CHALMERS

Motion compensation for IMT measurements

PETER HULTQVIST

Digital Imaging and Image Analysis Group
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden, 2010

EX002/2010.

Abstract

Atherosclerosis is a disease where deposits on the arteries have grown, and thereby decreasing the diameter of the artery, so that they affect the blood flow. These deposits can be observed by measuring the thickness of the two innermost layers of the artery, the intima and adventitia. This thickness is commonly referred to as the intima-media thickness (IMT). The IMT of the carotid artery is about 0.5 mm for young people, and increases slowly with age up to over 1 mm.

Ultrasound is a non-invasive method that can image the arterial wall with 0.1 mm accuracy. To be able to perform measurements with only a few years interval there is a need for an even more precise method.

Our first step has been to transfer the video data from a video tape into a digital format. A continuous full frame and full frame-rate video was recorded onto a hard drive.

We have worked with a method that aims at improving the accuracy of IMT measurement using information found in full video ultrasound measurements rather than an image from a single frame.

A motion tracking technique called phase correlation was used to determine the translation between frames in the video. The motion information has then been used to extract a motionless video of the arterial wall. The video has been merged along the time axis with a few different techniques giving output images with higher resolution.

The choices of merging technique was studied to determine how they affected the separation of the layers in aspect to their intensity distribution. This was done by measuring the intensity distribution over time for each layer. The overlap between the layers intensity distribution was measured on the merged images and compared to that of a single frame.

We found that the layers overlap in intensity distribution did decrease. For the lumen-intima edge the overlap was reduced by 50 – 100%, whereas for the media-adventitia edge the reduction was about 20%.

Keywords: IMT, Motion compensation, Phase correlation, Ultrasound

Table of Contents

Figure Index

1 Background

1.1 Cardiovascular Diseases

Atherosclerosis

The cardiovascular system consists of the blood, the heart and the blood vessels. As the human body ages, the walls of the arteries gets harder and thicker, which decreases the flow through the arteries.

Atherosclerosis is a type of cardiovascular disease, which causes the artery walls to thicken. This is caused by the buildup of cholesterol deposits on the inside of the artery. The effect of these deposits is a decreased diameter of the lumen – the artery opening, which reduces the flow. If the buildup of the deposits continues further, there is an increased risk of deposits breaking loose. The loose deposit can cause a clog, blocking the flow further down the arteries.

One indicator of the risk of cardiovascular diseases is to measure the narrowing of the lumen. While some suggest that the risk of cardiovascular diseases can be better determined by the thickness of the wall [1].

The artery wall has three innermost layers called intima, media and adventitia respectively. These layers can be seen in Figure 1. The density of the adventitia is higher than the other two, making it possible to distinguish it on an ultrasound image. The intima and media layers, however, are similar in density and can therefore not be distinguished. The deposits are building up between the intima and media layers, as seen in Figure 2. Therefore the intima-media thickness (IMT) is used as a measurement of the buildup.

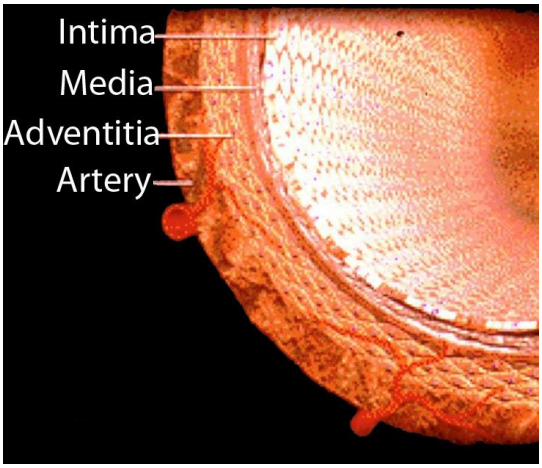


Figure 1: *The layers of the artery wall*

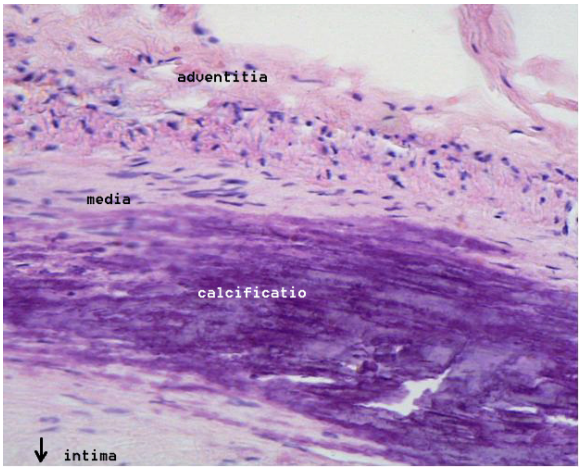


Figure 2: *Micro-photography of arterial wall tissue with calcified atherosclerotic plaque*

1 Emile Mohler, http://knol.google.com/k/emile-mohler/carotid-artery-disease/vSQWW7xh/ieaBPQ
2 Wikimedia Commons, http://commons.wikimedia.org/wiki/File:Calcificatio_atherosclerotica.jpg

The carotid IMT is 0.5 mm for young people and grows to over 1 mm during a normal person's lifetime [2].

When studying the IMT, the most commonly studied artery is the carotid artery, which is close to the skin along the neck, see Figure 3 . This makes it easy to access and measure.

Screening

The study of the IMT can give an indication on the health status of the person, also known as the cardiovascular age [3].

Screening is being done today to assess the risk for arteriosclerosis [4]. Future screening could include people in much earlier stages [5]. To do this it must be possible to measure very small changes in the IMT with an accuracy better than 0.1 mm. The measurements must also be invariant between each examination, which could be as little as a few years apart.

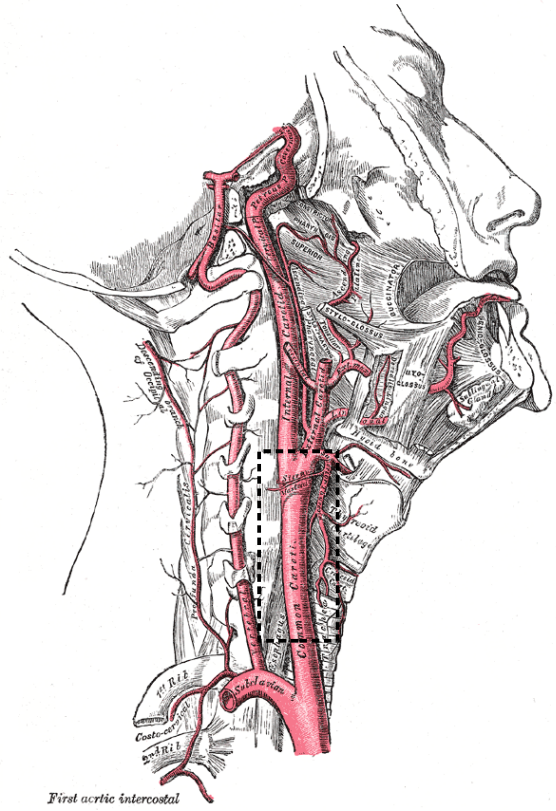


Figure 3: *The carotid arteries*

1.2 Ultrasound

Ultrasound is a non-invasive method to image the inside of the body. It is based on the transmission and reflection of acoustical ultrasonic waves in the 100 kHz range.

The transceiver is an electro-acoustic transducer using a piezoelectric crystal. This device generates the pulses from electrical signals, and receives the reflections, which it transforms into electrical signals that can be detected by the measurement equipment.

As the ultrasonic pulse travels between tissues of different density, part of the energy is reflected in the opposite direction. The reflection going back to the transceiver is measured in intensity, and the time from since the pulse was transmitted. This information is recorded and used to generate an 1-d image of the body.

This is what makes it possible to image tissue of different density, but it also makes it harder to distinguish tissue with similar density.
Using higher frequencies gives a shorter wavelength and thereby higher resolution. Higher frequencies though have higher attenuation, therefore the depth that can be measured is decreased .

An array probe, see Figure 4 , contains a linear array of transceivers that work together to transmit waves and detect the reflections. The recording of the transmitted and the received wave is used to generate a 2D image that represents a slice through the body.

The IMT is measured using this kind of probe. An example of an ultrasound image is shown in Figure 5, where we see the cross section of the carotid artery with the boundaries of the layers marked.

Figure 4: *Ultrasound Probe*

3 Anatomy of the Human Body: http://www.bartleby.com/107/146.html#i513
4 Daniel W. Rickey, http://en.wikipedia.org/wiki/File:UltrasoundProbe2006a.jpg

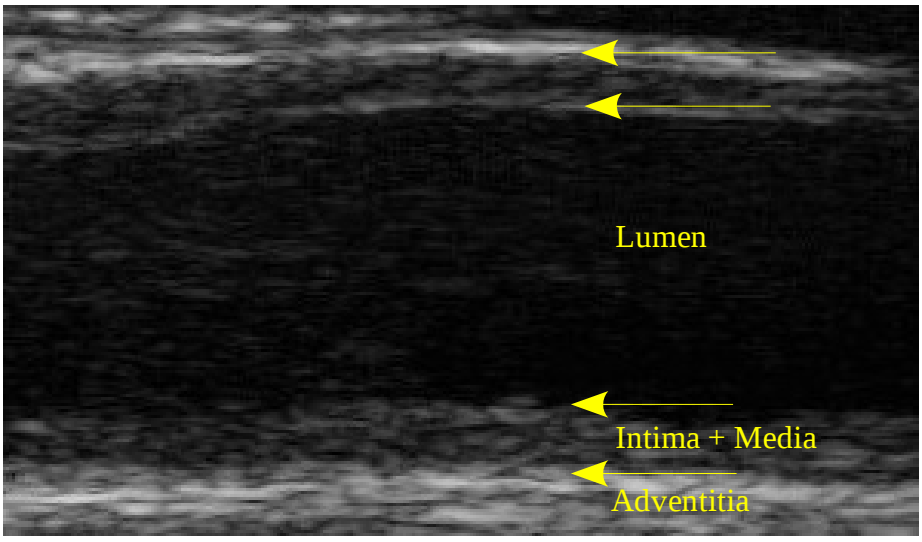


Figure 5: *Example of an ultrasound image of the carotid artery*

Using this technique there are several well known issues that reduce the quality of the imaging technique.

The detected reflections are assumed to have traveled a straight path to the edge being reflected at and back. When the detected wave has instead traveled along another, longer path, it will be represented as being located further away. This is known as false echoes, it is most noticeable after tissue having high density giving strong reflections.

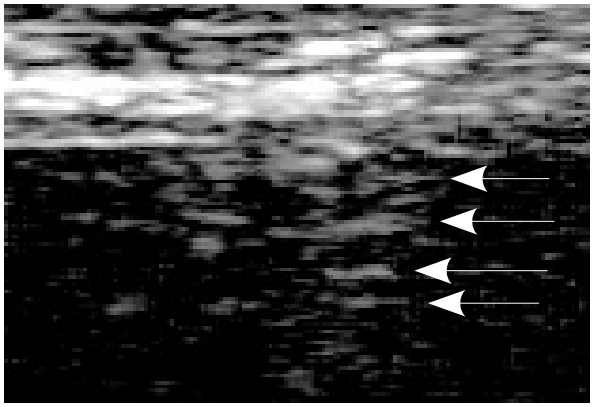


Figure 6: *False echoes from the wall above*

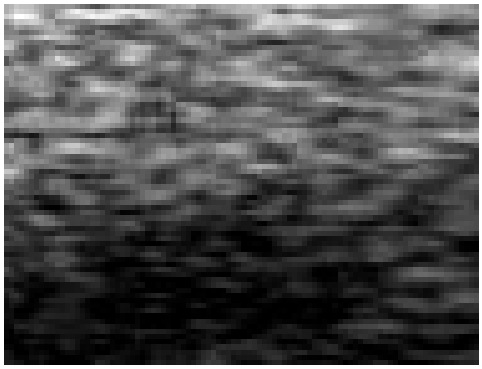


Figure 7: *A typical pattern in ultrasound known as speckle*

Echo dropouts are echoes of tissue that disappear from the image for a few frames. It occurs when the reflecting wave does not returned to the transceiver.

The most known disturbance of ultrasound is the speckle effect. This is the coarse time invariant noise found in Figure 7. They represent properties in the tissue that is smaller than the wavelength of the ultrasound, and is therefore not correctly represented. However, since they represent properties in the tissue they will remain invariant over time.

2 Proposed work

In this project we have been working with a video recording of ultrasound. In our data the arterial wall has movements caused by the heartbeats. During these movements we have seen that the imaging is affected mostly because of the motion, but also because of the change in angle that affects the imaging process.

We want to extract a time sequence of a static arterial wall. This sequence can be used to generate new improved input to edge detection algorithms. This is done by first determining the motion of the wall and then extracting the new motion compensated time sequence. In the new sequence we are now able to detect new variables such as intensity variation over time. The sequence can also be merged into one single frame with less noise and higher resolution than a single frame from the original video.

Since there are material being stored on a prerecorded video cassette we will construct a system for importing them into a digital format that can be used in further digital processing. The goals here are to be able to transfer a continuous playing video into a computer with limited computational resources.

2.1 Limitations

The work will be focused on improvements from using video data rather than single images. Methods applied to single images will not been considered during this work.

2.2 Material

Our material is four video sequences from different measurements. The videos have a resolution of 768 × 576 pixels, that is the resolution of traditional TV. The ultrasound data are located in a window of the image having the size of 529 × 277 pixels, where the area of interest is a smaller part of that image. The ultrasound resolution is 13.8 pixels/mm. The length of all video sequences are 75 frames at 25 Hz frame rate, corresponding to 3 seconds.

Figure 8: *Full frame from ultrasound instrument video*

A fifth measurement is located on a video cassette with about 15 minutes of video. This will be transferred into a digital format.

2.3 Tools

The main numerical tool used was Matlab, which is a mathematical program focused on working with data sets in matrices. Its matrix base is a good start for storage of the image and video data. The interactive features allowed us to start up fast with our ideas, and the scripting possibilities were used to automate the final method over all data. Built in functions such as the Fourier transform and a native use of complex numbers have made it a good environment to work with.

The acquisition software was written in C, using a library from the manufacturer of the frame grabber card. The development environment was Visual Studio.

2.4 Methods

Our work can be divided into several parts. Each will be described later in more detail.

Imaging (Ch 1.2), where the tissue in the body is imaged onto an electrical representation, here a video signal. This is done using an ultrasound equipment and is not within the scope of our work.

Video acquisition (Ch 3) is the part where the analog video signal is recorded into a sequence of digital images that are usable for our later work. Modern ultrasound equipment is fully digital and thus will make this step unnecessary. However, older measurements recorded on analog tape will have to be transferred into digital form before they can be analyzed.

Motion estimation (Ch 4) analyzes the images and extracts how the subject, in our case the arterial wall, has been moving over time. In this step we have tried several techniques to handle noise and errors in the motion measurement.

In **processing** (Ch 5.1, 5.2), we further reduce errors and extract the most correct data, even though this means loosing data to work with in the end. Here, we finally combine the motion data with the image data onto what looks like the original video with the motion compensated. From this volume we extract some new parameters otherwise not available in a single image.

3 Video acquisition

Analog video data from ultrasound measurements can be found stored on video cassette. Video can also be acquired directly via video output from older analog as well as newer digital ultrasound equipment. For our work these data needs to be digitized into a form that can be used in further analysis. A goal was to be able to acquire continuous video, that is video at full frame rate during at least 15 minutes. This would be enough to be sure that there is no time limit to the recording other than the storage capacity.

