UiO : Department of Mathematics
University of Oslo

Automatic feature extraction and parameter estimation from unipennate muscles in B-mode ultrasound images

Sissel Fladby

Master's Thesis, Spring 2017

This master's thesis is submitted under the master's programme *Computational Science and Engineering*, with programme option *Computational Science*, at the Department of Mathematics, University of Oslo. The scope of the thesis is 60 credits.

Abstract

Muscle ultrasound imaging is a convenient technique to visualize normal and pathological muscle tissue. Studying muscle fascicle pennation angle and fascicle length and their dynamic changes during contraction are important measures in skeletal muscle studies. In this thesis, we propose an automatic method that detects the pennation angle and the fascicle length of the vastus lateralis and the gastrocnemius muscle from B-mode ultrasound images. The method is based on the Radon transform for detection of aponeuroses and fascicle orientation. The performance of the method was tested on two different data sets, and produces mostly good results when the aponeuroses are detected correctly. However, there are some difficulties with the aponeurosis detection for the images where the aponeuroses were less prominent.

Acknowledgements

This thesis has been a collaboration between the Department of Mathematics at the University of Oslo, the Department of Informatics at the University of Oslo and the Norwegian School of Sport Sciences. It was started in August 2016 and finished May 2017.

I would especially like to thank my supervisor at the Department of Informatics at the University of Oslo, Professor Anne H. Solberg. Her knowledge, advice, and helpful comments during the last year is immensely appreciated. I would also like to thank my supervisor at the Norwegian School of Sport Sciences, Professor Olivier Seynnes, for procuring data sets and measurements, and for his encouragement and helpful comments.

I want to thank my friends at the Department of Mathematics, and especially Lars Andreas and Paul, for your continued support and good times.

Finally, I would like to thank my family for their love, help and support. To my father, who has often been in my thoughts on this journey – you are missed.

Contents

CHAPTER 1

Introduction

1.1 Area of research

Image analysis is widely used on medical images with all sorts of usages in mind. In ultrasound imaging, we see many techniques used to contribute in diagnostics. Image analysis is used to provide better image quality, by for instance despeckling or denoising images, and it is used in differential diagnostics to discover subtle differences in healthy and unhealthy tissue. It can be used as a tool for an operator to locate areas of interest, especially in cases where there is a large quantity of images to look through.

In many cases when doing research, the problem is not that there is not enough data at hand, the problem is that there is not enough time to go through all of the data manually. If a computer can search through the data, and sort out the relevant information in a fraction of the time a human can, a lot of time and costs will be saved.

Muscle ultrasound imaging is a convenient technique to visualize normal and pathological muscle tissue as it is non-invasive and real-time, and studying muscle fascicles using ultrasound imaging can be applied in both diagnosis and rehabilitation assessment. Fascicles are usually detected and measured manually, which is subjective and time consuming, especially when there is a large number of images to be measured.

Because of this, a number of articles and experiments have been done to automatically measure muscle properties like pennation angle and fascicle length from ultrasound images, and it is a field in development.

1.2 Muscle architecture

Knowledge of the geometric arrangement of muscle fibers, i.e., muscle architecture, is important when studying muscle functions and the resultant

joint actions, and it has been shown that the muscle architecture considerably affects the manner in which muscle force is transmitted to the tendons and bones [Fuk+97]. Muscle fascicle pennation angle and fascicle length and their dynamic changes during muscle contraction have become important measures for skeletal muscle studies [Zho+12]. The fascicle length directly determines the excursion of the muscle and subsequently the velocity of shortening, and the pennation angle directly affects both the force production and the excursion [GD87].

In this thesis we use ultrasound images depicting the vastus lateralis muscle (VL) and the gastrocnemius muscle (GM). The vastus lateralis is located on the side of the thigh. This muscle is the largest of the quadriceps group, and the specific task of the vastus lateralis muscle is to extend the lower leg [Teab]. The gastrocnemius muscle is a muscle located on the back portion of the lower leg, being one of the two major muscles that make up the calf. The flexing of this muscle during walking and bending of the knee creates traction on the femur, pulling it toward the tibia in the lower leg and causing the knee to bend [Teaa].

A B-mode ultrasound image of the vastus lateralis muscle is shown in Figure 1.1. The pennation angle is defined as the angle between the deep aponeurosis and the muscle fascicles. The aponeuroses are layers of flat, broad tendons, and are marked on the image in Figure 1.2, and the fascicles are bundles of muscle fibers, one of them marked on Figure 1.3.

1.3 Scope of the thesis

In this thesis we aim to create an automatic method for estimation of fascicle length and pennation angle from individual B-mode ultrasound images of the vastus lateralis muscle and the gastrocnemius muscle.

As a first step in the algorithm, we will need to locate the region of interest in ultrasound images. The next step is to determine the locations of the aponeuroses, and then model them in such a way that we get a representation of the true shape of the aponeuroses. We will assume that both the aponeuroses are visible in the image, and that there is one aponeurosis in the top half, and one in the bottom half of the region of interest, respectively.

