Real-Time Imaging of Elastic Properties of Soft Tissue with Ultrasound

by

Reza Zahiri-Azar

B.Sc., Sharif University of Technology, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

September 2005

Abstract

Current imaging devices such as computed tomography (CT), ultrasound (US) and magnetic resonance imaging (MRI) are not directly capable of measuring the mechanical properties of soft tissue even though such measurement would have a high clinical demand. Elastography with the aid of ultrasound has been well established in the literature as a strain imaging technique. Under certain conditions, these strain images can give a clear illustration of the underlying tissue stiffness distributions which has been shown to provide useful clinical information. Vibro Elastography is another new imaging system that performs a transfer function analysis of the tissue motion. The shape of the transfer function can be analyzed further and the stiffness of tissue can be estimated from the magnitude of the transfer functions at low-frequencies.

This thesis introduces a fast and accurate motion tracking algorithm which is at the heart of both strain imaging and stiffness imaging. The algorithm achieves real-time performance ($\geq$ 20 fps) without any need for additional hardware and its overhead. The performance of the proposed method is evaluated quantitatively according to its signal-to-noise ratio, contrast-to-noise ratio, dynamic range, resolution and sensitivity with both simulation data and phantom data. Also, the computational efficiency of the algorithm is compared with current real-time motion tracking algorithms. The results show that it is the most time efficient algorithm to date. Furthermore the performance of the proposed method is evaluated qualitatively from the real-time images that are generated in both tissue mimicking phantoms and real tissues in vivo.

By using this method two real-time elastography packages have been implemented which can easily be clinically applied. These implementations run at 35fps for strain images and 2fps for transfer function images of 16,000 pixels on an Ultrasonix RP500 ultrasound machine.

Contents

List of Tables

List of Figures

Glossary

1D	One dimensional.
2D	Two dimensional.
3D	Three dimensional.
ACC	Cost of conversion from real signal to analytic signal for a RF A-line.
ACCW	Cost of conversion from real signal to analytic signal for a single RF window.
BCC	Cost of conversion from real signal to base-band signal for a RF A-line.
BCCW	Cost of conversion from real signal to base-band signal for a single RF window.
B-mode	Brightness mode ultrasound image.
CC	Correlation Cost.
CAM	Combined Autocorrelation Method.
CNRe	Contrast to Noise Ratio in Elastography.
CSD	Cross Spectral Density.
CT	Computed tomography.
D	Depth of Ultrasound Image.
DRe	Dynamic Range in Elastography.
Elastography	Techniques for imaging mechanical properties of tissue.
Elastogram	An image obtained by elastography.
L	Length of an RF A-line.
LSE	Least Square Estimator.
MRI	Magnetic resonance imaging.
NCC	Normalized Correlation Cost.

PRS	Phase Root Seeking.
PSA	Peak Search Algorithm.
PSD	Power Spectral Density.
PSF	Point Spread Function.
RF	Radio Frequency Data.
SNRe	Signal to Noise Ratio in Elastography.
SNRs	Ultrasound Signal to Noise Ratio.
TDE	Time Domain Motion Estimation.
TDPE	Time Domain Motion Estimation with Prior Estimates.
US	Ultrasound.
ZCT	Zero Cross Tracking.
R_a	Axial Resolution in Elastography.
R_l	Lateral Resolution in Elastography.
c	Speed of Ultrasound in tissue.
ΔW	Window Shift.
W	Window Length.
nW	Number of Windows in a single RF line.
f_s	Sampling Frequency of Ultrasound Machine.
f_o	Centroid Frequency of Ultrasound Probe.
λ	Ultrasound Pulse wavelength.
E	Young's modulus.
d_i	Displacement measurement of window i.
k_i	Stiffness between windows i and $i+1$.
$H_i^j(w)$	Transfer function from window i to window j.

Acknowledgments

- I would primarily like to thank my supervisor Dr. Tim Salcudean for his guidance and support. Without his guidance and inspiration throughout the process, none of this would have been possible.
- I would also like to thank Trevor Hansen, Kris Dickie and Laurent Pelissier from Ultrasonix Medical Corporation, for their technical support and help.
- Many thanks to all my friends in the Robotics and Control lab, Julian for his help on the basics of Ultrasound imaging, Ehsan for linear algebra, Hani for signal processing, Hassan for continuous mechanics, Neerav for programming and Danny for his continually helpful advices.
- I owe my loving thanks to my wife Sara. Without her support, encouragement and understanding it would have been impossible for me to finish this work.
- Lastly, I am sincerely grateful to my parents who have always been the biggest motivation throughout my studies and whose years of love and hard work created the opportunity for me to chase my dreams.

Chapter 1

Introduction

1.1 Background

The basis of medical imaging is the measurement of a property of tissue that varies with tissue composition. Medical images are formed by displaying these properties measured at multiple locations in the body. From such images, a depiction of anatomy or pathology is gained. Each imaging modality in common use, such as X-ray, computed tomography (CT), ultrasound (US) and magnetic resonance imaging (MRI), measures a different property of tissue. But none of the properties measured by those modalities describe directly the mechanical properties of tissue.

Mechanical properties are properties that describe the deformation and mechanical behavior of tissue under a static or varying force. Elasticity (Young's modulus), compressibility (Poisson's ratio), stiffness, viscosity and density are examples of mechanical properties. Creating images from one or more mechanical properties is useful in medical practice, if the variation of the measured property creates sufficient contrast in the image to depict anatomy or pathology.

Elastography was first introduced in [81] and it aims to produce a new type of image that depicts the mechanical properties of tissue. The idea is to apply an external deformation to the tissue (such as axial compression) or to use an available internal deformation (like the heart beat or deformation of arterial walls from oscillating blood pressure) and investigate its corresponding interactions inside the tissue. An imaging device is employed to record these deformations and interactions. A motion estimation algorithm is then applied to find the displacement of each region in sequences of images. Finally different mechanical properties of each region are estimated from the recorded displacements.

Elastography has been well established in the literature as a strain imaging technique [18,78–82]. The strain distributions in tissues in response to an external deformation are closely related to the distribution of tissue elasticity [108]. It has been shown that the strain images are useful in different clinical applications such as tumor detection in the breast [30, 39, 40] , brain [8] and prostate [69], monitoring high intensity focused ultrasound lesions [95], imaging the myocardium [54], studying deep venous thrombosis [1, 32, 100], studying renal pathology [33], monitoring thermal changes and ablation [114, 128], intravascular applications such as finding vulnerable plaque [25, 26], mechanical behavior of skin [134] and obtaining the mechanical structural properties of tissue [50].

1.2 Ultrasound Imaging

Ultrasound has become a widely used medical scanning method due to its cost, ability for real time performance, quick setup procedures and easy access to its digital data. Therefore it has been widely used in motion estimation, flow estimation and elastography. In this thesis, the tissue's deformation is imaged by a conventional ultrasound system.

An ultrasound imaging system acquires data through the generation of an ultrasound wave directed toward the area to be examined, followed by measurement of the echoes generated by the interaction of the ultrasound wave with the tissue. The system consists of an ultrasound transducer for generating the ultrasound pulses and measurement of the echoes, and a computation system to convert the echo amplitudes into an image. The ultrasound transducer sends out a short burst of ultrasound, followed by a period of silence to listen for the returning echoes. For example, the ultrasound wave transmitted to the body strikes an interface and is partially reflected back to the transducer. The amplitude of the reflected echo depends on the ultrasonic properties of the tissue at the interface. The time between the sent and the received echo is used to calculate the depth of the interface assuming the speed of sound is constant throughout the media. The form of the transmitted wave is an amplitude modulated signal with a fixed carrier frequency determined by the probe. The returning echoes are rapidly sampled during the listening interval. The unprocessed samples are known as RF data. A set of RF data collected along one axial line is called an RF A-line (where A stands for amplitude). The set of RF lines (collected on a column-to-column basis) are referred to as RF-images throughout the thesis. Each RF data set is first amplified. The amplification increases as a function of time to compensate for the fact that echoes from deep locations are weaker and more attenuated than echoes from shallow locations. This operation is called time gain compensation (TGC) and is necessary to compensate for the attenuation of ultrasound in tissue. The next steps are envelope detection of the RF data and conversion to regular spatial coordinates (called scan conversion). An image formed this way is called a B-mode (Brightness mode) image or B-scan (see Figure 1.1) because the image pixels have brightness proportional to each amplitude [21, 61, 73].

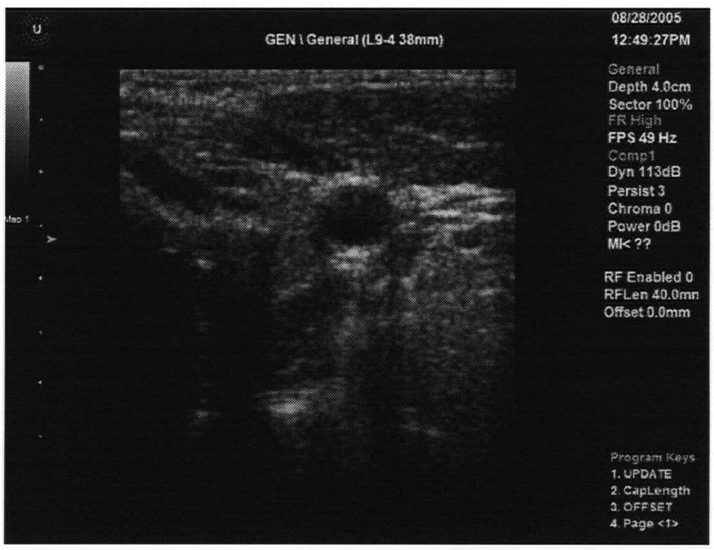


Figure 1.1: An example of the ultrasound B-mode image. The image is capture from the RP 500 ultrasound machine (Ultrasonix Corp., Burnaby, Canada).

1.3 Elastography Methods

Strain estimation methods based on ultrasound fall into two main groups depending on their external deformation method: 1) Methods where a quasi-static compression is applied to the tissue and the resulting components of the strain tensor are estimated; and 2) methods where a low-frequency vibration is applied to the tissue and the resulting tissue behavior is inspected.

The first method, strain imaging with a quasi-static compression [18, 75, 78–81], is performed by (i) obtaining a set of ultrasonic echo signals from a target, (ii) subjecting the target to a small axial deformation and (iii) obtaining a second set of echo signals. Motions along the direction of the applied load are estimated by performing piecewise motion estimation on corresponding pairs of signal segments (Figure 1.2). Strain estimation algorithms can then be applied to generate the strain images named elastograms. It has been shown that the strain distributions in tissues are closely related to the distribution of tissue elasticity [108]. Fast acquisition and/or computation are not an important factor in this kind of imaging, because the scatterer motion can be controlled by the applied compression and all the computation is done off line.

Strain imaging can also be performed by applying a dynamic compression (free hand deformation or low frequency vibration to the tissue) [62, 64, 65, 83, 84, 133, 135]. Ultrasound echo signals from the target are obtained simultaneously. The echo signals can be recorded for off line processing or they can be processed on the fly in real time. Elastograms are generated similarly to previous methods. Fast data acquisition is an important factor in this kind of imaging and, if real-time processing is preferred, fast computation is also necessary.

1.4 Axial Strain

Although images of axial strain, lateral strain, Poisson's ratio and Young's modulus can be generated [56, 78, 82] in this thesis we deal only with the estimation and imaging of axial strain which is called an elastogram.

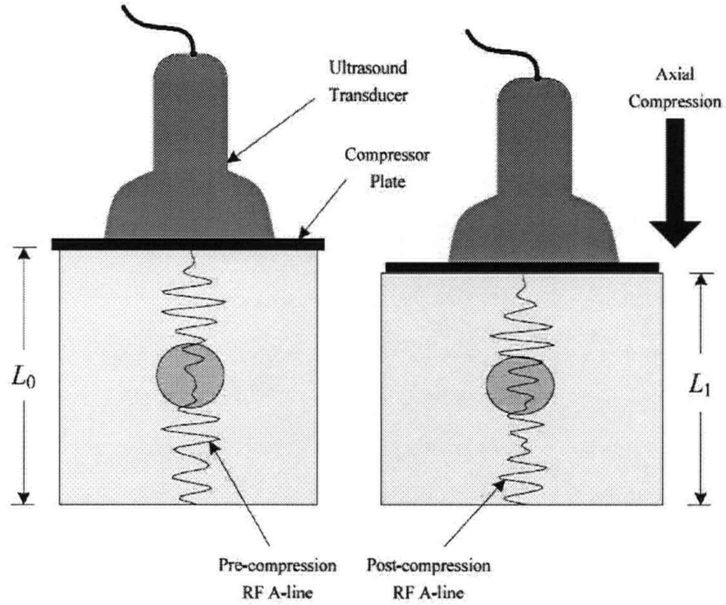


Figure 1.2: Illustration of the static elastographic process. Pre- and post-compressed tissue states are also shown. The figure is reproduced, with permission, from [38].

When an elastic medium, such as tissue, is compressed by a constant uniaxial stress, all points in the medium experience a resulting level of longitudinal strain whose main components are along the axis of compression. If one or more of the tissue elements has a different stiffness parameter than the others, the level of strain in that element will generally be different. A stiffer tissue element will generally experience less strain that the softer one. The axial strain is estimated in one dimension from the analysis of ultrasonic signals obtained from standard medical ultrasound. This is accomplished by (i) acquiring a set of digitized RF echo A-lines from the tissue region of interest, (ii) compressing the tissue with the ultrasonic transducer (or with a combination of transducer and compressor) along the ultrasonic radiation axis by a small amount (about 1%, or less, of the tissue depth) and (iii) acquiring a second, post-compression set of echo lines from the same region of interest. As explained before this approach can also be used in real time by acquiring continuous RF data from the tissue while an external deformation is applied to the tissue.

Corresponding echo lines are then subdivided into small temporal windows which are compared pair-wise by using motion estimation techniques, from which the change in arrival time of the echoes before and after compression can be estimated (Fig. 1.3). Due to the small magnitude of the applied compression in static compression, or high data acquisition rate in dynamic compression, there are only small distortions of the echo lines, and the changes in arrival times are small. The local longitudinal strain is then estimated as Equation (1.1):

$$e_{11} = \frac{(t_{1b} - t_{1a}) - (t_{2b} - t_{2a})}{t_{1b} - t_{1a}}, \tag{1.1}$$

where t_{1a} is the arrival time of the pre-compression echo from the current window, t_{1b} is the arrival time of the pre-compression echo from the next window, t_{2a} is the arrival time of post-compression echo from the current window and t_{2b} is the arrival time of the post-compression echo from the next window [81]. Therefore Equation (1.1) can be rewritten in the form of time-shifts as Equation (1.2)

$$s(i) = \frac{\Delta t(i) - \Delta t(i-1)}{\Delta T}, \tag{1.2}$$

where $t(i)$ is the time-shift between windows in the indexed segment pair and ΔT is the spacing between windows. The windows are usually translated in overlapping steps along the axis of the echo line, and the calculation is repeated for all depths. It has been assumed that speckle motion adequately represents the underlying tissue motion for small uniaxial compressions [18].

1.5 Previous Research on Strain Imaging

Following the seminal paper by Ophir et al [81], a number of methods have been developed and a number of methods have been borrowed from other research areas to evaluate the strain distribution due to an applied external force. Most methods fall into two main categories; gradient-based strain estimation and direct strain estimation [121]. Each of these categories is discussed in detail in following sections.

1.5.1 Gradient-based Strain Estimation

Axial strain in elastography is typically estimated with gradient-based methods from the gradient of the displacement (time-shift or phase-shift) estimates. Therefore motion tracking algorithms are the main part of these methods. Although gradient operation can be swapped with other strain estimation techniques, the term gradient-based methods are mostly used for this family of algorithms. Common displacement and strain estimation techniques are presented in the following sections.

1.5.1.1 Displacement Estimation Techniques

Motion estimation methods on ultrasound images were first used for blood flow estimation [12, 115] where a window in one RF A-line is tracked in its corresponding RF A-line (Figure 1.3). Time domain cross correlation was among the first methods [18, 37, 43, 47, 78–81, 110] in which the time-shift is estimated as the peak of the cross-correlation function between the pre- and post-compression RF A-lines (see Appendix B.1). Alternatively, cross correlation can be done on the envelope of the signals [59, 119, 120] instead of RF data to estimate the displacements. As the evaluated cross correlation function is spatially discrete, it must be interpolated to detect the exact position of the peak. In order to do subpixeling a number of interpolation techniques have been introduced in the literature (see Appendix B.2). The *quadratic parabola fitting* method was introduced in [37] which is equivalent to finding the zero crossings of the derivative of the signal. The *Sinc* interpolation was introduced in [16], and because of the sinusoidal nature of the ultrasound signal, the *cosine fitting* method was used in [16, 24]. If needed, the correct value of the correlations can also be recalculated to measure the accuracy of the motion tracking process at each window (see Appendix B.3).

In addition to time domain methods, a number of phase domain methods have also been introduced. These involve the estimation of the phase-shift from the complex cross correlation of the

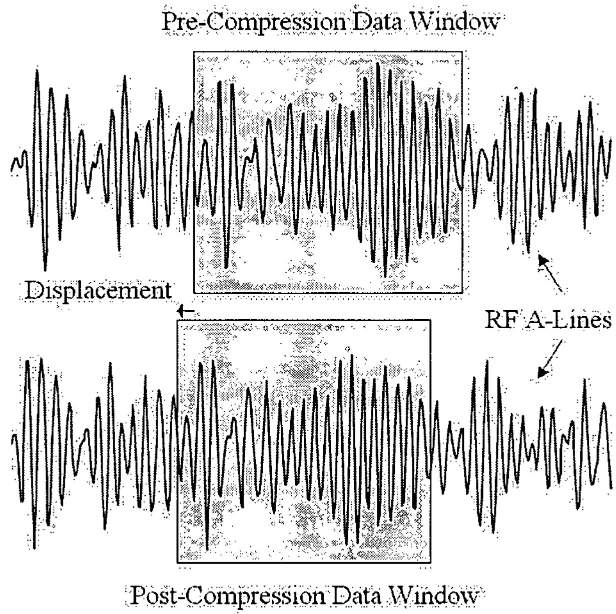


Figure 1.3: After dividing the RF A-lines to a number of windows the motion at each window is estimated by using a motion tracking algorithm. Taken from reference [38].

pre- and post-compression RF A-lines. Due to aliasing, phase domain methods fail when a single-step displacement exceeds a quarter of the ultrasonic wavelength. To avoid aliasing, phase unwrap methods have been introduced to extend the aliasing limit from $\lambda/4$ to $\lambda/2$ [75–77, 131]. It has been suggested in Phase Root Seeking (PRS) [90] to use *Newton's iteration* to unwrap the phase-shift (see Appendix A.1). To avoid aliasing, the Combined Autocorrelation Method (CAM) [102] suggested the use of the envelope correlation coefficient (see Appendix A.2). In [113] aliasing is simply ignored and the gradient operator is used directly on the phase-shift without converting the phase shift to time-shift. Because phase information is used directly, the author calls this method "direct strain estimation". According to the categorization in this thesis, the method falls into gradient-based methods since it uses the gradient operator on the phase shift estimates to estimate the strain.

A number of strain imaging algorithms, instead of using cross correlation, use feature extraction methods to find the displacement. The Zero Cross Tracking algorithm (ZCT) [109] identifies the zero-crossing instants of the pre- and post-compression RF A-lines. The sign change at the occurrence of the zeros is used to obtain two groups of zero-crossing locations (corresponding to a positive or negative sign change) for each RF A-line. A temporal tracking of such zero-crossings for successive steps or a cumulative accumulation of the displacements of the zero-crossings can be performed at each compression step. The distance between the zeros at each window is equal to the displacement estimate.

The Peak Search Algorithm (PSA) [34] is another motion tracking method that works similarly to ZCT but instead of finding the zeros of the RF signals it finds the peaks of RF signals. A peak assignment algorithm is then applied to assign each peak in the pre-compression RF data to its corresponding peak in the post-compression RF data. The shift between the peak in pre- and post-compression RF data is equal to the displacement estimates.

An iterative algorithm has also been introduced [117] that uses both time-shifts and local stretching factors to increase the accuracy of motion estimation. It estimates the displacements initially by doing time domain cross correlation. Then, local stretching factors are calculated according to the estimation of the displacements. Following that, stretching factors are applied to the pre-compression RF data to increase the correlation between the pre- and post-compression signals. The new pre-compression RF data is used in the next iteration to find the displacements again. Estimated motions at each iteration step are added to the previously estimated motion. The iteration stops after a predefined number of iterations or when a certain stopping condition is satisfied.

The staggered strain estimator as another iterative method was introduced in [110]. It estimates the strain in nonoverlapping windows at each step and shifts the windows by a small portion for the next step and estimate the strain again. The strain estimates for all such shifts are staggered to produce the final elastogram.

1.5.1.2 Strain Estimation Techniques

After finding the time-shifts with any of the above methods, the strain is then estimated according to Equation (1.2). Since estimating the strain from the motion estimate is a slope estimation problem, all previously used slope estimators can be used as strain estimators.

The gradient operator was used in the first strain estimation method [81]. It was used on the previously obtained time-shift estimates to estimate the strain. Since the gradient operator introduces additional amplification of the noise into the strain estimation process [60, 121], filtering methods such as averaging or median filtering, can be used before and/or after applying the gradient operator [57, 109, 118].

Alternatively the strain can be estimated by a least squares strain estimator [49] or higher order numerical differentiator with smaller error instead of simple gradient operator.

1.5.2 Direct Strain Estimation

Direct methods such as the *adaptive strain estimator* [4] and *spectral strain estimation* methods [59, 60, 121] estimate the strain directly without involving the use of motion estimation algorithms and the gradient operator.

The main idea behind the *spectral strain estimators* [59, 60, 121] is based on the Fourier scaling property, i.e., that a compression or expansion of the time-domain signal should lead to an expansion or compression of its power spectrum. Unlike gradient-based phase domain methods, spectral methods do not use the phase shift of the echo signals. They use the whole power spectrum.

The *adaptive strain estimator* [4] is another direct method that uses the local stretching factors as an estimator of the local strains. It uses an iterative algorithm that adaptively maximizes the correlation between the pre- and post-compression echo signals by appropriately stretching the latter. The final stretching factors are then reported as local strains. This method is different from [107, 117], where they use the stretching factor to improve the correlation between pre- and post-compression signals and then estimate the displacement. Here, the stretching factors themselves represent the local strain.

The correlation coefficient itself can be used to estimate the strain directly. The correlation between the pre- and post-compression echoes can decrease with applied strain. However, decorrelation itself has been used to estimate delay and/or time-shifts. Various researchers used the correlation coefficient to estimate tissue motion [28,116]. In [6], it was proposed to use the decorrelation coefficient (1 minus the correlation coefficient) for the envelope signals for free hand elasticity imaging but it was demonstrated in [122] that the correlation coefficient has a poor precision as a strain estimator. Because of its speed and simplicity, this estimator may be a valuable tool for free-hand real time elasticity imaging. However by using simulated data, it was also shown in [2] that changes in the center frequency and SNRs introduce unknown errors in the correlation coefficient.

1.5.3 Multi Dimensional Strain Imaging

Typically in elastography, the axial component of the strain is estimated by taking the gradient of the axial (along the beam propagation axis) displacement occurring after a quasi-static tissue compression [18, 78–81]. In general, however, the tissue motion that occurs during compression is three dimensional (3D). Because the lateral (perpendicular to the beam propagation axis and in the scan plane) and elevational (perpendicular to the beam propagation axis and to the scan plane) motions are not measured, two major disadvantages may occur [56]. First, the axial elastogram takes into account only a small part of the mechanical tissue motion information. Second, undesirable lateral and elevational motions are the primary causes of signal decorrelation [51, 52].

The lateral or elevational decorrelation can be prevented by appropriate confinement of the tissue under study [51, 52]. However, the confinement may not always be practical in clinical applications, especially when the tissues under study are not easily accessible. Therefore, confinement of the tissue has not been considered in this thesis.

Initially, multidimensional motion estimation started in the field of blood velocity estimation, to estimate velocity components other than the axial one. Later, in the field of elasticity imaging, some of the methods developed in the flow measurement field were borrowed to find axial, lateral and even elevational components of the strain.

Many papers in the literature have dealt with the problem of motion estimation in 2D [10, 12, 13, 19, 29, 56, 71, 93, 115, 135], that try to find components of the motion in the scan plane (axial and lateral), and in 3D [13, 42, 55, 58, 63, 68, 74], that try to find all the spatial components of the motion (axial, lateral and elevational). These two categories are discussed in detail in the following sections.

1.5.3.1 2D Strain Estimation

Similar to 1D strain imaging, time domain cross correlation can be used in 2D motion estimation by tracking the motion once in axial direction and once in lateral direction independently [56] or it can be extended to 2D block matching instead of 1D vector matching [10, 12, 19, 42, 115]. 2D block matching methods detect the peak position of the 2D cross-correlation function evaluated with respect to each local RF echo data. Because of the trade-off between the smoothness of motion tracking and spatial resolution depending on the size of the 2D block, it was suggested in [135] to use a hierarchical coarse-to-fine approach in the selection of the sizes of matching blocks.

A method similar to 2D block matching, called 2D companding [19], was developed for elastography to reduce decorrelation noise in elastograms using applied axial strains and correcting for the lateral decorrelation (elevational motion was ignored). It uses speckle block-matching techniques [12, 115] to track the 2D motion of the scatterers after static compression. This method has recently been extended to 3D companding [20, 46], in order to compensate for both lateral and elevational displacement.

In [70] the speckle-tracking technique was used in lateral direction. They showed that lateral displacement images obtained by traditional 2D correlation tracking are noisy. They also showed theoretically that, when envelope data are used to track lateral motion, the variance of the lateral displacement estimation is larger than the variance of the axial displacement estimate by orders of magnitude. Based on these results, they proposed the estimation of lateral displacement by making the assumption of isotropic incompressibility in the tissue and, thus, making use of the precision of the axial measurement.

In [56, 58] the lateral component of the strain was estimated by tracking the phase of single A-lines in the lateral direction through the use of a weighted interpolation method. Since the resolution in the lateral direction is poor, they used interpolation techniques to reconstruct the

RF signal in the lateral direction [56,70]. Accurate estimation of lateral strain made it possible to estimate the Poisson's ratio, which is defined as the ratio of the lateral strain over axial strain, and is an important mechanical property of tissue [56,97,98]. It has been shown that, if there is a strain contrast between an inclusion and the background, the Poisson elastogram is able to show whether this strain contrast is due to a Poisson's ratio contrast or an elastic modulus contrast. Furthermore, by detecting the motion in the lateral and/or elevational direction, correction algorithms were used to increase the accuracy of motion detection in the axial direction [56,58].

In addition to 2D time domain methods, a number of 2D frequency domain methods have also been introduced that involve the estimation of the phase-shift from the complex correlation of the pre- and post-compression RF images. A phase correlation method was used in [131] to find the motion in each block. The 2D Fourier transform of a 2D block in pre-compression RF data was multiplied by the conjugate of a 2D Fourier transform of a 2D block in post-compression RF data, and the result was normalized by the norm of the multiplication. The inverse Fourier transform was then applied to the result to find the estimated motion in a 2D window. Alternatively an iterative phase matching method was introduced in [112], which uses the 2D phase characteristics of the pre-compression RF as the index to iteratively search for the corresponding local data in the post-compression RF data.

1.5.3.2 3D Strain Estimation

2D motion tracking methods are used for estimating the motion only in axial and lateral direction in the plane of ultrasound image. In order to find all three components of the motion, elevational motion is also need to be estimated. Two distinct estimators for tracking elevational motion can be developed using (i) a single plane or (ii) multiple planes.

Among the methods that use a single plane to track the elevational motion, in [7] the correlation coefficient of the signal envelope was used to generate lateral displacement images. As a similar approach when only one plane is available, motion in the elevational direction can be tracked by using the decorrelation information [66]. But these approaches have several limitations.

To track the motion in multiple plane one needs to capture multiple planes. There are three distinct ways in which multiple planes can be generated in the elevational direction. Firstly, this may be done by displacing a 1D linear array in the elevational direction and acquiring successive frames at different parallel elevational planes. This method will most probably not be workable in an experimental setting and medical applications since it is not always possible to keep the tissue steady for a long time. Secondly, a 1.5D array can be used to provide three to four elevational planes by firing different sets of elements in the elevational direction. This method may yield the elevational component. Lastly, a 2D (or $N \times N$) array may be used, providing as many planes in the elevational direction as there are beams in the lateral direction. In this case, the elevational motion can be estimated in a plane perpendicular to a certain scanning plane.

Assuming that multiple planes of ultrasound images are available, previous motion tracking methods can be extended to 3D. An interplanar tracking method, similar to the one used for interbeam lateral tracking in [56] was applied in [58] to interpolate the RF line between the original beams to increase the resolution in both lateral and elevational direction. Axial, lateral and elevational motions are found independently by using 1D time domain cross correlation and correction algorithm is used to recorrelate all components of the motion. It should be noted that, although interpolating RF lines may increase accuracy, it is very time consuming and it is not always applicable if the computational cost is an important factor.

In [63, 68], 2D block matching is used in each plane and its neighboring planes to find and correct for both lateral and elevational motions of the current plane. The PRS, which is a phase domain algorithm, is then used in the axial direction to find the axial component of the motion. If the 2D phase array ultrasound probe is used to acquire RF data, the CAM method can be used to find the motion in 3D [74].

1.6 Comparison Metrics in Elastography

The entire elastographic process can be described using three fundamental concepts [127]. The block diagram is shown in Figure 1.4. The first concept is the contrast transfer efficiency (CTE) [48, 91], that determines the distribution of the ideal local axial strains in the compressed tissue according to its intrinsic tissue parameters. CTE is independent of the imaging system and strain estimation algorithm and it describes the fundamental relationship between modulus and strain contrasts.

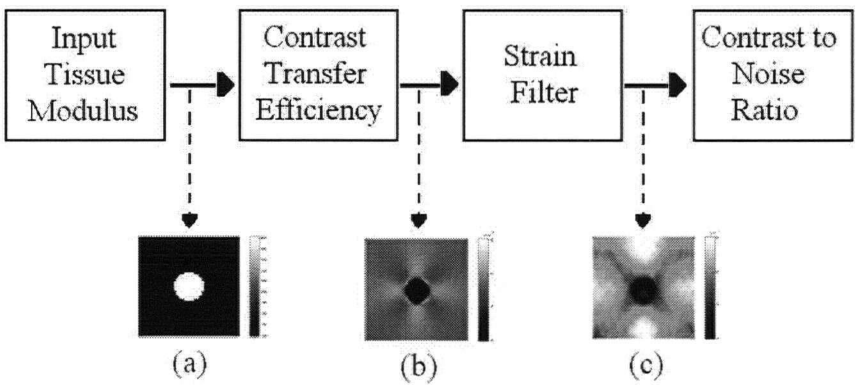


Figure 1.4: Simplified block diagram of the elastographic process. The compression, together with the intrinsic stiffness distribution (a) in the tissue and the boundary conditions, causes a strain distribution to be set up in the tissue (b). The statistical properties of the sonographic tissue motion estimation process give rise to the SF. The strain distribution filtered by the SF is the elastogram (c).

The second concept of the elastographic process characterizes the formation of the elastogram via the use of ultrasonic pulse-echo techniques. This process is described in terms of a general theoretical framework, namely the strain filter (SF), which characterizes the noise properties of the elastographic system [51, 106, 123, 125]. The SF is a statistical upper bound of the transfer characteristic that describes the relationship between actual tissue strains and the corresponding strain estimates corrupted by noise effects. It describes the filtering process in the strain domain, which allows quality elastographic depiction of only a limited range of strains in tissue. The SF provides upper and practical performance bounds for strain estimation and their dependence on the various system parameters. The SF is formed by plotting the upper bound of the elastographic signal-to-noise ratio (SNRe) [106, 123] for each estimated value of the tissue strain at a given axial and lateral resolution (R_{axial},$R_{lateral}$) [15, 96, 99]. The SF allows only a restricted range of strain values to be included in the elastogram and has a band pass characteristic.

A sample of the SF is shown in Figure 1.5. The sensitivity (smallest strain that can be estimated correctly) of the system is limited on the low end by sonographic noise effects. The saturation, (largest strain that can be estimated correctly) of the system is limited on the high end by signal decorrelation, resulting in a band pass filter in the strain domain with a specific dynamic range (DRe). The deviation of the SF from an ideal all-pass characteristic in the strain domain is due to the ultrasound system parameters, the finite value of the sonographic SNR (SNRs) and the effects of signal decorrelation. The SF can be modified and improved upon within certain limits by proper selection of the ultrasonic system parameters and by algorithms used for signal processing (motion tracking and strain estimation). The trade offs between achievable SNRe and the R_{axial} in elastography has been explored in [111].

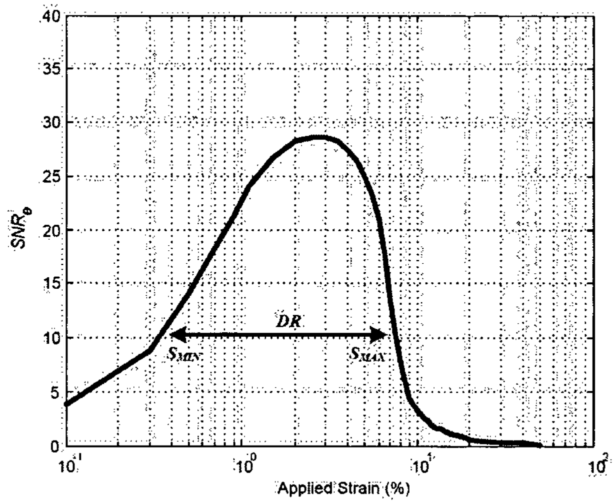


Figure 1.5: A sample of strain filter is shown. For clarification dynamic range, sensitivity and saturation have also been shown on the figure.

Finally, the third concept encompasses the contrast characteristics of the image produced, the elastogram, whose lesion detectability is characterized by the contrast-to-noise ratio (CNRe) parameter [11, 126]. Details of the fundamental concepts of elastography are given in the following chapters.

Based on the comparison metrics that were introduced in the SF (SNRe, DRe, sensitivity and saturation), the performance of different strain estimation algorithms have been studied. Gradient-based strain estimators that use RF data generally have the advantage of being highly precise and sensitive and are suitable for estimating small strain with high accuracy [60,123]. However the strain filter (SF) study has shown that they are not very robust in the presence of even a moderate amount of deccorrelation between the pre- and post-compression signals that mostly happens at high compression ratios [52,120]. Therefore, these methods mostly do not have a large DRe and fail to estimate high strains. In contrast, direct strain estimation methods such as *adaptive strain estimation*, *spectral strain estimators* [4,59,60,121], and gradient-based strain estimators that use the envelope of the signals [59,119,120], are not sensitive to small strains. However the SF study has shown that they are robust in the presence of the noise and decorrelation and mostly have a larger DRe and are capable of estimating very large strains.