Our setup consisted of a VHS player, a video frame-grabber card and a PC. The output of the VHS player is a composite video signal that is transferred via a coaxial cable to the frame grabber. The frame grabber[5] records uncompressed full video images at a rate of at least 25 fps. The dynamic range is adjustable, but the digitizer is limited to 8 bit depth of gray scale only. The card is operated on a regular computer with a PCI slot and from the manufacturer supplied drivers, recording software and a library for interfacing to the driver directly.

5 Matrox Meteor-II/Standard

The software accompanying the card could, used on our PC, record single frames and very short video. This limitation was caused by the limited RAM and computational power needed to compress the video resulting in buffers filling up faster than they could be compressed. In practice at most a few seconds of video data could be recorded.

Using the programming libraries for controlling the frame grabber, the recording process was controlled in more detail. A software written in C using the Visual Studio development environment was used to interface to the library.

Figure 9: *Matrox II frame grabber card*
Source: Matrox product sheet

To minimize the risk of dropped frames caused by other IO activity, our program was designed with a circular buffer and two threads. One thread for interfacing to the frame grabber library recording to memory and the other to store the data onto disk. This enabled a continuous acquisition. This design prevents either thread to be locked by the other.

The task of the first thread is to respond to captured frames from the library, transfer them from the buffers in the card via the library into the circular buffer and prepare a new buffer for the next frame to be captured. Most of this was done using the library's functions.

The task of the second thread is to transfer the buffers from the circular buffer onto the disk. This was done using the operating systems standard function calls.

A main thread controls the two worker threads. Its main purpose was to configure the frame grabber card via the library and initiate the process with the two treads.

The stored video is a binary file with each frame stored uncompressed one byte per pixel representing the gray scale intensity.

This setup was enough to perform the video capturing. Using a faster computer the acquisition program could be extended to include image processing algorithms or user interface for processing the data live during acquisition.

4 Motion estimation

We want to determine the relative motion in a small area between two images. This estimation of image shift is done through the whole video time and at many positions. This information will later be used in section 5.1 to generate the final motion compensated video.

The motion of the entire wall is tracked by following the motion of a small number of windows. These windows are chosen to be evenly distributed along a rough estimation of the wall. The crosses in Figure 10 represent the center of each window. The rectangle show the size of one window. Since only relative motion is detected, as long as the wall is within the frame, there is no need to know the exact location of the wall. Therefore we could use manual input.

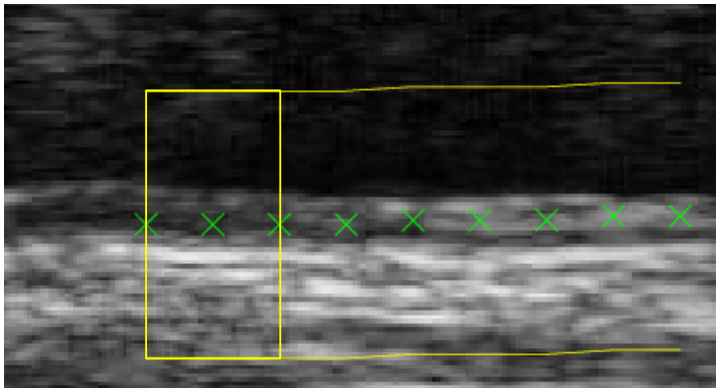


Figure 10: *Window for motion tracking.*
The crosses mark the center for every window.

We want to estimate the relative motion of the tissue centered at each window. We want to locate the same area of the tissue in every frame. We will only be interested in the relative positions of these areas compared to neighboring frames. Thus the output will be a trail representing the motion of the wall in the video in reference to an arbitrarily chosen point within the window.

The trail is detected by comparing the relative translation between each neighboring frame. The relative translations are then accumulated over time to get the position of the wall in all frames relative to the first one.

The method is working under the assumption that the entire window is only translated. Other effect such as rotation and distortion are not detected, and if present will be a factor decreasing the precision of the detected translation.

Two methods of detecting the translation was tried, cross correlation and phase correlation. Both methods use the overall surface structure, the wall as well as the time invariant speckles, to detect the translation between them. This is required since the object being tracked will never fully be within a window.

The motion tracking detects both translation along the x and y axis.

4.1 Cross Correlation

This is a well known method used to detect the existence and position of one object inside another, usually larger, image. The discrete version of cross correlation is defined in one dimension by (4.1), and in two dimensions by (4.2). In (4.1) the cross correlation(r) is itself a function of x. The variables x, y, xi and zeta are all integers, and f and g is sampled at integer variables, for this discrete version of the cross correlation. Even though the sum limits are infinite, they are in calculations limited by the area that f and g overlap, outside the cross correlation is zero.

$$r_{fg}[x] = (f*g)[x]$$

$$(f*g)[x] \stackrel{def.}{=} \sum_{\substack{\xi=-\infty \\ x,\xi\in\mathbb{Z}}}^{\infty} f^*[\xi]\,g[x+\xi] \qquad \text{(4.1) Cross Correlation}$$

$$r_{fg}[x,y] = (f*g)[x]$$

$$(f*g)[x,y] \stackrel{def.}{=} \sum_{\substack{\xi=-\infty \\ x,y,\xi,\zeta\in\mathbb{Z}}}^{\infty} \sum_{\zeta=-\infty}^{\infty} f^*[\xi,\zeta]\,g[x+\xi,y+\zeta] \qquad \text{(4.2) 2D Cross Correlation}$$

Here f is the image being searched for, and g is the image searched within. The output is a translation image where the coordinate and value of each pixel corresponding to how well the two images match at each translation vector. The most likely translation between the two images is found by locating the maximum of the output.

The size of g was set to the window size and the size of *f* was set to twice that size along both dimensions. This was done to get a translation image that, within the window size, corresponds to the cross correlation where f and g fully overlaps.

Figure 11 shows a typical cross correlation between two frames of the arterial wall. In both examples the X in the top images mark the maximum, the detected translation, which corresponds well with the true one. The bottom two images show the corresponding maps as meshes where the height/z-axis represent the scaled cross correlation magnitude.

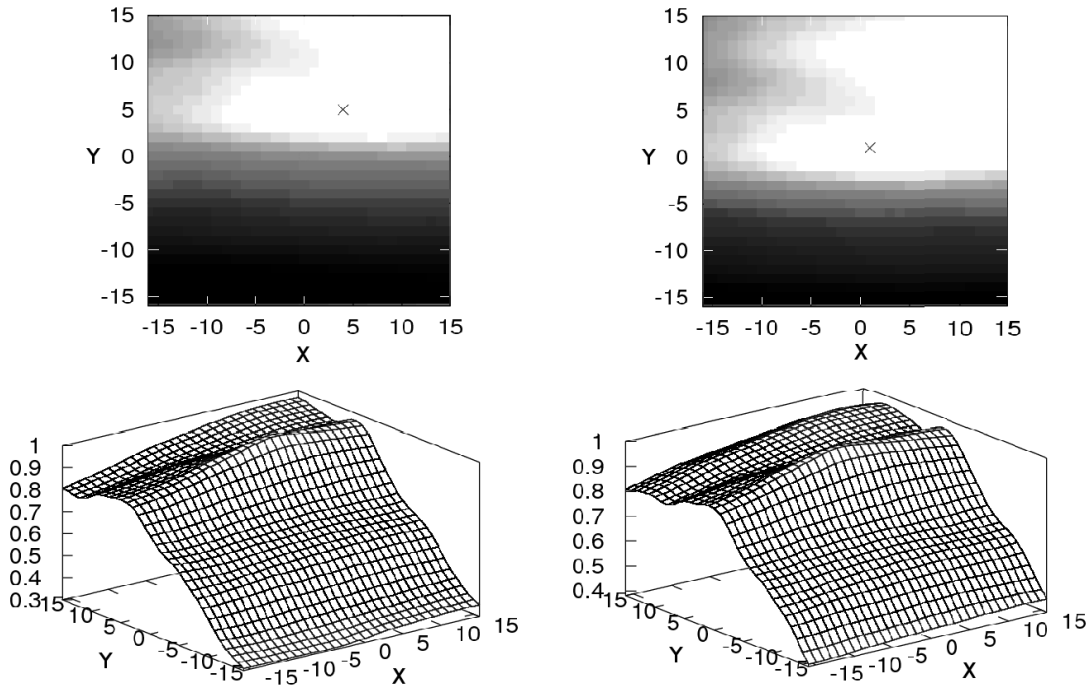


Figure 11: *Cross correlation magnitude scaled to 1*
True shift: left image (5,5), right image (1,1)
X mark the detected shift at these positions.

In general, cross correlation was found too often to yield too large errors in its detection of motion to be useful. We found a problem with this method in combination with the material used. Since there is only one edge in our material there is no matching “force” that prevents the estimation to go too deep into the wall. The detected translation had a tendency to be drawn into brighter areas, which commonly are located in the adventitia, further from the edge. Figure 12 shows how a true translation of (0, 0), marked with a circle, is detected as a much larger translation in the upper right corner, marked with a X.

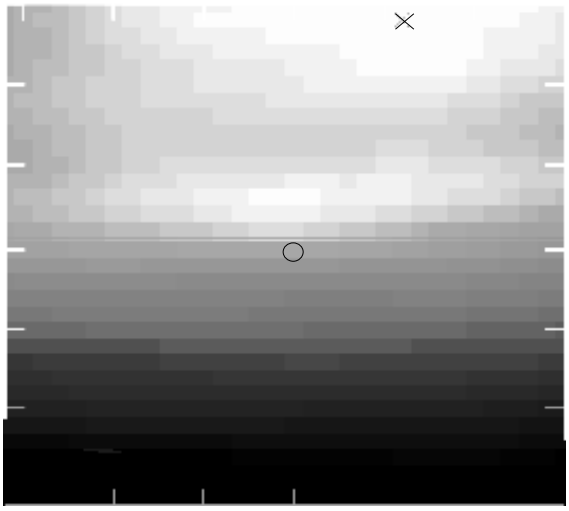


Figure 12: *Failed cross correlation match Match marked with X at the edge. True shift is at the center, marked with O.*

4.2 Phase Correlation

Phase correlation is based on a translation property in the Fourier transform. This will be described later.

In this work the same phase correlation method will be used as in many other methods [6] [7] [8] [9]. What differentiates are mostly the methods used to determine the sub-pixel translation as well as the processing of the detected data. A few differs more by using the cosine transform, DCT [10].

In one method to generate super-resolution images [11], the source images were up-sampled to be able to read sub-pixel translation. We found in our tests that up-sampling did not generate any more information for our material compared to what an up-scaled peak map would generate.

Another publication [12] refers to phase correlation as POC (Phase Only Correlation) to reflect that the intensity is not used in the method.

Phase correlation is comparing the translation between two images. As opposed to cross correlation the basic phase correlation compare two equally sized images. From both images the Fourier transform is calculated, which represent the image in sinusoidal basis-functions, each with a phase and amplitude component.

In phase correlation, the difference of each frequency component's phase is calculated. From each shared frequency and phase difference, a limited number of possible translations are calculated. By combining the possible translations for each frequency component, a combined translation probability map is constructed.

The Fourier transform

The definition of the Fourier transform can be found in its continuous form (4.3) and its discrete form (4.4) where f is N-periodic and sampled at integer values.

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i x \omega} dx$$

(4.3) Continuous Fourier Transform

$$F[k] = \sum_{n=0}^{N} f[n] e^{-i \frac{2\pi k}{N} n} dx$$

(4.4) Discrete Fourier Transform

One principle used in the Fourier transform is that any function can be represented by combining a set of orthogonal base functions. For the Fourier transform these base functions are the sinusoidal functions $f_b = e^{-i\omega}$. For each ω and its corresponding base function f_b there is a complex constant $F(\omega)$ that is given by the Fourier transform. This constant represents the amplitude and phase of each base function. As the name implies only the phase component of this constant is used in the phase correlation method.

Phase translation statistics

We now have the two images, a and b, their corresponding Fourier transforms F_a and F_b and their phases from $P = \arg(F)$. The phase difference between the two images is calculated for each ω, $R = \text{modulus}_{2\pi}(P_a - P_b)$. The amplitude information is now discarded as it is of no interest.

According to our explanation later, the amplitude would not have to be discarded by us, since it would be canceled out using the calculation $R = F_a / F_b$. In practice we found that there are amplitude differences and these differences do not relate to the translation calculations. Therefore we normalize the amplitude of F before calculating R by the division formula.

The translation for each frequency component in R can be calculated using the formula $T(\omega) = 2\pi/\omega(R/2\pi + n)$, where n is an arbitrary integer. From the formula we have an infinite number of possible translations, but we can already now conclude that the translation must be within the size of the window.

Since most components periodicity is less than the window, there are multiple translations possible for each components phase difference. However, these possible translations are different for each component, except the true one, where they all coincide. This is determined by generating a probability map for all possible translations that summarizes all probable translations from every frequency component.

A simplified example can be seen in Figure 13 where top plot represents two signals where the red/dashed is shifted 5 lu. to the left of the blue/solid. The three plots below show the three frequency components that the signals are made from.

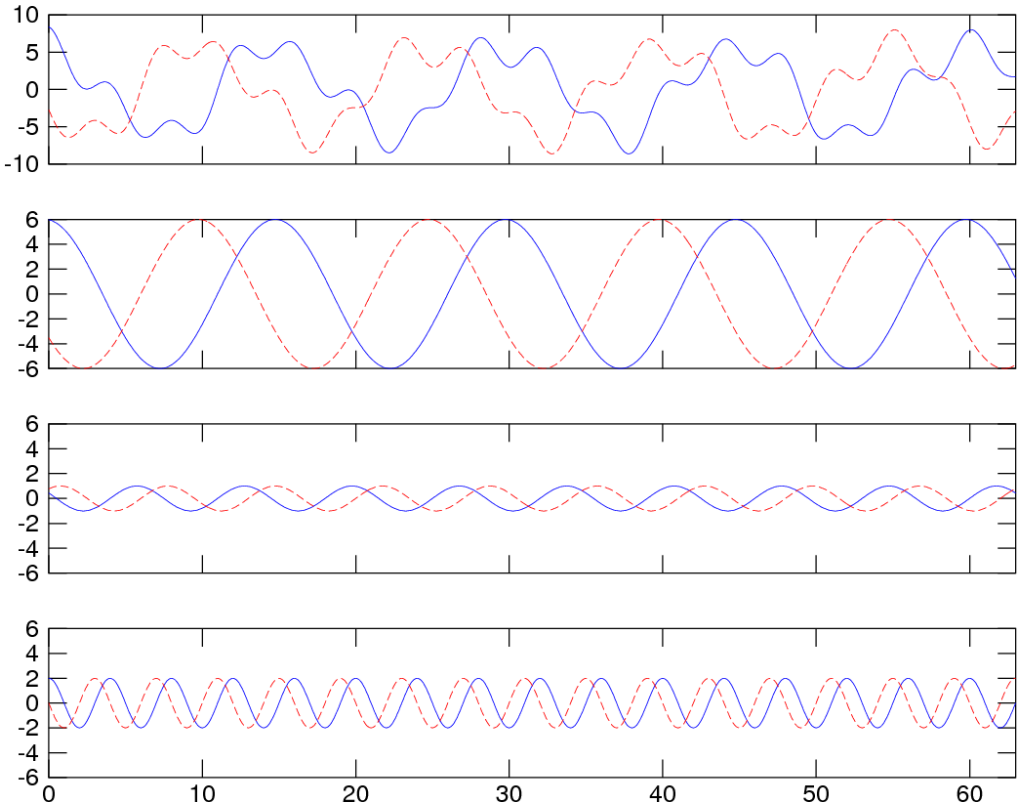


Figure 13: *Illustration of phase difference for a signal composed of three sinusoids.*
Solid/blue: Original signal, dashed/red: translated signal.
Top graph: two signals translated 5 lu. apart.
Second to fourth graph: The three frequency components of the two signals.

