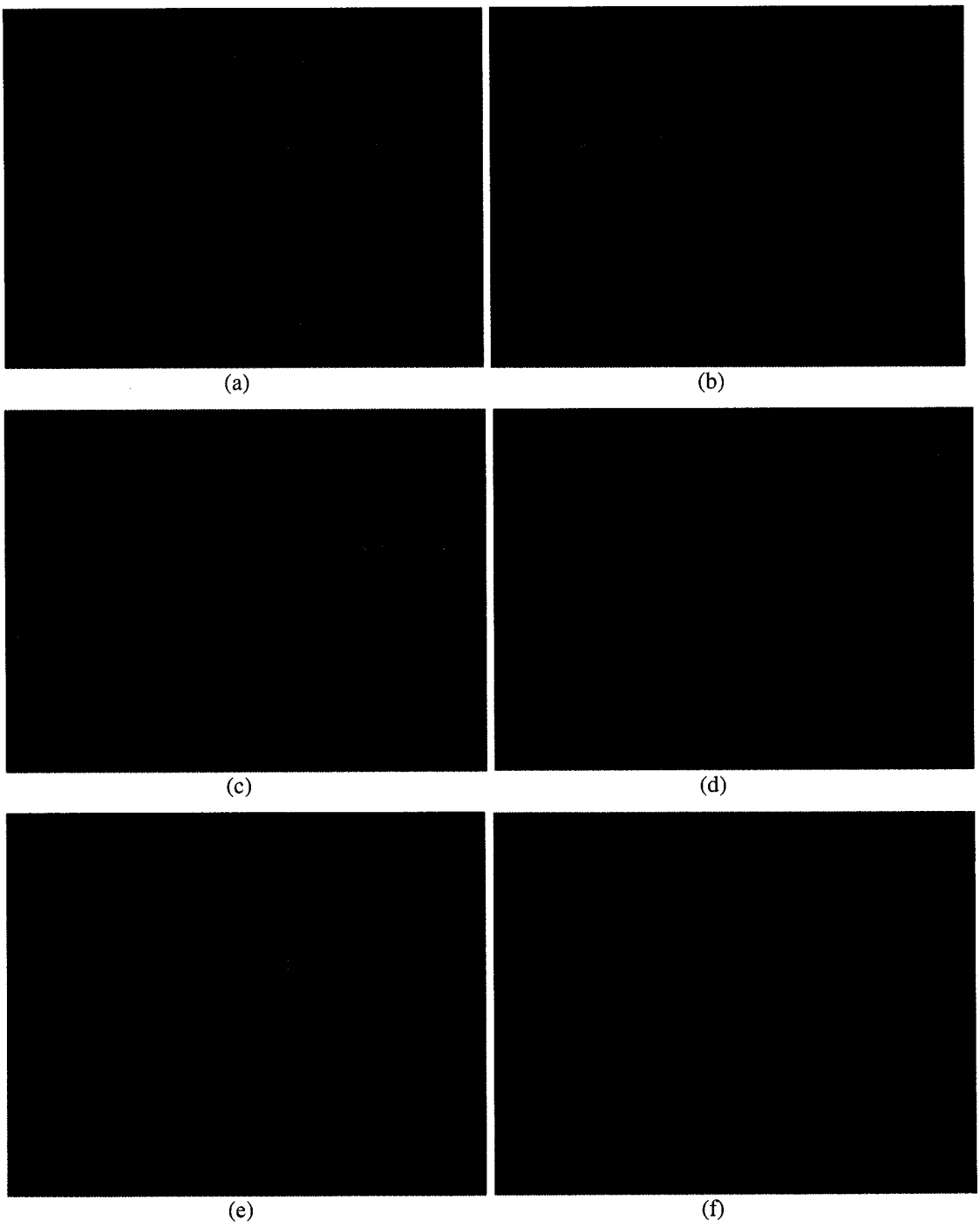


Figure 6.4 Several color Doppler datasets for the carotid waveform with different Doppler steering angles and the same PRF. The velocities decrease as the Doppler steering angle decreases. (a) Steering angle= 20°. (b) Steering angle= 16°. (c) Steering angle= 12°. (d) Steering angle= 8°. (e) Steering angle= 4°. (f) Steering angle= 0°.

The advantage of this approach is the correspondence between those features and the visual clues used by human experts to distinguish vessels. The other related advantage is that these features represent clinically relevant parameters that are used by ultrasound physicians to interpret the vessel hemodynamics and pathology states. However, physicians typically rely on the spectral Doppler data for this assessment. A nice extension would be to recognize different pathology states of the same vessel by correlating the estimated features with the published pathology numbers. The disadvantage of this approach is the reliance on a small number of features that are extracted from the color Doppler data. Some vessels may have similar values for these features and hence the discriminating task would rely on a very few parameters.

The second approach for feature extraction is to reduce the dimensionality of the unwrapped color Doppler data while maintaining the temporal pulsatility signature of the underlying vessel, which represents the most powerful discriminating feature of the blood flow within human vessels. Then this signature would be used directly, or through linear transformations, as an input to the recognition system without attempting to extract discrete features from the data. One advantage of such an approach is to retain the most useful information inherent in the color Doppler cineloops while reducing the data dimensionality via removing redundancy from the data. A second advantage is to avoid the development of robust feature estimators that would be needed to handle degraded data due to segmentation and phase unwrapping errors discussed in the previous chapters.

The second approach for feature extraction was adopted in this work. The data reduction goal was achieved in two steps. In the first step a histogram-based technique was developed to project the three-dimensional input data (2-D color Doppler frames in time) into a two-dimensional histogram image of the velocity distribution in time. We called the output 2-D histogram image the "*virtual spectrogram*" or VS due to its resemblance to the spectral Doppler display mode of the ultrasound systems, as shown previously in Figure 6.2.

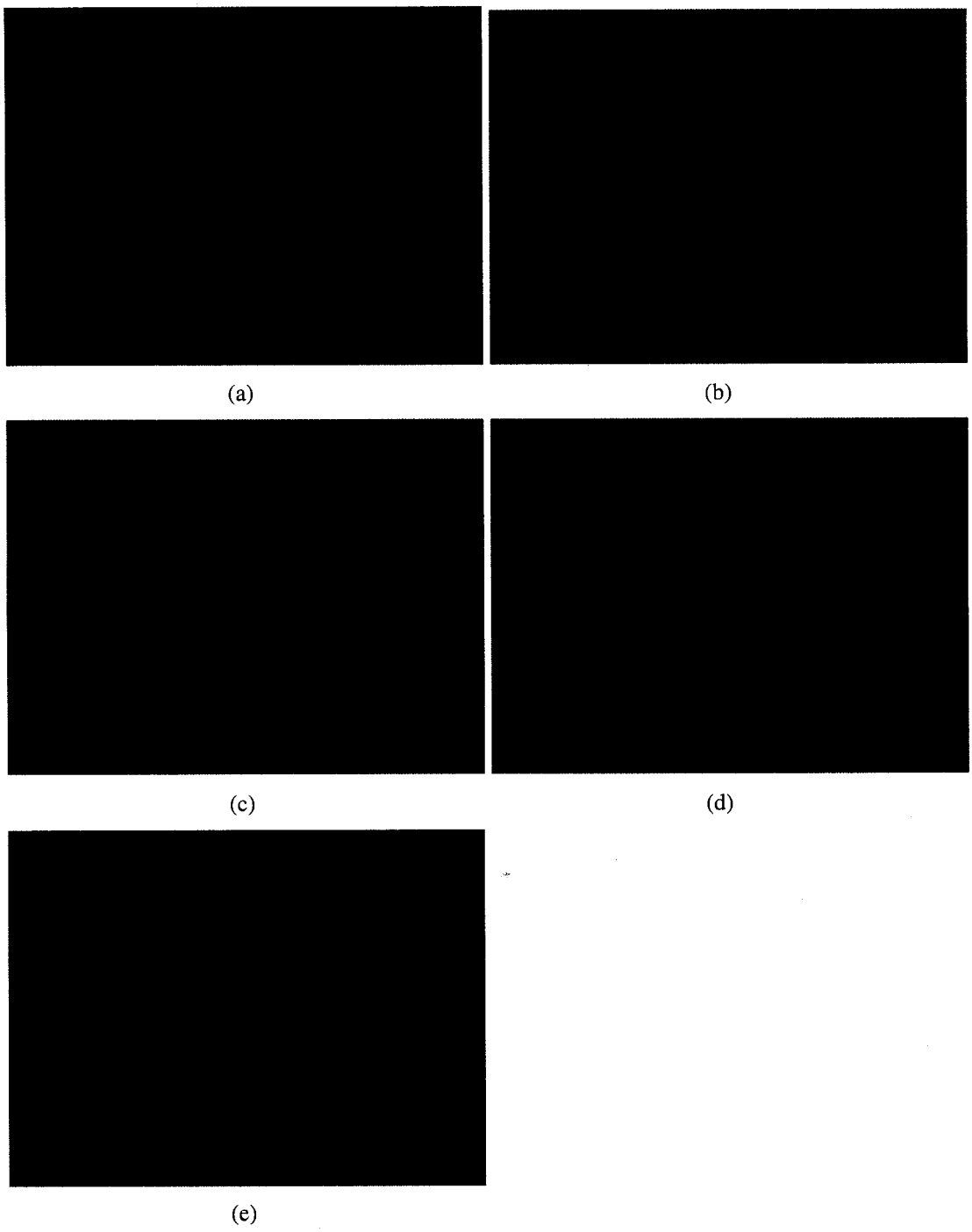


Figure 6.5 Unwrapped color Doppler datasets for the carotid waveform with different sampling rates and the same Doppler steering angle. (a) Nyquist velocity= 86.6 cm/s. (b) Nyquist velocity= 67.4 cm/s. (c) Nyquist velocity= 48.1 cm/s. (d) Nyquist velocity= 38.5 cm/s. (e) Nyquist velocity= 33.7 cm/s.

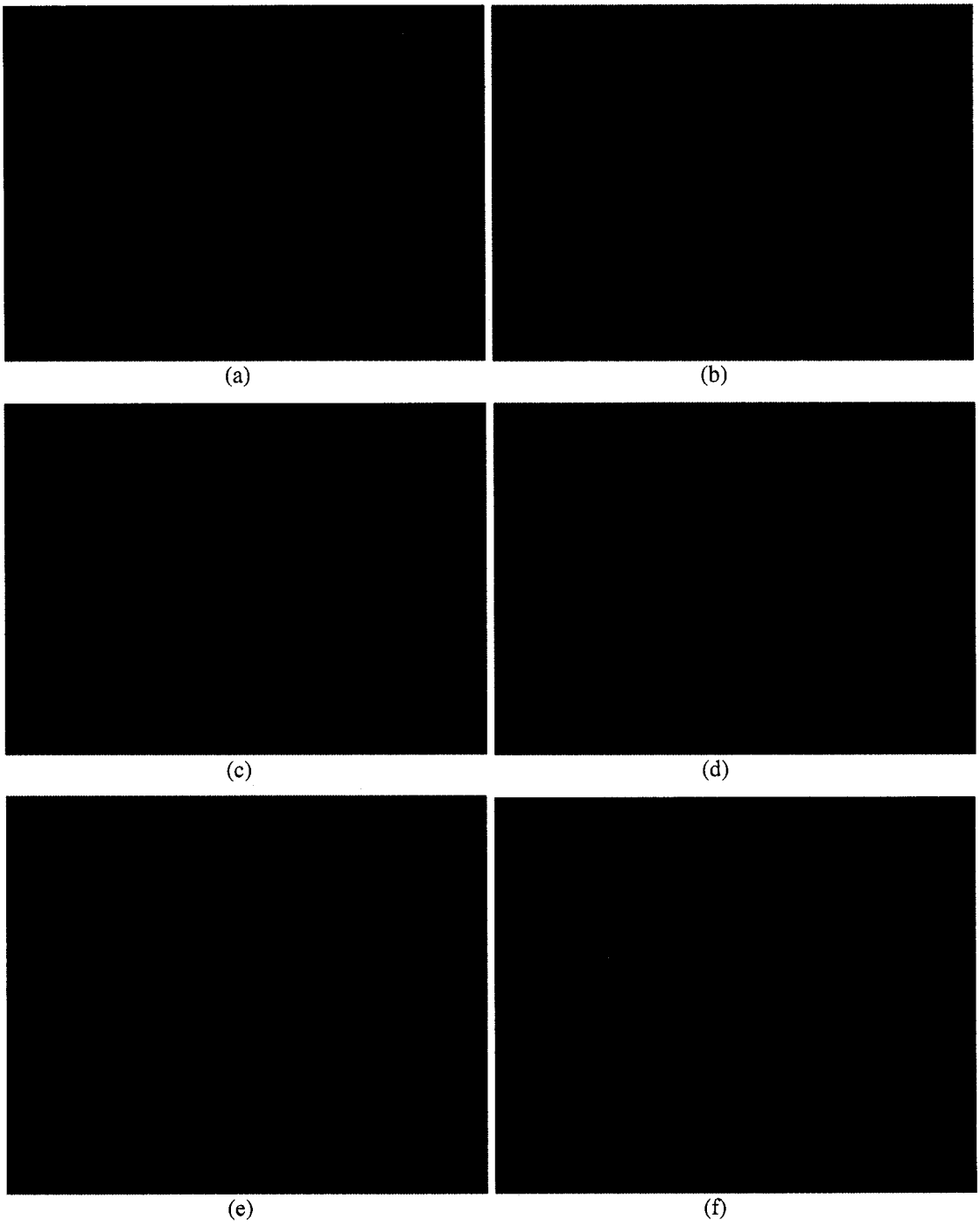


Figure 6.6 Unwrapped color Doppler datasets for the carotid waveform with different Doppler steering angles and the same PRF. (a) Steering angle= 20°. (b) Steering angle= 16°. (c) Steering angle= 12°. (d) Steering angle= 8°. (e) Steering angle= 4°. (f) Steering angle= 0°.

To construct the virtual spectrogram, a 2-D histogram of the velocity distribution is constructed from the unwrapped velocity data. The x-axis represents the frame number, while the y-axis represents counts of the discretized values of the velocities. The color Doppler pixels in each frame of the unwrapped dataset are converted to velocities using the scaling information associated with the captured frames. The velocities are then discretized and the corresponding vertical slot is incremented. The 2-D histogram is binarized by assigning a non-zero value to any velocity slot that has a non-zero count.

Figure 6.7 shows an example of the virtual spectrogram of a carotid waveform. It is clear from the figure that the virtual spectrogram maintains the signature of the carotid flow where the pulsatility is evident and the flow is uniphasic. It is important to note that all the velocities in the acquired training data were inverted, and this is why the carotid velocities are displayed as negative ones. This is just an aesthetics issue which should not affect the performance of the recognition system.

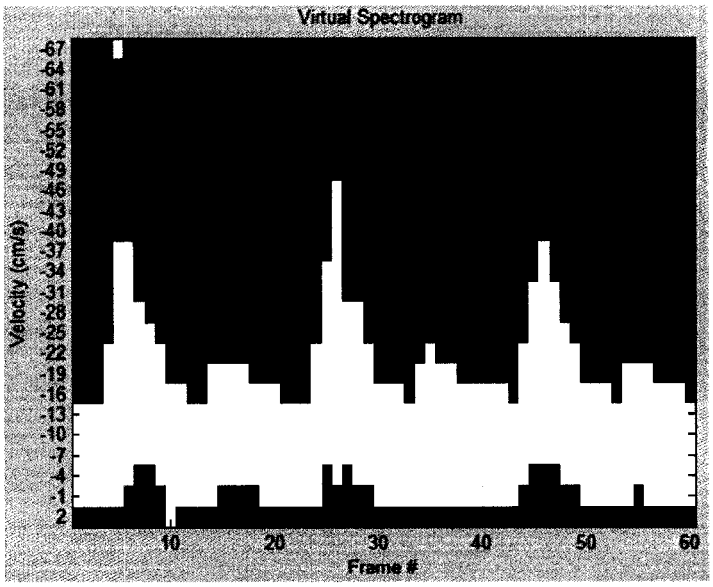


Figure 6.7 Color Doppler virtual spectrogram of a carotid waveform. The x-axis represents frame number. The y-axis represents discretized velocity. White slots indicate non-zero counts of the corresponding velocity within the corresponding frame.

Figure 6.8 and Figure 6.9 show the virtual spectrograms of the carotid datasets shown in Figure 6.3 and Figure 6.4 respectively. By carefully studying the carotid virtual spectrograms we can conclude that the virtual spectrograms offer good representations of the vessel hemodynamic while greatly reducing the data dimensionality. The pulsatiliy, phasicity, and peak velocity of the carotid artery are maintained in the carotid virtual spectrograms. There is a great resemblance between the virtual spectrograms and the true spectral spectrograms shown previously in Figure 6.2.

The quality of the VS is not as good as the spectral spectrograms; this is expected due to the inherent differences between the spectral Doppler and color Doppler acquisition processes, as described previously in Chapter 2. The degradation is also due to the limitations inherent in the phase unwrapping process, which may introduce inaccurate or wrong velocities in the unwrapped data.

The errors introduced by the phase unwrapping algorithms are clearly visible on the VS, for example negative velocities, which should not exist in the uniphasic carotid waveform, are occasionally shown. The peak velocity seems to be inaccurate in some cases as well. The binarization of the velocity histogram contributed to this inaccuracy, since even noise pixels contribute to the VS. All these factors strengthen the argument to rely on the overall appearance of the flow instead of trying to derive discrete features that will be prone to estimation errors.

Another important observation about the carotid VSs is that the waveforms show a great deal of variability although they share a common appearance. This is very similar to the hand-written character recognition problem where the letter patterns can change from person to person, which complicates the recognition task. Both the feature extraction and the recognition system tasks will need to account for such great variability.

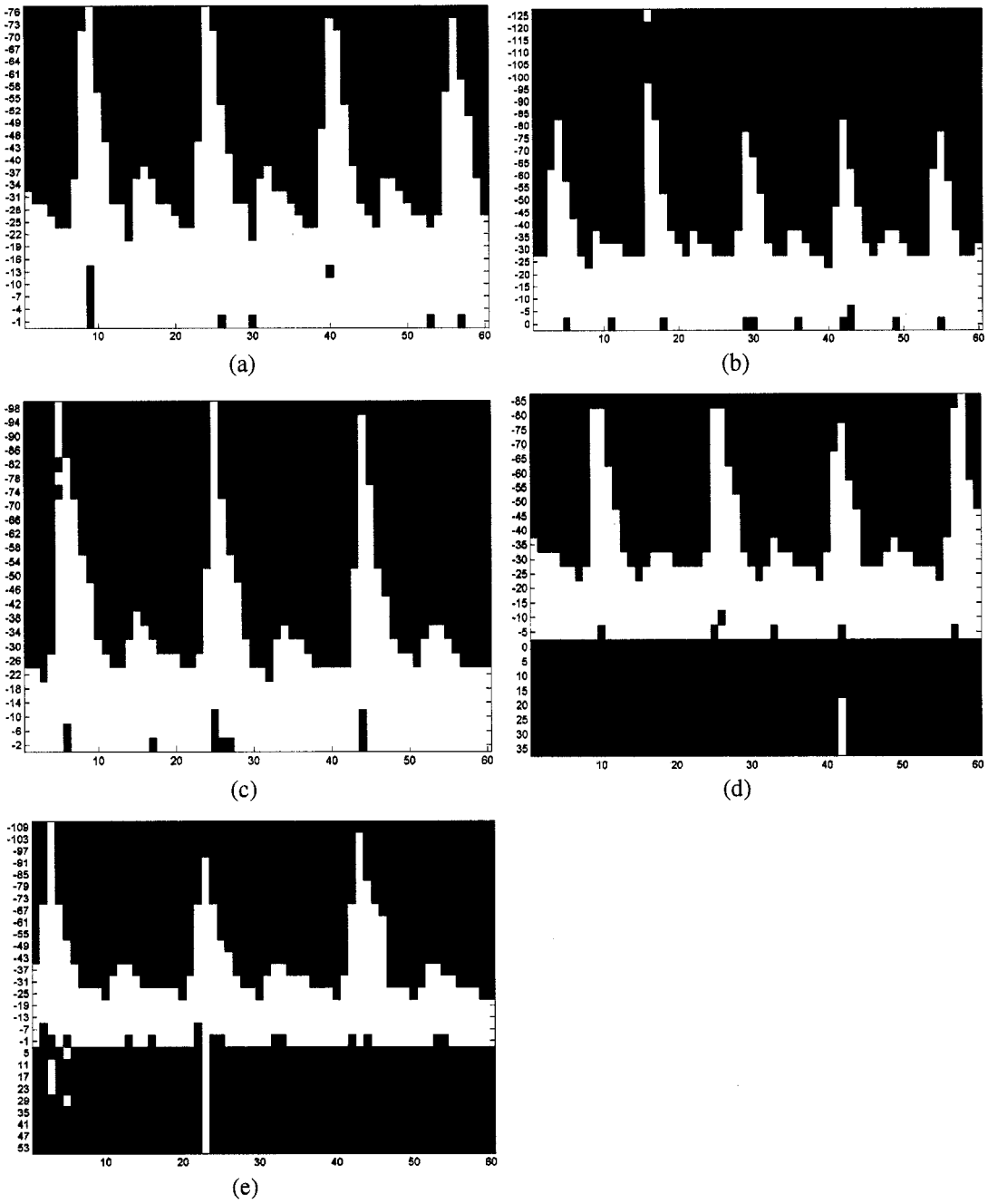


Figure 6.8 Virtual spectrograms for the carotid waveform with different sampling rates and same Doppler steering angle. (a) Nyquist velocity= 86.6 cm/s. (b) Nyquist velocity= 67.4 cm/s. (c) Nyquist velocity= 48.1 cm/s. (d) Nyquist velocity= 38.5 cm/s. (e) Nyquist velocity= 33.7 cm/s.

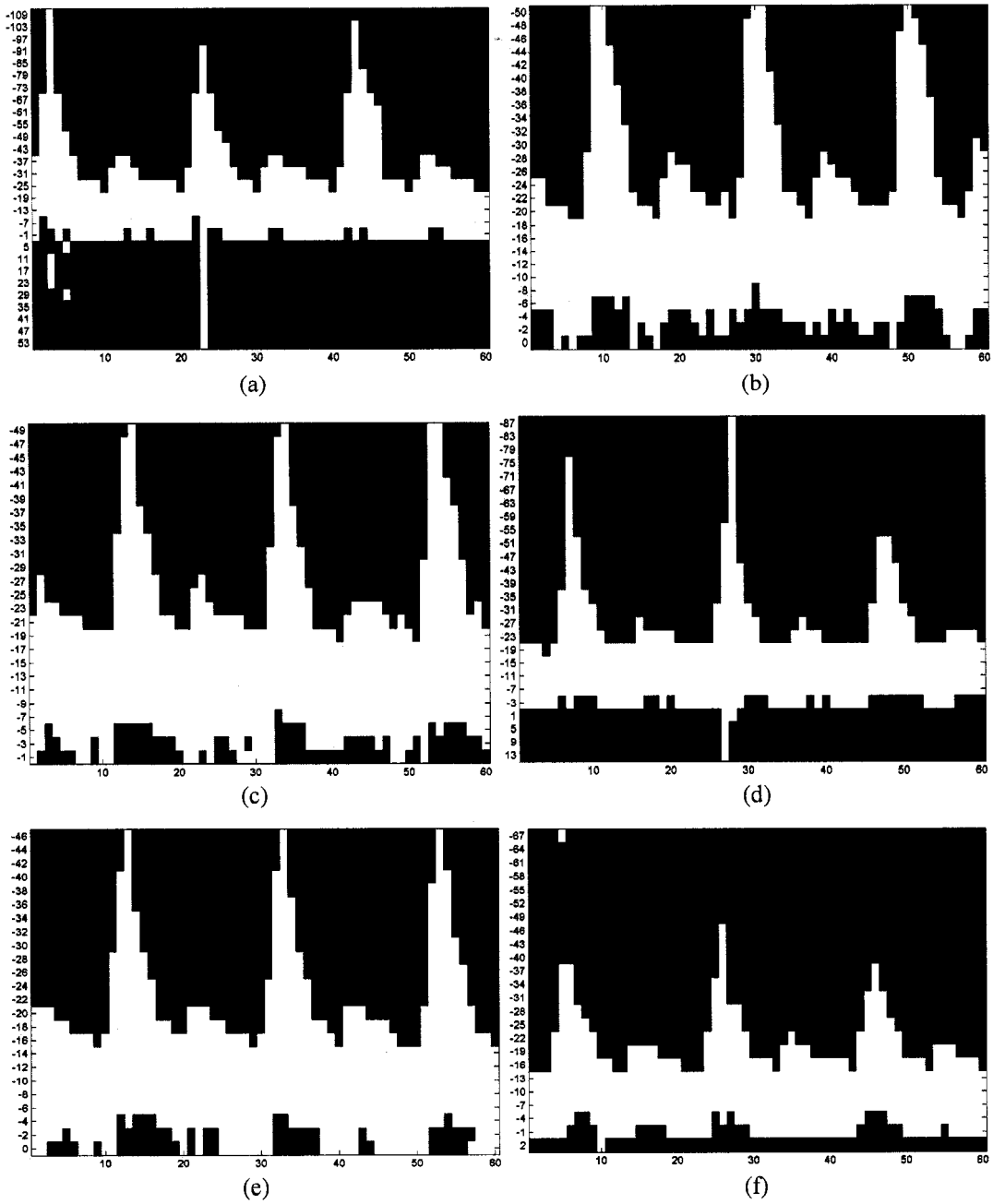


Figure 6.9 Virtual spectrograms for the carotid waveform with different Doppler steering angles and same PRF. (a) Steering angle= 20°. (b) Steering angle= 16°. (c) Steering angle= 12°. (d) Steering angle= 8°. (e) Steering angle= 4°. (f) Steering angle= 0°.

6.5. The Virtual Spectrogram Profile (VSP)

The virtual spectrogram developed in the previous section represents an efficient data reduction tool, which still maintains the main characteristics of the underlying vessel flow. VSs can be directly used as inputs to a recognition system or 2-D features can be extracted from them to be fed to the recognition system. 2-D texture features as well as transform features (Fourier, Wavelets, and Gabor) can be applied on the VS images and extract useful attributes that can be used as discriminating features. However carefully studying the VSs reveals that the relevant data that discriminate between vessels is confined to the external boundary of the VS. The boundary carries information about the peak velocity, pulsatility, and phasicity. So if we are able to trace the boundary of the VS, this will represent even further data reduction without sacrificing relevant information. For that reason, we developed a simple boundary tracking algorithm and applied it on the VS images to extract the boundary, which we called the "*virtual spectrogram profile*" or VSP.

In the VS boundary extraction, each column of the VS image is considered. The max and min velocities are detected as the highest and lowest non-zero pixels in the column. The absolute maximum and minimum velocities are compared and the highest is included in the profile. Due to the inaccuracy of the data, a 3-point median filter is applied to the original profile to remove the spikes that may occur. Figure 6.10 shows both the raw and median-filtered profiles of the carotid VS of Figure 6.7. The median-filtered profile removed the spikes from the raw profile, but at the same time the peak velocity is underestimated. This should not be a problem for the recognition system; it does not rely on the absolute velocity values due to the data inaccuracy.

Figure 6.11 and Figure 6.12 show the virtual spectrogram profiles, overlaid on top of the VS images, for the carotid waveforms of Figure 6.3 and Figure 6.4 respectively. To show examples of the VSPs for other waveforms, Figure 6.13 shows different constant waveforms, Figure 6.14 shows different femoral waveforms, Figure 6.15 shows different sinusoidal waveforms, and Figure 6.16 shows different square waveforms. All the previously mentioned observations apply to the other waveforms as well. In summary, the virtual spectrogram

profiles are compact representations of the underlying waveforms; that greatly reducing the data dimensionality.

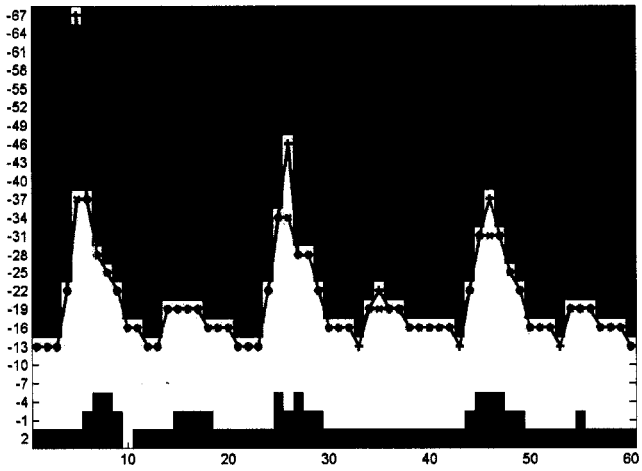


Figure 6.10 Virtual spectrogram profile (VSP) of a carotid waveform overlaid on top of its VS. The raw VSP is shown in blue. The median-filtered VSP is shown in red.

