

---

# **Comparison of three ultrasound velocity estimators for strain imaging of the brain**

---

By  
Thomas Kristoffersen Børstad

Trondheim, Desember 2010



## Abstract

Strain imaging is a technique where the strain experienced in tissue due to compression/decompression is estimated and visualized as an image. Taking advantage of the natural pulsation of the arteries, ultrasound strain imaging of the brain can be performed during craniotomy. Strain values in brain tumors are believed to differ from strain values in normal brain tissue, potentially allowing strain visualization to improve the detection of tumorous tissue using ultrasound.

In this project strain values were calculated by axial differentiation of a velocity field, where the velocity field has been estimated from two consecutive RF-frames. Three different time-domain, phase-based velocity estimators have been compared: the 1D autocorrelator (1D-AC), the 2D autocorrelator (2D-AC) and the crosscorrelator (CC). The estimators were tested on simulated data and on recordings of an elasticity phantom. The variance of the estimators is found to increase for higher velocities, with the CC method breaking down for velocities higher than a system parameter-dependent critical velocity. The 2D-AC method outperforms the 1D-AC and CC methods for high velocities, while for low velocities the accuracy difference between the methods diminish. Effects on the velocity estimates by varying the size of the sample region is explored and is shown to significantly affect the results. Strain images are found to be highly dependent upon the accuracy of the estimated velocity field, and the length of the spacing used when calculating the axial derivative.

The strain images are found to be significantly better for separating between structures of varying stiffness, compared to conventional B-mode images. Presenting the strain image in real-time is found to be problematic; the image is difficult to interpret due to large fluctuations of the strain values. A post-processing technique which uses the average value and standard deviation of an image to normalize and scale it, is used to compensate for this problem. Post-processing is found to improve the image quality on phantom data, but the method is highly sensitive to noise.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Strain and post processing</b>	<b>3</b>
2.1	Strain . . . . .	3
2.2	Calculating a strain image . . . . .	3
2.3	Problems associated with strain imaging . . . . .	4
2.4	Strain magnitude . . . . .	4
2.5	Thresholding and normalization . . . . .	5
<b>3</b>	<b>Velocity estimation</b>	<b>6</b>
3.1	Echo signals from a moving reflector . . . . .	6
3.2	Mathematical notation . . . . .	7
3.3	1D autocorrelator (1D-AC) . . . . .	7
3.4	2D autocorrelator (2D-AC) . . . . .	8
3.5	Crosscorrelation method (CC) . . . . .	8
3.6	Computational complexity . . . . .	9
<b>4</b>	<b>Methods</b>	<b>10</b>
4.1	Algorithmic implementation . . . . .	10
4.2	Simulation parameters . . . . .	10
4.3	Ultrasonix data . . . . .	11
<b>5</b>	<b>Simulated results - velocity estimation</b>	<b>12</b>
5.1	Velocities below $v_c$ . . . . .	12
5.2	Velocities close to $v_c$ . . . . .	13
5.3	Estimator mean and variance over a range of velocities . . . . .	13
<b>6</b>	<b>Phantom result - velocity estimation</b>	<b>15</b>
6.1	Effects of varying the algorithmic parameters . . . . .	15
6.2	Estimator comparison . . . . .	16
<b>7</b>	<b>Phantom results - strain imaging and post processing</b>	<b>18</b>
7.1	Comparison of the different estimators after strain processing . . . . .	18
7.2	Image post processing . . . . .	18
7.3	Result as it may be presented on operator screen . . . . .	19
<b>8</b>	<b>Discussion and observation</b>	<b>21</b>
8.1	Estimators . . . . .	21
8.2	Strain and post processing . . . . .	21
8.3	Future work . . . . .	21
<b>A</b>	<b>1D autocorrelator</b>	<b>23</b>
<b>B</b>	<b>Gaussian correlation coefficient</b>	<b>23</b>

## 1 Introduction

Medical ultrasound (US) images are made by transmitting sound waves, generated by a transducer, into the body and then recording the echoes returned. The real-time 2D image normally presented to the operator of a medical ultrasound scanner, the B-mode image, visualizes the power of the returned echo-signals. Echoes are generated when the propagating sound wave experiences a change in the characteristic acoustic impedance  $Z_0$  of the tissue.  $Z_0$  is dependent on two tissue properties, namely the tissue density  $\rho$  and the speed of sound  $c$  in the tissue. The relationship,  $Z_0 = \rho c$ , may be found in any text book describing waves, for example [1]. The variation of  $\rho$  and/or  $c$  for different kinds of tissue and structures, together with echo-attenuation, are the physical properties responsible for the majority of the information visible in a B-mode image. However other properties such as tissue elasticity and experienced strain, which can be of pathological importance, is generally not correlated with what is seen on the B-mode image [2]. Using different signal processing techniques and physical setups, the extraction of a wide range of properties in the imaged region is possible. This project deals with the calculation of axial strain values using signal processing, to generate strain images that can complement the normal B-mode image.

The motivation for this project is to attempt to improve (compared to B-mode) the ability to separate between a tumor and normal tissue on intraoperative images of the brain. Normally US is not used to image the brain, because the skull effectively blocks sound pulses from reaching the brain. However during brain tumor removal a hole is made in the skull, a procedure known as *craniotomy*, to give the neurosurgeon access to the brain. Covered in a sterile drape, the transducer is placed in the skull opening, it is possible to perform ultrasound imaging of the brain. Using a 3D navigation system to combine intraoperative ultrasound images with preoperative images from other modalities, is a technique that has been shown to have advantages in a clinical setting [3]. My co-supervisor, Tormod Selbekk, has in Selbekk et al. [4, 5] showed that axial strain can be used for brain tumor detection. The RF-data recorded and used in the Selbekk et al. studies [4, 5], which are from previous brain tumor surgeries performed St. Olavs hospital, has been made available for use in this project. The Regional Research Ethics Committee of Central Norway has approved the use of this data for retrospective studies.

This project report starts with describing how axial strain values are calculated using axial differentiation (section 2) of velocity estimates. The mathematical description of pulsed echo-signals from a moving target is then presented (section 3), and three velocity estimators are introduced. The data sets used to test the velocity and strain estimations are then explained (section 4), followed by the results on simulated and phantom data (sections 5 to 7). Finally a summary of the results, and an preliminary image from testing on clinical data, is presented at the end of the report, together with suggestions for future work. An electronic version (PDF) of this report is available for download on the project webpage [6].

## 2 Strain and post processing

Quantifying tissue stiffness is in general a challenging task, as tissue properties are non-linear and dependent on numerous properties such as time and moisture [2]. One possibility is to use a simplified model, and characterize tissue by calculating different moduli, such as Young's modulus or the Poisson ratio. Calculating tissue modulus using ultrasound traditionally involved using a vibrating actuator to generate standing waves in the tissue while doing US imaging [7]. Such methods have been tested on the brain by Scholz et al. [8], but was not used during the surgeries at St. Olavs hospital. This was due to concerns and uncertainty of the potential impact on the patient by using external forces on the brain. Newer techniques can measure tissue elastic properties without external forces, an example is supersonic shear imaging [9]. In this project axial strain values are calculated to detect regions of different stiffness without calculating elasticity constants.

### 2.1 Strain

The axial strain  $\varepsilon$  of an object being axially compressed or stretched is the ratio:

$$\varepsilon = \frac{L - L_0}{L_0} \quad (2.1)$$

where  $L_0$  is the initial axial length of the object and  $L$  is the length after compression/stretching. Notice that the strain  $\varepsilon$  is a unit-less quantity, which can have positive or negative values. Strain is a property of an object, as an example the strain value changes with the length of the object. Elasticity constants however, such as Young's modulus, are inherent to the material and is invariant to the current shape of an object.

It is seen from eq. (2.1) that to calculate strain, the length of the object needs to be known before and after compression. In order to avoid dealing with pre- and post-compression lengths, which would involve tracking of objects, the strain experienced in a fixed region between two consequent RF-frames is calculated. This simplifies computations considerably, allowing axial strain to be calculated by numeric differentiation of a velocity field.

### 2.2 Calculating a strain image

Let the entire ultrasound sample volume be represented as a 2D matrix with entries  $[m, n]$ , with  $m$  representing the depth (increases with  $m$ ) and  $n$  the beam number of the corresponding sample. If there has been axial movement in the tissue between the two RF-frames, there will be a time delay difference between a pulse in the first frame and the matching pulse of the second frame. This time delay will depend on the movement in the tissue. This is explained in more detail in section 3.1.

Let  $\hat{\tau}[m, n]$  be the estimated time delay difference delay at a point  $[m, n]$ , then the axial strain over a region with length  $\Delta m$  can be estimated as:

$$\hat{\varepsilon}[m, n] = \frac{\hat{\tau}[m - \frac{\Delta m}{2}, n] - \hat{\tau}[m + \frac{\Delta m}{2}, n]}{\Delta m T_s} \quad (2.2)$$

where  $\hat{\varepsilon}$  is the strain estimate and  $T_s$  is the echo sampling frequency. The time delays  $\hat{\tau}[m, n]$  are directly related to the velocity estimate  $\hat{v}[m, n]$  between two pulse sets as:

$$\hat{v}[m, n] = \frac{c}{2} \frac{\hat{\tau}[m, n]}{T_{PR}} \quad (2.3)$$

where  $T_{PR}$  is the fixed transmission delay between the pulse sets. By combining equations the axial strain over the region defined by  $\Delta m$  can be calculated using velocity estimates:

$$\varepsilon[m, n] = \alpha(\Delta m) \cdot \left( \hat{v}[m - \frac{\Delta m}{2}, n] - \hat{v}[m + \frac{\Delta m}{2}, n] \right) \quad (2.4)$$

with  $\alpha(\Delta m) = \frac{2}{c\Delta m} \frac{T_{PR}}{T_S}$ . This method is similar to the method used by Heimdal et al [10], strain rate imaging. The difference between the methods is in the scaling constant  $\alpha(\Delta m)$ . A physical explanation of eq. (2.4), is that of the two endpoints of a one-dimensional object are moving with different velocities, then the object must either compress or stretch.