In each frequency component plot, the possible translations can be read as the distance between two peaks, from the solid/blue to the dashed/red plot. As seen, the length of the possible translations, as well as the number of them, is dependent on the frequency of its component.

In the top plot of Figure 14 the possible translations are shown as circles, together with a smoother probability curve, one for each frequency component. The bottom plot shows the summarized result of these three plots.

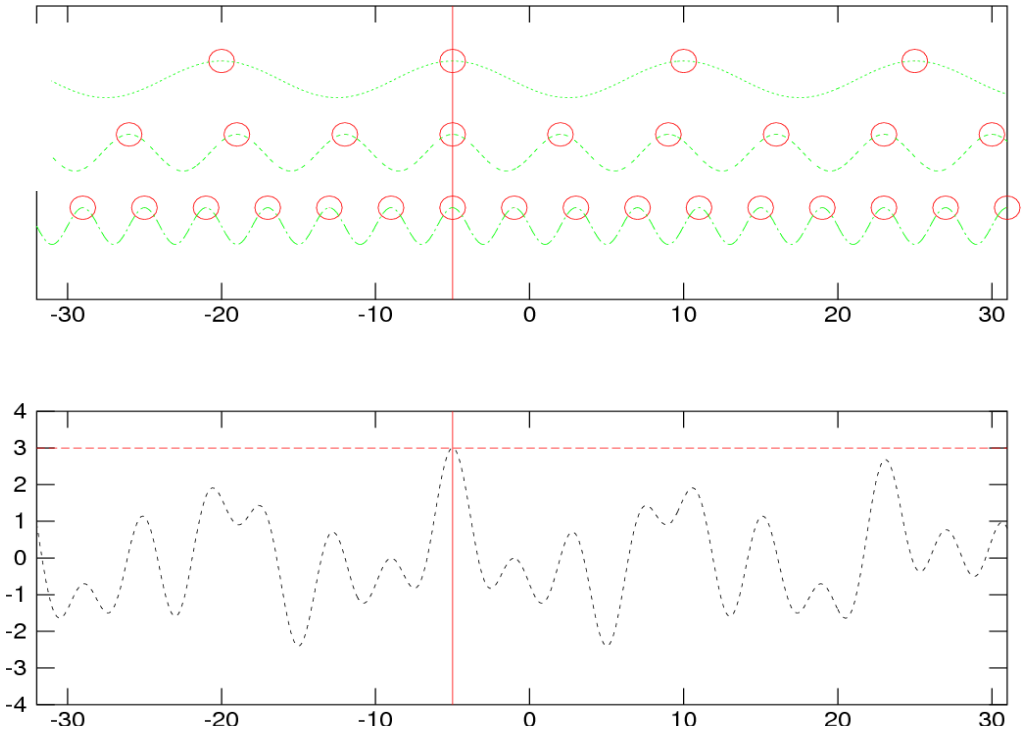


Figure 14: *Relative probability for each translation*

The outcome of this example shows that the most probable translation is at -5, which is also the true value. Although the outcome was correct the margin is very small, the second largest peak at 23 is very close. Here we used three wave components. In practice the most probable peak will be more distinct when more components are used. For a 16 × 32 pixel sized window, 512 wavelengths are combined, which generates a very distinct peak as shown later in Figure 21, page 14.

To generate the translation probability map, sinusoidal functions were used to represent distinct possible translations. The formula for generating the translation probability can thus be expressed, within a constant 1/N, as in (4.5), which is the same as the inverse discrete Fourier transform.

$$r(n) = \frac{1}{N} \sum_{k=0}^{N} R_C(k)\, e^{i\frac{2\pi n}{N}k}$$
$$R_C(k) = e^{iR(k)}$$

(4.5) Discrete Inverse Fourier Transform

Hough transform analogy

This last step in generating a probability map has similarities with the Hough transform. Both methods import a matrix of data where each single value inconclusively can indicate many solutions. For the Hough transform, a continuous line, and for the phase correlation a limited number of points. When all these inconclusive results are combined we get an image of the most probable ones.

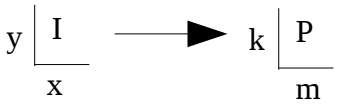


Figure 15: *Hough Transform*

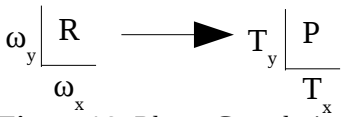


Figure 16: *Phase Correlation*

The Hough transform for straight lines, described as y = kx + m, translates the intensity in an x-y image onto the probability in a k-m map. Using phase correlation the map of phase differences are transformed into a map of translation probabilities.

Tracking multiple motions

As the Hough transform can detect multiple lines we made a test to see if the phase correlation had similar properties.

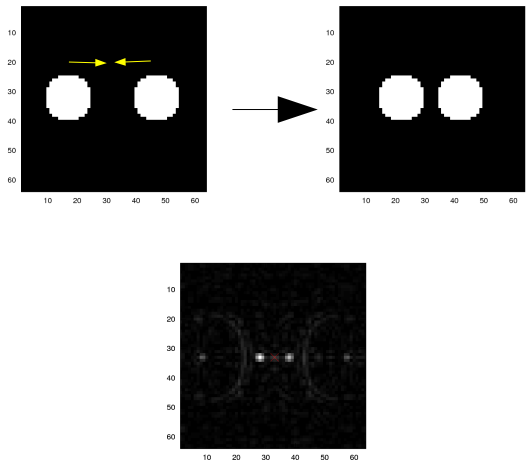


Figure 17: *Phase Correlation test 1*
True motion: 5 right, 5 left

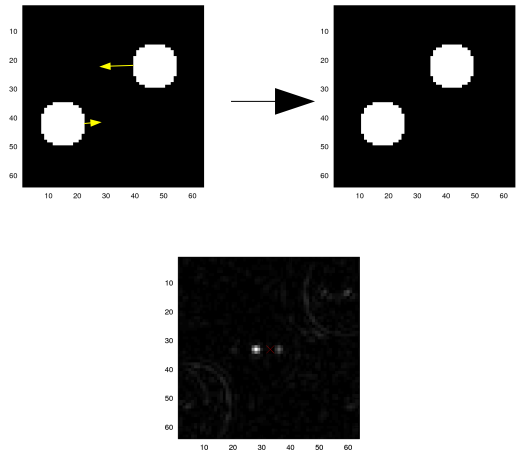


Figure 18: *Phase Correlation test 2*
True motion: 3 right, 5 left

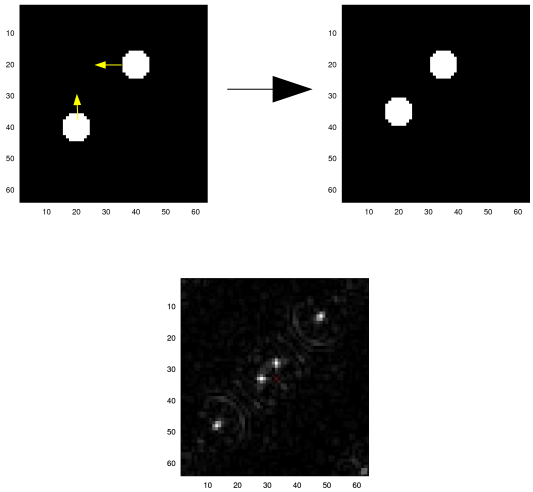


Figure 19: *Phase Correlation test 3*
True motion: 5 up, 5 left

In each group of images the two at the top are frames 1 and 2, and the bottom image shows the translation probability map. The true motion found in the title is also detected correctly. Some additional peaks can be seen that are especially noticeable in Figure 19. These extra peaks represent the motion interpreted as if the two discs switched places. In this situation with the two discs being identical, it is a valid conclusion that can be drawn from only the two images.

We notice that a smaller area moving in a different directions would cause another peak in the translation probability map. If this peak is separated enough from the main peak, so that they can be distinguished, the disturbance from this smaller area motion will not affect the measurement. However, when the motions are close we will have a disturbance. As we will see later, the motion from a smaller area will not affect our measurement of the larger area when their directions and speed differs.

Determining image translation

From the translation probability map the translation can be observed as the coordinate of the maximum value represented as a distinct peak. For Figure 20 we have used a 64 × 64 window over the edge of an IMT ultrasound video. It shows this translation probability map between 5 images, with a slightly larger motion between the second and third image as seen in the second map below. Figure 21 shows a cross section in one map for x = 0.

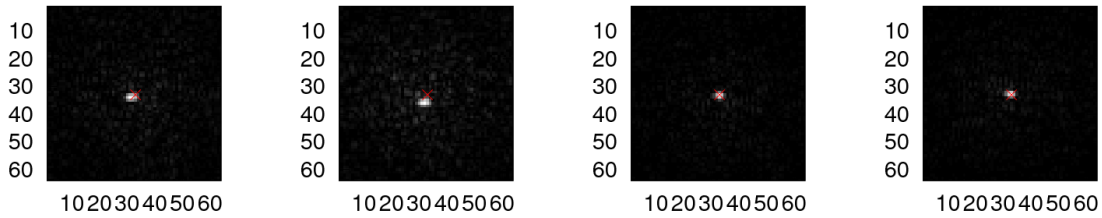


Figure 20: *Translation maps between 5 images in sequence*
The cross in the center of each image mark the position for zero translation.

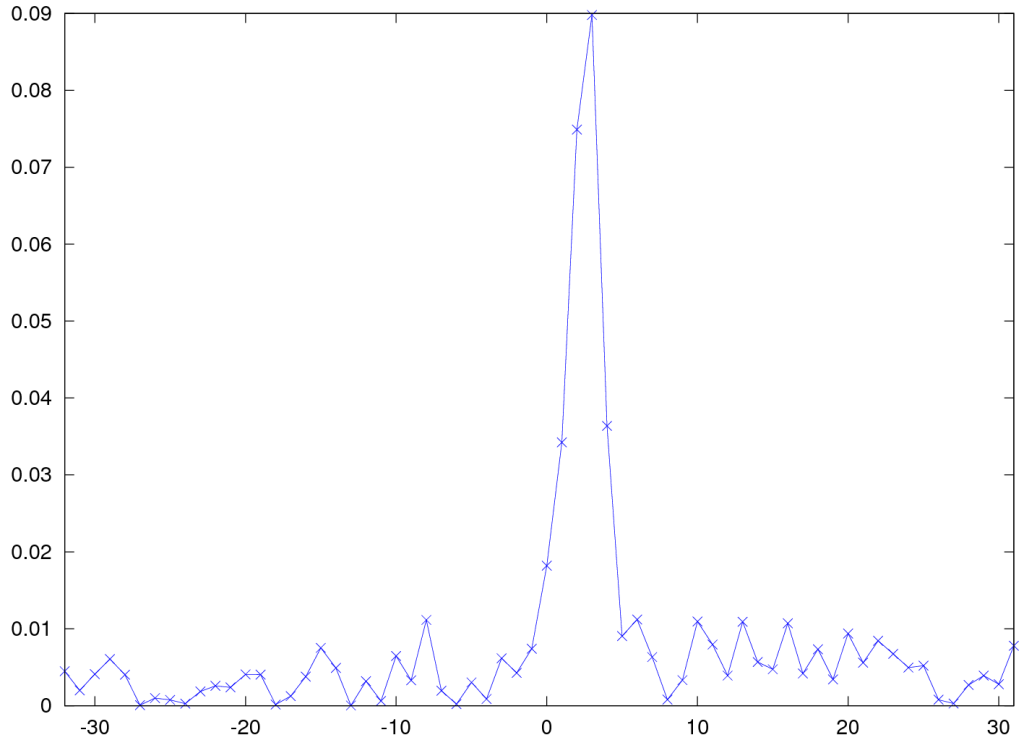


Figure 21: *Y-slice across the main peak in a translation map*

Since the motion in our video material was of low speed, 0-3 pixels/frame, the accumulating error from only using pixel resolution detection would grew too large over the full video. Therefore a sub-pixel accuracy was needed. To get sub pixel resolution in the shift estimation the maximum was first located. After that, the sub-pixel was calculated by using a mass weight center formula (4.6).

$$\Delta x = \frac{\sum_x r[x]x}{\sum_x r[x]} \qquad \Delta x = \frac{\sum_{x_{max}-1}^{x_{max}+1} r[x]x}{\sum_{x_{max}-1}^{x_{max}+1} r[x]}$$

(4.6) Mass weight *(4.7) Localized mass weight*

When we calculated this over all x, we found that even though the peak was significantly larger than the noise, in the order of 10, the large area (here 64 × 64) of the noise made the noise more dominant. Therefore the weight center calculation (4.7) was only applied using the nearest 3 × 3 pixels surrounding the maximum.

Mathematically derivation of phase correlation

So far we have tried to use an intuitive motivation for the phase correlation method. A more strict mathematical description will now follow.

We start with two functions, *f* and *g*. The function *g* has the same values as f, but is shifted T pixels to the right. This is described by the convolution with the shifted Dirac function.

$$g[n] = (f * \delta_T)[n] = f[n-T]$$

The Fourier transform of *f* and *g* are related as:

$$G[\mu] = F[\mu] e^{-i\frac{2\pi}{N}\mu T}$$

We want the phase difference between F and G. In complex mathematics a division equals subtraction of the arguments and division in their amplitude. Since the difference in f and g is only a translation, F and G will have the same amplitude for each μ. Thus, dividing *G* by *F* will give us the phase shift function *R*:

$$\frac{G}{F} = \frac{|G|}{|F|} e^{i(arg(G)-arg(F))} = \left[|F| \equiv |G|\right] = \frac{|G|}{|F|} e^{i(arg(G)-arg(F))} = R$$

$$R[\mu] = \frac{G[\mu]}{F[\mu]} = e^{-i\frac{2\pi}{N}\mu T}$$

Now transforming *R* back using the inverse discrete Fourier transform we get:

$$r[n] = N\, d_T[n]$$
$$d_T[n] = \begin{cases} 1, n = T + mN, m \in \mathbb{Z} \\ 0, \text{otherwise} \end{cases}$$

This is the same translation probability map found previously. It shows that the phase correlation method will give a distinct peak at the translation coordinate.

From this derivation we found that the amplitude is canceled and thus is not part of the final result. When the two images are not perfect translations of one another, we must discard the amplitude difference in the phase correlation calculations.

4.3 Combining frame translations

The translation between two single frames can be detected using phase correlation. To get the motion over time for all frames, the detected translations are combined for which there are several methods. In all of them, the result is a single vector with the position relative to an arbitrarily chosen reference frame.

Accumulated motion track

The basic method is to compare each frame to the next one and accumulate the detected motions. This method starts with the first frame as a reference, and the translation between each frame is accumulated for each step. The final frame position is thus accumulated from all previously detected translations.