1.7 Performance Optimization in Elastography

Signal decorrelation due to tissue compression is a significant source of error in tissue displacement estimates. This reduction in the value of the correlation can have a significant impact on the performance of the strain estimator [123,127]. Similar to 1D companding which is a noise-reduction technique (that applies single compression at the transmitter and complementary expansion at the receiver) temporal stretching is widely used in strain imaging with a static deformation to improve the SNRe and DRe [3,78,109,124]. After applying a known compression to the tissue (1%), the post-compression signal is stretching globally (1%) to remove the mechanical artifacts. Temporal stretching was extended in 2D [19] and 3D [20] motion tracking to remove the decorelation noise due to lateral and elevational motions [51]. After the correction process, axial strain estimation is applied to the signal.

Logarithmic amplitude compression is another technique which is used widely [18,90] that achieves the desired reduction of the variability of the RF echo signals without sacrificing the resolution and reduces the uncertainty in the time shift estimation. In order to implement logarithmic compression the whole RF A-lines needs to be compressed which is computationally intensive.

Multicompression is another performance optimization technique that suggests dividing a big static compression to several small compressions [118]. Averaging multiple displacement or strain estimates reduces the variance by a factor of N (where N is the number of estimates) and improves the SNRe by a factor of $\sqrt{N}$ [118]. Since the strain between the RF data can be kept within the DRe, if the multicompression technique is used, the SF will have the shape of a high pass filter instead of band pass [109, 118], which is close to ideal strain estimator. A similar method was introduced in [57] that selects the optimal strain corresponding to each compression level according to its value in the SF. High strain regions are selected from small compressions and low strain regions are picked from high compression regions and a final strain image is then generated. Similarly to this idea in real time elastography, temporal filtering can be used to generate the elastogram from previously generated strain images [88].

As mentioned before, typically in elastography, time-domain techniques are used that involve the computation of the time-shift to estimate the displacement following an applied compression, and the estimation of the strain by applying gradient operations on the previously obtained time-shifts. Since the gradient operation introduces additional amplification of the noise into the strain estimation process the Least Squares Strain Estimator (LSE) was introduced in [49]. They proposed to use a piecewise approach in order to reduce the variance of the measured strain profile. The improvement in the SNRe was also shown in the same paper.

Post processing of the strain images makes it possible to improve the quality of elastograms even further. The median filter is normally used to reduce noise in an image, somewhat like the mean filter [72, 86]. However, it often does a better job than the mean filter of preserving useful detail in the image. Median filters with 1D and 2D kernels are widely used to remove the noisy regions in elastograms [109].

1.8 Real-time Strain Imaging and Its Computational Issues

Most strain estimation methods are used for off-line processing and are typically compared according to their signal-to-noise ratio (SNRe), contrast-to-noise ratio (CNRe), dynamic range (DRe), sensitivity and resolution of elastograms [96, 99, 111, 123, 126, 127]. For implementations that preserve the real time aspect of ultrasound imaging, the computational efficiency of such algorithms is also important. However, very little quantitative information is available on timing and computational efficiency issues in elastography. Although the term *time efficient* has been widely used recently [89,103,109], a formal definition has not been used in the literature to compare different methods.

A lot of work has been done to have a real-time elastography system that has a high clinical utility and to preserve the real-time aspect of ultrasound imaging. In order to do high frame rate ($\geq$ 20 fps [88]) processing, the computational cost of the signal processing algorithms becomes an important factor. Furthermore the amounts of strain between the sequences of RF data are very small ($< 1\%$) with negligible decorrelation noise [87] therefore a sensitive strain estimator is preferred. It should be noted that small strains lead to low SNRe strain images [123], however, multiple successive frames of strain images can also temporarily be filtered for an improvement in SNRe [87].

Direct strain estimators are mostly computationally intensive and are not sensitive to small strains. Therefore for now, they do not have a potential for real-time applications. The current implementation of the adaptive strain estimator takes an hour to generate a single elastogram.

Gradient-based strain estimators are better suited for real-time purposes due to their sensitivity and speed. Since estimating the strain from the displacement estimates has a negligible cost (using a simple differentiator or LSE), the motion estimation process becomes the most important step in real-time elastography.

In order to speed up the motion tracking process, a number of hardware architectures have been introduced in the literature [44,53,92,101]. Using just one digit of the RF data instead of all digits to estimate the motion was another approach that was introduced in [17]. They showed that if just one bit of the RF data is considered (sign function in match) it is still possible to estimate the motion. This way the summation and multiplication operator can be replaced with simple XOR and AND gates in hardware. Based on this theory, a number of real-time correlation estimators architecture have been introduced [136]. Further research showed that one bit resolution for motion estimation is not enough for accurate results [78–80,82]. Therefore, these architectures are not used anymore for elastography.

A number of researchers have claimed methods that are capable of real-time motion tracking, such as the Zero Cross Tracking (ZCT) method [109], Peak Search Algorithms (PSA) [34]. However, implementations have not been reported yet.

1.8.1 Current Real-Time Motion Tracking Algorithms

Research on accurate real time strain imaging and motion estimation algorithms has focused mainly on phase domain algorithms that have been shown to be capable of rapid determination of displacement. As mentioned before due to aliasing such methods fail when a single-step displacement exceeds a quarter of the ultrasonic wavelength. To avoid aliasing, phase unwrap methods have been introduced to extend the aliasing limit from $\lambda/4$ to $\lambda/2$ [77]. These include Phase Root Seeking (PRS) [90] that uses Newton's iteration to unwrap the phase-shift and Combined Autocorrelation Method (CAM) [102, 103] that uses the envelope correlation coefficient to avoid aliasing.

PRS was used by the LP-IT company (Lorenz and Pesavento Ingenieurburo fur Informationstechnik, Bochum, Germany) and was the first algorithm to calculate strain images of significant size in real time [89]. For the world's first real time implementation of strain imaging, which allows strain imaging in a clinical setting, they won the 2nd award of the Innovation Reward Ruhrgebiet in November 2000. Currently they can generate up to 30 frames of strain per second. CAM is another strain imaging method that achieves real time performance. The CAM method is used by Hitachi (Hitachi Medical Corporation, Chiba, Japan) and currently they are able to generate up to 20 frame of strain images per second [104, 105].

1.9 Vibro Elastography

Since strain is not an intrinsic property of soft tissue and its value at each point and each time depends on a number of parameters such as mechanical properties of tissue and amount of compression, a number of different methods have been introduced to produce images of more stable properties of tissue.

Vibro Elastography (VE) is a new imaging system that estimates the elastic properties of soft tissue [117]. The tissue is vibrated externally with known frequency components and a high frame rate ultrasound machine is used to image the tissue. A motion estimation method is then applied to the captured RF data and estimated displacements are buffered. After this step, the tissue motion is analyzed with a Transfer Function. According to this method, the tissue dynamics between two locations along the axis of motion is considered as a linear dynamic system. The transfer function between the two locations is obtained by spectral analysis with the buffered estimated motions used as inputs and outputs. The shape of the transfer function can then be analyzed further. The stiffness of tissue can be estimated from the magnitudes of the transfer function at low-frequencies.

Similar to strain imaging, VE uses the information of the displacement estimates and motion tracking plays an important rule in VE, but in contrast to strain imaging, motion tracking is not the only computationally intensive part in VE. Transfer function analysis is another time consuming part in the VE and because of these two parts, until now, VE was only used off-line and processing was only done on the previously captured RF data.

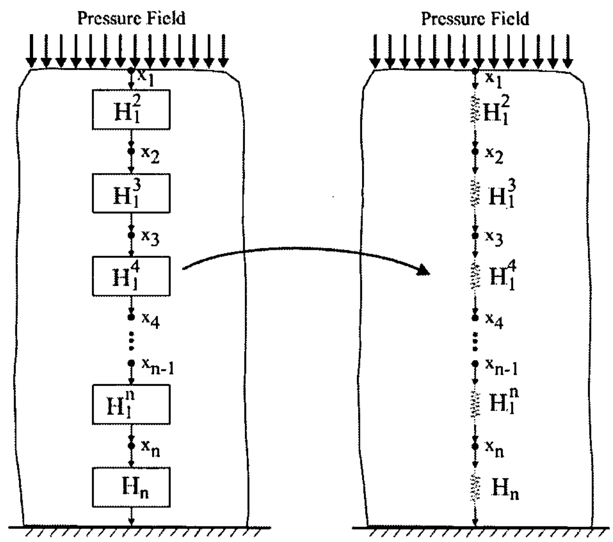


Figure 1.6: Sketch showing the simplification of a linear dynamic system into a static one under low frequency excitations [117].

1.9.1 Transfer Function Analysis

In Figure 1.6, the sequence of tissue displacements at regions i and j are $x_i(t)$ and $x_j(t)$ respectively. These displacements are the inputs and outputs of a linear dynamic system shown as H_j^i in Figure 1.6. The transfer function between element i and element j is calculated with standard techniques [67]. First, the power spectral density $PSD_{x_i,x_i}(\omega)$ of element $x_i(t)$ is computed. Then the cross spectral density $CSD_{x_i,x_j}(\omega)$ between elements i and j is computed [117]. After that the complex transfer function is computed as follows [132]:

$$H_i^j(\omega) = \frac{CSD_{x_i,x_j}(\omega)}{PSD_{x_i,x_i}(\omega)}, \tag{1.3}$$

The value of the transfer function is then averaged in the range of frequency that we are interested in. The shape of the transfer function can then be analyzed further. Mechanical properties of the tissue such as stiffness, damping and mass can then be estimated from these transfer function values. For example, the stiffness k_j of tissue at region j can be estimated from the magnitudes of the transfer function at low-frequencies:

$$k_j = \frac{k_1(1 - H_1^2)}{H_1^j - H_1^{j+1}}. \tag{1.4}$$

Dynamic elastography methods have been well studied in [117]. VE is a model-free method, which uses spectral analysis of the tissue motion. It investigates the behavior of the tissue at different frequencies and finds the frequency response of the tissue. Both static and dynamic parameters of the system can then be extracted by focusing on different ranges of frequencies. This way VE eliminates the need for solving the inverse problem to identify the parameters of the system. Solving the inverse problem is computationally intensive and not applicable in real time. Furthermore, the implementation of the transfer function according to Equation (1.3) makes the algorithm very immune to the noise while most parameter identification methods are very sensitive to it.

1.10 Thesis Objectives

The required processing components of both strain imaging and stiffness imaging is shown in Figure 1.7. The goal of this work is the real time implementation of these packages which are used to measure the mechanical properties of soft tissue including static and dynamic properties based on ultrasonic measurements of tissue motion under free hand deformations or low frequency mechanical vibrations.

To have both systems in real time, all blocks need to be optimized. Since the motion tracking block is the most essential part of both systems, a fast and accurate motion tracking algorithm that can be used for all elastography methods that require displacement estimation is provided. To validate the algorithm from its time-efficiency point of view, the computational cost of the introduced algorithm needs to be compared with current real time algorithms such as PRS [90] and

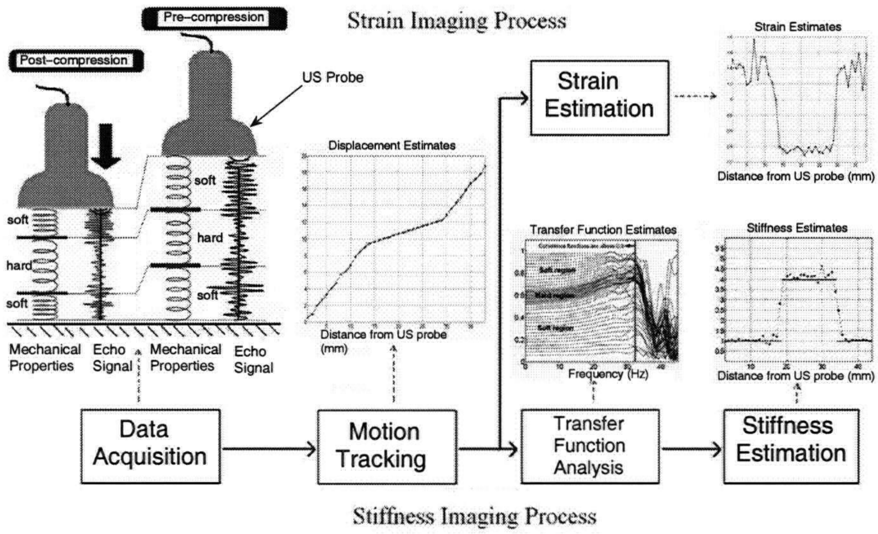


Figure 1.7: Sketch showing the simplification of both strain imaging and stiffness imaging process.

CAM [102, 103]. In addition to the computational cost, the motion tracking algorithm also needs to be validated according to its accuracy of estimating the motion to make sure it is both fast and accurate.

Following the motion estimation block in the strain estimation process and following the transfer function analysis in stiffness imaging process, strain and stiffness need to be estimated (Figure 1.7). Therefore current estimators like the gradient operator [81] and the least squares estimator [49] need to be studied. Similar to the motion tracking block, strain and stiffness estimation methods need to be fast and accurate to be applicable in real time.

A simulation environment and an experimental set-up need to be provided to validate the proposed real time imaging systems based on important metrics, namely signal-to-noise ratio [51, 106, 123, 125], resolution [15, 96, 99] and contrast-to-noise ratio [11, 126]. Simulation results will help us to determine the best parameter settings for each block of the system. Following that, experiments will validate the performance of the whole system, mainly its motion tracking block with real data when tissue mimicking materials with known mechanical properties are externally deformed and imaged with an ultrasound system.

Finally both strain and stiffness imaging packages need to be implemented and tested for real time performance that can be used in clinical applications.

1.11 Thesis Outline

In this chapter, a review of the current elastography approaches and motion tracking methods in successive ultrasound images was given.

Chapter 2 presents the standard *Time Domain Cross Correlation* (TDE) technique as a motion estimation method. After discussing the short comings of TDE, a new technique is introduced that we call *Time Domain Cross Correlation with Prior Estimates* (TDPE) since the main structure of the algorithm is similar to that of the TDE method but it is faster and more accurate.

Chapter 3 discusses the computational cost of the motion tracking algorithm. It starts with an introduction of two popular motion tracking algorithms that achieve real time performance. A comparison key is then introduced and is used to compare the introduced method in Chapter 2 with current available real-time methods.

In chapter 4, a number of strain estimation methods are introduced. The chapter starts with a standard gradient operator which is the fastest and simplest strain estimator. It then introduces other versions of the differentiator that have smaller error. Finally, the Least Squares method is explained as another option to estimate the strain. The chapter finishes by comparing these strain estimation algorithms.

Chapter 5 presents the simulation study of the TDPE algorithm. The chapter starts by quantitative evaluation of the TDPE method by means of signal-to-noise ratio ($SNRe$), dynamic range (DRe), sensitivity, contrast-to-noise ratio ($CNRe$) and axial resolution (R_a) using simulated RF data. Furthermore the performance of the median filter and least squares strain estimation and their effect on SNRe, CNRe and resolution with the available trade offs are investigated.

In Chapter 6, two types of tissue mimicking materials (phantoms) are tested to validate the TDPE algorithm with real RF signals from ultrasound machine. The chapter starts with a description of the experimental setup. Then, the construction of two types of tissue mimicking materials is detailed. The experimental results are presented next, and remarks on the results conclude the chapter.

In Chapter 7, the current version of both strain imaging and stiffness imaging packages are explained. The flow chart of both programs is shown and details about each box in the flow chart are explained. The interface of the current version of both packages with their features is demonstrated and the same tissue mimicking phantoms that were used in previous chapter to generate the elastograms are used to validate the performance of the software. The chapter finishes by showing the current images that are generated in real time from the tissue mimicking materials and real prostate tissue in *vivo* when external deformation is applied to the phantoms.

Finally Chapter 8 presents the conclusions of the research together with suggestions for future work.

Chapter 2

Motion Estimation

In this chapter, first the standard *Time Domain Cross Correlation Estimator* (TDE) technique is presented as a motion estimation method. TDE was among the first algorithms used to perform speckle tracking and its properties have been well established using the strain filter, SNRe, DRe, sensitivity, resolution and CNRe measures [96, 99, 111, 123, 126, 127]. After discussing the shortcomings of TDE, a new technique called *Time Domain Cross Correlation with Prior Estimates* (TDPE) is introduced. The main structure of the algorithm is similar to that of the TDE method but TDPE performs much faster and is more accurate.

2.1 Standard Time Domain Cross Correlation Technique

The standard time domain cross correlation (TDE) was the first algorithm used to perform speckle tracking [81]. With this technique small displacements between pairs of ultrasonic images that are acquired under different axial compressions are determined using a cross-correlation analysis of the corresponding A-lines of an RF-data set.

A-lines of RF data are split into a number of overlapping windows with a specific length (Fig. 2.1). The local axial displacements are then computed by cross correlating two corresponding windows over a predefined search area (Fig. 2.2(b)). A peak in the correlation occurs at the actual displacement.

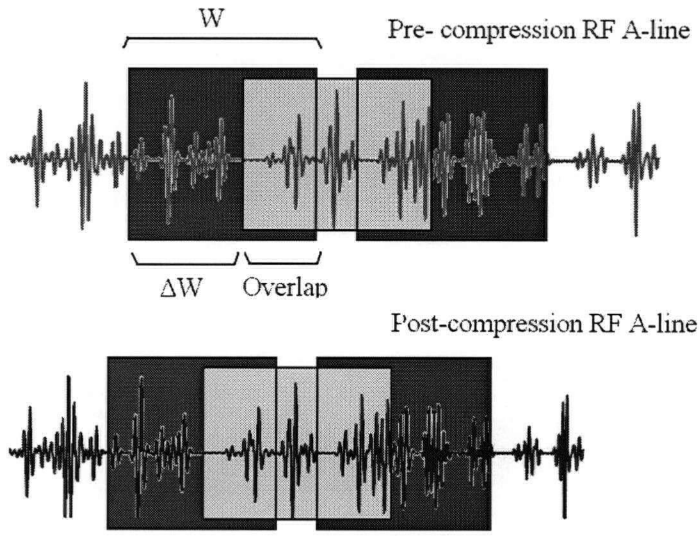
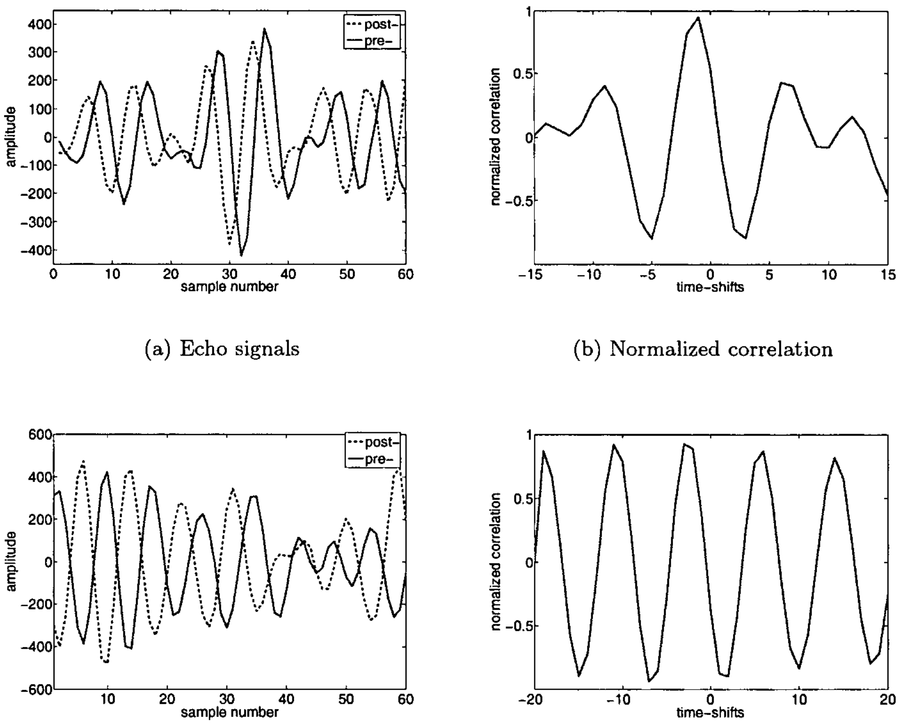


Figure 2.1: The TDE method splits the pre-compression RF A-line into a number of overlapping windows and finds their corresponding windows in the post-compression RF A-line by cross correlation method. The important parameters are shown in the figure. W is the length of each window and ΔW is the shift between windows. Some researchers use the overlap between windows instead of ΔW as a signal processing parameter.

The exact position of the peak is usually calculated by subpixelling techniques to avoid resampling [16]. After this step a gradient operator or least squares strain estimation (LSE) algorithm [49] is used to calculate the strain from the displacement estimates (Equation (1.2)).

The size of the search region is one of the parameters that needs to be initialized in the TDE algorithm and it depends on the range of possible motion. For small compression ratios the search region can be set to be small since the motions are also small but for higher compression ratios the search region should be selected much larger since the TDE fails to find the motions that happen outside of its search area.

The temporal stretching of the post-compression A-lines is typically done to undo the effects of the mechanical compression on the signal [123, 124, 127]. As explained in the previous chapter, this stretching improves the correlation between the pre- and post-compression A-lines and reduces the strain noise which leads to higher SNRe and DRe.



(a) Echo signals

(b) Normalized correlation

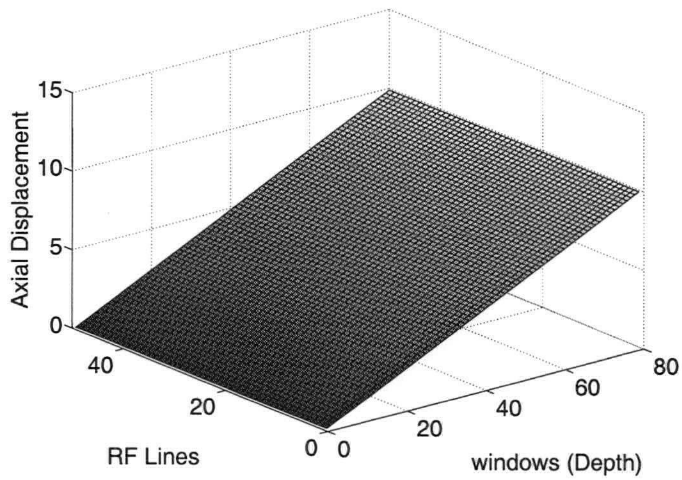
(c) Echo signals

(d) Normalized correlation

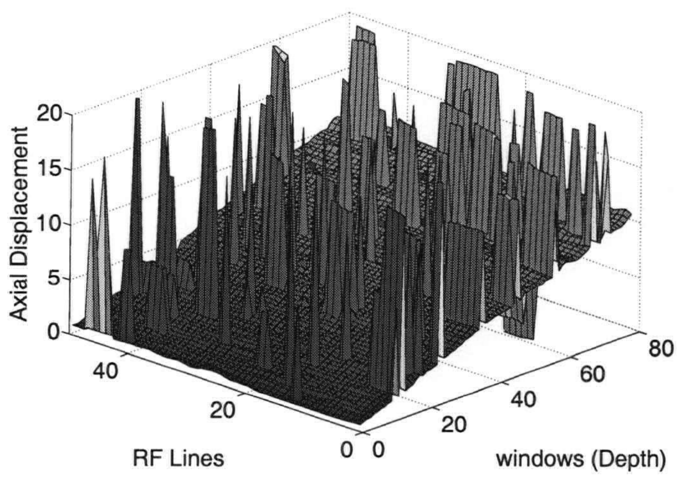
Figure 2.2: Pre- and post-compressed RF data (Left), and the corresponding normalized cross-correlations (Right). The peak in the correlation function shows the displacement.

2.1.1 Pros and Cons of the Standard Method

The properties of TDE have been well established in the literature [96, 99, 111, 123, 126, 127]. TDE is easy to implement. Its Strain Filter [123] shows that it is sensitive to small strain and has a high SNRe for low strains which is what we need in real time applications. For higher strains, because of de-correlation noise, the method often fails to find the displacement correctly. Therefore, if estimating the strain at high compression ratios are required TDE will not perform very well.

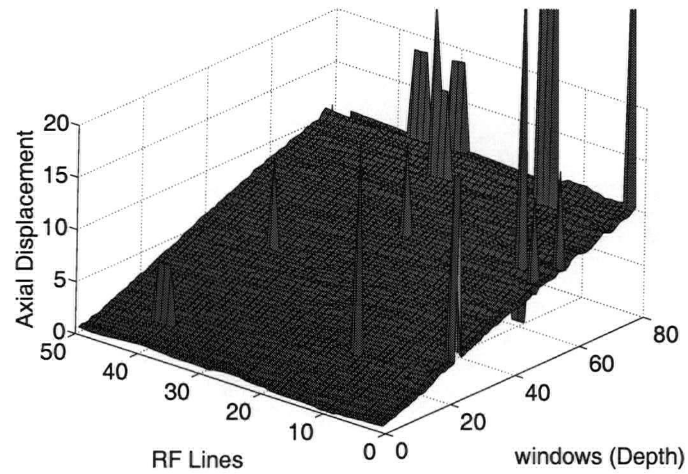


(a) Ideal displacement image on a pair of simulated RF frames on a homogeneous tissue.

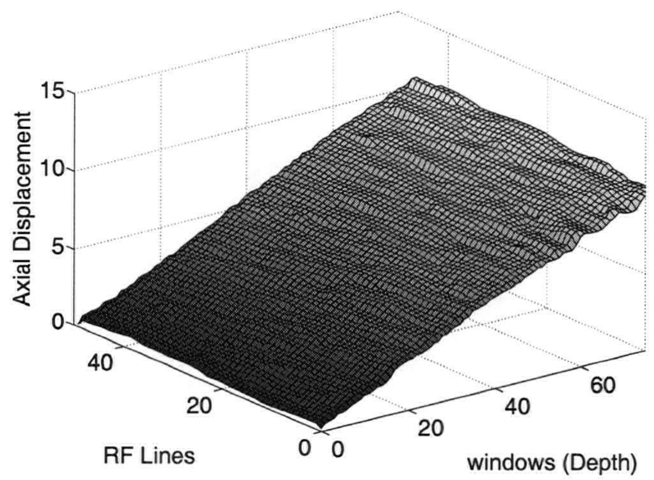


(b) Estimated displacement image with standard TDE with cross correlation.

Figure 2.3: Theoretical displacement image on a pair of simulated RF frames on a homogeneous tissue (Top) and estimated displacement image on a pair of simulated RF frames when TDE with cross correlation is used (Bottom). The z-axis shows the axial displacement, the y-axis shows the RF A-lines and the x-axis shows the windows starting from the window proximal to the probe to the distal window deep in the tissue. Large peaks are the result of selecting false peaks. The image shows how often does this happen when non-normalized cross correlation is used.



(a) Estimated displacement image with standard TDE when normalized cross correlation.



(b) Estimated displacement image with standard TDE with normalized cross correlation following a 2D median filter.

Figure 2.4: Estimated displacement image on a pair of simulated RF frames when TDE with normalized cross correlation is used (Top) following a 2D median filtering with a kernel size of 3x3 (Bottom). The remaining false peaks are removed by using a median filter.

The dynamic range of 0% to 2% is enough for the high frame rates typically encountered in real time strain imaging since the strain between consecutive sets of RF data is usually small and as the frame rate goes up the required dynamic range becomes smaller. The only reason why TDE is not used in real time applications is its computational inefficiency. Indeed, since TDE cross-correlates two signals over a large area to find displacements, it is slow.

Searching over a large area also causes another problem that is referred to in most papers as *false peak* [130] or *false match* [85] and that is similar to aliasing in the time domain. Since the RF data are essentially sinusoidal in shape, their cross correlation is also similar to a sinusoid. Thus searching over a large area may cause another large peak to appear in the correlation coefficients (Fig. 2.2(d)). Choosing the wrong correlation peak leads to false matching which appears in the displacement vector as a sharp peak and causes the displacement images to become noisy (Fig. 2.3(b)). The problem of finding false peaks can be alleviated somewhat if the normalized correlation [9, 127, 130] is used rather than the cross correlation itself (Fig. 2.4(a)). However, the normalized cross correlation does not eliminate all false peaks and makes the algorithm slower.

For further improvements in displacement estimation, usually the normalized correlation is followed by a 1D or 2D median filtering [23] operation in order to remove the remaining false matches and noisy regions [96, 109]. This improves the performance of the method but adds more overhead and makes the algorithm even slower (Fig. 2.4(b)).

2.2 Time Domain Cross Correlation with Prior Estimates

Although TDE is easy to implement and very accurate when the cross-correlation function has a single well-determined peak, it is not suitable for real time applications since it is time consuming. The inefficiency of the algorithm and the problem of finding false peaks are TDE problems that are related to each other. Both result because the cross correlation is performed over a large interval. Since correlation is just local information, it is not always sufficient to find displacements accurately. Therefore TDE is slow and error-prone.

A new motion tracking algorithm is introduced in this chapter. We call it *Time Domain Cross Correlation with Prior Estimates* or TDPE since the main structure of the algorithm is similar to that of the standard TDE method. TDE uses previous estimated data to predict the motion at a current position. This prediction process makes the algorithm much faster, and since all the predictions are based on boundary conditions and internal constraints, the estimated motions are more accurate.

The TDPE algorithm exploits the fact that neighboring windows in an RF frame correspond to small regions that are close to each other (Fig. 2.5) so their displacements should also be close to each other. This property can be seen in all displacement estimate figures (Fig. 2.3(a) 2.3(b) 2.4(a) 2.4(b)).

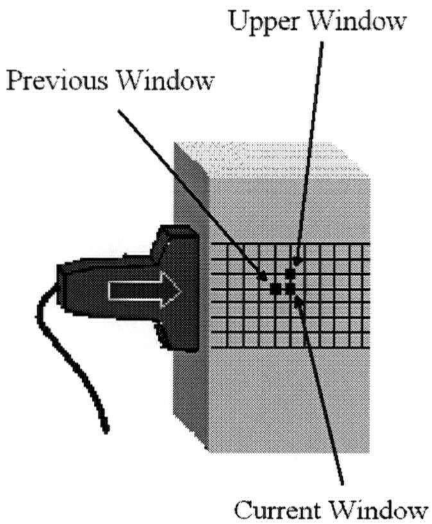


Figure 2.5: In computing the displacement of the center window, TDPE exploits the fact that displacements of two of its neighbors (leading window to the left and upper above) have already been calculated.

TDPE is similar to TDE, but it collects information about the displacements of neighboring blocks (global information) and uses it to predict and reduce search areas to small regions. In this way TDPE becomes much faster than TDE because it searches a very small region. It also becomes more accurate and without false peaks because, in the small region employed, the correlation coefficient has a single peak, which is easy to detect (Fig. 2.6).

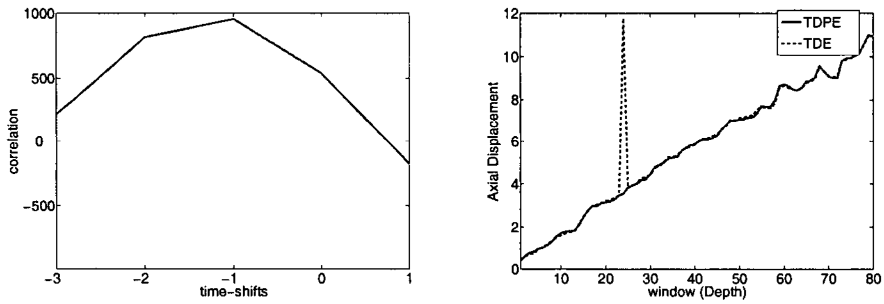


Figure 2.6: A sample of the correlation function used by TDPE is shown on the left. The correlation is not normalized and it has only been calculated for a small number of time-shifts. A sample of estimated displacement vectors generated by TDE and TDPE is shown on the right. Even though TDE uses normalized correlation, it may still find false peaks while TDPE can work even with non-normalized correlation and estimates the motion correctly.

2.2.1 The Main Structure of the Algorithm

Similarly to the TDE method, TDPE splits each RF line into a number of windows. Each window has four adjacent neighbors - a leading window to the left and a lagging window to the right on the same RF line, and windows at the same position on the upper and lower RF lines (Fig. 2.5). In elastography, windows have large overlap with each other (50% to 75%) and the spaces between the windows are very small. Because of this physical proximity, adjacent windows have similar displacements and strains.

We use the displacement estimates of the leading window (previous window) and windows at the same position on the upper RF line (upper window) which already have been calculated to bracket the displacement of the current window (internal conditions). The cross-correlation is then used to determine displacements within the bracket. Typically, only three lags inside the predicted brackets are checked to detect the displacement quickly and with high accuracy. Furthermore, since the region is small, normalization is not necessary.

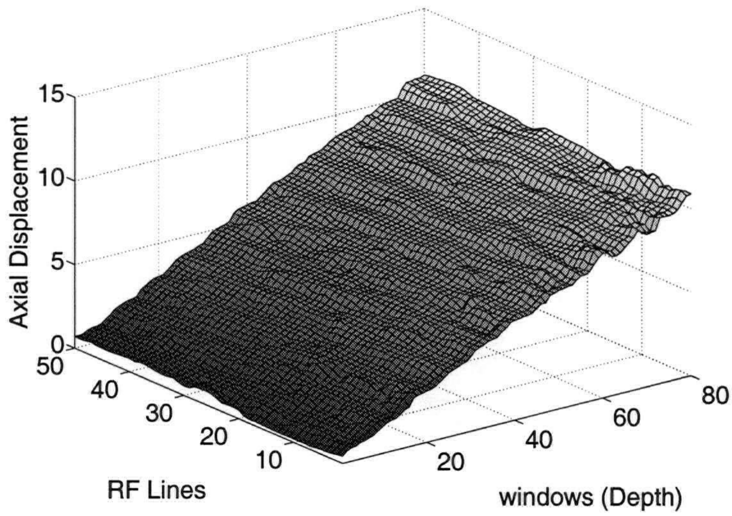


Figure 2.7: Estimated displacement image on a pair of simulated RF frames when TDPE with non-normalized cross correlation is used (median filter is not applied). The image shows that even though TDPE uses the non-normalized correlation it still finds the correct displacements.

2.2.2 Prediction Algorithm

The prediction algorithm is based on two types of information: (i) a boundary condition at the probe where the displacement is almost zero since the estimation is always done with respect to the probe itself and, (ii) internal conditions that state that the displacement of adjacent neighbors are always close to each other due to their physical proximity.