Further, we will need to estimate the orientation of the fascicles. This will be done with two intentions; we want to determine the pennation angle with regards to the deep aponeurosis, and secondly we want to create a reference fascicle determined by the detected orientation of the fascicles in the different parts of the image. The reference fascicle will need to follow

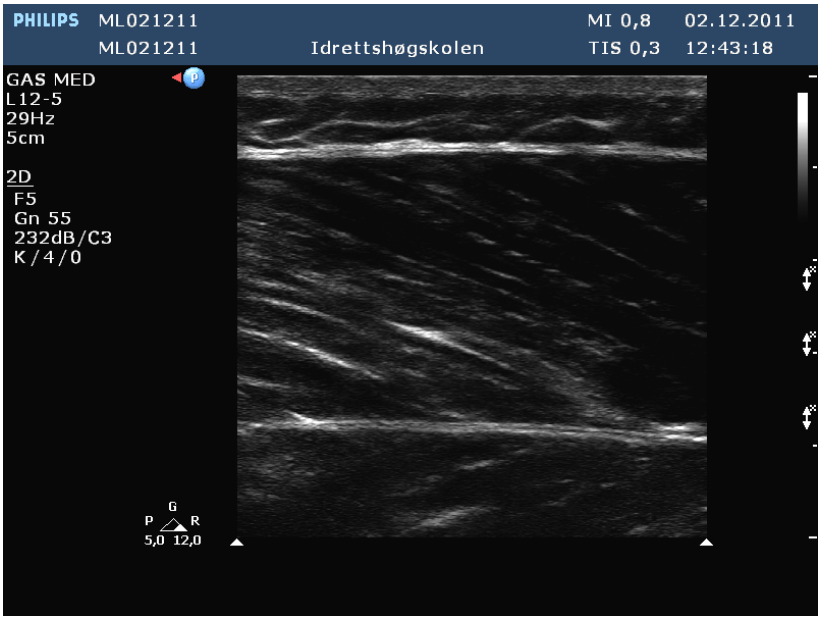


Figure 1.1: B-mode ultrasound image of the vastus lateralis muscle.

.

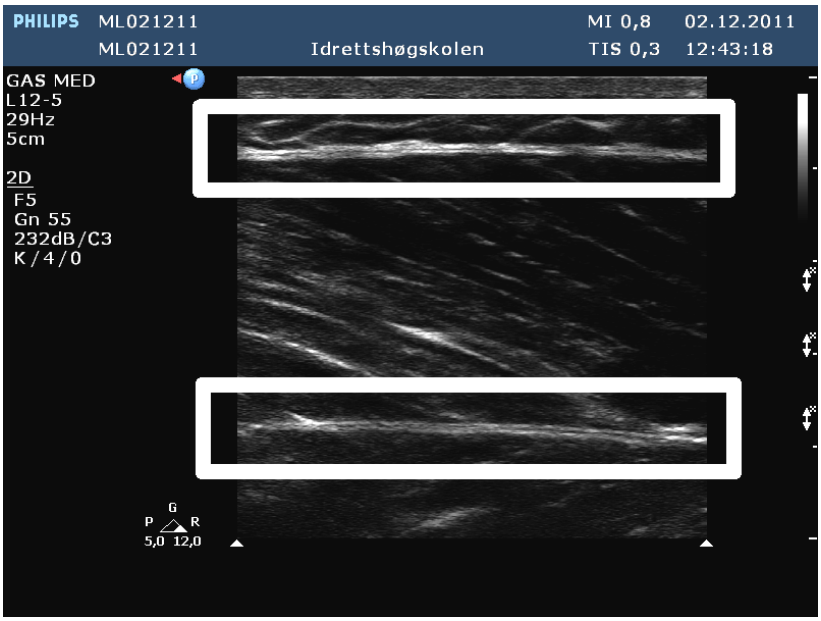


Figure 1.2: B-mode ultrasound image of the vastus lateralis muscle, with the deep and superficial aponeurosis marked.

.

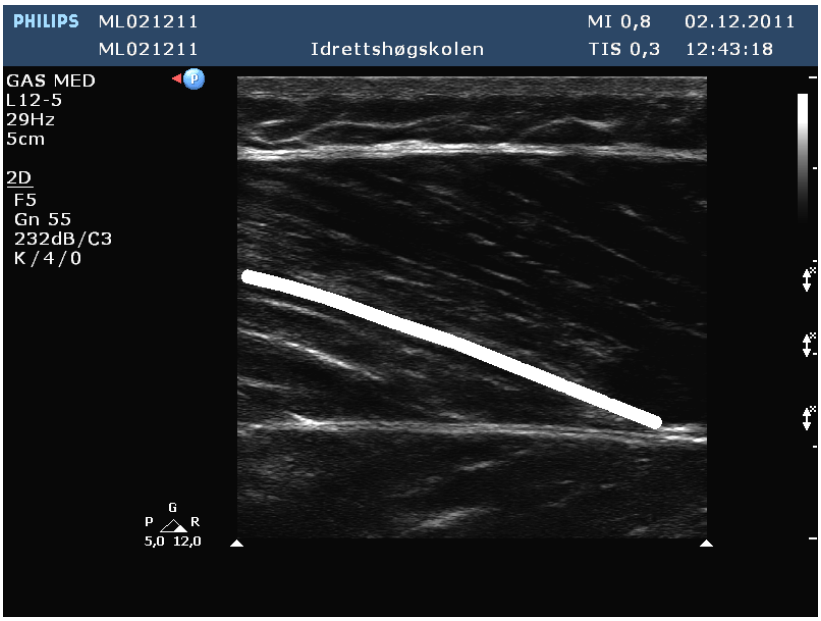


Figure 1.3: B-mode ultrasound image of the vastus lateralis muscle, with a fascicle marked.

.

the curvature of the fascicle, and a model that can reflect the shape of the fascicle should be used for this.

The fascicle length will be determined by the length of the fascicle from the deep aponeurosis to the superficial aponeurosis, to mimic how it is measured manually. In the cases the reference fascicle extend outside the field of view, we need to extrapolate the fascicle and the aponeuroses, and calculate where these two curves would meet to obtain the start point and the end point. In this step we need to assume that the extrapolated fascicles and aponeuroses actually can be transferred to the area outside field of view.