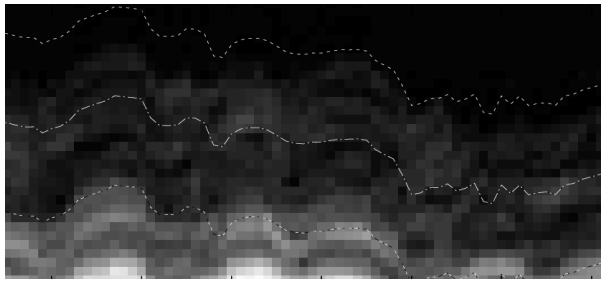


Figure 22: *Result from motion tracking*

With this method the errors are also accumulated. Sometimes there is a “jump” between single frames as result of faulty translations being detected.

Overdetermined systems

By comparing the translation between more frames than the neighboring ones as in basic matching, errors from bad matches can be reduced. This gives us an overdetermined system that can be solved in different ways. We choose to compare every frame to the next five frames. We found that comparing to more than five neighbors did not improve the result.

Given this extra data two methods with different goals were tried, least squares solving and an frame following method.

Least squares

A system of equations having *n* unknowns was constructed, the unknowns being the relative position of all frames where the first one is set to zero as a reference. Then, one equation for each frame matching is added. The system of linear equations is combined into a matrix equation, where A is constructed from the frames to compare , b is the detected translation between the frames and x represent the unknown positions.

$$Ax = b$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & . & . \\ -1 & 1 & 0 & 0 & . & . \\ -1 & 0 & 1 & 0 & . & . \\ 0 & -1 & 1 & 0 & . & . \\ 0 & -1 & 0 & 1 & . & . \\ 0 & 0 & -1 & 1 & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ . \\ . \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \\ . \\ . \end{bmatrix}$$

In A and b every row comes from one phase correlation measurement. The first one, being an exception is used to set the reference position of the first frame to zero($x_0 = 0$). Each column in A represent one frame from the complete video sequence. For example, in the second row we have the equation: $x_1 - x_0 = 0.1$. This comes from one phase correlation measurement between the first and second frame for which the difference was measured to 0.1. In the third row the first frame was compared to the third frame. In this example we are comparing each frame to the two next (and thus also two previous) frames.

For the equation to be solvable, at least n-1 comparisons are needed with a relation between all frames. Usually as in basic matching the neighboring frames are compared. As more frames are compared such as the second, next we get an overdetermined system that cannot be solved exactly because of errors in the input. Therefore, it is solved using the least squares method that calculate the positions, **x**, so that the sum squared difference between the input and the solution is minimized.

With this method the translation detection between all frames is done before any calculations. Therefore, frames can be compared to both previous and later frames in time. A problem occurs if the area of interest is moving more than a fraction of the window being used for the frame matching. There is also a risk of losing the motion track, especially if most of the tracking area is looking into the lumen where there is no pattern to follow. An attempt to handle this is done with the method we call frame following.

Frame following

Rapid or large motions will result in translations up to half the tracking window, which greatly reduces the accuracy of the translation detection. The frame following method tries to keep this difference as small as possible by using previously known detected motions.

This method has two components. First the following method, that use the previously detected motions to place the window for the next frame matching. Second, the next frame is compared to each of the previous 5 frames.

The absolute position derived from each match is calculated. From all these positions an average position of the corresponding frame is calculated. This process is repeated for every frame.

One risk is that the accumulated trail can travel outside the image. This occur if multiple subsequent frames falsely detect large motions in one single direction.

4.4 Mesh combination

The output from the methods in Section 4.3 is several trails representing the motion along a single point in the initial frame. These trails have been detected using evenly distributed starting points along the edge of the initial frame, seen as crosses in Figure 23.

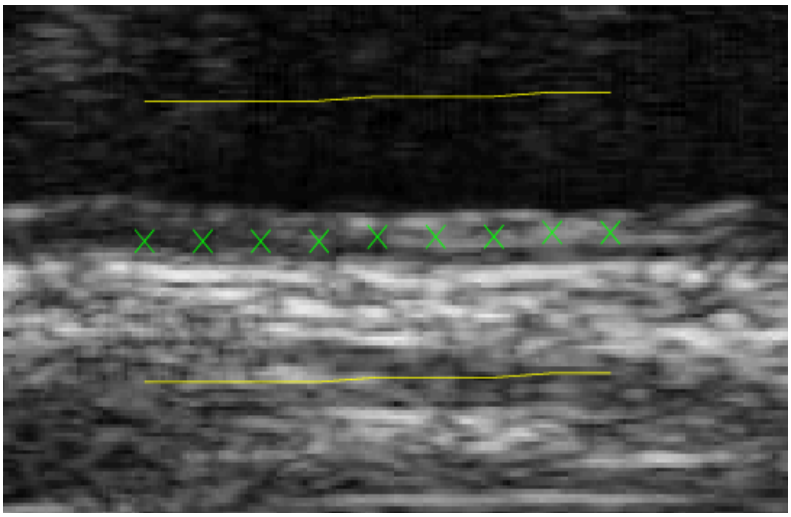


Figure 23: *Starting points of trails*

The data of these trails are combined into a single mesh, Figure 24 , which is a representation of the wall position over time. The data is stored in a 2-dimensional matrix with time/frame along one dimension and the x-dimension of the original image along the other. The z-axis represents the relative position of the motion in the y-direction of the video.

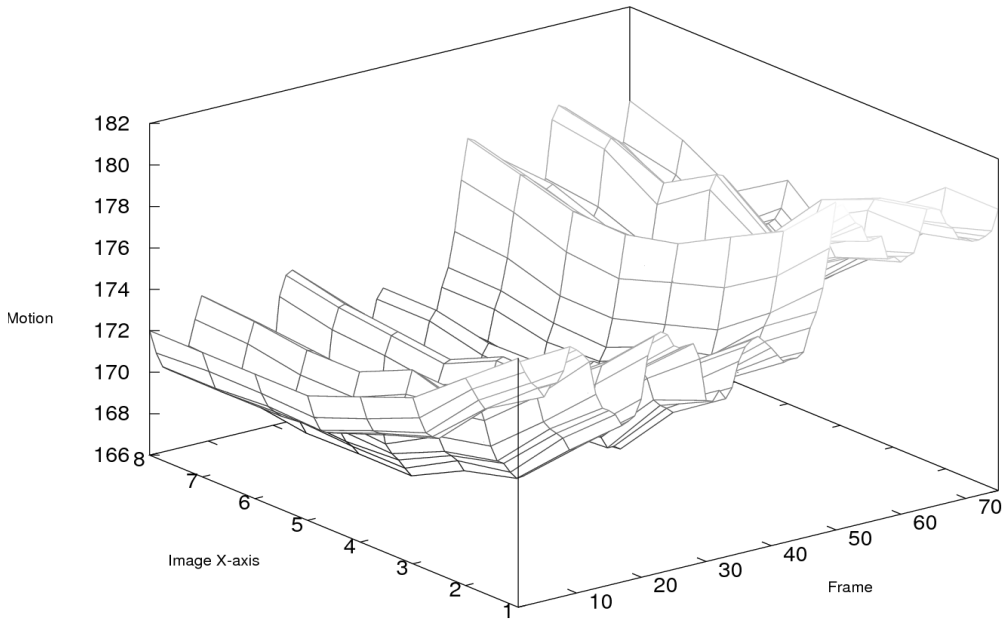


Figure 24: Mesh of wall motion in the video

4.5 Extracting the area of interest

The motion tracking errors was found to increase over time because of jumps in the detected shifts that did not correlate with the wall movements. This happened mostly around the time of heartbeats giving rapid motions combined with blurry images.

Along the wall, some areas were more prone to result in errors than others. We want to avoid these areas as well.

From the video a region of interest was extracted, avoiding both errors localized in time and space. This region was chosen manually since other attempts to automatically detect these errors failed. A typical size of this region was 1 s, 25 frames, and spatially of over 4 mm along the wall.

4.6 Noise filtering

We want to apply a low-pass filtering onto these data to reduce the noise in our motion mesh.

As can be seen in Figure 24, the motion is often rapid compared to the errors. We observe that the shape along the x-axis is constant over time. The motion of the wall is mainly caused by the pressure in the artery. Our observation of the motion indicates that the pressure can be considered constant over the entire imaged artery. We also conclude that any spatial difference in pressure would be smooth and therefore not affected by the following filtering method.

Using a low-pass filtering applied to this data would wrongly reduce this rapid motion. Therefore, we first extracted these motion features from the data set before applying the low-pass filtering.

The line at the first frame is set to all zero, since the motion data only has relative information. The average position for each frame in time is calculated. The average is subtracted from the data with only the local variations left, as seen in Figure 25.

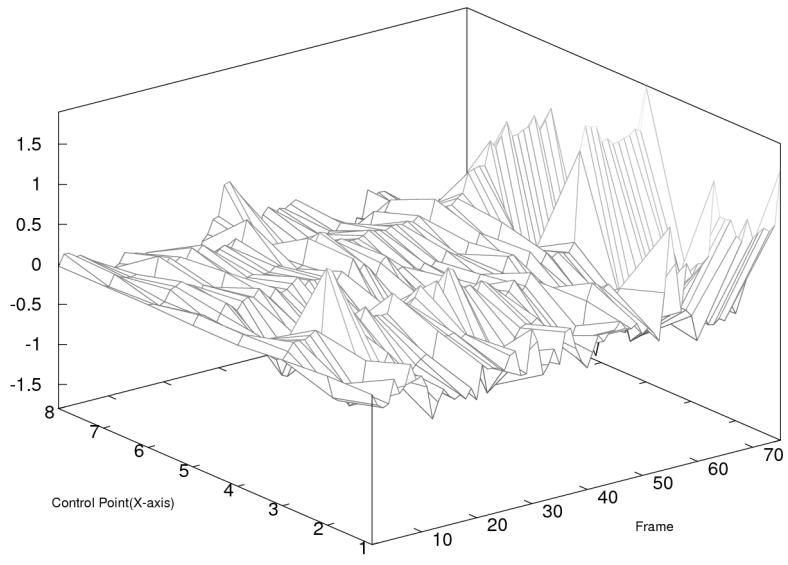


Figure 25: *Motion with larger features extracted*

A low-pass filter is then applied to this data using a Gaussian kernel.

Finally the averaged data, previously subtracted, is added back onto the filtered data giving us the now smoother motion. The straightening of the reference line is kept, as the data lost there comes from the manual user input, which is of no interest.

5 Static video generation

Using the detected motion mesh and the original video, a new compensated video was extracted. The purpose was to get a video where the wall is motionless. Differences between frames will then be the results of other defects in the ultrasound imaging than the motion of the subject.

The method of generating this video, which we call frame stacking, will be described in the next section. The remaining sections will then describe how we used this data.

5.1 Frame Stacking

A reference frame was first chosen, usually in the middle of the time axis. The motion data was then used to calculate an extraction map for all the other frames so that the subject of all frames would be located in the same position as in the reference frame. Using these maps, each frame from the original video was extracted into a new image. Since the motion detection has sub-pixel accuracy, the images had to be interpolated to get the exact position.

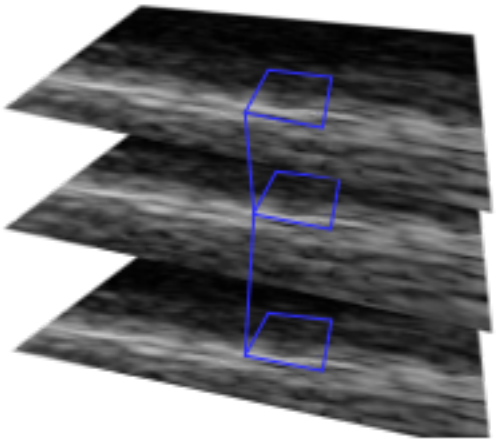


Figure 26: *Original video with detected translation between frames*

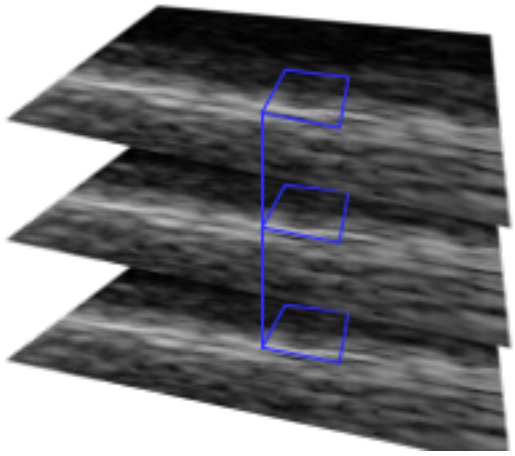


Figure 27: *Stacked video with frame translation compensated*

The method is not specific for the resolution of the input image, therefore, it is possible to extract each frame onto an image with higher resolution.

Our final data is a video sequence of the arterial wall appearing motionless over time.

5.2 Merging video

By merging the new video sequence along the time axis we get a single image of the arterial wall. Using various techniques for merging, we get different images. In all our examples, the merging has been done on a pixel basis in the stacked video sequence.

Mean is given by calculating the average over time for each pixel. This is the typical noise reduction method.

Filtered 10th/90th percentile(min, max) was generated by taking the 10% from the min/max thereby removing extreme values.

5.3 Test for classification

From the merged images we want to test for a difference between the layers in the arterial wall that would make them easier to distinguish from one another. We made a test to see if the data could be used with a classification method to distinguish the layers.

From the stacked video we manually defined areas such as in Figure 28. Three areas were sampled: lumen near the wall, intima-media, and adventitia near the wall. Black areas are not sampled. We also cut out the pixels closest to the edge between the layers seen as the black stripes separating the white areas.

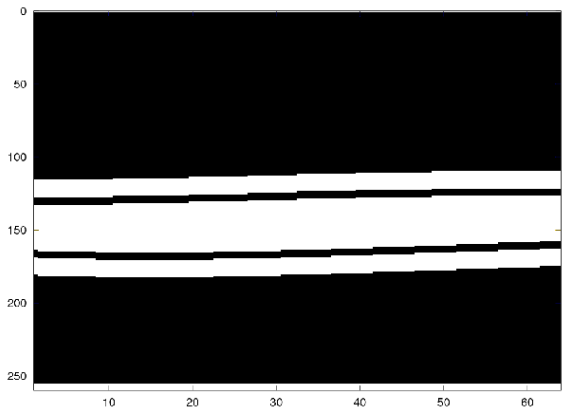


Figure 28: *Manual classification input*
White areas from the top: Lumen, intima-media, adventitia.
Black areas are not sampled.

By combining the manual classification map and the various merged images the intensity distribution in the lumen, intima-media and adventitia was plotted against the different image types.