Finding the displacement is done from the first window (the closest window to the probe) to the last window at the end of the RF line. RF lines are processed starting from the top. Thus in Fig. 2.5 RF windows are processed from the top left to the bottom right. Considering the boundary conditions, at the first window of each RF line the correlations for the lag -1, lag 0 and lag +1 are calculated and saved. A three-point parabolic interpolation [16] is then applied to find the exact position of the correlation peak. This value is then recorded as the displacement of the current window. The closest integer value to this displacement is then used to bracket the search area for the next window (internal condition).

For the first RF line, since the information on the upper window does not exist, the search area will be:

$$[Round(previousWindowDis.) - 1, Round(previousWindowDis.) + 1], \tag{2.1}$$

which are just three lags and for the rest of RF lines the search area will be:

$$[min(Round(previousWindowDis.), Round(upperWindowDis.)) - 1, \tag{2.2}$$
$$max(Round(previousWindowDis.), Round(upperWindowDis.)) + 1], \tag{2.3}$$

where $previousWindowDis.$ is the displacement estimate from the previous window (lagging window on the same RF A-line) and $upperWindowDis.$ is the displacement estimate from the upper window (window at the same position on the preceding RF A-line)(Figure 2.5).

Because of physical proximity, the previous-window and upper-window typically point to the same position. This means that in most cases a bracket of 3 lags is enough for the cross-correlation search area. In the cases in which the previous-window and upper-window point to different locations, more than 3 lags are used to bracket the displacement and the one that points to the highest correlation coefficient will be selected. Again a three-point parabolic interpolation is then applied to find the exact position of the correlation peak.

By using the information of the upper window we make the prediction more precise but we also run the risk of carrying through an erroneous estimate from one RF line to the next. Therefore, as another implementation version, Equation (2.1) can be used for all RF lines. This means only the information form previous window will be considered in the prediction algorithm and the upper window will be ignored. This way we may lose track of the motion for a single RF line but it does not affect the rest of the RF lines. Furthermore a single corrupted line can be filtered later with the aid of median filter while a corrupted region can not be removed easily.

As another extension to the TDPE method the maximum correlation coefficient at each search can be used to validate the accuracy of the present motion estimation at that window. If this correlation coefficient is smaller than what we expect the search bracket can be enlarged. For example 5 lags can be used instead of 3 lags.

It should be noted that it is because of physical proximity fact that median filter is used in the standard TDE method to remove the remaining false peaks, since it replaces the value at each pixel by the median value of its neighborhood. But this fact is mostly used as a post processing key and it has never been used inside the process of motion estimation itself. TDPE uses this fact inside the search process to speed up the search and to make it more accurate.

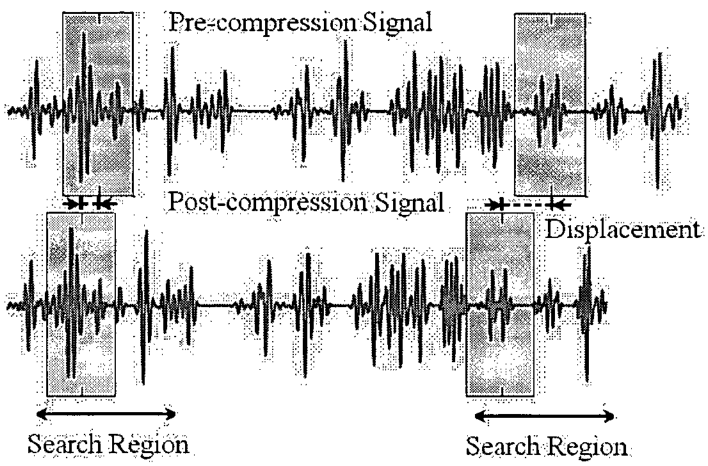
2.2.3 Pros and Cons of the Algorithm

TDPE is very fast since it only searches very small regions to find the displacements. TDPE calculates the correlation values at 3 lags and since all subpixelling methods need at least 3 correlation estimates to find the exact position of the correlation peak (see Appendix B.2), its computational cost is close to minimal among the correlation based algorithm. And because its main structure is the same as the standard TDE method, it keeps its sensitivity and high SNRe at low compression ratios and it is also easy to implement without additional hardware overhead. It is because of these properties that the algorithm offers an attractive option for real time applications since these applications need a fast and sensitive algorithm which is also easy to implement.

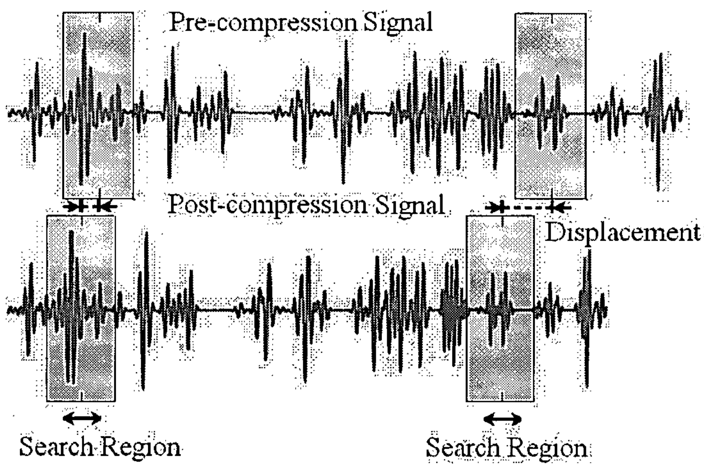
The current version of the TDPE algorithm predicts the motion at each window based on its neighbor's previously estimated motion. Therefore, if for any reason (like RF data corruption or high compression ratios) motion estimates from neighboring windows become imprecise the prediction algorithm will fail. In other words, the method relies on old information and if this information is incorrect the algorithm can not predict and estimate the motion correctly. Therefore, corrupted regions start to appear in displacement image.

2.3 Conclusion

In this chapter the standard Time Domain Cross Correlation (TDE) was overviewed and its shortcomings were pointed out. Following that, the TDPE, as a new motion estimation method



(a) The TDE algorithm.



(b) The TDPE algorithm.

Figure 2.8: The search region of the TDE algorithm needs to be greater than the largest possible motion (a), while the search region of the TDPE algorithm is fixed (b).

was introduced that works similarly to the TDE but performs much faster and is more accurate. These properties make TDPE an excellent choice for real time motion estimation with ultrasound images.

Chapter 3

Computational Cost

The chapter starts with the introduction of the computational cost as a new strain imaging factor. Following that it introduces a comparison factor and uses that to compare the current available algorithms that achieve real-time performance such as *Phase Root Seeking* (PRS) [90] and *Combined Auto Correlation Method* CAM [102] with the TDPE. The computational cost of standard TDE method is also computed as a reference and is compared with these methods.

3.1 Preliminary Considerations and Comparison Measures

PRS and CAM are the only motion tracking algorithms that are used in a real-time elastography system. PRS is used by the LP-IT company (Lorenz and Pesavento Ingenieurburo fur Informationstechnik, Bochum, Germany) company and was the first algorithm to calculate strain images of significant size in real time [89]. Currently they can generate up to 30 frames of strain per second. The CAM method is used by Hitachi (Hitachi Medical Corporation, Chiba, Japan) and currently they are able to generate up to 20 frames of strain images per second [104,105]. Therefore in order to validate the introduced TDPE algorithm as a real-time motion tracking algorithm, it is better to compare it with these algorithms.

For the algorithms considered here and for the standard TDE method, the computation time depends on digital signal processing parameters and is independent of the acoustic and mechanical properties of the tissue. It should be noted that this statement is not true for all motion tracking algorithms. As an example, in some iterative algorithms that try to maximize the correlation, the number of iterations depends on the amount of local compression which is high for softer regions and low for harder regions. Therefore the number of iterations at each region may depend on the mechanical properties of that region.

Computational performance is determined by the number of RF lines, number of windows in each RF line and the computation needed to find the displacement at a particular window. Let the window-length be W, the shift between windows be ΔW (Figure 2.1) and the number of windows nW. ΔW can be calculated directly from W, nW and the RF data length L ($\Delta W\% = [L/(nW \times W)] \times 100$). RF data length depends on the depth of the tissue D being examined, the sampling frequency f_s and the speed of sound c ($L = f_s \times 2 \times D/c$).

The TDE, TDPE, PRS and CAM algorithms are compared based on the *number of multiplications and summations* they require $[multiplies, adds]$. Since summation and multiplication are the main operations in hardware, by counting them a very accurate comparison can be made.

3.2 Computational Cost of the TDE algorithm

The number of iterations for the cross correlation calculation (nI) is a predefined variable in the TDE algorithm. It shows how many correlations are needed at each window to find the displacement of the window. The exact value of nI depends on the largest possible motion. Without stretching, the largest displacement usually happens at the end of the RF line since displacement accumulates and it is proportional to the amount of compression applied to the tissue. This fact can be seen in all displacement estimate images (Fig. 2.3(a) 2.3(b) 2.4(a) 2.4(b)).

Estimating the number of iterations for each algorithm makes the comparison easier. For a depth of 50 mm, $c = 1540$ m/s, $f_s = 40$ MHz, the length of the RF-data would be more than 2500 samples. For 1% strain, without temporal stretching, the worst case speckle motion is almost 25 samples (2500×0.01). Since in real time the external deformation could be both compression and relaxation the direction of the motion is also unknown therefore the search area should be at least

$2 \times 25 = 50$ samples around its center point for each window.

In off-line processing since the deformation is mostly in compression the search area can be reduced to 25. Therefore 25 is selected as a default size for search area for 1% compression. For higher compression ratios the search area is typically much larger.

It should be noted that for the same system, if the compression is increased to 10%, the search area for motion estimation should be at least 250 samples in each direction (2500×0.1) which is so large that it makes the algorithm very slow and error prone. In practice the size of the search region for TDE is a predefined value and set to be a constant which defines the maximum displacement that can be captured correctly by the algorithm and the algorithm simply fails for larger displacements.

Considering this the computational cost C_{TDE} of the TDE for calculating the displacement vector for a single RF line is given by Equation (3.1) below:

$$C_{TDE} = nW \times (nI \approx 25) \times NCC, \tag{3.1}$$

where nW is the number of windows, nI is the number of iterations and NCC is the normalized correlation cost. C_{TDE} is equal to the number of windows times the computational cost at each window, which itself is equal to the number of correlations that need to be estimated at each window.

3.3 Computational Cost of the TDPE algorithm

Similar to TDE, in the TDE algorithm A-lines of RF data are split into a number of overlapping windows with a specific length. The local axial displacements are then computed by cross correlating two corresponding windows but in contrast with TDE it typically checks just 3 lags to detect the displacement. Therefore its computational cost C_{TDPE} can be described similarly as:

$$C_{TPDE} = nW \times (nI \approx 3) \times CC, \tag{3.2}$$

where CC is the correlation cost (without normalizing). If TDPE with normalized correlation is needed its cost should be considered in the amount of computation at each window which means CC should be replaced with NCC which is the normalized correlation cost.

3.4 Computational Cost of the PRS algorithm

The PRS algorithm has additional overhead in order to convert the RF signals to base-band signals. It estimates the time shifts from phase shifts of the base-band signal by using a modified Newton iteration (please see Appendix (A-1)). The algorithm iterates at least twice [90]. PRS needs to consider the subsample time-shifts of base-band signals. Therefore, at each iteration step it needs to resample the complex data of the window in post-compression RF data and calculate its complex correlation with the window in pre-compression RF data in order to estimate the phase shift between them. As a result the computational cost C_{PRS} of the PRS algorithm can be written as:

$$C_{PRS} = BCC + nW \times (nI \approx 2) \times (RC_{complex} + CC_{complex}), \tag{3.3}$$

where BCC is the base-band conversion cost for a whole RF line, and $RC_{complex}$ and $CC_{complex}$ are resampling cost and cross correlation costs for a complex signal, respectively.

3.5 Computational Cost of the CAM algorithm

Similar to PRS, the CAM algorithm has an additional overhead in order to convert the RF signals to analytic signals. It estimates time shifts from the phase shifts of analytic signals that are estimated from the correlation of two complex signals. To unwrap the phase shift, the CAM algorithm estimates the phase shift at several locations ($M + N + 1$ locations according to Equation (A-8) in the Appendix) with the spacing of $\lambda/2$ where λ is the wave length of the ultrasound signal. In parallel with this process it also calculates the envelope correlation at the same locations.

The phase shift corresponding to the maximum envelope correlation is unwrapped. This phase shift is then used to estimate the displacement [102].

N and M are predefined variables in CAM and they show how many correlations need to be calculated with respect to the current position to the left (M) and to the right (N) to extend the aliasing limit from $\lambda/4$ in both directions. The exact value of N and M depends on the largest possible motion.

Similar to the TDE algorithm, it is preferred to estimate the number of iterations for the CAM algorithm to make the comparison easier. For c = 1540 m/s, f_o = 5 MHz and λ = 0.31 mm, if we want to capture the motion due to 1% compression (0.5 mm for the 50 mm phantom depth), the algorithm should be able to unwrap the aliasing up to $3\lambda/2$ (0.46 mm) in both directions (compression and relaxation). This means M and N in the CAM algorithm should be at least set to 3, since increasing M and N increases the unwrapped region by $\lambda/2$. This means that the autocorrelation function needs to be run $M + N + 1 = 3 + 3 + 1 = 7$ times. Considering that the deformation is just compression just one of M or N should be set to 3 and the other one can be set zero and the calculation of autocorrelation function reduces to 4 times ($M+N+1 = 3+0+1 = 4$). Thus $M + N + 1 = 4$ is an estimated value for the number of iterations for the CAM algorithm for 1% compression, while for higher compression ratios M and N should be considered much larger.

As a result for the CAM algorithm the computational cost C_{CAM} can be written as:

$$C_{CAM} = ACC + nW \times (nI \approx 4) \times (CC_{complex} + ENC)\,, \tag{3.4}$$

where ACC is the cost of converting to analytic signal for a complete RF line and ENC is the envelope normalized correlation cost.

3.6 Comparison of the Computational Costs

In order to compare the computational costs of these methods we need to go into further detail. Since windows have overlap, a more efficient conversion to base-band and analytic signals is done once per line before processing as shown in Equation (3.3) and Equation (3.4). In order to consider the cost of conversion for each window individually rather than the whole RF line we assume that windows have 50% overlap with each other. So the cost of conversion is divided between neighboring windows. Therefore Equation (3.3) and Equation (3.4) can be revised as follows:

$$C_{PRS} = nW \times [BCCW/2 + (nI \approx 2) \times (RC_{complex} + CC_{complex})] \,, \tag{3.5}$$

$$C_{CAM} = nW \times [ACCW/2 + (nI \approx 4) \times (CC_{complex} + ENC)] \,, \tag{3.6}$$

where $BCCW$ is the cost of base-band conversion of a single window, $ACCW$ is the cost of analytic conversion of a single window.

The cost of a correlation is that of inner product of vectors (Equation (B-2)). For the normalized correlation one needs to calculate the autocorrelation of the two windows as well, thus making it necessary to calculate 3 inner products of vectors (Equation (B-3)). Furthermore linear resampling calculations consist of 2 complex multiplications and one complex summation [89, 90] for each sample. Moreover complex multiplication needs 4 multiplications and 2 summations and complex summation needs 2 summations for each element:

$$\begin{aligned}(a, jb) \times (c, jd) &= (a \times c - b \times d, j\,(a \times b + c \times d)) \,, \qquad (3.7)\\ (a, jb) + (c, jd) &= (a + c, j\,(b + d)) \,,\\ |(a, jb)| &= \sqrt{(a \times a + b \times b)},\end{aligned}$$

Therefore the cost of each operation according to the $[multiplies, adds]$ on a vector with length N can be written as follow:

- Element-wise Multiplication of two real signals: $[N, 0]$
- Inner Product of two real signals: $[N, N-1]$
- Element-wise Multiplication of two complex signals: $[4N, 2N]$

- Inner Product of two complex signals: $[4N, 4N-2]$
- Linear Resampling of two real signals: $[2N, N]$
- Linear Resampling of two complex signals: $[8N, 6N]$
- Norm calculation of a complex signal (ignoring the cost of *sqrt*): $[2N, N]$

Furthermore, as shown in Appendix (Equation (A-3)), the conversion from analytic signals to base-band signals consists only of element-wise multiplication of two complex signals of the size of each window. But the conversion from real signals to analytic signals is calculated by adding an imaginary part to the signal, which is equal to its Hilbert transform. There are several ways of computing the Hilbert transform; in this paper we will use the most pertinent 14-point FIR filter approach presented in [89]. Calculating the 14 point FIR in time domain for each point requires 14 multiplications and 14 summations.

Since both PRS and CAM algorithms use the analytic signals this conversion overhead should be added to the overall cost of each algorithm. The current implementation of CAM uses a quadrature detector to capture real-time analytic signals [102] but since this feature is not available on all ultrasound machines the analytic conversion cost is considered here for the CAM algorithm. But if a quadrature detector is already available on the ultrasound machine the cost of converting the real signal to the analytic signal should be omitted from the overall cost for both CAM and PRS algorithms.

Considering these assumptions the computational requirement of each algorithm, based on $[multiplies, adds]$ reduces to:

$$\begin{aligned}
C_{TDE} &= nW \times \overbrace{(nI \approx 25)}^{SearchRegion} \times \overbrace{3 \times [Wlength, Wlength - 1]}^{NormalizedCorrelation}, \\
C_{TDPE} &= nW \times \overbrace{(nI \approx 3)}^{SearchRegion} \times \overbrace{[Wlength, Wlength - 1]}^{Correlation}, \\
C_{PRS} &= nW \times [\overbrace{[14+4, 14+2] \times Wlength/2}^{Analytic+Base-bandConversion} + (nI \approx 2) \times \\
&\quad (\overbrace{[8,6] \times Wlength}^{Resampling} + \overbrace{[4 \times Wlength, 4 \times Wlength - 2]}^{ComplexCorrelation})], \\
C_{CAM} &= nW \times [\overbrace{[14, 14] \times Wlength/2}^{AnalyticConversion} + (nI \approx 4) \times \\
&\quad (\overbrace{[4 \times Wlength, 4 \times Wlength - 2]}^{ComplexCorrelation} + \overbrace{3 \times [2Wlength, Wlength]}^{EnvelopeNormalizedCorrelation})],
\end{aligned} \tag{3.8}$$

Now it is easy to compare the computational efficiency of these algorithms. Since nW (number of windows in each line) and $Wlength$ (length of each window) are all variables common to all methods, their product can be replaced by a constant τ

$$nW \times Wlength = \tau, \tag{3.9}$$

Therefore the simplest form for the number of operations [multiplies, adds] for each algorithm reduces to:

$$\begin{aligned}
C_{TDE} &= [75 \times \tau, 75 \times \tau], \\
C_{TDPE} &= [3 \times \tau, 3 \times \tau], \\
C_{PRS} &= [33 \times \tau, 28 \times \tau], \\
C_{PRS}^{QD} &= [26 \times \tau, 21 \times \tau], \\
C_{CAM} &= [47 \times \tau, 35 \times \tau], \\
C_{CAM}^{QD} &= [40 \times \tau, 28 \times \tau]
\end{aligned} \tag{3.10}$$

where C^{QD}_{CAM} and C^{QD}_{PRS} are the computational costs of the CAM and the PRS algorithms when a quadrature detector is available on the ultrasound machine.

From the above, the computational cost of the TDPE algorithm is almost 1/25 of that of TDE and 1/10 of those of CAM and PRS algorithms for small compressions ($\leq 1\%$). For high compression ratios, as mentioned before, the search area for TDPE and PRS remains the same since the number of iterations for both of them is constant while the search area increases for TDE and CAM which requires them to run more calculations. The result also shows that number of operations for CAM is almost close to that of TDE while it was reported to be 6 times faster [102].

3.7 Conclusion

In this chapter the computational cost was introduced as a new strain imaging factor. The number of multiplications and summations $[multiplies, adds]$ was used as a comparison key. By using this factor, the TDPE algorithm (which was introduced in Section 2.2) was compared with current real-time algorithms and it was shown that it estimates the motion with 1/10 of the computational cost of both *Phase Root Seeking* [90] and *Combined Auto Correlation Method* [102] and with 1/25 of the computational cost of the standard TDE method.

Chapter 4

Strain Estimation

A well defined approach does not exist to validate the motion tracking algorithm independent of the strain estimation process in elastography. However, the comparison factors for strain estimation such as the *strain filter* [51, 106, 123, 125], *axial resolution* and *lateral resolution* [15, 96, 99] and *contrast-to-noise ratio* [11, 126] have been well specified in the literatures. Therefore, this chapter discusses the estimation of strain from the displacement estimates and the TDPE algorithm will be validated on the next chapter based on the strain images that it generates.

As explained in the Introduction, in gradient-based strain estimation methods, strain is calculated from the estimated displacement. The chapter starts with the overview of two standard strain estimation techniques namely, the *gradient operator* [81] and the *least squares strain estimator* [49]. Following those, new extensions for both techniques will be introduced, that can be used in the strain estimation process.

It should be noted that in this chapter *strain estimation* is the name given to the process of estimating the value of strain from the displacement estimates which have already been estimated, instead of the whole process of estimating strain from the sequences of raw RF data.

4.1 Gradient Operator

After estimating the displacement at each window the strain needs to be estimated at that

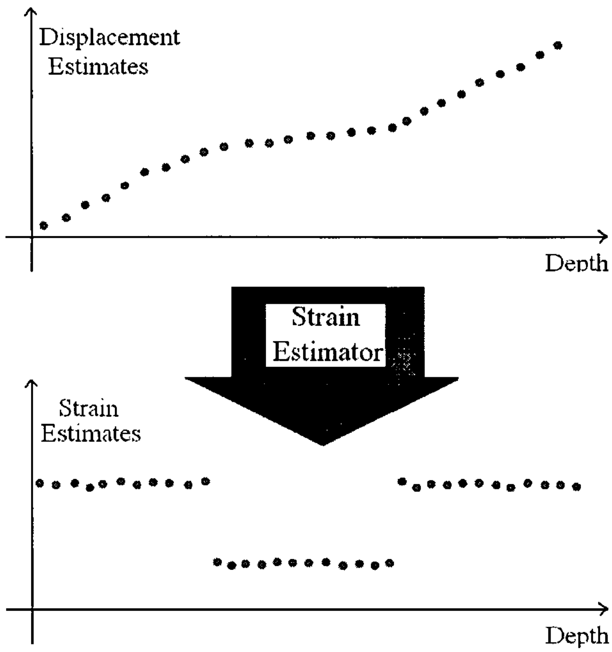


Figure 4.1: Sketch showing the process of estimating the strain from the displacement estimates.

region. According to Equation (1.2) local strain is equal to the difference between the estimated displacement at the current window and its previous window, normalized by the space between the windows. This is shown in Equation (4.1):

$$s(i) = \frac{d(i) - d(i-1)}{\Delta T}, \tag{4.1}$$

where $d(i)$ is the displacement estimate in the indexed segment and ΔT is the spacing between windows. To estimate the strain we can simply differentiate the displacement estimates. Therefore the strain estimation problem is similar to the slope estimation problem and all slope estimation techniques can be applied.

Typically in elastography, the strain is estimated by applying a gradient operation on the previously obtained time-shift estimates. Strain estimation by a simple gradient operator is very fast.

4.2 Least Squares Strain Estimation (LSE)

Least Squares Strain Estimator (LSE) was introduced in [49] to improve the process of strain estimation. The main goal of LSE is the reduction of noise amplification due to the gradient operation and this is achieved by a piecewise linear curve fit to the estimated displacement field. LSE finds the best-fitting line to a set of points inside a region around the point of interest (kernel), by minimizing the sum of the squares of the offsets of the points from the fitted line (Figure 4.2). The slope of the line is then reported as the estimate value of the gradient of the time-shifts at the center of the kernel. Similarly to the gradient approach, this value is then normalized by the space between the windows to estimate the real value of the strain at each window.

A displacement vector d with length M, around a given kernel of N points, may be modeled by:

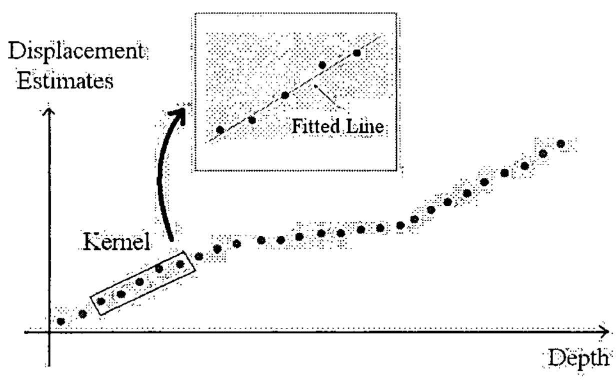


Figure 4.2: LSE with a kernel size of 5 is shown. A line is fitted to the data points with a least squares error and the value of the slope is then reported as the estimate of the gradient at the center of the kernel.

$$d(i) = a \times z(i) + b, \tag{4.2}$$

where z is the tissue depth and a and b are constants to be estimated. Equation (4.2) can be written in matrix format as:

$$d = A \begin{bmatrix} a \\ b \end{bmatrix}, \tag{4.3}$$

where A is an $N \times 2$ matrix for which the first column represents the depth position $d(i)$ and the second is a column of ones. The classical least squares solution of Equation (4.3) is given by [14]:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = [A^T A]^{-1} A^T \hat{d}, \tag{4.4}$$

where $\hat{d}$ contains the noisy displacement data, $\hat{a}$ and $\hat{b}$, are the LSE estimates of a and b and $\hat{a}$ is an estimate of the gradient at the middle of the kernel. In other words in LSE method the differentiation of the displacement estimates is calculated by fitting a straight line to a number of neighboring windows that has a least squares error. The slope of this line is then reported as local differentiation.

4.3 Higher Order Numerical Differentiators

The gradient operation amplifies small differences in displacements and has a linear frequency response ($j\omega$). This operation can be thought of as the crudest form of a high-pass filter. It has low gains at low frequencies and high gains at high frequencies and therefore amplifies high frequency noise [60, 121]. Numerical differentiation methods with smaller error can also be used in the strain estimation process. These techniques are widely used for slope estimation but they have not been used in elastography yet.

Numerical differentiation formulas can be derived by (i) constructing the Lagrange interpolating polynomial through a number of points, (ii) differentiating the Lagrange polynomial, and (iii) finally evaluating the differentiation at the desired point. A three points, five points and seven points rule for calculating the differentiation for strain estimation can be written as Equation (4.5), Equation (4.6) and Equation (4.7):

$$s\,(i) = \frac{d\,(i+1) - d\,(i-1)}{2\Delta T}, \tag{4.5}$$

$$s\,(i) = \frac{-d\,(i+2) + 8d\,(i+1) - 8d\,(i-1) + d\,(i-2)}{12\Delta T}, \tag{4.6}$$

$$s\,(i) = \frac{d\,(i+4) - 40d\,(i+2) + 256d\,(i+1) - 256d\,(i-1) + 40d\,(i-2) - d\,(i-2)}{360\Delta T}, \tag{4.7}$$

where again $d\,(i)$ is the displacement estimate in the indexed segment and ΔT is the spacing between windows.

By using more displacement estimates from neighboring windows (more points) numerical differentiators can estimate the gradient more accurately. But since displacement estimates are not accurate the interpolated polynomial will not be accurate either and therefore increasing the number of points may not necessarily improve the strain estimation process.

4.4 LSE with Higher Order Polynomials

The idea of LSE can be extended to higher order polynomial fitting. This means instead of best-fitting line (first order polynomial) we can fit a higher order polynomial to data points ($LSE_i(N)$ where i is the order of polynomial and N is the size of the kernel) to improve the bandwidth of the LSE. For example for a second order polynomial a displacement vector d, around a given kernel of N points, can be modeled by:

$$d(i) = a \times z(i)^2 + b \times z(i) + c, \tag{4.8}$$

where z is the tissue depth and a, b and c are constants to be estimated. Equation (4.8) can be written in matrix format as:

$$d = A \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \tag{4.9}$$

where A is an $N \times 3$ matrix for which the first column represents the squares of the depth $d(i)^2$, the second column represents the depth position $d(i)$ and the third is a column of ones. The classical least squares solution of Equation (4.9) is given by [14]:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix} = [A^T A]^{-1} A^T \hat{d}, \tag{4.10}$$

where $\hat{d}$ contains the noisy displacement data, $\hat{a}$, $\hat{b}$ and $\hat{c}$, are the LSE_2 estimates of a, b and c and $2 \times \hat{a} \times \lfloor \frac{i+N-1}{2} \rfloor + \hat{b}$ is an estimate of the gradient at the middle of the kernel where $\lfloor \frac{i+N}{2} \rfloor$ is the center of the kernel.

Although going to higher order polynomials can help increase the bandwidth of the strain estimator and increase its resolution by providing faster transitions from one region to another, but since the displacement estimates themselves are not very accurate, fitting a higher order polynomials to these inaccurate data is not the best solution. The trade-off between the improvement in signal-to-noise ratio and loss of resolution needs to be studied in order to choose an optimum kernel size and polynomial.

4.5 Conclusion

Standard strain estimation techniques have been overviewed in this chapter. Furthermore two slope estimation techniques have also been introduced, that can be used for strain estimation. Numerical differentiation methods have been introduced as an extension to the gradient operator and the least squares strain estimation method was extended to higher order polynomials.

It should be noted that estimating the stiffness in vibro elastography according to the Equation (1.4) is also a slope estimation process:

$$\frac{1}{k_j} \approx H_1^j - H_1^{j+1}, \tag{4.11}$$

Therefore, all strain estimation techniques can be used to estimate the stiffness from the magnitudes of the transfer function at low-frequencies.

Chapter 5

Simulation Results

This chapter presents a simulation study of the TDPE algorithm. The chapter starts by a quantitative evaluation of the TDPE method by means of signal-to-noise ratio ($SNRe$), dynamic range (DRe), sensitivity, contrast-to-noise ratio ($CNRe$) and axial resolution (R_a) using simulated RF data. Furthermore, the performance of the least squares strain estimation (LSE) is investigated. The effect of the LSE on SNRe have been studied in [49] but its effect on the axial resolution and CNRe with the available trade-offs have not been studied yet. In addition to that, for the first time, the effect of the median filter on the SNRe and axial resolution with the available trade-offs is investigated.

5.1 Validation

The performance of TDPE is validated using 1D simulation in Matlab (MathWorks, Inc., Natick, MA, USA) via three important metrics, namely (i) the strain filter (SF) [51, 123, 125] which represents the behavior of SNRe as a function of compression ratios and includes some information about dynamic range and sensitivity of the algorithm, (ii) the elastographic contrast-to-noise ratio (CNRe) which is proportional to the detectability of the lesion in the tissue [11, 126, 127] and (iii) elastographic axial resolution (R_a) that shows the smallest lesion that can be visualized in the elastogram [5,96].

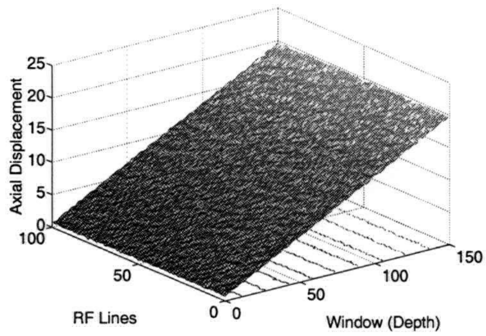
5.2 1D Simulation of a 2D Image

1D simulations were preferred over 2D simulations for two reasons: 1. lateral motion and beam effects have been accounted for as derating factors in the SF [51,125] and, hence, the results are expected to be similar to those detailed in those papers, and 2. significantly smaller simulation time is required when performing a statistical analysis over several parameters, like the window size, overlap, filtering etc.

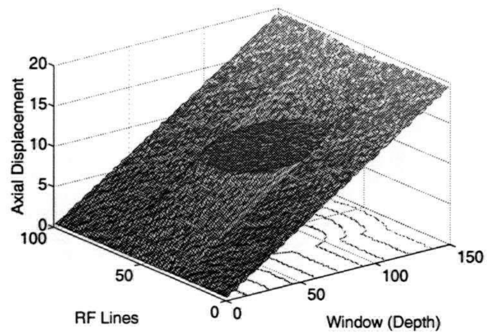
2D RF frames were generated with 1D simulations for each RF A-line. Similarly to previous work [117], pre- and post-compression RF signals corresponding to a 40-mm (D) uniformly elastic target segment were generated using a convolution between the impulse-response of the system (point spread function, PSF) and a normal distribution of scatterer amplitudes [130]. The point scatterers were uniformly distributed with density of at least 10 scatterers per pulse width. The speed of sound in tissue was assumed to be constant at 1540 m/s. The PSF was simulated using a Gaussian-modulated cosine pulse with a 5 MHz center frequency (f_o). The sampling frequency (f_s) was set to 40 MHz. Although the value of each of these parameters has a direct effect on the performance of the strain estimator [123], these parameters are chosen to simulate the RP 500 ultrasound system (Ultrasonix, Burnaby, Canada) as closely as possible.

To compute the post compression signal the phantom was modeled as a serial connection of springs [81]. A stiffness map was assigned to each line and the displacements of each point scatterers were calculated by compressing the model and calculating the displacement at each region.

For simulating a homogeneous phantom the value of the stiffness is set to be the same throughout the image which leads to uniform compression of the point scatterers [15]. For simulating a phantom with a circle inclusion, the stiffness of a circle inside a phantom was set to be 4 times stiffer than its background and for simulating a three layered phantom, the stiffness of the middle layer was set to be 4 times stiffer than other two layers. The post-compression signal is then calculated by convolving the compressed point scatterers with the original PSF.



(a) Homogeneous Phantom.



(b) Phantom with Inclusion.

Figure 5.1: Estimated displacement at 1% strain on both simulated homogeneous phantom and phantom with inclusion when TDPE is used as a motion tracking algorithm (D= 40mm, W=1mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

A sample of the displacement estimate from both simulated homogeneous phantom and simulated phantom with inclusion is shown in Fig. 5.1(a) and 5.1(b). TDPE was used to estimate the motion from simulated RF images. The inclusion is almost detectable in the displacement estimates itself before strain estimation process.

5.2.1 Evaluation of Strain Estimators

The choice of strain estimator has a direct effect on all of the metrics. Therefore, before a quantitative evaluation of the TDPE method by means of $SNRe$, DRe, sensitivity, $CNRe$ and R_a, it is important to evaluate the strain estimator that we are going to use.

Standard strain estimation techniques have been overviewed and new strain estimation techniques have also been introduced in the previous chapter. This section evaluates these methods based on their strain images' signal-to-noise ratio.

The elastographic signal-to-noise ratio characterizes the noise at which a value of strain is estimated and it is defined by [123]:

$$SNRe = \frac{m_s}{\sigma_s}, \tag{5.1}$$

where m_s denote the statistical mean strain estimate and σ_s denotes the standard deviation of the strain noise estimated from the elastogram.