6.6. Waveform Pitch Detection

The virtual spectrogram profiles detected in the previous section could contain an unspecified number of heart cycles, which will make it difficult to develop similarity measures between profiles of the same waveform. Also any transform-based feature extractor, such as the Fourier transform, is defined for aperiodic signals that represent one cycle of periodic signals with infinite period. The other alternative, the Fourier series expansion, is defined for infinite periodic signals. Our VSPs represent finite periodic signals, which are not suitable for either form of the transform-based methods.

For these reasons, we decided to apply a detection algorithm to detect a single period of the vessel waveform. In the speech processing and recognition literature, the fundamental period of a periodic signal is called the signal "pitch." We will adopt the same name here and call it the color Doppler waveform pitch. Pitch detection is a classical and still-active research

problem in speech signal processing literature. Pitch detection algorithms can be classified as time domain algorithms, frequency domain algorithms, and hybrid algorithms [82].

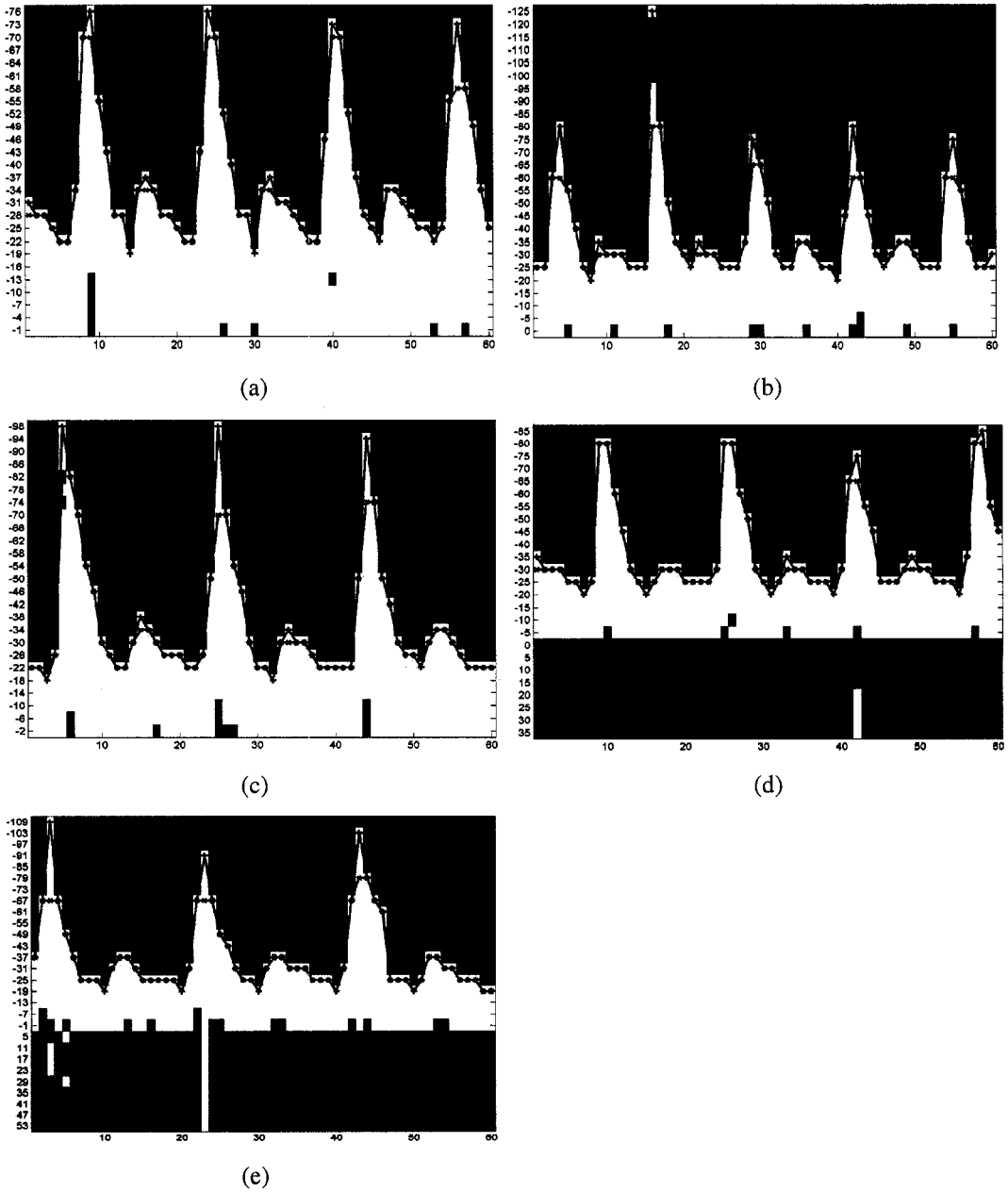


Figure 6.11 Virtual spectrogram profiles (VSPs) for the carotid waveform with different sampling rates and same Doppler steering angle. Raw profiles are in blue and final profiles are in red. (a) Nyquist velocity= 86.6 cm/s. (b) Nyquist velocity= 67.4 cm/s. (c) Nyquist velocity= 48.1 cm/s. (d) Nyquist velocity= 38.5 cm/s. (e) Nyquist velocity= 33.7 cm/s.

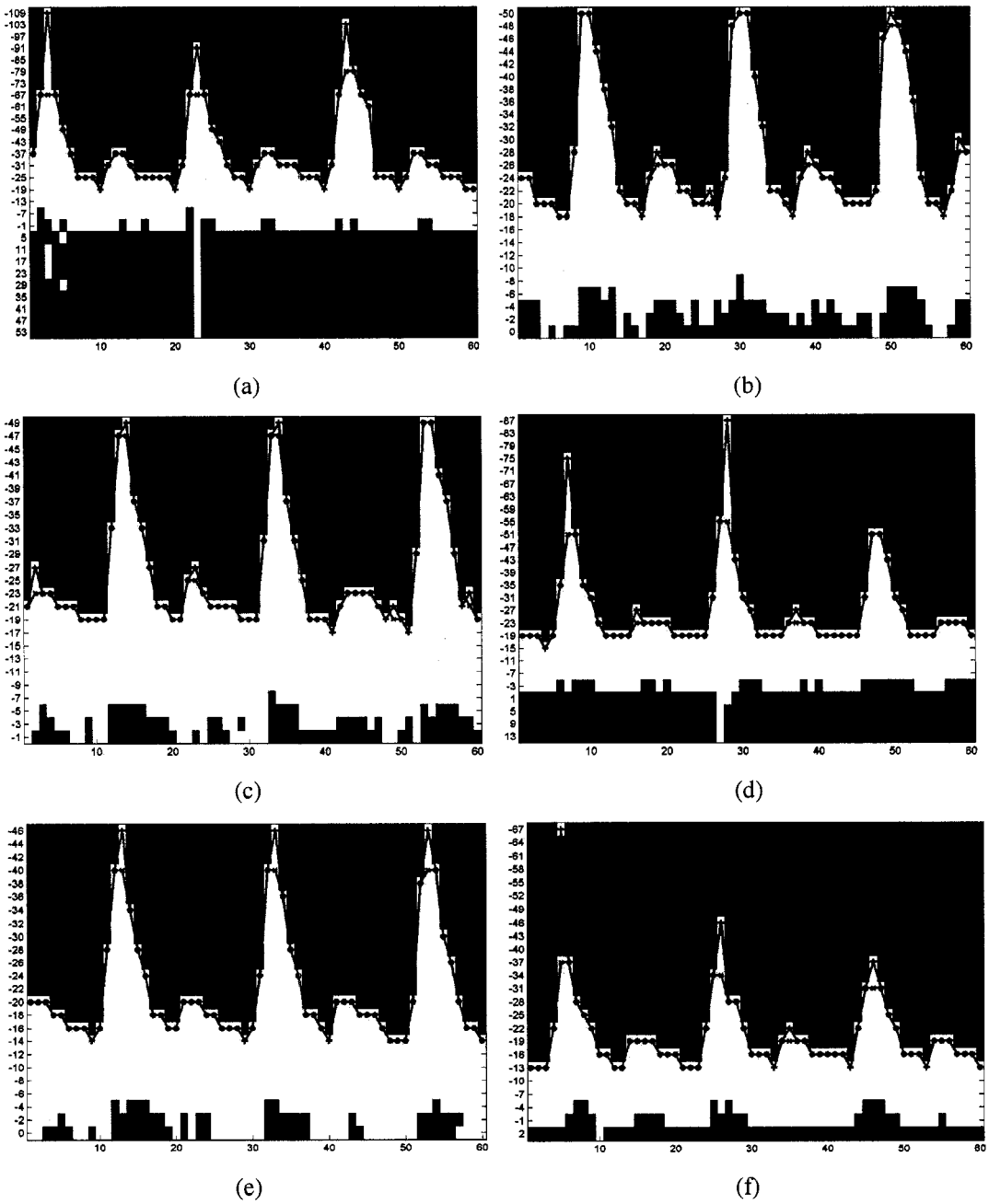


Figure 6.12 Virtual spectrogram profiles (VSPs) for the carotid waveform with different Doppler steering angles and same PRF. Raw profiles are in blue and final profiles are in red. (a) Steering angle= 20°. (b) Steering angle= 16°. (c) Steering angle= 12°. (d) Steering angle= 8°. (e) Steering angle= 4°. (f) Steering angle= 0°.

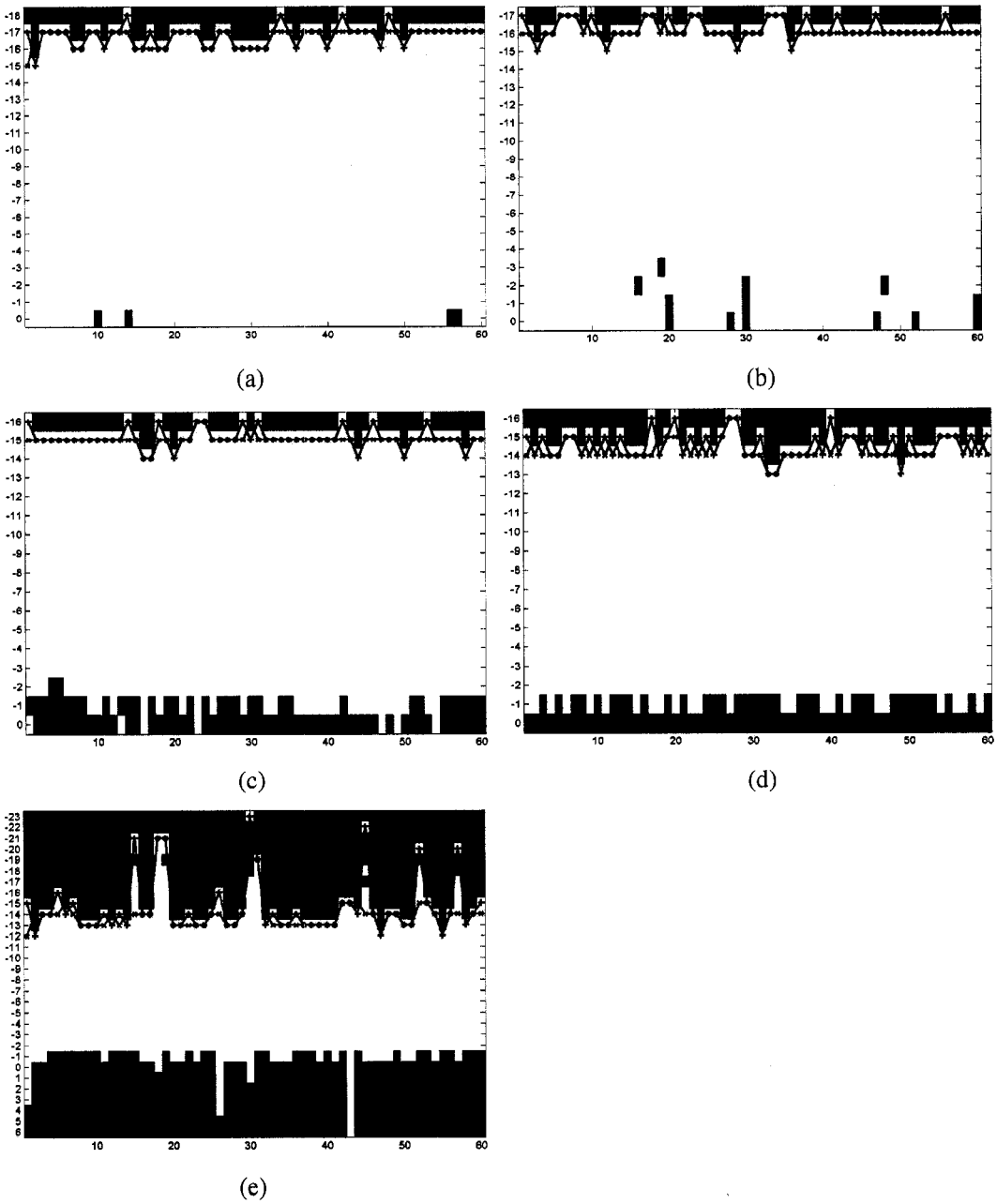


Figure 6.13 Virtual spectrogram profiles (VSPs) for the constant waveform with different sampling rates and same Doppler steering angle. Raw profiles are in blue and final profiles are in red. (a) Nyquist velocity= 43.3 cm/s. (b) Nyquist velocity= 33.7 cm/s. (c) Nyquist velocity= 24.1 cm/s. (d) Nyquist velocity= 14.4 cm/s. (e) Nyquist velocity= 9.6 cm/s.

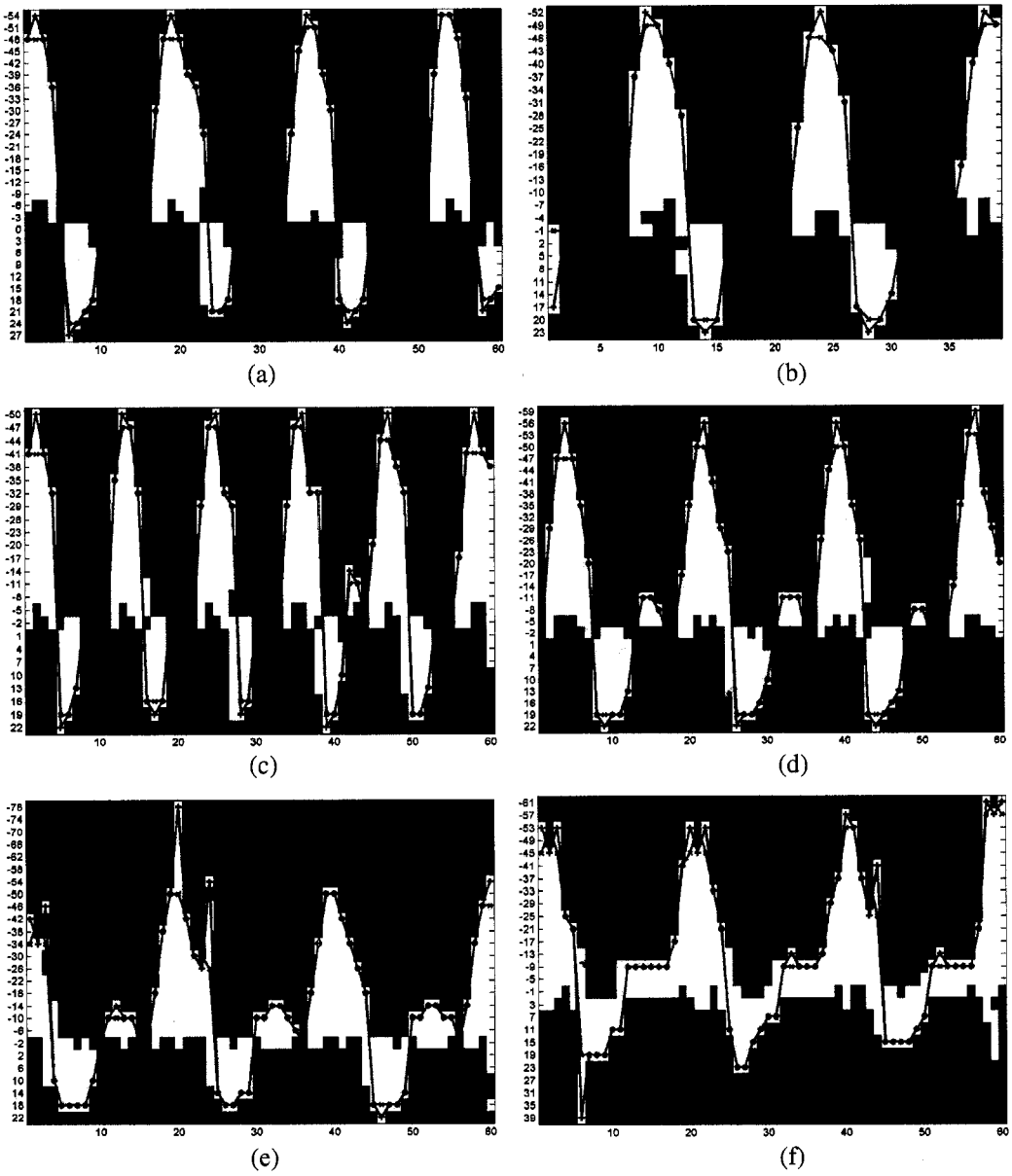


Figure 6.14 Virtual spectrogram profiles (VSPs) for the femoral waveform with different sampling rates and same Doppler steering angle. Raw profiles are in blue and final profiles are in red. (a) Nyquist velocity= 96.3 cm/s. (b) Nyquist velocity= 77.0 cm/s. (c) Nyquist velocity= 57.8 cm/s. (d) Nyquist velocity= 43.3 cm/s. (e) Nyquist velocity= 33.7 cm/s. (f) Nyquist velocity= 24.1 cm/s.

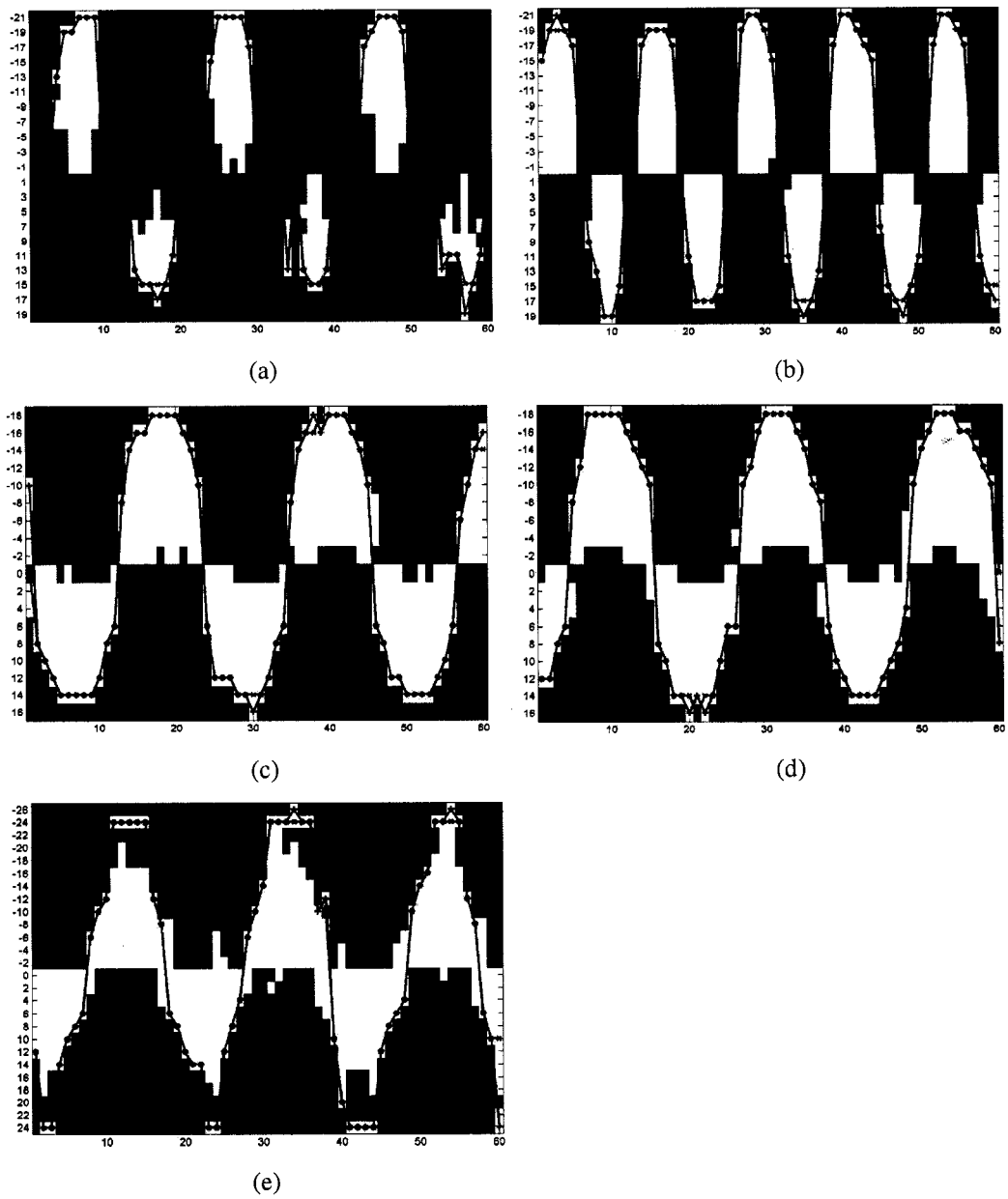


Figure 6.15 Virtual spectrogram profiles (VSPs) for the sinusoidal waveform with different sampling rates and same Doppler steering angle. Raw profiles are in blue and final profiles are in red. (a) Nyquist velocity= 96.3 cm/s. (b) Nyquist velocity= 67.4 cm/s. (c) Nyquist velocity= 28.9 cm/s. (d) Nyquist velocity= 19.3 cm/s. (e) Nyquist velocity= 12 cm/s.

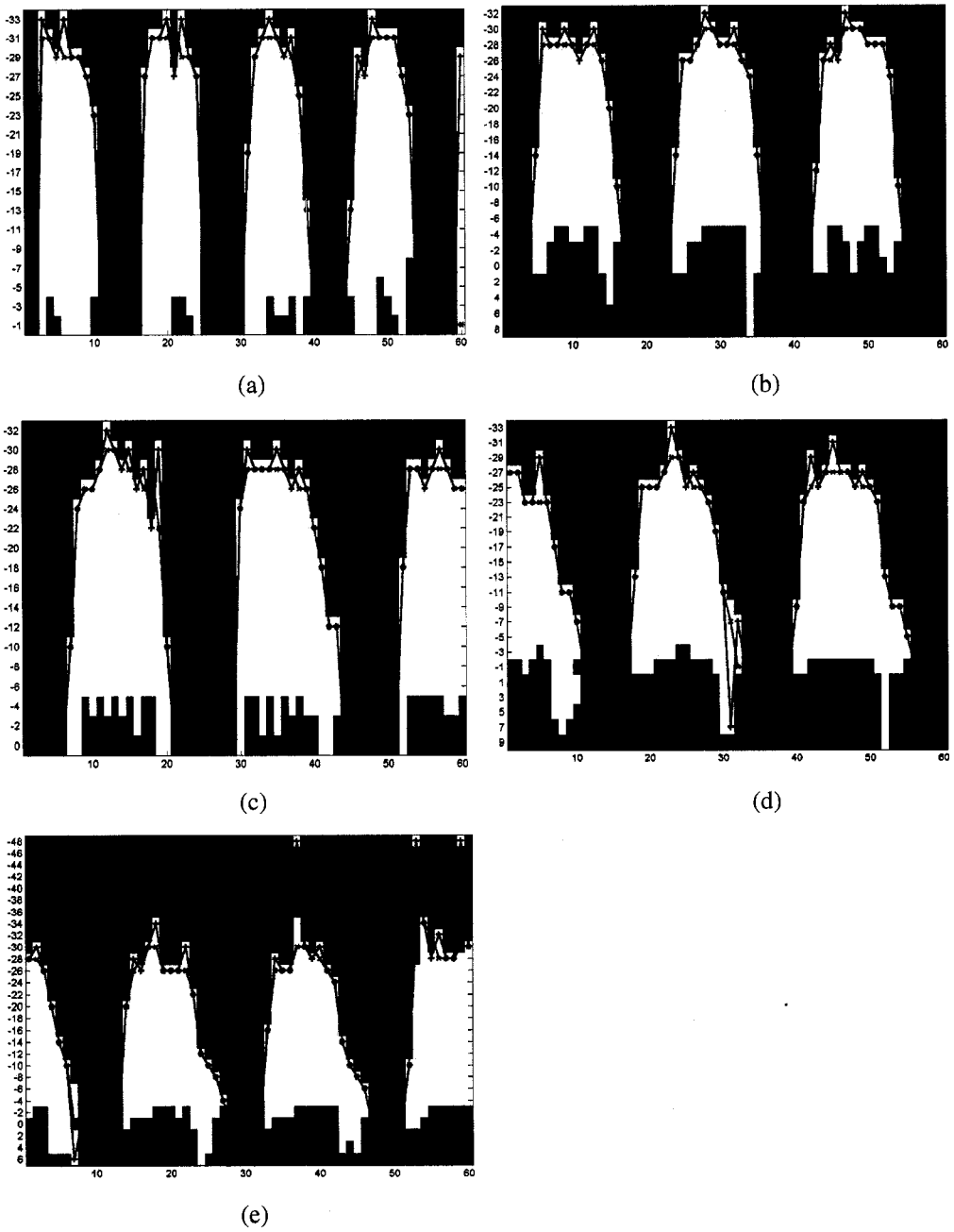


Figure 6.16 Virtual spectrogram profiles (VSPs) for the square waveform with different sampling rates and same Doppler steering angle. Raw profiles are in blue and final profiles are in red. (a) Nyquist velocity= 77 cm/s. (b) Nyquist velocity= 48.1 cm/s. (c) Nyquist velocity= 38.5 cm/s. (d) Nyquist velocity= 28.9 cm/s. (e) Nyquist velocity= 24.1 cm/s.

Time-domain algorithms detect the signal pitch period directly from the waveform. The pitch detection relies on measurements such as peaks, valleys, zero-crossings, and auto-correlations. Dubnowski et al. [83] developed pitch detection algorithm based on searching the autocorrelation function of the speech signal for its maximum. If the maximum exceeds a predefined threshold, then the location of the maximum is assigned to the pitch period. Miller [84] detected excursion cycles in the signal based on intervals between major zero crossings. The remainder of the algorithm tries to identify the principal excursion cycles, which correspond to true pitch periods.

Frequency-domain algorithms detect the signal pitch period from the impulses of the frequency spectrum, which show up at the fundamental frequency of the pitch period and its harmonics. Schafer and Rabiner [85] developed a cepstral pitch detector algorithm. They computed the cepstrum of each block of the Hamming-weighted speech signal samples. The pitch periods correlate to the location of peak cepstral values.

Hybrid pitch detectors combine features of both the time-domain and the frequency-domain approaches. One of the pitch detection algorithms compared in [82] was proposed by Atal to apply a Newton transformation of the speech signal to spectrally flatten the speech, i.e., to transform the signal into one that has sharp peaks at the pitch impulses and is approximately zero everywhere else. More recently Charalampidis and Kura combined the wavelet transform with autocorrelation techniques to enhance the pitch detection robustness [86].

We adopted a classical autocorrelation (AC) approach to detect the pitch of the virtual spectrogram profiles with some minor changes that enhanced its performance. The classical autocorrelation-based pitch detection relies on detecting the maximum of the autocorrelation function, which should correspond to the pitch period. There are many versions of the autocorrelation function [87]; the raw AC function for a discrete signal $x[n]$ of length N is defined as:

$$AC(m) = \sum_{n=0}^{N-m-1} x[n+m]x[n] \qquad for\, m \geq 0 \qquad (52)$$

The AC function is symmetric, i.e. $AC(m) = -AC(-m) \quad for\, m < 0$. A biased estimate of the AC function is defined as $AC_{biaed}(m) = \frac{1}{N} AC(m)$, while an unbiased version of the AC function is defined as $AC_{unbiaed}(m) = \frac{1}{N-|m|} AC(m)$. The autocorrelation coefficient is a normalized version of the raw AC function where the AC value at zero lag $(m = 0)$ is equal to 1.0. We used the AC coefficient formula in our pitch detection algorithm. Figure 6.17b shows the AC coefficient function for the femoral waveform VSP shown in Figure 6.17a. It is easy to see that the highest peak of the AC coefficient function corresponds to the fundamental pitch of the waveform. However, the peak of the AC function may not always correspond to the pitch duration due to the presence of noise or secondary peaks in the signal. For example, the secondary peak of the carotid waveform VSP shown in Figure 6.17c causes a false maximum peak in its AC coefficient function as shown in Figure 6.17d. In this case the pitch period will be underestimated.

To overcome this problem we applied a simple modification to the AC pitch detection method, which is to raise the VSP $x[n]$ to the power of an odd number p to be $x^p[n]$; we found that $p = 3$ is a reasonable choice. The effect of raising the profile to a power of an odd value is to amplify the fundamental peak against the secondary ones, while preserving the sign of the original signal. Figure 6.18 shows the modified AC coefficient functions after raising the VSPs to the third power. Now the maximum peak of the modified AC coefficient function corresponds to the pitch period for both the femoral and carotid waveforms.