By increasing the parameter  $\Delta m$  the average strain over a longer image region is calculated, and the effects on the image by varying this parameter is shown later. The quality of the strain estimate  $\hat{\varepsilon}$  is directly dependent on how well the velocity  $\hat{v}[u, m]$  can be estimated. Thus three different velocity estimators are studied in this report, together with their effect on the strain estimator.

### 2.3 Problems associated with strain imaging

If the same force is applied on two equally sized, initially non-compressed object such that they are compressed, then the strain values seen will depend on the object stiffness. Intuitively the stiffer object will compress less (low strain value) than the softer object (high strain value). In the strain image, calculated using eq. (2.4), the equivalent is that stiffer regions will have higher strain values than less stiff regions, *if the regions experience equal velocities*. Strain imaging is a qualitative method, where objects can only be compared relatively to other objects if they are imaged under equal conditions (such as velocity). This can make comparing strain values between different patients difficult. Still strain and strain rate imaging has been proven useful in for example cardiac applications [10, 11] and, as previously mentioned, in the brain [4, 5].

If there is no movement in the tissue then, from eq. (2.4), the strain will necessarily be zero. So to perform strain imaging movement in the imaged region is needed. In the brain application, this movement is generated by the natural arterial pulsation. Assuming that the strain values are presented on screen using color-mapping, two related problems occur when viewing the strain image in real-time:

1. It is difficult to find a fixed colormap that works well for both the low and the high velocity periods of arterial pulsation.
2. Due to the periodic compression/decompression, regions with constant stiffness will show a periodic change in color values. This can create a confusing image which is difficult to interpret, due to the time varying dynamic region of the strain values.

These problems are addressed using image post-processing, presented in the next sections.

### 2.4 Strain magnitude

Following the same technique used by Selbekk et all. [4], the absolute strain value, i.e the strain magnitude, is calculated as:

$$|\hat{\varepsilon}[m, n]| = \alpha(\Delta m) \cdot \left| \left( \hat{v}[m - \frac{\Delta m}{2}, n] - \hat{v}[m + \frac{\Delta m}{2}, n] \right) \right| \quad (2.5)$$

By looking at strain magnitude only, the dynamic color range needed to present the values is halved. It is also reasonable to assume that most of the information about the stiffness of a region is contained in the magnitude of the strain, regardless of the sign. An obvious drawback is the loss of knowing whether a region is experiencing contraction or stretching. This makes it more difficult to interpret if the results seen are physically reasonable, and could potentially disguise artifacts. An example is if the strain values indicate that a small region is expanding, while all surrounding regions are compressing. Such a scenario would not be detectable in a magnitude image.

## 2.5 Thresholding and normalization

To deal with problem of the dynamic region of the strain values being time varying, the strain images were normalized and scaled using statistical image processing.

Define the mean value  $\mu$  and standard deviation  $\sigma$  of an image  $x[m, n]$  with  $M_{\text{tot}}$  rows and  $N_{\text{tot}}$  columns as:

$$\mu = \frac{1}{M_{\text{tot}}N_{\text{tot}}} \sum_{m=1}^{M_{\text{tot}}} \sum_{n=1}^{N_{\text{tot}}} x[m, n] \quad (2.6)$$

$$\sigma = \left( \frac{1}{M_{\text{tot}}N_{\text{tot}} - 1} \sum_{m=1}^{M_{\text{tot}}} \sum_{n=1}^{N_{\text{tot}}} (x[m, n] - \mu)^2 \right)^{\frac{1}{2}} \quad (2.7)$$

To scale the strain images, the strain magnitude image was first computed followed by the computation of  $\mu$  and  $\sigma$  of this magnitude image. Then strain magnitude values  $|\hat{\varepsilon}[m, n]|$  were thresholded to the range  $\mu \pm 3\sigma$ . Finally the scaled, shifted absolute strain value  $\hat{\varepsilon}'[u, v]$  is calculated using the relationship:

$$\hat{\varepsilon}'[m, n] = \frac{|\hat{\varepsilon}[u, v]| - (\mu - 3\sigma)}{6\sigma} \quad (2.8)$$

The range of  $\hat{\varepsilon}'[u, v]$  will then be from 0 to 1 inclusive. A normalized value of 0.5 corresponds to the mean absolute strain value  $\mu$ , while 0 and 1 represents strain magnitudes that are exactly on, or under/over the  $3\sigma$  bounds.

This kind of statistical thresholding should be used with care. As will be seen later this method is non-robust, meaning that it breaks down if the signal-to-noise ratio is low. This is illustrated by assuming that the image consists of only noise, then the scaling in eq. (2.8) would scale this noise to cover the whole range of the colormap. This is unacceptable in a clinical setting, and thus this method needs to be refined before it can be potentially used in a clinical system. Despite the shortcomings, the scaling method still produces good results for the phantom data.

### 3 Velocity estimation

Spatial velocity estimates are needed if strain is calculated using eq. (2.4). Velocity estimation using sound echoes is a thoroughly studied topic, and has applications in for example radar, sonar and medical ultrasound. In ultrasound, velocity measurements have been especially important due to the clinical applications of the estimation of blood flow. Thus a wide variety of methods for ultrasonic velocity estimation have been proposed in the literature [12, 13, 14, 15]. To understand the velocity estimators an explanation of the velocity estimation problem follows.

#### 3.1 Echo signals from a moving reflector

A simplified illustration of the principles of pulsed wave velocity estimation is shown in Fig. 1. The figure shows a set of seven returned RF-signals from a single reflector moving away from the transducer. Each pulse was emitted with a delay of  $T_{PR}$  compared to the previous pulse, where the  $PR$  subscript is an abbreviation for Pulse Repetition. The samples, shown with circles, along the vertically dashed

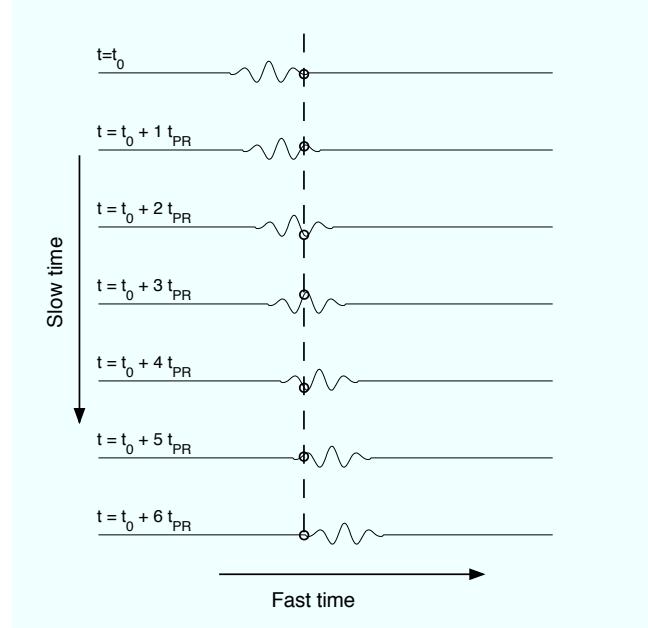


Fig. 1. Ideal received RF-pulses from a moving target. Figure inspired by similar figure in [1, page 659]

lines represents the doppler, or "slow time", signal. It is well known that the frequency of the doppler signal,  $f$ , carries information about the axial velocity  $v_a$  of the moving reflector. This velocity can be calculated using the following approximation, valid for  $v_a \ll c$  [1, page 660]:

$$v_a = |\vec{v}| \cos(\theta) \approx \frac{c}{2} \frac{f}{f_c} \quad (3.1)$$

where  $\vec{v}$  is the velocity vector of the moving reflector,  $\theta$  is the angle between this vector and ultrasonic beam and  $c$  is the speed of sound in the propagation medium. Thus by measuring the doppler frequency  $f$ , and assuming  $c$  and  $f_c$  to be known and constant, the axial velocity  $v_a$  can be estimated. The terms velocity  $v$  and axial velocity  $v_a$  will from now on be used interchangeably.

### 3.2 Mathematical notation

Assume that the signal data available is from a 2D rectangular region with  $\mathbf{M}$  depth samples,  $\mathbf{N}$  adjacent linear scan lines and with  $\mathbf{O}$  "slow time" frames. Each sample represents the RF-signal at a certain point  $[m, n, o]$  in space-time. The mean axial velocity in this 2D region, between frame 0 and frame  $\mathbf{O} - 1$ , is the quantity that is being estimated. Following the notation of Loupas. et al [12],  $\mathbf{M}$  is known as the range gate length and  $\mathbf{O}$  as the ensemble length. Extending on this notation, the term lateral gate length will be used for  $\mathbf{N}$ . Note that this quantity is not used in [12], which is equivalent to fixing  $\mathbf{N} = 1$ .