In order to compare the strain estimation algorithms, a three layered phantom ($4 \times 4cm^2$) with a four times stiffer layer inside was simulated and two RF images were generated with 100 RF A-lines each (1% compression). Displacements were estimated at each of the lines by using the TDPE algorithm (W=1mm, overlap=75%). The displacement estimates of the first soft layer were then used to compare the strain estimation methods (50 windows).

The performance of the numerical differentiators is shown in Table 5.1. As expected, the results from the numerical differentiators show a little improvement with respect to the gradient method (smaller deviations and larger SNRe).

Strain Estimator	**Standard Deviation**	**Mean**	$SNR = \frac{STD}{Mean}$
Ideal	0	1.4134	∞
Gradient	0.9283	1.4175	1.4959
3 Points Differentiator	0.7055	1.4239	1.9735
5 Points Differentiator	0.8455	1.4256	1.6504
7 Points Differentiator	0.8755	1.4271	1.5963

Table 5.1: Comparison of gradient operator with numerical differentiators. All values are averaged over 100 realizations. To compare the method the result from ideal strain estimator is also included. The table compares the algorithms' standard deviation and their resulting signal-to-noise ratios (SNR).

The performance of the LSE_1 is shown in Table 5.2. As the size of the kernel increases the algorithm's performance improves (smaller deviations and larger $SNRe$):

Finally the performance of the LSE_2 is shown in Table 5.3. Similarly to the LSE_1, as the size of the kernel increases the algorithm's performance improves (smaller deviations and larger $SNRe$). However, the improvement is less marked as in the case of the LSE_1 method. This could be explained by the fact that in LSE_2 we are fitting a second order polynomial (to increase the bandwidth of the estimator) which is not a line anymore and have its own ripple.

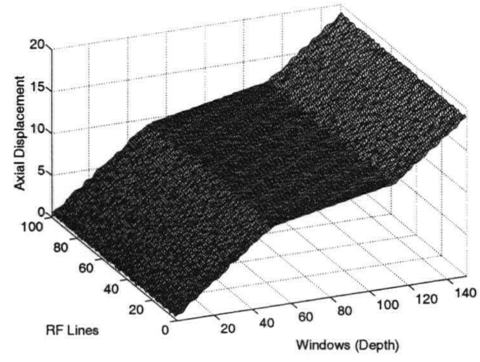
Strain Estimator	**Standard Deviation**	**Mean**	$SNR = \frac{STD}{Mean}$
Gradient	0.9283	1.4175	1.4959
$LSE_1(4)$	0.5489	1.4216	2.5293
$LSE_1(6)$	0.3204	1.4224	4.2678
$LSE_1(8)$	0.1950	1.4232	6.9533
$LSE_1(10)$	0.1298	1.4223	10.4761
$LSE_1(12)$	0.0917	1.4215	14.7107
$LSE_1(14)$	0.0673	1.4200	19.9715
$LSE_1(16)$	0.0520	1.4188	26.0171
$LSE_1(18)$	0.0436	1.4164	31.3489

Table 5.2: Comparison of $LSE_1(N)$. All values are averaged over 100 realizations.

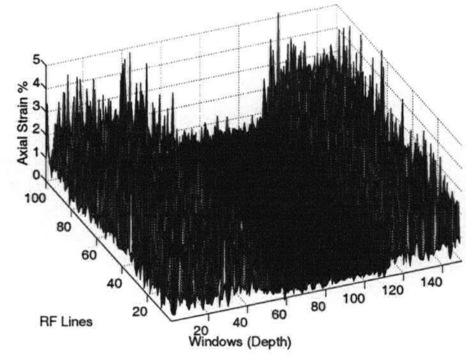
Strain Estimator	**Standard Deviation**	**Mean**	$SNR = \frac{STD}{Mean}$
Gradient	0.9283	1.4175	1.4959
$LSE_2(4)$	1.2878	1.6617	1.2596
$LSE_2(6)$	0.9718	1.5700	1.5828
$LSE_2(8)$	0.7302	1.4658	1.9201
$LSE_2(10)$	0.5465	1.4287	2.4496
$LSE_2(12)$	0.4012	1.4233	3.2980
$LSE_2(14)$	0.3073	1.4225	4.3012
$LSE_2(16)$	0.2360	1.4198	5.4440
$LSE_2(18)$	0.1860	1.4175	6.8615

Table 5.3: Comparison of $LSE_2(N)$. All values are averaged over 100 realizations.

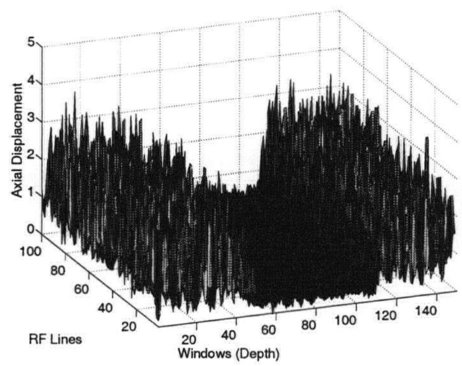
For the qualitative comparison of the strain estimation method, a result of motion estimation of a simulated three layered phantom was used as initial data in the strain estimation process. The result of strain estimation with the gradient operator is shown in Figure 5.2(b). It has been shown that without any filtering this operator results in a noisy elastogram with high resolution (sharp edges). The result of strain estimation with a numerical differentiator is also shown in Figure 5.2(c) and 5.2(d). Both methods show and improvement with respect to the simple gradient method (smaller upper bound) but the difference between three points differentiator and seven points differentiator is not quite clear. Finally the performance of LSE_1 method is shown in for two kernel sizes of 6 (Fig. 5.2(e)) and 12 (Fig. 5.2(f)). The improvements in the elastograms are evident. As the kernel size increases the ripple in the estimated strain decreases which shows the improvement in SNRe.



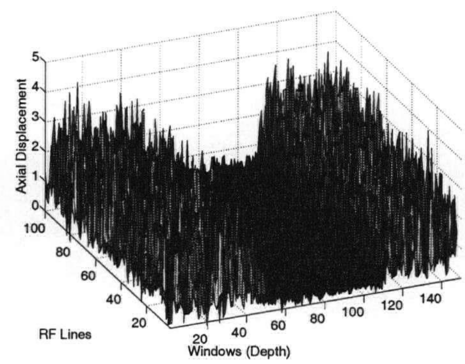
(a) Displacement estimates.



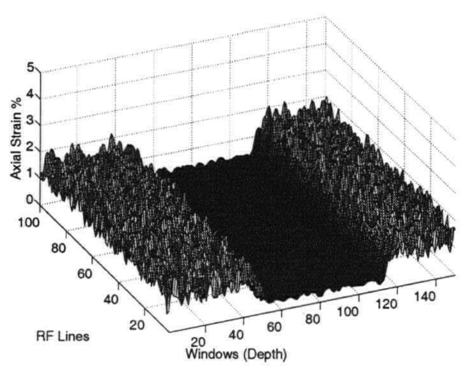
(b) Elastogram with Gradient operator.



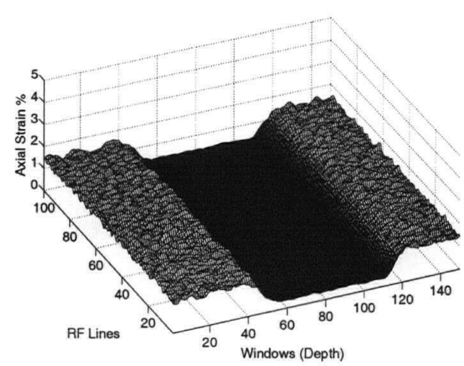
(c) Elastogram with three points differentiator.



(d) Elastogram with seven points differentiator.



(e) Elastogram with LSE_1 with kernel size of 6.



(f) Elastogram with LSE_1 with kernel size of 12.

Figure 5.2: A sample of displacement estimates of a three layered phantom at 1% compression with its corresponding elastogram when different strain estimators are used.

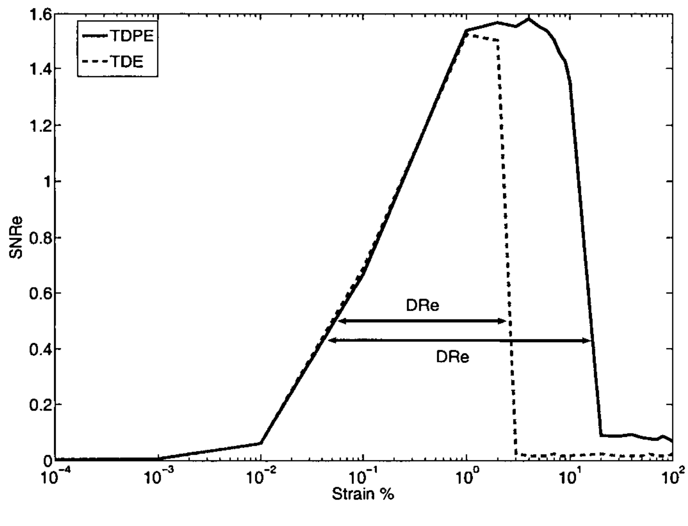
To be consistent with other researchers, in this thesis a standard LSE method (LSE_1) and the gradient operator will be used as strain estimators. The trade-offs involved in their use will be presented in the following sections. Studying the trade-offs between the SNRe and resolution for higher degree polynomial fittings and LSE_2 will be the topic of future work.

5.2.2 Strain Filter Study (SF)

The SF [123] describes the important relationship among the resolution, dynamic range (DRe), sensitivity and elastographic SNRe (defined as the ratio of the mean strain to the standard deviation of the strain in the elastogram), and may be plotted as a graph of the upper bound of the SNRe vs. the strain experienced by the tissue, for a given elastographic axial resolution (as defined by the data window length [96]). The SF is a statistical upper bound of the transfer characteristic that describes the relationship between actual tissue strains and the corresponding strain estimates depicted on the elastogram. It describes the filtering process in the strain domain, which allows quality elastographic depiction of only a limited range of strains from tissue. This limited range of strains is due to the limitations of the ultrasound system and of the signal processing parameters and algorithms. The SF is obtainable as the ratio between the mean strain estimate and the appropriate lower bound on its standard deviation.

The low-strain behavior of the SF is determined by the acoustic properties of the ultrasound machine and the high-strain behavior of the SF is determined by the rate of decorrelation of a pair of corresponding signals due to tissue distortion that arises due to several reasons. These include nonuniform and/or nonslip boundary conditions, lateral and elevational motion, differing angular orientation of the transducers relative to the direction of compression (resulting in depth-dependent decorrelation), and stress-strain nonlinearities in tissues and tissue-mimicking phantoms [111].

The general appearance of the SF in three dimensions, demonstrating the trade offs between SNRe and strain at all resolutions is available in [79] which investigates the behavior of the SF for different signal processing parameters such as window size (W) and window overlap. In addition to the selection of the signal processing parameters, the SF can be derated by other corrupting processes such as frequency dependent attenuation [125] and lateral and elevational signal decor-



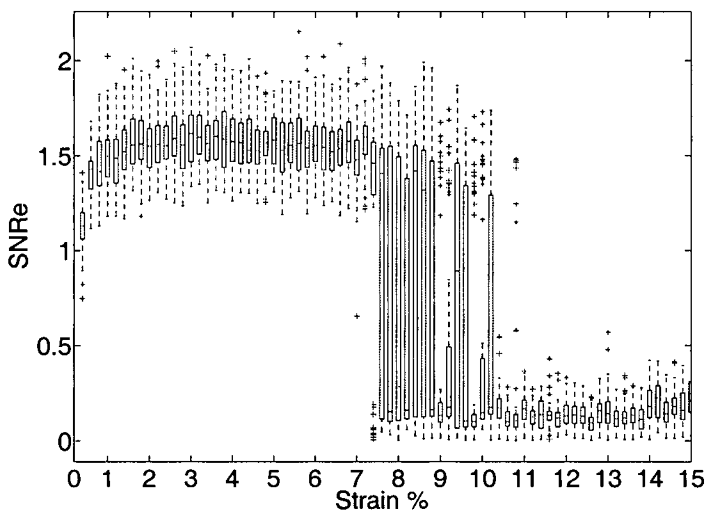


Figure 5.3: The strain filter of the TDPE algorithm. Compression is done once from 0.001% to 100% in logarithmic steps (Top) and once from 0% to 15% in 0.2% steps to increase the accuracy (Bottom). 100 realizations are used to show the mean values (Top, Bottom) and the deviation from the mean values (Bottom). The gradient operator is used as a strain estimator without any filtering (D=40mm, W=1mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

relation [51].

To generate the SF of the TDPE method and compare it with the SF of the standard TDE algorithm, a homogeneous 40 mm phantom was simulated using a 1D simulation. The phantom was compressed from 0.001% to 100% in logarithmic scale. Both TDE and TDPE with $W = 1$ mm and 75% window overlap (160 window in 40 mm) were used to estimate the motion at each step and the gradient operator was used to estimate the strain. The SNRe was estimated at each compression ratio according to Equation (5.1) and the SFs are shown in (Fig. 5.3(a)). The mean values of the SNRe for 100 simulations are used as the SNRe at each compression.

It can be seen from Figure 5.3(a) that for small compression ratios the TDPE and the TDE performs similarly (sensitive to 0.01% compression). For higher compression ratios, the TDPE shows a larger dynamic range (DRe) compared to the TDE. The SNRe of the TDE algortihm drops for compressions higher than 3% while the TDPE algorithm starts to saturate at compressions higher than 10%.

To investigate the behavior of the TDPE in the high SNRe region (0.2% to 15%) another simulation was run. A homogenous 40 mm phantom was simulated using a 1D simulation. The phantom was compressed from 0% to 15% in steps of 0.2% strain (76 frames). TDPE with $W = 1$ mm and 75% window overlap (160 window in 40 mm) was used to estimate the motion at each step and the gradient operator was used to estimate the strain. The SNRe was estimated at each compression ratio (Fig. 5.3(b)). The Matlab boxplot function was used to show the mean value and the deviation from the mean value of the SNRe for 100 simulations. It can be seen from Figure 5.3(b) that TDPE is capable of estimating strains of up to 8% compression.

5.2.2.1 Effect of LSE and Median Filter on the Strain Filter

As discussed in the previous chapter, the LSE can be applied instead of a simple gradient operator to improve the SNRe which leads to elastograms with higher quality. The effect of the LSE on the SF of the standard TDE algorithm has been presented in [49]. To study the effect of the LSE on the proposed TDPE algorithm, the improvement in SF is shown in Fig 5.4(a) and an improvement in SNRe at a single compression ratio is shown in Fig 5.4(b) when the LSE method with different kernel sizes (2-12) are used instead of the gradient operator on the same data that was used in

Figure 5.3(b).

The result shows more than linear improvement in SNRe when LSE is used which confirms the results from [49] that state that a kernel size of N improves the SNRe by $N^{3/2}$. To show this, the improvement in SNRe at a single compression ratio (1%) is shown in Fig 5.4(b). LSE with a kernel size of 2 is the same as the gradient operator and as the kernel size increases, the SNRe improves.

Median filtering is another popular filtering type in elastography that helps to remove the noisy regions and improves the quality of the elastograms. Median filtering can be applied either after motion estimation to remove false peaks [130] or after strain estimation to remove noisy regions [109].

Although median filtering has been widely used in elastography its effect on the SF and resolution has not been investigated in the literature yet [111].

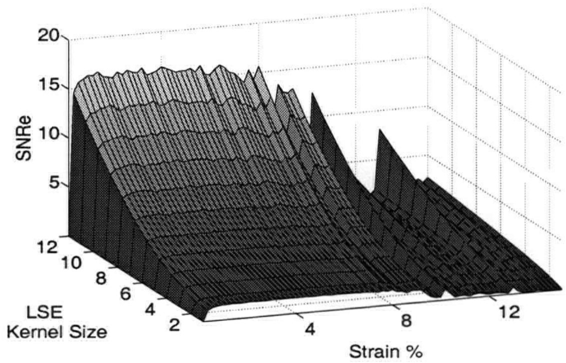
To demonstrate the performance of median filtering, the simulated RF data were used by TDPE to estimate the motions. 2D median filters with variable kernel sizes were used once to filter the displacement estimates before applying the strain estimator (Fig. 5.4(c), 5.4(d)) and once after the strain estimation process (Fig. 5.4(e), 5.4(f)). The gradient operator was used in both cases as a strain estimator.

A 2D median filter was preferred over 1D because it has a larger effect and it makes it possible to remove corrupted lines while a 1D filter selects the median value from the same corrupted line.

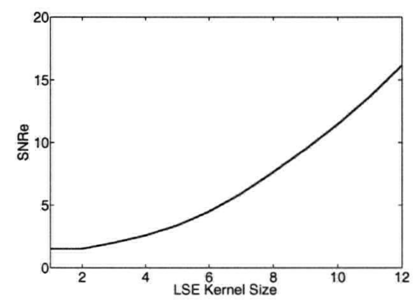
As we expected, an improvement in both cases is not linear since the median is a nonlinear filter. The results also shows that the median filter performs better if it is applied on the strain estimates instead of displacement estimates which could be explained by the fact that in a single neighborhood strain estimates are much closer than displacement estimates since the displacement accumulates while the strains stay the same. Therefore, we will apply it after the strain estimation process for the rest of simulations.

5.2.3 Contrast-to-Noise Ratio Study

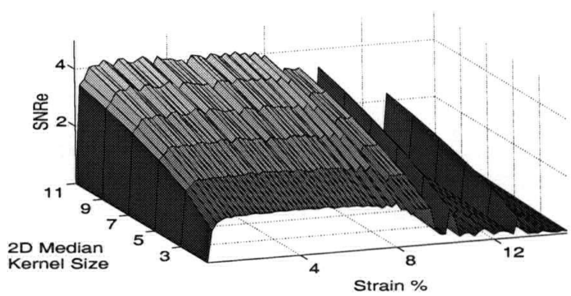
The contrast-to-noise ratio (CNRe) in elastography is an important quantity that is related



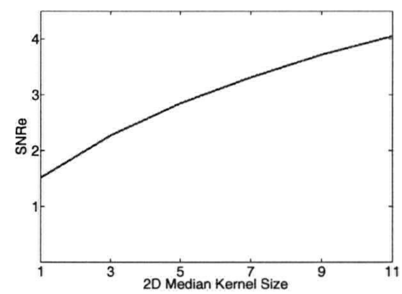
(a) Improvement in SF with LSE.



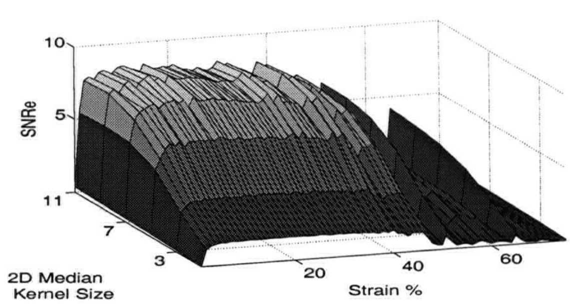
(b) Improvement in SNRe at 1%.



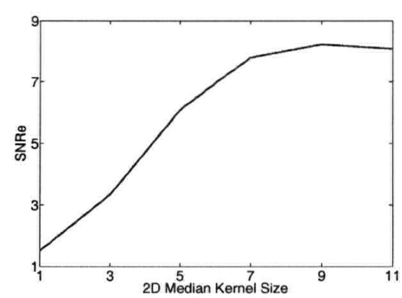
(c) Median filter is applied to displacement estimates.



(d) Improvement in SNRe at 1%.



(e) Median filter is applied to strain estimates.



(f) Improvement in SNRe at 1%.

Figure 5.4: Improvement in strain filters for TDPE when (a,b) LSE, (c,d) gradient operator following a 2D median filter and (e,f) 2D median filter following a gradient operator is used as a strain estimator. Compression is done from 0% to 15% in 0.2% steps. 100 realizations are used to show the mean values. The corresponding improvements in SNRe at 1% compression are also shown in (b, d and f) (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

to the detectability of a lesion [79]. The properties of the ultrasound imaging system and signal processing algorithms described by the SF can be combined with the elastic contrast properties of tissues with simple geometry [48], enabling prediction of the elastographic contrast-to-noise ratio (CNRe) parameter. This combined theoretical model enables prediction of the elastographic CNRe for simple geometry such as layered (1D models) or circular lesions (2D models) embedded in a uniformly elastic background. This quantity includes the contrast characteristics of the output elastogram and is given by [11,126]:

$$CNRe = \frac{2 \times (s_t - s_b)^2}{\sigma_t^2 - \sigma_b^2}, \tag{5.2}$$

where s_t and s_b represent the mean values of the estimated strain in the target and the background, respectively and σ_t and σ_b are the standard deviations in the estimated strains.

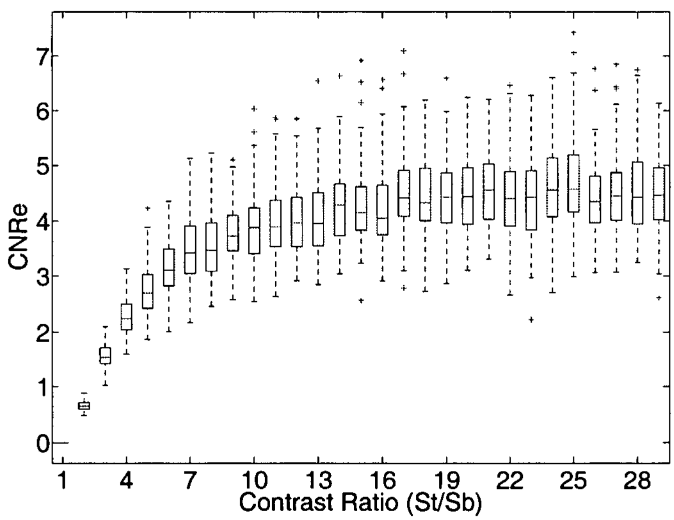


Figure 5.5: CNRe of TDPE plotted as a function of contrast ratio. CNRe is calculated from the SF and the strain and deviation values up to 6% strain are used. Using 0.2% strain in the inclusion and 0.2% to 6% strain in the background (D=40mm, W=1mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

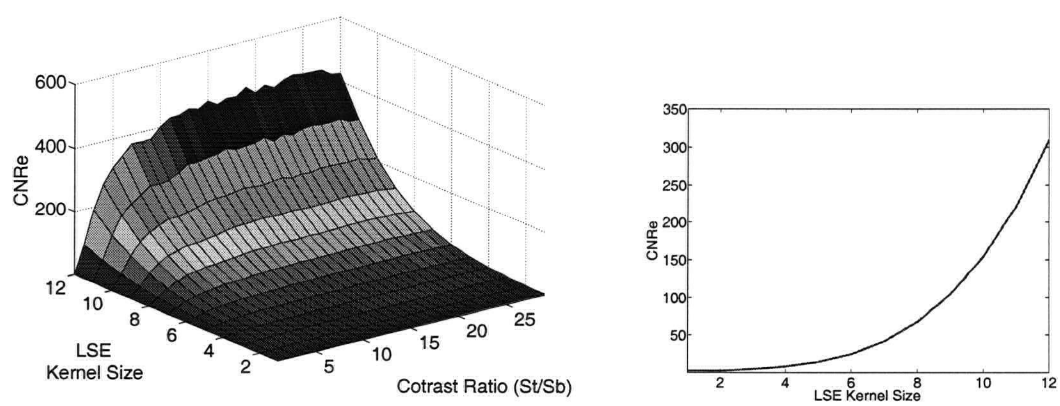
The same data from the homogeneous phantom at different compression ratios were used to calculate the CNRe of TDPE at different contrast ratios between the target and background (Fig.

5.5). A strain of 0.2% was used as the target strain (inclusion) and 0.2% – 6% strain was selected as the background strain. Similarly to SNRe, the Matlab boxplot function was used to show the mean value and the deviation from the mean value of CNRe for 100 simulations.

5.2.3.1 Effect of LSE on Contrast-to-Noise Ratio

The effect of the LSE on CNRe has not been presented in the literature yet. The improvement in CNRe is shown in Figure 5.6(a) when LSE with different kernel sizes (2-12) is used instead of the gradient operator on the same data that was used in Figure 5.5. As we expected it shows more than quadratic improvement in CNRe which confirms the results from [111], that estimates that $CNRe \approx SNRe^2$, and [49] concludes that for a kernel size of N, the SNRe improves with $N^{3/2}$. Therefore the CNRe should improve with N^3.

To investigate the improvement of CNRe more precisely when LSE is used, CNRe at a single compression ratio (1%) is shown in Figure 5.6(b). It shows a large improvement in CNRe as the kernel size of LSE goes up which means the detectability of the lesion increases rapidly as the kernel



(a) Improvement in CNRe with LSE.

(b) Improvement in CNRe at contrast ratio of 5.

Figure 5.6: CNRe of TDPE using 1D simulated data and LSE(2-12) as an strain estimator. CNRe is calculated from SF and the strain and deviation values up to 6% strain are used, by using 0.2% strain in the inclusion and 0.2% to 6% strain in the background (a). 100 realizations are used to show the mean values. The improvement in CNRe at contrast ratio of 5 is also shown in (b) (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

size of the LSE increases.

As a result it seems that it is better to use LSE with large kernel size during the detection process. But as we will discuss later, increasing the LSE kernel size degrades the resolution. Therefore the operator needs to change the kernel size in real time to optimize the elastogram from both resolution and CNRe points of view.

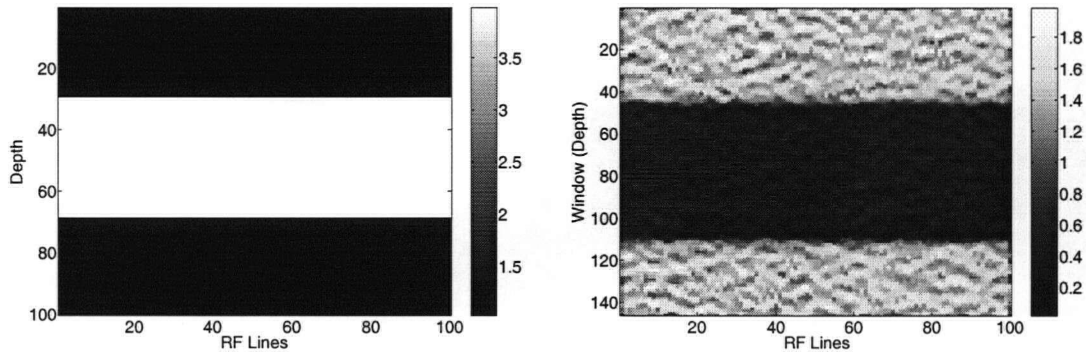
5.2.4 Axial Resolution Study (R_a)

A number of definitions for the estimation of the axial resolution exist in the literature [5,15,96]. It has recently been demonstrated in [96] that the upper bound on the elastographic axial resolution is proportional to the length of the point spread function (PSF) of the transducer. The constant of proportionality depends on the exact definition that is used for the axial resolution, but is generally a number between 1 and 2 [79]. This limit is not always achieved due to noise considerations [5]. The practically attainable resolution for a given system, however, is generally limited by the choice of signal processing parameters such as window size and overlap, as long as the window size is larger than the PSF of the system and other signal processing parameters such as strain estimation algorithm.

Similar to [15], in this thesis axial resolution is defined as the length of the time segment over which the strain changes from the value in the background to the value in the target (90%-10%). This definition is close to the definition of the filter accuracy in signal processing. Similarly to signal processing where filters with sharp transition from pass-band to stop-band are preferred, elastograms with high resolutions are also desired.

To derive the axial resolution of our strain estimator a 3 layered phantom was simulated with a four times stiffer homogeneous layer inside (Fig. 5.7). Similarly to previous sections, the RF data were simulated at different compression ratios and time-delays were estimated by using the TDPE algorithm. Strain was then estimated as the gradient of the time-delay estimates.

The resolution was estimated at each line and the results were averaged over 100 independent lines. For $W = 1$ mm, overlap$= 75\%$ ($\Delta W = 25\%$) the axial resolution estimated to be $R_a = 0.3$ mm. This result is consistent with results in the literature, since it has been predicted that for a gradient



(a) Stiffness map.

(b) Estimated elastogram.

Figure 5.7: To simulate a 3 layered phantom, the stiffness of the inside layer was set to be 4 times stiffer than its background. The stiffness map is shown in (a) and its corresponding estimated elastogram image at 1% compression is shown in (b). LSE with a kernel size of 6 is used as an strain estimator and a 3×3 median filtering was applied after the strain estimation

strain estimator the axial resolution is almost equal to the window shift ($R_a \approx \Delta W$) [5, 111, 127].

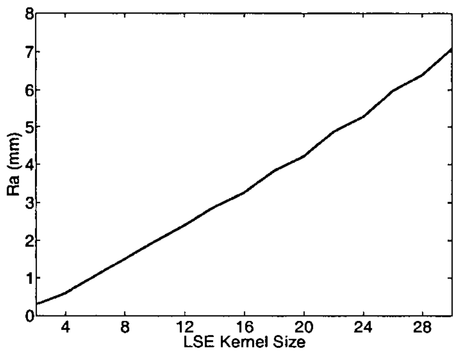
5.2.4.1 Effect of LSE and Median Filtering on Axial Resolution

The trade-off between achievable SNRe (CNRe) and axial resolution in elastography has been well studied in [111] with respect to the window size and window overlap. The trade-off was demonstrated for two important signal processing parameters (W, ΔW) but other signal processing parameters have not been studied yet.

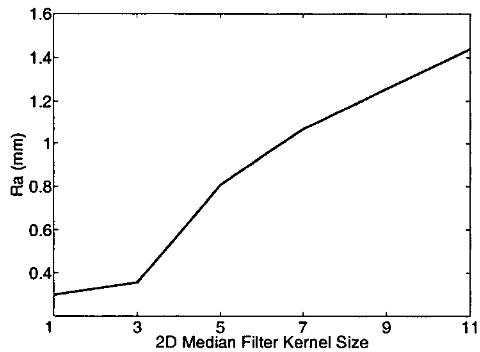
In this section, for the first time the trade-off with two other parameters, namely kernel size of LSE and median filter is presented.

The use of filtering typically improves the SNRe at the expense of spatial resolution. Therefore, we expect the commonly used least-squares strain estimator to have a poorer spatial resolution than those estimated using the gradient operation. However the trade-off between SNRe and R_a for LSE has not been studied in literature [111].

In order to measure the effect of LSE on the axial resolution of the elastogram, the same simulated data from the three layered phantom used, but instead of the gradient estimator, LSE with different kernel sizes varying form 2 to 30 were used to estimate the strain. The resolution



(a) Resolution of LSE at different kernel sizes.



(b) Axial Resolution when elastogram is median filtered.

Figure 5.8: Effect of filtering on the axial resolution. For LSE the kernel size was varied from 2 to 30 (a) and for 2D Median filtering the kernel size was changed from 3×3 to 11×11 (b). 100 realizations are used to show the mean values. (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

was then estimated as the length of the time segment over which the strain changes from the value in the background to the value in the target (90%-10%) (Fig. 5.8(a)). As predicted by [111] the result shows a linear degradation in resolution. Therefore, for a known window-shift (ΔW), and LSE kernel size (N), the axial resolution can be estimated according to Equation (5.3):

$$R_a \approx \Delta W \times N, \tag{5.3}$$

The gradient strain estimation technique results in a linear transition of strain from the value in the background to the value in the stiff layer inside, however, it amplifies the noise. On the other hand LSE results in a nonlinear transition from soft region to hard inclusion but it reduces the noise significantly [111].

The performance of the LSE can be concluded according to the results in Sections 5.2.3 and 5.2.2. For a kernel size of N, on can state that:

- SNRe improves with $N^{3/2}$
- CNRe improves with N^3

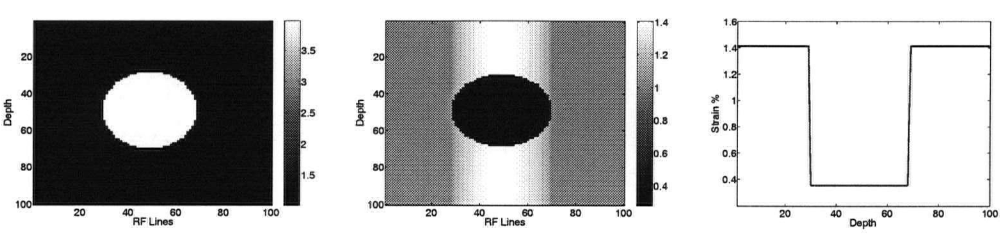
- R_a degrades linearly with N

To study the effect of median filtering, the 3 layered elastogram generated by the gradient operator was median filtered with a number of different kernel sizes varying from 3×3 to 11×11. The median filter was applied after the strain estimation process. Again the resolution was estimated as the length of the time segment over which the strain changes from the value in the background to the value in the target (90%-10%). The result is shown in Figure 5.8(b). As we expected due to the nonlinearity of the median filter the resolution degrades nonlinearly. It is also clear from Figure 5.8(b) that while the median filter improves the elastogram (SNRe) it doesn't degrade the resolution very much. Therefore it is because of these property that median filter is being widely used in elastography.

5.2.5 Qualitative Evaluation

For a qualitative evaluation of the TDPE algorithm a 2D phantom (40 mm × 40 mm) with a 4 times stiffer circular inclusion was simulated using a 1D simulation to generate the RF data (Fig. 5.9). These types of phantoms are widely used since they simulate the tumors inside the body.

In order to see the dynamic range of the algorithm, the phantom was compressed from 0.5% to 10% with a 0.5% strain step (20 frames). TDPE with W = 1 mm and 75% overlap was used to estimate the motion and the standard gradient operator was used to estimate the strain at different



(a) Stiffness map. (b) Ideal elastogram. (c) Ideal strain at center.

Figure 5.9: To simulate a phantom with a circle inclusion the stiffness of a circle inside a phantom was set to be 4 times stiffer than its background. The stiffness map is shown in (a) and its corresponding ideal strain image at 1% compression is shown in (b,c).

compression ratios. As shown in Figure 5.10, the algorithm is capable of estimating strains at high compression ratios. The colorbar besides each elastogram shows the maximum available strain which is proportional to the compression ratio. As the compression goes up the elastogram starts to degrade due to the decorrelation but the inclusion is still detectable even at 10% compression ratio (Fig. 5.10(f)).

TDPE fails at very high compression ratios because the displacements of the neighbors are not accurate anymore. Since the TDPE method depends on this information it fails to predict the search area correctly and cannot estimate the motion. As shown in Figure 5.10 corrupted regions start to appear in the elastograms at very high strains.