The other periodic waveforms also indicate strong correlation between the maximum peak of their AC coefficient function and their pitch period, as shown in Figure 6.19b and d for the sinusoidal and square waveforms respectively. Since the constant waveform is not periodic, its AC coefficient function does not contain any peaks, as shown in Figure 6.19f. In some cases

of noisy waveforms, some false peaks may show up in the constant waveform AC function and the pitch period may be misestimated. To increase the robustness of our pitch detection algorithm, the peak detection step mandates that a point is higher than all neighboring points in a small neighborhood around the peak. This was necessary to avoid detecting false peaks.

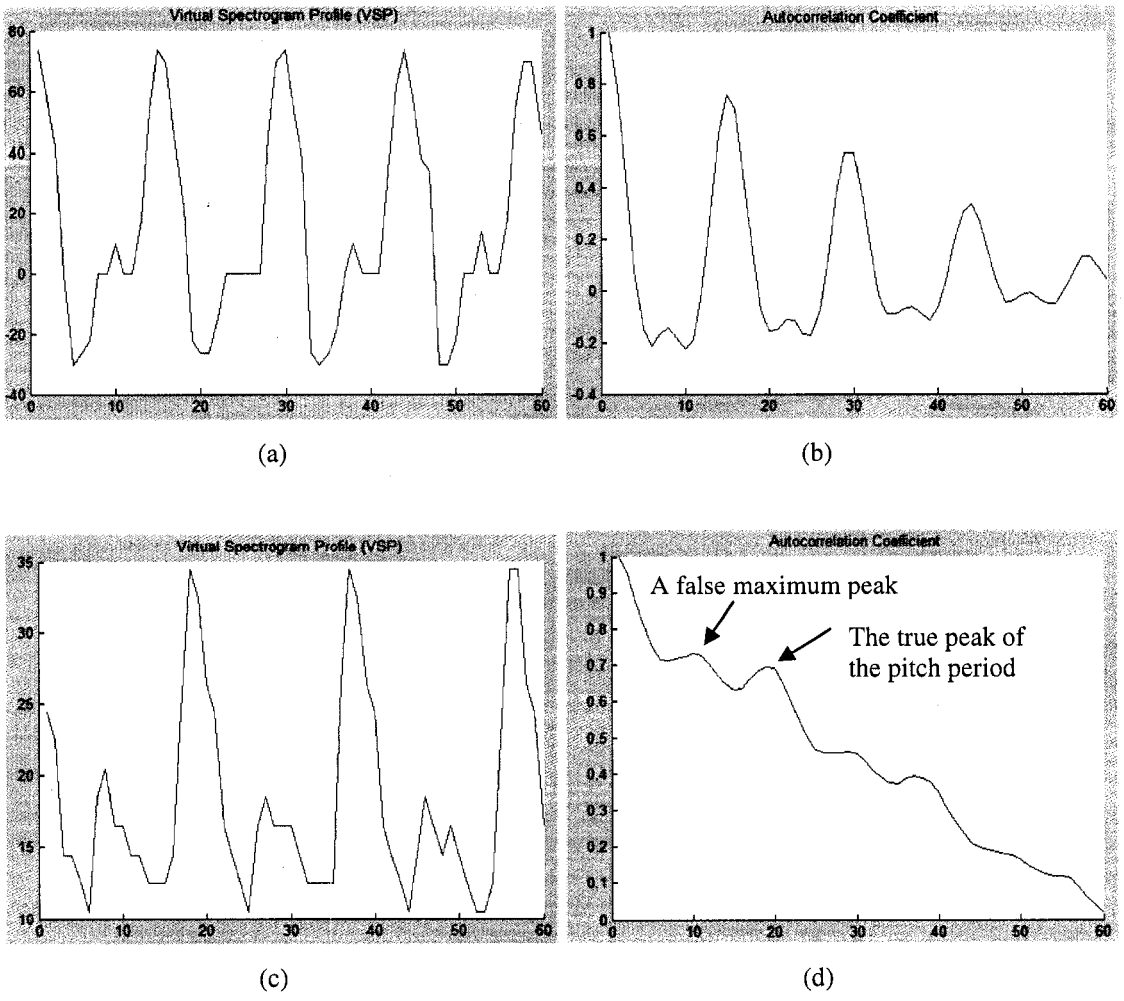


Figure 6.17 Autocorrelation coefficient functions for vessel VSPs. (a) Femoral VSP. (b) Femoral autocorrelation function. (c) Carotid VSP. (b) Carotid autocorrelation function.

6.7. Data Normalization

So far we have applied several data reduction steps to the original unwrapped color Doppler cineloops to end up with a compact representation of the vessel waveforms. The final representation is a single pitch of the 1-D profile of the waveform. In this section we will apply some data normalization steps to further simplify the vessel recognition task. Although most transform-based feature extraction methods account for data normalization in the transform domain, to achieve invariance to possible distortions such as affine transformations, we decided to investigate spatial domain data normalization techniques in order to allow the usage of the raw data as input features for the recognition system.

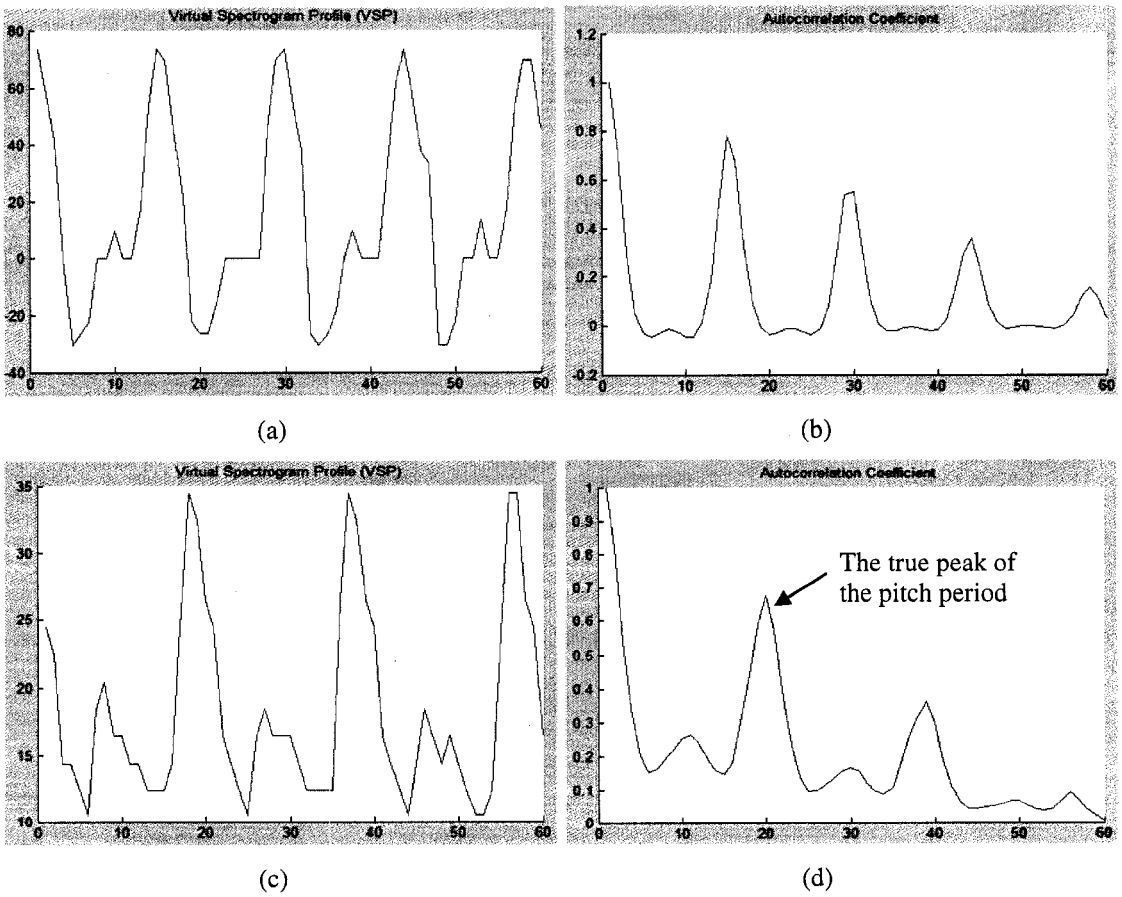


Figure 6.18 Autocorrelation coefficient functions for vessel VSPs raised to third power. (a) Femoral VSP. (b) Femoral autocorrelation function. (c) Carotid VSP. (b) Carotid autocorrelation function.

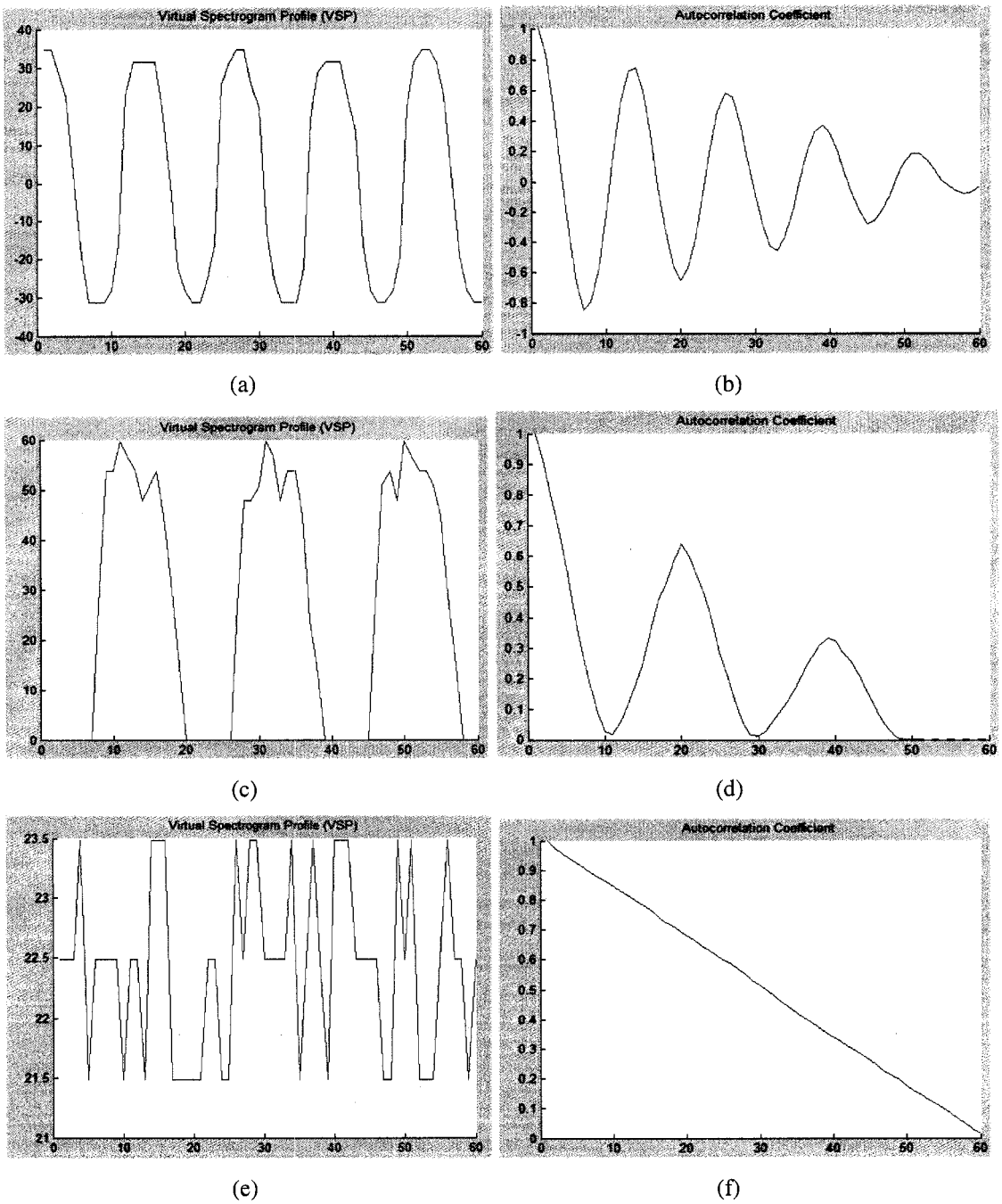


Figure 6.19 AC coefficient functions for other VSPs raised to third power. (a) Sinusoidal VSP. (b) Sinusoidal waveform autocorrelation function. (c) Square VSP. (d) Square waveform autocorrelation function. . (e) Constant VSP. (f) Constant waveform autocorrelation function.

6.7.1. Shift Invariance Normalization

Shift invariance is one of the sources of variation for the color Doppler vessel recognition problem as discussed previously in section 6.2. The acquired data can start anywhere in the heart cycle and the detected waveform single-pitch profiles in the previous section can need to be aligned, before they can be tested for similarity or dissimilarity.

There have been some prior attempts to achieve shift-invariance normalization in the spatial domain. Haitao et al. [88] developed spatial normalization techniques for the Fourier shape descriptors. Their shift invariance approach relied on an evaluation function defined for each shape contour point and derived from the central radius, the curvature, and the gradient magnitude of each contour point. The starting point of the shape is taken as the point with the highest function value. Li et al. [89] developed spatial normalization methods for the shape wavelet descriptors. To achieve rotation invariance, they rotate the shape contour coefficients so that the center of mass is minimized.

Our shift invariance approach is similar to the one developed in [89], but applied to the 1-D profile data instead of the 2-D shape contour data. The single-pitch profile $x[n]$ is circularly shifted for each possible shift k. The center of mass of the shifted profile $x_k[n]$ is calculated as $m_k = \sum_{n=0}^{n=N-1} n x_k[n]$. The shift $k_{\min}$ with the minimum center of mass is identified, and the profile is shifted with this value.

Figure 6.20 shows examples of the shift-invariance normalization results. The different single-pitch waveforms are shown in the left column while the shifted profiles are shown in the right column. The shifted profiles are clearly aligned with each other for the periodic waveforms, such as the carotid and femoral ones. For the aperiodic constant waveform, the shift operation is irrelevant.

6.7.2. Sample Size Invariance Normalization

The pitch detection step resulted in variable-size profiles due to the inaccuracy of the input data. Normalizing the size of the feature vectors is important to allow comparing feature vectors for similarity and dissimilarity measures. Zero-padding and data resampling are well-known methods for size normalization in transform-based feature generation methods such as Fourier and Wavelet transforms. Spatial data resampling methods have also been developed for shape recognition tasks [89]. We chose to resample the single-pitch profiles to obtain fixed size input vectors to be used in the recognition system. The build-in MATLAB© "resample" function, which implements a polyphase resampling filter, was used to achieve fixed size profiles. The size of "32" points was chosen due to its convenience to the transform-based feature extraction methods. The right column of Figure 6.20 shows the resampled profiles of the different waveforms.

6.7.3. Scale Invariance Normalization

Scale invariance is important for the vessel recognition problem in color Doppler ultrasound imaging due to the size variation among different humans for the same vessel and due to system setting variations such as the Doppler steering angle. We applied two scale invariance methods on the waveform profile data. The first scaling method maps the input profile range into the unity range [0:1]. This scaling method was used by different researchers to achieve scale-invariant shapes for the shape recognition task [89].

The input profile $x\ [n]$ is scaled using the formula $x_s[n] = \frac{x[n] - x_{\min}}{x_{\max} - x_{\min}}$, where $x_{\min}$ and $x_{\max}$ are the minimum and maximum velocities of the profile. Figure 6.21 shows examples of the different waveform shifted profiles in the left column and the unity scaled profiles in the middle column.

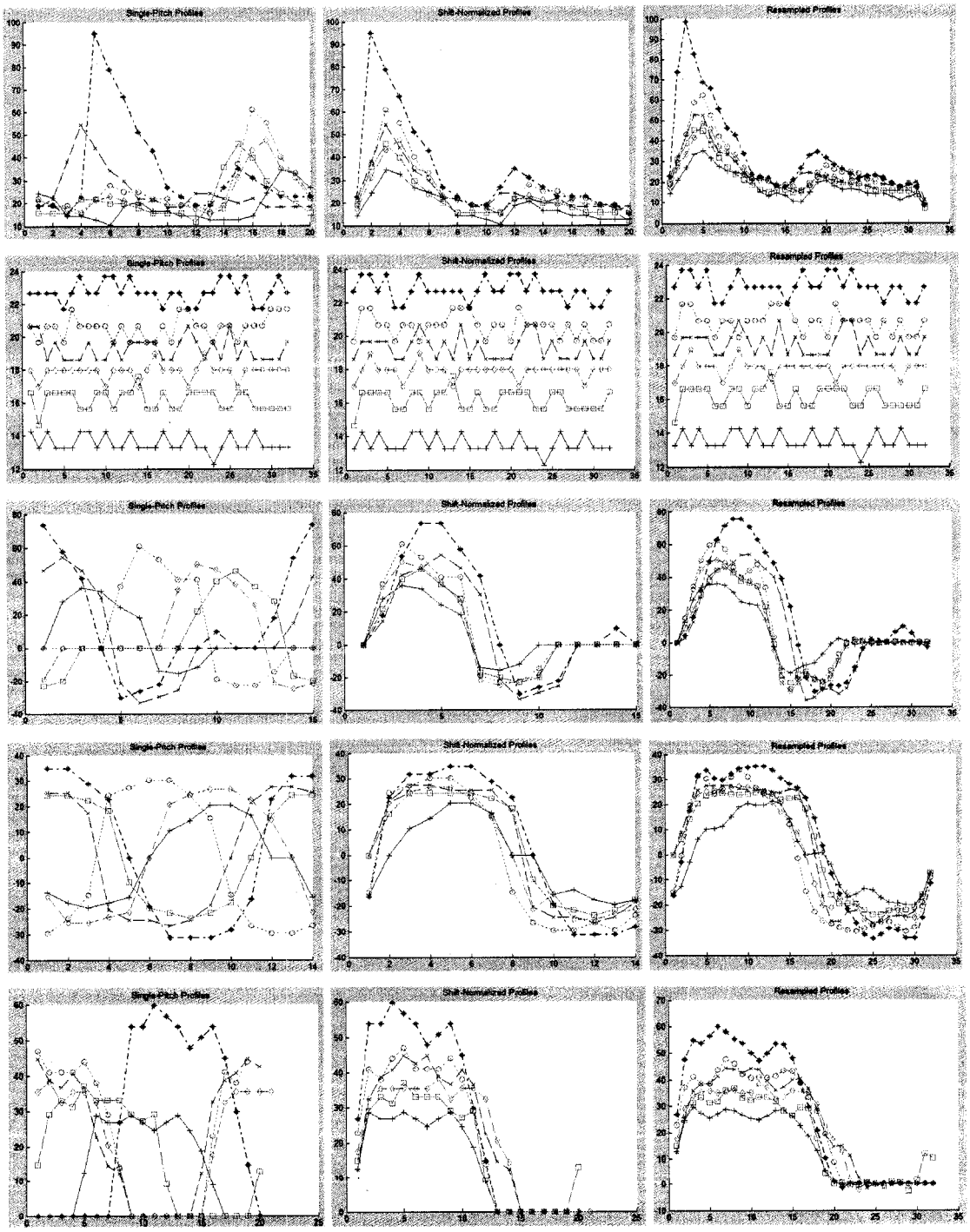


Figure 6.20 Shift-invariance normalization and resampling of the waveform profiles. Left Column shows the single-pitch VSPs, middle column shows the shift-normalized VSPs, and right column shows the resampled shift-normalized VSPs for the carotid, constant, femoral, sinusoidal, and square waveforms respectively.

One main drawback of the unity scaling method is that the phasicity of the waveform is lost due to the mapping of all the waveforms in the range [0:1]. In other words, all negative velocities are mapped to positive ones above zero velocity. This is a concern that may affect the performance of the recognition system and will be further evaluated in the next chapter. For that reason we call the unity scaling method the "lost-sign" scaling method.

To overcome the lost phasicity drawback, another scaling method was developed to achieve scale-invariance while preserving the phasicity, or the sign, of the original profile. In this method the input profile $x\ [n]$ is scaled using the formula $x_s[n] = \frac{x[n]}{x_{\max}}$, where $x_{\max}$ is the maximum velocity of the profile. This formula guarantees that the scaled profile will be bounded from the top by 1 and it will not be bounded from the bottom. However knowing that the minimum velocity will not exceed the maximum velocity in magnitude, the profile will be bounded by -1 in reality.

To give solid examples, both the carotid and square waveforms will be bounded by the range [0:1] since the minimum velocity is 0. If we assume the magnitude of the minimum velocity of the femoral waveform is half of its maximum one, i.e. $x_{\min} = -x_{\max}/2$, then the scaled profile will be bounded by the range [-0.5:1]. For the sinusoidal velocity, the magnitude of the minimum velocity is roughly equal to its maximum velocity, i.e. $x_{\min} = -x_{\max}$, thus its scaled profile will be bounded by the range [-1:1]. The right column of Figure 6.21 confirms these findings where it shows examples of different waveform scaling using this method, which we called "*preserved-sign*" scaling. The two scaling methods will be evaluated against the performance of the recognition system in the next chapter.

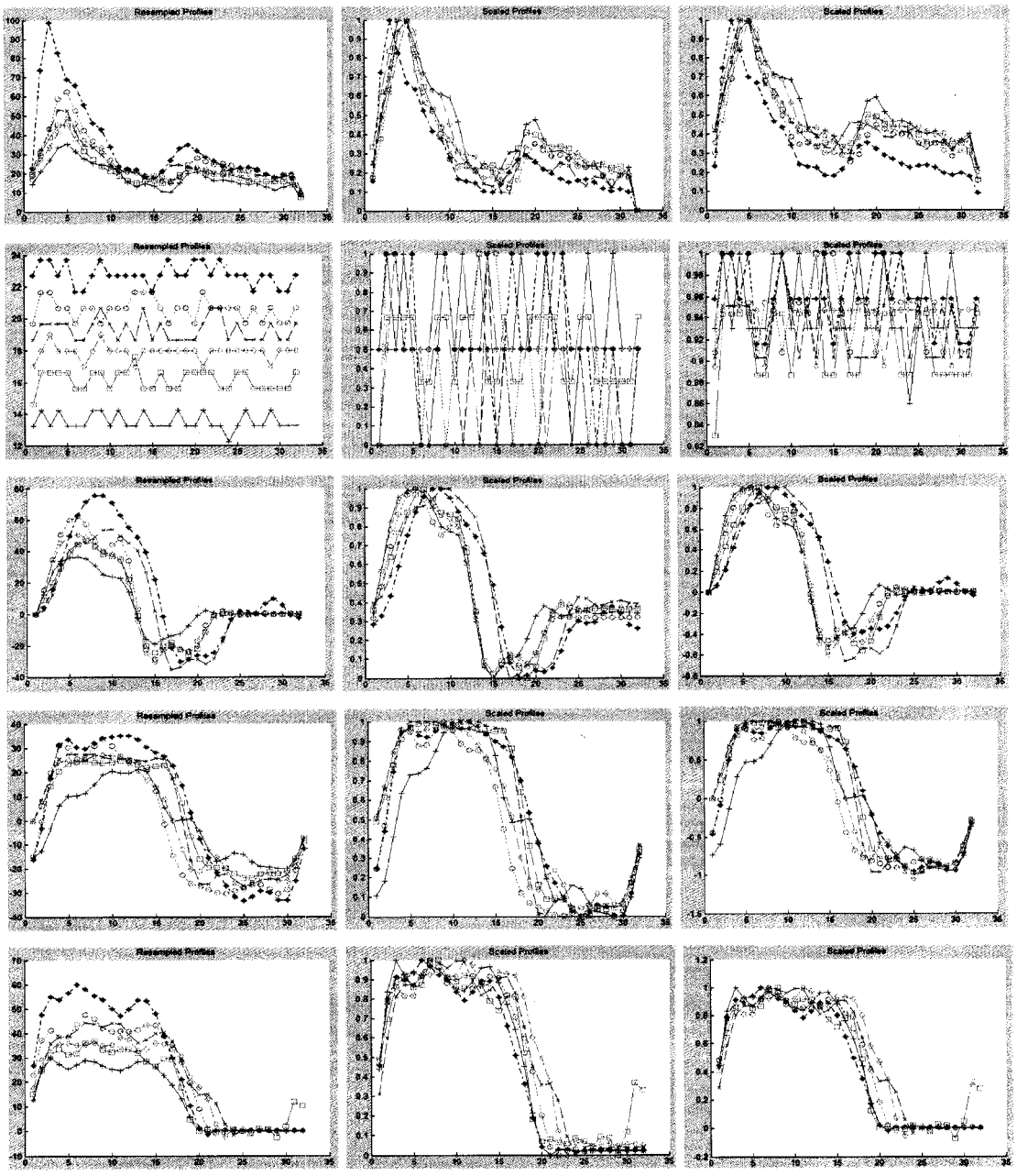


Figure 6.21 Scale-invariance normalization of the waveform profiles. Left Column shows the single-pitch VSPs, middle column shows the lost-sign scaled VSPs, and right column shows the preserved-sign scaled VSPs for the carotid, constant, femoral, sinusoidal, and square waveforms respectively.

6.8. Transform-based Feature Extraction

The normalized VSPs obtained from the unwrapped color Doppler cineloops represent waveforms of different vessels. Our remaining task is to build a recognition system that can distinguish the underlying vessel. In the literature, there has been similar attempts for recognizing other biomedical waveforms. Whitehouse et al. [90] developed a statistical pattern recognition for spectral Doppler waveforms of amniotic fluid displacement during fetal respiration to distinguish healthy from upnormal waveforms. Kinematic Doppler features, such as peak velocity, cycle duration, and pause time, were extracted and fed to a linear discriminant analysis system. Christodoulou and Pattichis [91] developed an unsupervised recognition system for EMG signal classification for the diagnosis of neuromuscular disorders. The raw EMG waveforms were segmented into atomic components that represent the smallest functional units of the muscle that can be voluntarily activated. Two classifiers, a neural network classifier and a statistical classifier that relies on the Euclidean distance between the waveforms, were evaluated. Cowe and Evans [92] applied a principal component analysis technique to discriminate true embolic signals from artifacts in transcranial Doppler ultrasound signal.

In our case, the normalized VSPs can be used directly as inputs to the vessel recognition system. Moreover, they can be used as inputs to linear transformations, which can pack the relevant information in a fewer number of features and achieve higher performance and robustness. Transform-based feature extraction methods represent popular choices for many recognition problems, such as character, shape, and face recognition tasks. Usually in these tasks, the raw feature vector is large and redundant. Transform-based methods help with reducing its size and remove redundancy. At the same time they help with data normalization to achieve invariance against different sources of distortions, such as translation, scaling, rotation, and starting point. Normalization can be much easier and more robust in the transform domain than the spatial domain for some applications.

The raw feature vector length is not an issue for our vessel recognition problem, since the normalized VSPs represent a compact size feature vector (32 points). However the data

could be very noisy due to the inherent nature of the color Doppler data and the inaccuracy of the different processing steps involved so far, particularly the segmentation, phase unwrapping, and data normalization steps. For that reason we decided to apply transform-based feature extraction methods and compare them against the raw feature vectors in terms of impact on the recognition system performance. The three well-known transform-based feature extraction methods considered are Fourier descriptors, wavelet descriptors, and moment descriptors.

6.8.1. Fourier Descriptors

The Fourier descriptors for shape representation and recognition have been used since the seventies. Zahn and Roskies [93] developed a Fourier descriptor for planar closed curves. They developed an angular difference representation of the shape contour to which the Fourier transform was applied. The coefficients of their Fourier descriptors are calculated as follows:

$$a_n = -\frac{1}{n\pi}\sum_{k=1}^{m}\Delta\varphi_k \sin\frac{2\pi n t_k}{T}, \quad b_n = \frac{1}{n\pi}\sum_{k=1}^{m}\Delta\varphi_k \cos\frac{2\pi n t_k}{T}, \tag{53}$$

where $\Delta\varphi_k$ is the angular difference between two successive line segments on the contour. T is the length of the boundary curve, consisting of m line segments, t_k is the accumulated length of the boundary from the starting point to the k^{th} point. The coefficients a_n and b_n are size and translation invariant. Rotation invariance can be obtained by transforming to polar coordinates. Then the amplitudes are independent of rotation and mirroring, while the phase angles are not.

Granlund [94] applied the Fourier series transform to the complex number representation of hand-written character contours $z(t) = x(t) + jy(t)$ as:

$$a_n = \frac{1}{T}\int_0^T z(t)e^{-j2\pi nt/T}dt \qquad (54)$$

The coefficients a_n are translation-invariant except a_0 which is the center of gravity of the character contour. T is the total contour length. Granlund derived scale and rotation invariant coefficients as follows:

$$b_n = \frac{a_{1+n}a_{1-n}}{a_1^{\,2}}, \qquad d_{mn} = \frac{a_{1+m}^{n/k}a_{1-n}^{m/k}}{a_1^{(m+n)/k}} \qquad (55)$$

where $n \neq 1$ and k is the greatest common divisor of m and n.