Let the RF-samples from the set of 2D frames be represented as a real, discrete 3-dimensional signal  $x[m, n, o]$ . Here  $m$  is the depth index,  $n$  the beam number and  $o$  the frame number of a single sample. The depth samples are ordered from near to far and the beams from left to right, using the transducer frame of reference. Further more frames are ordered increasingly according to their sample time, e.g the samples in frame  $o + 1$  are newer than the samples of frame  $o$ . Representing the frames as a set of  $\mathbf{O}$  matrices  $X_0, X_1 \dots X_{\mathbf{O}-1}$  with  $\mathbf{M}$  rows and  $\mathbf{N}$  columns gives:

$$X_0 = \begin{bmatrix} x[0, 0, 0] & x[0, 1, 0] & \dots & x[0, \mathbf{N} - 1, 0] \\ x[1, 0, 0] & x[1, 1, 0] & \dots & x[1, \mathbf{N} - 1, 0] \\ \vdots & \vdots & \ddots & \vdots \\ x[\mathbf{M} - 1, 0, 0] & x[\mathbf{M} - 1, 1, 0] & \dots & x[\mathbf{M} - 1, \mathbf{N} - 1, 0] \\ & & \vdots & \\ & & & \end{bmatrix} \quad (3.2)$$

$$X_{\mathbf{O}-1} = \begin{bmatrix} x[0, 0, \mathbf{O} - 1] & x[0, 1, \mathbf{O} - 1] & \dots & x[0, \mathbf{N} - 1, \mathbf{O} - 1] \\ x[1, 0, \mathbf{O} - 1] & x[1, 1, \mathbf{O} - 1] & \dots & x[1, \mathbf{N} - 1, \mathbf{O} - 1] \\ \vdots & \vdots & \ddots & \vdots \\ x[\mathbf{M} - 1, 0, \mathbf{O} - 1] & x[\mathbf{M} - 1, 1, \mathbf{O} - 1] & \dots & x[\mathbf{M} - 1, \mathbf{N} - 1, \mathbf{O} - 1] \end{bmatrix}$$

The *analytical signal*  $x_+[m, n, o]$  is then defined as:

$$x_+[m, n, o] = x[m, n, o] + i \cdot \mathcal{H}\{x[m, n, o]\} \quad (3.3)$$

where  $\mathcal{H}\{\cdot\}$  represents the 1D Hilbert transform along the first (depth) dimension of the signal. It will prove useful to define the 2D autocorrelation of the signal  $x_+$  as:

$$\begin{aligned} \gamma[m, o] &= \sum_{u=0}^{\mathbf{M}-m-1} \sum_{v=0}^{\mathbf{N}-1} \sum_{w=0}^{\mathbf{O}-o-1} x_+[u, v, w] x_+^*[u+m, v, w+o] \\ &= \sum_{v=0}^{\mathbf{N}-1} \left( \sum_{u=0}^{\mathbf{M}-m-1} \sum_{w=0}^{\mathbf{O}-o-1} x_+[u, v, w] x_+^*[u+m, v, w+o] \right) \end{aligned} \quad (3.4)$$

where the  $*$  denotes complex conjugation. If  $\mathbf{N} = 1$ , then eq. (3.4) is identical to eq. (11) in [12]. Note that the expression for  $\gamma$  is the complex conjugate of the definition normally used in signal processing literature.

### 3.3 1D autocorrelator (1D-AC)

A breakthrough in pulsed doppler velocity estimation occurred in 1985, when Kasai et al. demonstrated real-time pulsed velocity estimation using analog electronics. The velocity estimator used was first presented by Barber et al [13]. The key idea behind this estimator is that the doppler frequency  $f$  can

be estimated by calculating the autocorrelation of the doppler signal. The fact that the estimation can be performed purely in the time domain was an advantage when using analogue techniques, since no costly frequency domain transformations is needed.

The 1D autocorrelator estimates the mean doppler frequency in the  $M \times N$  region over  $O$  frames as (see appendix A):

$$f \approx \frac{1}{2\pi} \frac{\arg\{\gamma[0, 1]\}}{T_{PR}} \quad (3.5)$$

By combining eq. (3.1) and (3.5) the 1D autocorrelator estimate for the axial velocity is obtained as:

$$\hat{v} = \frac{c}{2\omega_c} \frac{\arg\{\gamma[0, 1]\}}{T_{PR}} \quad (3.6)$$

where  $\omega_c = 2\pi f_c$ .

### 3.4 2D autocorrelator (2D-AC)

Loupas et al. further improved the idea of the 1D autocorrelator in their 1995 paper [12]. The idea was to no longer assume that the transmitted signal center frequency  $f_c$  is constant over the imaged region. Instead an estimator which also estimated  $f_c$  was introduced, making better use of the information present in 2D autocorrelation function. In the same article it is argued that this gives a more robust estimator, which will compensates for values of  $f_c$  fluctuating due to, for example, frequency-dependent attenuation.

In addition to using the correlation in the slow time direction  $\gamma[0, 1]$  as in eq. (3.6), the correlation of the signal in the depth direction, i.e  $\gamma[1, 0]$  is used to estimate the pulse center frequency as:

$$f_c \approx \frac{1}{2\pi} \frac{\arg\{\gamma[1, 0]\}}{T_s} \quad (3.7)$$

where the  $T_s$  is the sampling time of the RF-pulses, where the derivation of eq. (3.7) is similar to that given in appendix A. Thus by combining equations (3.6) and (3.7) the 2D-AC estimator is given as:

$$\hat{v} = \frac{c}{2} \frac{T_s}{T_{PR}} \frac{\arg\{\gamma[0, 1]\}}{\arg\{\gamma[1, 0]\}} \quad (3.8)$$

The lateral gate  $N$  parameter, in eq. (3.4), must be 1 for this estimator to be identical to the one introduced in [12].

### 3.5 Crosscorrelation method (CC)

The third and final method is that presented by Simon et al. [16], using a similar idea to another estimator also presented in the Loupas et al. paper [12]. By fixing  $O = 2$ , and thus only using two frames, the time shift between two RF-frames is estimated as:

$$\hat{\tau} = \frac{2\arg\{\gamma[0, 1]\}}{\arg\{\gamma[1, 1]\} - \arg\{\gamma[-1, 1]\}} T_s \quad (3.9)$$

In the derivation of eq. (3.9), without notifying the reader, Simon et al. implicitly assume that the phase of the one-dimensional discrete signal  $\gamma[m, 1]$  is equal to the phase of the analytical crosscorrelation signal  $\tilde{\gamma}[m]$ . With  $\tilde{\gamma}[m]$  defined as:

$$\tilde{\gamma}[m] = \gamma'[m] + i \cdot \mathcal{H}\{\gamma'[m]\} \quad (3.10)$$

where

$$\gamma'[m] = \sum_{u=0}^{M-m-1} \sum_{v=0}^{N-1} x[u, v, 0] x[u + m, v, 1] \quad (3.11)$$

In other words the assumption is that  $\arg\{\gamma[m, 1]\} = \arg\{\tilde{\gamma}[m]\}$ , which is generally not true. However the method is validated through simulations, and gives good results under certain conditions. Using eq. (2.3) and (3.9) gives the following estimator:

$$\hat{v} = c \frac{T_s}{T_{PR}} \frac{\arg\{\gamma[0, 1]\}}{\arg\{\gamma[1, 1]\} - \arg\{\gamma[-1, 1]\}} \quad (3.12)$$

This method appears similar to the 2D-AC method. By inspection the difference is that instead of estimating  $f_c$  using eq. (3.7) the following relationship is used:

$$f_c = \frac{1}{2\pi} \frac{\arg\{\gamma[1, 1]\} - \arg\{\gamma[-1, 1]\}}{2T_s} \quad (3.13)$$

which again implies that:

$$\arg\{\gamma[1, 0]\} = \frac{\arg\{\gamma[1, 1]\} - \arg\{\gamma[-1, 1]\}}{2} \quad (3.14)$$

The assumption is that the phase  $\arg\{\gamma[1, m]\}$  varies linearly for  $m = -1$  to  $m = 1$ . According to Simon et al. this is only a valid assumption if  $|\tau| \leq T_s$ , i.e if the time shift between two frames is less than the echo sampling period. In terms of velocity, the equivalent restriction is  $|v| \leq v_c = c \frac{T_s}{2T_{PR}}$ . From results presented later in this report, it is shown that the CC method breaks down if this assumption is not met.

### 3.6 Computational complexity

It is assumed that the computational complexity of the three estimators is proportional to the number of 2D autocorrelation estimates needed. This means that the 1D-AC method is the fastest one, with the 2D-AC and CC methods being 2 and 3 times as more computationally expensive the 1D-AC method. This is something that has to be considered when dealing with real-time implementation.

## 4 Methods

The 1D-AC and 2D-AC estimators were implemented using MATLAB (The MathWorks Inc., Natick, MA, USA), version 7.10.0.499 (R2010a). The CC estimator implementation was provided by Sintef, and apart from interface changes, no modifications to the Sintef implementation was made. To test and evaluate the methods two different data sets were used. The first set is simulated data, where the true mean velocity being estimated is known in advance. This makes it possible to determine if the estimators are unbiased, and also compare the estimators through their variance. The second data set is phantom recordings done by the company Ultrasonix (Richmond, BC, Canada), made available for users of their SonixRP scanner.

### 4.1 Algorithmic implementation

The three estimators all make use of the 2D autocorrelation function as defined by eq. (3.4), which has three adjustable parameters:

- Range gate **M**. The number of samples used in the axial direction for each velocity estimate. Since axial correlation is used for the 2D-AC and CC methods, this parameter has a lower limit  $M \geq 2$ .
- Lateral gate **N**. The number of samples used in the lateral direction for each velocity estimate. Can be disabled if  $N = 1$ .
- Ensemble length **O**. The number of RF-frames used for each estimate. Must always be greater than 1, since slow time correlation is always used. Note that the CC method is defined with  $O = 2$ .

In order to compare the algorithms the same parameters were used for all estimators, thus it will be assumed that  $O = 2$  from now on. The output of each method is an estimate of the mean axial velocity in the two-dimensional subregion spanned by  $M \times N$  (axial  $\times$  lateral) occurring between two RF-frames. To acquire velocity estimates for each point in the entire image, the  $M \times N$  subregion is translated to cover the entire image. To ensure that all velocity estimates have been calculated using the same parameters (**M** and **N**), estimates where the 2D subregion extends beyond the image (where no data is available) are not used. This will occur at the edges of the image. An example of this effect is seen at the top and bottom of the image in Fig. 10, visualized by the increasingly larger black bars at the top and bottom of the image with increasing **M**.