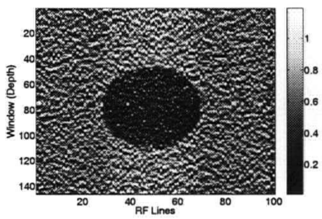
To show the effect of the Least Squares Strain Estimator (LSE), the time-shift estimates at 1% compression were selected and elastograms were obtained with LSE of different kernel sizes (Fig. 5.11). It can be seen that as the kernel size (N) of the LSE increases, the elastogram improves rapidly and looks more like the ideal elastogram (Fig. 5.9) which is due to the improvement in SNRe ($N^{3/2}$).

Fig. 5.11 also shows the trade-off between the SNRe and axial resolution (R_a). As the kernels size of LSE increase the boundary of the inclusion also fades. To clarify this, the strain at the center line corresponding to each elastogram is also shown below the elastograms. Comparing it with an ideal strain (Fig. 5.9(c)), the ripples in the high strain and low strain regions are inversely proportional to the SNRe and the slope of the transitions between high strain in the background and low strain in the inclusion are also inversely proportional to the resolution.

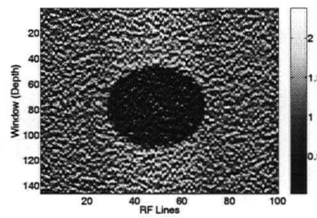
Therefore, as the kernel size of the LSE increases, the ripple in both regions decreases which results in an improvement in SNRe. However, as the kernel size increases, the slope of the transition region between high strain and low strain region decreases which means the resolution degrades.

Similar to LSE, to show the effect of median filtering on the estimated elastograms, the time-shift estimates at 1% compression were selected and the gradient operator was used to create the strain image. To generate the final elastogram, the strain image was median filtered with different 2D kernel sizes (Fig. 5.12). It can be seen that as the kernel size ($N \times N$) of the 2D median filter increases, the elastogram improves rapidly, which is due to the improvement in SNRe.

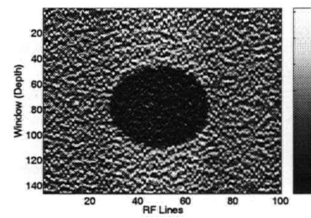
Fig. 5.12 also shows the trade-off between the SNRe and R_a. Similar to LSE, as the kernel



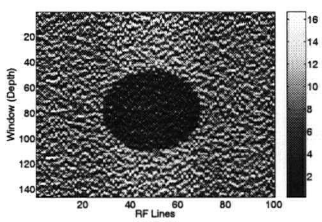
(a) 0.5% compression.



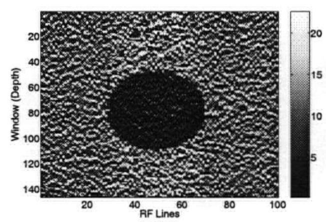
(b) 1% compression.



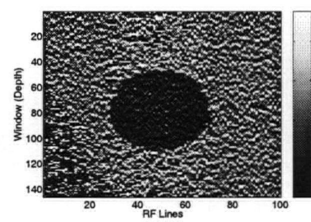
(c) 3% compression.



(d) 7% compression.



(e) 9.5% compression.

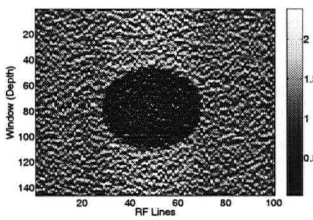


(f) 10% compression.

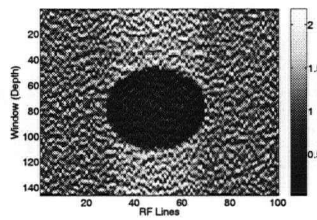
Figure 5.10: To demonstrate the dynamic range of the TDPE, the elastograms at different compression ratios are shown in this figure. The inclusion is still detectable at 10% compression (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

size of the median filter increases, the boundary of the inclusion fades. The strain at the center line of each elastogram shows this fact better. As the kernel size of the median filter increases, the ripple in both regions decreases which shows the improvement in SNRe. However the slope of the transition region decreases which means the resolution degrades. But this time the resolution does not degrade with the same rate as LSE.

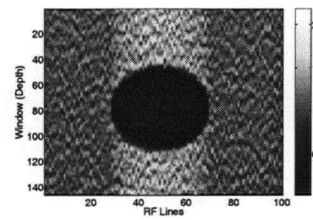
For further qualitative evaluation of the TDPE, another 2D phantom was simulated which consists of three layers with a harder layer inside (Fig. 5.7). These types of phantoms are also widely used since they simulate different layers of tissue inside the body. Typically ultrasound images from the body start with a layer of fat, which is soft, and another layer of muscle, which is hard, following another soft layer. 1D simulation was used to generate the RF data at 1% compression and TDPE was applied to estimate the time-shifts. The effect of the LSE on the elastogram and the trade-off between SNRe and R_a is shown in Figure 5.13 and the performance of the median filter on the strain image, estimated by the gradient operator, is shown in Figure 5.14.



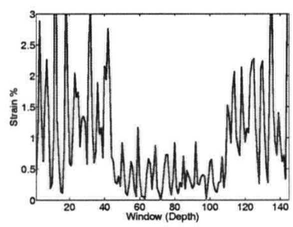
(a) Elastogram with gradient.



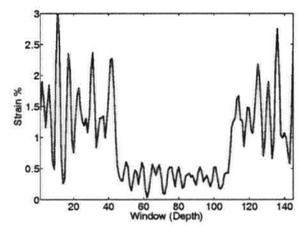
(b) LSE with kernel size of 4.



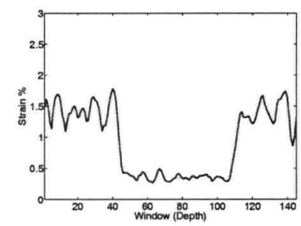
(c) LSE with kernel size of 8.



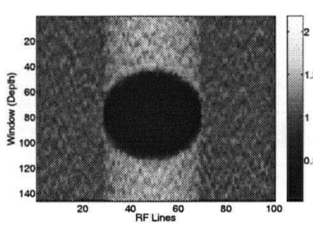
(d) Center line of elastogram.



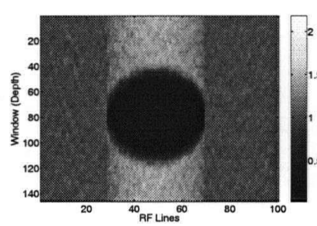
(e) Center line of LSE(4).



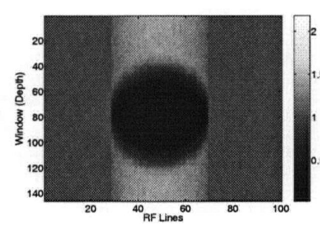
(f) Center line of LSE(8).



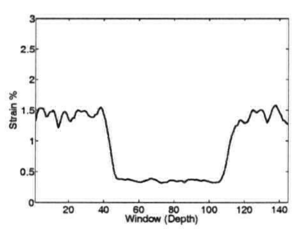
(g) LSE with kernel size of 12.



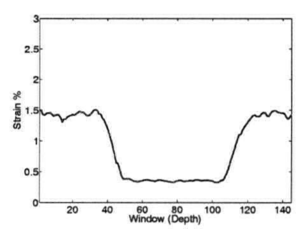
(h) LSE with kernel size of 16.



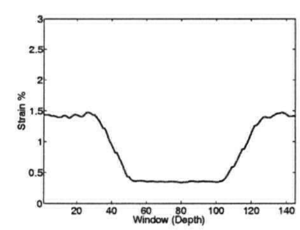
(i) LSE with kernel size of 24.



(j) Center line of LSE(12).

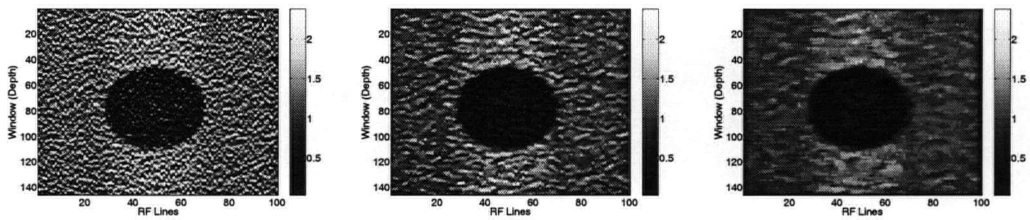


(k) Center line of LSE(16).

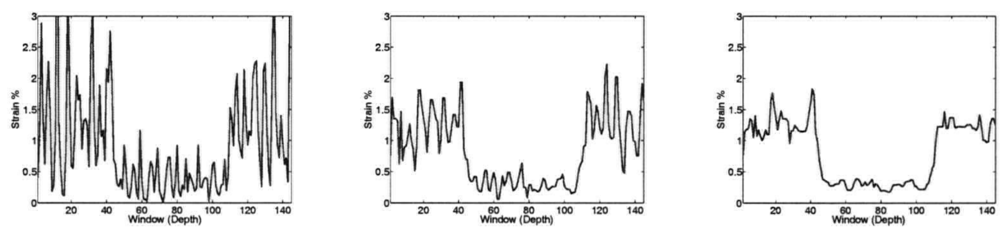


(l) Center line of LSE(24).

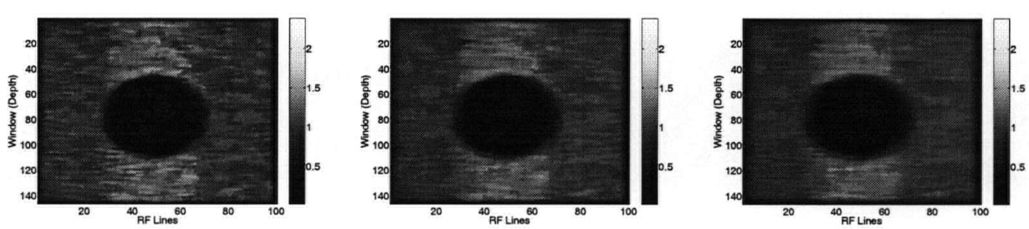
Figure 5.11: The improvement in elastogram when the Least Squares Strain Estimator (LSE) with different kernel sizes are used instead of gradient estimator. As the kernel size increases the elastograms improve rapidly. The smoother elastograms also show the loss of resolution. The boundary of the inclusion fades as the kernel size becomes bigger (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).



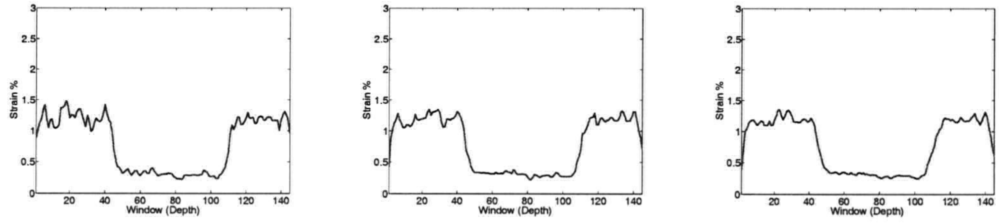
(a) Elastogram with gradient (1%). (b) Median with 3 × 3 kernel. (c) Median with 5 × 5 kernel.



(d) Center line of elastogram. (e) Center line of 3 × 3 kernel. (f) Center Line of 5 × 5 kernel.

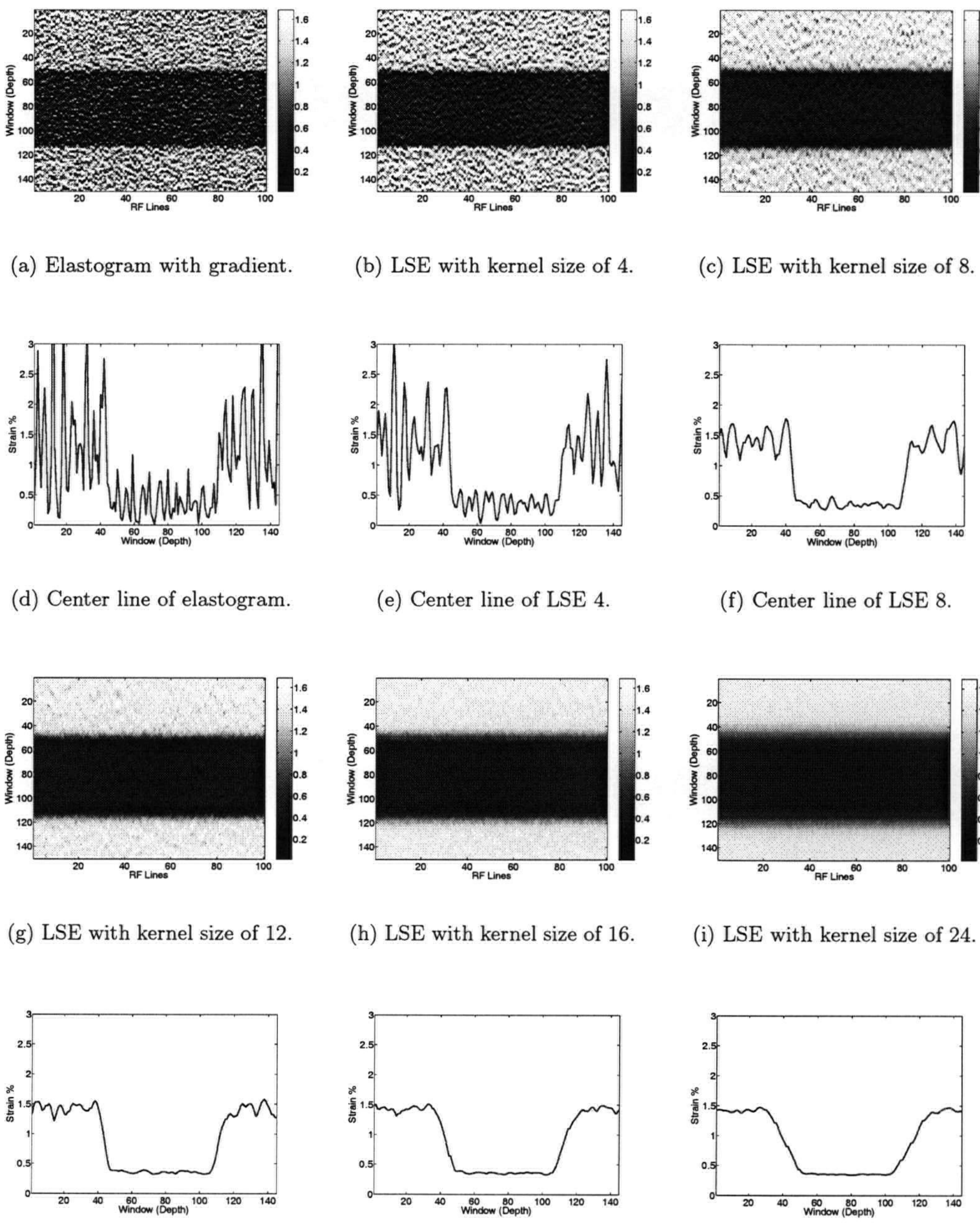


(g) Median with 7 × 7 kernel. (h) Median with 9 × 9 kernel. (i) Median with 11 × 11 kernel.



(j) Center line of 7 × 7 kernel. (k) Center line of 9 × 9 kernel. (l) Center line of 11 × 11 kernel.

Figure 5.12: To demonstrate the effect of median filtering in the elastograms' (SNRe) the original elastogram is compared with median filtered elastograms with different kernel sizes. As the kernel size increases the elastograms improve rapidly. The filtered elastograms also show the loss of resolution since the boundary of the inclusion fades as the kernel size becomes bigger (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).



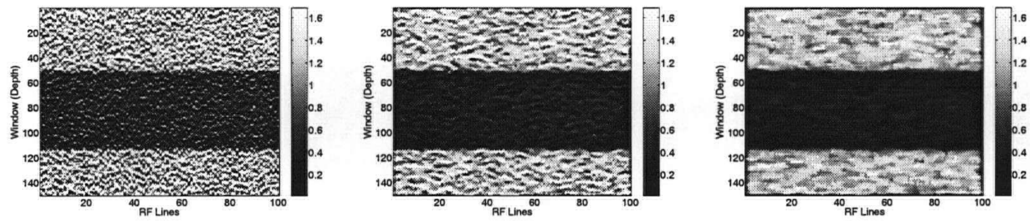
(a) Elastogram with gradient. (b) LSE with kernel size of 4. (c) LSE with kernel size of 8.

(d) Center line of elastogram. (e) Center line of LSE 4. (f) Center line of LSE 8.

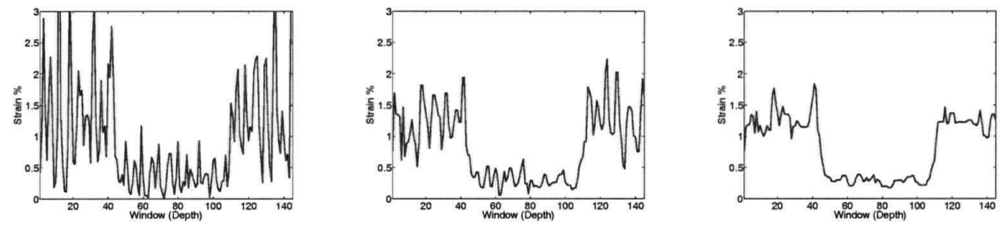
(g) LSE with kernel size of 12. (h) LSE with kernel size of 16. (i) LSE with kernel size of 24.

(j) Center line of LSE 12. (k) Center line of LSE 16. (l) Center line of LSE 24.

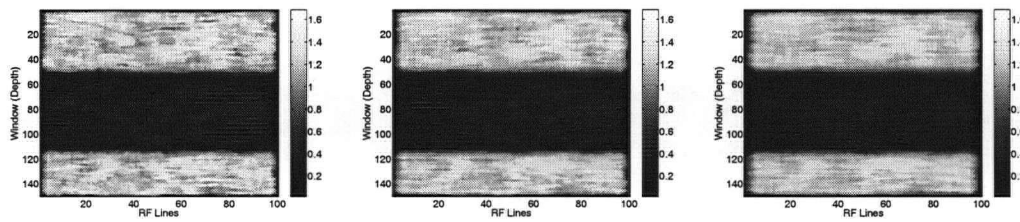
Figure 5.13: The improvement in elastogram when the Least Squares Strain Estimator (LSE) is used instead of gradient estimator. As the kernel size increases the elastograms' (SNRe) improves rapidly while the resolution degrades (the boundary of the layeres fade) (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).



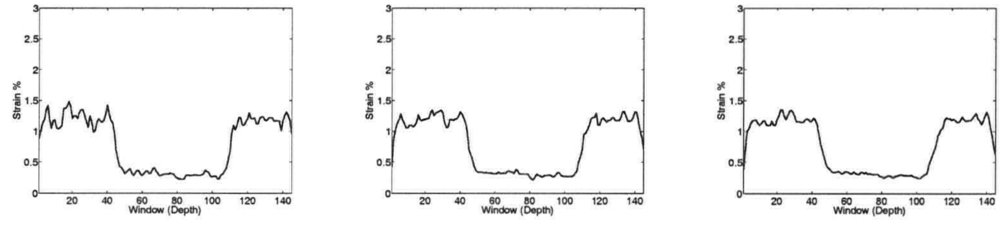
(a) Elastogram with gradient (1%). (b) Median with 3 × 3 kernel. (c) Median with 5 × 5 kernel.



(d) Center line of elastogram. (e) Center line of 3 × 3 kernel. (f) Center Line of 5 × 5 kernel.



(g) Median with 7 × 7 kernel. (h) Median with 9 × 9 kernel. (i) Median with 11 × 11 kernel.



(j) Center line of 7 × 7 kernel. (k) Center line of 9 × 9 kernel. (l) Center line of 11 × 11 kernel.

Figure 5.14: The original elastogram is compared with Median Filtered elastograms with different kernel sizes. As the kernel size increases the elastograms' (SNRe) improve rapidly while the resolution degrades (the boundary of the layers fade) (D=40 mm, W=1 mm, overlap=75%, f_o=5 MHz and f_s=40 MHz).

Similar to previous simulations, Fig. 5.13 and 5.14 show the trade-offs between the elastographic signal-to-noise ratio (SNRe) and axial resolution (R_a). As the kernel size increases for both LSE and Median Filter the elastogram improves rapidly due to the improvement in SNRe while the boundary between different layers start to fade due to the degradation in axial resolution.

5.3 Conclusion

The TDPE algorithm was validated in this chapter. While the computational cost of TDPE was shown to be at least 1/25 of TDE and 1/10 of CAM and PRS, it maintains the high SNRe, CNRe and R_a. 1D simulations of RF data showed that TDPE is even sensitive to 0.01% compression and is capable of estimating strain up to 10% while the TDE fails. Study of the axial resolution of the TDPE showed that it is capable of detecting the inclusions as small as 0.3 mm (W=1mm, 75% window overlap and $f_o = 5$ MHz). Having these properties makes the TDPE a suitable method for real time motion tracking and strain imaging.

It was also shown that the SNRe of TDPE can be improved if the Least Squares Estimator is used instead of the gradient operator but a trade-off exists, between the improvement in SNRe and degradation in axial resolution. We showed that for a kernel size of N the SNRe of TDPE improves with $N^{3/2}$ while the resolution degrades linearly with N.

The effect of median filtering was also presented in this chapter. It was shown that it causes both a nonlinear improvement in SNRe and a nonlinear degradation in axial resolution. It was shown that the effect of median filtering on the axial resolution was negligible while it provided a good improvement in SNRe. Therefore it seems to be a good filter for elastogram.

Chapter 6

Experiments

In this chapter, two types of tissue mimicking materials (phantoms) are tested to validate the TDPE algorithm with real RF signals from the Ultrasonix Rp500 ultrasound machine (Ultrasonix Medical Corporation, Burnaby, BC). The chapter starts with a description of the experimental setup. Then, the construction of two types of tissue mimicking materials is detailed. The experimental results are presented next, and remarks on the results conclude the chapter.

6.1 Experimental Setup

In order to validate the algorithm qualitatively with real RF data rather than simulated RF data, a number of experiments were run. An overview of the set-up is illustrated in Figure 6.1. The experimental set-up is composed of an ultrasound machine, an accurate compression device, a platform for analyzing tissue mimicking materials and a PC for off line processing.

A 5-12 MHz linear probe connected to a PC-based ultrasound machine (Ultrasonix RP500, Ultrasonix Medical Corporation, Burnaby, BC) was used for all scans and RF data capturing. The probe was placed on one side of the rectangular tissue mimicking material, and the compressor was placed on the opposite side. No lateral confinement was used in the whole experiment. The tissue mimicking material was lifted from the bottom surface during the experiments to prevent friction

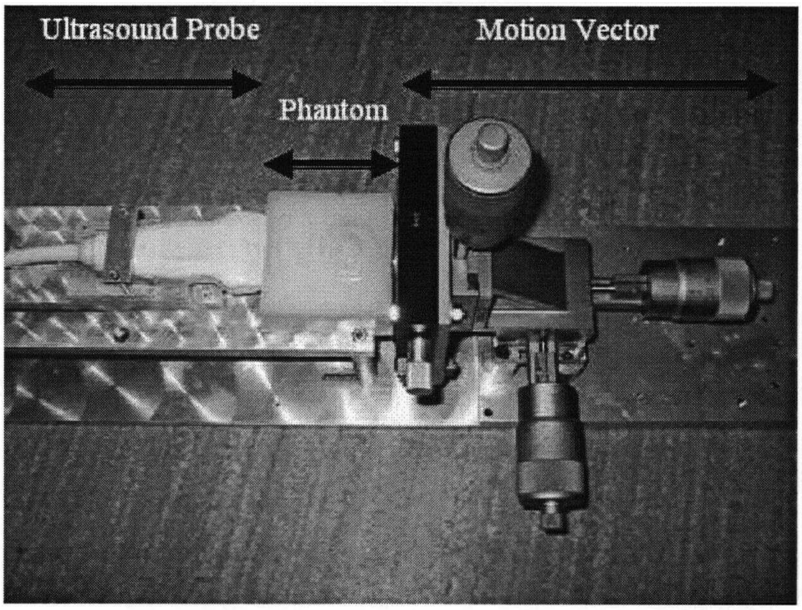


Figure 6.1: Overview of the experimental set-up

between the tissue mimicking material and the test platform. The tissue mimicking material was squeezed from both sides between the probe and the compressor, to reduce the lateral motion on the contact surfaces. The phantom was placed in a pre-compression mode and pre-compression RF data were captured by ultrasound machine.

To capture the post-compression RF data at a certain compression ratio, by considering the size of the phantom, the amount of compression was converted to an axial displacement. For instance, if the phantom is 6 cm in its initial state, in order to apply a 1% axial compression the phantom needs to be compressed 0.6 mm in the axial direction. By using the motion vector the phantom was compressed in the axial direction until the required compression was applied to the phantom. After this step the second set of RF data were captured as a post-compression data. These two sets of RF data were used later in a motion tracking algorithm to generate the static elastogram on another PC in an off line process by using Matlab code.

The same software that had been used for simulated RF data was used for off line processing. The software reads the two sets of RF data (one from pre- and one from post-compression) and runs a TDPE motion tracking algorithm (Section 2.2) to find the displacement estimates. LSE was used as a strain estimation method. The generated elastograms were then color coded and displayed on the screen.

Because of the slippery surface of the agar phantom, and to reduce the experimental error, each experiment was done separately, with separate pre- and post-compression RF data, instead of capturing one pre-compression data and several post-compression data.

6.2 Phantom Materials

A phantom is a tissue mimicking material. In the thesis, both acoustic and mechanical properties of tissue are considered in phantom construction. This includes the texture of the ultrasound image, the speed of sound, attenuation and stiffness [18, 78–82].

The phantoms used in this work are made from agar. Agar is widely used in the field of ultrasound elastography because it is easy to produce, non-toxic and its acoustic and mechanical properties is close to the human tissue [41]. Agar is a water-based phantom. In order to produce an agar phantom the water is heated up to 90° C. The required percentage of agar powder is then added to the water. This step needs to be done very slowly to make sure all the powder is disolved in the water. The stiffness is controlled by adjusting the proportion of agar powder in the aqueous solution [41]. The mixture is then cooled down to 80° C. Then a realistic speckle pattern of the B-mode image is produced for agar phantoms by adding cellulose scattering particles (2% cellulose in these experiments) into the liquid mixture solution. Then the final mixture is cooled in a mould until it congeals (around 26 ° C [41]). To be consistent with the simulations, two types of phantoms were made, one with a harder inclusion and one layered phantom.

6.3 Strain Filter Study

To generate the SF of the TDPE algorithm a homogeneous phantom ($6 \times 6 \times 6cm^3$) was made. The phantom was prepared using a 2% agar solution with 2% cellulose as scatterers. The phantom was imaged to a depth of 40 mm with a linear array of 100 elements with a 10-MHz centroid frequency, 80% fractional bandwidth transducer and digitized at 40 MHz. The phantom was compressed from 0.2% to 5% in steps of 0.2% strain (25 experiments) and in each compression ratio it was imaged once for pre-compression RF data and once for post-compression RF data.

TDPE with $W = 1$ mm and 75% window overlap (160 window in 40 mm) was used to estimate the motion at each step and the gradient operator was used to estimate the strain. The SNRe was estimated at each compression ratio according to Equation (5.1).

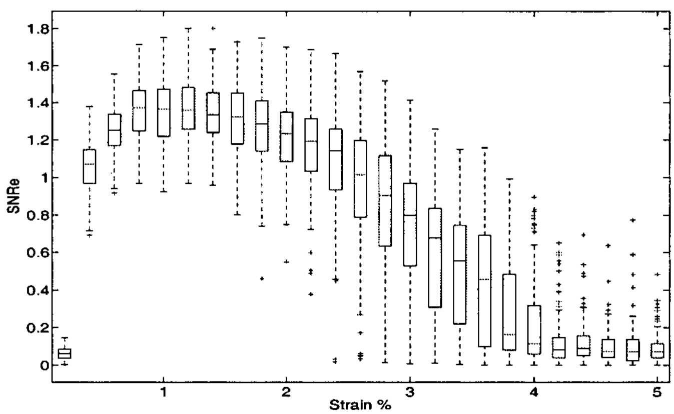


Figure 6.2: The strain filter of the TDPE algorithm with real data from experiments. Compression is done from 0.2% to 5% in steps of 0.2%. The result from 100 RF lines are used to show the mean values and the deviation from the mean values. The gradient operator is used as a strain estimator without any filtering.

The Matlab boxplot function was used to show the mean value and the deviation from the mean value of the SNRe for 100 RF A-lines. It can be seen from Figure 6.2 that TDPE is capable of estimating strains of up to 4% compression.

As we expected the result from simulation data over-estimates the performance of the TDPE algorithm. This could be explained by the fact that the tissue motion that occurs during compression is three dimensional (3D) while it was assumed to be 1D in simulation data. Furthermore agar phantoms have slippery boundaries and they trend to slip mostly at high compression ratios. This change in the probe's field of view will cause the motion tracking to fail.

6.4 Qualitative Results

An experiment was run for a phantom with a harder inclusion. The experiment was performed on a $6\times6\times6cm^3$ uniformly elastic phantom with a cylindrical inclusion. The center of the cylindrical inclusion (radius of 7 mm) was placed 15 mm below the surface. The phantom was prepared using a 2% agar solution with 2% cellulose as scatterers for the background and 4% agar and 2% cellulose for the hard inclusion.

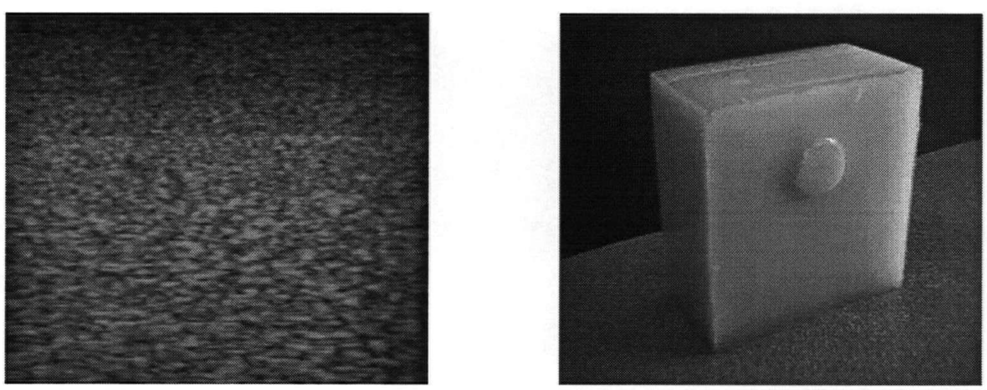
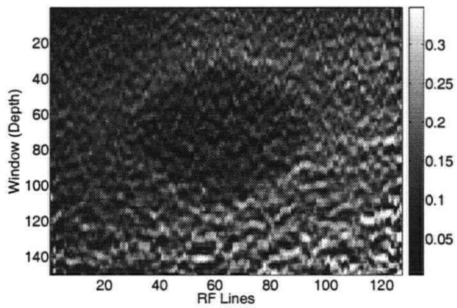
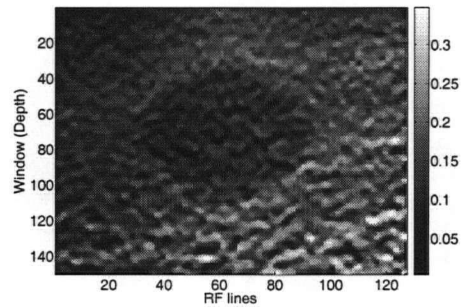


Figure 6.3: Image on right shows the phantom with cylindrical inclusion and the image on left shows its corresponding B-mode image.

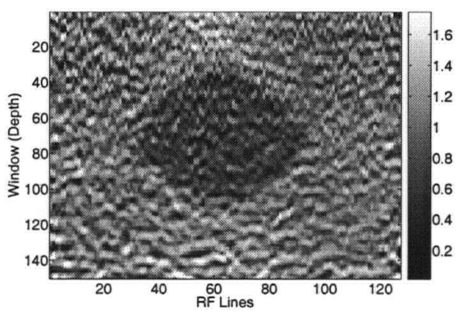
The phantom was imaged to a depth of 50 mm with a linear array of 100 elements with a 10-MHz centroid frequency, 80% fractional bandwidth transducer and digitized at 40 MHz. An image of the phantom with a regular ultrasound B-mode image is shown in Figure 6.3. Since the



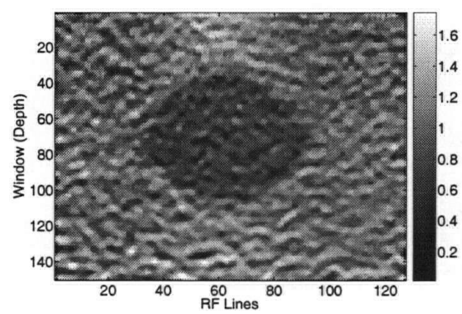
(a) Elastogram at 0.1% compression.



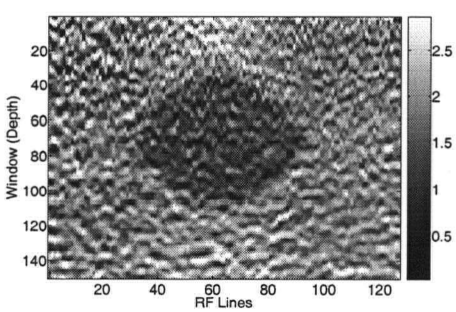
(b) Median filtered elastogram at 0.1%.



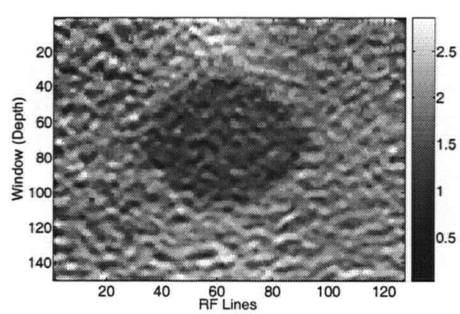
(c) Elastogram at 1% compression.



(d) Median filtered elastogram at 1%.



(e) Elastogram at 2% compression.



(f) Median filtered elastogram at 2%.

Figure 6.4: Elastograms of a phantom with 14 mm inclusion at 0.1%, 1% and 2%. LSE with kernel size of 6 is used to estimate the strain. Elastograms without applying a median filter (left) and median filtered with kernel size of 3×3 (right) are shown (D=50 mm, W=1 mm, overlap=75%, f_o=10 MHz and f_s=40 MHz).

amount of cellulose (scatterers) in both the background and the inclusion is the same, the inclusion is not visible in the regular B-mode.

The phantom was imaged once in rest position (pre-compression data). It was then compressed uniformly and it was imaged again to capture the post-compression RF data. The experiment was repeated once for 0.1%, once for 1% and once for 2% compression. After capturing RF data time shifts were estimated by applying the TDPE algorithm. Strain was then estimated by the LSE method with a kernel size of 6. The elastograms at strains of 0.1%, 1% and 2% are shown in Figure 6.4. Due to the stiffness contrast between the background and inclusion, the inclusion is quite visible in all elastograms.