Chen et al. [95] compared the Fourier descriptors and moment invariant descriptors for character shape recognition. They applied the discrete Fourier transform (DFT) to the complex representation of the contour points defined as:

$$a_n = \frac{1}{K}\sum_{k=0}^{K-1} z(k)e^{-j2\pi nk/K} \qquad n = 0,1,\ldots,K-1 \qquad (56)$$

where K is the total number of contour points. The translation, scale, and rotation invariant descriptors are derived from the original DFT coefficients as follows:

$$c_{n-2} = \frac{|a_n|}{|a_1|} \qquad n = 2,3,\ldots,K-1 \qquad (57)$$

We adopted the invariant Fourier descriptors as calculated in equation (57) when using the raw waveform profiles without scaling. When using the scaled profiles, we use the DFT coefficients directly, except the first DC coefficient a_0. Since the Fourier transform is symmetrical, only half of the coefficients are considered. Figure 6.22 shows a simulation of

possible variations (scaling, translation, shifting, and mirroring) applied to a femoral profile and their corresponding Fourier descriptors as calculated in equation (57). It is clear that all the descriptors are the same for all the simulated variations. Figure 6.23 shows the Fourier descriptors for all the training data of all waveform types. The waveform profiles were scaled using the preserved-sign scaling technique that is described in section 6.7.3. The direct DFT coefficients from equation (56) are used as descriptors since the scaling is done in the spatial domain. One observation about the figure is that all waveform types include some inaccurate and noisy datasets that are easily spotted on the profile plot. Their corresponding Fourier descriptors are easily identified on the Fourier descriptor plots.

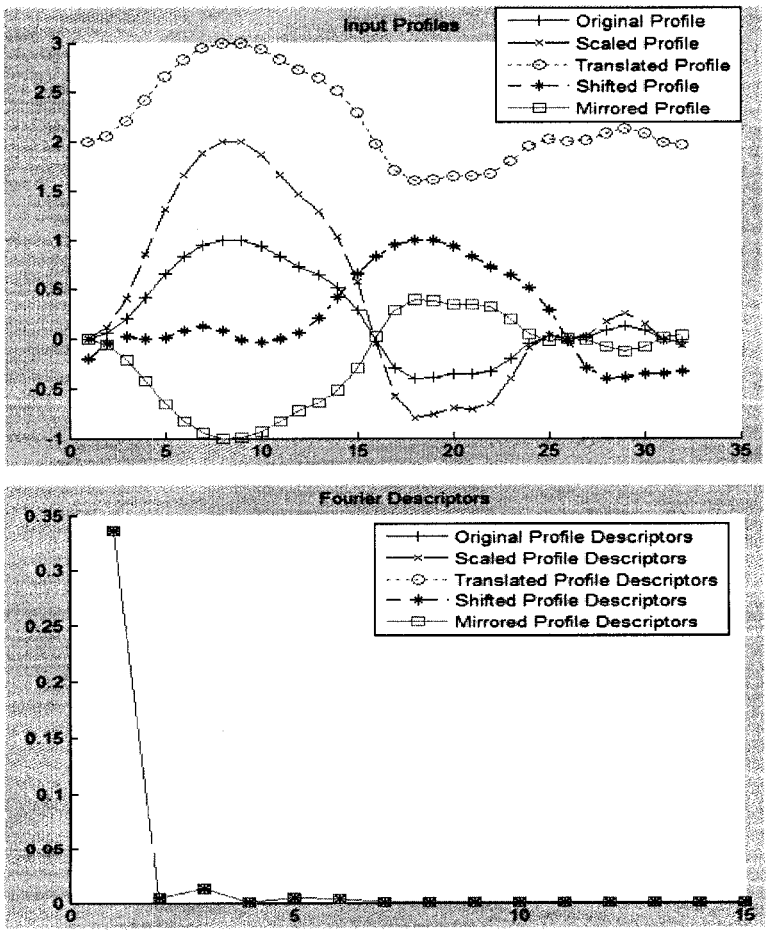


Figure 6.22 Invariant Fourier Descriptors Evaluation. Top: simulation of different sources of variation (scaling, translation, shift, and mirroring) applied to a femoral profile. Bottom: the calculated invariant Fourier descriptors.

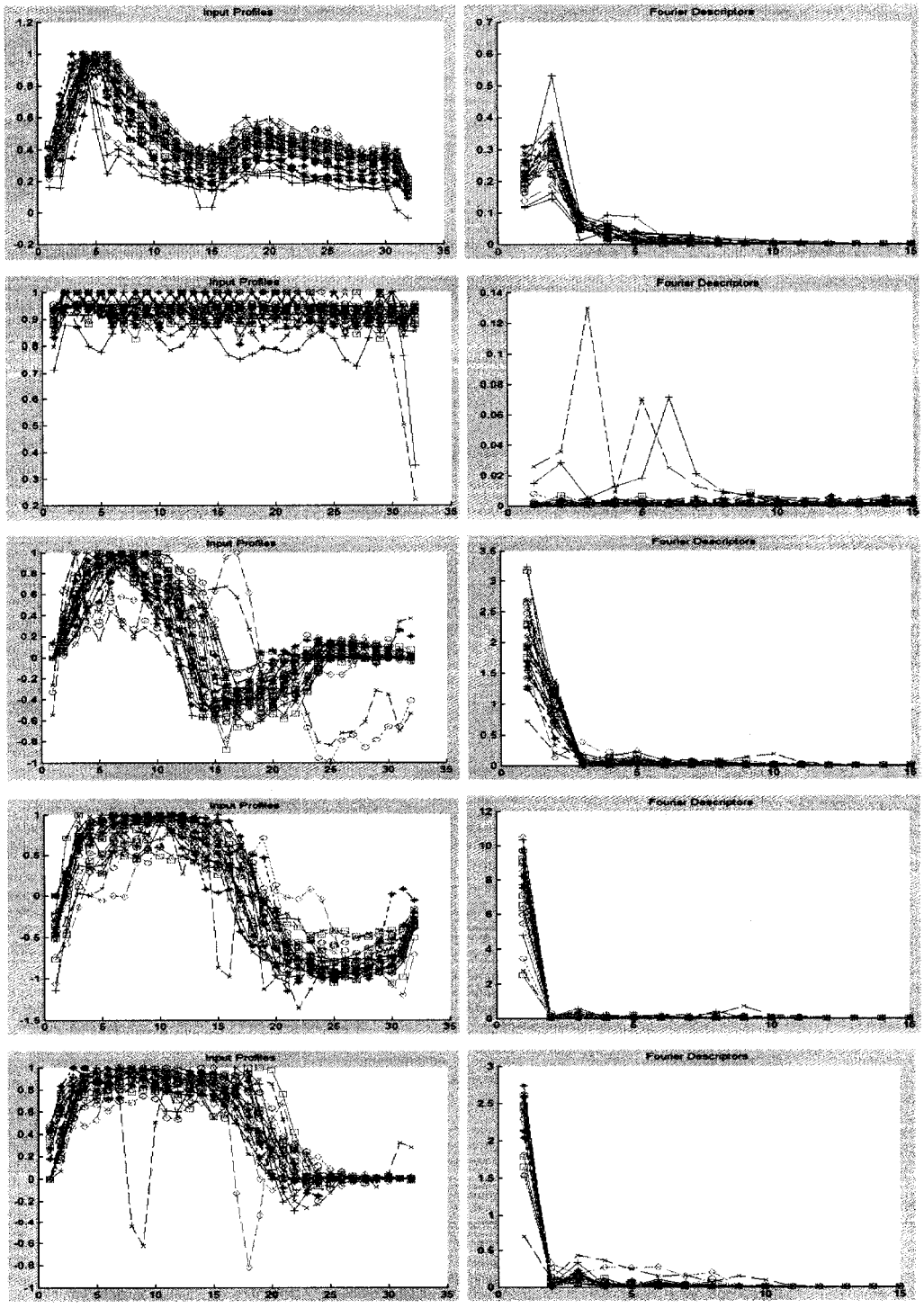


Figure 6.23 Invariant Fourier Descriptors for different waveforms. The left column shows the preserved-sign scaled profiles, while the right column shows the Fourier descriptors for carotid, constant, femoral, sinusoidal, and square waveforms.

6.8.2. Wavelet Descriptors

The Fourier descriptors described in the previous section uses global sinusoids as basis functions. In recent years, the wavelet transform has become an active area of research for multiresolution signal and image analysis. The wavelet descriptor is more efficient in representing local features of a curve due to the spatial and frequency localization property of wavelet basis functions. One main advantage of the wavelet representation is that a small change in the shape of a curve will not cause a large change in its wavelet representation, and hence, the wavelet representation is stable with respect to noise.

Chuang and Kuo [96] described the theoretical foundations and some applications of wavelet descriptors of planar curves. The periodized wavelet transform is characterized by a scaling function $\phi(t)$ whose translations, defined as $\phi_n^m(t) = 2^{-m/2}\phi(2^{-m}t - n) \quad for\ n,m \in Z$, form an orthonormal basis for the wavelet subspace. For each scaling function there is a corresponding wavelet function $\psi(t)$. Its translations, defined as $\psi_n^m(t) = 2^{-m/2}\psi(2^{-m}t - n) \quad for\ n,m \in Z$, also form an orthonormal basis for the wavelet subspace. Periodic scaling and wavelet functions are defined as $\hat{\phi}_n^m(t) = \sum_{l \in Z} \phi_n^m(t+l)$ *and* $\hat{\psi}_n^m(t) = \sum_{l \in Z} \psi_n^m(t+l)$. A continuous function $f(t)$ can be expanded using the periodic wavelet transform as follows:

$$f(t) = \sum_n c_n^{M_c} \hat{\phi}_n^{M_c}(t)dt + \sum_{m=M_f}^{M_c} \sum_n d_n^m \hat{\psi}_n^{M_c}(t)dt \qquad (58)$$

Where $c_n^{M_c}$ and d_n^m are the approximation and detail coefficients of the wavelet transform and defined as:

$$c_n^{M_c} = \int_0^1 f(t)\hat{\phi}_n^{M_c}(t)dt; \quad and \quad d_n^m = \int_0^1 f(t)\hat{\psi}_n^m(t)dt \tag{59}$$

Chuang and Kuo used a parameterized representation of the closed planer curve as:

$$\alpha(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \quad t(l) = l/L \quad 0 \le l \le L \tag{60}$$

where t is the normalized arc length, l is the arc length, and L is the total arc length. The parameterized curve can be expanded by the approximation and wavelet coefficients as follows:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_a^M(t) \\ y_a^M(t) \end{bmatrix} + \sum_{m=M-m_0}^{M} \begin{bmatrix} x_d^m(t) \\ y_d^m(t) \end{bmatrix} \tag{61}$$

where

$$x_a^M(t) = \sum_n a_n^M \hat{\phi}_n^M(t), \quad y_a^M(t) = \sum_n c_n^M \hat{\phi}_n^M(t) \tag{62}$$

are called the approximation signals at scale M and

$$x_d^m(t) = \sum_n r_n^m \hat{\psi}_n^m(t), \quad y_d^m(t) = \sum_n d_n^m \hat{\psi}_n^m(t) \tag{63}$$

are called the detail signals at scale m with $m = M - m_0$ being the finest scale and $m = M$ being the coarsest scale. The approximation coefficients (a_n^M, c_n^M) and the detail coefficients (r_n^m, d_n^m) are used as the planar curve descriptors. The planer shape can be reconstructed using the approximation coefficients and a subset of the detail ones. The more coefficients included the more details appear in the constructed shape. Hence the wavelet transform represents a multiresolution representation of the planer shapes.

Chuang and Kuo also derived direct relationships between the spatial domain transformations, scaling, translation, and rotation, and simple transformations of the wavelet coefficients as follows:

- Scaling a planar curve will affect both the approximation and detail wavelet coefficients with the same scaling level. For the polar representation of the wavelet coefficients, scaling will affect the magnitude of the coefficients but not the phase.
- Translating a planar curve will affect the approximation coefficients with the same displacement, but not the detail coefficients.
- Rotating a planar curve will affect the phase of both the approximation and detail wavelet coefficients, but not the magnitude.

To normalize the wavelet coefficients in order to achieve scale, translation, and rotation invariant shape description, Chuang and Kuo suggested the following:

- For displacement normalization, subtract the mean of the approximation coefficients from themselves. The detail coefficients are already translation-invariant.
- For scale normalization, divide the magnitudes of the coefficients by the average magnitude of all coefficients.
- For rotation normalization, subtract the average phase of all coefficients from the coefficients.

One source of variation that was omitted by Chuang and Kuo in [96] is the starting point of the planar curve. The authors assumed the starting point of the shape is known and fixed through out their experiments. A few publications have tried to address this problem. Hung [97] derived a generalized uniqueness property inherent in the discrete periodized wavelet transformation. The uniqueness property facilitates a quantitative analysis of the one-to-one mapping between the variation of wavelet coefficients and the starting point shift of the planar curve. Hung proposed a new shape descriptor called the uniqueness wavelet descriptor by which the starting point is fixed entirely within the context of the wavelet representation is proposed. He applied a fast binary search method to find the optimal starting point, in the noise resistance sense.

Another alternative to normalize the starting point is the one suggested by Li et al. [89], which applies a spatial domain method to rotate the shape in order to minimize its center of mass. We adopted this method in section 6.7.1 in order to achieve shift invariance normalization, which can be applied to any transform-based descriptors, not only the wavelet transform.

We adopted the wavelet descriptor concept for planar curve to our waveform profiles problem. We chose a 1-level wavelet decomposition using the "Daubechies" wavelet filter type. To obtain scaling, translation, and rotation invariant coefficients, we applied the following normalization steps:

- Subtract the mean of the approximation coefficients from the approximation coefficients to achieve translation-invariance.
- Divide all approximation coefficients by the maximum of the absolute approximation coefficients and divide all detail coefficients by their absolute mean to achieve scale-invariance. The maximum is used for the approximation coefficients instead of the mean since the mean is already zero due to the translation normalization.
- Consider only the absolute value of both the approximation and detail coefficients as the wavelet descriptors to achieve rotation invariance.

Figure 6.24 shows a simulation of different variations applied to a femoral waveform. The left column shows that both the approximation and detail coefficients are invariant to all transformations except the shifting one. The right column shows the effect of applying the spatial domain shift operation to the profiles to achieve invariant descriptors for all transformations including the shift operation. Figure 6.25 shows the approximation and detail wavelet descriptors for all the training dataset for all waveforms.

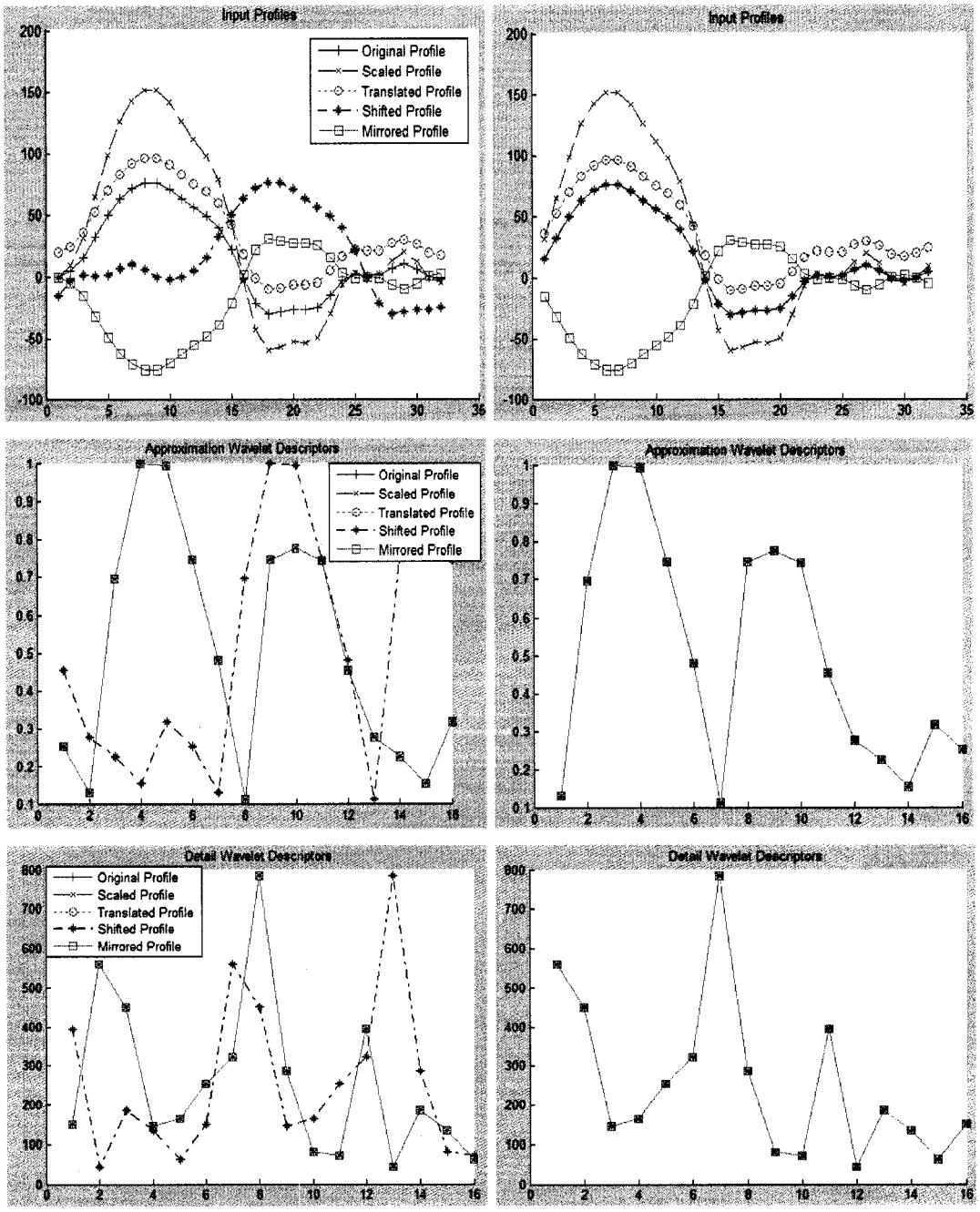


Figure 6.24 Invariant Wavelet Descriptors Evaluation. Top left: simulation of different sources of variation (scaling, translation, shift, and mirroring) applied to a femoral profile. Middle left: the approximation coefficients. Bottom left: the detail coefficients. Top right: the same waveform simulation with spatial shift operation. Middle right: the approximation coefficients. Bottom right: the detail coefficients.

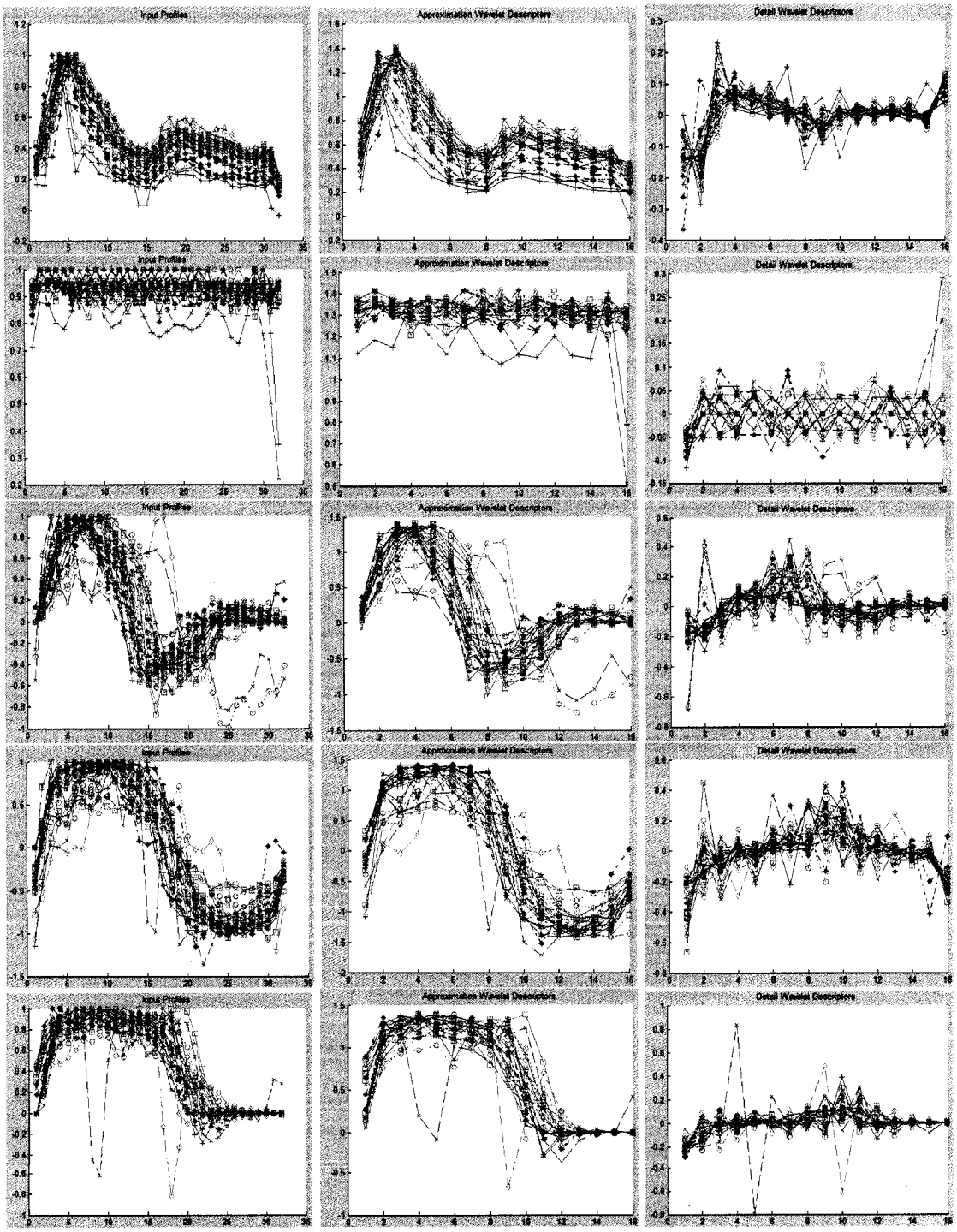


Figure 6.25 Invariant Wavelet Descriptors for different waveforms. The left column shows the preserved-sign scaled profiles, the middle column shows the wavelet approximation descriptors, and the right column shows the wavelet detail descriptors for carotid, constant, femoral, sinusoidal, and square waveforms.

6.8.3. Moment Descriptors

Moment-based invariants are the most common region-based image invariants and have been used as pattern features in many applications, such as character recognition [98]. While both the Fourier and wavelet descriptors are boundary-based image features that only compute the pixels along the image contours, moment invariants are region-based image features that take all of the pixels of the image into account and can be calculated for grayscale as well as binary image regions.

Hu [99] was the first to introduce the use of moment invariants for pattern recognition. He introduced orthogonal moment invariants (invariant to translation, scale and rotation) using nonlinear combinations of regular moments. Regular moments for a continuous image function (or region) $f(x, y)$ are defined as:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x, y)dxdy \quad p, q = 0,1,2,... \tag{64}$$

where m_{pq} is the $(p+q)^{th}$ order moment of the image region $f(x, y)$. The translation-invariant central moments are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x, y)dxdy \quad p, q = 0,1,2,... \tag{65}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$ represent the centroid of the image region. The relationship between the central moments of a scaled image region, with a scaling factor of α, and the original region's moments is given by:

$$\mu'_{pq} = \alpha^{(p+q+2)}\mu_{pq} \tag{66}$$

where μ'_{pq} is the $(p+q)^{th}$ order moments of the scaled image region $f(\alpha x, \alpha y)$. Thus scale-invariant, as well as translation-invariant, moments can be defined as:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = (p+q+2)/2, \; p+q = 2,3,... \tag{67}$$

Rotation-invariant moments can be derived from η_{pq}. For example the following second-order moment functions are rotation-invariant, as well as translation and scale-invariant:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi 2 &= (\eta_{20} - \eta_{02})^2 + \eta_{11}^2 \end{aligned} \tag{68}$$

Hu [99] derived moment invariants for general linear transformations that are functions of 2nd and 3rd order central moments. Reiss [100] has shown that Hu invariants are in fact incorrect and provided corrected expressions for them. Chen et al. [95] have compared Hu's invariants against the Fourier descriptors for character recognition. Recently Mamistvalov [101] published a generalization theory for n-dimensional moment invariants. Zernike moment invariants, which are derived from well-known complex sets of orthogonal functions known as Zernike polynomials, have also been used for shape and character recognition [98]. Mertzios et al. [102] derived a 1-D shape representation function of the boundary distances from the shape centroid, and then applied 1-D moments as features for shape recognition.

For our problem, we decided to derive 1-D moments from the waveform profiles and use them as discriminating features. The 1-D moments characterize the shape of a PDF distribution. The first few moments have direct physical meaning, such as mean, variance, skewness, and kurtosis. The mean is the average value of the distribution. The variance measures the deviation from the mean value. The skewness is a measure of the asymmetry of the data around the mean. If skewness is negative, the data are spread out more to the left of

the mean than to the right. If skewness is positive, the data are spread out more to the right. The Kurtosis is a measure of how outlier-prone a distribution is. The kurtosis of the normal distribution is 3. Distributions that are more outlier-prone than the normal distribution have kurtosis greater than 3; distributions that are less outlier-prone have kurtosis less than 3. We expected that the 1-D moments will capture the important global characteristics of the different waveforms. The central moment of order p of a distribution is defined as:

$$\mu_p = E(\mu - \bar{\mu})^p = \sum_{n=1}^{N} (x_n - \bar{x})^p f(x_n) \quad for\ p = 1,2,3,... \tag{69}$$

where $\bar{\mu}$ is the sample mean of the distribution, $E()$ is the expected value of a discrete random variable (RV), $f(x)$ is the RV PDF or in our case the profile 1-D function, and N is the total number of discrete points of the function.

To investigate the invariance properties of the 1-D moments, we simulated different valid geometric transformations (scaling, translation, shift, and mirroring) on a femoral waveform, as shown in the left column of Figure 6.26. Since the values of the different moments can be quite different due to the higher order powers used in equation (69), we applied two different scaling methods to the calculated moments. The first scaling method is to calculate the logarithms of the absolute values of the moments ($\log|\mu_p|$), which is the most common moment scaling method in the literature; we call this scaling method the "log-scaled moments". The top right figure in Figure 6.26 shows the log-scaled moments of the transformed waveforms on the left side; it is evident that the log-scaled moments are invariant to translation, shift, and mirroring, but not to scaling. The second scaling method is to multiply all moments by a constant value to avoid very small or very large numbers; we call this method the "linearly-scaled moments" since it preserves the distance between all the moments. The bottom right figure of Figure 6.26 shows the linearly-scaled moments of the transformed waveforms; it is evident from the figure that the linearly-scaled moments are invariant to translation and shifting, but not to scaling or mirroring. The mirroring moments differ only by an opposite sign from the original moments. However the advantage of the

linearly-scaled moments could be the ability to differentiate between different phasic waveforms by preserving the sign of the moments, as we will investigate in the next chapter. The linearly-scaled moments for all the training datasets of the different waveforms are shown in Figure 6.27. The waveforms are scaled in the spatial domain using the preserved-sign scaling methods.

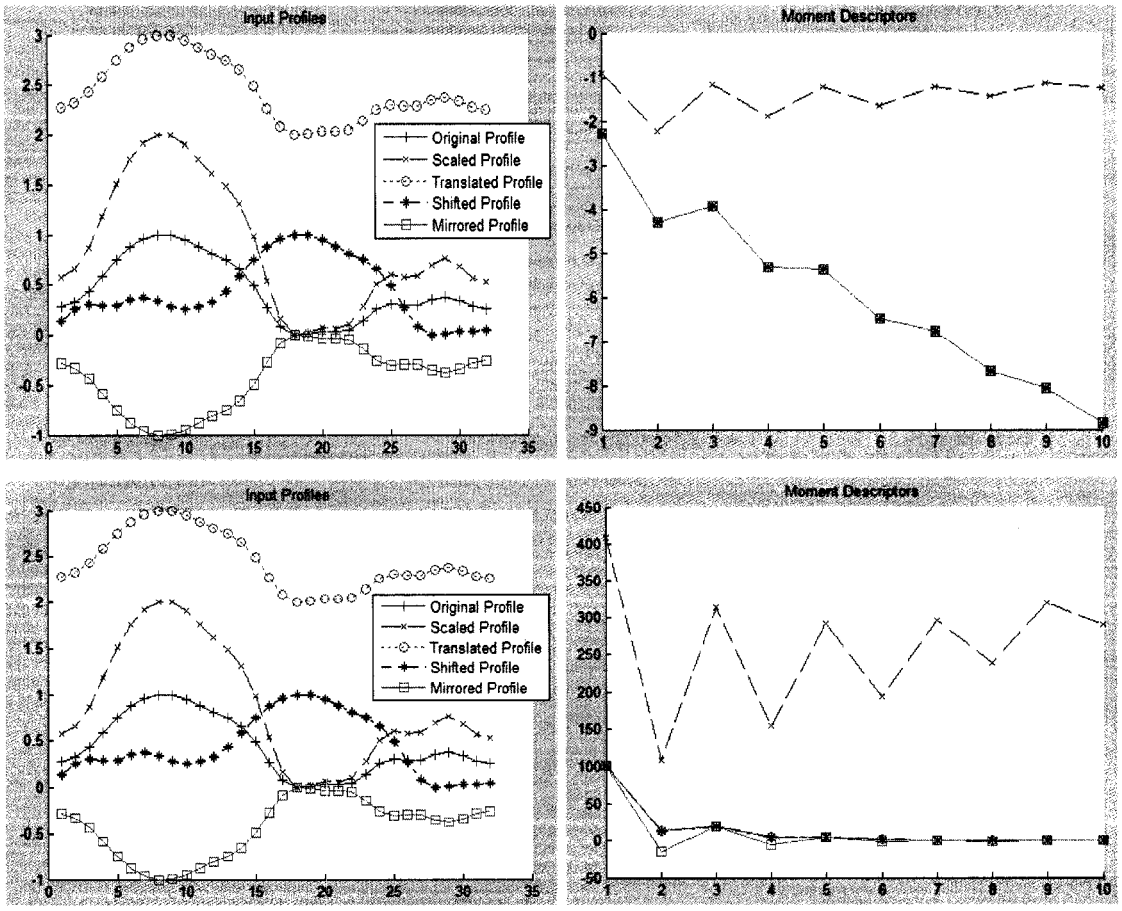


Figure 6.26 Invariant Moment Descriptors Evaluation. Top left: simulation of different sources of variation (scaling, translation, shift, and mirroring) applied to a femoral profile. Top Right: the log-scaled moments. Bottom left: the same simulation profiles. Bottom right: the linearly-scaled moments.