1.4 Structure of the thesis

- In chapter 2, we have a look at what has prevously been done of research that is relevant to the thesis, with a following discussion.
- In chapter 3 we have a look at the avaliable data sets, and discuss some of the challenges with it.
- In chapter 4 we explain what we aim to achieve with the region of interest algorithm, and then we proceed to explain how we develop

the algorithm for images with both a rectangular and parallelogram shaped region of interest. Finally we discuss the resulting algorithm.

- In chapter 5, we define the discrete Radon transform, and the discrete normalized Radon transform.
- In chapter 6, we discuss how to detect the aponeuroses and determine an algorithm for detecting the location of the aponeuroses. Further, we discuss representation of the aponeuroses using spline curves, and define splines and least squares approximation with splines. Finally, we evaluate the resulting implementation of the aponeuroses detection.
- In chapter 7, we discuss filtering of the fascicle plane, and how to detect the orientation in the fascicle plane. Further, we discuss construction of the reference fascicle, and modelling of this. Finally, we evaluate the result of the reference fascicle estimation.
- In chapter 8, we discuss determination of the fascicle length and pennation angle.
- In chapter 9, we discuss the results of the algorithm as a whole, testing the algorithm on test set 1 and test set 2.
- In chapter 10, we summarize the problem, discuss the main finding, and consider what further work can be done.

CHAPTER 2

Background

2.1 Previous research

There has already been done research on line detection in ultrasound images, and so we will have a look at some of the articles published in that area.

Feature extraction from ultrasound images of muscles

Zhou and Zheng [ZZ08] wrote in 2008 an article where they estimated fascicle orientation in ultrasound imaging by using revoting Hough transform. The new method first located the global maximum in the Hough transform (HT) accumulator matrix, which corresponded to the most dominant collinear feature points globally, using the standard HT; then the pixels close to the detected line were removed from the edge map, the HT accumulator matrix was calculated again, i.e., revoting, and a new line was detected. The results of both computer-generated images and clinical ultrasound images demonstrated that revoting HT could provide an approach for the orientation estimation for the lines in the ultrasound images. Further studies are required to achieve adaptive parameter selection for deciding the number of lines to be detected.

Rana, Hamarneh and Wakeling [RHW09] did a study aiming to develop an automated method to quantify the orientation of the fascicles within a muscle.

First they used multiscale vessel enhancement filtering to enhance fascicle structures developed by [Fra+98]. Then they compared two methods, a) Radon transform to quantify the dominant orientation in the image, and b) and ultrasound-spesific wavelet analysis quantified the local orientation around each pixel.

When the methods were applied to an actual ultrasound sequence, the Radon transform identified orientations with a closer match to the manual values than the wavelet analysis, the wavelet analysis resulted in angles that were 1.35° less than the manually digitized values.

Zhao and Zhang [ZZ11] wrote in 2011 an article where they proposed to use localized Radon transform to detect and track muscle fascicles in ultrasound images. They compare this to using Hough transform to track muscle fascicle. A revoting strategy such as in [ZZ08] is used in both algorithms.

The aponeuroses are thus extracted first by using the Radon transform, and then fascicles can be detected in the region between the aponeuroses. The aponeuroses are assumed to be straight lines, and so are the fascicles.

Using a *localized* Radon transform means they are only searching where one would expect to locate the fascicles, and limiting the angle range to the angles one would expect to detect fascicles. As they explain, this is possible because the fascicles must be located in betweeen the superficial and deep aponeuroses. It is also known that the pennation angle of fascicles in a muscle are within a certain range, even under "pathological conditions" - i.e abnormal muscles. Because of this, the Radon transform can be performed within a certain angle range smaller than the usual 180°.

Results show that the localized Radon transform outperforms both the revoting Hough transform using a Canny edge detector, and the Radon transform on the entire image.

They conclude that because the revoting Hough transform relies greatly on the performance of the edge detector chosen, and edge detectors rarely works well with the blurry line edges in both simulated and clinical ultrasound images, it is not as suited as the Radon transform. Not only does the Radon transform not require edge detection, its inherent integration feature is also less susceptible to background noise.

When measuring the fascicle length, the algorithm measured the length of the straight lines located as fascicles, and images with curved fascicles were not studied in the paper.

Zhou, Chan and Zheng [ZZ15] wrote in 2015 an article about automatic measurement of pennation angle and fascicle length of gastrocnemius muscles. They preprocessed the image by convolving a Gabor wavelet to enhance line-like structures in the image. Then, using the normalized Radon transform they detected the position and orientation of the aponeuroses. The fascicle orientation was determined as the predominant orientation by using the maximum variance in the Hough transform. Finally, they extracted the pennation angle as the angle between individual fascicle and

deep aponeuroses while the fascicle length was defined as the length along the fascicle from the superficial aponeurosis to the deep aponeurosis.

The results showed a good agreement between the automatic measurements and the traditional manual measurement.

Jalborg [Jal16] wrote in 2016 a master thesis where an algorithm was presented for estimating pennation angle and fascicle length in ultrasound images of skeletal muscle. In this thesis, the region of interest was detected automatically. Then, using the Radon transform to detect the orientation in the image, the approximate location of the aponeuroses is detected by looking for a change in the dominant angle. The accurate location is detected by doing the Radon transform on the Knutsson directional filtered images on the approximate locations of the aponeuroses. The aponeuroses were modelled as straight lines.

To find the dominant orientations in different parts of the image they performed a normalized Radon transform in local windows linearly spaced throughout the Knutsson filtered image, and the mean angle for each depth of the image was calculated. In order to create a reference fascicle, in each depth a line piece with the corresponding mean angle was constructed, the line pieces 'stitched' together, and a quadratic curve was fit to the data.

Further, the fascicle length was estimated by calculating the arc length of the quadratic curve, from the superficial to the deep aponeuroses. The reference fascicle and the aponeuroses was extrapolated outside the field of view if necessesary. The pennation angle was estimated by calculating the median angle of the fascicles located two thirds into the fascicle plane, relative to the deep aponeurosis.