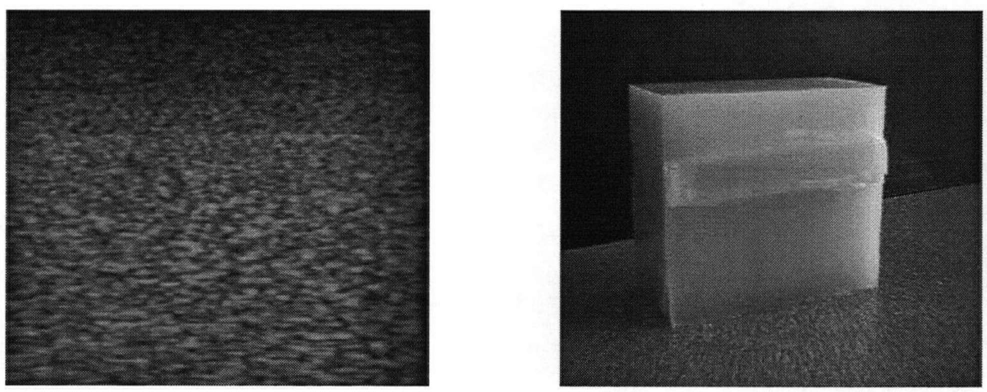
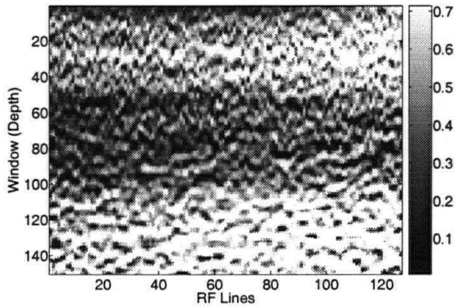


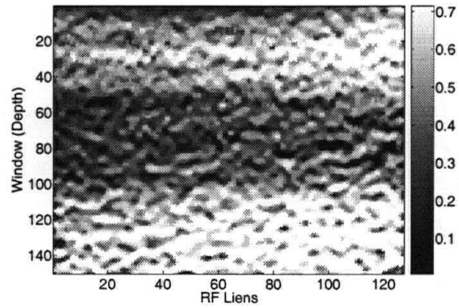
Figure 6.5: The image on the right shows the phantom with the cylindrical inclusion and the image on the left shows its corresponding B-mode image.

Another experiment was run for a three layered phantom. The experiment was performed on a $6 \times 6 \times 6cm^3$ uniformly elastic phantom with a harder layer inside. The center of the harder layer was placed 15 mm below the surface. The phantom was prepared using a 2% agar solution with 2% cellulose as scatterers for upper and lower layers and 4% agar and 2% cellulose for the hard layer inside.

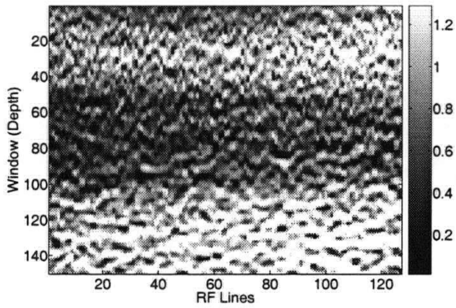
Similarly to previous experiments, the phantom was imaged to a depth of 50 mm with a linear array of 100 elements with a 10-MHz centroid frequency, 80% fractional bandwidth transducer and digitized at 40 MHz. An image of the phantom with a conventional ultrasound B-mode image is



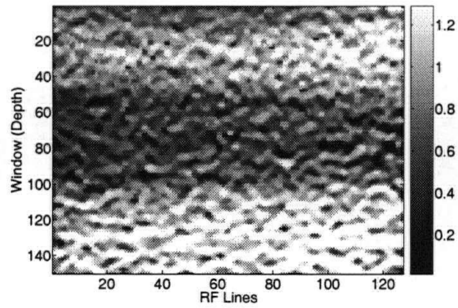
(a) Elastogram at 0.5% compression.



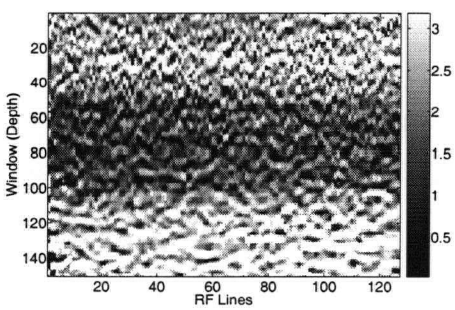
(b) Median filtered elastogram at 0.5%.



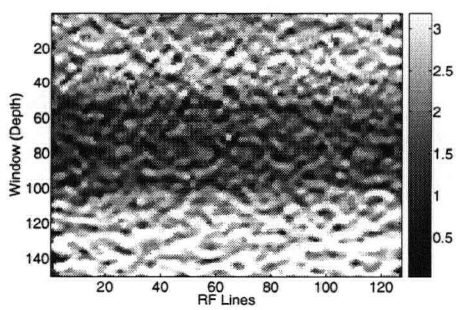
(c) Elastogram at 1% compression.



(d) Median filtered elastogram at 1%.



(e) Elastogram at 2% compression.



(f) Median filtered elastogram at 2%.

Figure 6.6: Elastograms of a phantom with 14mm inclusion at 0.5%, 1% and 2%. LSE with kernel size of 6 is used to estimate the strain. Elastograms without applying a median filter (Left) and median filtered with kernel size of 3×3 (right) are shown (D=50mm, W=1mm, overlap=75%, f_o=10 MHz and f_s=40 MHz).

shown in Figure 6.5. As in the previous experiment, since the amount of cellulose (scatterers) in both background and inclusion is the same the inclusion is not visible in the regular B-mode image.

The phantom was imaged once in rest position (pre-compression data) and once after the compression (post-compression data). The phantom was compressed uniformly once 0.5%, once 1% and once 2%. After capturing RF data time shifts were estimated by applying the TDPE algorithm. Strain was then estimated by the LSE method with a kernel size of 6. The elastograms at strains of 0.5%, 1% and 2% are shown in Figure 6.6. Due to the stiffness contrast between the inside layer and background, the hard layer is detectable in all elastograms. The elastograms also show the improvement when median filter is applied to the strain images.

6.5 Conclusion

In this chapter, two types of tissue mimicking materials (phantoms) were tested to validate the TDPE algorithm with real RF signals from ultrasound machine. The results show that the TDPE algorithm is capable of estimating the motion from the sequences of RF data, captured from an ultrasound machine. Considering its computational cost, SNRe, CNRe, R_a and sensitivity TDPE seems to be an attractive option for real time application.

Chapter 7

Implementation

In this chapter, the current version of both strain imaging and stiffness imaging packages are explained. The flow chart of both programs is shown and the details in each box in the flow chart are explained. The interface to the current version of both programs with their features is demonstrated and the same tissue mimicking phantoms that were used in the previous chapter to generate the elastograms are used to validate the performance of the software. The chapter finishes with showing the current images that are generated in real time from the tissue mimicking materials when external deformations are applied to the phantoms.

7.1 Real Time Strain Imaging

A real time strain imaging system has been implemented on the Ultrasonix 500RP ultrasound machine (Ultrasonix Medical Corporation, Burnaby, BC), which is a PC-based machine with a single 3.2 GHz CPU that provides RF data in real time. 30% of the CPU is used by the main program (Ultrasonix service pack) itself and the remaining 70% of the CPU was used for RF data acquisition, motion tracking with TDPE algorithm, strain estimation with LSE, scaling, color-coding and superimposed display on the B-mode images, or on the screen beside B-mode images.

The flow chart of the software is shown in Figure 7.1. The main server which is the original software on ultrasound machine should be started before the strain imaging software since it provides both B-mode and raw RF data for elastography programs. Initially in the software a connection establishes with the main server. Following the initialization two independent threads are started (Fig. 7.1). One of the threads is used to generate B-mode and the other one is used for strain imaging. The details about each thread are explained in following sections.

7.1.1 B-mode Thread

The B-mode thread captures the B-mode data from the main server. Similarly to all conventional ultrasound machines, B-mode is then color coded in gray scale in RGB format. In order to display the image correctly in both B-mode and strain image, the software should know the information about the current ultrasound probe such as curvature, number of elements, pitch and so on. This information is then used to display the image on the screen correctly.

Ideally the information about the probe should be retrieved from the server, but in the current version of the software this information is set inside the software for different ultrasound probes and the operator needs to select the type of probe manually.

7.1.2 Strain Imaging Thread

Strain is estimated from the displacement estimate. To estimate the displacement two sets of RF data are needed. As shown in Fig 7.1, the thread is an infinite loop. It always uses the current RF frame and the RF frame from the previous cycle to do the motion tracking.

Before performing motion estimation, the thread checks whether two RF frames have the same number of lines and number of samples at each line. This similarity check makes it possible to change the parameters of the main server such as depth, line density or even changing the probe without any need to close the strain imaging software. Therefore all parameter settings can be done in real time.

After the data acquisition process, RF frames are sent to the motion estimation block. The

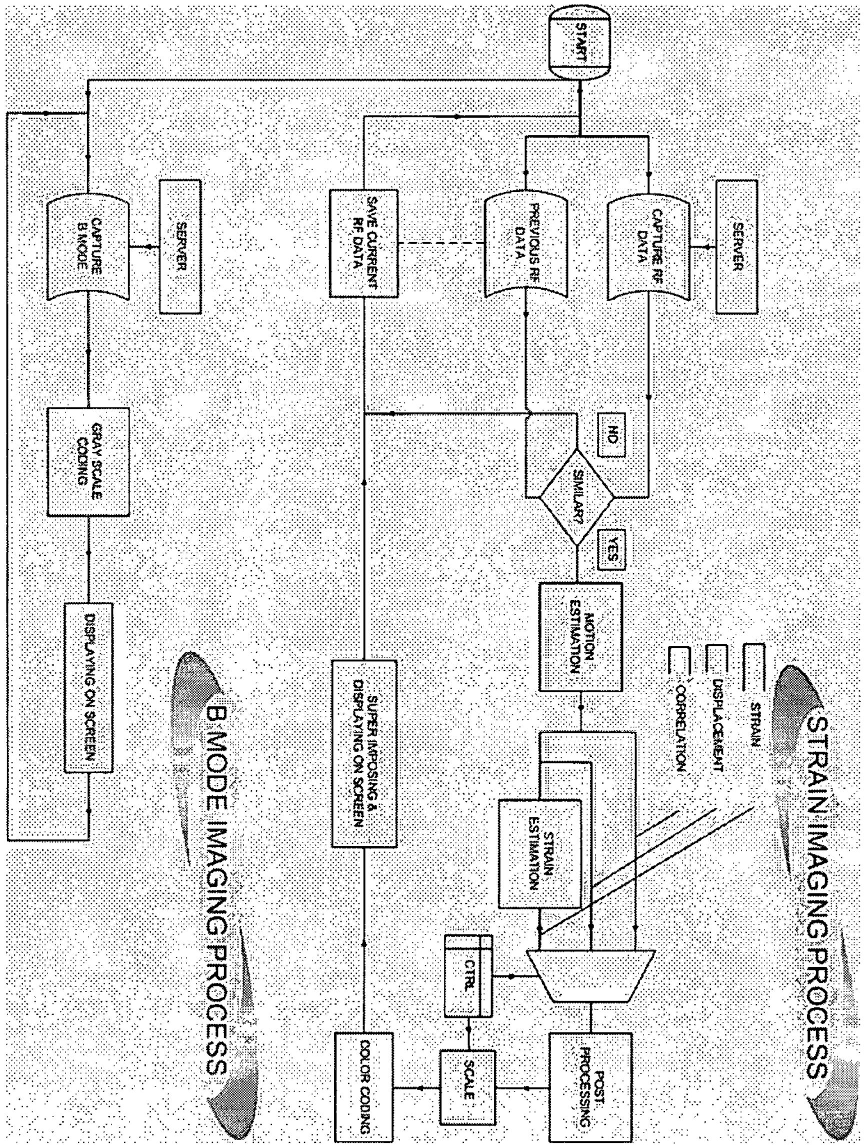


Figure 7.1: Flow Chart of the real time strain imaging system.

TDPE was used as a motion estimator due to its fast performance. Quadratic parabola fitting (please see Appendix B.2) was used as a subpixelling method to increase the accuracy of motion tracking and the three point Lagrangian interpolation method was used to recalculate the value of the correlation at the exact position of the correlation peak (please see Appendix B.3). 1D median filter was then applied on the displacement estimates to improve the process of motion tracking [130].

While it is not necessary to calculate the correlation image, it was added to the software since it shows the accuracy of the displacement estimation process. Although the cross correlation was set as a default correlation estimator, other correlation estimator methods (see Appendix B.1) are implemented in the software and it can be set for the TDPE to use each of them.

In addition to the selection of the correlation estimator, the number of windows, the length of each window and the size of the search region can also be set for the algorithm. Therefore the motion estimation block generates both an estimated displacement image and its corresponding correlation image.

7.1.2.1 Strain Estimation

LSE finds the best-fitting line to a set of points inside each kernel. The slope of the line is then reported as the estimate value of the gradient of the time-shifts at the center of the kernel [49]. As shown if Figure 7.2, these kernels have a large overlap with each other. Therefore, computation at each kernel may be useful for its ovelapping kernels.

Therefore, in order to investigate this idea, LSE was implemented. According to Equation (4.3) we can write A and d which are $N \times 2$ and $N \times 1$ matrices as follows:

$$A = \begin{bmatrix} i & 1 \\ i+1 & 1 \\ \vdots & \vdots \\ i+N-1 & 1 \end{bmatrix}, \hat{d} = \begin{bmatrix} d(i) \\ d(i+1) \\ \vdots \\ d(i+N-1) \end{bmatrix}, \tag{7.1}$$

then $[A^T A]^{-1}$ reduces to a 2×2 matrix and $A^T \hat{d}$ to a 2×1 matrix. Now if we define

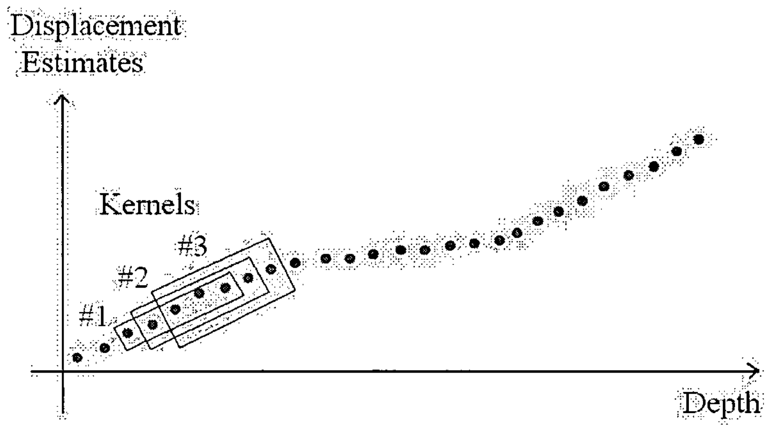


Figure 7.2: In the LSE method, neighboring kernels have large overlap with eachother. Three overlapping kernels of size 5 is shown.

$$A^T A = \begin{bmatrix} u & v \\ w & x \end{bmatrix}, A^T \hat{d} = \begin{bmatrix} y \\ z \end{bmatrix}, \tag{7.2}$$

the Least Squares Estimation (Equation (4.4)) reduces to

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} u & v \\ w & x \end{bmatrix}^{-1} \begin{bmatrix} y \\ z \end{bmatrix} = \frac{1}{ux - vw} \begin{bmatrix} x & -v \\ -w & u \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}. \tag{7.3}$$

Since we are looking for the value of $\hat{a}$ that is the slope of the crossing line, its value can be calculated as

$$\hat{a} = \frac{xy - vz}{ux - vw}. \tag{7.4}$$

The value of all u,v,w,x,y and z can be calculated from Equation (7.1) and Equation (7.2) which will reduce to:

$$\begin{aligned} u &= i^2 + (i+1)^2 + \ldots + (i+N-1)^2, \\ v = w &= i + (i+1) + \ldots + (i+N-1), \\ x &= 1 + 1 + \ldots + 1 = N, \\ y &= d(i)^2 + d(i+1)^2 + \ldots + d(i+N-1)^2, \\ z &= d(i) + d(i+1) + \ldots + d(i+N-1), \end{aligned} \tag{7.5}$$

All of the above equations can be calculated in a single iteration loop which iterates up to the size of the kernel. It should be noted that the iteration comes from a matrix multiplication in which each row should be multiplied by its corresponding column.

As can be seen from Equation(7.5), the values of u, v, w, x, y for estimating the slope of the current position can be used to calculate the values of u, v, w, x, y for the next position. Therefore after first calculation of u, v, w, x, y, the rest of them can be calculated as follows:

$$\begin{aligned} u_{new} &= u_{old} - i^2 + (i+N)^2, \\ v_{new} = w_{new} &= w_{old} - i + (i+N-1), \\ x_{new} &= x_{old} = N, \\ y_{new} &= y_{old} - d(i)^2 + d(i+N)^2, \\ z_{new} &= z_{old} - d(i) + d(i+N), \end{aligned} \tag{7.6}$$

This implementation of LSE makes it possible to change the kernel size of the least squares method in real time to improve the quality of the elastogram's or its resolution. The computational cost of this implementation of the LSE method has been presented in Appendix (C.1).

7.1.2.2 Post Processing

Post processing of the strain images is becoming a new field of research in elastography [72, 86]. Therefore, this block is considered in the current implementation of the software and a number of linear and nonlinear filters are implemented in this block.

Following the strain estimation block a control switch is used to select between the available images from correlation, displacement and strain. The same control switch is then used to scale the data [0 255]. In this way the software is capable of showing strain, displacement and correlation images in real time.

1. Median Filtering: A 2D median filter has been implemented in the software and is being used as a post processing filter. The kernel size can be set in the software and the operator has an option to apply it to the strain image or not.

   The Median Filter is normally used to reduce noise in elastography [23, 109]. It often does a better job than the Mean Filter of preserving useful detail in the elastogram. It considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings [36]. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.

   The most time-consuming part of the median filtering is the sorting process such that for a median filter with a kernel size of $n \times n$ one needs to sort n^2 numbers and select its median for each pixel. For small kernel sizes the selection of the sort algorithm has not much effect on the performance of the filter but as the kernel size of the increases the correct selection of the sort algorithm becomes important.

   If fast sorting algorithms such as *Heap sort*, *Quick sort* and *Merge sort* are used the order of complexity of median filter would be $n^2 Log_2(n^2)$ for each pixel [22] which makes it too intensive and not applicable for real-time applications. A fast 2D median filtering algorithm was introduced in [45] that reduces the complexity of 2D median filtering to $2n + 10$ which is much faster than the original version. They used the previously sorted data of the neighbor pixel to speed up the sorting process for the current pixel.

   In elastography software since the required kernel size was small (3×3) a simple implementation of the Median Filter was used by applying *insertion sort* [22].

2. Temporal Filtering: Strain is not an intrinsic property of the tissue and the amount of strain at each region depends on the amount of compression that is applied at that moment [88]. As a result, in high frame rate strain imaging systems, the strain images are not stable and they change rapidly on the screen. In order to have a more stable elastogram on the screen, several approaches have been implemented and embedded in the post processing block.

   In order to make the strain images more stable, strain images can be averaged and the average strain can be shown instead of showing each image independently. To avoid the extra buffer for this process, averaging was implemented according to the Equation (7.7) and the user can set the number of frames to be averaged:

$$S_{average,K} = \frac{K \times S_{average,K-1} + S_{current}}{K+1}, \tag{7.7}$$

   where K is a counter that resets to one, whenever it reaches its predefined limit, $S_{average,K-1}$ is the averaged strain at step $K-1$ and $S_{current}$ is the latest estimated strain.

   In this way the software starts to average two images then three and so on until it reaches the maximum averaging. Then it starts again and averages the first two images. In the current implementation, averaging does not register the elastograms which leads to losing the spatial resolution for large compressions, but since small compressions are mostly used in elastography this is not a significant problem.

   Exponential averaging is another method to generate more stable strain images [88]. It is simple to implement and does not need extra buffering. It works according to Equation (7.8). By changing the value of the α from 0 to 1 different elastograms can be generated. For values of α close to 0 the elastogram will mostly show the most current strain image which is close to unfiltered elastograms and for the values of α close to 1 the elastogram will mostly show the history of the strain which is weighted average and is much more stable. This post processing function was implemented in the software and the amount of filtering can be selected by setting the value of α in real-time.

$$S_{exponential,k} = (1-\alpha)S_{current} + \alpha S_{exponential,k-1}, \tag{7.8}$$

   In addition to these linear filters, nonlinear filters can also be applied to generate more stable elastograms. For example, instead of just average, maximum or median value of the strain among a number of images can be shown and the result can be called *median strain imaging* and *maximum strain imaging.* But since these kinds of imaging have not been studied yet they are not implemented in the current version of the software.

7.1.2.3 Displaying

Following the post processing procedure the final image is color coded. Five standard colormaps are implemented in the software (NORMAL, HOT, HSV, JET and GRAY) and the operator can easily switch between these colorcodes in real-time. Color coding provides much more dynamic range than the gray scale color mapping. For example in the HOT colormapping the input data should have a range of [0 768], in HSV it should have a range of [0 1280] while in the gray scale the data needs to be in the range of [0 255].

Initially, to superimpose the strain image on top of the B-mode image, the B-mode image was drawn on the screen first. The strain image was then converted to transparent and the transparent strain image was drawn on top of the B-mode image at the same location on the screen. This implementation made the software very slow and inappropriate for real time performance.

To fix this problem the superimposing was simply implemented by averaging B-mode and strain images that were already converted to RGB format. To keep the flow chart simple, the B-mode and strain threads are shown to be quite independent. But in order to do the averaging and superimposing, the two threads need to communicate with each other. In the current implementation, the strain thread provides the B-mode thread with its strain image in RGB format. In the B-mode thread, after drawing the original B-mode image on one side of the screen, to draw the superimposed image on the other side of the screen, the B-mode thread shrinks the same data to make the size of both B-mode and strain images equal. It then averages both images with adjustable opacity. It then draws the final image on the other side of the screen beside the original B-mode. To have an adjustable opacity the averaging was simply replaced with weighted averaging. Therefore to see more strain image the operator needs to increase the weight of the strain image and to see more B-mode the operator needs to increase the weight of the B-mode in the averaging process.

After color coding the data it is superimposed on the B-mode with adjustable opacity. It is then drawn on the screen correctly depending on the probe's type and specifications.

Finally at the end of the cycle the previous RF frame is dropped and the current RF frame is recorded in its place since it should be used as a previous frame for the next motion tracking cycle.

7.1.3 Software Features

Currently the system is capable of computing high frame rate strain images which are color coded and then superimposed on the B-mode image with adjustable opacity. The current frame rate of the strain imaging is provided in Table 7.1. The present interface of the software is also shown in Figure 7.3.

Important settings are gathered in the control console that can be located on the left side of the screen. The control box is divided into smaller boxes. The first box on top (*Configuration Options*) provides the operator to switch from simple averaging (Equation (7.7)) to exponential averaging (Equation (7.8)). The first slider on top makes it possible to change the value of K or α depending whether the simple or exponential averaging is desired. The second slider changes the kernel size of the LSE method which can be applied in real time. The operator can also select to activate the superimposing or deactivate it. If superimposing is active the third slider adjusts the opacity of the superimposing process.

The second box on the control box (*Color Map*) provides the user to select the appropriate colormap for the strain images. By clicking on each color map its corresponding colorbar appears in a box below. Currently the range of [0% to 2%] is colorcoded and higher strains are colorcoded similarly to 2% strain. This range can be set in the software.

The third box on the control box (*settings*) provides a number of options for the user. The first check box selects the processing of the full image or just a smaller rectangle at the center of the image. This helps the viewer to focus on the strain distribution at the center of the image. The second check box switches between cross correlation and normalized cross correlation (see Appendix B) in the TDPE method. The third check box activates or deactivates the median filter as a post processing filter. The fourth check box is considered for prostate imaging to switch the drawing

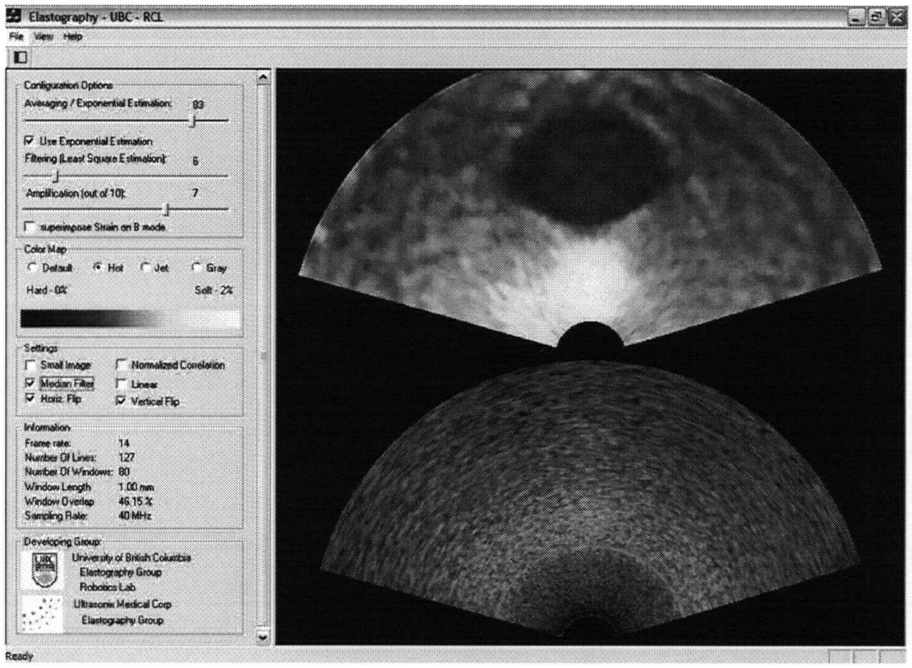


Figure 7.3: Current interface of the real time strain imaging system. The control box is on the left side of the screen and B-mode and strain images are drawn in the right side of the screen.

from a rectangular view (which is required for a linear probe in sagittal view), to triangular view (which is necessary for a sector probe in transverse view). The last two check boxes flip both B-mode and strain images in horizontal and vertical direction. This feature helps the doctor to understand the anatomy by aligning physical probe orientation with the image displayed.

There is another window (*Information*) which does not provide any setting and it only shows pertinent information in real time, such as frame rate, number of RF lines and number of windows which are being processed, the length of each window, window overlap in percentage and the sampling frequency of the ultrasound machine.

Finally, at the bottom of the control box (*Developing Group*) the information about the developers is included in front of their institute's logo.

RF Lines× windows	Pixels	Processing time	Algorithm frame rate	Sys. frame rate
256 × 256	65,000	55 ms	18 fps	14 fps
192 × 192	37,000	34 ms	29 fps	20 fps
128 × 128	16,000	20 ms	50 fps	30 fps
100 × 100	10,000	13 ms	77 fps	40 fps
64 × 64	4,000	5 ms	200 fps	60 fps

Table 7.1: The current frame rate of the real time strain imaging system on Ultrasonix RP500 with a single Pentium 4, CPU 3.2GHZ, 1GB RAM for a window size of 1mm (52 samples).

7.2 Real Time Vibro Elastography

Similarly to real time strain imaging, a real time stiffness imaging system based on vibro elastography [117] has been implemented. The flow chart of the software is shown in Figure 7.4 and the process at each box with its setting parameters are explained below. Like strain imaging, the main server should be started before the stiffness imaging software since it provides both B-mode and raw RF data. Initially, a connection is established with the main server. Following the initialization two independent threads are started. One of the threads is used to generate B-mode and the other one is used for stiffness imaging. The B-mode thread is the same thread as we used for strain imaging (see Section 7.1.1) except superimposing was not considered for stiffness images due to its low frame rate. The stiffness thread is discussed in following section.

7.2.1 Stiffness Imaging Thread

Stiffness imaging thread buffers the displacement estimates and runs a transfer analysis on the estimation displacement. The stiffness is then estimated by using the value of the transfer function at a specific range of frequency. Therefore, like strain imaging, two sets of RF data are used at each buffering cycle to estimate the motion. Due to its real time nature, TDPE is used as a motion estimator and Quadratic Parabola fitting B.2 was used as a subpixelling method to increase the accuracy of motion tracking. Similarly to strain imaging, the cross correlation was set as a default correlation estimator while all of the correlation estimator methods that introduced in B.1 have

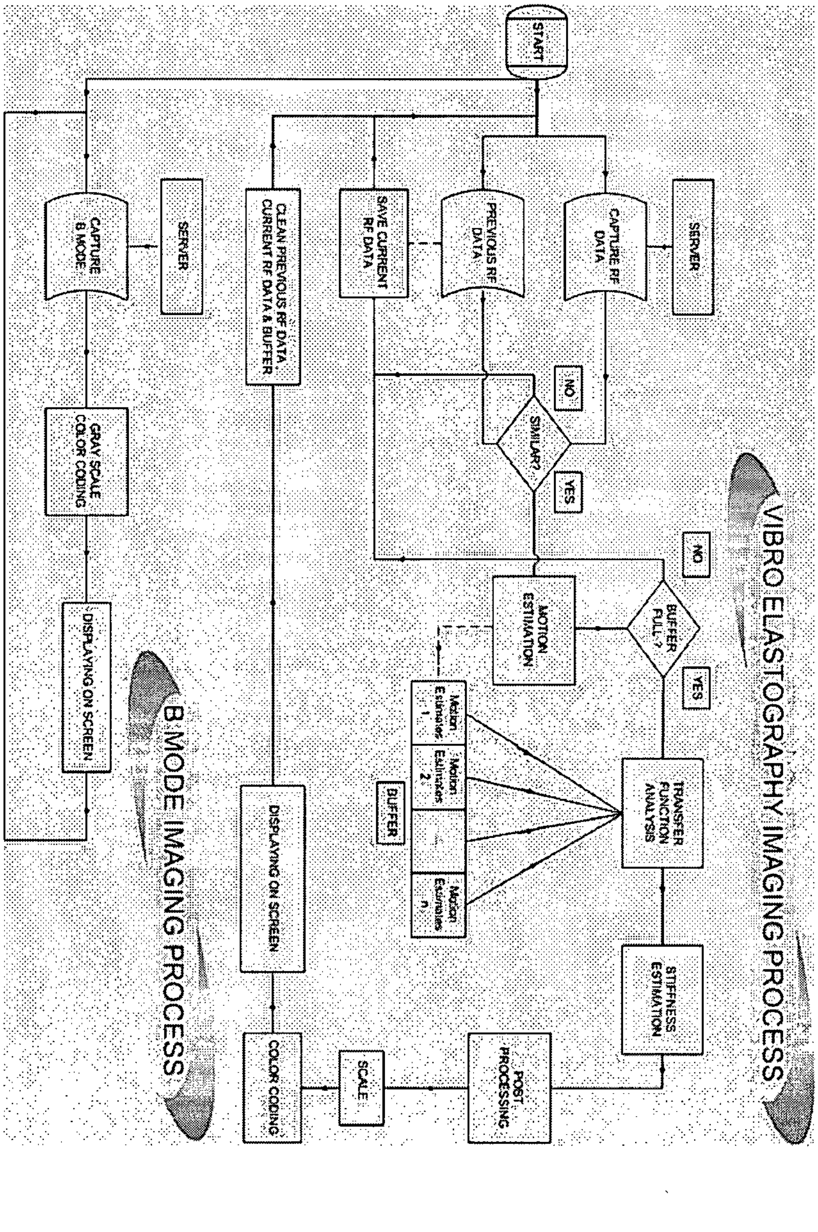


Figure 7.4: Flow Chart of the real time vibro elastography system.

been implemented in the software and can be used.

The motion estimation images are recorded in a large buffer. The software fills the buffer until it is full. Therefore, in addition to the selection of the correlation estimator, number of windows, length of each window and size of the search region, the size of the buffer, length of FFT window and percentage of overlap for FFT windows, can also be set for the algorithm. Currently the software buffers 16 frames of displacement estimates and FFT windows are 8 points with 50% overlap (3 windows).

Similarly to strain imaging, the software always checks two sets of RF data to make sure they are similar according to their number of lines and length of the lines. This similarity check makes it possible to change the parameters of the main server such as depth, line density or even changing the probe in real time without any need to close the software.

7.2.1.1 Transfer Function Estimation

As can be seen from Equation (1.3) $x_i(t)$ is repeated three times in the calculation of the PSD and CSD therefore if we combine two functions with a single Transfer Function Estimation, the number of required FFT calculations reduces to two series instead of four series since we just need to calculate the FFTs once for $x_i(t)$ and once for $x_j(t)$. Therefore the transfer function estimate at each window can be rewritten as:

$$h_i^j(\omega) = \frac{FFT(x_i(t)) \times conj(FFT(x_j(t)))}{FFT(x_i(t)) \times conj(FFT(x_i(t)))}, \tag{7.9}$$

The transfer function is then estimated by averaging this partial estimate of the transfer function at each window [132].

The performance of the transfer function estimate can be improved even further by noticing the fact that the transfer function is always estimated with respect to the reference input $(H_r ef^j(\omega))$. This means that during the whole process, the input of all transfer functions is the same therefore we just need to calculate its FFTs once and use it to estimate the transfer functions for other blocks. This implementation reduces the number of FFT calculations to one series for each block.

7.2.1.2 Stiffness Estimation

After estimating the value of the transfer function at each block with respect to the reference block, the relative stiffness value at each block is estimated according to [117]:

$$\begin{bmatrix} k_2 \\ k_3 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} H_1^2 - H_1^3 & 0 & \cdots & 0 & 0 \\ H_1^3 - H_1^2 & H_1^3 - H_1^4 & \cdots & 0 & 0 \\ \vdots & & \cdots & & \vdots \\ 0 & 0 & \cdots & H_1^n - H_1^{n-1} & H_1^n \end{bmatrix}^{-1} \begin{bmatrix} k_1(1 - H_1^2) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7.10}$$

This approach to estimate the relative stiffness has a lot of computational cost and is not applicable for real time purposes. Therefore in order to have this implemented in real time:

$$\begin{bmatrix} k_2 \\ k_3 \\ k_4 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} k_1(1 - H_1^2)/(H_1^2 - H_1^3) \\ k_1(1 - H_1^2)/(H_1^3 - H_1^4) \\ k_1(1 - H_1^2)/(H_1^4 - H_1^5) \\ \vdots \\ k_1(1 - H_1^2)/(H_1^n) \end{bmatrix} \tag{7.11}$$

This implementation for the stiffness estimation was used in real time applications. For further improvement in the result of stiffness estimates, similar to strain estimation (as explained in Section 4.5), the difference between the values of the transfer function was estimated by using the LSE method. This method results in a much smoother stiffness estimate. The tradeoff for using least squares method to estimate the stiffness has not been studied yet.