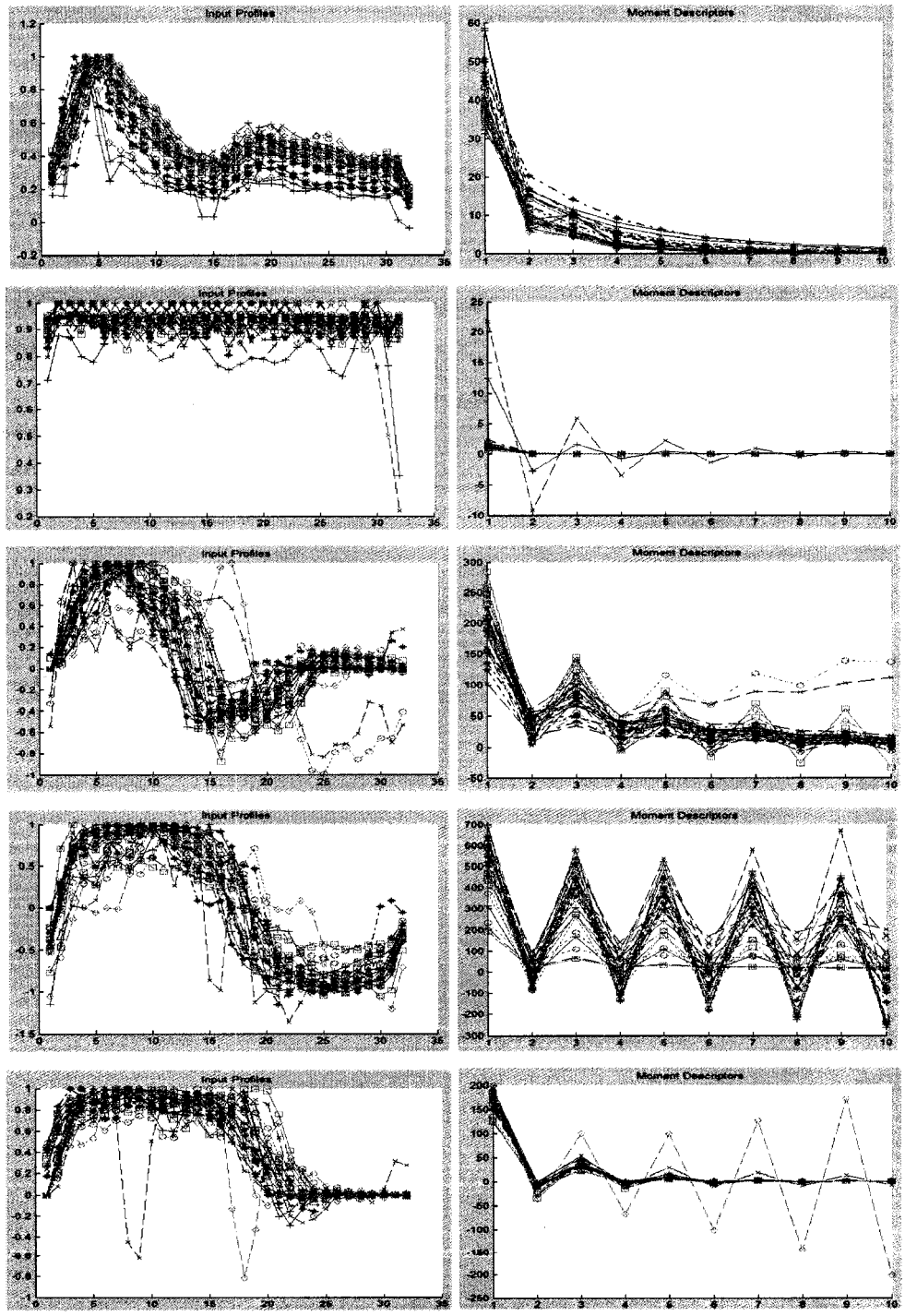


Figure 6.27 Linearly-scaled Moment descriptors for different waveforms. The left column shows the preserved-sign scaled profiles, the right column shows the moment descriptors for carotid, constant, femoral, sinusoidal, and square waveforms.

6.9. Conclusions and Future Work

In this chapter we have identified the different sources of variation in color Doppler ultrasound imaging. We have developed different data reduction techniques to be applied on the unwrapped phase cineloops. The final outputs represent 1-D waveform profiles of a single heart cycle. These waveform profiles can be used directly as inputs to a classifier system to recognize different vessels. In addition we have also applied different linear transformations including the Fourier, wavelet, and moment transforms to the waveform profiles. Several data normalization techniques, both in the spatial domain and in the transform domain, were suggested to achieve invariant features for identified sources of variation. In the following chapter we will discuss the last phase of our vessel recognition system and build a classifier that can identify vessels from the different input forms that were developed here.

The approach we have chosen for the vessel recognition problem is a statistical pattern recognition approach in which a vessel is represented by a feature vector that represents the waveform profile or its transformation coefficients such as the Fourier, wavelet, or moment coefficients. Structural pattern recognition, where an entity is represented by its primitive parts, their attributes, and their relationships, as well as by its global features, is another possibility. More specifically the vessel waveform profiles can be represented symbolically by relational trees (RTs) that emphasize their structural composition. Relational trees are data structures that describe the relative size and placement of peaks and valleys in a waveform. Each nonterminal node in an RT represents a valley in the waveform and each terminal node represents a peak. The root node of the RT is chosen to represent the deepest valley in the waveform, and the valleys are nested according to relative depth. Relational trees were first introduced by Erich and Foith to represent waveforms [103]. The authors have also developed tree similarity measures to indicate waveform similarity based on waveform shape. Structural waveform analysis techniques have proved to be particularly suitable in some application areas where there is a large amount of natural distortion and noise present, such as seismic and ECG waveform classification [104], [105]. As a future work, the relational-tree similarity approaches may be tried on our vessel profile waveform classification to handle the increased level of inherent noise due to the segmentation and preprocessing steps involved.

Chapter 7 Vessel Classifier Design and Evaluation

7.1. Introduction

In the previous chapter we applied different preprocessing steps and feature extraction techniques to the unwrapped color Doppler ultrasound cineloops. In this chapter we will handle the last step of our vessel recognition system which is the classification task. The training datasets, which were acquired according to the protocol in section 6.3, will be used to build a classifier that can recognize the vessels of an unknown color Doppler cineloop.

This chapter represents the final stage of the vessel recognition system where the classifier is designed and evaluated, as shown in Figure 7.1. The chapter describes a number of experiments that we conducted to answer important questions regarding the vessel recognition system: Which preprocessing method is more efficient? Which feature extraction method is best? What is the optimal number of features, or attributes, to use? Which classifier is best for our problem? Which descriptor is more noise resilient than the others? We chose to use a well-known open source data mining software package called WEKA, or "Waikato Environment for Knowledge Analysis" [106], for all the experiments. WEKA is a collection of machine learning algorithms for data mining tasks implemented in the JAVA programming language. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization and it offers a user-friendly experimental environment to facilitate working with data mining tasks. The data preprocessing algorithms, described in the previous chapter and implemented in Matlab©, output the different feature descriptors in a text format (ARFF) that is readable by the WEKA software package [106]. Figure 7.2 shows a screen shot of the WEKA explorer GUI that is used to preprocess and classify the waveform datasets.

Figure 7.1 The stages of the vessel recognition system. The last stage is for the design and evaluation of the classifier.

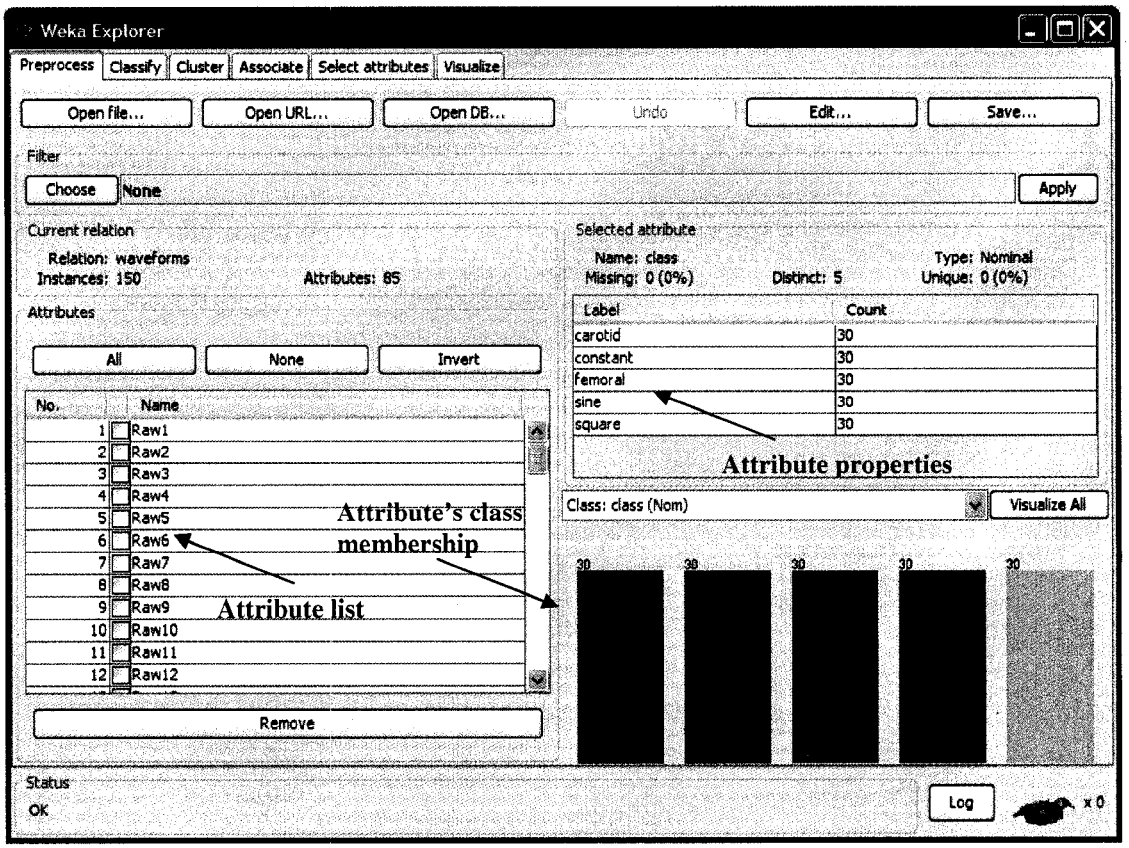


Figure 7.2 A screen-shot of the WEKA Explorer interface.

7.2. Pattern Recognition Concepts

In this section we will briefly introduce the relevant pattern recognition concepts and tools that will be used in this chapter.

7.2.1. The Naive Bayes Classifier

Before we start describing the experiments, we will discuss briefly a classical classifier that will be used for most of the experiments in this chapter. The "Naive Bayes Classifier" is a well-known classifier that is based on Bayesian decision theory [107]. The Bayes classifier accounts for the statistical nature of the generated features, which fits our problem with the expected variations in the extracted features due to the nature of the color Doppler acquisition process and the different image processing steps involved.

For a given classification task of M classes, $w_1, w_2, ..., w_M$, and an unknown pattern with feature vector x, the Bayes classifier attempts to maximize the *a posteriori* probability $P(w_i \mid x),\ i = 1,2,...,M$. The *a posteriori* probability represents the probability that the unknown pattern belongs to the respective class w_i. The Bayes rule calculates the *a posterior* probability as follows:

$$P(w_i \mid x) = \frac{p(x \mid w_i)P(w_i)}{p(x)} \qquad (70)$$

where $p(x)$ is the PDF of x and can be ignored since it is independent of all the classes, $P(w_i)$ is the a priori probability of the classes, which is either known or can estimated from the frequency of each pattern in the training set, and $p(x \mid w_i)$ is the class-conditional PDF that describes the distribution of the feature vectors in each of the classes. The Naïve Bayes classifier assumes class-conditional PDFs with normal distributions, which are fully

characterized by their mean values and covariance matrices. Another simplification applied by the Bayes classifier is the independence of all the features in the feature vector [107]. The unknown pattern x is assigned to class w_i if $P(w_i \mid x) > P(w_j \mid x) \quad \forall j \neq i$.

7.2.2. Cross-validation

Cross-validation, sometimes called rotation estimation [108], is the statistical practice of partitioning a sample of data into subsets such that the analysis is initially performed on a single subset, while the other subsets are retained for subsequent use in confirming and validating the initial analysis. The initial subset of data is called the training set; the other subsets are called validation or testing sets. Common types of cross-validation are: Holdout validation, K-fold cross-validation and Leave-one-out cross validation.

Holdout validation is not cross-validation in the common sense, because the data is never crossed over. Observations are chosen randomly from the initial sample to form the validation data, and the remaining observations are retained as the training data. Normally, less than a third of the initial sample is used for validation data.

K-fold cross-validation, partitions the original sample into K subsamples. A single subsample of the K subsamples is retained as the validation data for testing the model. The remaining K – 1 subsamples are used as training data. The cross-validation process is then repeated K times, with each of the K subsamples used exactly once as the validation data. The K results from the folds can then be averaged to produce a single estimation.

Leave-one-out cross-validation involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. This is the same as a K-fold cross-validation with K being equal to the number of observations in the original sample.

7.2.3. T-test

A t-test is any statistical hypothesis test in which the test statistic has a Student's t distribution if the null hypothesis is true. Among the most frequently used t-tests is a test of the null hypothesis that the means of two normally distributed populations are equal. Given two data sets, each characterized by its mean, standard deviation and number of data points; we can use some kind of t-test to determine whether the means are distinct, provided that the underlying distributions can be assumed to be normal. The t statistic is given by the following equation:

$$t = \frac{(\bar{x} - \bar{y})}{\sqrt{\frac{(\sigma_x^2 - \sigma_y^2)}{n}}} \qquad (71)$$

where $\bar{x}$ and $\bar{y}$ are the means of the two populations, σ_x^2 and σ_y^2 are the variances of the two populations, n is the size of each population, and $\sqrt{\frac{(\sigma_x^2 - \sigma_y^2)}{n}}$ is called the standard error of the difference between the two means. For significance testing, the number of degrees of freedom for this test is $2n - 2$.

The t-test can be used to compare different classifiers by giving a confidence bound of the hypothesis that one classifier is better than the other. K datasets are drawn from the problem domain, and cross validation is used with each classifier to predict the error (or success) rates for the two classifiers $(x_1 ... x_k, y_1 ... y_k)$. The difference between the two corresponding sets $(d_i = x_i - y_i)$ is modeled as a t-statistic with zero-mean, unit-variance, and $K - 1$ degrees of freedom. The null hypothesis $(d = 0)$ is tested for a certain confidence level (1% or 5%) to check if the actual difference exceeds the confidence limit. In this case the two classifiers will be different; otherwise one is better than the other. With a limited dataset,

the corrected resampled t-test is used instead. Either hold-out or cross-validation method is used with randomized allocations of the same dataset [106].

7.3. Data Preprocessing Evaluation

Four feature types were proposed in the previous chapter: raw profiles, Fourier descriptors, wavelet descriptors, and moment descriptors of the profiles. We proposed different normalization methods to be applied to the profiles extracted from the unwrapped color Doppler cineloops. The sample size-invariance normalization is straightforward and necessary for all feature types to allow similarity comparisons. The shift-invariance normalization is mainly necessary for the raw feature profiles and the wavelet descriptors, since the shift-invariant wavelet descriptors are difficult to achieve and still an active research area. However the Fourier and moment descriptors can also benefit from the shift invariance preprocessing step and can be used directly without normalization in the transform domain. Two scale-invariant methods were proposed in section 6.7.3, the lost-sign and preserved-sign scaling methods. The lost-sign scaling method is a classical one in the recognition literature, while the preserved-sign method was invented specifically for the vessel recognition problem. We claimed in the previous chapter that the preserved-sign scaling should have advantages over the classical lost-sign scaling method, however we need to confirm it with the classifier performance.

To evaluate the different preprocessing methods, four configurations of the scaling methods and the descriptor types were tested as follows:

1. No scaling is applied to the profiles and variant descriptors are used; the raw transform coefficients are used without any processing. This case is valid for the training cases where fixed amplitude was used for each waveform for all instances, while the Doppler steering angle changes caused slight amplitude change. However eventually we chose not to rely on the amplitude of the vessel waveform, due to the high variation among different humans and among disease states.

2. No scaling is applied to the profiles and invariant descriptors are used. The invariant descriptors are obtained from the raw transform attributes as described in the previous chapter. This case is very useful to avoid relying on the spatial domain normalization methods as much as possible. However scale invariance is not applied to the moment descriptors since the training data had almost fixed amplitude for each waveform type.
3. Lost-sign scaling method is applied to the profiles, while variant descriptors are used.
4. Preserved-sign scaling method is applied to the profiles, while variant descriptors are used.

The four test cases described above are applied to each of the feature types with an arbitrary number of coefficients (or attributes). We used all attributes of the raw normalized profiles, 32 attributes; we used 5 Fourier descriptors, the 16 approximation wavelet descriptors, and 5 moment descriptors. We had 30 instances of each waveform. The Naïve Bayes classifier is used with equal probability for all classes. Cross-validation method is used to evaluate the classification performance. Table 7.1 shows the classification success rate for the different scaling methods for each feature type. Figure 7.3 plots the data in Table 7.1 in a graphical representation.

Table 7.1 Classification success rates of different preprocessing methods.

	No scaling/ variant descriptors	No scaling / invariant Descriptors	Lost-sign scaling/ variant descriptors	Preserved-sign scaling/ variant descriptors
Raw (32)	96.0%	NA	90.0%	97.3%
FD (5)	73.3%	77.1%	85.3%	94.6%
WD (16)	96.6%	97.3%	86.6%	98%
MD (5)	60.6%	68%	60.6%	95.3%

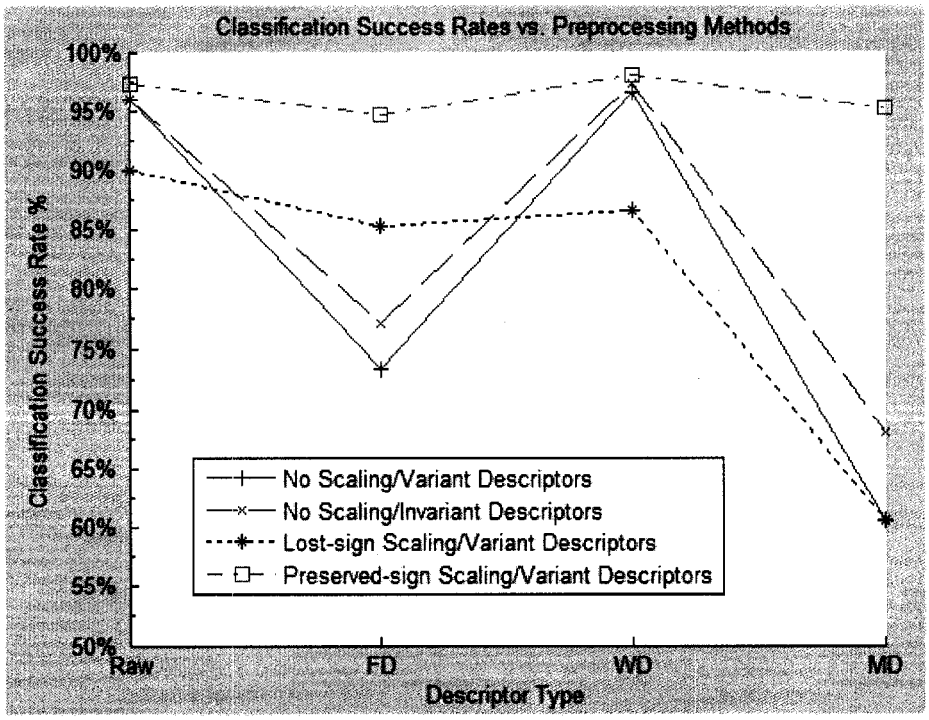


Figure 7.3 Classification success rates of different preprocessing methods.

The following conclusions can be drawn from the success rates in Table 7.1 and Figure 7.3:

- The preserved-sign scaling method is superior to all other scaling methods for all descriptor types. This is aligned with our expectations set in the previous chapter, when we discussed the properties of this scaling method, specifically how it preserves the phasicity of the underlying waveform, which gets lost with either the invariant descriptors or lost-sign scaling methods.
- Applying invariant descriptors on the non-scaled data slightly enhanced the recognition rates. This is due to the scale variation in the training data, because of varying the Doppler steering angle on the sample-amplitude waveform.
- Both the raw profile coefficients and wavelet descriptors show better performance than the Fourier and moment descriptors. However it is still too early to draw such a conclusion due to the limited size of the training set.

Moving forward with the recognition system evaluation, we chose to use the preserved-sign scaling method with the raw transform coefficients as descriptors, since the performance gain was noticeable with this scaling method.

The moment descriptors scaling methods, the linear-scaling and log-scaling methods, were briefly evaluated using the Naïve Bayes classifier with cross validation and preserved-scaling preprocessing for all 10 moment descriptors. The success rates were very comparable (92.3% for the log-scaling method and 93% for the linear-scaling method). This is due to the preserved-sign scaling step which constrains the moment coefficients to lower ranges and alleviate the need for a log scaled step. For this reason we decided to use the linear-scaling method for the moment descriptors for further experiments.

7.4. Test Data Acquisition

The preliminary success rates listed in Table 7.1 were very promising, especially for the preserved-sign scaling method. The high success rates for all descriptor types confirmed the validity of our approach of the vessel recognition problem. However the training set size was somehow limited and further evaluations of the recognition system would require larger dataset size and independent test data.

We have acquired two additional test datasets (TEST1 & TEST2), in addition to the original training set (TRAINING1). The first test set was extracted from the same acquisition data used for the training set. Each color Doppler cineloop of 60 frames contained more than one cycle; only the first cycle or pitch was extracted and added to the training set. The remaining cycles in each loop were extracted and added to the first test set. To extract as many cycles as possible, the cycles were allowed to overlap by half cycle length. Figure 7.4 shows an example of extracting the test pitches from a femoral waveform; the left figure shows the original 60-frame profile and the right figure shows the six pitches extracted and scaled from the profile with a half-pitch overlap, excluding the first pitch which is used for the training set.

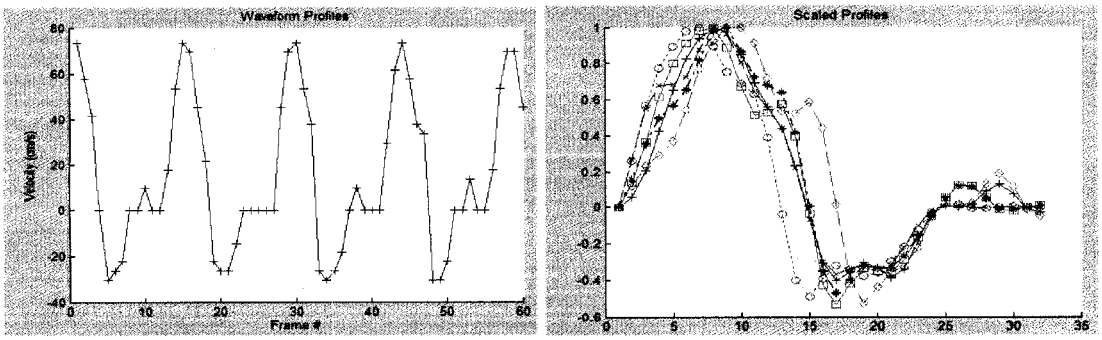
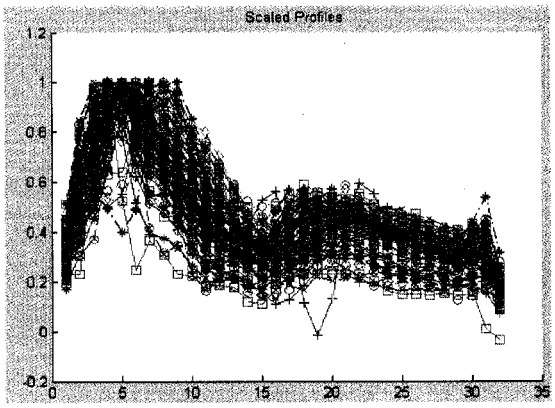


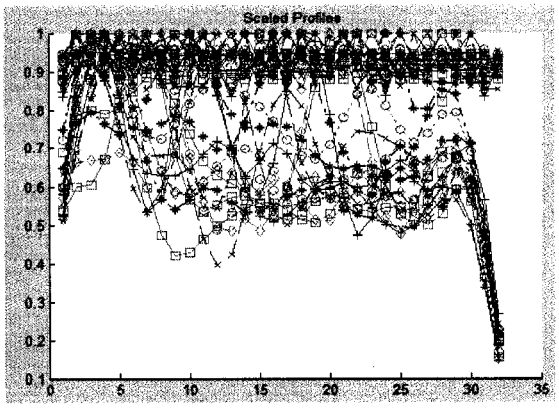
Figure 7.4 TEST1 dataset extraction. Left: A 60-frame femoral waveform. Right: all 6 extracted (and scaled) pitches with half-pitch overlap, excluding the first pitch.

Figure 7.5 shows all the test profiles extracted from the original training acquisitions for the different waveforms. It should be noted that there is much more variation among the different cases of the same waveform due to preprocessing failures or noisy data, especially for the constant waveforms. It is also important to mention that TEST1 dataset is different than the training set (TRAINING1), however the two sets are not completely independent since they share the same acquisition run. The cycle overlap in the test dataset also minimizes the degree of independence among the different test cases. The total number of case in the TEST1 set is 586 instances compared to the 150 instances in the TRAINING1 set.

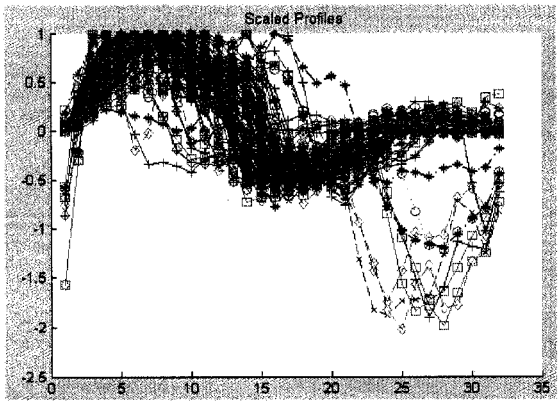
The second test set (TEST2) was constructed using a completely new acquisition run, but still following the same protocol described in section 6.3. The only deviation from the protocol is to relax the amplitude condition and vary the signal amplitude using the Doppler phantom control software. This change was necessary since we concluded from the previous preprocessing analysis that the preserved-sign scaling method is the best choice that achieves the highest recognition rates. Hence the fixed amplitude assumption is not necessary for our application, and relaxing that condition will also benefit the expected variation from human to human. Different degrees of aliasing were included for each dataset to exercise the unwrapping algorithms as well.