In the thesis, it was pointed out that a straight line was not sufficient to represent the aponeuroses, and that a more robust manner of aponeuroses detection was necessary. The fascicle detection was promising, but they pointed out that it might be worth looking into using the median angle instead of the mean angle.

Estimation of the fascicle length was not evaluated because of lack of data, estimation of the pennation angle gave mixed results, mainly because the deep aponeurosis detection failed in many of the images.

Because the speckle noise on the ultrasound images is simular to that in synthetic aperture radar (SAR) images, some research on line detection in SAR images is also interesting to have a look at when determining methods for line detection in ultrasound images.

Line detection in SAR images

Because the speckle noise on the ultrasound images is simular to that in synthetic aperture radar (SAR) images, some research on line detection in SAR images is also interesting to have a look at when determining methods for line detection in ultrasound images.

Hellwich, Mayer and Winkler [HMW96] proposed to use a Markov random field (MRF) and Bayesian classification to detect lines in SAR-images. These images are prone to have a high amount of speckle noise, which is also a challenge in ultrasound imaging

Using Bayes' theorem, assuming the data to be MRF, and using the equivalence of MRF and neighborhood Gibbs fields, they were able to express the posterior probability density of the object parameters given the scene data in terms of energies, and given the independence of the intensity and coherence of the image data.

The results after 25 iterations of simulated annealing were not completely satisifying, and they suggest that more work should be done on the algorithm.

Wei and Feng [WF16] proposed a line detection method called Image Edge Field Accumulation (IEFA) that extracts line features from the image edge fields of SAR-images.

Inspired by Hough Transform, they replace the standard binary edge map with a continous edge field. This way, they can keep both direction and strength information about each pixel. This means that an edge pixel can have different accumulation weights for the lines that pass through it.

Compared to using Hough Transform on a binary edge map constructed from the GGS biwindow edge detector and the Canny edge detector, the results are quite good. The algorithm has strong antinoise ability and good antiocclusion ability. They conclude that the IEFA is applicable to line detection and runway recognition, but that the computational complexity depends on the degree of complexity of the image, and that the method must be further improved for a wider range of applications.

Discussion

Not completely convinced that the methods we looked at on line detection in SAR-images were completely applicable in this thesis, we think it might be more fruitful to continue development on some of the results from the research on feature extraction from ultrasound images. In both [ZZ11] and [ZZ15] they used a form of Radon transform to detect the aponeuroses with success. Both the Radon transform and the Hough transform has been used with success in order to detect the fascicle orientation. However, as

[ZZ15] points out, the Hough transform relies greatly on the performance of the edge detector chosen, while the Radon transform does not necessarily require edge detection. Also, [RHW09] got good results detecting the fascicle orientation when performing the Radon transform on images filtered with a multiscale vesselness filter. The Radon transform was also used extensively by [Jal16], both for aponeurosis detection and fascicle detection, and some of the work done in that thesis would be very interesting to build upon.

CHAPTER 3

Data material

3.1 Data sets for algorithm development

In order to develop the algorithm for this thesis, we need some ultrasound images we can use as examples and as a way to determine parameters. These images are acquired using a Philips HD11 XE scanner operating at 12 MHz. The datasets depict the vastus lateralis. In this thesis, there has been two sets of data we have used while developing the algorithm, and evaluating the intermediate steps in the algorithm. We will refer to them as training set 1 and training set 2.

Training set 1 consists of 391 images. The ultrasound images depict the vastus lateralis. This data set is a sequence of images from one person during a muscle contraction. In this data set, the muscle fascicles are mostly quite distinct and clear, and the fascicles are mostly continous. The superficial aponeurosis is clearly visible, while the deep aponeurosis is more indistinguishable. Two examples of images from this data set are shown in Figure 3.1.

Training set 2 consists of 24 images and it also depicts the vastus lateralis. Unlike the previous set, this set of images is not taken in one sequence, but rather at several different times, in a span of three days. These images are mostly of a lower quality, with areas with fewer visible fascicles and speckle noise. Many of the fascicles also appear discontinous. Both the superficial and the deep aponeuroses are clearly visible, but in a few images, they are blended slightly into some of the fascicles. Examples of these types of images is show in Figure 3.2.