Exact centering is not possible when **M** or **N** is even, since discrete data is used. The velocity estimate  $\hat{v}[u, v]$  is therefore made using axial samples in the range  $[u - \lceil \frac{M}{2} \rceil - 1, u + \lfloor \frac{M}{2} \rfloor]$ , where  $\lfloor \cdot \rfloor$  is the floor operator, and  $\lceil \cdot \rceil$  is the ceiling operator. Similarly the lateral samples used are  $[v - \lceil \frac{N}{2} \rceil - 1, v + \lfloor \frac{N}{2} \rfloor]$ . In this way the centering is perfect for odd values, and off by a half sample for even values.

Increasing **M** or **N** means that more samples are used for each velocity estimate. It is shown that this gives better estimates (lower noise), but also causes smoothing i.e. degrading the spatial resolution. Thus there is a tradeoff between reducing noise and increasing spatial resolution.

### 4.2 Simulation parameters

The simulated data set is generated using a MATLAB script, provided by professor Hans Torp at the Department of Circulation and Medical Imaging, NTNU. The simulator first approximates the 2D discrete power spectral density (PSD) of an RF pulse set from a region with constant axial velocity. Analytically such a spectrum is described in [1, page 714]. White noise is then shaped using this

spectrum resulting in a stochastic signal with the desired PSD, and the output is the simulated RF ensemble signal. The adjustable parameters of the simulator, together with the values used in this project, are shown in Table 1. For the most part the simulated values were chosen equal to the parameters of the Ultrasonix data set.

Table 1. Simulation parameters

Symbol	Value	Description
$F_s$	20 MHz	RF sampling frequency
PRF	49 Hz	Pulse repetition frequency
$f_c$	5 MHz	Pulse center frequency
BW	2 MHz	Pulse bandwidth (-6 dB)
c	1540 m/s	Speed of sound in tissue
$\theta$	$10^\circ$	Angle of incidence
$f_\#$	2.5	F-number
$M_{\text{tot}}$	948	Number of depth samples
$O_{\text{tot}}$	350	Number of pulse returns

### 4.3 Ultrasonix data

The Ultrasonix (UX) recordings consists of 194 RF-frames, acquired using a 10 MHz linear probe with the SonixRP system. Only the parameters  $F_s$  and PRF were known, and they are equal to the values in Table 1. The estimated pulse power spectrum, was inspected in order to get an estimate for  $f_c$  and BW, with the estimates shown in Table 1. To suppress noise, the RF signal was filtered with a FIR bandpass filter of order 32, centered at the pulse center frequency  $f_c$ . The estimated pulse power spectrum, before and after filtering, together with the filter response is shown in Fig. 2. The number of beams used,  $N_{\text{tot}}$ , is 128. The spacing between beams was not made available.

The UX recordings are of an elasticity phantom, which is being repeatedly compressed and decompressed with a mechanical actuator with a frequency close to  $\frac{1}{3}$ . Embedded in the phantom are three spheres, which are stiffer than the surrounding material. The spheres and the background material have equal echogenicity, meaning that they are not distinguishable on the B-mode image. Thus if the strain processing is successful, the spheres should show up in the strain image.

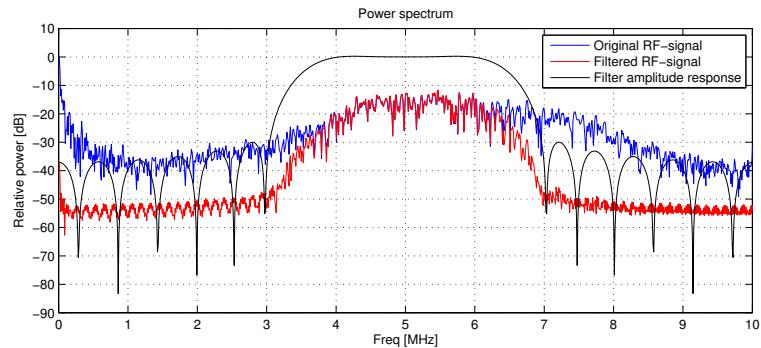


Fig. 2. The estimated power spectrum of the RF-signal in a single frame from the ultrasonix data, before and after bandpass filtration. Also shown is the bandpass filter response

## 5 Simulated results - velocity estimation

The three different velocity estimators were tested on identical RF-data generated using the simulator described in section 4.2. Since only one scan line is generated by the simulator, the lateral gate  $\mathbf{N}$  was set to 1. The range gate was  $\mathbf{M} = 60$ , the reasoning for this choice was based on visual evaluation of the UX data discussed in section 6.1. By varying the simulated mean axial velocity  $v$  the effects of different velocities on the algorithms is studied. The critical axial velocity  $v_c$  leading to wrong results for the CC method, as discussed in section 3.5, is 1.88 mm/s.

### 5.1 Velocities below $v_c$

The first simulation was run with mean velocity  $v = 1$  mm/s. This is well below the critical velocity  $v_c$ , meaning that the assumptions made by the CC method should hold. The estimated velocity along the scan line is shown in Fig. 3 for the different estimators. The 2D-AC and CC methods produce results which are practically inseparable in the image, while the 1D-AC estimates fluctuates noticeably more. The differences between the estimators are quantified in Fig. 4 which shows the estimate distribution from each of the algorithms using histograms. All three methods produce a distribution which resembles a Gaussian, all with a high "gaussian correlation coefficient"  $\rho > 0.99$  (as defined in appendix B). The 1D-AC method has significantly higher standard deviation of  $\sigma = 31.2 \mu\text{m/s}$  than the two other methods, which both have  $\sigma = 10.1 \mu\text{m/s}$ .

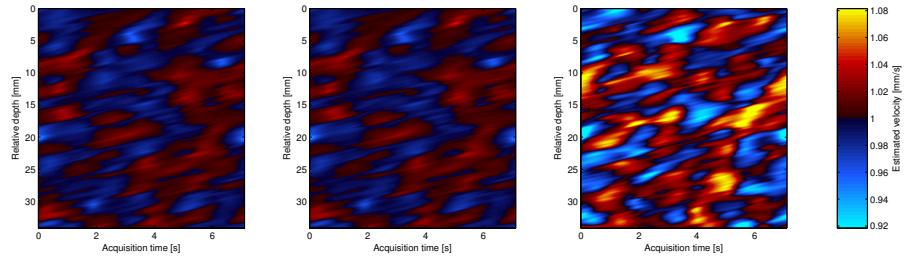


Fig. 3. Velocity estimates from simulated data, with axial velocity equal to 1 mm/s ( $v < v_c$ ). The estimates shown are from the 2D-AC (left), CC (middle) and 1D-AC (right) methods

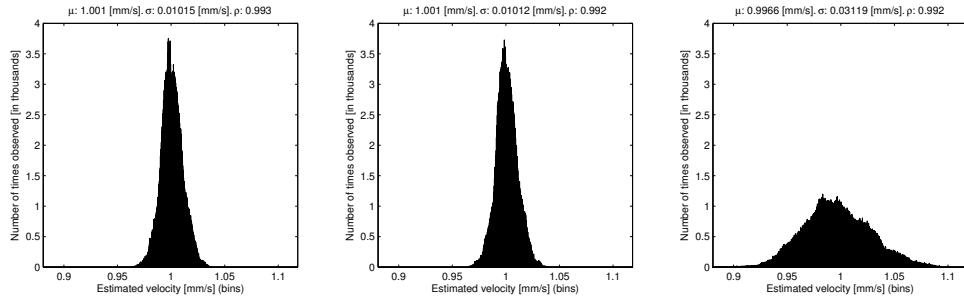


Fig. 4. Histogram showing the estimate distribution for the three methods. The true mean is 1 mm/s. 2D-AC (left), CC (middle) and 1D-AC (right). Shown above each plot is the estimator mean  $\mu$ , standard deviation  $\sigma$  and gaussian correlation coefficient  $\rho$

## 5.2 Velocities close to $v_c$

To study the effect on the CC method by breaking the assumption  $v < v_c$ , the simulations were run again using a mean axial velocity of 1.8 mm/s. This velocity is close to the maximal tolerated velocity  $v_c = 1.88$  mm/s, and the velocity fluctuations in the simulator causes velocities above  $v_c$ . The effect of breaking this assumption is evident from Fig. 5 showing the estimates, where the bright blue spots shown for the CC method correspond to *negative* velocity estimates. The 2D-AC and 1D-AC however, still gives seemingly unbiased estimates. The corresponding estimate distribution histogram in Fig. 6 reveals that the CC estimates are centered around two different values, and no longer resembles a Gaussian ( $\rho < 0.08$ ). The 2D-AC and 1D-AC are still Gaussian ( $\rho > 0.99$ ), and centered around the true mean.

It is worth noting the increase in standard deviation compared to the low velocity scenario. The 2D-AC method this time has a standard deviation of  $\sigma = 15.68 \mu\text{m/s}$  (compared to  $10.1 \mu\text{m/s}$  previously), while the standard deviation for the 1D-AC method has increased to  $\sigma = 57.05 \mu\text{m/s}$  (from  $31.2 \mu\text{m/s}$ ). The standard deviation seems to be increasing with increasing velocity, however it increases more rapidly for the 1D-AC estimator than the 2D-AC estimator.