7.2.1.3 Post Processing and Displaying

Following the stiffness estimation block is the post processing block which is much simpler compared to the strain imaging software. Because of the buffering process, the frame rate of the stiffness imaging is slow (1 fps for 128 lines of RF data and 2 fps for 64 lines of RF data with 16 frames buffering). Therefore the averaging was not implemented and the post processing block for stiffness imaging only consists of a 2D median filtering. The kernel size of the filter can be set in the software.

The frame rate of the stiffness imaging improves if a ring buffer is used instead of simple buffer which means that each time, instead of removing all the data from the buffer, only a number of displacement estimates is removed and replaced with new displacement estimates. This implementation may help to increase the frame rate and will be considered in the next version of the software.

Similar to strain imaging software, after post processing the stiffness estimate, the data are scaled and colorcoded. To increase the contrast another amplification step was added to this process so that the user can scale the result in real time by changing the position of a slider in the control box. Five standard colormap was implemented in the software (NORMAL, HOT, HSV, JET and GRAY) and the operator can easily switch between these colorcodes. The same displaying process as in Section 7.1.1 was used for the stiffness image.

In the current implementation, the transfer function estimation and motion tracking run in the same thread (sequential) and the transfer function runs if the motion tracking buffer is full. Therefore, it is important to note that transfer function analysis is computationally intensive and takes some time. Therefore the current RF frame is not valid anymore for the next motion detection cycle and the software should drop both current and previous RF frames at the end of transfer function estimation cycle. The thread will start to capture RF data in its next cycle and will start to fill the buffer from the next stiffness estimation.

7.2.2 Software Features

Currently the system is capable of computing real time stiffness images. The most recent interface of the software is shown in Figure 7.5.

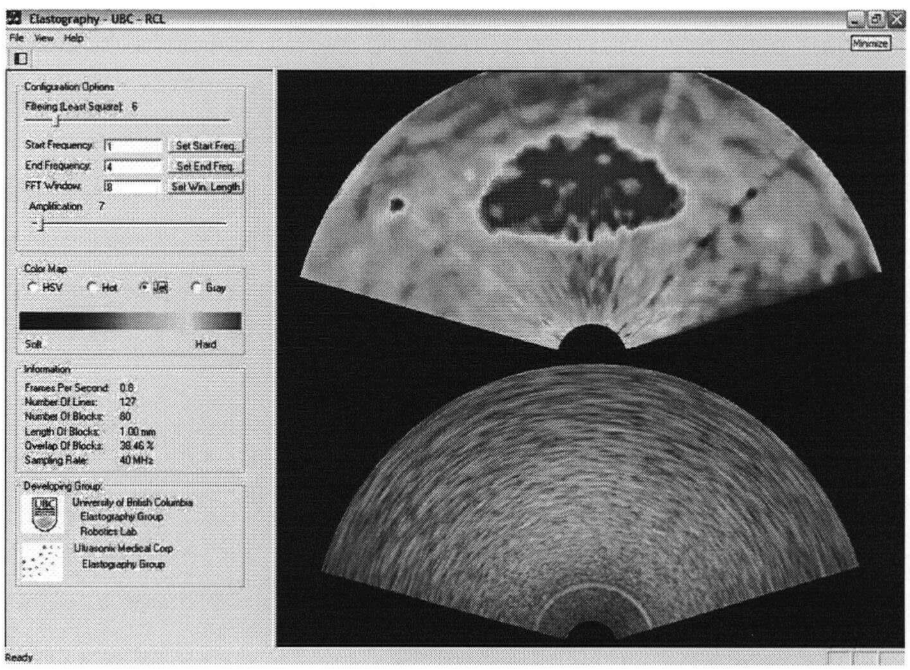


Figure 7.5: Current interface of the real time strain imaging system. The control box is on the left side of the screen and B-mode and strain images are drawn in the right side of the screen

The control box on the left is very similar to the strain imaging software except for the Configuration Box. In the vibro elastography software, besides the kernel size of the least squares and amount of amplification, the user can also choose the start frequency and end frequency of which to average the value of the transfer function with. The user can also change the length of the FFT window used in the transfer function estimate.

The real value of the *start frequency* and *end frequency* depends on the buffering frequency of the software. For example if the software is buffering with 16 fps, according to Nyquist rate the maximum frequency component that we can capture is 8 Hz. Therefore for a FFT window length of 8 (4 positions are used for real parts and 4 positions are used for imaginary parts). The position of 0 will be 0 Hz, position of 1 will be 2 Hz, position of 2 will be 4 Hz, position of 3 will be 6 Hz and position of 4 will be 8 Hz. Increasing the FFT length may help increase the resolution of each frequency components but it also requires more data to be buffered which makes the algorithm even slower.

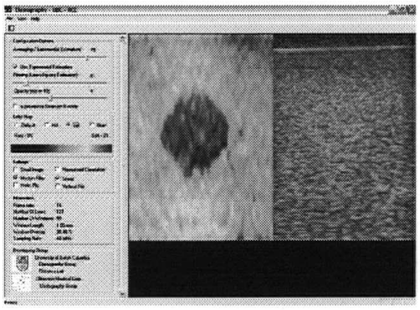
7.3 Verification

To show the current performance of both packages, they were tested on the same phantoms that were used in Section 6.4. One sample of both real time strain image and stiffness image from the phantom with inclusion and the three layered phantom is shown in Figure 7.6.

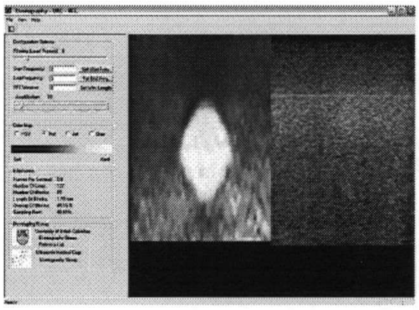
To generate the strain image both phantoms were deformed with an external vibrator (single frequency sinusoid with amplitude of 1.5 mm and frequency of 0.5 Hz). As it can be seen from the control box, the kernel size of LSE was set to be 6, exponential averaging ($\alpha = 0.79$) was used to stabilize the images and a 2D median filter (with a kernel size of 3×3) was applied to the strain images.

To generate the stiffness image both phantoms were deformed with an external vibrator (amplified white Gaussian noise bandpassed from 4Hz to 6Hz). The value of the transfer function was averaged from 4Hz to 6Hz to include the excitation frequency component. As it can be seen from the control box, the kernel size of LSE was set to be 6 and a 2D median filter (with a kernel size of 3×3) was applied to the stiffness images.

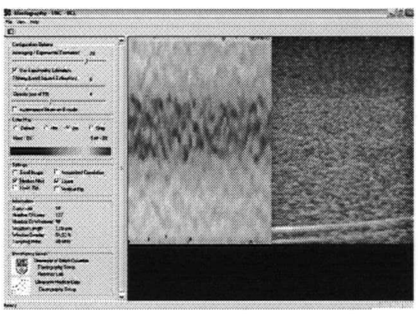
The results from Figure 7.6 confirm the correct implementation of both strain imaging and stiffness imaging programs. It should be noted that the same colorcoding for both strain image and stiffness image result in reverse colormap. This means dark region in strain image appears as a bright region in stiffness image. This phenomenon is due to the fact that soft region experience more strain and hard region experience less strain. Therefore hard inclusion will appear dark in



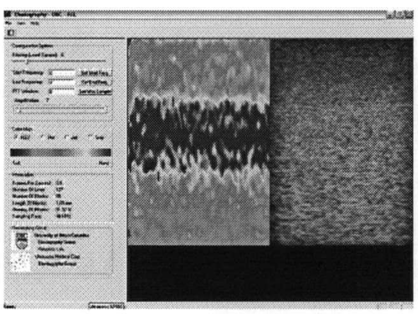
(a) Sample of real time strain image of a phantom with inclusion.



(b) Sample of real time stiffness image of a phantom with inclusion.



(c) Sample of real time strain image of a three layered phantom.



(d) Sample of real time stiffness image of a three layered phantom.

Figure 7.6: Screen shots of the interface and acquired images olbained with two developed software. Two types of tissue mimicking phantoms are tested. The strain image and stiffness image of a phantom with inclusion (top), and a three layered phantom (bottom) are shown.

the strain image and will appear bright in stiffness image.

For further verification the real time system was taken to the operating room and it was used on a real tissue in vivo. The idea was to check if the prostate will show up in the strain and transfer function images or not. Generally prostate tissue is stiffer than its surrounding tissue.

The actuation system is shown in Figure 7.7. The dual imaging probe with two sets of crystals make it possible to image in both sagittal direction (with linear array crystals) and in transverse direction (with convex array crystals). The probe was mounted in housing and was vibrated along its sagittal plane by two motor-driven four-bar linkages, along a direction set by the rotational stepper, with its depth adjusted by the translational stepper. A Simulink code was used as a controller to vibrate the probe with adjustable amplitude and range of frequencies.

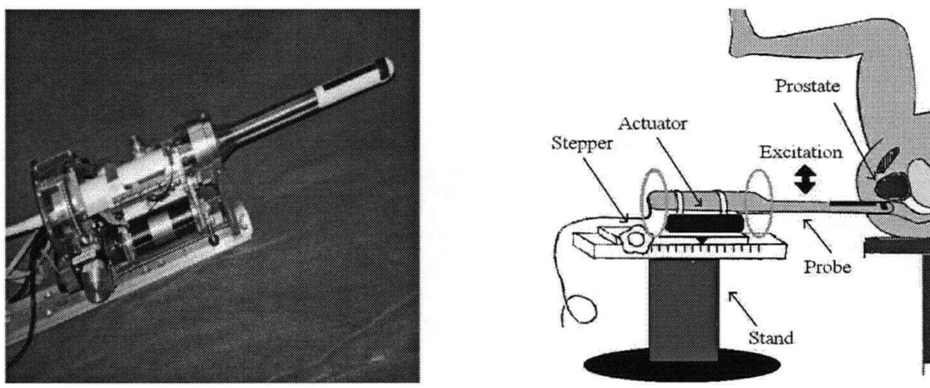


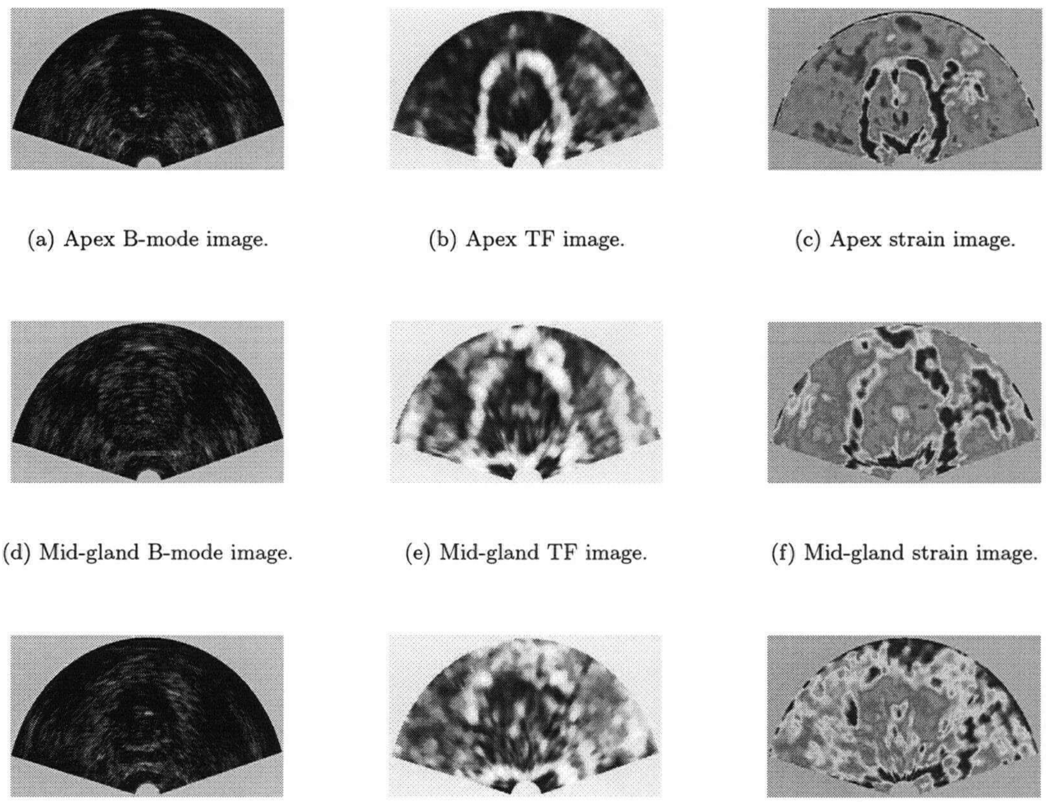
Figure 7.7: Photograph of actuation system, with the transrectal ultrasound (TRUS) probe mounted in it (left). A sketch of the procedure is also shown (right).

At each step, the probe was positioned with the doctor into the patient's rectum and the prostate was deformed with an external vibrator (amplified white Gaussian noise bandpassed from 0.5Hz to 5Hz) with maximum 2 mm peak to peak amplitude.

To calculate the transfer function, the length of FFT windows was set to be 8 and 50% overlap between windows were considered. The value of the transfer function was then averaged from 1Hz to 6Hz to include the excitation frequency components. The difference between the value of the transfer function was estimated with the LSE(10) and this difference was then shown as a transfer function image.

The strain images were estimated with LSE(8) and exponentially filtered ($\alpha = 0.9$) to generate stable images. The transfer function (TF) and strain images from the convex crystal array in the transverse direction are shown in Figure 7.8 besides their corresponding B-mode images.

As shown in Fig. 7.8, both elastograms (transfer function images and strain images) show the boundary of the prostate clearly mostly in apex and mid-gland.



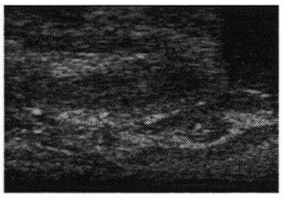
(a) Apex B-mode image. (b) Apex TF image. (c) Apex strain image.

(d) Mid-gland B-mode image. (e) Mid-gland TF image. (f) Mid-gland strain image.

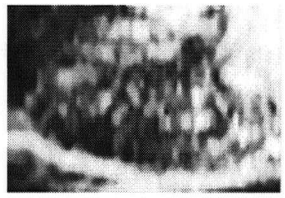
(g) Base B-mode image. (h) Base TF image. (i) Base strain image.

Figure 7.8: Prostate transverse B-mode images (left), transfer function images (middle) and strain images (right). The motion applied was 2 mm peak to peak, band-pass filtered (0.5-4.5Hz) white noise. The images show the prostate in apex (top), mid-gland (center) and base (bottom).

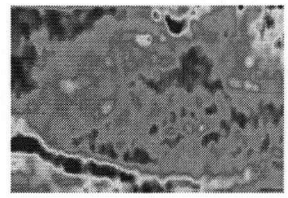
For further evaluation of the software, the images from sagittal planes are also presented. The elastograms are shown in Figure 7.9 with their corresponding B-mode images. As the probe rotates (counter clockwise), the prostate starts to move out of the probe's filed of view. This fact is quite visible in both transfer function and strain images while it is hard to see the boundary of the prostate in the B-mode images. As shown in Figure 7.9, both elastograms show much more contrast compare to the original B-mode images.



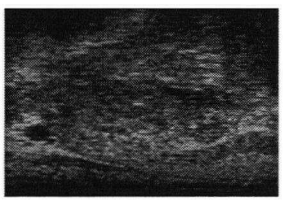
(a) Sagittal plane B-mode.



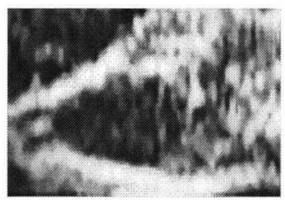
(b) Sagittal plane TF.



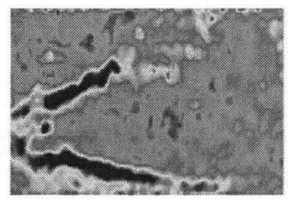
(c) Sagittal plane strain.



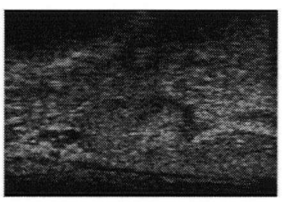
(d) Sagittal plane -10° B-mode.



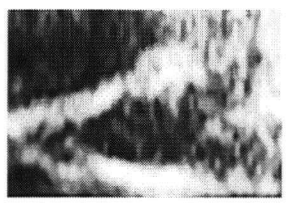
(e) Sagittal plane -10° TF.



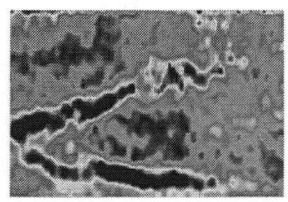
(f) Sagittal plane -10°strain.



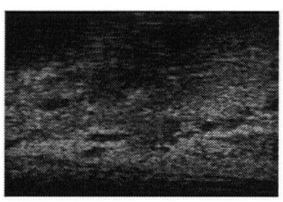
(g) Sagittal plane -20°B-mode.



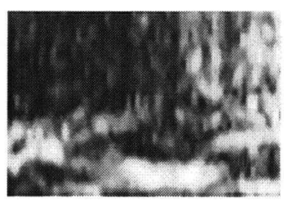
(h) Sagittal plane -20° TF.



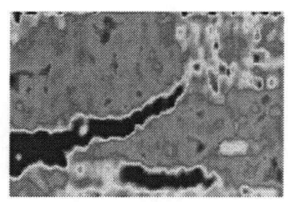
(i) Sagittal plane -20°strain.



(j) Sagittal plane -30° B-mode.



(k) Sagittal plane -30° TF.



(l) Sagittal plane -30° strain.

Figure 7.9: Prostate sagittal B-mode images (left), transfer function images (middle) and strain images (right). The motion applied was 2 mm peak to peak, band-pass filtered (0.5-4.5Hz) white noise.

7.4 Conclusion

In this chapter, the current version of both strain imaging and stiffness imaging programs were explained in detail according to their flow chart. The interface of the current version of both programs with their features was demonstrated. The result from tissue mimicking phantoms demonstrates the correct implementation of both programs. The software can perform in real time and in clinical applications without extra hardware overhead.

Chapter 8

Summary and Future Work

8.1 Summary

Two real time elastography packages (strain imaging and stiffness imaging) have been presented. Both programs include a new motion tracking algorithm. The strain imaging algorithm estimates the strain directly from the displacement estimates. The stiffness imaging buffers the displacement estimates and analyze the tissue motion with a Transfer Function. The stiffness of tissue is then estimated from the magnitudes of the transfer function. The specific contributions of this thesis can be summarized as follows:

- **Fast Motion Tracking Algorithm:** This thesis introduced a new method for real time motion tracking in ultrasound images which is very fast. The algorithm was named *Time Domain Cross Correlation with Prior Estimates* since it has a similar structure to that of standard *Time Domain Cross Correlation.* The standard method is very accurate for small strain but it is computationally intensive and not applicable for real time performance.

- **Comparison of Computational Cost:** *Computational cost* is introduced as a new comparison factor for different motion tracking algorithms and the numbers of multiplies and adds was used as a comparison factor. We compared the computational cost of the introduced method with the PRS [68] and CAM [102] methods that are the state-of-the-art methods in real time motion tracking algorithms [86, 105]. We showed that the proposed TDPE method is the most efficient algorithm among the current real time methods and is at least 25 times faster than the standard TDE method. Therefore TDPE offers an attractive option for real time motion tracking.

- **Strain Estimation Algorithms:** In addition to the gradient operator and LSE method as standard strain estimation algorithms, *numerical differentiation methods* have been introduced as an extension to the gradient operator and the least squares strain estimation method was extended to higher order polynomials.

- **Studying the Strain Filter of TDPE:** The performance of the TDPE was studied based on three important metrics, namely the strain filter, contrast-to-noise ratio and resolution with simulation data. The study of the strain filter showed that the proposed algorithm is very sensitive and that it can track motions due to very small compressions ($\approx 0.01\%$) and has a large dynamic range so it can track motions due to compressions as large as 10%. Furthermore by studying the axial resolution of the algorithm we showed that the TDPE algorithm is capable of detecting an inclusion as small as 0.3 mm (for $W = 1$ mm, 75% window overlap and $f_o = 5$ Mhz).

  In addition to the simulation data, results from tissue mimicking phantoms showed that the proposed algorithm is capable of accurately estimating the motion with small amount of computation. These properties make the proposed method an attractive option for the applications that need real time motion tracking.

- **Studying the Median Filter and the Least Squares Estimator:** For the first time the performance of the median filter has been presented in elastography. Quantitative data showed that the median filter improves the strain filter while it has a small negligible effect on the axial resolution. Qualitative data was then used to confirm these results. Moreover, the effect of the LSE on the contrast-to-noise and axial resolution has been studied.

- **Real Time Elastography System:** The proposed motion tracking algorithm was used in two real time elastography systems on the Ultrasonix 500 RP machine that require fast and accurate displacement estimator. The first system is real time strain imaging which shows the distribution of the strain due to an external deformation in real time. The strain is estimated from the displacement estimates. Therefore the most important part in real time strain imaging is its real time motion estimation which needs to be as fast as possible. Using TDPE as a motion estimator made it possible to have a high frame rate strain imaging system.

The second system is real time vibro elastography that extracts not only static but also dynamic properties of the tissue. A computer-controlled vibrator induces motion over a range of frequencies and simultaneously the resulting displacements are estimated and recorded at multiple locations and time instants. The motion data is then used to extract the mechanical properties of tissue. Therefore, similarly to strain imaging the bottle neck of the vibro elastography is its motion tracking part. To have it in real time, the motion estimation part needs to be as fast as possible.

Strain imaging software consists of data acquisition, motion tracking, strain estimation, color coding, superimposing and displaying the images on the screen. In addition to using TDPE as a motion estimator, the Least Squares Strain Estimator was also implemented according to Section 7.1.2.1 to speed up the strain estimation part.

Vibro elastography software consists of data acquisition, motion tracking, buffering, transfer function analysis, stiffness estimation and color coding, superimposing and displaying the images on the screen. Similarly to strain imaging, TDPE was used as a motion estimator.

Currently both elastography programs can perform in real time and are ready to use in clinical applications.

8.2 Future Work

The idea used in this paper to speed up the computation of displacement using cross-correlation can also be used for phase domain algorithms to do the phase unwrapping. Most of the current phase domain algorithms must perform additional calculations to avoid aliasing and to unwrap the phase. As in our method, the phase shift of neighbors can be used to unwrap the phase shift of the current window without additional calculations since the phase shifts of neighboring windows are expected to be close to each other.

Typically in elastography, the axial component of the strain tensor is estimated by taking the gradient of the axial (along the beam propagation axis) displacement occurring after a quasi-static tissue compression [18, 78–81] which requires a 1D motion tracking algorithm as proposed in this thesis. In general, however, the tissue motion that occurs during compression is 3D. Because the lateral (perpendicular to the beam propagation axis and in the scan plane) and elevational (perpendicular to the beam propagation axis and to the scan plane) motions are not measured, two major drawbacks are encountered [56]. First, the axial elastogram takes into account only a small part of the mechanical tissue motion information. Second, undesirable lateral and elevational motions are the primary causes of signal decorrelation [51, 52]. Therefore 2D and 3D motion tracking algorithms are preferred and need to be considered in the next version of the elastography programs.

The simulation environment will be upgraded to 2D and 3D model for generating the RF data which are more realistic compared to 1D RF data. 2D and 3D RF data will be used to investigate the performance of the TDPE algorithm. The same data will be used to investigate 2D and 3D motion tracking algorithms.

Currently the frame rate of the elastography software is limited by the ultrasound machine that can not provide RF data at a faster rate. This problem will be solved soon by the Ultrasonix Company.

Finally, clinical applications for *in vitro* and *in vivo* experiments will be investigated starting with prostate imaging for cancer detection, and RF ablation for cancer treatment.

Bibliography

[1] SR Aglyamov, AR Skovoroda, JM Rubin, M O'Donnell, and Emelianov SY. Young's modulus reconstruction in DVT elasticity imaging. In *In: Linzer M, ed. 27th International Symposium on Ultrasonic Imaging and Tissue Characterization*, page 174, Arlington, VA, 2002. Ultrason Imaging.

[2] S.K. Alam and J. Ophir. On the use of baseband and RF signal decorrelation as tissue strain estimators. *Ultrasound in Medicine and Biology*, 23:1427–1434, 1997.

[3] S.K. Alam and J. Ophir. Reduction of signal decorrelation from mechanical compression of tissues by temporal stretching: applications to elastography. *Ultrasound in Medicine and Biology*, 23:95–105, January 1997.

[4] S.K. Alam, J. Ophir, and E.E. Konofagou. An adaptive strain estimator for elastography. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 45:461–72, March 1998.

[5] S.K. Alam, J. Ophir, and T. Varghese. Elastographic axial resolution criteria: an experimental study. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 47:304–309, January 2000.

[6] JC Bamber and NL Bush. Freehand elasticity imaging using speckle decorrelation rate. In *IEEE Symposium on Acoustical Imaging*, pages 285–292 vol. 22, Florence, Italy, 1996. IEEE.

[7] JC Bamber and NL Bush. Freehand elasticity imaging using speckle decorrelation rate. In *Acoustical Imaging*, pages 285–292 Vol.22, Plenum Press, New York, 1996.

[8] J. Bang, T. Selbekk, and G. Unsgard. Strain Processing of Intraoperative Ultrasound Images of Brain Tumours. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 56, 2004.

[9] J.S. Bendat and A.G. Piersol, editors. *Random Data. Analysis and Measurement Procedures, 2nd Edition.* John Wiley and Sons, 1986.

[10] M Bertrand, M Meunier, M Doucet, and G. Ferland. Ultrasonic biomechanical strain gauge based on speckle tracking. In *IEEE Ultrasonics Symposium*, page 859864 vol.2. IEEE, 1989.

[11] M. Bilgen and Insana MF. Predicting target detectability in acoustic elastography. In *IEEE Ultrasonics Symposium*, pages 1427–1430 vol.2. IEEE, 1997.

[12] L.N. Bohs and G.E. Trahey. A novel method for angle independent ultrasonic imaging of blood flow and tissue motion. *IEEE Transactions on Biomedical Engineering*, 38:280–6, March 1991.

[13] O. Bonnefous. Measurement of the complete (3D) velocity vector of blood flows. In *IEEE Ultrasonics Symposium*, page 795799. IEEE, 1988.

[14] W.L. Brogan, editor. *Modern Control Theory, Third Edition*, chapter 6. Prentice Hall, 1991.

[15] I. Cespedes. *Elastography. Imaging of biological tissue elasticity.* PhD thesis, University of British Columbia, 1993.

[16] I. Cespedes, Y. Huang, J. Ophir, and S. Spratt. Methods for the estimation of subsample time-delays of digitized echo signals. *Ultrasonic Imaging*, 17:142–171, 1995.

[17] I. Cespedes and J. Ophir. Reduction of image noise in elastography. *Ultrasonic Imaging*, 15:89–102, 1993.

[18] I. Cespedes, J. Ophir, H. Ponnekanti, and N. F. Maklad. Elastography: elasticity imaging using ultrasound with application to muscle and breast in vivo. *Ultrasonic Imaging*, 15:73–88, 1993.

[19] P. Chaturvedi, M. Insana, and T. Hall. 2D companding for noise reduction in strain imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 45:179191, 1998.

[20] P. Chaturvedi, M. Insana, T. Hall, and M. Bilgen. 3D companding using linear arrays for improved strain imaging. In *IEEE Ultrasonics Symposium*, page 14351438, Toronto, Canada, October 1997. IEEE.

[21] A.D. Christensen, editor. *Ultrasonic Bioinstrumentation*, chapter 6. John Wiley and Sons, 1988.

[22] T. Cormen, C. Leiserson, R. Riverst, and C. Stein, editors. *Introduction to Algorithms. Second Edition*, chapter 3. MIT Press, 2001.

[23] E. Davies, editor. *Machine Vision: Theory, Algorithms and Practicalities*, chapter 3. Academic Press, 1990.

[24] P. de Jong, T. Arts, A. Hoks, and R. Reneman. Determination of Tissue Motion Velocity by Correlation Interpolation of Pulsed Ultrasonic Echo Signals. *Ultrasonic Imaging*, 12:84–98, 1990.

[25] C. De Korte, E. Cespedes, A. Van Der Steen, and C. Lancee. Intravascular elasticity imaging using ultrasound: Feasibility studies in phantoms. *Ultrasound in Medicine and Biology*, 23:735–746, 1997.

[26] C.L. De Korte, A.F.W. van der Steen, E.I. Cspedes, G. Pasterkamp, S.G. Carlier, F. Mastik, A.H. Schoneveld, P.W. Serruys, and N Bom. Characterisation of plaque components and vulnerability with Intravascular Ultrasound Elastography. *Physics in Medicine and Biology*, 45:1465–1475, 2000.

[27] P deJong, T. Arts, A. Hoeks, and R. Reneman. Experimental evaluation of the correlation interpolation technique to measure regional tissue velocity. *Ultrasonic Imaging*, 13:145–161, April 1991.

[28] RJ Dickinson and Hill CR. Measurement of soft tissue motion using correlation between A-scans. *Ultrasound in Medicine and Biology*, 8:263–271, 1982.

[29] D Dotti, R Lombardi, and P. Piazzi. Vectorial measurement of blood velocity by means of ultrasound. *Medical and Biological Engineering and Computing*, 30:219225, March 1992.

[30] M. Doyley, J. Bamber, F. Fuechsel, and N. Bush. A freehand elastographic imaging approach for clinical breast imaging: System development and performance evaluation. *Ultrasound in Medicine and Biology*, 27:1347–1357, 2001.

[31] V.N. Dvornychenko. Bounds on (deterministic) correlation functions with application to registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:206–13, March 1983.

[32] SY. Emelianov, X. Chen, M. O'Donnell, B. Knipp, D. Myers, T. Wakefield, and JM. Rubin. Triplex ultrasound: Elasticity imaging to age deep venous thrombosis. *Ultrasound in Medicine and Biology*, 28:757–767, 2002.

[33] SY Emelianov, MA Lubinski, WF Weitzel, and et al. Elasticity imaging for early detection of renal pathology. *Ultrasound in Medicine and Biology*, 21:871–883, 1995.

[34] H. Eskandari and T. Salcudean. Tissue Strain Imaging Using A Wavelet-Based Peak Search Algorithm. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 31, 2004.

[35] A. Fertner and A. Sjolund. Comparison of various time delay estimation methods by computer simulation. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 1329–1330 vol.34. IEEE, 1986.

[36] D. Forsyth and J. Ponce, editors. *Computer Vision: A Modern Approach.*, chapter 3. Prentice Hall, 2003.

[37] S.G. Foster, P.M. Embree, and W.D. O'Brien. Flow velocity profile via time-domain correlation: error analysis and computer simulation. *Ultrasonics, Ferroelectrics and Frequency Control*, 37:164–175, May 1990.

[38] D. Fu. *Spectral Strain Estimation Techniques for Tissue Elasticity Imaging.* PhD thesis, Drexel University, 2005.

[39] BS Garra, EI Cespedes, J Ophir, and et al. Elastography of breast lesions: Initial clinical results. *Radiology*, 202:79–86, January 1997.

[40] JT Hall, Zhu Y, CS Spalding, and LT Cook. In-vivo results of real-time freehand elasticity imaging. In *IEEE Ultrasonics Symposium*, page 1649:1653. IEEE, 2001.

[41] T.J. Hall, M. Bilgen, M.F. Insana, and T.A. Krouskop. Phantom materials for elastography. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 44:1355–65, November 1997.

[42] IA. Hein. Multi-directional ultrasonic blood flow measurement with a triple beam lens. In *IEEE Ultrasonics Symposium*, page 10651069. IEEE, 1993.

[43] I.A. Hein, V. Suorsa, J. Zachary, R. Fish, J. Chen, W.K. Jenkins, and W.D. O'Brien. Accurate and precise measurement of blood flow using ultrasound time domain correlation. In *IEEE Ultrasonics Symposium*, pages 881–886 vol.2. IEEE, Oct 1989.

[44] H. Honma, M. Ohta, and T. Nishitani. A simplified method of motion vector estimation for VLSI implementation. In *ACM/SIGDA ninth international symposium on Field programmable gate arrays*, pages 216–219. IEEE International Conference on Systems Engineering, September 1992.

[45] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Acoustics, Speech, and Signal Processing*, 27:13–18, Feb. 1979.

[46] M. Insana, P. Chaturvedi, T. Hall, and M. Bilgen. 3D companding using 1.5D arrays for improved strain imaging. In *IEEE Ultrasonics Symposium*, page 14271430, Toronto, Canada, October 1997. IEEE.

[47] J.A. Jensen and I.R. Lacasa. Estimation of blood velocity vectors using transverse ultrasound beam focusing and cross-correlation. In *IEEE Ultrasonics Symposium*, pages 1493–1497 vol.2. IEEE, Oct 1999.

[48] F. Kallel, M. Bertrand, and J. Ophir. Fundamental limitations on the contrast-transfer efficiency in elastography: an analytic study. *Ultrasound in Medicine and Biology*, 22:463–470, 1996.

[49] F. Kallel and J. Ophir. A least-squares strain estimator for elastography. *Ultrasonic Imaging*, 19:195–208, 1997.

[50] F. Kallel, J. Ophir, K. Magee, and T. Krouskop. Elastographic imaging of low-contrast elastic modulus distributions in tissue. *Ultrasound in Medicine and Biology*, 24:409–425, 1998.

[51] F. Kallel, T. Varghese, J. Ophir, and M. Bilgen. The nonstationary strain filter in elastography, Part II. Lateral and elevational decorrelation. *Ultrasound in Medicine and Biology*, 23:1357–69, 1997.

[52] Ophir J. Kallel F. Three-dimensional tissue motion and its effect on image noise in elastography. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 44:12861296, 1997.

[53] S. Khawam, T. Arslan, and F. Westali. Embedded reconfigurable array targeting motion estimation applications. In *International Symposium on Circuits and Systems*, pages II–760–II–763 vol.2. IEEE, May 2003.

[54] E. Konofagou, J. D'hooge, and J. Ophir. Myocardial elastography - A feasibility study in vivo. *Ultrasound in Medicine and Biology*, 28:475–482, October 2002.

[55] E. Konofagou, F. Kallel, and J. Ophir. Three-dimensional motion estimation in elastography. *1998 IEEE Ultrasonics Symposium. Proceedings*, pages 1745–8 vol.2, October 1998.