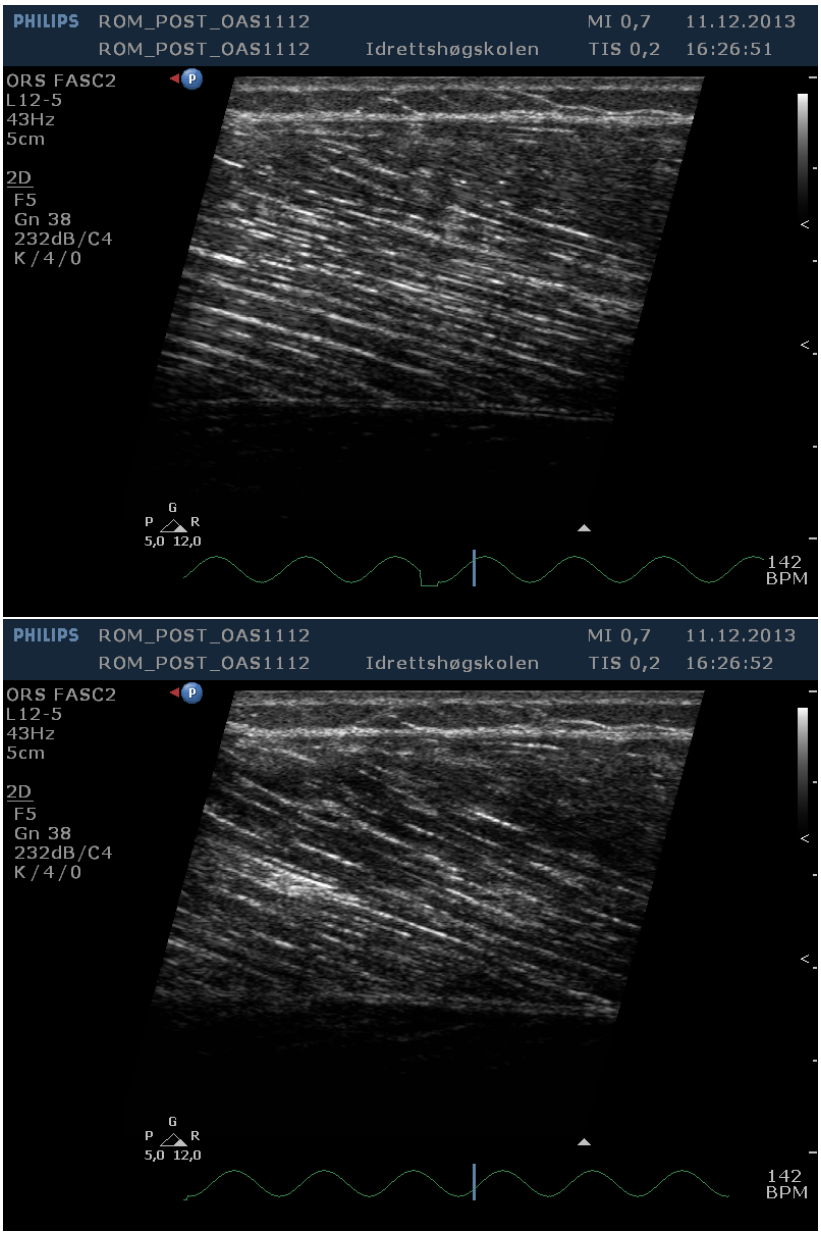


Figure 3.1: Images nr. 187 and 288 from training set 1.

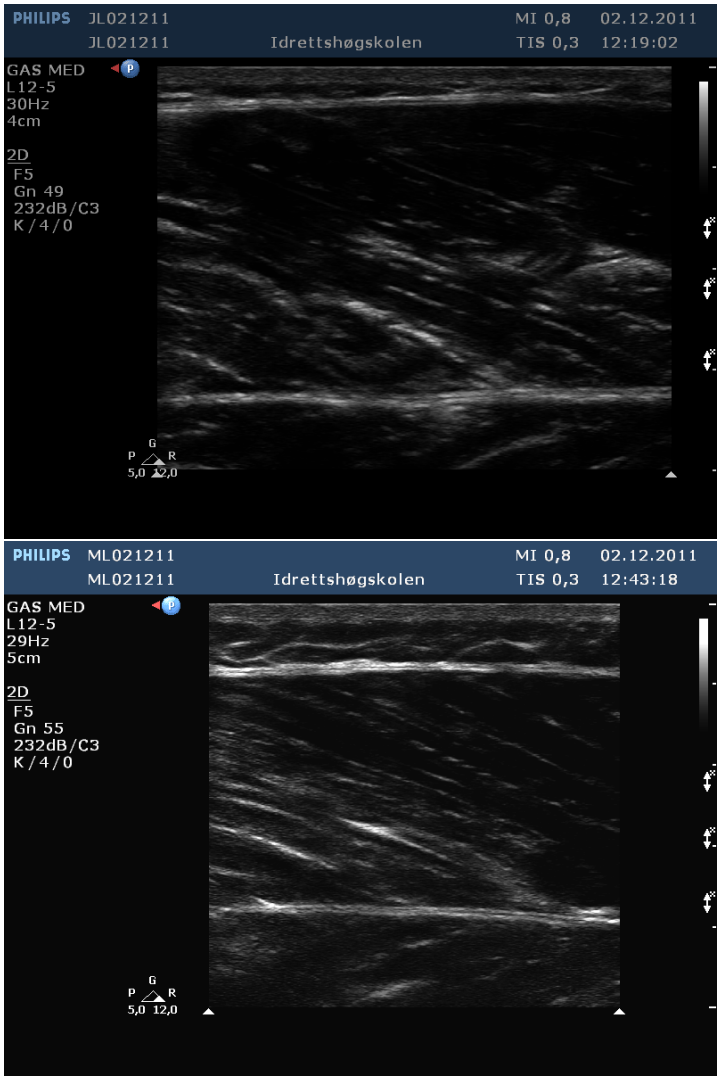


Figure 3.2: Images nr. 12 and 18 from training set 2.

3.2 Challenges with this type of datasets

In general when working with ultrasound images, one has to account for a certain amount of noise and variation in the image quality. Images of the same object can be of great variety depending on the type of ultrasound scanner, user determined settings on the machine, and the handling of the probe while scanning. Because of this, it is important to create a robust algorithm that can handle many of these variations in ultrasound images.

Both the aponeuroses are visible in all the images. However, while they are some of the most prominent parts of many of the images, the deep aponeurosis in training set 1 is almost indistinguishable in many of the images. This means that even if it might be tempting, methods using pixel brightness to locate the aponeuroses are not straight forward, and a potential algorithm can not rely exlusively on that.

Instead, one can analyze the structures in the image, and create an algorithm that relies on these. However, here comes the next challenge; many of the images have a large amount of noise. Not only speckle noise, but also structural noise, like artefacts and acoustic shadowing. This means that there may be little data, or erroneous data used as input to the algorithm.

In some of the images, there are also some structures that have a similar shape as the aponeuroses, close to the aponeuroses. A big example of this is the skin layer, which can be spotted on all the image examples from the data sets.