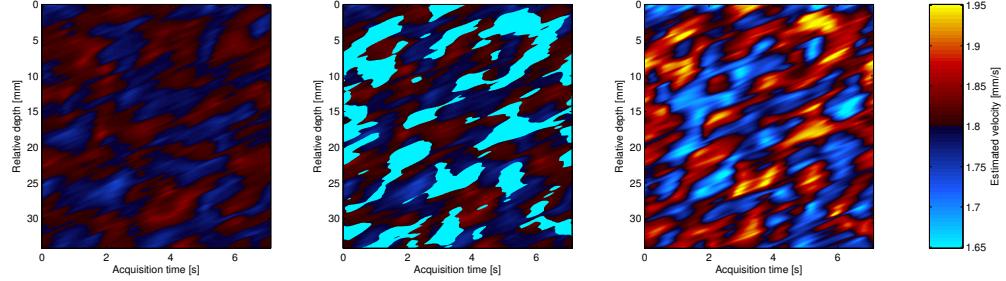


Fig. 5. Velocity estimates from simulated data, with axial velocity equal to 1.8 mm/s ( $v \approx v_c$ ). The estimates shown are from the 2D-AC (left), CC (middle) and 1D-AC (right) methods

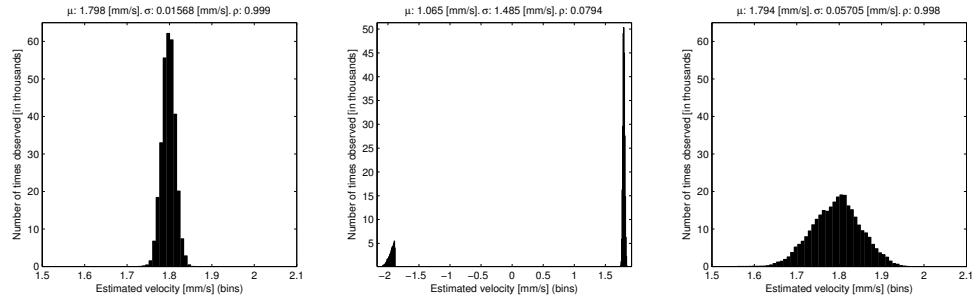


Fig. 6. Histogram plot of high velocity estimates, 2D-AC (left), CC (middle) and 1D-AC (right)

## 5.3 Estimator mean and variance over a range of velocities

To further study the effects on the standard deviation of the estimators with varying velocity, simulations were run over a range of axial velocities. The simulated velocities  $v$  ranged from 0.5 mm/s to 3 mm/s, in

0.05 mm/s intervals. For each of the runs, the average estimate  $\mu$  and corresponding standard deviation  $\sigma$  of the estimators was calculated. Fig. 7 shows the mean error  $\epsilon = v - \mu$  and the corresponding  $3\sigma$  bounds for the different estimators. The differences between the estimators are obscured due to the breakdown of the CC estimator for high velocities.

A zoomed-in version of Fig. 7 is shown in Fig. 8, where it is seen that the CC and 2D-AC methods have practically the same standard deviation as long as the simulated mean axial velocity is below 1.5 mm/s. In the low velocity case all estimators seem to be unbiased, with the mean error fluctuating around zero. The 1D-AC has significantly higher standard deviation than the other methods, and this difference grows bigger with increasing velocity.

For velocities over 1.5 mm/s the CC method breaks down. It is interesting to see from the un-zoomed version in Fig. 7, that the CC error does not converge towards a fixed value for higher velocities. An explanation is found in both [12] and [16].

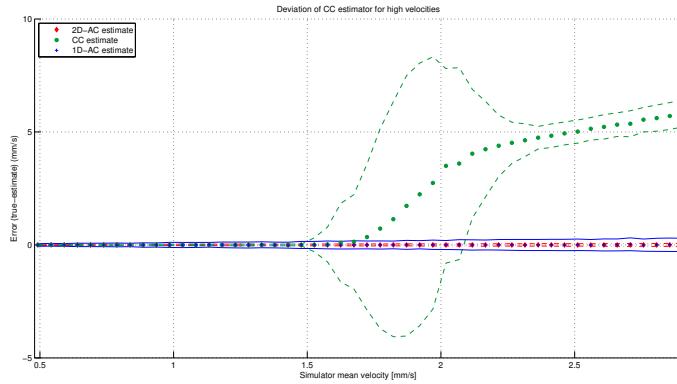


Fig. 7. Mean error (markers) and  $3\sigma$  bounds (lines) for the different estimators. Notice the sudden change in behavior for the CC estimator around the 1.5 mm/s

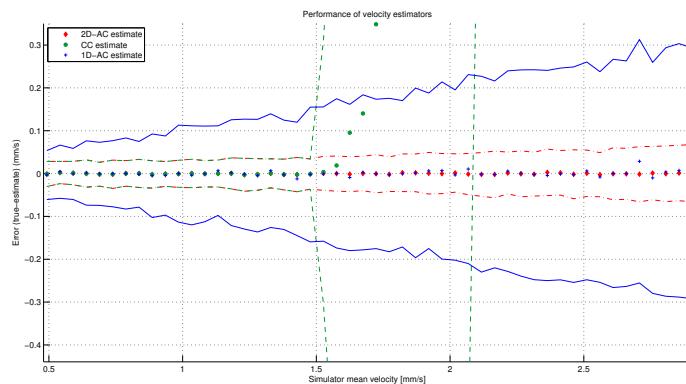


Fig. 8. Mean error (markers) and  $3\sigma$  bounds (lines) for the different estimators. Notice the increase in variance for higher velocities

## 6 Phantom result - velocity estimation

The second data set used for testing is the RF-data from Ultrasonix, described in section 4.3. Some of the figures presented are still frames from a movies, where the results are shown in real-time. Figures which are from a movie has the name of the movie in the figure caption, and the movies are available on the project webpage [6]. This webpage also has an electronic version of this report, presumably with better image quality.

To better visualize the results, figures where the estimators are compared are shown in two scenarios (i.e two frames). The first frame is from velocity estimation between RF-frame 28 and 29, while the other is between RF-frame 98 and 99. The first set corresponds to a point in time when the velocity is low and while the other set is from a time where the velocity is high.

The conventional B-mode image (frame 98), after bandpass filtration and envelope detection, is shown in Fig. 9. The red rectangle marks the user-specified region of interest (ROI), the subset of the data which the algorithms were tested on. Notice that the image inside the ROI looks fairly homogenous on the B-mode image.

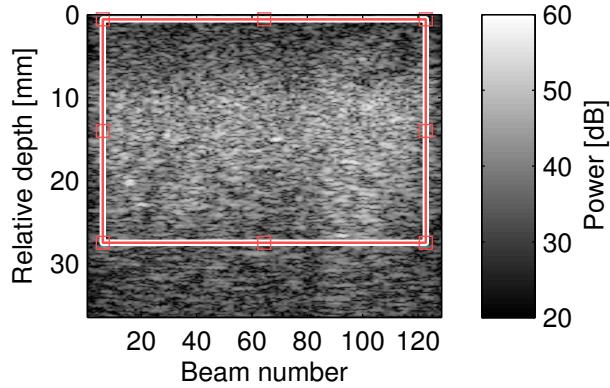


Fig. 9. Ordinary B-mode image of the UX data, with the ROI marked by a red rectangle. **Movie:** bmodeUx.mov [6]

### 6.1 Effects of varying the algorithmic parameters

The range gate  $\mathbf{M}$  and lateral gate  $\mathbf{N}$  are the common tunable parameters of the estimators. To determine appropriate values for these parameters, the effects on the 2D-AC estimator image from changing these parameters was inspected. The values which appeared to give the best image were then chosen.

With  $\mathbf{N} = 1$ , the velocity estimates for a range gate  $\mathbf{M}$  varying from 10 samples (0.385 [mm]) to 100 samples (3.85 [mm]) is displayed in Fig. 10. The black regions at the top and bottom of the image corresponds to points where there are not enough data points to fill the entire range gate, as explained in section 4.1. From Fig. 10 it is seen that increasing the range gate gives smoother results, at the cost of a lowered axial resolution. A range gate of 60 samples (2.31 mm) was chosen as this seemed to give good estimates without significant blurring the results.

In the same manner, the lateral gate  $\mathbf{N}$  was varied from one to five beams, using  $\mathbf{M} = 60$ . The effects on the image is illustrated in Fig. 11. The lateral range gate was set to  $\mathbf{N} = 4$  beams by inspection.

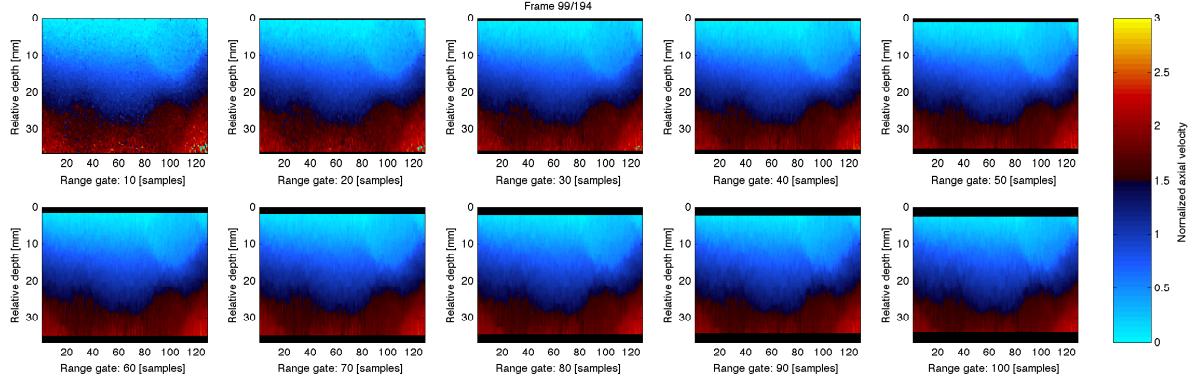


Fig. 10. Velocity images obtained using the 2D-AC method, by varying the size of the range gate ( $\mathbf{M}$ ). Each image has been normalized by the mean estimated velocity in the image