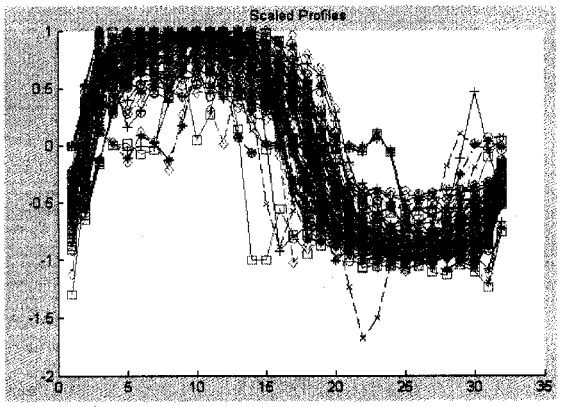
Carotid waveforms



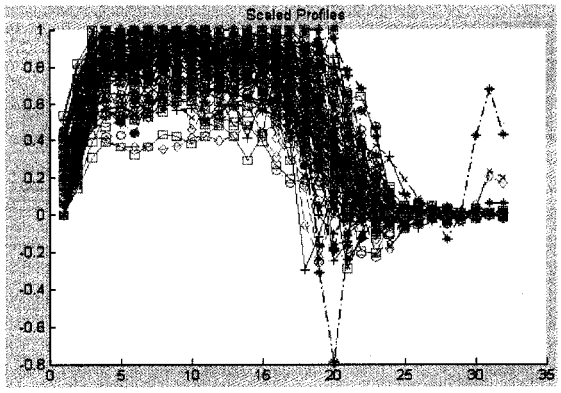
Constant waveforms



Femoral waveforms



Sine waveforms

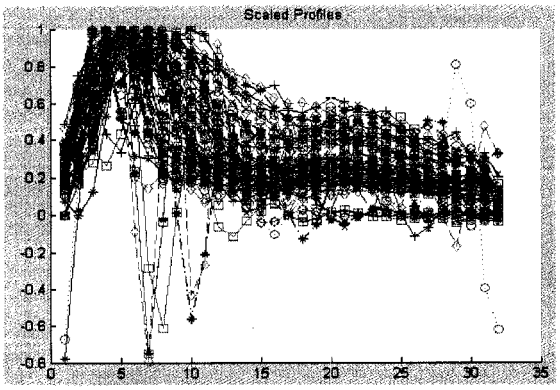


Square waveforms

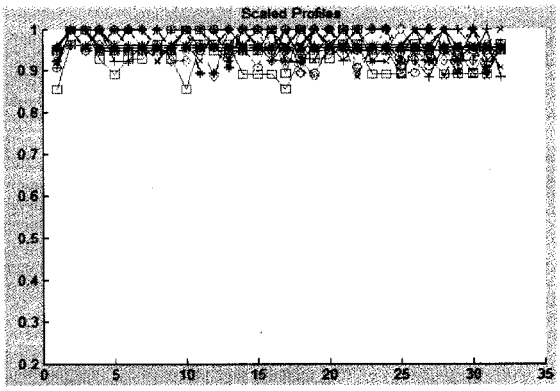
Figure 7.5 TEST1 dataset for the different waveforms.

Figure 7.6 shows the scaled pitches extracted from the test set acquisition run for the different waveforms. There are many noisy datasets in each waveform but the different

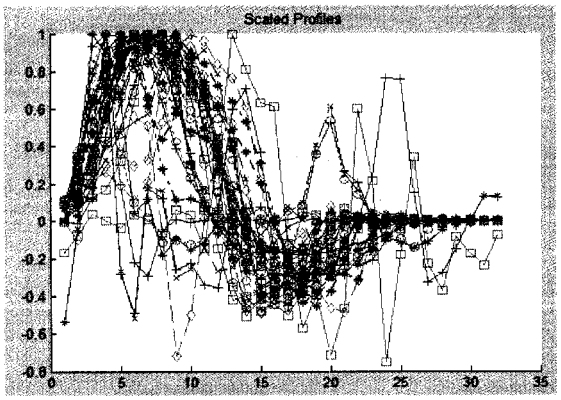
amplitude waveforms mapped to the same scale after applying the preserved-sign scaling method. The total number of cases in the TEST2 set is 330 instances.



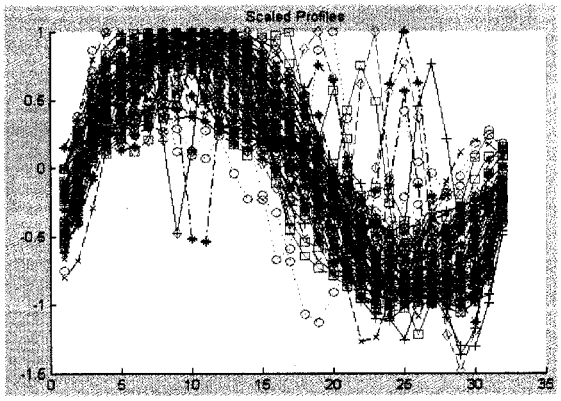
Carotid waveforms



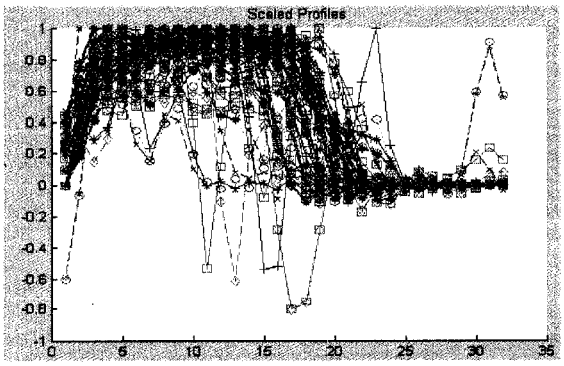
Constant waveforms



Femoral waveforms



Sine waveforms



Square waveforms

Figure 7.6 TEST2 dataset for the different waveforms.

7.5. Attribute Selection Evaluation

The next critical question to answer is how many attributes should be used for each descriptor type. In the literature the number of transform-based attributes is typically selected either randomly without justification or based on the understanding of the physical meaning of the descriptors. For example the lower order attributes of the Fourier descriptors are usually retained, while the high order ones that correspond to higher frequencies are excluded due to their high noise content. Chen et al. [95] chose the first ten Fourier descriptors for character recognition, since the remaining attributes are closer to zero. Similarly the lower scale wavelet attributes are usually used, while the higher scale ones are discarded; they carry little information about the underlying objects. Chang and Kuo [96] used six-level wavelet decomposition with four coefficients at the coarsest scale, and they used only the four coarsest, or approximation, coefficients as descriptors for character recognition. Mertzois et al. [102] used four moment descriptors for shape recognition.

In general, there are many attribute selection methods in the literature, which can be broadly classified into two main categories. The first category methods use a specific classifier to evaluate the performance against each subset of the attributes and select the subset that gives the highest success rates. The second category methods work independently from any classifier and search the attribute space to find the attribute combination that has the optimal discriminating power. We investigated the two categories for our vessel recognition problem.

7.5.1. Classifier Performance-based Attribute Selection

For our vessel recognition application we investigated the optimal number of attributes that would achieve the highest recognition performance. Due to the nature of our attributes, which represent either the raw profiles or the transformed coefficients that pack the information in the lower order attributes, we chose to consider only adjacent attributes starting from the first one. Other recognition problems may consider all combinations of their attributes, but this is not necessary for our problem. The experiment we conducted is to

evaluate the classification success rate while incrementing the number of attributes. We again used the Naïve Bayes classifier and 10-folds cross validation method on the preserved-sign scaled data to evaluate the results. The experiment was conducted on both the Fourier and moment descriptors, but not the wavelet ones. This is due to the spatial locality property of the wavelet descriptors, which requires including at least all the approximation coefficients to fully represent the underlying waveform. Figure 7.7 shows the plot of the classification success rate versus the number of attributes for the Fourier, wavelet and moment descriptors.

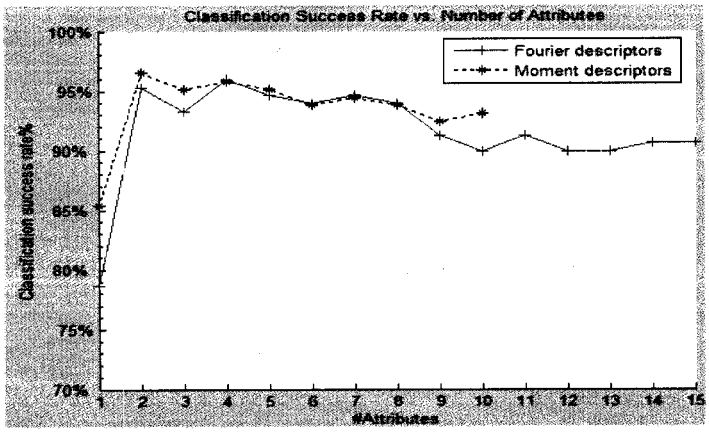


Figure 7.7 Classification success rates vs. number of attributes for the Fourier and moment descriptors.

The following conclusions can be drawn from Figure 7.7:

- The optimal number of attributes of the Fourier descriptors lies between 2 and 8. Including the higher-order Fourier attributes (9-15) slightly degrades the classification performance. This is expected due to the little information content of the high-order Fourier coefficients and the high degree of noise they include.
- The success rate is almost stable with the number of moment attributes between 2 and 10.

7.5.2. Discrimination Power-based Attribute Selection

We used WEKA's attribute selection toolkit for our second attribute selection experiment, which is independent of any classifier. In WEKA we specify an attribute

evaluator and a search method. The evaluator determines what method is used to assign a discriminating power weight to each subset of attributes. The search method determines what style of search is performed. The provided dataset can be used in two ways to evaluate the attributes. The discriminating power weight of the attribute subset can be determined using the full set of training data, or cross-validation can be used, in which case the discriminating power weight of the attribute subset is determined for each cross-validation fold. The output will represent the number of folds (or percentage) where the attribute was selected. For example using 10-fold cross-validation, the output for a certain attribute will look like 6 or 60% which means that attribute is selected 6 out of 10 times as a discriminating feature. Figure 7.8 shows a screen shot of WEKA's attribute-selection tool with an example of Fourier descriptor evaluation.

We applied the cross-validation method with the "Best First" greedy search method for attribute subset selection for all four descriptor types: raw, Fourier, wavelet, and moment descriptors. The attribute evaluator method used was the "Subset Evaluation" method, which evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. We repeated the experiment for three different datasets (TRAINING1+TEST1, TEST2, TRAINING1+TEST1+TEST2). Table 7.2 shows the results of the three experiments with the selected attributes listed for each descriptor type. The criterion for considering an attribute was that it was selected 50% or 5 times out of 10 cross-validation runs.

The following observations can be drawn from the results of Table 7.2:

- The general observation is that the attribute selection varies from one dataset to another. However there are some commonalities between the selected attributes in the three experiments.
- For the raw profile descriptors, a large number of attributes is needed to achieve the best classification rates.
- For the Fourier descriptors, the low-order descriptors are consistently selected by the three experiments. Very few high-order Fourier descriptors are randomly selected.

This conclusion agrees with the results of our first attribute evaluation method described in the previous section.

- For the wavelet descriptors, a large number of descriptors are also needed to achieve better classification rates. A large number of the wavelet approximation coefficients are consistently chosen by the three experiments. Also, a smaller number of the wavelet detail coefficients were also selected by the three experiments.
- For the moment descriptors, a few attributes were consistently chosen by the three experiments (1, 3, 5, and 7).

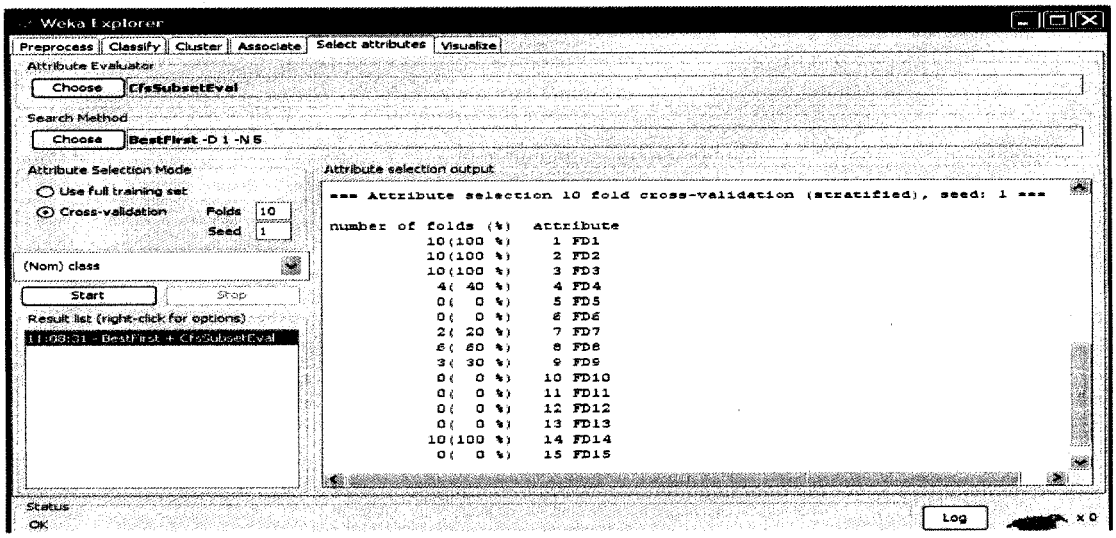


Figure 7.8 WEKA's attribute selection tool. An example of attribute selection for the Fourier descriptors of all training and test sets combined.

Table 7.2 Attribute selection results for the different descriptor types.

	TRAINING1+TEST1	TEST2	TRAINING1+TEST1+TEST2
Raw (32)	1,4,12,14-20, 22-31	2,5,6,8,11,13,16-18, 22-25, 27-29, 31	1,5,7,12,14-20, 22-30
FDs (15)	1-4, 7, 9, 14	1-3	1-3,8,14
WDs (32)	1, 7-16, 19-21, 23-27	1, 3, 4, 7-15, 17, 20, 25, 26	1,7-15,19-21, 23-28,
MDs (10)	1,3,4,5,7,9	1,3,5,7,8	1,3,5,7,9,10

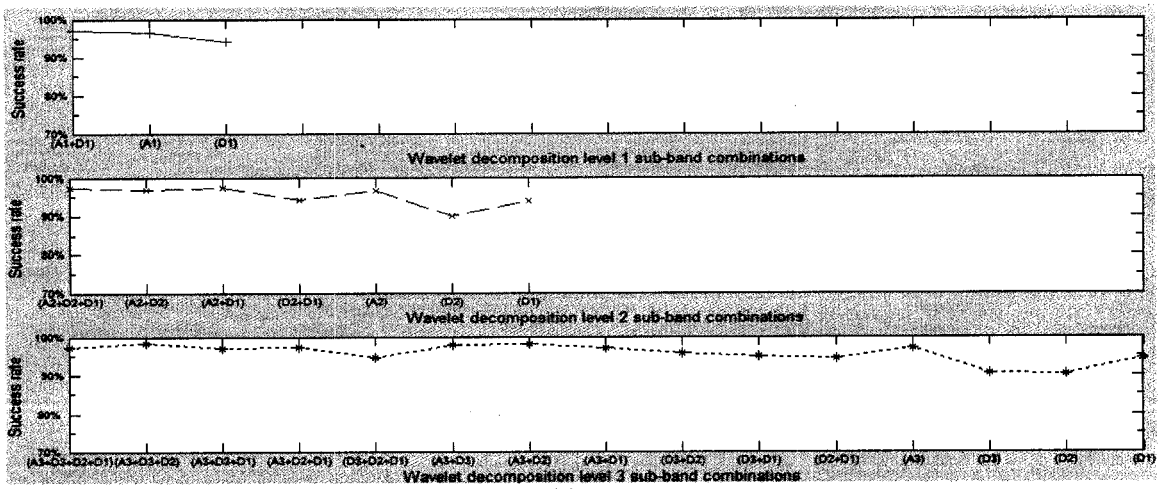
7.5.3. Wavelet Descriptor Attribute Selection

For the wavelet descriptors, a separate experiment was conducted to explore the attribute selection for the different wavelet decomposition levels. The recognition success rates were evaluated for all combinations of wavelet subbands for three decomposition levels; the Naïve Bayes Simple classifier and 10-fold cross validation were used. Figure 7.10 shows the successs rates for all combinations of the wavelet subbands for three decomposition levels. The experiment was run twice (for TRAINING1 and TEST1 sets combined and for TEST2 set). In the figure, Ax and Dx represent the approximation and detail subbands of the wavelet decomposition level x, respectively.

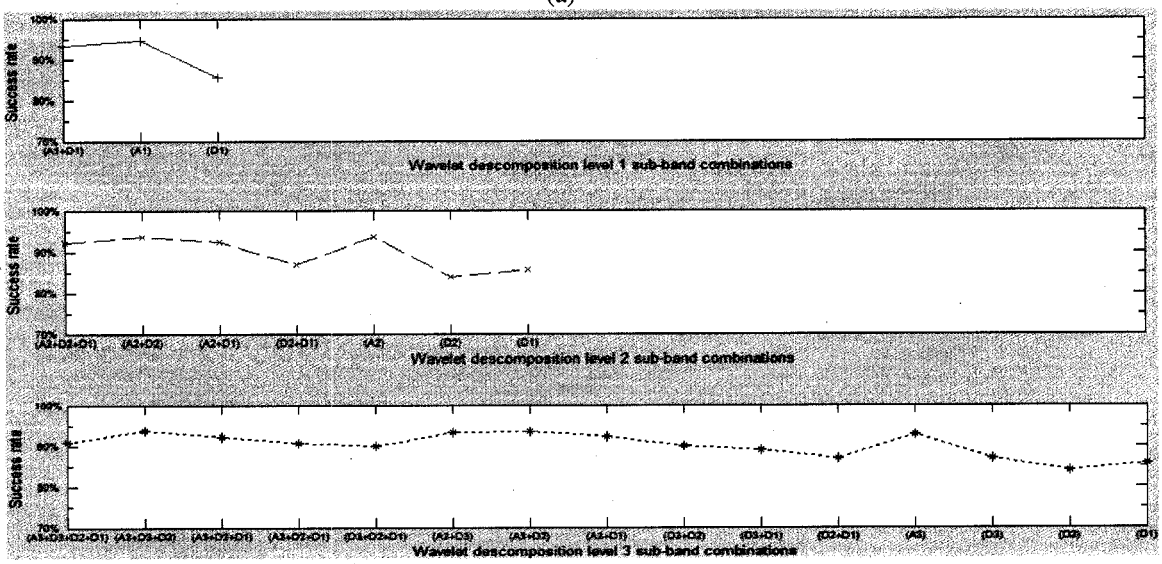
Based on the results shown in Figure 7.9, the following conclusions can be drawn:

- The exclusion of the approximation coefficients degrades the classifier performance. This is due to the fact that the approximation coefficients capture the global (low-frequency) features of the underlying waveforms.
- Some wavelet subband combinations achieve very high recognition rates. However the results do not promote a single wavelet decomposition nor a single subband combination conclusively.
- The approximation coefficients of a single level wavelet decomposition (A1) is a reasonable choice for the forthcoming experiments.

Based on the results drawn from all attribute selection methods and the fact that the waveforms we considered only represent a subset of all possible blood flow waveforms, we decided to use a fixed number of adjacent attributes for each descriptor type for further evaluations of the recognition system. We used all 32 raw descriptors, the first 5 Fourier descriptors, the 16 wavelet approximation descriptors, and the first 5 moment descriptors.



(a)



(b)

Figure 7.9 Recognition success rates for all combinations of the wavelet subbands for 3 decomposition levels. (a) Success rates for TRAINING1 and TEST1 sets. (b) Successs rates for TEST2 set.

7.6. Classifier Evaluation

The experiments conducted so far suggested the optimal preprocessing method and the optimal number of attributes for each descriptor type. The classification success rates of the preserved-sign scaling methods listed in Table 7.1 and obtained using the Naïve Bayes classifier with the training dataset (TRAINING1) were very promising for all descriptor types. However we need to confirm these results with independent test cases. In this section we will

evaluate the classification performance of the same Naïve Bayes classifier with the test cases described in section 7.4.

During this experiment we used one of the three datasets (TRAINING1, TEST1, and TEST2) as a training set while the other two are used as testing sets. 10-fold cross validation was used to obtain the classification success rates. The number of attributes of each descriptor type decided in the previous section was used: 32 Raw attributes, 5 Fourier descriptors, 16 wavelet descriptors, and 5 moment descriptors. Table 7.3, Table 7.4, and Table 7.5 list the classification rates of the three experiments. The following conclusions can be drawn from the listed rates:

- The raw and wavelet descriptors were consistently superior to the Fourier and moment descriptors, in agreement with the results of Table 7.1. This is most probably due to the locality property of both the raw and wavelet descriptors, which preserves the discriminating features among the different waveforms.
- The wavelet descriptors slightly enhanced the performance of the classifier over the raw descriptors in many cases. This is most probably due to reducing the noise level by discarding the detail coefficients while preserving the local information of the underlying waveforms.
- The success rates of both TRAINING1 and TEST1 sets were very comparable in the three experiments. This is due to the fact that they both share the same acquisition set and thus they are not completely independent.
- The success rates of TEST2 when either TRAINING1 or TEST1 set was used for training were not satisfactory in general, while the success rates of both TRAINING1 and TEST1 sets when TEST2 set was used for training were more acceptable and comparable to their own numbers. This indicates that the classifier model built using TEST2 set is more versatile and applicable than the models built using TRAINING1 or TEST1 set. This is most probably due to the fact that the acquisition process of TEST2 relaxed the fixed-amplitude condition and it offers more generic representation of the problem domain. Another reason is the existence of many failed cases in TEST1 dataset, especially in the

constant and femoral waveforms, as shown in Figure 7.6. These noisy cases will influence the designed classifier model and degrades its performance with other test cases.

Table 7.3 Classification success rates for using TRAINING1 set as a training set while TEST1 and TEST2 sets are used as test sets.

	TRAINING1	TEST1	TEST2
Raw (32)	97.3%	94.4%	87%
FD (5)	94.6%	91.1%	67.3%
WD (16)	98%	94.9%	90.3%
MD (5)	95.3%	93.7%	69.7%

Table 7.4 Classification success rates for using TEST1 set as a training set while TRAINING1 and TEST2 sets are used as test sets.

	TEST1	TRAINING1	TEST2
Raw (32)	96.8%	95.3%	70.9%
FD (5)	95.9%	92.7%	69.7%
WD (16)	96.4%	95.3%	71.2%
MD (5)	93%	91.3%	65.8%

Table 7.5 Classification success rates for using TEST2 set as a training set while TRAINING1 and TEST1 sets are used as test sets.

	TEST2	TRAINING1	TEST1
Raw (32)	94.2%	94.6%	92%
FD (5)	81.8%	90%	89%
WD (16)	94.5%	94.7%	91.5%
MD (5)	88.5%	83.3%	80.5%

7.7. Classifiers Comparison

The next evaluation step of our vessel recognition system is to compare several well-known classifiers and find out which one of them yields the best recognition rate. We used the WEKA's experiment environment to evaluate 10 different classifiers on the four different descriptor types. The training (TRAINING1) and test datasets (TEST1 and TEST2) were combined together to form a large evaluation set for each descriptor type. 10 iterations were run with a random seeded 10-fold cross-validation applied for each run. Figure 7.10 shows a screen shot of WEKA's experiment environment with a setup to evaluate 10 classifiers for the 4 descriptor types.

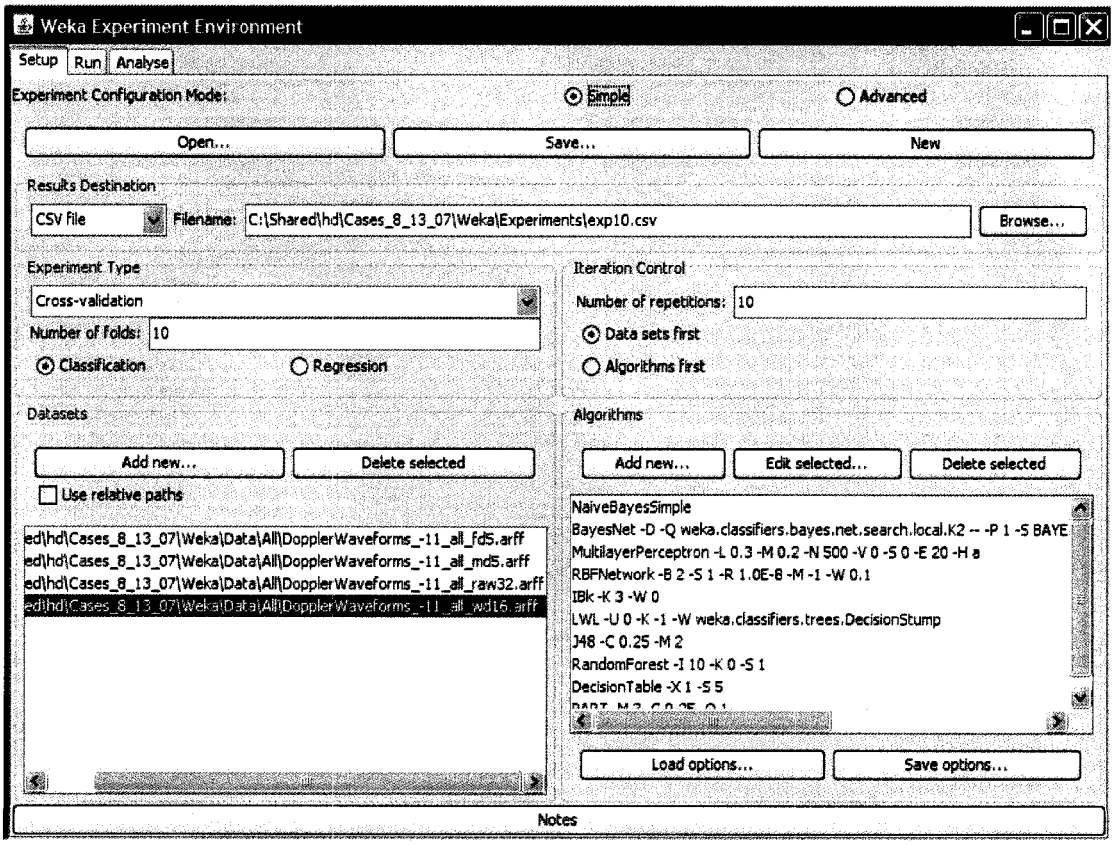


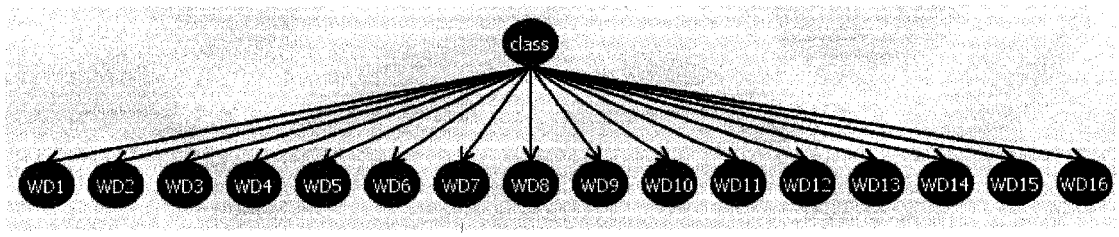
Figure 7.10 WEKA's experiment environment. A classifier comparison experiment is setup with four descriptor types and ten classifiers.

The ten evaluated classifiers belong to five different classifier types: Bayesian classifiers, function classifiers, lazy classifiers, tree classifiers, and rule-based classifiers. Bayesian classifiers rely on Bayes decision theory and try to minimize the classification error probability. Function classifiers are described by linear or nonlinear discriminant functions that separate the different classes and do not assume any probabilistic distribution of the underlying training data. The lazy or instance-based classifiers do not build any model from the training data, but use a similarity measure to compare an unknown pattern against all training patterns and assign the unknown pattern to the majority class of the x nearest training patterns, where x is set according to the problem domain. Tree classifiers build a nonlinear multistage decision system in which classes are sequentially rejected until a finally accepted class is reached. Rule-based classifiers build a set of rules to test the attribute values and assign a class based on the test outcomes. The ten evaluated classifiers are:

1. The Naïve Bayes Simple (NBS) classifier [107]
2. The Bayesian Networks (BN) classifier [106]
3. The Multilayer Perceptron Neural Network (MLPs) classifier [106]
4. The Radial Basis Function Network (RBF) classifier [106]
5. The k-Nearest Neighbor (KNN) classifier, a.k.a IBK classifier within WEKA's environment [109]
6. The Locally-weighted Learning (LWL) classifier in conjunction with the Decision Stump classifier [106]
7. The C4.5 Decision Tree (C4.5) classifier, a.k.a J48 within WEKA's environment [110]
8. The Random Forest (RF) classifier [111]
9. The Decision Table (DT) classifier [112]
10. The PART Decision List (PART) classifier [113]

The Naïve Bayes classifier, which was described previously in section 7.2.1, is used in its simplest form in which feature independence and normal distributions are assumed for all attributes. Bayesian networks represent a major enhancement to the classical Bayes classifier; they can represent probability distributions concisely and comprehensibly in a graphical manner. A Bayesian network is a directed acyclic graph with each node representing a single attribute. Some advanced Bayesian networks can also derive new attributes from the existing

ones and create hidden nodes for them, but we will not consider this level of complexity for our problem. Each node of the network contains a table of conditional probabilities of the specific attribute it represents given all combinations of its parent nodes. The root node, which represents the class attribute or the underlying vessel waveform in our case, contains the *a priori* probability of each class. Bayesian networks assume *conditional independence* which implies that each node is conditionally independent of grandparents and all higher ancestors given its parents. Figure 7.11 shows an example of a learned Bayesian network for the approximation wavelet descriptors with all the training and test datasets combined using 10-fold cross validation. The top part of the figure shows a graphical representation of the network, while the bottom part shows one example of the conditional probability tables of the first wavelet descriptor. The resultant network has all the edges extending from the class attribute to all the wavelet attributes due to the default maximum number of parent nodes used is equal to one. If we use 3 for the maximum number of parents with the moment descriptors, we get the Bayesian network shown in Figure 7.12.



class	'(-inf-0.0...	'(0.0333...	'(0.1521...	'(0.3629...	'(0.80213...	'(0.9680...
carotid	0.013	0.017	0.132	0.776	0.059	0.002
constant	0.005	0.005	0.005	0.005	0.071	0.907
femoral	0.094	0.535	0.352	0.015	0.002	0.002
sine	0.776	0.046	0.173	0.002	0.002	0.002
square	0.011	0.091	0.411	0.395	0.091	0.002

Figure 7.11 A Bayesian network for the approximation wavelet descriptors with maximum no. of parents=1 for the K2 algorithm. Top: the network graphical representation. Bottom: the conditional probability of the first wavelet descriptor.