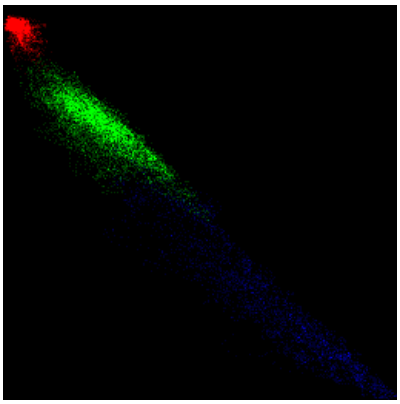


Figure 29: *Intensity distribution*
Single image (x-axis) vs maximum merge (y-axis) merged image.
In order from top-left to bottom-right(red, green, blue): lumen, intima+media, adventitia

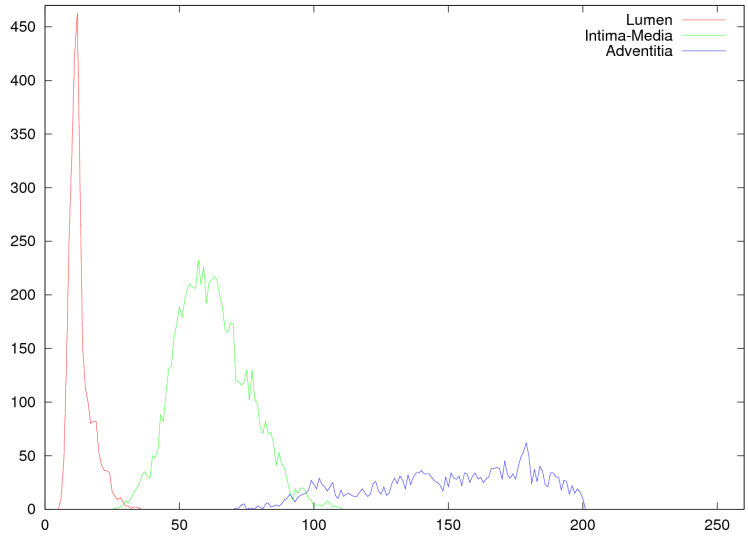


Figure 30: *Max-Intensity distribution of the layers.*
From left to right: lumen, intima+media, adventitia.

6 Results

6.1 Video acquisition

With our video acquisition software we were able to record the uncompressed video into permanent storage. The software was able to record full frames at full frame rate (25 fps), without any frames being dropped. The recording was done continuously to permanent storage.

6.2 Motion detection

During the development of our program we got more parameters to our method that affected the performance of the motion tracking. Some configuration alternatives did not have a conclusive best performing setting. For these we found that the configuration performing best was different on different sets of videos. Other parameters did have a natural option for better performance. These were intentionally set to the worst performing setting when experimenting with each new parameter. This was done to more easily detect the effect each new parameter had. Therefore, the errors presented here are not as noticeable in the final set of motion tracking results.

We applied our motion detection method to each of our four original videos. The result is shown in Figure 31-34.

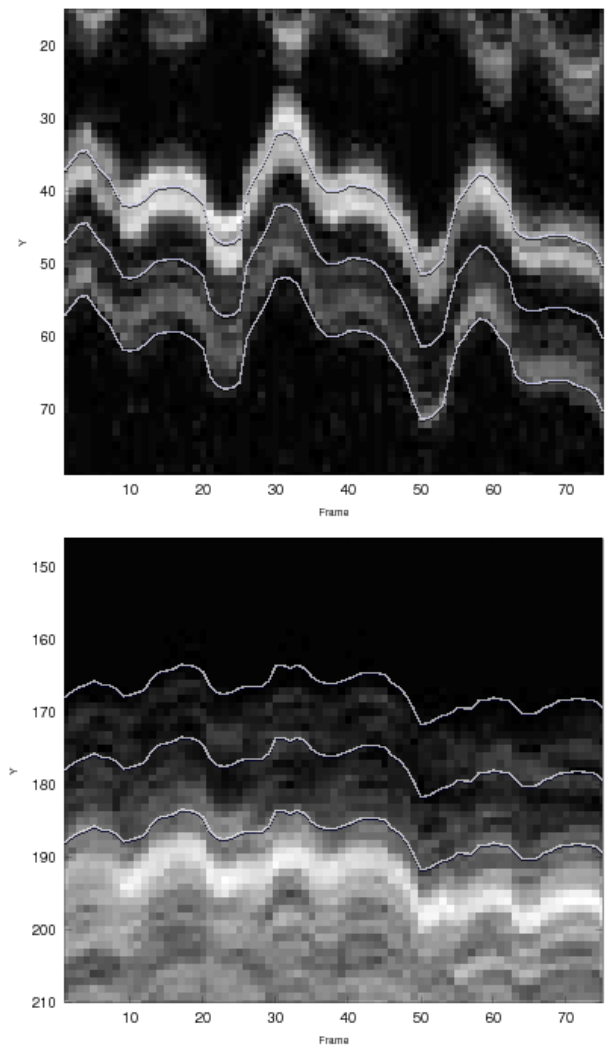


Figure 31: Motion tracking result, Sequence 1

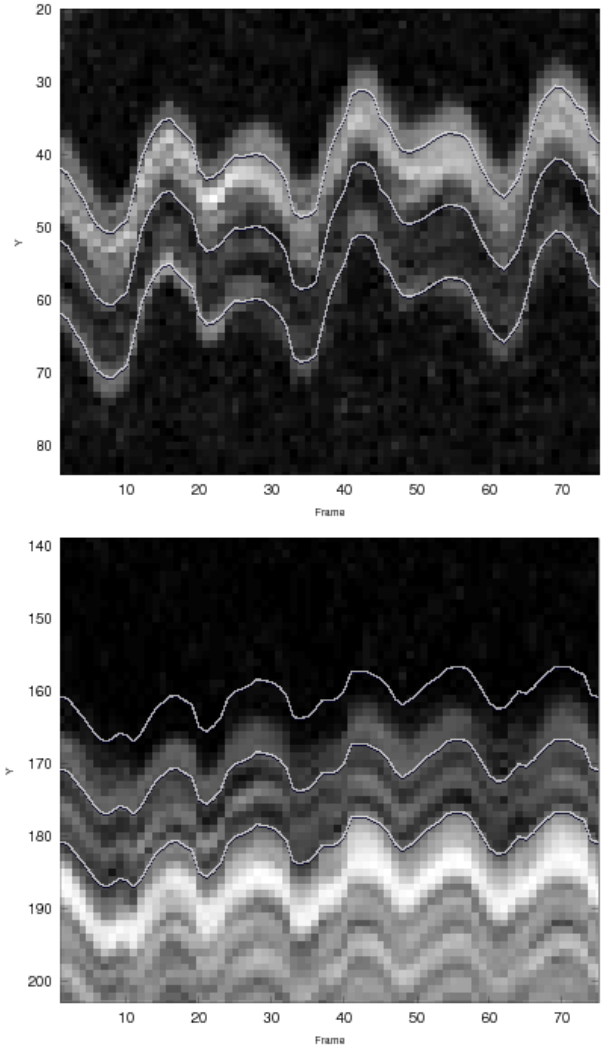


Figure 32: Motion tracking result, Sequence 2

Each figure represent one video. The top half of each figure is of the near wall and the bottom half is the far wall of the artery. The images are cut out along the y- and t-axis using a constant x. The result is similar to M-mode ultrasound images. The detected motions are shown as three parallel lines along each image. The detected motion is not related to any specific Y-coordinate, but only a relative motion. Therefore, we draw a few lines with equal y-distance over the image as an representation for the detected motion.

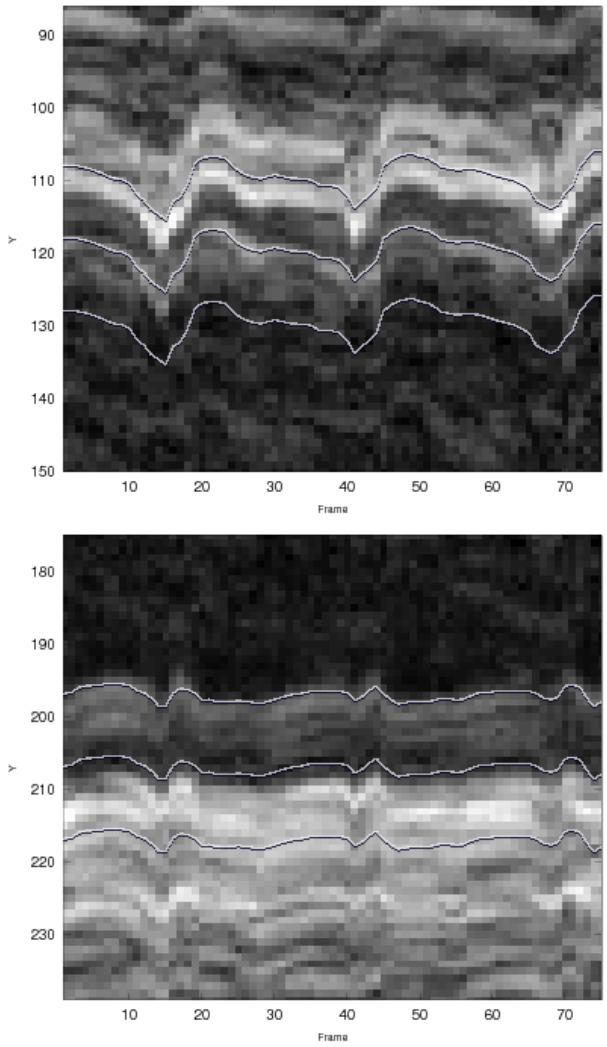


Figure 33: Motion tracking result, Sequence 3

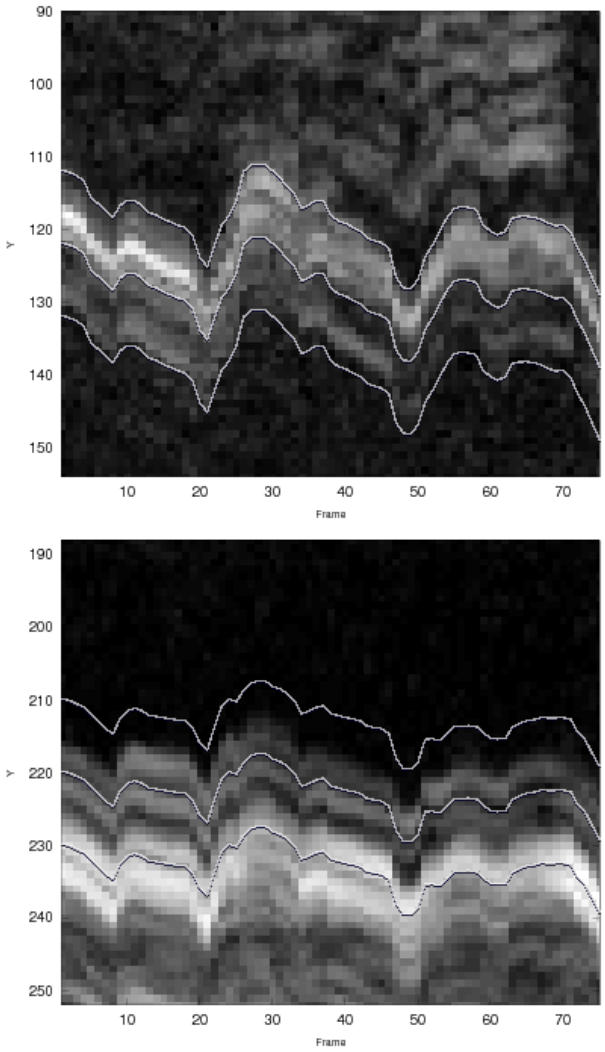


Figure 34: Motion tracking result, Sequence 4

Tracking window size

The motion tracking is done on a window with a predefined size during the whole process. The size we finally choose, 32 × 64 pixels, is drawn as the yellow rectangle in Figure 35.

Using larger windows the stability of the measurements increased when more information was involved in determine the motion. However, a too large window will include areas in the video moving differently, as in Figure 36, which introduces an error in the detected motion.

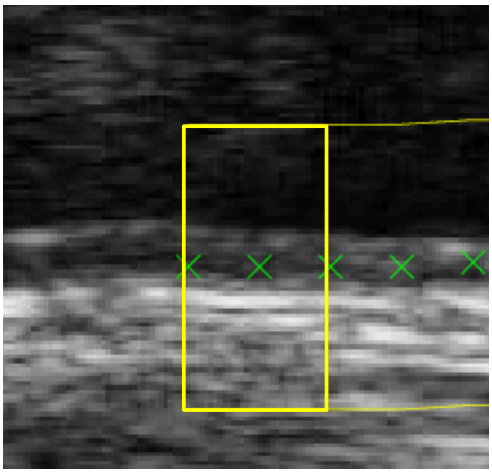


Figure 35: *Window Size*
X-Y image of an original frame

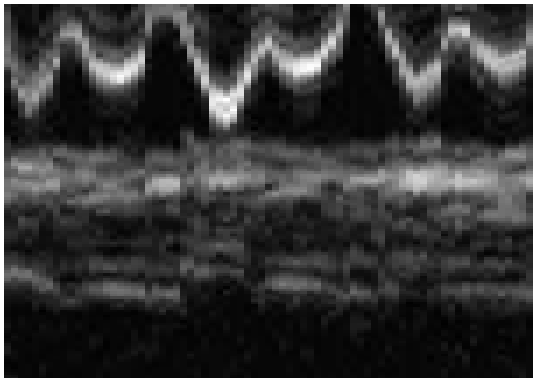


Figure 36: *Contradictory motions within single window*
T-Y image of motion compensated video

The area must also be large enough to fully catch the most rapid motions that can occur in the video, otherwise the tracking might guess wrong and continue following another area. This is shown as a smaller or larger jump in the detected motion.

Since a lot of computation is done on these data sets the time to run the program was largely affected by the size of the window.

True motions as high as 8 pixels between two frames were found. We finally found that a window of 32 × 64 gave the best results.

Number of neighbors to match against

Using larger number of comparisons with each frame increased the stability of the measurement up to about 5 frames, more than that did not increase the stability and only increased the computational work.

Frame follow and static position differences

Two methods were tried to determining the location of the windows for motion comparison. The frame following and the least squares method, which is using a static located window.

In Figure 37 we used the least squares method with a static window and in Figure 38 we applied the frame follow method onto the same data.

We found that with the frame follow method, there was a higher risk that several subsequent mismatches could introduce a false jump in the trail as seen in Figure 38 starting at frame 25. On the other hand, using a static window introduced a lot of smaller errors seen in Figure 37 as failure to follow the motion at frame 50 – 55, or the drift where the bottom line in Figure 37 lies, relative to the tissue, higher at frame 40 than frame 30. The rapid movements near the peak at frame 50 is also much better caught using the frame follow method in Figure 38.

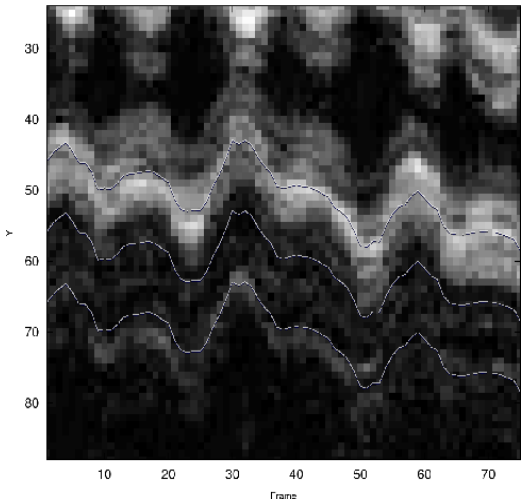


Figure 37: *Image A, Least squares, static window*

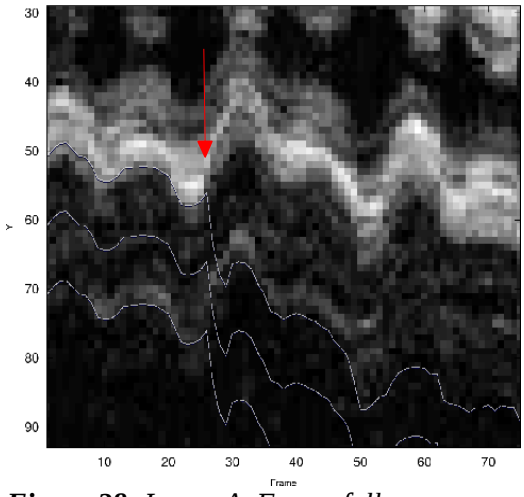


Figure 38: *Image A, Frame follow*
A few failed detections affect the following positions

In Figure 39 and 40 we do the same comparison using a new video. In this example we can see a situation where the static window method in Figure 39 at frame 47 begin an erroneous jump.