Finally, the data sets we have here are just some examples of how an ultrasound image of this type can appear, and the algorithm will naturally be designed to fit these types of images.

3.3 Data sets used for testing

In order to test the final product, we got access to two data sets to be used for testing only.

Test set 1 consists of 11 images. These images are not in the DICOM-format, so there is not any information to gather about the acquisition frequency or even whether or not the images are taken in a series or not. These are also of lower quality, with a high amount of noise, and with areas of few visible fascicles. Both the superficial and the deep aponeurosis is visible, but as in the second data set, in some of the images the fascicles may blend with the aponeuroses slightly. Examples of images from this data

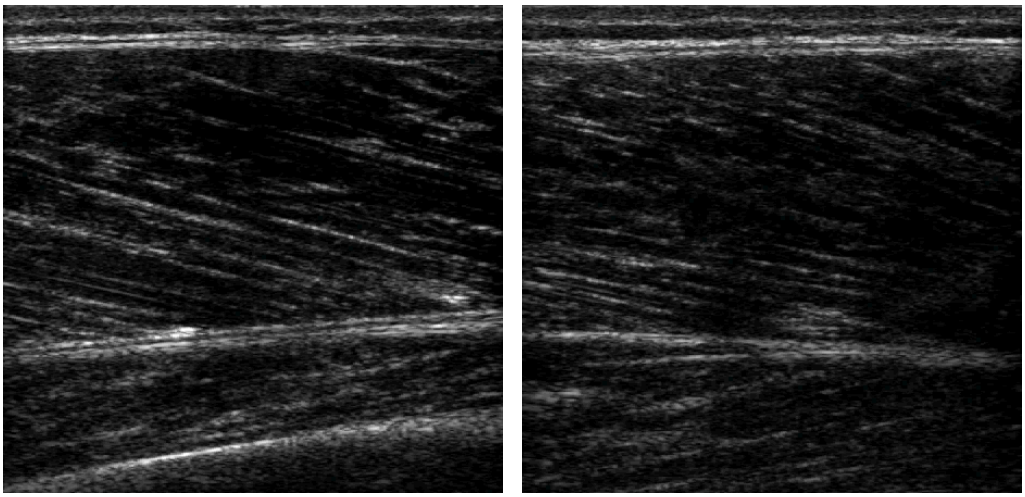


Figure 3.3: Images nr. 5 and 7 from the first test data set consisting of the 11 lower quality images.

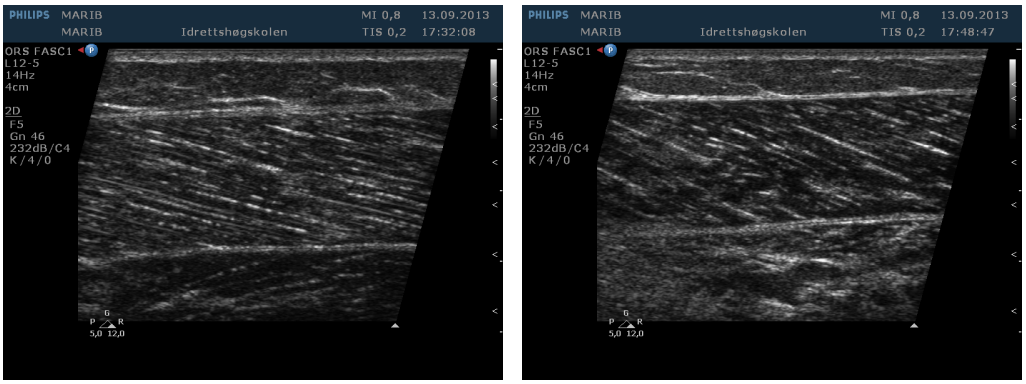
set is shown in Figure 3.3. There are no manual measurements for this set, and the images will be evaluated by visual inspection only.

Test set 2 consists of 60 images taken from 10 subjects with manual measurements for each subject. This set was gained access to one week before the deadline. In the image set, the operator held the probe at an angle relative to the aponeuroses in order to accentuate the fascicles. This leads to the aponeuroses being less prominent than if the probe was perpendicular to the aponeuroses.

For each subject, there are three images from the vastus lateralis muscle, and three images from the gastrocnemius muscle. The three images for each muscle are of the same region, but the probe has been handled slightly different in each of the three images. This gives images that are slightly different from each other, and the measurements will as a result also be. However, the measurements will still be in the same area.

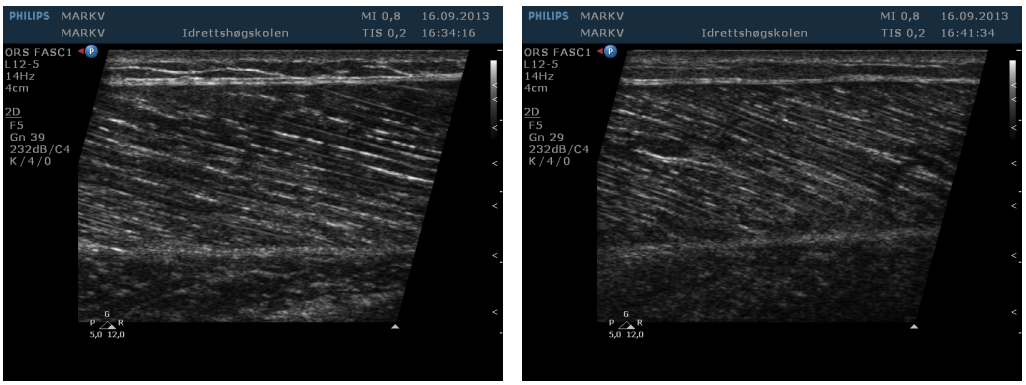
As for the manual measurements, we have the manual measurements for the pennation angle and fascicle length for each image. Each image has three manual measurements, since the manual measurements may vary slightly on the same image. Example images from the data set can be seen in Figure 3.4 and Figure 3.5.