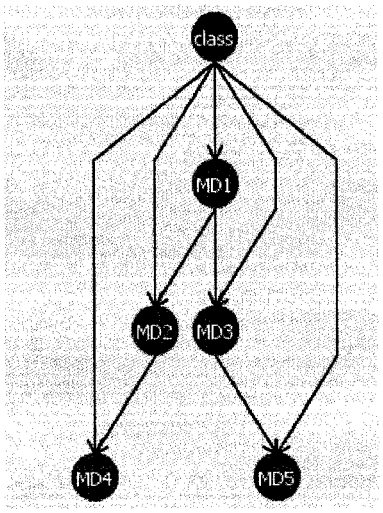


Figure 7.12 A Bayesian network for the moment descriptors with maximum no. of parents=3 for the K2 algorithm.

There are many algorithms for constructing and learning Bayes network; a simple and fast one, which was introduced by Cooper and Herskovits [114], is called K2 . The K2 algorithm is a greedy one which starts with a given ordering of the attributes, with the class attribute in the beginning, and considers adding edges from previously processed nodes to the current one. In each step it adds the edge that maximizes the joint probability of each training instance.

Multilayer perceptron neural networks (MLPs) and radial basis function networks (RBFN) are nonlinear classifiers that develop functions that separate nonlinear class boundaries. The building block of a multilayer perceptron neural network is called a *perceptron*. A perceptron represents a hyperplane in instance space that linearly separates the classes. The perceptron can be represented as a graph with input and output layer of nodes and weighted connections. The input layer has a node for each attribute plus an extra node that is always set to one to represent the constant term of the hyperplane equation. The output layer consists of one node that is connected to all input nodes. A multilayer perceptron neural network consists of a number of interconnected neurons or perceptrons arranged in layers as input, output and hidden layers. The structure of the network typically solved through experimentation, while the connection weights are learned via a well-known algorithm called

backpropagation, the details of which are described in [106]. Figure 7.13 shows the layout of a multilayer perceptron neural network used to classify the vessel waveforms based on the Fourier descriptors.

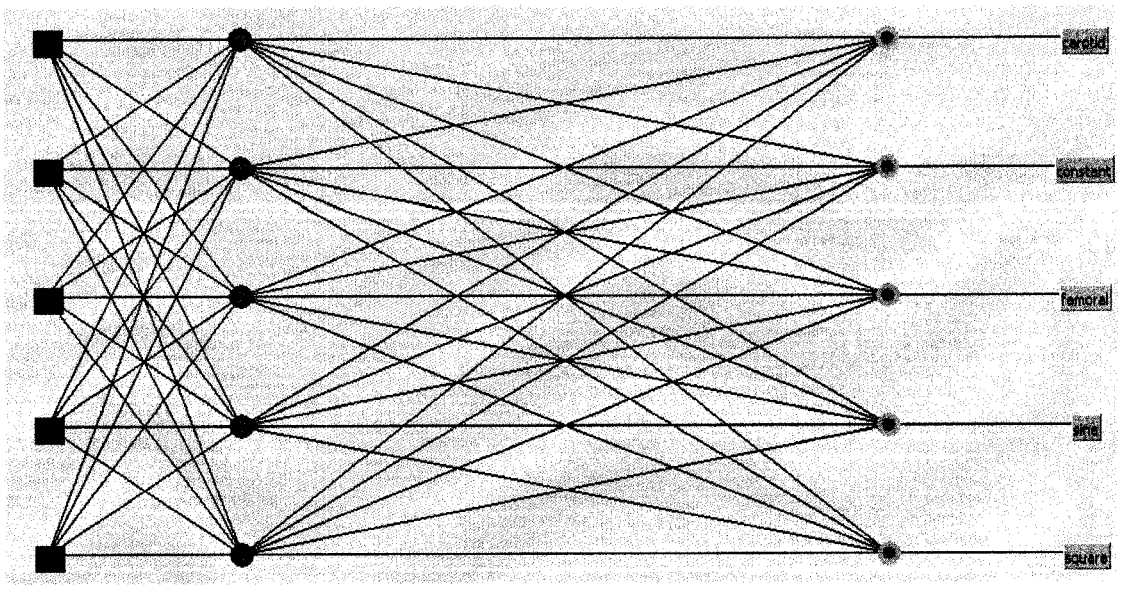


Figure 7.13 A multilayer perceptron neural network for the Fourier descriptors with an input layer that represents the attributes, one hidden layer, and an output layer that represents the vessel classes.

The Radial basis function networks differ from MLPs in the way that hidden nodes perform computations. In RBFN, each hidden node represents a particular point in input space and uses a Gaussian activation function that constitutes a hypersphere or hyperellipsoid in instance space, instead of a hyperplane in case of MLPs.

The K nearest neighbor classifier is a classical instance-based learning algorithm. In our case, Euclidean distance is used with attribute scale normalization to the 0-1 range. The nearest neighbor algorithm is simple but could be slow if the training data is large. One remedy is to try to reduce the number of exemplars and reduce redundancy. Another alternative is to use efficient tree structures, such as KD-trees and Ball trees, that are used to speed up the comparison process of finding the nearest neighbor [106]. Other drawbacks of the simple nearest neighbor approach are the negative effects of noisy instances and irrelevant

attributes. However, there are pruning techniques to discard noisy exemplars, and attribute weighting schemes to reduce the influence of irrelevant exemplars [106].

The concept of locally weighted learning generates local models at classification time, not training time, by weighing the training instances according to their distance to the test instance. This concept is used for enhancing linear regression models as well as enhancing classifiers that can handle weighted instances. For example, locally weighted Naïve Bayes proved to outperform both Naïve Bayes and the K-nearest neighbor techniques [106]. This is mainly due to relaxing the intrinsic independence assumption of the classical Bayes classifier. The locally weighted Bayes classifier assumes only independence within a neighborhood of the test instance. We decided to apply the locally-weighted training concept to a simple tree classifier, the Decision Stump classifier, that consists of only a single depth to see how much enhancement can be achieved by the locally-weighted training approach. The decision stump classifier uses a single feature for classification.

The popular decision tree algorithm C4.5 was developed by Quinlan and fully explained in his book [110]. The decision tree is constructed using the training set to select an attribute to place at the root node and an evaluation rule for this attribute that will split the training set into two subsets. The process is then repeated recursively for each subset by selecting other attributes to branch on. The branching process ideally stops when all instances in the branch belong to the same class. The criterion for selecting attributes to branch on is to increase the purity of the developed branches, or in other words maximize the information gain, or entropy, of the resultant tree to classify a newly seen instance. There are many tree pruning algorithms that attempt to simplify the original complete tree that is usually overfitted to the training data and does not generalize well to independent test sets [106]. Figure 7.19 shows an example of C4.5 decision tree for the raw descriptor type; 13 out of the 32 attributes were used to construct the tree.

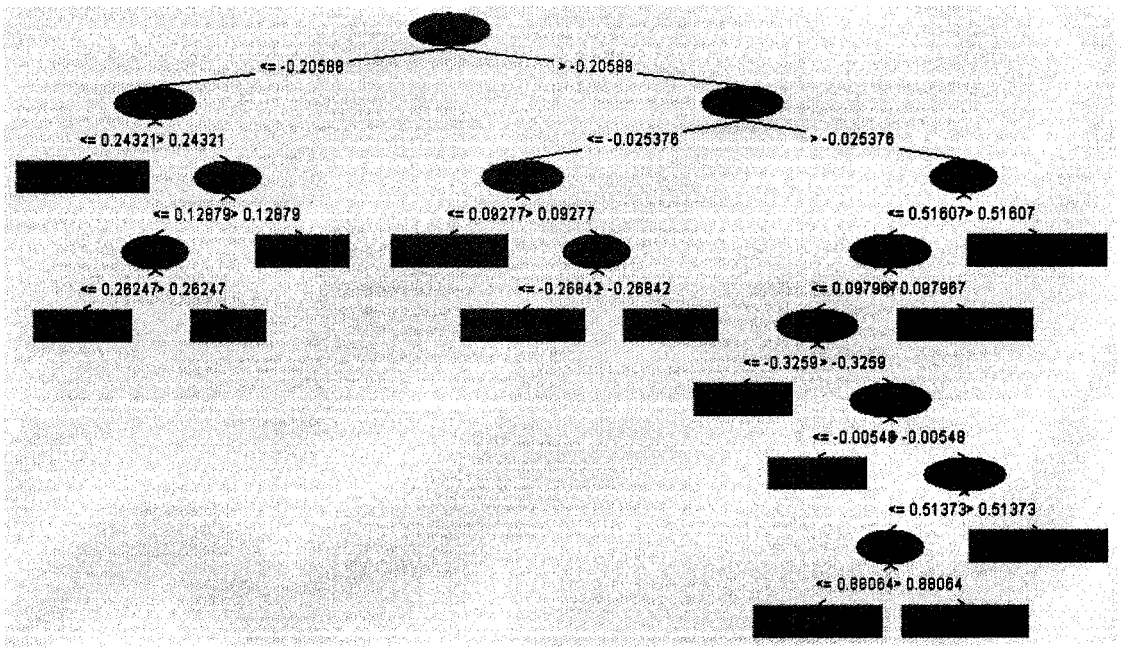


Figure 7.14 A C4.5 decision tree classifier for the raw descriptors.

The concepts of combining several models, such as bagging, randomization, and boosting, have appeared in the literature and they seem to greatly enhance the performance of a single model. Bagging exploits the instability inherent in learning algorithms by generating different datasets from the original training set and building different models. The new datasets are generated through sampling with replacement from the original training set. A new test instance is classified using the different models and a majority vote decides the final class of that test case.

Randomization achieves different models through the randomization of the algorithm parameters, not the training data. For example, a decision tree can be randomized by randomly selecting one of the N best options at each node instead of the absolute best option as was the case with the C4.5 decision tree. Boosting techniques combine different complementary models. Boosting is an iterative process where new models are influenced by the performance of those models built previously. Unlike bagging, which uses equal weights, boosting voting is weighted.

The Random Forest classifier is an example of a classifier that combines multiple models through the application of two powerful concepts, bagging and randomization. The classifier builds a randomized decision tree in each iteration of the bagging algorithm. The randomized tree chooses a test based on a given number of random features at each node.

The Decision Table classifier builds a decision table whose columns represent a set of selected attributes and whose rows represent tests over the attributes and the assigned class label. The classifier evaluates feature subsets using "Best First" search algorithm and cross-validation for evaluation. Given a new test instance, the classifier searches for matching rows of the instance attribute values. If no instances are found, the majority class is returned; otherwise, the majority class of all matching rows is returned. Figure 7.15 shows a snapshot of a decision table for the wavelet descriptors. The classifier selected only two wavelet descriptors: WD9 and WD13.

Rules:

WD9	WD13	class
'(0.21473-0.64863]'	'(0.692205-inf)'	sine
'(0.64863-0.81565]'	'(0.692205-inf)'	constant
'(0.81565-1.22915]'	'(0.692205-inf)'	constant
'(1.22915-inf)'	'(0.692205-inf)'	constant
'(0.81565-1.22915]'	'(0.253965-0.692205]'	carotid
'(-0.09472-0.21473]'	'(0.253965-0.692205]'	carotid
'(-0.915935--0.311715]'	'(0.253965-0.692205]'	femoral
'(0.64863-0.81565]'	'(0.253965-0.692205]'	carotid
'(0.21473-0.64863]'	'(0.253965-0.692205]'	carotid
'(-0.09472-0.21473]'	'(0.15774-0.253965]'	carotid
'(1.22915-inf)'	'(0.15774-0.253965]'	square
'(-0.915935--0.311715]'	'(0.15774-0.253965]'	femoral
'(0.21473-0.64863]'	'(0.15774-0.253965]'	carotid
'(0.64863-0.81565]'	'(0.073197-0.15774]'	square
'(-0.311715--0.09472]'	'(0.073197-0.15774]'	femoral
'(-0.09472-0.21473]'	'(0.073197-0.15774]'	femoral
'(-0.915935--0.311715]'	'(0.073197-0.15774]'	femoral
'(-inf--0.915935]'	'(0.021496-0.073197]'	square
'(1.22915-inf)'	'(0.021496-0.073197]'	square
'(0.81565-1.22915]'	'(0.021496-0.073197]'	square
'(0.21473-0.64863]'	'(0.021496-0.073197]'	square
'(0.64863-0.81565]'	'(0.021496-0.073197]'	square
'(-0.311715--0.09472]'	'(0.021496-0.073197]'	femoral
'(-0.09472-0.21473]'	'(0.021496-0.073197]'	femoral

Figure 7.15 A snapshot of a decision table of the wavelet descriptors.

The PART classifier obtains rules from partial decision trees. It builds a partial C4.5 decision tree in each iteration and makes the "best" leaf node into a rule. This algorithm combines the divide-and-conquer strategy for decision tree learning with the separate-and-conquer strategy for rule learning. The separate-and-conquer strategy implies building a rule, removing the instances it covers, and continuing to create rules recursively for the remaining instances until none is left. The rule creation involves building a pruned decision tree, making the leaf with the largest coverage into a rule, and discarding the tree [113]. Figure 7.16 shows a snapshot of the decision list for the moment descriptors generated by the PART algorithm.

```
PART decision list
------------------

MD5 > 14.794 AND
MD1 > 312.81 AND
MD4 > -266.66: sine (146.0/2.0)

MD5 > 14.794 AND
MD4 > -8.8853 AND
MD2 > 23.737: femoral (131.0/1.0)

MD2 > 0.13487 AND
MD5 > 12.73 AND
MD4 <= 19.861: femoral (46.0/1.0)

MD2 > 0.13487 AND
MD1 > 69.924 AND
MD3 <= 51.74: square (26.0)

MD2 > 0.13487 AND
MD2 > 4.0258 AND
MD1 <= 124.96 AND
MD1 <= 61.767: carotid (165.0)

MD1 > 1.8532 AND
MD3 > 0.16761 AND
MD1 <= 191.81 AND
MD3 > 12.473: square (124.0/1.0)
```

Figure 7.16 A snapshot of a decision list of the moment descriptors generated by the PART classifier.

The t-test was used to statistically compare the different classifiers. One base classifier, the Naïve Bayes classifier, was used to compare to the other 9 classifiers. Table 7.6 shows the classification success rates of the 10 classifiers for the four different descriptors. A "B" letter adjacent to the success percentage indicates that the corresponding classifier was

statistically better than the Naïve Bayes classifier; a "W" letter indicates that the classifier is statistically worse than NBC. If there is no letter adjacent to the success rate, then the t-test was not conclusive.

Table 7.6 Classification success rates (in percentages) of the 10 classifiers for the four different descriptors using the NBS as a base classifier for comparison. B stands for statistically better, and W stands for statistically worse.

	NBS	BN	MLPs	RBF	KNN	LWL	C4.5	RF	DT	PART
Raw (32)	94.6	97.7 B	98.0 B	96.9 B	97.4 B	86.8 W	96.7 B	98.4 B	94.8	96.8 B
FD (5)	90.3	89.2	92.9 B	90.0	91.6	77.7 W	91.0	93.2 B	90.2	90.9
WD (16)	94.6	97.5 B	97.5 B	97.1 B	97.5 B	86.8 W	97.0 B	98.4 B	95.1	97.5 B
MD (5)	65.3	89.1 B	68.2 B	88.5 B	93.0 B	62.4	92.4 B	94.2 B	90.3 B	91.9 B

The following conclusions can be drawn from the results listed in Table 7.6:

- The previous conclusion regarding the better recognition performance achieved by both the raw and wavelet descriptors is confirmed with all the 10 classifiers; the success rates of the raw and wavelet descriptors are higher than those of the Fourier and moment descriptors.
- The differences between the recognition rates of the raw and wavelet descriptors are negligible. Hence we can conclude that the two descriptor types are equally powerful.
- The Fourier descriptors' performance is slightly better than the moment descriptors with some classifiers, such as NBS, MLPs, and LWL classifiers.
- All classifiers are statistically better than the classical Naïve Bayes Simple classifier in three or more descriptor types, except the Decision Table classifier which is statistically better only with the moment descriptor and the LWL classifier which is statistically worse than the NBS in three descriptor types.
- In general, the Random Forest classifier gave the best performance with all four descriptor types. To confirm this conclusion we changed the base class to be the RF classifier to compare its performance against the rest of the classifiers. Table 7.7 shows the same success rates of the 10 classifiers, but with comparison to the RF classifier. It is evident from the table entries that the majority of the cases of the other classifiers are statistically

worse than the RF, except for a few cases which are indeterminate. The two classifiers that are almost comparable to the RF are Bayesian Networks (BN) and Multilayer Perceptron Neural Network (MLPs) classifiers.

- The Locally-Weighted Learning (LWL) classifier in conjunction with the decision stump classifier gave the worst performance with all four descriptor types. To better illustrate this conclusion, we changed the base class to be the LWL classifier to compare its performance against the rest of the classifiers. Table 7.8 shows the same success rates of the 10 classifiers, but in comparison to the LWL classifier; it is evident from the table entries that all the cases of the other classifiers, except for one entry, are statistically better than the LWL one. This result was expected since the decision stump classifier is a weak one that relies only on a single attribute with a one-level tree. For example, the tree resulted from training the decision stump classifier with the moment descriptor data relied only on the fifth moment descriptor to classify either the carotid or femoral waveforms; all other waveforms were misclassified. Adding the locally-weighted training approach enhanced the performance of the decision stump by more than 50%. However, the simplicity of the underlying classifier still falls behind all other more sophisticated algorithms.
- Excluding the worst-performing classifier (LWL), the classification success rates of the raw, wavelet, and Fourier descriptors are within a close range for all classifiers (within 5% difference), while the range of the moment descriptor varies much among all the classifiers (almost 30% difference). This indicates that the classifier choice for the first three descriptors is not as critical as the classifier choice for the moment descriptor.

Table 7.7 Classification success rates (in percentages) of the 10 classifiers for the four different descriptors using the RF classifier as a base classifier for comparison. B stands for statistically better, and W stands for statistically worse.

	RF	NBS	BN	MLPs	RBF	KNN	LWL	C4.5	DT	PART
Raw (32)	98.4	94.6W	97.7	98.0	96.9W	97.4W	86.8W	96.7W	94.8W	96.8W
FD (5)	93.2	90.3W	89.2W	92.9	90.0W	91.6W	77.7W	91.0W	90.2W	90.9W
WD (16)	98.4	94.6W	97.5	97.5	97.1W	97.5W	86.8W	97.0W	95.1W	97.5
MD (5)	94.2	65.3W	89.1W	68.2W	88.5W	93.0	62.4W	92.4W	90.3W	91.9W

Table 7.8 Classification success rates (in percentages) of the 10 classifiers for the four different descriptors using the LWL as a base classifier for comparison. B stands for statistically better, and W stands for statistically worse.

	LWL	NBS	BN	MLPs	RBF	KNN	C4.5	RF	DT	PART
Raw (32)	86.8	94.6 B	97.7 B	98.0 B	96.9 B	97.4 B	96.7 B	98.4 B	94.8 B	96.8 B
FD (5)	77.7	90.3 B	89.2 B	92.9 B	90.0 B	91.6 B	91.0 B	93.2 B	90.2 B	90.9 B
WD (16)	86.8	94.6 B	97.5 B	97.5 B	97.1 B	97.5 B	97.0 B	98.4 B	95.1 B	97.5 B
MD (5)	62.4	65.3	89.1 B	68.2 B	88.5 B	93.0 B	92.4 B	94.2 B	90.3 B	91.9 B

7.8. Noise Resilience Evaluation

The final evaluation step of our vessel recognition system is to test the performance of the four suggested descriptors with increasing noise level added to the input waveforms. This evaluation is necessary since the noise level in the *in vivo* data is expected to be higher than the noise level in the *in vitro* flow phantom data that we have been trying so far. To test the noise resilience level of the different descriptor types, we added Gaussian noise to the original vessel waveforms with increasing variance level. Figure 7.17 shows an example of a femoral waveform with added Gaussian noise of different variances. With the preserved-sign scaling methods, the variance of the added noise has the following values (.005, .01, .02, .04, .08), as shown in the figure.

The degradation of the recognition rate with increasing the noise level was measured for each descriptor type. The Random Forest classifier, which is the best performing one, was used for this experiment with 10-fold cross validation on all the training and test datasets combined (TRAINING1+TEST1+TEST2). The recognition rates versus the increasing noise level are displayed in Figure 7.18 for the four descriptor types. The following conclusions can be drawn from Figure 7.18:

- The wavelet descriptors are the most noise resilient descriptor type for the vessel recognition application since their performance degradation has the least slope among all four descriptors.

- The moment descriptors are the least resilient descriptor type for our application. The moment descriptor degradation rate is the steepest one with increasing noise level.
- The degradation rate of both the raw and Fourier descriptors are almost the same. The raw descriptor may be slightly better. However the raw descriptors still give better recognition rates than the Fourier and moment descriptors for all noise levels.

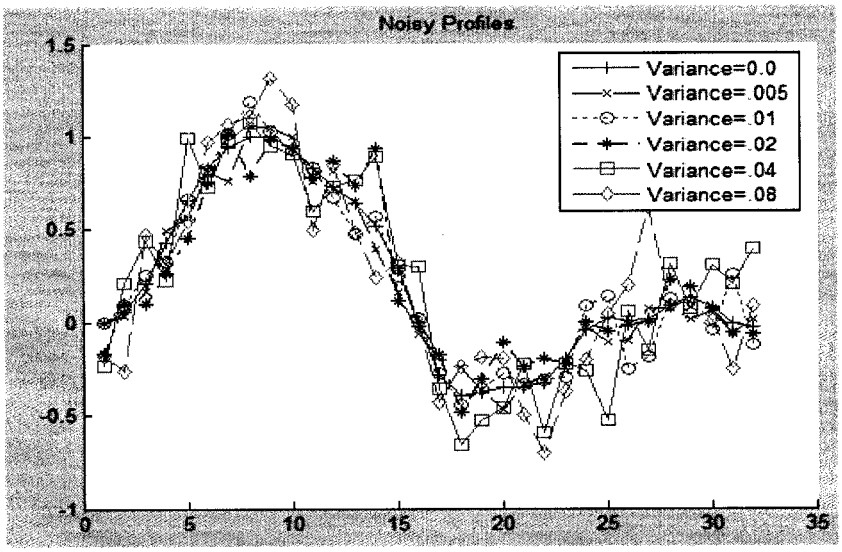


Figure 7.17 A femoral waveform with additive Gaussian noise with increasing variance.

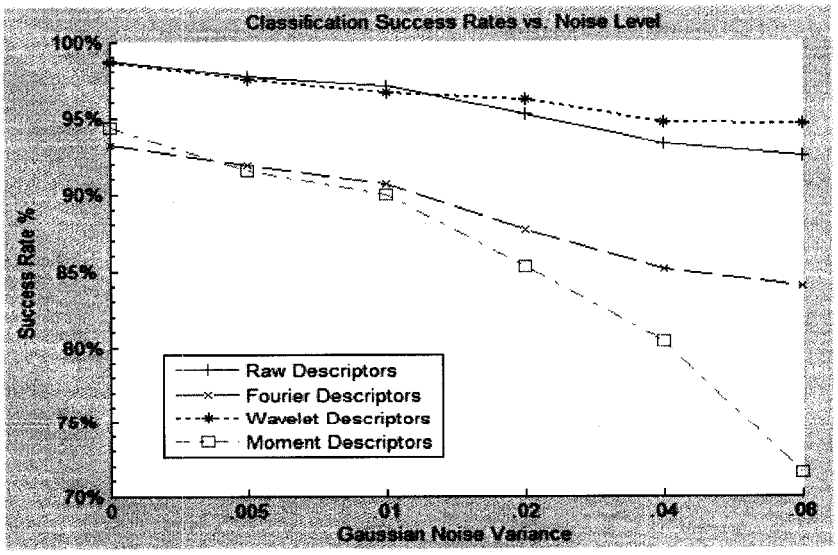


Figure 7.18 Recognition rates for the different descriptor types with increasing Gaussian noise levels.

7.9. Conclusions and Future Work

In this chapter we have concluded our research work by evaluating the last phase of our vessel recognition system, which is the classifier performance. The different data preprocessing and normalization methods developed in the previous chapter were evaluated and the newly-developed preserved-sign scaling method achieved the best classifier performance. More datasets were acquired and processed in order to further evaluate the classifier performance. Two different mechanisms for attribute selection were conducted and the optimal number of attributes for each descriptor type was determined. Classifier performance was evaluated with independent datasets and 10 different classifiers were tested. The Random Forest (RF), Bayesian Networks (BN), and Multilayer Perceptron Neural Network (MLPs) classifiers were the best classifiers for our vessel recognition system, while the Locally-weighted Learning (LWL) classifier, in conjunction with the Decision Stump classifier, was the worst. Finally the wavelet descriptors proved to be the best noise resilient descriptor type among all four descriptors evaluated when increasing level of additive Gaussian noise is added to the original waveforms.

While the segmentation and phase unwrapping stages of our vessel recognition system have been tested with both *in vitro* and *in vivo* data, the classifier stage has only been tested with *in vitro* flow phantom data. This is mainly due to the large number of datasets needed for both the training and test phases of designing and testing the classifier, which was not feasible with *in vivo* data due to limited access to human models and expert sonographers' time. However the waveforms of the *in vitro* data represent a variety of *in vivo* cases and the results obtained with the *in vitro* data are very promising. The next logical step is to carry out large scale clinical trials with *in vivo* data, while focusing on specific area of the vascular applications, such as the carotid tree including the common carotid, internal, carotid, external carotid, jugular vein, and subclavian vein. Building a recognition system for these close-by vessels should greatly simplify the manipulation and optimization of the ultrasound system for this type of vascular exam.

The descriptor types that we tried, including Fourier, wavelet, and moment descriptors, represent a broader class of linear transforms that can be applied to the input data in order to achieve information packing compared to the original input data [107]. The other transforms include the Karhunen-Loeve transform, the discrete cosine transform, the Haar transform, and the Hadamard transform. Any of these transforms could be used with our vessel recognition problem.

The Karhunen-Loeve transform, or the principle component analysis (PCA) transform, is an interesting one since it generates optimally uncorrelated features Y_{nx1} from the input feature vector X_{nx1} as $Y=A^TX$. Each feature or measurement of X is treated as a random variable. Matrix A is chosen such that its columns are the eigenvectors V_i of the covariance (correlation) matrix of the training input feature vector. The correlation matrix of Y is a diagonal matrix with uncorrelated features. X can be approximated by $m < n$ basis vectors, which represent the projection of X onto a subspace spanned by the m eigenvectors. Choosing the eigenvectors corresponding to m largest eigenvalues minimizes the mean square approximation error. The concept of the principal eigenvectors subspace has also been exploited as a classifier [107]. First the sample mean of the training set is subtracted from the feature vectors. For each class w_i, the correlation matrix is estimated and the principal m eigenvectors of the largest m eigenvalues are computed. A matrix A_i is then formed using the respective eigenvectors as columns. An unknown feature vector x is then classified in class w_i for which $\| A_i x \| > \| A_j x \| \ \forall i \neq j$. The main drawback of the PCA transform is that its basis functions are data dependent, in contrast to the basis functions of the other transforms such as the Fourier or the Wavelet.

We did some preliminary PCA analysis of our vessel waveforms and we show here some interesting results. Table 7.9 shows the result of the PCA analysis of the vessel waveform training data (TRAINING1), where the largest 10 eigenvalues are shown for each waveform type. It is evident that the waveform information is concentrated in the first few eigenvalues for all waveform types except the constant waveform, which has its energy spread across all eigenvalues almost equally. This is expected since the constant waveform is almost

a DC frequency, while all other waveforms show some degree of pulsatility. Figure 7.19 shows the PCA reconstruction of the different waveform if only the largest eigenvector is used to reconstruct the waveforms back. One observation about the reconstructed waveforms is that they maintain the main shape information of the underlying waveforms but with much lower noise level. Another observation is that the reconstructed constant waveforms deviate by a large degree from the original ones, in contrast with the other four waveforms. This is expected, since the largest eigenvector does not carry much information about the constant waveform, as indicated in Table 7.9.