As in previous example we can see that the frame follow method has less transient errors, whereas we in Figure 39 can see several misses at frame 8, 25 and 35.

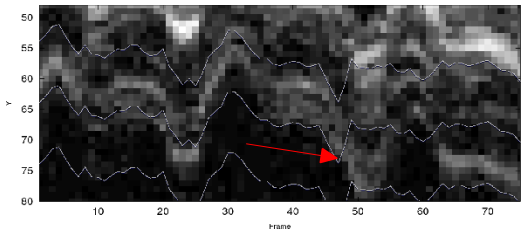


Figure 39: *Image B, Least square, static window*

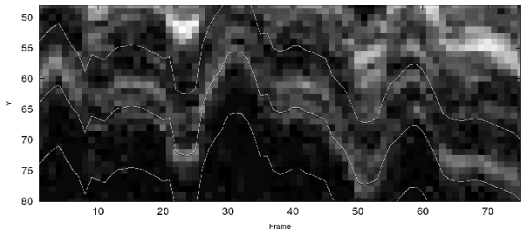


Figure 40: *Image B, Frame follow*

Our conclusion is that the frame matching, although it can match arbitrarily areas, perform better when the same area is used over time.

6.3 Frame stacking

The output of the frame stacking is a video in itself where the motion of the arterial wall is compensated by the detected motion. Figure 41 And 42 below show the result from our four videos with near and far wall from the same video is shown in the same column, although the x coordinates may differ. The window size used for frame matching is half of the images height. In each pair, the left is the T-Y slice of the original video and the right is the equivalent motion compensated version of the same slice.

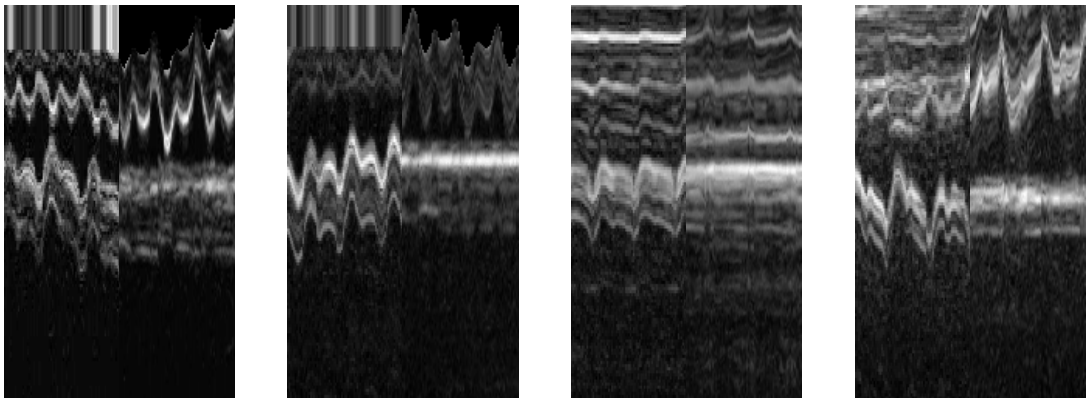


Figure 41: *Seq. 1-4 near wall*

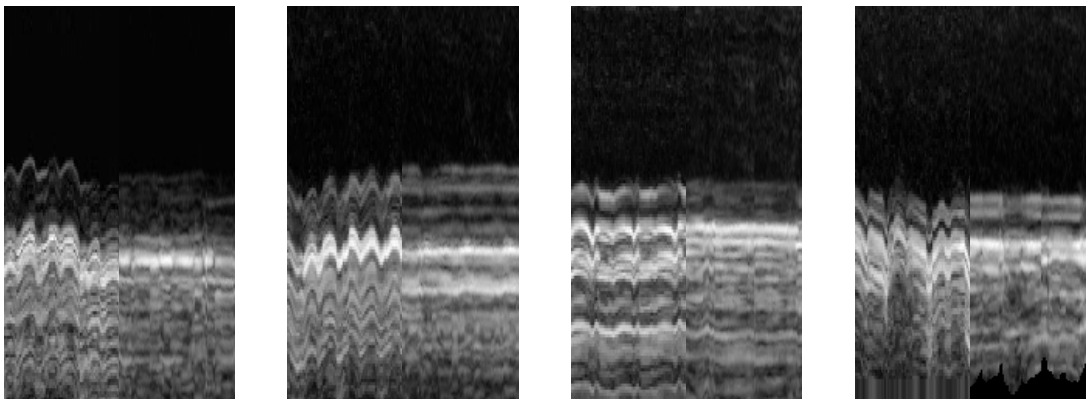


Figure 42: *Seq. 1-4 far wall*

The right image should therefore at best show a horizontal straight line.

The additional movements found in the near wall images (Figure 41), further away from the artery, has less or no effect on the frame matching. This is mostly because the movements are outside the frame.

These movements, when inside the frame would give rise to another peak in the translation probability map from the phase correlation method. Thus this different movement do cause errors when its motion is close to that of the wall. This can be seen in Figure 41, the fourth pair where there is additional noise in the motion compensated image at the frames where the motion of the other movements is close to zero.

6.4 Merging

From the motion compensated videos we first extracted a subset of each sequence to exclude erroneous motions. This was done both along the time and the x-axis. The following results will therefore have variations in their sizes. The near wall detections had usually smaller areas without errors, which gives the smaller merge images for sequence 2-4.

All subsets were then merged using the three different methods, minimum, mean and maximum.

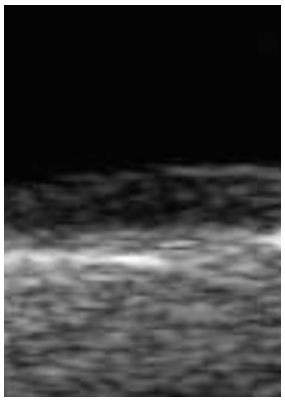


Figure 43: Original, single frame

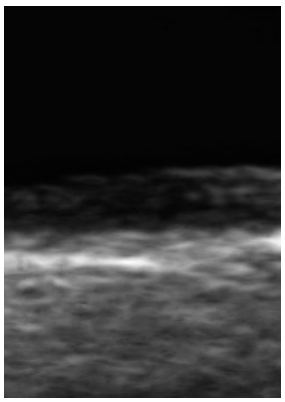


Figure 44: Merged, Minimum

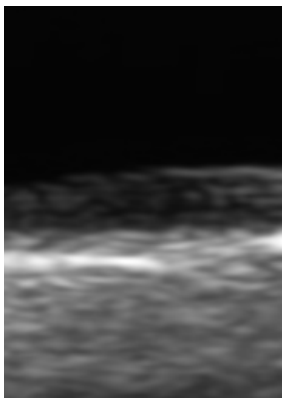


Figure 45: Merged, Mean

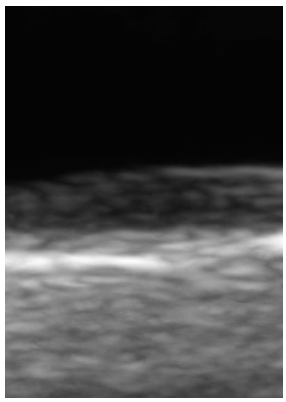


Figure 46: Merged, Maximum

Since all three methods cut out the extreme values, 10% for maximum, the lumen area is mostly darker in all results compared to a single frame. The media-adventitia layer is the darker stripe in the middle of the images.

In Figure 46, the maximum merge, the edge between lumen and the intima-media layer is more distinct than in the other cases.

In Figure 44, the minimum merge, there is a higher contrast between the intima-media layer and the adventitia layer, which otherwise is the most difficult edge to measure.

6.5 Intensity distribution

A typical intensity distribution for a single frame was compared to that of the three different merges.

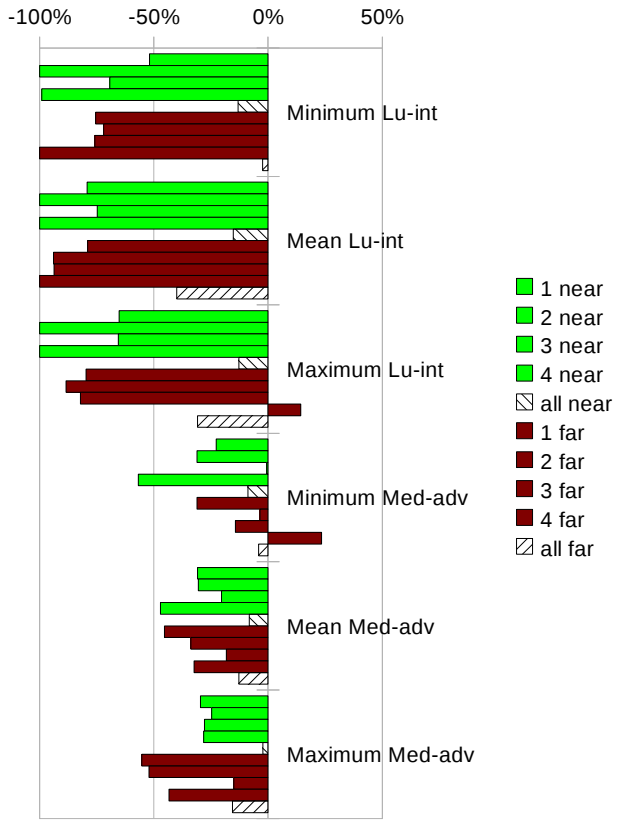
Using the manually generated mask of layer information we could extract the intensity for the various layers; lumen, intima-media and adventitia. Each layers intensity distribution was plotted in the same figure.

The overlapping of each area was calculated. The difference in overlapping between a single frame and each merging technique is shown in two representations. In the first chart in Figure 47, the value represents the percentage as compared to the single frame. Here a -100% means that there were no intensity distribution overlap for the merged version as compared to the single image. In the second chart in Figure 48 the value represents the percentage as compared to the entire intima-media layer.

Change in overlap

Figure 47 show the measured change in overlap for various merge methods compared to a single image.

In both Figure 47 and Figure 48, the bars are grouped by merge method(minimum, mean and maximum) and edge to separate(Lumen-intima and media-adventitia). For every group we have first four bars(light green) for measurement on the near wall for sequence 1-4 respectively. The next striped bar is the result when all four previously intensity distributions are combined before they are compared to the single images. After that we have the same order of measurements(dark red) for measurements on the far wall.

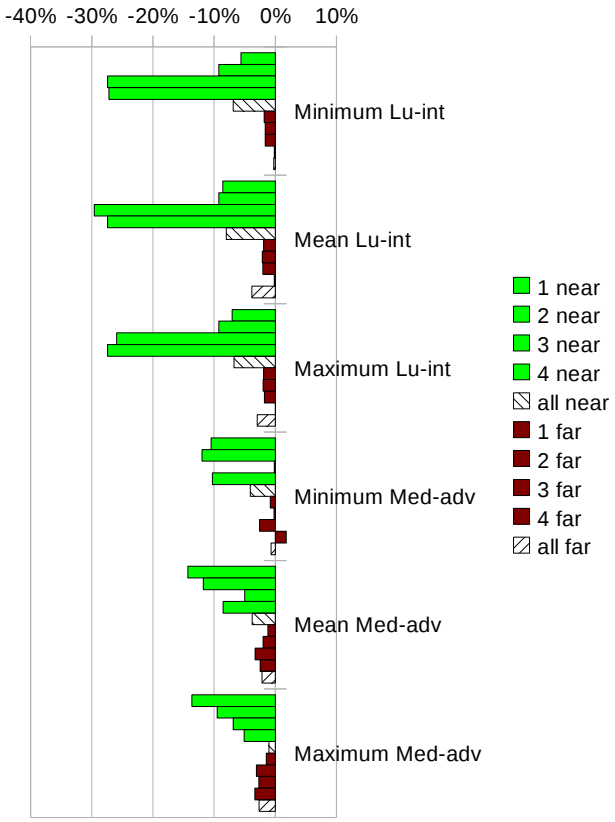


	Minimum Lu-int	Mean Lu-int	Maximum Lu-int	Minimum Med-adv	Mean Med-adv	Maximum Med-adv
1 near	-52%	-79%	-65%	-23%	-31%	-30%
2 near	-100%	-100%	-100%	-31%	-31%	-25%
3 near	-69%	-75%	-66%	-1%	-20%	-28%
4 near	-99%	-100%	-100%	-57%	-47%	-28%
all near	-13%	-15%	-13%	-9%	-8%	-2%
1 far	-76%	-79%	-80%	-31%	-45%	-55%
2 far	-72%	-94%	-88%	-3%	-34%	-52%
3 far	-76%	-94%	-82%	-14%	-18%	-15%
4 far	-100%	-100%	14%	23%	-32%	-43%
all far	-2%	-40%	-31%	-4%	-13%	-16%

Figure 47: *Relative change in overlap*

Change in overlap relative to intima-media layer size

Figure 48 comes from the same measurement as in Figure 47, but here we present the reduction in overlap compared to the intima-media layer size. A low reduction in Figure 48 combined with a high in Figure 47 would indicate that the overlap was small in the single frame to start with.



	Minimum Lu-int	Mean Lu-int	Maximum Lu-int	Minimum Med-adv	Mean Med-adv	Maximum Med-adv
1 near	-6%	-9%	-7%	-10%	-14%	-14%
2 near	-9%	-9%	-9%	-12%	-12%	-9%
3 near	-27%	-30%	-26%	0%	-5%	-7%
4 near	-27%	-27%	-27%	-10%	-9%	-5%
all near	-7%	-8%	-7%	-4%	-4%	-1%
1 far	-2%	-2%	-2%	-1%	-1%	-1%
2 far	-2%	-2%	-2%	0%	-2%	-3%
3 far	-2%	-2%	-2%	-3%	-3%	-3%
4 far	0%	0%	0%	2%	-2%	-3%
all far	0%	-4%	-3%	-1%	-2%	-3%

Figure 48: *Change in overlap relative to intima-media size*

Sequence 2, near wall

In Figure 49, we show the distribution for a near wall sequence. Lumen-intima differentiation is 100% where there is some overlapping for the single image, but none in any of min, mean or max.

The media-adventitia differentiation is not so good though. The numbers give a 20-30% less overlap, which at first sounds good, however, looking at the images, we see a total overlap of the intima-media and the adventitia distributions. We can, in all four plots, see that the adventitia distribution(rightmost, blue) fully overlaps the range of the intima-media(middle, green).