Finally, two more images was gained access to in order to test the accuracy of the fascicle orientation part of the algorithm. The images in Figure 3.6 are of hair, and we know the exact underlying (main) orientation for both of these images. They are however, not completely suited for this algorithm, as is discussed in the results chapter in Section 9.4.



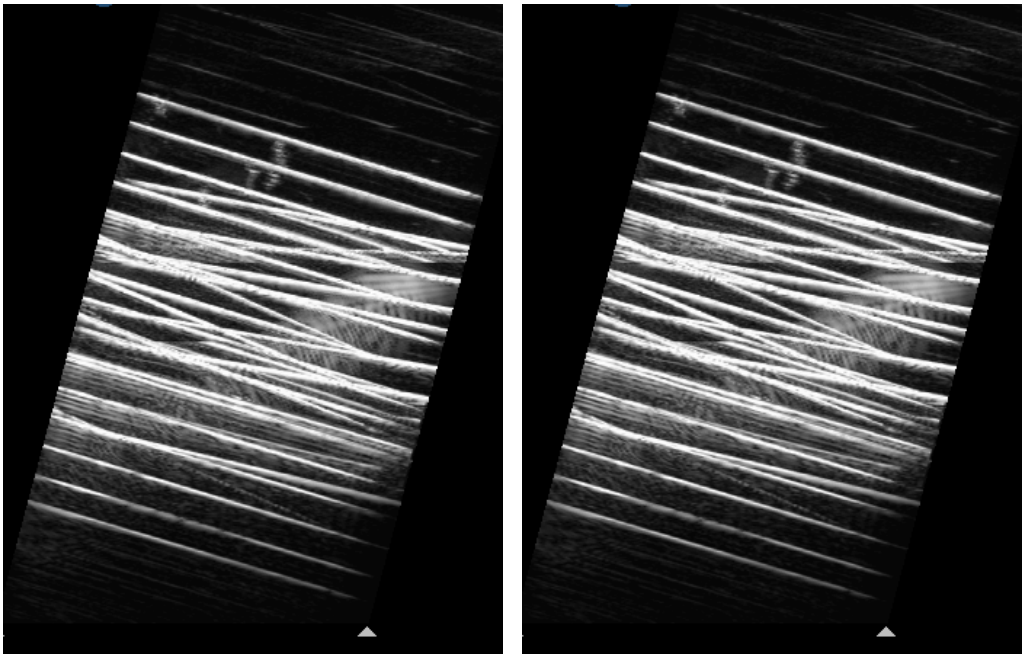
(a) (b)

Figure 3.4: Image nr. 3 from subject nr. 8 for a) the vastus lateralis and b) gastrocnemius muscle.



(a) (b)

Figure 3.5: Image nr. 3 from subject nr. 9 for a) the vastus lateralis and b) gastrocnemius muscle.



(a) (b)

Figure 3.6: The two ultrasound images of hair.

CHAPTER 4

Determining the region of interest

4.1 The problem

Since the ultrasound images from the dataset used in this thesis are stored using the DICOM-format, there is more information than simply the image of the muscle in the input image file. When processing the image, that information will contribute to noise and inaccurate results. Because of this, it is imperative that we locate the region of interest before other operations are done.

The images used for developing the algorithm have been of two types, the images where the region of interest has a rectangular shape, and the images where it has the shape of a parallelogram. The two types are illustrated in Figure 4.1.

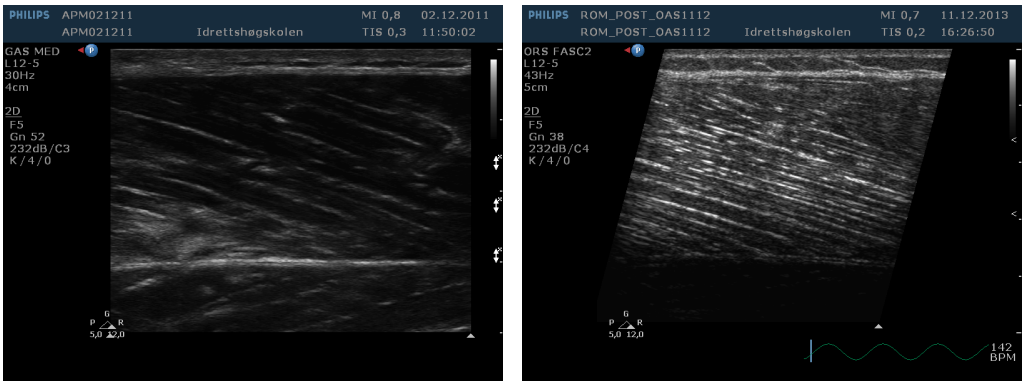


Figure 4.1: Ultrasound images that illustrate a rectangular and a parallelogram shaped region of interest, respectively.