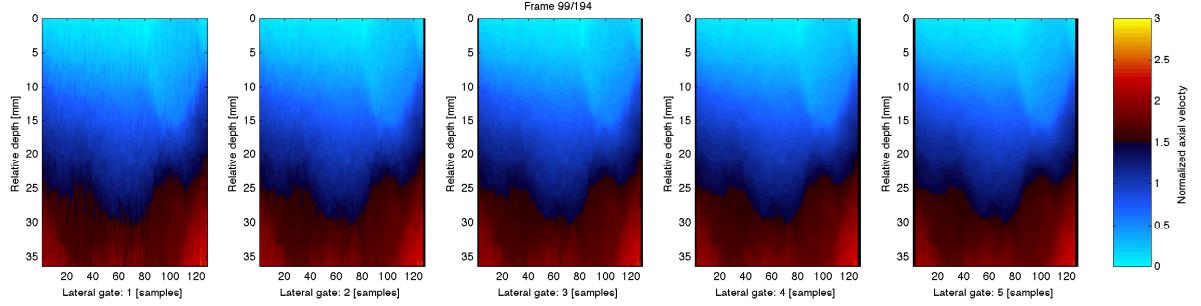


Fig. 11. Velocity images obtained using the 2D-AC method, by varying the size of the lateral gate ( $\mathbf{N}$ ). Each image has been normalized by the mean estimated velocity in the image

## 6.2 Estimator comparison

The algorithmic parameters used from now is thus  $\mathbf{M} = 60$ ,  $\mathbf{N} = 4$  and  $\mathbf{O} = 2$ . The velocity image produced by the estimators in the high velocity scenario shown in Fig. 12 and for the low velocity scenario in Fig. 13. Common to both scenarios is that velocity magnitude increases with depth, because the phantom is being compressed from underneath and speeds are highest nearest to the compressive force. In the high velocity scenario the maximal positive (upwards) velocity reached are estimated to be around 3 mm/s. This is far above the critical velocity  $v_c$  of 1.8 mm/s found earlier, and leads to wrong results in the deeper parts of the phantom for the CC method. For the low velocity scenario, there is no artifacts for the CC method and it is in general difficult to distinguish between any of methods. In regions with high velocity the 1D-AC seems to produce more noisy results compared to the 2D-AC method, which is consistent with what was seen in the simulations.

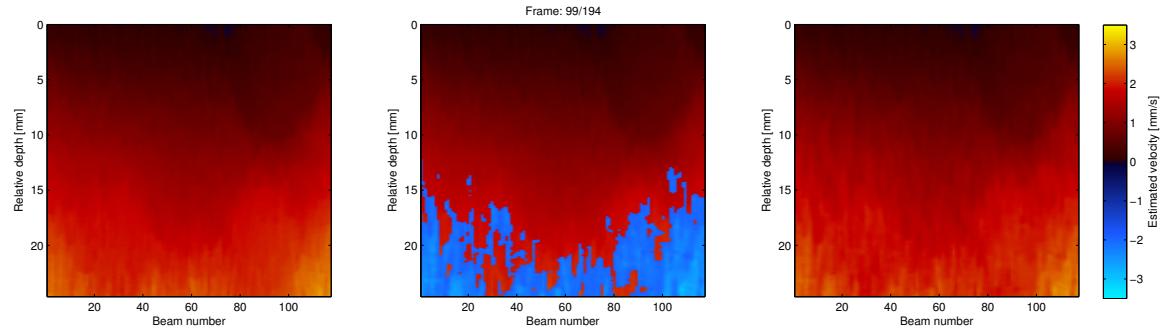


Fig. 12. Velocity images from the UX data (high velocity scenario). 2D-AC (left), CC (middle) and 1D-AC (right) methods. **Movie:** velocityUx.mov [6]

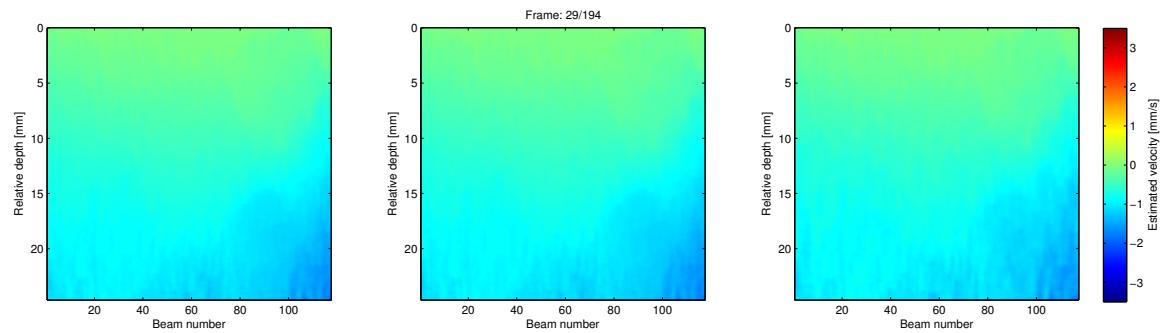


Fig. 13. Velocity images from the UX data (low velocity scenario). 2D-AC (left), CC (middle) and 1D-AC (right) methods. **Movie:** velocityUx.mov [6]

## 7 Phantom results - strain imaging and post processing

Axial strain, using eq. (2.4), can be calculated using the known velocity estimate. A decision must be made on how large the axial distance  $\Delta m$  in eq. (2.4) should be. Using the velocity estimates from the 2D-AC algorithm, shown earlier in Fig. 12, the effect of varying  $\Delta m$  is shown in Fig. 14. It is seen increasing  $\Delta m$  seems to increase the accuracy, but degrade spatial resolution. A  $\Delta m$  value of 90 samples (3.465 mm) was chosen as this seemed to give good results. Notice that, to a varying degree, the three lesions are visible in the strain images. This makes it possible to detect lesions that are not visible in B-mode from Fig. 9.

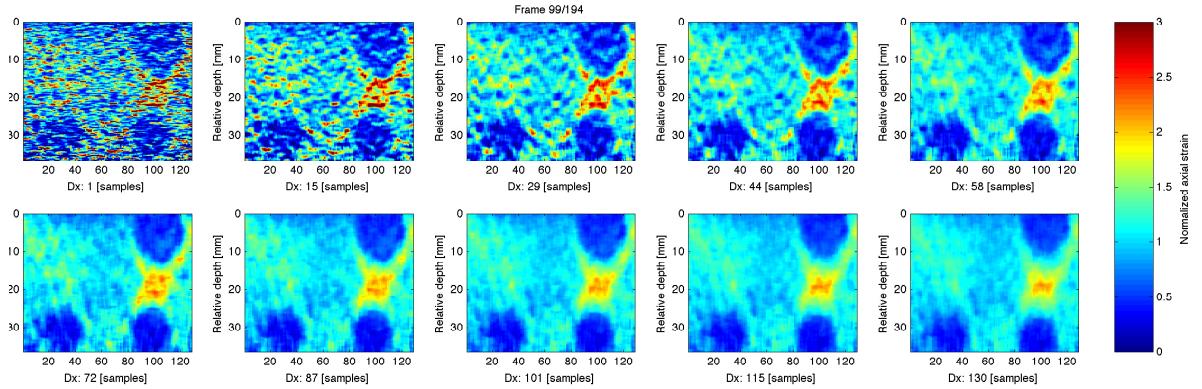


Fig. 14. Effects on the strain images by varying  $\Delta m$ . Each image has been normalized by the mean strain in the frame

### 7.1 Comparison of the different estimators after strain processing

The axial strain image using  $\Delta m = 90$  was calculated and is shown in Fig. 15 for the high velocity scenario and in Fig. 16 for the low case scenario. The units are displayed in parts per thousand (per mille). The first thing noticed is that the artifacts in the high velocity scenario for the CC method. These artifacts are caused by the incorrect velocity estimates seen in Fig. 12, completely obscuring the two embedded spheres in the bottom of the image. In the low velocity case the 2D-AC and CC methods produce nearly identical results, which agrees with what was seen in simulations. The 2D-AC method gives better results than the 1D-estimator for higher velocities, especially evident at the lower part in Fig. 15. The two deep spheres are less clear and their strain values fluctuate more for the 1D-AC estimator. Again this is consistent with the simulated results.

The problems mentioned in section 2.3 can be seen by comparing the high ( Fig. 15 ) and low ( Fig. 16 ) velocity strain images. In the two figures the values are displayed using the same fixed colormap. Notice that the soft region between the two right spheres, changes colors from dark blue (high compression) to red (expansion). Fluctuating colors on the same image region makes the image difficult to interpret, some that becomes apparent when the real-time image is watched. The contrast between lesion and non lesion is also better for the high velocity case, than the low velocity case, primarily due to the difference in dynamic region. These problems are addressed next.