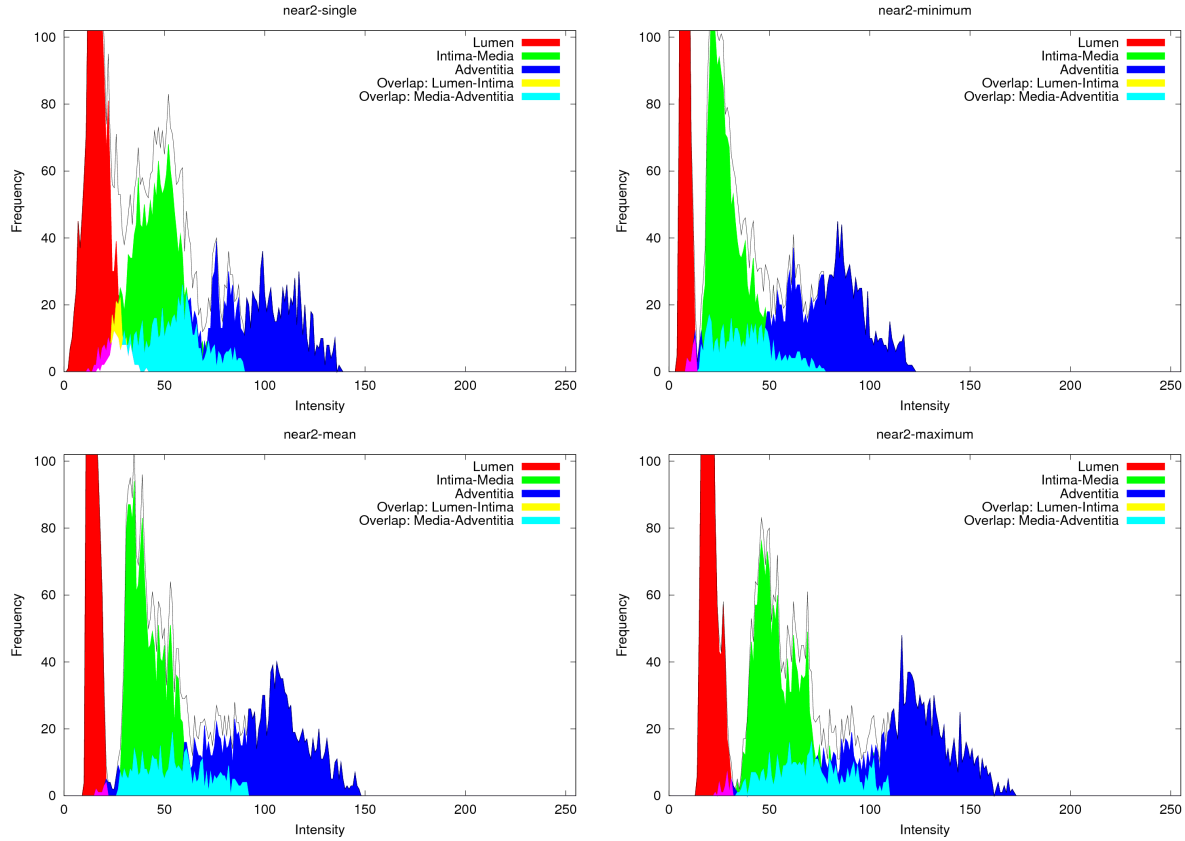


Figure 49: *Intensity distribution for sequence 2, near wall*

In Figure 50 we can see the same four images as represented in Figure 49.

The narrowed intensity distribution of the lumen is verified in all three merged images. This can bee seen as the lumen area being distinct darker.

The separation of intima-media and adventitia is also improved in all three merged images, and the best one being the minimum merge where the intima-media layer is distinct darker than the adventitia. In the images we find that the part of the intima-media layer closest to the adventitia layer represent the darker region of the entire intima-media intensity, thus the area close to the intima-media – adventitia edge can have a better separation, intensity wise, than compared to the entire intima-media area.

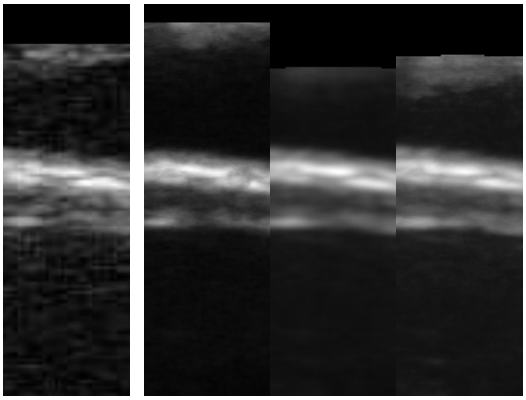


Figure 50: *Merged X-Y images.*
Original and merged min, mean and max

Sequence 3, near wall

The lumen-intima separation is slightly less successful here with a 70% less overlap.

The media-adventitia overlaps slightly less, 20-30%.

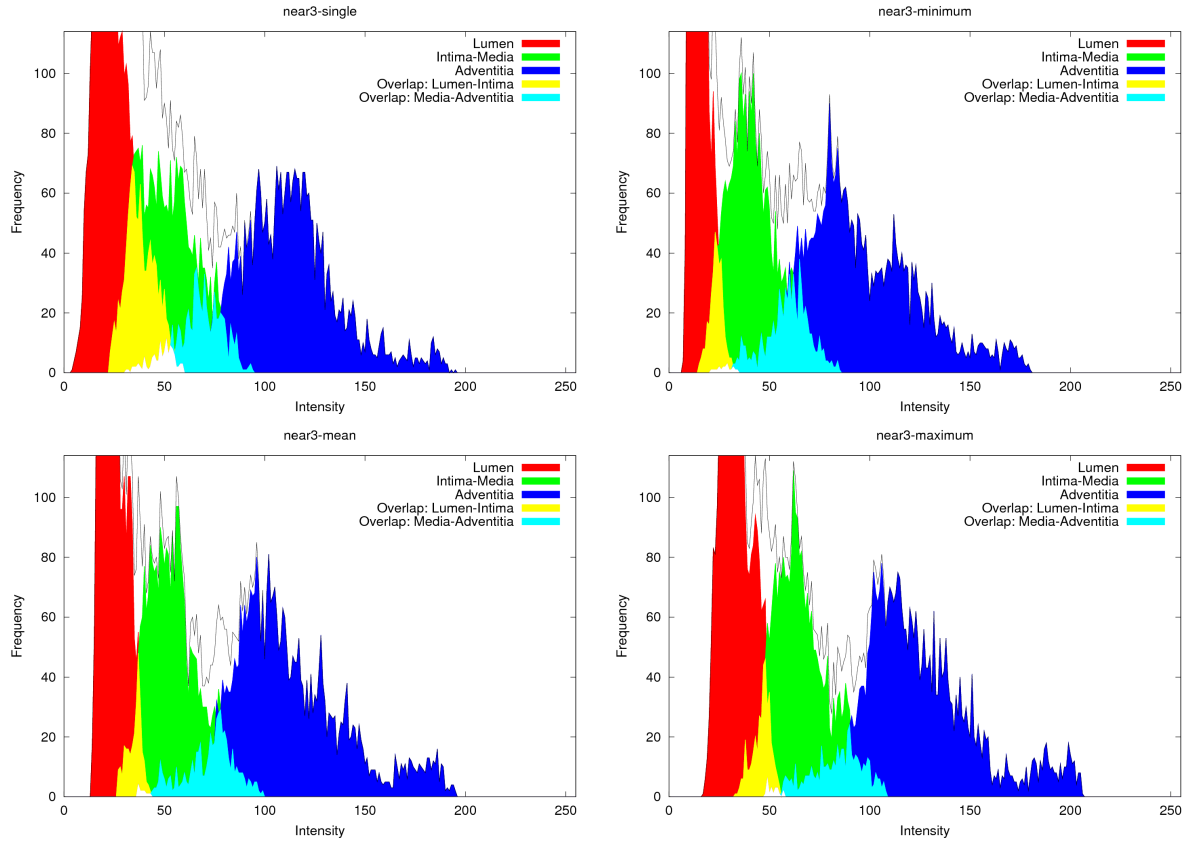


Figure 51: *Intensity distribution for sequence 3, near wall*

In Figure 52 there is a false echo effect in the bottom left part in all images. This affects the intensity distribution for lumen which now extends further into the brighter areas.

Visually we can see a better distinction between the intima-media and the adventitia layer in the min merge image. This image can make it easier for other edge detection methods to detect the true edge.

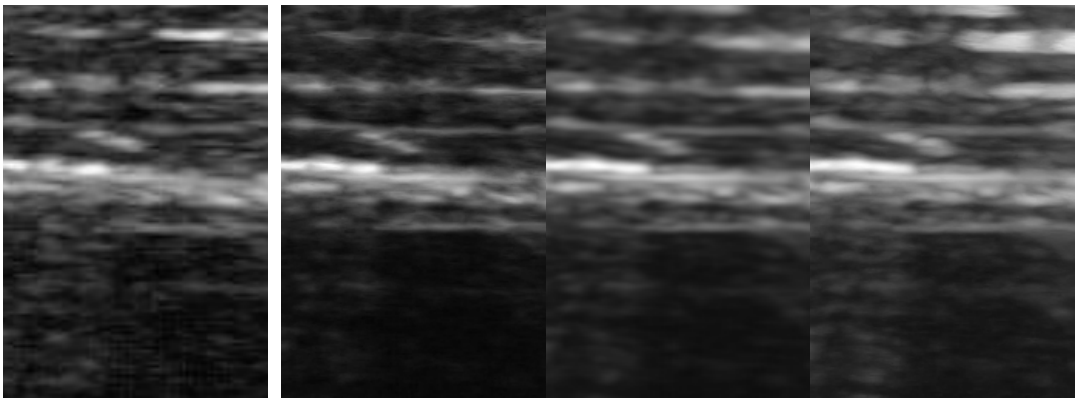


Figure 52: *Merged X-Y images*
Original and merged min, mean and max

Sequence 2, far wall

Here we have a small overlap in lumen-intima that again is almost separated.

For the media-adventitia we get less overlap, from a few percent using minimum merging to at most 50% reduction using the maximum merging.

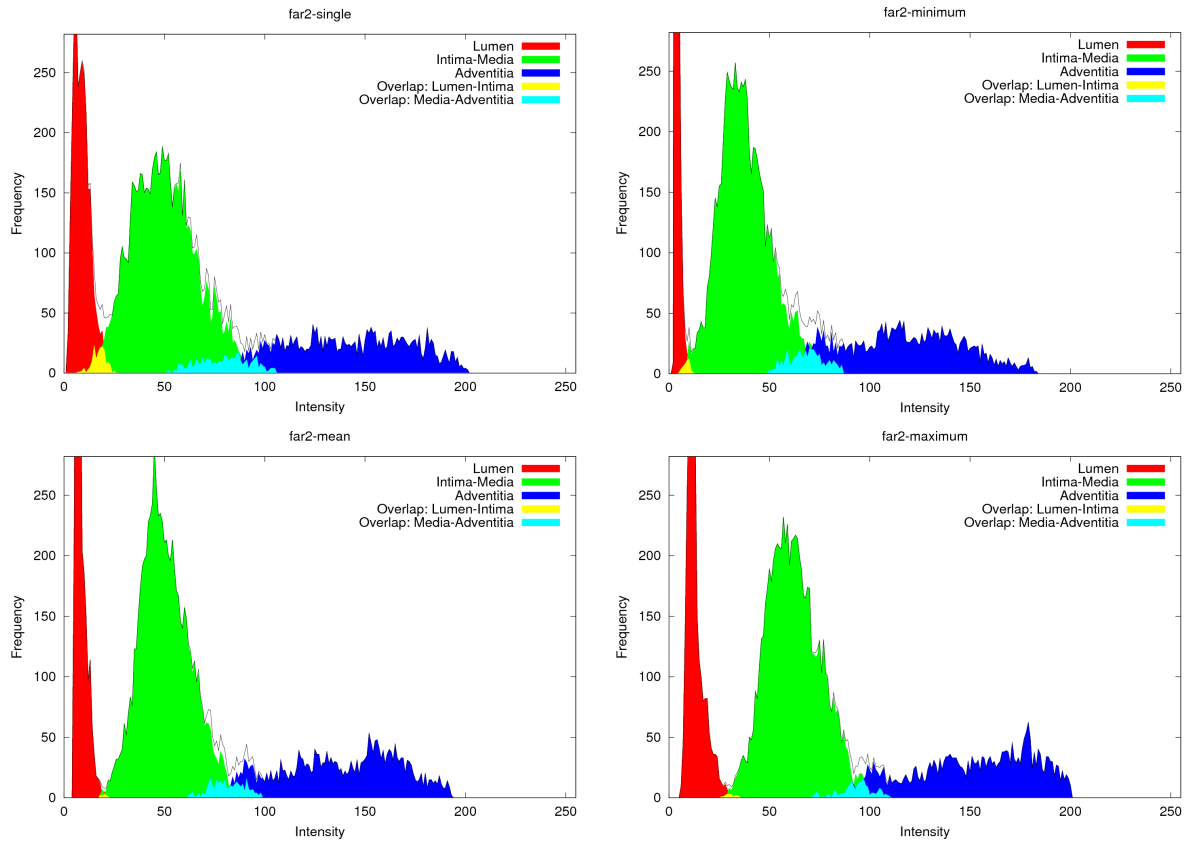
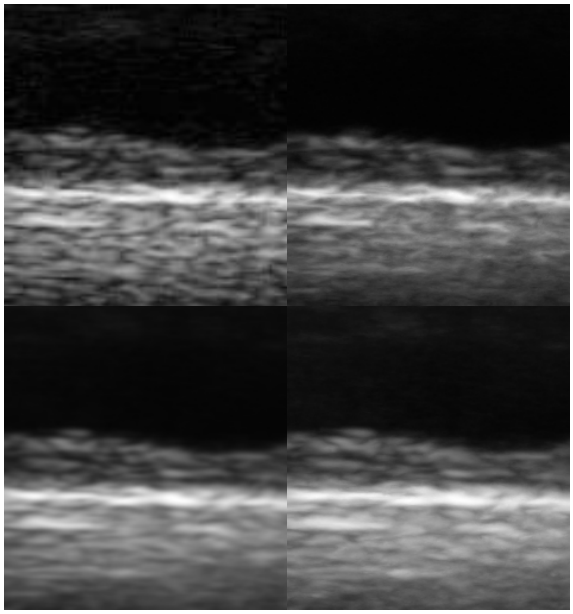


Figure 53: *Intensity distribution for sequence 2, far wall*



Sequence 3, far wall

Here we found the typical result with very good lumen-intima separation. The reduction in media-adventitia overlap is there, however, the overlap is still large.