[56] E. Konofagou and J. Ophir. A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and Poisson's ratios in tissues. *Ultrasound in Medicine and Biology*, 24:1183–99, October 1998.

[57] E. Konofagou, J. Ophir, F. Kallel, and T. Varghese. Elastographic dynamic range expansion using variable applied strains. *Ultrasonic Imaging*, 19:145–166, 1997.

[58] E.E. Konofagou and J. Ophir. Precision Estimation and Imaging of Normal and Shear Components of the 3D Strain Tensor in elastography. *Physics in Medicine and Biology*, 45:1553–1563, 2000.

[59] E.E. Konofagou, T. Varghese, and J. Ophir. Spectral Estimators in Elastography. *Ultrasonics*, 38:412–416, 2000.

[60] E.E. Konofagou, T. Varghese, J. Ophir, and S.K. Alam. Power spectral strain estimators in elastography. *Ultrasound in Medicine and Biology*, 25:1115–29, September 1999.

[61] F. Kremkau, editor. *Diagnostic Ultrasound: Principles and Instruments, 4th Edition*, chapter 1. W.B. Saunders Company, 1993.

[62] TA Krouskop, S Vinson, B Goode, and D. Dougherty. A pulsed Doppler ultrasonic system for making noninvasive measurements of the mechanical properties of soft tissue. *Journal of Rehabilitation Research and Development*, 24:1–8, 1987.

[63] M. Krueger, A. Pesavento, H. Ermert, K.M. Hiltawsky, L. Heuser, H. Rosenthal, and A. Jensen. Ultrasonic strain imaging of the female breast using phase root seeking and three-dimensional optical flow. In *IEEE Ultrasonics Symposium*, pages 1757–1760 vol.2. IEEE, October 1998.

[64] R.M. Lerner, S.R. Huang, and K.J. Parker. 'Sonoelasticity' images derived from ultrasound signals in mechanically vibrated tissues. *Ultrasound in Medicine and Biology*, 16(3):231–239, 1990.

[65] RM Lerner and KJ. Parker. Sono-elasticity in ultrasonic tissue characterization and echographic imaging. In *Proceedings of the 7th European Comm Workshop*, Nijmegen, The Netherlands: Office of Publications of the European Community, 1987.

[66] W Li, F Mastik, I Cespedes, S Carlier, and A Steen. Echo decorrelation estimated from signal powers Ultrasound. *Ultrasound in Medicine and Biology*, 254:405409, 1999.

[67] L. Ljung. *System Identification, Theory For The User.* Prentice Hall PTR, NJ, 1999.

[68] A. Lorenz, A. Pesavento, M. Pesavento, and H. Ermert. Three-dimensional strain imaging and related strain artifacts using an ultrasonic 3D abdominal probe. In *IEEE Ultrasonics Symposium*, pages 1657–1660 vol.2. IEEE, October 1999.

[69] A. Lorenz, HJ Sommerfeld, MG Schurmann, and et al. A new system for the acquisition of ultrasonic multicompression strain images of the human prostate in vivo. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 46:969–1147–1154, 1999.

[70] AM Lubinski, SY Emelianov, KR Raghavan, AE Yagle, AR Skovoroda, and M. ODonnell. Lateral displacement estimation using tissue incompressibility. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 43:247255, 1996.

[71] GE Mailloux, F Langlois, PY Simard, and M. Bertrand. Restoration of the velocity field of the heart from two-dimensional echocardiograms. *IEEE Transactions on Medical Imaging*, 8(2):143153, June 1989.

[72] T. Matsumura, S. Tamano, R. Shinomura, T. Mitake, M. Yamakawa, T. Shiina, A. Itoh, and E. Ueno. Preliminary Evaluation of Breast Disease Diagnosis Based on Real-Time Elasticity Imaging. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 58, 2004.

[73] W.N. McDicken, editor. *Diagnostic Ultrasonics: Principles and use of Instruments, 3rd Edition*, chapter 1. Churchill Livingstone, 1991.

[74] N. Nitta and T. Shiina. Myocardial strain imaging based on three dimensional motion tracking. In *IEEE Ultrasonics Symposium*, pages 1258–1261 Vol.2. IEEE, October 2003.

[75] M. O'Donnel, A.R. Scovoroda, B.M. Shapo, and S.Y. Emelianov. Internal Displacement and Strain Imaging Using Ultrasonic Speckle Tracking. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 41:314–25, May 1994.

[76] M. O'Donnell, S.Y. Emelianov, A.R. Skovoroda, M.A. Lubinski, W.F. Weitzel, and R.C Wiggins. Quantitative elasticity imaging. In *IEEE Ultrasonics Symposium Proceedings*, pages 893–903 vol.2. IEEE, 1993.

[77] M. O'Donnell, A.R. Skovorada, and B.M. Shapo. Measurement of arterial wall motion using Fourier based speckle tracking algorithms. In *IEEE Ultrasonics Symposium Proceedings*, pages 1101–1104 vol.2. IEEE, 1991.

[78] J. Ophir, S.K. Alam, B. Garra, F. Kallel, E. Konofagou, T. Krouskop, and T. Varghese. Elastography: ultrasonic estimation and imaging of the elastic properties of tissues. *Ultrasonic Imaging*, 213:203–33, 1999.

[79] J. Ophir, S.K. Alam, B.S. Garra, F. Kallel, E.E. Konofagou, T. Krouskop, C.R.B. Merritt, R. Righetti, R. Souchon, S. Srinivasan, and T. Varghese. Elastography: Imaging the Elastic Properties of Soft Tissues with Ultrasound. *Medical Ultrasound*, 29:155–171, 2002.

[80] J. Ophir, I. Cespedes, B. Garra, H. Ponnekanti, Y. Huang, and N. Maklad. Elastography: ultrasonic imaging of tissue strain and elastic modulus in vivo. *European Journal of Ultrasound*, 3:49, January 1996.

[81] J. Ophir, I. Cespedes, H. Ponnekanti, Y. Yazdi, and X. Li. Elastography: a quantitative method for imaging the elasticity of biological tissues. *Ultrasonic Imaging*, 13:111–134, April 1991.

[82] J. Ophir, B. Garra, F. Kallel, E.E. Konofagou, T. Krouskop, R. Righetti, and T. Varghese. Elastographic Imaging. *Ultrasound In Medicine and Biology, The Future of Biomedical Ultrasound*, 26:S23–S29, 2000.

[83] J. Park, S. Kwon, and MK. Jeong. A Study of Vibration Characteristics for Ultrasonic Elasticity Imaging. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 106, 2004.

[84] K.J. Parker, S.R. Huang, R.A. Musulin, and R.M. Lerner. Tissue response to mechanical vibrations for 'sonoelasticity imaging'. *Ultrasound in Medicine and Biology*, 16(3):241–246, 1990.

[85] F. Pascal. A parallel stereo algorithm that produces dense depth maps and preserves image features. In *Machine Vision and Applications*, pages 35–49 vol.6, 1993.

[86] A. Pesavento and A. Lorenz. Real Time Strain Imaging a new Ultrasonic Method for Cancer Detection: First Study Results. In *IEEE Ultrasonics Symposium*, pages 1647–1652. IEEE, 2001.

[87] A. Pesavento, A. Lorenz, H. Ermert, H. Sommerfeld, M. Garcia-Schurmann, Th. Senge, and S. Philippou. Frame-to-frame statistics of real-time strain images. In *IEEE Ultrasonics Symposium*, pages 1809–1812 vol.2. IEEE, October 2000.

[88] A. Pesavento, A. Lorenz, Siebers S., and H. Ermert. Frame To Frame Filtering Of Real Time Strain Images. In *IEEE Ultrasonics Symposium*. IEEE, 2002.

[89] A. Pesavento, A. Lorenz, S. Siebers, and H. Ermert. New real-time strain imaging concepts using diagnostic ultrasound. *Physics in Medicine and Biology*, 45:1423–1435, 2000.

[90] A. Pesavento, C. Perrey, M. Krueger, and H. Ermert. A time efficient and accurate strain estimation concept for ultrasonic elastography using iterative phase zero estimation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 46:1057–1067, 1999.

[91] H Ponnekanti, J Ophir, Y Huang, and I Cespedes. Fundamental mechanical limitations on the visualization of elasticity contrast in elastography. *Ultrasound in Medicine and Biology*, 21:533–543, 1995.

[92] S. Ramachandran and S. Srinivasan. FPGA implementation of a novel, fast motion estimation algorithm for real-time video compression. In *ACM/SIGDA ninth international symposium on Field programmable gate arrays*, pages 213–219. International Symposium on Field Programmable Gate Arrays, 2001.

[93] BS Ramamurthy and GE. Trahey. Potential and limitations of angle-independent flow detection algorithms using radio-frequency and detected echo signals. *Ultrasonic Imaging*, 13:252268, 1991.

[94] J. Revell, M. Mirmehdi, and D. McNally. Technical report on Ultrasound Speckle Tracking For Strain Estimation, December 2003.

[95] R Righetti, F Kallel, RJ Stafford, and et al. Elastographic characterization of HIFU-induced lesions in canine livers. *Ultrasound in Medicine and Biology*, 25:1099–1113, 1999.

[96] R. Righetti, J. Ophir, and P. Ktonas. Axial Resolution in Elastography. *Ultrasound in Medicine and Biology*, 28:101–113, January 2002.

[97] R. Righetti, J. Ophir, S. Srinivasan, and T. Krouskop. The Feasibility of Using Elastography for Imaging the Poissons Ratio in Porous Media. *Ultrasound in Medicine and Biology*, 30:215–228, 2004.

[98] R. Righetti, J. Ophir, and Krouskop T. Poisson's Ratio Imaging and Poroelastography in Biological Tissues: A Feasibility Study. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 94, 2004.

[99] R. Righetti, S. Srinivasan, and J. Ophir. Lateral Resolution in Elastography. *Ultrasound in Medicine and Biology*, 29:695–704, January 2003.

[100] JM Rubin, SR Aglyamov, T. Wakefield, M. O'Donnell, and SY Emelianov. Clinical Application of Sonographic Elasticity Imaging for Aging of Deep Venous Thrombosis: Preliminary Findings. *Ultrasound Medicine*, 22:443–448, 2003.

[101] A. Ryszko and K. Wiatr. An assessment of FPGA suitability for implementation of real-time motion estimation. In *Euromicro Symposium on Digital Systems and Design*, pages 364–367. IEEE, September 2001.

[102] T. Shiina, M. Doyley, and J. Bamer. Strain imaging using combined RF and envelope autocorrelation processing. In *IEEE Ultrasonics Symposium*, pages 1331–1336. IEEE, 1996.

[103] T. Shiina, N. Nitta, E. Ueno, and J.C. Bamber. Real Time Tissue Elasticity Imaging using Combined Autocorrelation Method. *Medical Ultrasonics*, 26:57–66, 1999.

[104] T. Shiina, M. Yamakawa, N. Nitta, E. Ueno, T. Matsumura, S. Tamano, and T. Mitake. Clinical assessment of real-time, freehand elasticity imaging system based on the combined autocorrelation method. In *IEEE Symposium on Ultrasonics*, pages 664–667 Vol.1. IEEE, October 2003.

[105] T. Shiina, M. Yamakawa, N. Nitta, E. Ueno, T. Matsumura, S. Tamano, R. Shinomura, and T. Mitake. Breast and Prostate Cancer Diagnosis by Real-Time Tissue Elasticity Imaging System - A Feasibility Study. In *2nd International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 60, 2003.

[106] S. Srinivasan, F. Kallel, and J. Ophir. Estimating the elastographic signal-to-noise ratio using correlation coefficients. *Ultrasound in Medicine and Biology*, 28:359–368, March 2002.

[107] S. Srinivasan, F. Kallel, R. Souchon, and J. Ophir. Analysis of an Adaptive Strain Estimation Technique in Elastography. *Ultrasonic Imaging*, 24:109–118, April 2002.

[108] S. Srinivasan, T. Krouskop, and J. Ophir. Comparing Elastographic Strain Images with Modulus Images Obtained Using Nano-Indentation: Preliminary Results Using Phantoms and Tissue Samples. *Ultrasound in Medicine and Biology*, 30:329–3434, 2004.

[109] S. Srinivasan and J. Ophir. A zero-crossing strain estimator in elastography. *Ultrasound in Medicine and Biology*, 29:227–238, 2003.

[110] S. Srinivasan, J. Ophir, and S.K. Alam. Elastographic Imaging Using Staggered Strain Estimates. *Ultrasonic Imaging*, 25:229–245, May 2002.

[111] S. Srinivasan, R. Righetti, and J. Ophir. Tradeoffs between the axial resolution and the signal-to-noise ratio in elastography. *Ultrasound in Medicine and Biology*, 29:847–866, 2003.

[112] C. Sumi. Fine Elasticity Imaging Utilizing the Iterative RF-echo Phase Matching Method. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 46:158166, January 1999.

[113] C. Sumi. Direct Strain Measurement Based on an Autocorrelation Method. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 64, 2004.

[114] C. Sumi. Thermal Propeties Reconstruction Technique Based on Temperature Measurement. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 73, 2004.

[115] G.E. Trahey, J.W. Allison, and O.T. Von Ramm. Angle independent ultrasonic detection of blood flow. *IEEE Transactions on Biomedical Engineering*, BME-34:965–7, December 1987.

[116] M Tristam, DC Barbosa, DO Cosgrove, DK Nassiri, JC Bamber, and Hill CR. Ultrasonic study of in vivo kinetic characteristics of human tissues. *Ultrasound in Medicine and Biology*, 12:927–937, December 1986.

[117] E. Turgay. Imaging Visco-Elastic Properties of Soft Tissue with Ultrasound. Master's thesis, University of British Columbia, 2004.

[118] T. Varghese, , and J. Ophir. Performance optimization in elastography: Multicompression with temporal stretching. *Ultrasonic Imaging*, 18:193–214, 1996.

[119] T Varghese, M Bilgen, and J. Ophir. Multiresolution imaging in elastography. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 45:6575, 1998.

[120] T. Varghese and Ophir J. Characterization of elastographic noise using the envelope of echo signals. *Ultrasound in Medicine and Biology*, 24:543–555, 1998.

[121] T. Varghese, E.E. Konofagou, J. Ophir, S.K. Alam, and M. Bilgen. Direct strain estimation in elastography using spectral cross-correlation. *Ultrasound in Medicine and Biology*, 26:1525–1537, November 2000.

[122] T. Varghese and J. Ophir. Estimating tissue strain from signal decorrelation using the correlation coefficient. *Ultrasound in Medicine and Biology*, 22:1249–1254, 1996.

[123] T. Varghese and J. Ophir. A theoretical framework for performance characterization of elastography: the strain filter. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 44:164–72, January 1997.

[124] T. Varghese and J. Ophir. Enhancement of echo-signal correlation in elastography using temporal stretching. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 44:173–180, January 1997.

[125] T. Varghese and J. Ophir. The nonstationary strain filter in elastography, Part I Frequency dependent attenuation. *Ultrasound in Medicine and Biology*, 23:1343–56, 1997.

[126] T. Varghese and J. Ophir. An analysis of elastographic contrast-to-noise ratio performance. *Ultrasound in Medicine and Biology*, 24:915–924, 1998.

[127] T. Varghese, J. Ophir, E. Konofagou, F. Kallel, and R. Righetti. Tradeoffs In Elastographic Imaging, Ultrasonic Imaging. *Ultrasonic Imaging*, 23:216–248, October 2001.

[128] T. Varghese, J. Zagzebski, and J. Jee. Elastography imaging of thermal lesion in the liver in vivo following radio frequency ablation: Preliminary results. *Ultrasound in Medicine and Biology*, 28:1467–1473, 2002.

[129] F. Viola and W.F. Walker. Comparison of time delay estimators in medical ultrasound. In *IEEE Ultrasonics Symposium*, pages 1485–1488 vol.2. IEEE, October 2001.

[130] F. Walker and E. Trahey. A fundamental limit on delay estimation using partially correlated speckle signals. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 42:301–308, 1995.

[131] T.E. Weber and R.L. Hollis. A vision based correlator to actively damp vibrations of a coarse-fine manipulator. In *IEEE International Conference on Robotics and Automation Proceedings*, pages 818–825 vol.2. IEEE, 1989.

[132] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15:70–73, Junuary 1967.

[133] Y. Yamakoshi, J. Sato, and T. Sato. Ultrasonic imaging of internal vibration of soft tissue under forced vibration. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 37(2):45–53, March 1990.

[134] M. Yassine, F. Ossant, C. Imberdis, and F. Patat. New Elastographic Algorithm for in vivo Study of the Mechanical Behavior of Skin. In *Proceedings of the Third International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, page 59, 2004.

[135] F Yeung, SF Levinson, and KJ. Parker. Multilevel and motion model-based ultrasonic speckle tracking algorithm. *Ultrasound in Medicine and Biology*, 24:427441, 1998.

[136] Ko Young-Ki, K Hyun-Gyu, L. Jong, K Young-Ro, Oh Hyeong-Cheol, and Ko; Sung-Jea. New motion estimation algorithm based on bit-plane matching and its VLSI implementation. In *IEEE Region 10 Conference*, pages 848–851 vol.2. IEEE, September 1999.

Appendix A

Real Time Motion Tracking Algorithms

A.1 Phase Root Seeking (PRS)

The PRS algorithm is illustrated in Fig. A.1. It uses a modified Newton iteration method at each window to estimate the time-shift from the phase-shift of the signals:

$$\begin{aligned} t_{n+1} &= t_n - \frac{\varphi(t_n)}{\dot{\varphi}(t_n)} \\ &\approx t_n - \frac{\varphi(t_n)}{\omega_0}. \end{aligned} \tag{A-1}$$

Similarly to the TDE method, time shifts are estimated by the PRS method using a discrete number of windows at each RF line [90]. The time shift τ_k of the k-th window of two A-lines centered around $t_k = k\Delta T$ is estimated by the following iterative formula

$$\overbrace{\tau_{k,0}}^{FirstGuess} = \overbrace{\tau_{k-1,L}}^{TimeShiftOfPreviousWindow} \tag{A-2}$$

$$\overbrace{\tau_{k,l}}^{CurrentGuess} = \overbrace{\tau_{k,l-1}}^{PreviousGuess} + \frac{1}{\underbrace{\omega_0}_{SlopeOfPhaseShift}} \overbrace{arg\left[e^{-j\omega_0\tau_{k,l-1}} \int_{t_k-T_w/2}^{t_k+T_w/2} a_b^*(t)\, b_b(t-\tau_{k,l-1})\right]}^{EstimatedPhaseShift},$$

where $a_b(t)$ and $b_b(t)$ denote the corresponding pre-compression and post-compression base-band echo data. l is the iteration index and L is the number of iterations that is mostly between 2 to 6. ω_0 denotes the transducers nominal center frequency, T_w denotes the window length and the base-band signals are calculated from analytic signals by [90]:

$$a_b(t) = a_+(t)\, e^{-j\omega_0 t} = [a(t) + jHilbert(a(t))]\, e^{-j\omega_0 t}, \tag{A-3}$$

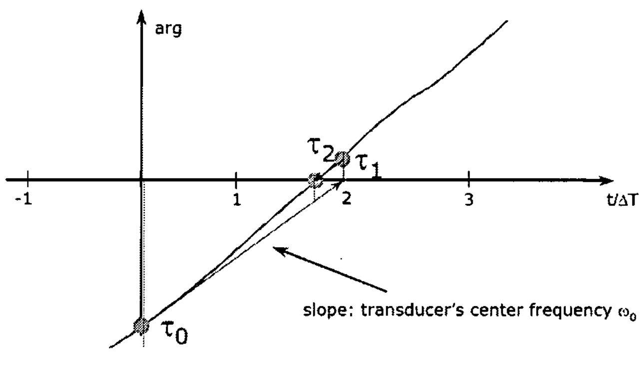


Figure A.1: A Modified Newton search far the estimation of the time delay. The root of the phase is iteratively searched. In each step of the iteration, the phase is calculated and approximated by a linear function with a slope equal to the transducers centroid frequency. The intercept of this linear function with the abscissa is the new estimate for the time delay [90].

Furthermore the post-compression signal has to be re-sampled by a sub-sample time shift [90]. Among all available re-sampling methods, *linear interpolation* is used for the purpose of real time applications since it is more accurate for base-band data [89].

A.2 Combined Autocorrelation Method (CAM)

The method has the merits of phase domain processing but without aliasing, since it combines the result of phase correlation with that of envelope correlation, both of which are calculated directly from the RF signal using complex autocorrelation processing [102, 103]. Phase correlation is used for displacement measurement and envelope correlation is used for phase unwrapping.

In order to perform the complex autocorrelation, phase information is needed. Similarly to the PRS method, the Hilbert transform can be used to convert the time-domain signals to analytic signals. After this step the complex cross-correlation function is used to detect the phase shift:

$$R_{ab}\left(t;n\right)=\int_{-t_0/2}^{t_0/2} a_{+}\left(t+v\right)b_{+}^{*}\left(t+nT/2+v\right)dv=R_u\left(t;\tau-nT/2\right)e^{-j\omega_0(\tau-nT/2)}, \qquad \text{(A-4)}$$

where T is period of ultrasonic wave, ω_0 is carrier angular frequency, τ is the time shift and $R_u(t;\tau)$ is the autocorrelation of the envelope.

For the special case of $n=0$, the displacement d can be obtained for the phase shift $\varphi=\omega_0\tau$. If the displacement is less than $\lambda/4$ it can be obtained directly and without ambiguity:

$$\delta=\frac{\varphi}{2\pi}\lambda=\frac{\omega_0\tau}{2\pi}\lambda, \qquad \text{(A-5)}$$

Since large displacements may be necessary for elasticity imaging, CAM uses another phase unwrapping step by using the envelope normalized correlation coefficient defined by

$$C_u\left(t;n\right)=\frac{\left|R_{ab}\left(t;n\right)\right|}{\left|a\left(t\right)\right|\left|b\left(t+nT/2\right)\right|}, \qquad \text{(A-6)}$$

According to Equation (A-4) and Equation (A-6), for each time t, two sets of C_u and $\varphi=\omega_0\tau$ may be obtained as

$$C_u\left(t\right)=\left\{C_u^{-M},\ldots,C_u^{-1},C_u^{0},C_u^{1},\ldots C_u^{N},\right\} \qquad \text{(A-7)}$$

$$\varphi\left(t\right)=\left\{\varphi^{-M},\ldots,\varphi^{-1},\varphi^{0},\varphi^{1},\ldots\varphi^{N},\right\}, \qquad \text{(A-8)}$$

where $C_u^n = C_u^n(t;n)$ and φ^n is the phase of $R_{ab}(t;n)$. If M and N (size of the search region) are selected to be sufficiently large, one component of $\{\varphi(t)\}$, φ^k, among the n components is not wrapped because it is obtained from two sequences in which the displacement at t is less than $\lambda/4$. At the same time C_u^n becomes maximum at $n = k$ (the smaller the displacement the higher the correlation coefficient).

Therefore the first step is to determine the value of n (among $N+M+1$ values) which maximizes the C_u^n , after which the unwrapped phase shift may be obtained as $\varphi(t) = arctan(R_{ab}(t;n))$ and the displacement can be calculated from Equation (A-5).

In other words, the CAM algorithm estimates the phase-shift at a number of locations with a $\lambda/2$ spacing between them. The phase shift that corresponds to the maximum envelope correlation at that position is then reported as the unwrapped phase shift and is used to estimate the displacement.

Appendix B

Correlation Based Motion Tracking Methods

B.1 Correlation Estimation Methods

Motion estimation is performed on a pair of signals which can be mathematically expressed as Equation (B-2)

$$\begin{aligned} a(t) &= r_1(i) + n_1(i) \\ b(t) &= r_2(i) + n_2(i), \end{aligned} \tag{B-1}$$

where $a(t)$ and $b(t)$ are pre-compression and post-compression digitized RF data. In this expression, $r_1(i)$ and $r_2(i)$ are the echo signals received by the transducer, which may have been decorrelated by physical processes, whereas $n_1(i)$ and $n_2(i)$ represent additive noise introduced from electronic sources. For a small window inside each RF data the stationary property of the process is valid. Under these conditions, a correlation estimation algorithm can be used to estimate the time-shift inside small windows in $a(t)$ and $b(t)$. Common pattern matching methods can be categorized in below.

- **Correlation:** The correlation (Corr) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$Corr(\tau) = \sum_{t=i}^{t=i+W} a(t) \times b(t+\tau), \tag{B-2}$$

where i is the first element of the current window and W is the length of the window. This method is widely used in signal processing applications [35, 129].

- **Normalized Correlation:**

The normalized correlation (NCorr) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$NCorr(\tau) = \frac{\sum_{t=i}^{t=i+W} a(t) \times b(t+\tau)}{\sqrt{\sum_{t=i}^{t=i+W} a^2(t) \sum_{t=i}^{t=i+W} b^2(t+\tau)}}, \tag{B-3}$$

The normalized Correlation scales the correlation coefficients between 0 and 1 and makes it possible to judge how accurate the pattern matching is accomplished, according to the coefficinet value (closer to 1 means more accurate) [127].

- **Normalized Covariance:**

The normalized covariance (NCov) between the signal $a(t)$ and signal $b(t)$ is defined as :

$$NCov(\tau) = \frac{\sum_{t=i}^{t=i+W} (a(t) - \overline{a}) \times \left(b(t+\tau) - \overline{b}\right)}{\sqrt{\sum_{t=i}^{t=i+W} (a(t) - \overline{a})^2 \sum_{t=i}^{t=i+W} \left(b(t+\tau) - \overline{b}\right)^2}}, \tag{B-4}$$

where $\overline{a}$ and $\overline{b}$ are the means of the current pre- and post-compression windows. The normalized covariance is mostly used in image processing applications [85] but it has also been used for ultrasound signals as well [37, 129].

- **Hybrid-Log Method:**

The hybrid-log method (HLM) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$HLM(\tau) = \sum_{t=i}^{t=i+W} \left[(a(t) - b(t+\tau)) - ln(1 + exp(2(a(t) - b(t+\tau)))) \right], \tag{B-5}$$

The *Log* compression transforms the multiplicative noise to additive [94].

- **Polarity-Coincidence Correlation:**

The polarity-coincidence Correlation (PCC) between the signal $a(t)$ and signal $b(t)$ was firstly used in [17] for elastography and is defined as:

$$PCC(\tau) = \sum_{t=i}^{t=i+W} sign(a(t)) \times sign(b(t+\tau)), \tag{B-6}$$

- **Hybrid-Sign Correlation:**

The hybrid-sign correlation (HSC) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$HSC(\tau) = \sum_{t=i}^{t=i+W} a(t) \times sign(b(t+\tau)), \tag{B-7}$$

- **Meyr-Spies Method:**

The meyr-spies method correlation between the signal $a(t)$ and signal $b(t)$ is defined as:

$$MSM(\tau) = \sum_{t=i}^{t=i+W} [a(t-2) - a(t)] \times b(t-1+\tau), \tag{B-8}$$

- **Sum of Absolute Differences:**

The sum of absolute differences (SAD) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$SAD(\tau) = \sum_{t=i}^{t=i+W} |a(t) - b(t+\tau)|, \tag{B-9}$$

In contrast to previous methods, to find the pattern with SAD we search for the position that minimizes the SAD while in previous methods we search for the position that maximizes the correlation [19,35,129]. The advantage of SAD as a similarity measure over normalized cross correlation algorithm is its simplicity in implementation. Like correlation, SAD algorithm requires one operation per pixel whereas the normalized cross correlation algorithm requires five and normalized covariance method requires eight operations per pixel. In [12] it was shown that SAD is as accurate as correlation for both RF data and B-mode data in detecting both axial and lateral motion.

- **Sum of Squared Differences:**

The sum of squared differences (SSD) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$SSD\,(\tau) = \sum_{t=i}^{t=i+W} \left(a\,(t) - b\,(t+\tau)\right)^2, \tag{B-10}$$

The difference between SSD and SAD is that it amplifies the large differences larger than SAD. Similar to SAD we search for the position that minimizes the SSD [35, 129].

- **Normalized Sum of Squared Differences:** The normalized sum of squared differences (NSSD) between the signal $a(t)$ and signal $b(t)$ is defined as:

$$NSSD\,(\tau) = \frac{\sum_{t=i}^{t=i+W}\left(\left(a\,(t)-\overline{a}\right)-\left(b\,(t+\tau)-\overline{b}\right)\right)^2}{\sqrt{\sum_{t=i}^{t=i+W}\left(a\,(t)-\overline{a}\right)^2 \sum_{t=i}^{t=i+W}\left(b\,(t+\tau)-\overline{b}\right)^2}}, \tag{B-11}$$

- **Logarithmic Compression Method:**

  Logarithmic compression helps to reduce the decorrelation noise [17] and can be used prior to all previous methods to improve the performance of pattern matching. It compresses both pre- and post-compression signals according to Equation (B-12). The compressed signals are then used for pattern matching. The logarithmic compression of signal $a(t)$ is defined as:

$$a_{Log}\,(t) = sign\,(a\,(t)) \times Log_{10}\,(1 + |a\,(t)|)\,, \tag{B-12}$$

Among all available methods it was shown that *correlation* and *normalized correlation* are among the most accurate methods for strain estimation [35, 85, 129] and because of this property they have been typically used in strain estimation methods.

B.2 Subpixelling

Since time delays are generally not integral multiples of the sampling period, the location of the largest sample of the cross correlation function is an inexact estimator of the location of the peak. Therefore, interpolation techniques must be used between the samples of the cross correlation to improve the estimation precision. A number of sub-sampling methods have been introduced in literatures that use different curve fitting methods to do the sub-sampling. Curve-fitting interpolation can yield biased time-delay estimates. The artifactual effect of these bias errors on elasticity imaging by elastography is discussed in [16]. In addition to curve-fitting methods, there are a number of *reconstructive interpolation* techniques that can be used to find the exact position of the peak in the cross correlation function [16] but since these methods are computationally very intensive, they are not considered here. The most popular sub-sampling methods that use curve fitting methods can be categorized as follow:

- **Quadratic Parabola fitting:**

  Three values of the correlation coefficients are considered: the highest value and its two neighbors. Quadratic interpolation is applied to the three points, and the location that gives the peak value of the curve is chosen. The peak of the curve is determined by finding the zero crossing of the first derivative [31, 37]. According to Figure B.1, if λ is the distance of the location of the maximum coefficient to the peak point, then λ is given in [31] as:

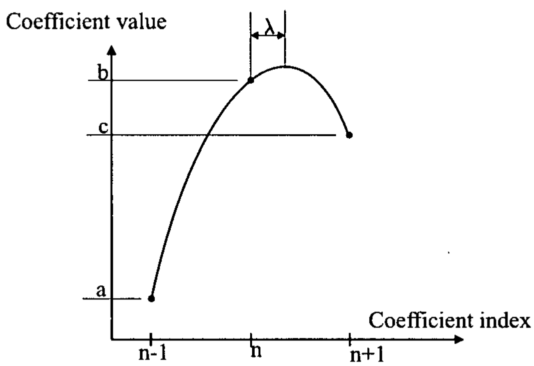


Figure B.1: Exact position of the peak of cross correlation function is typically estimated by using interpolation techniques. The image is provided by Turgay E.

$$\lambda = \frac{c-a}{2(-c+2b-a)}, \tag{B-13}$$

  where b is the value of the maximum correlation coefficient, and a and c are the values of the left and right samples respectively.

- **Cosine fitting:** Another curve that has been used in [16, 27] to interpolate the peak in the cross correlation function is the cosine function. This model is reasonable for some signals with rectangular and Gaussian spectra and it has been used in both blood flow estimation and elastography. Similar to previous method, the peak of a cosinusoid fitted to the three largest samples of the digital cross correlation values (a, b, c) was derived to be:

$$\lambda = -\frac{\theta}{\omega_0}, \tag{B-14}$$

where ω_0 is the angular frequency of the cosinusoid given by:

$$\omega_0 = cos^{-1}\left(\frac{a+c}{2b}\right), \tag{B-15}$$

and θ is the phase of the cosinusoid given by:

$$\theta = tan^{-1}\left(\frac{a-c}{2bsin\,(\omega)}\right), \tag{B-16}$$

These two curve-fitting methods were evaluated and compared in [16]. They showed that cosine interpolation performs better than parabolic interpolation in an arbitrarily small time interval near the peak and the goodness of the cosine-fit deteriorates with increased length of this interval.

B.3 Correlation Recalculation

After finding the exact position of the peak with one of the subpixelling methods the value of the correlation at that point also needs to be recalculated. Although the value of the correlation has no effect on the result of motion tracking but it can be used later to validate the estimated motion. In this thesis three points Lagrangian interpolation method is used to recalculate the value of the correlation. Similar to subpixelling, the same three values of the correlation coefficients are considered: the highest value and its two neighbors (Fig. B.1). Lagrangian interpolation is applied to the three points, and the value of the correlation at the peak position is calculated according to Equation (B-17).

$$Correlation = a \times \lambda \times (\lambda - 1)/2 - b \times (\lambda + 1) \times (\lambda - 1) + c \times (\lambda - 1) \times \lambda/2, \tag{B-17}$$

The accuracy of the recalculation can be improved by increasing the number of points and using higher order interpolation techniques which requires estimation of the correlation at more points.

Appendix C

Computational Issues

C.1 Computational Cost of the LSE Method

If the calculation is run without any simplification and according to the Equation (4.4) from left to right, the complexity of the algorithm for a kernel size of N will be:

1. A^T: $N \times 2$ Matrix Transposing, (cost is ignored).
2. $A^T A$: $N \times 2$ by $N \times 2$ Matrix Multiplication, $[4N, 4(N-1)]$.
3. $[A^T A]^{-1}$: 2×2 Matrix Inversion, (cost is ignored).
4. $[A^T A]^{-1} A^T$: 2×2 by $2 \times N$ Matrix Multiplication, $[4N, 2N]$.
5. $[A^T A]^{-1} A^T \hat{d}$: $2 \times N$ by $N \times 2$ Matrix Multiplication, $[4N, 4(N-1)]$.

which is $12N$ multiplication and $10N$ summations for each window ($[12N, 10N]$) for the original implementation of the LSE.

The implementation that is used in this thesis (Section 7.1.2.1) reduces the complexity of LSE to 2 multiplications and 8 summations (Equation(7.6)) for each window where N is the kernel size. Therefore the overall cost of the second fast version will be [2, 8] for each window which is independent from N.

Considering the number of multiplications and summation, the current implementation of the LSE performs almost $6N$ times faster than standard implementation. This means for the kernel size of 10 the second approach is almost 60 times faster than the original implementation. Using this approach the calculation of the least squares estimation becomes possible for even very large kernel size ($N \geq 50$) in the real time applications while the size of the kernel can also be adjusted in real time.