### 7.2 Image post processing

Using the strain magnitude and scaling discussed in sections 2.4 and 2.5 the post processed images were produced. The result is shown for the high velocity Fig. 17 and low velocity Fig. 18 scenario. In the

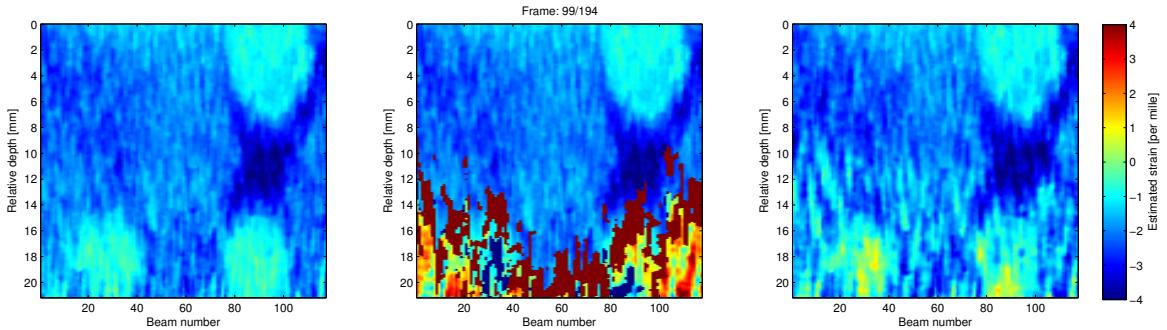


Fig. 15. Strain images from the Ultrasonix data (high velocity scenario). 2D-AC (left), CC (middle) and 1D-AC (right) methods. **Movie:** strainUx.mov [6]

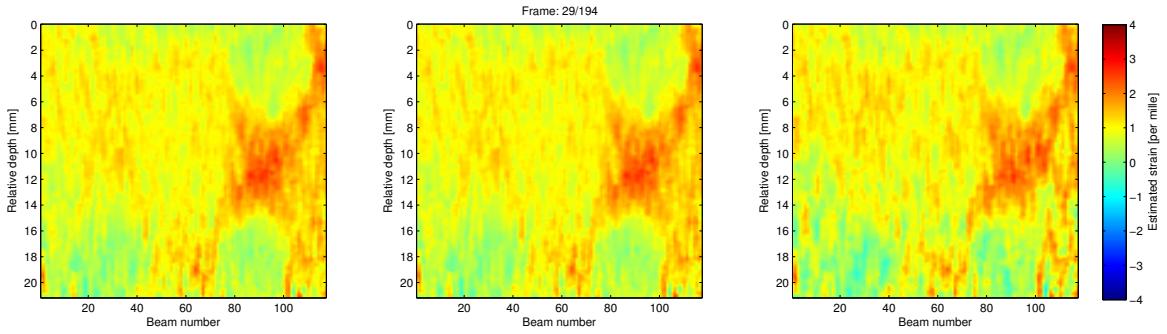


Fig. 16. Strain images from the Ultrasonix data (low velocity scenario). 2D-AC (left), CC (middle) and 1D-AC (right) methods. **Movie:** strainUx.mov [6]

high velocity case, the post processing algorithm breaks down when used on the CC method, due to the artifacts seen in Fig. 15. However for the 2D-AC and CC method, the soft and hard regions are displayed using similar colors for both the high and low velocity scenario. Thus the corresponding movie postProcessedUx.mov [6] is more pleasing to look at. However, as the results from the CC-method in Fig. 17 shows, this automatic scaling can not be used successfully if there is a lot of noise present in the image.

### 7.3 Result as it may be presented on operator screen

A common way to present a strain image on an ultrasound scanner is to overlay the strain data on top of the B-mode image. To show how the post-processed image might look on a scanner if implemented in real-time, frames showing the post-processed image mixed with B-mode image is shown in Fig. 19. The frames were mixed by replacing the intensity of the post-processed image with the intensity from the B-mode image (using the HSV color space). The post-processed images were smoothed with a gaussian filter of length 5 in the temporal direction to produce the result in Fig. 19.

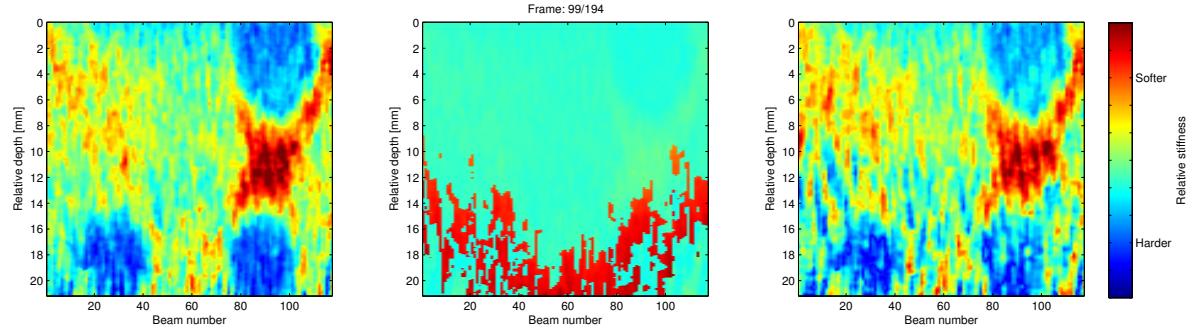


Fig. 17. Post processed images from the Ultrasonix data (high velocity scenario). 2D-AC (left), CC (middle) and 1D-AC (right) methods. **Movie:** postProcessedUx.mov [6]

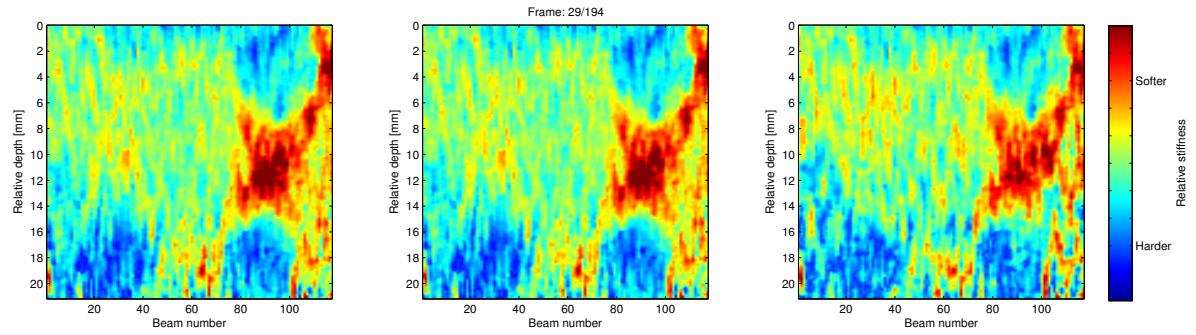


Fig. 18. Post processed images from the Ultrasonix data (low velocity scenario).. 2D-AC (left), CC (middle) and 1D-AC (right) methods. **Movie:** postProcessedUx.mov [6]

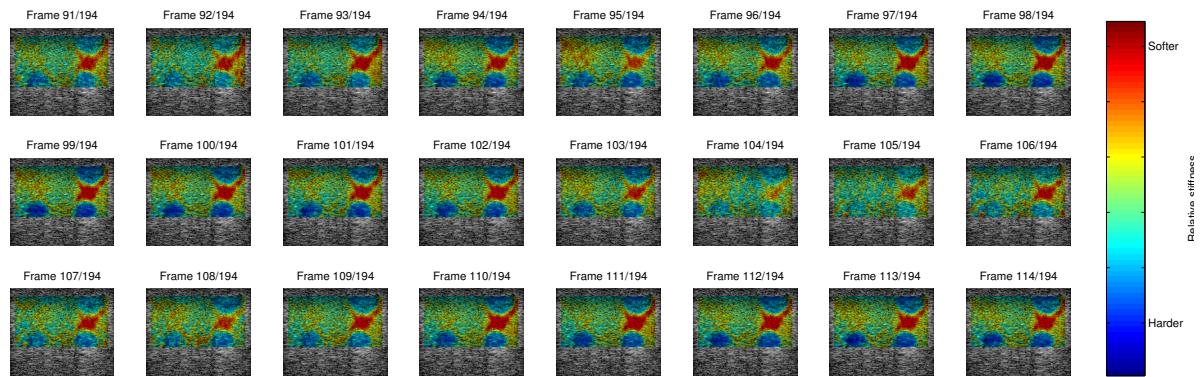


Fig. 19. Frames showing the 2D-AC estimator strain image mixed with the B-mode image over time. **Movie:** mixedBmodeUx.mov [6]

## 8 Discussion and observation

### 8.1 Estimators

The three velocity estimators have been compared using simulated and phantom data. The results from both velocity, strain and post processing shows that all stages depend on the accuracy of the velocity estimator. If all velocities in the imaged region are guaranteed to be below the critical velocity  $v_c$ , then the 2D-AC and CC estimators perform equally well. However if this assumption can not be guaranteed, then the CC method as currently implemented can not be used. The 1D-AC method consistently underperforms compared to the 2D-AC method, however the difference between the methods decreases with decreasing velocity. The most promising estimator is the 2D-AC method if it is computationally feasible, with the "light-weight" alternative being the 1D-AC method. The CC method should be avoided, unless it can be guaranteed that maximum velocities in the tissue are below the critical velocity  $v_c$ .

### 8.2 Strain and post processing

From the phantom data the difference between the B-mode and strain images is significant. The strain image visualizes the three lesions clearly, with good contrast compared to the B-mode image where they are not even visible. It is therefore reasonable to believe that the strain processing implemented here could have useful applications.

The problem on how to scale and normalize the strain values has not been solved in a satisfactory way in this project. This was attempted solved using post-processing techniques using strain magnitude and statistical signal processing. This gave good results for the phantom, as shown in the movie mixedBmodeUx.mov [6]. However the phantom data represent an ideal situation. The movement is purely in the axial direction, the phantom is compressed in a perfectly periodic fashion and the stiffness differences between the spheres and the surroundings is large. Such conditions are likely to be worse in all aspects for clinical data. This could severely limit the usefulness/correctness of the post processing setup. Further more the post-processed image shows only relative units, while clinicians prefer quantitative data for their measurements and comparisons. In spite of the limitations, an example from clinical data is shown in Fig. 20. The conventional B-mode image on the left, shows a metastasis in the brain (large structure on the left of the image). While little attempt has been made to interpret the post processed image on the right, regions of varying stiffness is clearly visible. It also looks like the metastasis is bordered by softer tissue.

### 8.3 Future work

This project has established that of the three velocity estimators, the 2D-AC method should be used if computational resources are available. It was also shown that axial strain measurements can reveal properties not visible in the B-mode image. Thus a natural next step would be to implement the 2D-AC method and strain estimation in real-time on an ultrasound scanner. Tinkering with parameters in real-time would probably be more fruitful than doing offline processing in matlab.