Table 7.9 PCA analysis for the vessel waveforms. The largest 10 eigenvalues are shown or each waveform.

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
Carotid	.792	.078	.036	.026	.012	.01	.008	.006	.005	.003
Constant	.011	.004	.004	.003	.003	.003	.002	.002	.002	.002
Femoral	.739	.662	.387	.11	.076	.062	.041	.033	.02	.014
Sine	.689	.403	.228	.171	.09	.07	.045	.039	.026	.024
Square	.272	.074	.062	.024	.02	.017	.01	.004	.003	.002

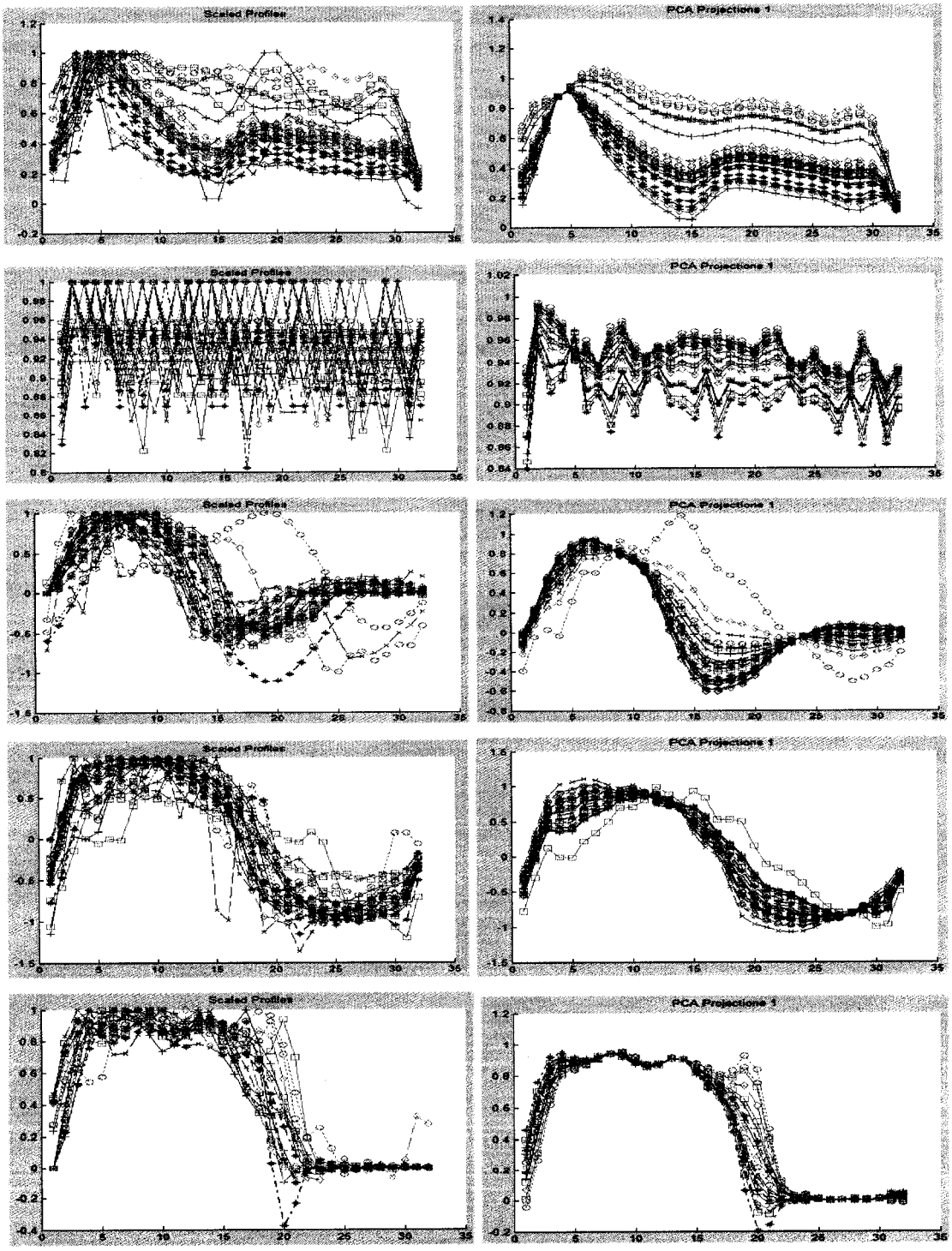


Figure 7.19 PCA reconstruction using only the largest eigenvector of the different waveforms. Left: the preserved-sign scaled training data. Right: the reconstructed waveforms using the largest eigenvector for the carotid, constant, femoral, sine, and square waveform.

Chapter 8 Conclusions and Future Work

In this thesis, we have developed a complete vessel recognition system based on the analysis and processing of color Doppler ultrasound images. Several novel clinical applications that directly benefit from the developed techniques have also been described. The contributions of this thesis can be summarized as follows: We introduced advanced computer vision and image understanding techniques to the field of ultrasound imaging. The main goal was to achieve more intelligent operation of current ultrasound systems by incorporating high-level knowledge about the image content to guide the optimization of the acquired images.

An important clinical application was targeted, which is the vascular disease that constitutes a main cause of mortality and morbidity in the world. The proposed solutions should enhance the quality of the vascular ultrasound exam outcome by providing the ease of use of the ultrasound machine and automating some of the frequent and error-prone tasks that are done manually nowadays.

A novel vessel segmentation technique was proposed that applies a shape decomposition approach to segment distinct vessels in color Doppler ultrasound images. The idea represents a new clinical application of the shape analysis literature.

We developed a fully-automated clinical application, which is the automated Doppler angle estimation based on high-level image processing techniques. The Doppler angle estimation is a very critical clinical problem for the assessment of the stenosis degree in human arteries. The proposed automation technique should streamline the outcome of the vascular exam and minimize the intra-observer and inter-observer variability of the stenosis grading. The preliminary clinical evaluations suggested a very positive agreement with the expert angle settings.

The color Doppler ultrasound aliasing problem was addressed, which is the main source of distortion in the velocity data. Although the phase unwrapping problem is a very

mature research field in other imaging modalities, such as radar and MRI imaging, a thorough analysis of the problem never existed for the ultrasound color Doppler imaging field. We have addressed the color Doppler aliasing problem in detail and proposed a novel phase unwrapping technique to restore the true velocity values in color Doppler images. The new algorithm was carefully tested with many *in vitro* and *in vivo* datasets, in addition to comparing its performance against some well-known phase unwrapping techniques.

New data reduction and feature generation techniques were specifically developed for color Doppler ultrasound images. The idea of generating virtual spectrogram profiles that retain the most relevant hemodynamic information of the underlying vessels is very powerful in terms of achieving high data reduction ratios and retaining high discrimination power.

Thorough evaluations were conducted for a number of well-known transform-based feature descriptors, data normalization methods, and different classifiers. The performance of the best combination of feature descriptors, scaling methods, and classifier was very satisfactory with high recognition rates.

The future directions of this research can be summarized as follows: The logical next step of this work is to integrate all the algorithms with the signal path of the ultrasound system to allow testing the recognition system in a real clinical environment. We have tried to include *in vivo* data in many stages of this research, such as the segmentation and phase-unwrapping stages, however the on-system implementation will allow for more extensive clinical evaluation, especially for those newly-developed clinical applications.

A large scale ultrasound controls automation project can be launched that benefits from the infrastructure building blocks developed in this thesis. A dramatic simplification of the vascular exam can be achieved. Clinical trials can be conducted to evaluate the degree of enhancement in terms of both workflow simplification and clinical outcome consistency.

Another extension for the vascular application is to consider the 3-D color Doppler ultrasound, which is still an active area of research and has not yet been widely accepted in the

clinical arena. Recognizing vessels from 3-D datasets in time instead of 2-D ones will have a larger impact on the clinical outcome since one main complaint of 3-D ultrasound nowadays is the complexity of operating the system to obtain relevant clinical data. Automation and quantification tasks should greatly simplify the 3-D ultrasound workflow. The main advantage will be the fast acceptance of these techniques by the medical community since 3-D ultrasound is still a novice clinical tool that does not have a widely acceptable workflow yet.

This same research direction can extend to other ultrasound clinical applications, such as obstetrics and cardiology, which are as important as the vascular application. Recognition systems can be developed for the fetus organs or the human heart parts to optimize the ultrasound system settings accordingly. Another important extension of this research will be considering pathological cases as well as normal ones. In our research we focused on either *in vitro* or normal *in vivo* data due to the availability of the data. However in the future the developed techniques can be extended to recognize disease states for the same vessel. For example, automating the grading of the carotid stenosis based on the changes in the Doppler waveforms could be possible. This should open the door for incorporating more advanced computer aided diagnosis, or CAD, systems inside the ultrasound system or within the associated archiving or reporting packages. The huge patient databases available today in the archiving, or PACs, systems can represent training data for building classification models for the disease state.

Bibliography

[1] J. Woo, "A short History of the development of Ultrasound in Obstetrics and Gynecology", http://www.ob-ultrasound.net/history1.html.

[2] C. Otto, Textbook of Clinical Echocardiography, 2nd edition, W.B. Saunders Company, Philadelphia, USA, 2000.

[3] C. Krebs, V. Giyanani, and R. Eisenberg, Ultrasound Atlas of Vascular Diseases, Appleton and Lange, Connecticut, USA, 1999.

[4] The Philips Healthcare iU22 high-end diagnostic ultrasound system for radiology labs, http://www.medical.philips.com/us/products/ultrasound/general/iu22/

[5] E. Grant, C. Benson, G. Moneta, A. Alexandrov, J. Baker, E. Bluth, et al. "Carotid artery stenosis: gray-scale and Doppler US diagnosis— Society of Radiologists in Ultrasound Consensus Conference," Radiology, vol. 229, pp. 340-346, 2004.

[6] A. Steinman, J. Tavakkoli, J. Myers, R. Cobbold, K. Johnston, "Sources of error in maximum velocity estimation using linear phased-array Doppler systems with steady flow," Ultrasound in Medicine and Biology, vol. 27, no. 5, pp. 655-664, May 2001.

[7] E. Lui, A. Steinman, R. Cobbold, K. Johnston, "Human factors as a source of error in peak Doppler velocity measurement," Journal of Vascular Surgery, vol. 42, no. 5, Nov 2005.

[8] C. Murphy, and A. Russo, "An Update on Ergonomic Issues in Sonography Report", http://www.sdms.org/pdf/sonoergonomics.pdf , Healthcare Benefit Trust July 2000.

[9] A. Smith, J. Wolf, G. Xie, and M. Smith, "Musculoskeletal Pain in Cardiac Ultrasonographers: Results of a Random Survey," Journal of the American Society of Echocardiography, vol. 10, no. 4, pp. 357-62, May 1997.

[10] N. Magnavita, L. Bevilacqua, P. Mirk, A. Fileni, and N. Castellino, "Work-related Musculoskeletal Complaints in Sonologists," Journal of Occupational Environment in Medicine, vol. 41, no. 11, pp. 981-988, Nov 1999.

[11] P. Mirk, N. Magnavita, L. Masini, M. Bazzocchi, and A. Fileni, "Frequency of Musculoskeletal Symptoms in Diagnostic Medical Sonographers. Results of a Pilot Survey," Radiol Med, vol. 98, no. 4, pp. 236-241, Oct 1999.

[12] D. White, "Johann Christian Doppler and his effect - a brief history," Ultrasound in Medicine and Biology, vol. 8, pp. 583-591, 1982.

[13] J. Zagzebski, Essentials of Ultrasound Physics, Mosby Inc., 1996.

[14] D. Evans and W. McDicken, Doppler Ultrasound Physics, Instrumentation, and Signal Processing, 2nd edition, John Wiley and Sons Ltd, Chichester, England, 2000.

[15] P. Wells, "A range-gated ultrasonic Doppler system," Engineering in Medicine and Biology, vol. 7, pp. 641-652, 1969.

[16] D. Baker, "Pulsed ultrasonic Doppler blood-flow sensing," IEEE Transactions on Sonics and Ultrasonics, vol. 17, pp. 170-185, 1970.

[17] Namekawa K, Kasai C, Tsukamoto M, Koyano A, "Imaging of blood flow using autocorrelation," Ultrasound in Medicine and Biology, vol. 8, 1982.

[18] C. Kasai, K. Namekawa, A. Koyano, and R. Omoto, "Real-time two-dimensional blood flow imaging using an auto-correlation technique," IEEE Transactions on Sonics and Ultrasonics, vol. 32, pp. 458-464, 1985.

[19] J. Willemetz et al., "Bias and variance in the estimate of the Doppler frequency induced by a wall motion filter," Ultrasonic Imaging, vol. 11, pp. 215-225, 1989.

[20] M. Shariati, J. Dripps, and W. McDicken, "Deadbeat IIR based MTI filtering for color flow imaging systems," Proceedings of the 1993 IEEE Ultrasonics Symposium, vol. 2, pp. 1059-1063, 1993.

[21] A. Hoeks et al., "An efficient algorithm to remove low frequency Doppler signals in digital systems," Ultrasonic Imaging, vol. 13, pp. 135-144, 1991.

[22] R. Collaris, and A. Hoeks, "Postprocessing of velocity distributions in real-time ultrasonic color velocity imaging," Ultrasonic Imaging, vol. 16, pp. 249-264, 1994.

[23] D. Lipschutz, "Combined color flow map and monochrome image," USA patent no. 4761740, 1988.

[24] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 8, no. 6, 1986.

[25] C. Kirbas, "A Review of Vessel Extraction Techniques and Algorithms," ACM Computing Surveys, vol. 36, pp. 81-121, June 2004.

[26] N. Armande, P. Montesinos, and O. Monga, "Thin nets extraction using multi-scale approach," Computer Vision and Image Understanding, vol. 73, no. 2, pp. 248–257, 1999.

[27] H. Blum, "A transformation for extracting new descriptors of shape," Models for the Perception of Speech and Visual Form. Cambridge, MA: M.I.T Press, 1967.

[28] E. Sorantin, C. Halmai, B. Erdohelyi, K. Palagyi, L. Nyul, K. Olle, B. Geiger, F. Lindbichler, G. Friedrich, and K. Kiesler, "Spiral-ct based assessment of tracheal stenosis using 3d-skeletonization," IEEE Transactions on Medical Imaging, vol. 21, no. 3, pp. 263–273, Mar 2000.

[29] J. Udupa and S. Samarasekea, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," Graphical Models in Image Processing, vol. 58, no. 3, pp. 246-261, 1996.

[30] S. Aylward, S. Pizer, E. Bullitt, and D. Eberl, "Intensity ridge and widths for tubular object segmentation and description," In Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 131–138, 1996.

[31] H. Schmitt, M. Grass, V. Rasche, O. Schramm, S. Haehnel, and K. Sartor, "An x-ray based method for the determination of the contrast agent propagation in 3-d vessel structures," IEEE Transactions on Medical Imaging, vol. 21, no. 3, pp. 251–262, Mar 2002.

[32] R. Poli, and G. Valli, "An algorithm for real-time vessel enhancement and detection," Computer Methods and Programs in Biomedicine, vol. 52, no. 1, pp. 1–22, Jan 1997.

[33] S. Eiho, and Y. Qian, "Detection of coronary artery tree using morphological operator," IEEE Computers in Cardiology, vol. 24, pp. 525–528, 1997.

[34] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," International Journal of Comput. Vision, vol. 1, no. 4, pp. 321–331, 1988.

[35] A. Singh, D. Terzopoulos, and D. Goldgof, Deformable Models in Medical Image Analysis, Wiley, John & Sons, Incorporated, 1998.

[36] H. Luo, L. Qiang, R. Acharya, and R. Gaborski, "Robust snake model," Proceedings of the IEEE Conference on CVPR, vol. 1, 2000.

[37] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," Journal of Numerical Mathematics, vol. 66, no. 1, pp. 1–32, 1993.

[38] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 2, pp. 158–175, Feb 1995.

[39] T. O'Donnell, T. Boult, X. Fang, and A. Gupta, "The extruded generalized cylinder: A deformable model for object recovery," Proceedings of the IEEE Conference on CVPR, 1994.

[40] M. Goldbaum, S. Moezzi, A. Taylor, S. Chatterjee, J. Boyd, E. Hunter, and R. Jain, "Automated diagnosis and image understanding with object extraction, object classification, and inferencing in retinal images," IEEE International Conference on Image Processing, vol. 3, 1996.

[41] C. Perry, U. Scheipers, W. Bojara, M. Lindstaedt, S. Holt, H. Ermert, "Computerized Segmentation of Blood and Luminal Borders in Intravascular Ultrasound," IEEE Ultrasonic Symposium, vol. 2, pp. 1122-1125, 2004.

[42] D. Zheng, and G. Lu, "Review of shape representation and description techniques," Pattern Recognition, vol. 37, no. 1, pp. 1-19, 2004.

[43] K. Siddiqi and B. Kimia, "Parts of Visual Form: Computational Aspects," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol.17, pp.239-251, 1995.

[44] L. Li and W. Chen, "Corner Detection and Interpretation on Planar Curves Using Fuzzy Reasoning," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 21, no. 11, Nov 1999.

[45] F. Arrebola, A. Bandera, P. Camacho, F. Sandoval, "Corner Detection by Local Histograms of Contour Chain Code," Electronics Letters, vol. 33, no. 21, Oct 1997.

[46] J. Hua, and Q. Liao, "Wavelet-based Multiscale Corner Detection," Proceedings of ICSP, 2000.

[47] A. Quddus, and M. Fahmy, "Fast wavelet-based corner detection technique," Electronics Letters, vol. 35, no. 4, Feb. 1999.

[48] J. Han, "Detection of convex and concave discontinuity points in a plane curve," 3rd International Conference on Computer Vision, pp.71-74, 1990.

[49] S. Marchand-Maillet and Y. Sharaiha, Binary Digital Image Processing, A Discrete Approach, Academic Press, London, UK, 2000.

[50] U. Montanari, "Continuous Skeletons from digitized images," Journal of the ACM, vol. 16, 1969.

[51] I. Bitter, A. Kaufman, and M. Sato, "Penalized-Distance Volumetric Skeleton Algorithm," IEEE Transactions on Visualization and Computer Graphics, vol. 7, no. 3, pp. 195-206, 2001.

[52] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C, second edition, Cambridge University Press, UK, 1999.

[53] A. Saad, and L. Shapiro, "Computer Vision Approach for Ultrasound Doppler Angle Estimation", The Society for Imaging Informatics in Medicine (SIIM) conference, Seattle, WA, May 2008.

[54] F. Mochtarian and A. Mackworth, "A Theory of Multiscale, Curvature-based Shape Representation of Planar Curves," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 14, no. 8, 1992.

[55] B. Jang and R. Chin, "Morphological Scale space for 2-D Shape Smoothing," Proceedings of the IEEE Conference on Image Processing (ICIP), vol. 2, 1994.

[56] Y. Wang, S. Lee, K. Toraichi, "Multiscale Curvature-based Shape Representation using B-Splines Wavelets," IEEE Transactions on Image Processing, vol. 8, no. 11, 1999.

[57] J. Lauchard, A. Vialard, and F. Vieilleville, "Analysis and Comparative Evaluation of Discrete Tangent Estimators," Proceedings of 12th International Conference on Discrete Geometry for Computer Imagery, pp. 240-251, 2005.

[58] J. Bland and D. Altman, "Statistical methods for assessing agreement between two methods of clinical measurement", The Lancet, vol. i, pp. 307 –310, 1986.

[59] A. Saad and L. Shapiro, "Shape Decomposition Approach for Ultrasound Color Doppler Image Segmentation", Proceedings of the IEEE International Conference of Pattern Recognition (ICPR), vol. 4, pp. 691-694, 2006.

[60] D. Ghiglia and M. Pritt, Two-Dimensional Phase Unwrapping Theory, Algorithms, and Software, John Wiley and Sons, Inc. NY, USA, 1998.

[61] K. Itoh, "Analysis of the phase unwrapping problem," Applied Optics, vol. 21, no. 14, pp. 2470, Jul 1982.

[62] D. Ghiglia, G. Martin, and L. Romero, "Cellular automata method for phase unwrapping," Journal of the Optical Society of America, vol. 4, no. 1, pp. 267-280, Jan 1987.

[63] R. Goldstein, H. Zebker, and C. Werner, "Satellite radar interferometry: two-dimensional phase unwrapping," Radio Science, vol. 23, pp. 713-720, 1988.

[64] R. Cusack, J. Huntley, and H. Goldrein, "Improved noise-immune phase-unwrapping algorithm," Applied Optics, vol. 34, no. 5, pp. 781-789, 1995.

[65] J. Buckland, J. Huntley, and S. Turner, "Unwrapping noisy phase maps by use of a minimum cost matching algorithm," Applied Optics, vol. 34, no. 23, pp. 5100-5108, 1995.

[66] D. Bone, "Fourier fringe analysis: the two-dimensional phase unwrapping problem," Applied Optics, vol. 30, pp. 3627-3632, 1991.

[67] H. Lim, W. Xu, and X. Huang, "Two new practical methods for phase unwrapping," IEEE Proceedings of the International Geoscience and Remote Sensing Symposium, vol. 1, pp. 196-198, Jul 1995.

[68] T. Flynn, "Consistent 2-D phase unwrapping and absolute values information," IEEE Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), pp. 2057-2059, 1996.

[69] T. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity," Journal of the Optical Society of America, vol. 14, no 10, pp. 2692-2701, Oct 1997.

[70] D. Ghiglia and L. A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods," Journal of the Optical Society of America, vol. 11, no. 1, pp. 107-117, 1994.

[71] M. Pritt, "Phase unwrapping by means of multigrid techniques for interferometric SAR," IEEE Transactions on Geoscience and Remote Sensing, vol. 34, no. 3, pp. 728-738, 1996.

[72] D. Ghiglia and L. Romero, "Minimum Lp-norm two-dimensional phase unwrapping," Journal of the Optical Society of America, vol. 13, pp. 1-15, 1996.

[73] M. Pritt, "Congruence in least-squares phase unwrapping," IEEE Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), 1997.

[74] Mark Pritt, "Comparison of Path-Following and Least-Squares Phase Unwrapping Algorithms," IEEE Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), vol. 2, pp. 872 – 874, 1997.

[75] S. Chavez, Q. Xiang, and L. An, "Understanding Phase Maps in MRI: A New Cutline Phase Unwrapping Method," IEEE Transactions on Medical Imaging, vol. 21, no. 8, Aug 2002.

[76] R. Kramer and O. Loffeld, "A novel procedure for cutline detection," International Journal of Electronics and Communications, vol. 50, no. 2, pp. 112–116, Mar 1996.

[77] A. Oppenheim and J.S. Lim, "The importance of phase in signals", Proceedings of the IEEE, vol. 69, no. 5, pp. 529-541, May 1981.

[78] D. Eugene Strandness, Jr., Duplex Scanning in Vascular Disorders, third edition, Lippincott Williams and Wilkins, Philadelphia, PA, USA, 2002.

[79] Matlab©, the language of technical computing. The MathWorks Inc. Natick, Massachusetts.

[80] B. Heinz, J. Gil, D. Kirkpatrick, and M. Werman, "Linear Time Euclidean Distance Transform Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 5, pp. 529-533, May 1995.

[81] A. Saad, and L. Shapiro, "A New Fringeline-Tracking Approach for Color Doppler Ultrasound Imaging Phase Unwrapping," SPIE Medical Imaging conference, San Diego, CA, Feb 2008.

[82] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A Comparative Performance Study of Several Pitch Detection Algorithms," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 5, Oct 1976.

[83] J. Dubnowski, R. Schafer, and L. Rabiner, "Real-time digital hardware pitch detector," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, pp. 2-8, Feb 1976.

[84] N. Miller, "Pitch detection by data reduction," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23, pp. 72-79, Feb 1975.

[85] R. Schafer and L. R. Rabiner, "System for automatic formant analysis of voiced speech," Journal of the Acoustical Society America, vol. 47, pp. 634-648, Feb 1970.

[86] D. Charalampidis, V. Kura, "Novel wavelet-based pitch estimation and segmentation of non-stationary speech", 8th International Conference on Information Fusion, vol. 2, Jul 2005.

[87] S. Orfanidis, Optimum Signal Processing. An Introduction. 2nd Edition, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[88] H. Jia and M. Xie, "Improvement of Fourier descriptor using spatial normalization," IEEE International Symposium on Communications and Information Technology (ISCIT), vol. 2, pp. 1284-1287, Oct 2005.

[89] J. Li and C. Kuo, "Automatic Target Shape Recognition via Deformable Wavelet Templates," SPIE vol. 2756.

[90] A. Whitehouse, X. Lu, J. Miller, "Computer-based statistical, pattern recognition of Doppler spectral waveform kinematics during fetal respiration," IEEE International Symposium on Biomedical Imaging: Nano to Macro, pp. 880-883, Apr 2006.

[91] C. Christodoulou, C. Pattichis, "Unsupervised pattern recognition for the classification of EMG signals," IEEE Transactions on Biomedical Engineering, vol. 46, pp. 169-178, Feb 1999.

[92] J. Cowe, D. Evans, "Automatic detection of emboli in the TCD RF signal using principal component analysis," Ultrasound in Medicine & Biology, vol. 32, no. 12, pp. 1853–1867, 2006.

[93] C. Zahn and R. Roskies, "Fourier descriptors for plane closed curves," IEEE Transactions on Computers, vol. 21, pp. 269-281, Mar 1972.

[94] G. Granlund, "Fourier preprocessing for hand print character recognition," IEEE Transactions on Computers, vol. 21, pp. 195-201, Feb 1972.

[95] Q. Chen, E. Petriu, and X. Yang, "A comparative study of Fourier descriptors and Hu's seven moment invariants for image recognition," Canadian Conference on Electrical and Computer Engineering, vol. 1, pp. 103 – 106, May 2004.

[96] G. Chuang and C. Kuo, "Wavelet Descriptors of Planar Curves: Theory and Applications," IEEE Transactions on Image Processing, vol. 5, no. 1, Jan 1996.

[97] K. Hung, "The Generalized Uniqueness Wavelet Descriptor for Planar Closed Curves," IEEE Transactions on image processing, vol. 9, no. 5, May 2000.

[98] O. Trier, A. Jain, and T. Taxt, "Feature extraction methods for character recognition – a survey," Pattern Recognition, vol. 29, no. 4, pp. 641-662, 1996.

[99] M. Hu, "Visual pattern recognition by moment invariants," IRE Transactions on Information Theory, vol. 8, pp. 179-187, Feb 1962.

[100] T. Reiss, "Recognizing of planar objects using invariant image features," Lecture Notes in Computer Science. vol. 676, Springer-Verlag, Berlin, 1993.

[101] A. Mamistvalov, "n-Dimensional moment invariants and conceptual mathematical theory of recognition n-Dimensional solids," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 20, no. 8, Aug 1998.

[102] B. Mertzios, K. Tsirikolias, "Fast shape discrimination using one-dimensional moments", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. 2473 – 2475, Apr 1991.

[103] R. Erich and J. Foith, "Representation of random waveforms by relational trees," IEEE Transactions on Computers, vol. C-25, pp. 725-736, Jul 1976.

[104] S. Shaw and R. Defiguiredo, "Structural Processing of Waveforms as Trees," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 2, Feb 1990.

[105] N. Parthasarathy, M. Ahmedi, and M. Shridhar, "Pattern Based ECG Recognition," IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, May 1991.

[106] I. Witten, E. Frank, Data Mining, Practical Machine Learning Tools and Techniques, second edition, Morgan Kaufmann, Jun 2005.

[107] S. Theodoridis, and K. Koutroumbas, Pattern Recognition, Academic Press, 1999.

[108] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, no. 12, pp. 1137–1143, 1995.

[109] D. Aha, and D. Kibler, "Instance-based learning algorithms", Machine Learning, vol.6, pp. 37-66, 1991.

[110] R. Quinlan, C4.5, Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[111] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, Oct 2001.

[112] R. Kohavi, "The Power of Decision Tables," Proceedings of the European Conference on Machine Learning, 1995.

[113] E. Frank and I. Witten, "Generating Accurate Rule Sets without Global Optimization," Shavlik Journal, Proceedings of the Fifteenth International Conference Machine Learning:, Morgan Kaufmann Publishers, San Francisco, CA, 1998.

[114] G. Cooper, and E. Herskovits. 1992, "A Bayesian method for the induction of probabilistic networks from data," Machine Learning, vol. 9, no. 4, pp. 309-347, 1992.

VITA

Ashraf Saad was born in Cairo, Egypt in 1972. At the University of Cairo he earned a Bachelor of Engineering degree in Biomedical and Systems Engineering in 1995 and a Master of Science degree in Biomedical and Systems in 1998. In 2000 he earned a Master of Science degree in Computer Engineering and Computer Science from Louisville University. In 2008 he earned a Doctor of Philosophy at the University of Washington in Electrical Engineering.