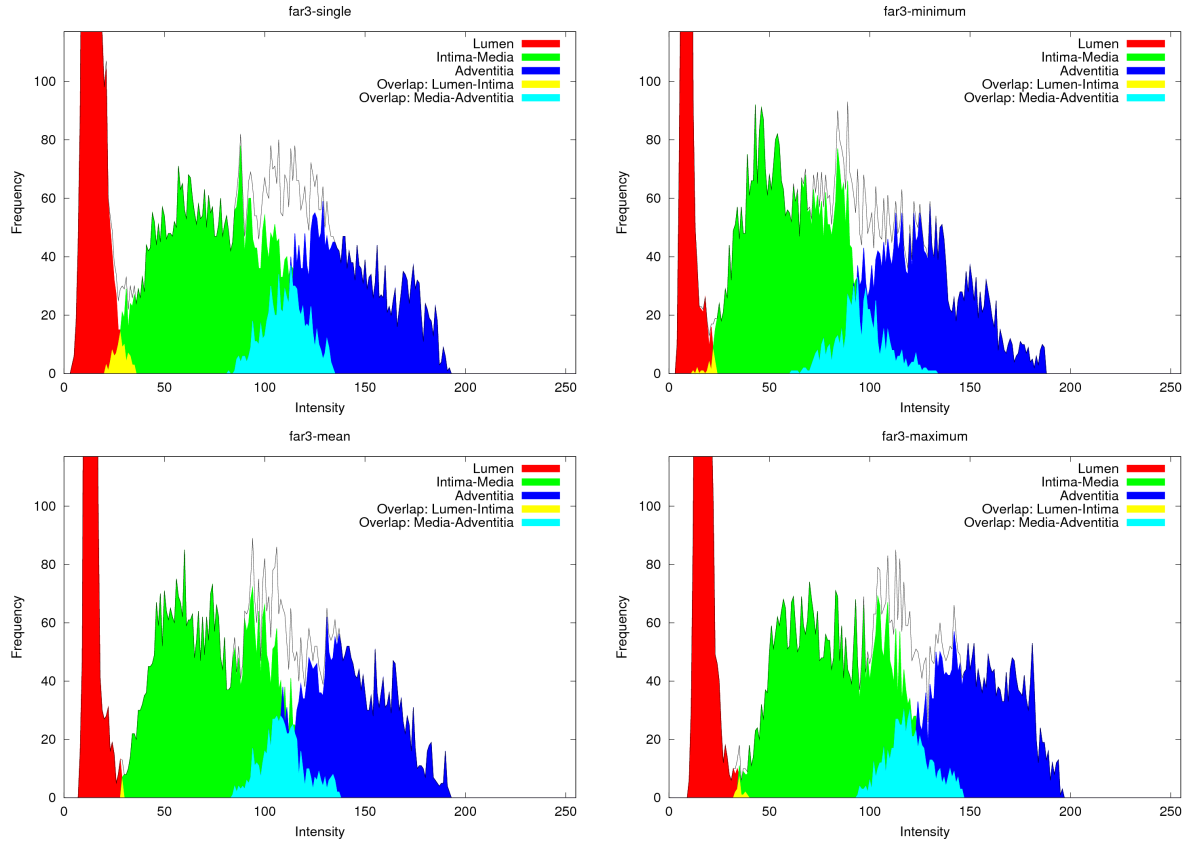
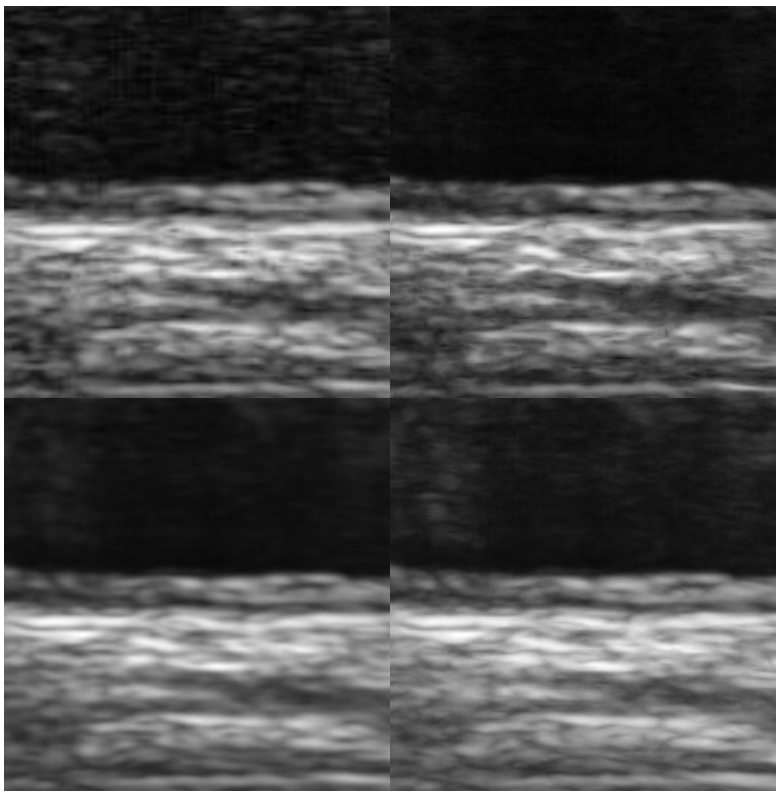


Figure 54: *Intensity distribution for sequence 3, far wall*



Sequence 4, far wall

Finally we look at the worst result according to the numbers.

The overlap in lumen-intima for the maximum merging are very few pixels, therefore, an insignificant difference appears as large, here, negative values.

For the media-adventitia overlap we got a worse result in the minimum merge method.

The mean and maximum did perform better separation between the layers than a single image.

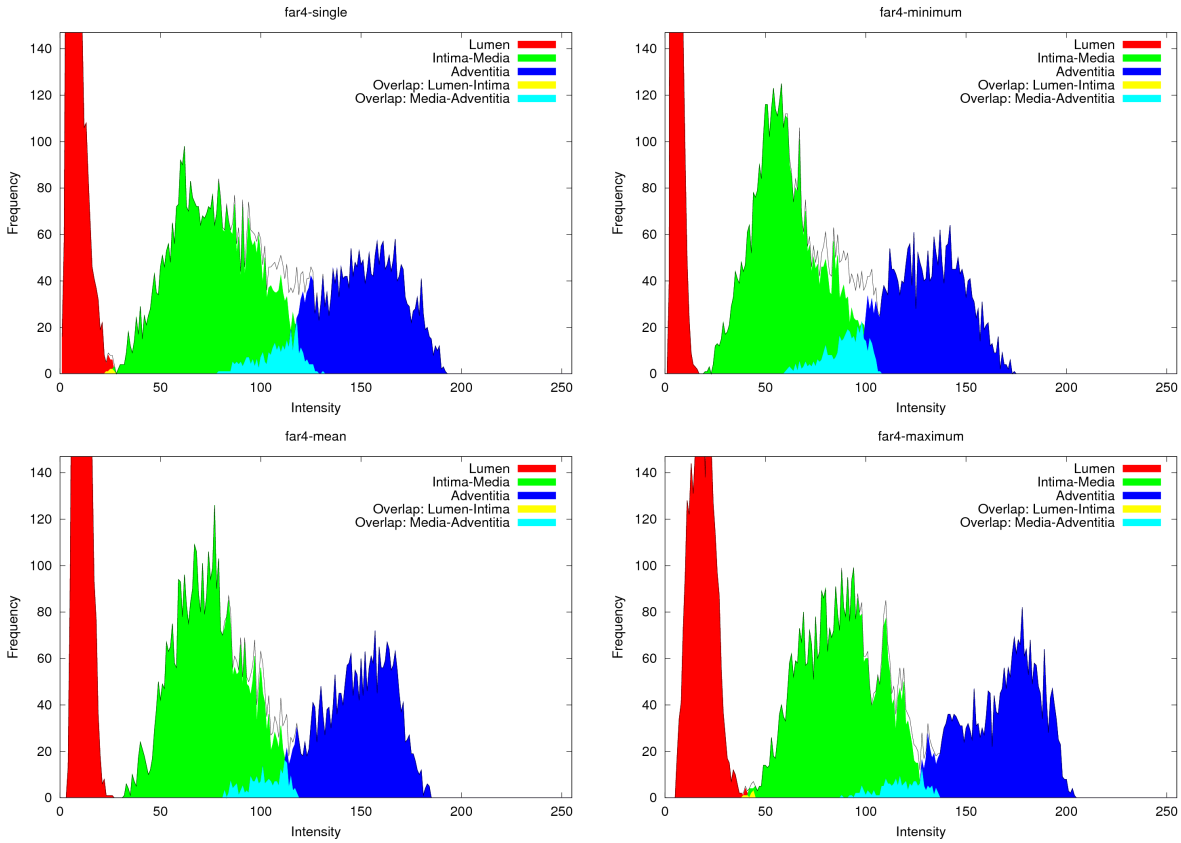
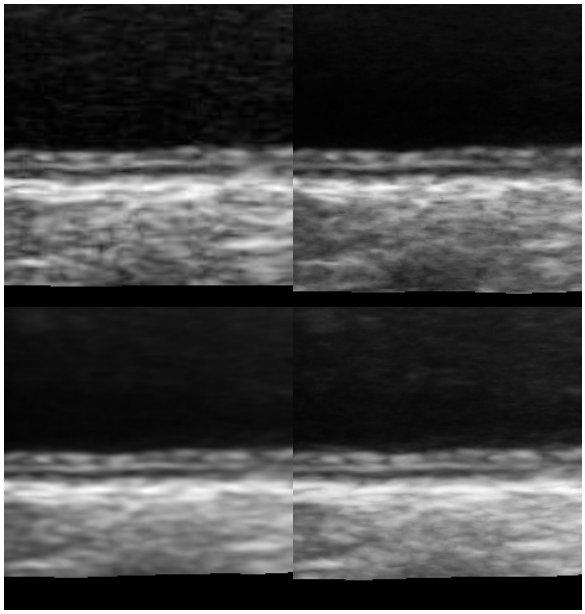


Figure 55: *Intensity distribution for sequence 4, far wall*



7 Conclusion

Our work started with an ultrasound video of the carotid artery. We wrote a software, that using a frame-grabber card could, on a regular computer, continuously record analog video onto disk. We developed a video processing method that, from the the original video, constructed a video of a static artery wall.

We used phase correlation to detect the motion and combined it into a model of the entire wall movement in the original video. This movement was used to generate the motion compensated video.

The final video was then merged along the time axis to generate new 2D images of the artery. Depending the the merging technique used we got images with various qualities that is usable when detecting the edges of the intima-media layer.

7.1 Video acquisition

The problem was to be able to continuously record full frame size and frame-rate video.

Using our program, specifically written for this purpose, we managed capture video with a margin even on a low end machine. Thus the compression was the limiting factor that prevented the shipped recording software to work.

7.2 Phase correlation

The method of phase correlation was implemented for detecting the arterial wall motion.

I was found that the errors in the motion detection data were either extremely small or very noticeable, which made it easy for manual selection of the useful area.

During the implementation of the motion detection there were one choice of configuration that did not have an obvious improvement. This was the combination of motion over an entire time frame. The two methods tried did both have their weaknesses and strengths. Our conclusion is that a combination of both of them can be a better choice. The best result was found when windows of the matching frames was located close to the same are of the artery wall.

7.3 Image stacking and merging

The video sequence generated got a visually appearance of motionless wall.

This video can be used in future area detection algorithms that then does not have to consider the otherwise rough motions.

The merged images can themselves directly be used in arbitrary 2D edge detection algorithms. From the averaging in the merging, we got images containing less noise. Since the stacking method work in with sub-pixel accuracy, the merged images got better quality compared to an ordinary up-scaling.

7.4 Intensity distribution

We got an improved separation of the lumen from the wall. Although this is generally an easy edge to detect there are circumstances, such as with very low echo intensity walls, where this improvement can be useful.

For our series of image there were almost no difference between the far and the near wall in the aspect of improvement in overlap.

The media-adventitia overlap was much harder to distinguish. We managed to reduce the overlap in the intensity distribution, but there is still such a large overlap so that intensity alone is not a possible indicator for edge separation.

Still we found the different merge techniques to have properties so that the layers could better be distinguished from each other, this was found by studying the intensity distribution. The maximum merge was found best at separating the lumen from the intima layer. The minimum merge technique was found best at separating the intima-media from the adventitia layer.

7.5 Implementation

Our method is a video to image processing method or, in a reduced form, a video processing method. Therefore it can be used as such in any existing IMT measurement tools that work with images or videos.

An implementation as a plug-in for existing tools would not require any further adaption.

So far there have been manual steps in the method where we have selected a smaller range for the final merge. This might not be practical in a final product where our method only will be a small part and thus cannot get this much attention. Improvements for making the method more automated is needed.

As we have seen evidence, for the different edges there are different merges that perform the best, the edge detection should not work on a single image. A somewhat modified implementation is needed to take this into consideration.

8 Future work

In this section we will comment on the possibility for improvement in three different areas. They are the phase correlation quality measurement, the choice of window to match and how the frame matching data is combined.

8.1 Detect and discard motion tracking defects

The most noise generating defects in the video samples are the heart beats. In our work we have therefore only extracted frames that had a good match in between two heartbeats. Since the video contains several heartbeats we are only using a fraction of the available data.

The methods developed in this work have the ability to match noncontinuous frames to one another such as the ones before and after a heartbeat.

At first the selection of frames to match can be done manually for testing. Later, however, there could be methods that can be used to automatically do this selection.

In the video there is also ECG information that could easily be extracted. This information indicates which frames are close to the heartbeats and its rapid motions. These frames would then be discarded before the main measurements start.

8.2 Phase correlation quality.

The translation probability map from the phase correlation method has so only been used to determine the position of the largest peak. Other qualities of this peak could possibly be an indicator on the accuracy of the match. They can be the height and shape of the peak and the noise level.

These and perhaps other properties of the map could be used to indicate the quality of the match. If a good indicator is found it can then be used to discard the worst measurements in the otherwise overdetermined systems determining the frame positions.

8.3 Iterative method

In section 4.3 we discussed two methods to combine an overdetermined system. The least squares where all frames were first matched, and the frame following method where each new frame is compared to the n previously ones.

So far the methods have only been applied in one iteration. A possible improvement is to modify or combine these methods using an iterative approach. There are two options.

I. An iteration using only the least squares method. The window for each frame is adjusted after each iteration to better fit the detected motion.

II. A two step method where the frame following method is used to determine the first placement of the windows. In a second iteration the least squares method is used to more accurately determine the final motion.

8.4 Using a rigid wall model

Our model allowed various points along the wall to be detected independently. This allowed them to accumulate errors on their own, which later made them unusable in combination with their neighbors.

By making an assumption of a non elastic property of the wall along the artery, the errors could be reduced. An even more rigid model where the wall edge is modeled as a rigid line could perhaps be of use on smaller segments along the wall.

This more rigid model could be used in merging the detected motion tracks into the single motion mesh for the whole video.

References

[1] Stein J. H. et al. (2008) Use of Carotid Ultrasound to Identify Subclinical Vascular Disease and Evaluate Cardiovascular Disease Risk: A Consensus Statement. American Society of Echocardiography Carotid Intima-Media Thickness Task Force Endorsed by the Society for Vascular Medicine

[2] Casella I. B. et al. (2008) A practical protocol to measure common carotid artery intima-media thickness. Division of Vascular Surgery, Hospital Regional Sul – São Paulo/SP, Brazil

[3] Stein J. H. et al. (2004) Carotid Intima-media Thickness And Vascular Age. American Society of Echocardiography

[4] Chodakauskas T, Feinstein S (2007) A non-invasive measurement of arterial wall atherosclerosis. European Hospital, Sept 1 2007

[5] Yamazaki Y. (2007 or earlier) The Usefulness of IMT Measurement Software in Ultrasound Systems. The First Department of Medicine, Faculty of Medicine, Osaka University

[6] William Scott Hoge (2003) A Subspace Identification Extension to the Phase Correlation Method. IEEE Trans. Medical Imaging, 22(2):277-280, Feb 2003

[7] Argyriou V., Vlachos T. (2006) A study of sub-pixel motion estimation using phase correlation. Centre for Vision, Speech and Signal Processing, University of Surrey

[8] David J. Fleet (1994) Disparity from Local Weighted Phase-Correlation. Department of Computing Science, Queen's University, Canada

[9] Haruhisa Shimoda, Fukue K. (2005) Experiments of Super-Resolution Utilizing Sub-Pixel Shifted Overlapping Images. Tokai University Research and Information Center

[10] Mahmoud S. A. (1991) Motion detection and estimation of multiple movingobjects in an image sequence using the cosinearea transform (CAT). IEE proceedings-I, Vol. 138, No. 5 , October 1991

[11] Young S. S., Driggers R. G. (2006) Superresolution image reconstruction from a sequence of aliased imagery. Applied Optics/ Vol. 45, No. 21 / 20 July 2006

[12] Takita K. et al. (2003) High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation. IEICE Trans. Fundamentals, Vol. E86-A, No. 8 August 2003