The problems associated with using the non-robust statistical scaling has been raised several times in this report, and it would be natural to look into different ways of visualizing the strain values. The values should be presented in a way that optimizes the detection of tumors compared to normal tissue. Another possibility is to expand the strain estimation to two dimensions, i.e calculate 2D strain, to see if the results can be improved. This is expected to provide better results in situations where the movement is considerable in the lateral direction.

It would also be interesting to see a clinical study of the results obtained after velocity and strain estimation. Could these images be of clinical value? Though similar studies have been done by Selbekk et al. [4, 5], the algorithmic parameters and post-processing methods used were different.

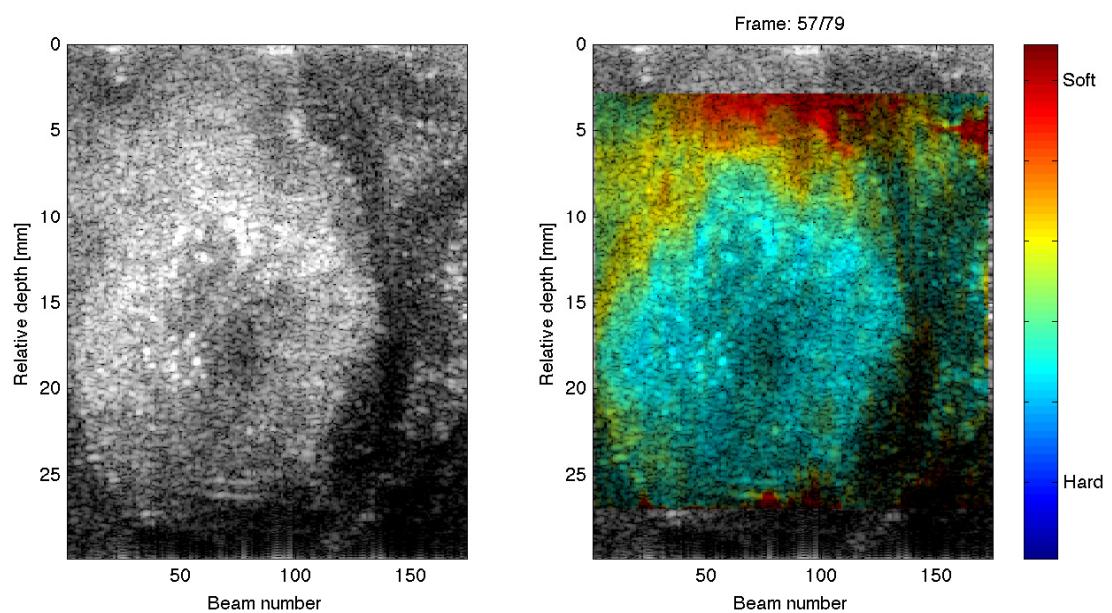


Fig. 20. B-mode and post-processed image from a metastasis in the brain. **Movie:** mixedBmodeMetastasis.mov

## A 1D autocorrelator

Following the derivation in [13], the mean frequency  $\bar{f}$  of a complex, band-limited signal  $x_+(t)$  can be expressed using the power spectral density  $G(f)$  of the signal as:

$$\bar{f} = \frac{\int_{-\infty}^{\infty} f G(f) df}{\int_{-\infty}^{\infty} G(f) df} \quad (\text{A.1})$$

Defining the 1D continuous time autocorrelation function  $\mathcal{R}(\tau)$  of a signal  $x(t)$ :

$$\mathcal{R}(\tau) = \int_{-\infty}^{\infty} x(t)x^*(t + \tau) dt \quad (\text{A.2})$$

Barber et al. used the well known Wiener-Kinchine theorem [17, page 285], relating the power spectral density to the autocorrelation function, to prove the following relationship [13]:

$$\bar{f} = \frac{1}{2\pi} \dot{\phi}(0) \quad (\text{A.3})$$

where  $\dot{\phi}(\tau)$  is the phase of autocorrelation function  $\mathcal{R}(\tau)$ . Using an approximation to the derivative  $\dot{\phi}(0)$  and exploiting the symmetry of  $\mathcal{R}(\tau)$  the the mean frequency can be expressed as [13]:

$$\bar{f} \approx \frac{1}{2\pi} \frac{\arg\{\mathcal{R}(T_{\text{PR}})\}}{T_{\text{PR}}} \quad (\text{A.4})$$

This result was developed using a continuous signal model, and is not immediately applicable to the sampled situation presented in eq. (3.2). In this report it is assumed that this result carries over to the discrete domain as:

$$\bar{f} \approx \frac{1}{2\pi} \frac{\arg\{\gamma[0, 1]\}}{T_{\text{PR}}} \quad (\text{A.5})$$

## B Gaussian correlation coefficient

Let  $x_i$ , where  $i = 1, 2 \dots N$  denote the centered x-coordinate of each of the  $N$  bars of a histogram generated from the values of an image  $v[u, v]$ . Let  $y_i$  be the number of observations for bar  $i$ , and let  $\mu$  be the mean value and  $\sigma$  the standard deviation of the image values. Using the gaussian function with  $\mu$  and  $\sigma$  as found from the image:

$$P(x) = e^{-(x-\mu)^2/2\sigma^2} \quad (\text{B.1})$$

the gaussian correlation coefficient used for this project is the correlation coefficient between  $P(x_i)$  and  $y_i$ . Treating  $P(x_i)$  and  $y_i$  as vectors, with means  $\mu_p$  and  $\mu_y$  and standard deviation  $\sigma_p$  and  $\sigma_y$  the gaussian correlation coefficient is:

$$\rho = \frac{\frac{1}{N-1} \sum_{i=1}^N (P(x_i) - \mu_p)(y_i - \mu_y)}{\sigma_p \sigma_y} \quad (\text{B.2})$$

## References

- [1] R. S. C. Cobbold, *Foundations of Biomedical Ultrasound*. Oxford University Press, 2007.
- [2] 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,” in *In Proc. Instn. Mech. Engrs., Part H: Journal of Engineering in Medicine*, pp. 203–233, 1999.
- [3] O. M. Rygh, *3D ultrasound based neuronavigation in neurosurgery*. PhD thesis, Norwegian University of Science and Technology, Faculty of medicine, Department of Neurosciene, 2008.
- [4] T. Selbekk, J. Bang, and G. Unsgaard, “Strain processing of intraoperative ultrasound images of brain tumours: initial results.,” *Ultrasound Med Biol*, vol. 31, pp. 45–51, Jan 2005.
- [5] T. Selbekk, R. Brekke, O. Solheim, S. Lydersen, T. A. N. Hernes, and G. Unsgaard, “Tissue motion and strain in the human brain assessed by intraoperative ultrasound in glioma patients.,” *Ultrasound Med Biol*, vol. 36, pp. 2–10, Jan 2010.
- [6] “<http://folk.ntnu.no/thomab/filmer/>, username: straininbrain. password: straininbrain. webpage containing the movies generated in this project.”
- [7] L. Gao, K. J. Parker, R. M. Lerner, and S. F. Levinson, “Imaging of the elastic properties of tissue—a review,” *Ultrasound in Medicine & Biology*, vol. 22, no. 8, pp. 959 – 977, 1996.
- [8] M. Scholz, V. Noack, I. Pechlivanis, M. Engelhardt, B. Fricke, U. Linstedt, B. Brendel, D. Ing, K. Schmieder, H. Ermert, and A. Harders, “Vibrography during tumor neurosurgery.,” *J Ultrasound Med*, vol. 24, pp. 985–992, Jul 2005.
- [9] J. Bercoff, M. Tanter, and M. Fink, “Supersonic shear imaging: a new technique for soft tissue elasticity mapping,” *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 51, no. 4, pp. 396–409, 2004.
- [10] A. Heimdal, A. Stoeylen, H. Torp, and T. Skjaerpe, “Real-time strain rate imaging of the left ventricle by ultrasound.,” *Journal of the American Society of Echocardiography*, vol. 11, no. 11, pp. 1013 – 1019, 1998.
- [11] J. D’hooge, A. Heimdal, F. Jamal, T. Kukulski, B. Bijnens, F. Rademakers, L. Hatle, P. Suetens, and G. R. Sutherland, “Regional strain and strain rate measurements by cardiac ultrasound: principles, implementation and limitations.,” *Eur J Echocardiogr*, vol. 1, pp. 154–170, Sep 2000.
- [12] T. Loupas, J. Powers, and R. Gill, “An axial velocity estimator for ultrasound blood flow imaging, based on a full evaluation of the doppler equation by means of a two-dimensional autocorrelation approach,” *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 42, pp. 672 –688, jul. 1995.
- [13] W. D. Barber, J. W. Eberhard, and S. G. Karr, “A new time domain technique for velocity measurements using doppler ultrasound,” *Biomedical Engineering, IEEE Transactions on*, no. 3, pp. 213–229, 1985.
- [14] P. Munk and J. A. Jensen, “A new approach for the estimation of the axial velocity using ultrasound.,” *Ultrasonics*, vol. 37, pp. 661–665, Jul 2000.
- [15] O. Standal, T. Tangen, and B. Angelsen, “P2d-4 a phase based approach for estimation and tracking of locally variable delays,” in *Ultrasonics Symposium, 2007. IEEE*, pp. 1583 –1585, oct. 2007.

- [16] C. Simon, P. VanBaren, and E. S. Ebbini, "Two-dimensional temperature estimation using diagnostic ultrasound," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 45, no. 4, pp. 1088–1099, 1998.
- [17] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing - Principles, algorithms, and applications*. Pearson Education, Inc, 4 ed., 2007.