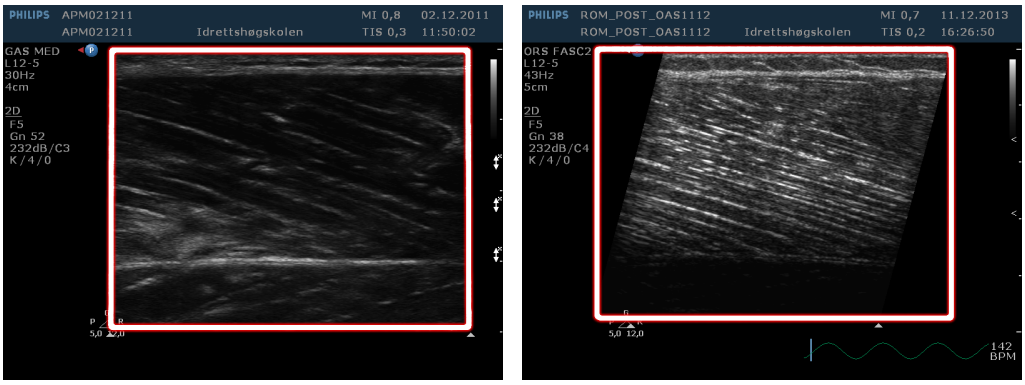


Figure 4.2: Images with the region of interest we hope to locate in the ultrasound images marked.

We want to create an algorithm that automatically locate the region of interest for both types of images, and the goal is that all unrelevant information that can provide errors in detection is removed at the same time as we preserve as much of the region of interest as possible. That final point holds particularily for the parallelogram shaped images, since there might be a trade off between keeping all the corners and keeping some of the unrelevant information. Below, an algorithm is presented that locates the regions of interest for both rectangular and parallelogram shaped images. The algorithm is designed to keep most of the image information in the input file. For the parallelogram shaped images, steps are taken when needed in the detection algorithms to ensure that the background does not interfere with the detections.

4.2 Algorithm for rectangular region of interest

We have chosen to expand on the Region of Interest-algorithm from [Jal16].

The idea is relatively simple. In the background information, there are mostly dark pixels, and usually the region of interest is generaly brighter than that. We want to locate the borders where the region of interest begins, and a simple way of doing so is comparing the mean of for instance the columns and see where there is sudden increase in values. If we want to locate the vertical borders, we compare the mean of the columns, to locate the horizontal borders, we compare the mean of the rows.

Based on the pixel values in the image, a small value and a high value is set, and will now be referred to as the low threshold value and the high

threshold value. A search is started from one of the four current borders, and the mean of each row or column is compared to the low threshold value.

Starting by searching for the top border, the algorithm begins with the threshold value selected, and checks if each of the rows in the top half of the image has a mean below the threshold. If none of the rows have a mean below the threshold, the threshold is increased slightly, and the search starts from the top again. If some of the rows are below the threshold, the row that is furthest toward the center of the image is chosen, since that supposedly is the last row before the region of interest begins. The new border is set a few pixels below this to ensure that we only get the region of interest in our new image. If the algorithm at any time hits the high threshold, the search is stopped, and the current border of that part of the image is used. This is a safety measure done to prevent the algorithm from choosing a wrong border far into the region of interest, and also to keep it from going into an eternal loop.

The bottom border is found by starting with the low value and comparing the rows starting at the bottom and going upwards. After the border has been located in the same manner as the top border, the image is cropped along these new borders so that the top and bottom area will not disrupt the detection of the left and right borders.

The left and right borders are found the same way as the top and bottom border, only using the means of the columns rather than the rows.

This algorithm however, will not give a good result for the parallelogram shaped regions of interest. That is because when we calculate the mean of the columns and detect the border from that, we will lose the corner pixels because the rest of the column pixel values are dark. Because of that, we have decided to modify the algorithm to include this case.

4.3 A modified version to include parallelogram shapes

It was desired to change the algorithm so that it would also get good results for parallelogram shaped regions of interest, as discussed in Section 4.1. A minor change is suggested to the detection of the left and right border. Instead of calculating the mean of the entire column to locate the left border, only the bottom 40 or so pixels of each column are used in the calculation. For this to work well, it is imperative that the lower border has been detected properly, so that the pixels used for calculating the mean on the left border are relevant, and that the algorithm avoids searching below the region of

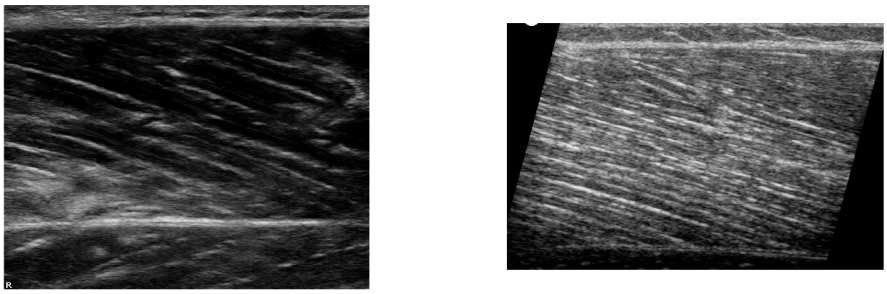


Figure 4.3: The results from using the modified region of interest algorithm on the example images.

interest.

The right border is located the same way, using only the top 40 or so pixels of each column in the calculations.

In addition, upon converting the image to a grayscale image, some of the background turns white in many of the images. In order to avoid those parts disrupting the algorithm, the computing image was thresholded so that the 10% brightest pixels were blackened for the search. When the border was located, this was used to crop the original image.

4.4 Results from the Region Of Interest-algorithm

The results when using this algorithm on the example images can be seen in Figure 4.3.

For our usage, this is a decent result. We should notice that for the image with the rectangular region of interest, some noise is detected in the bottom left corner, this seems to be so small that it probably will not affect the algorithm to any extent.

In the rightmost image, we should notice that quite a bit of the bottom of the ultrasound image has been cropped away. A look at the original image also reveals why, the pixels in that area are mostly so dark that it is difficult to distinguish the area from the backround. Since the lower aponeuroses is still in the image, and that no vital information is located in the area of the image that has been cropped away, this is not considered critical. Most of the pictures attempted to crop this way has given good results like these.

Other examples taken out of two different data sets where the region of interest algorithm has been used is seen in Figure 4.4. The algorithm seems

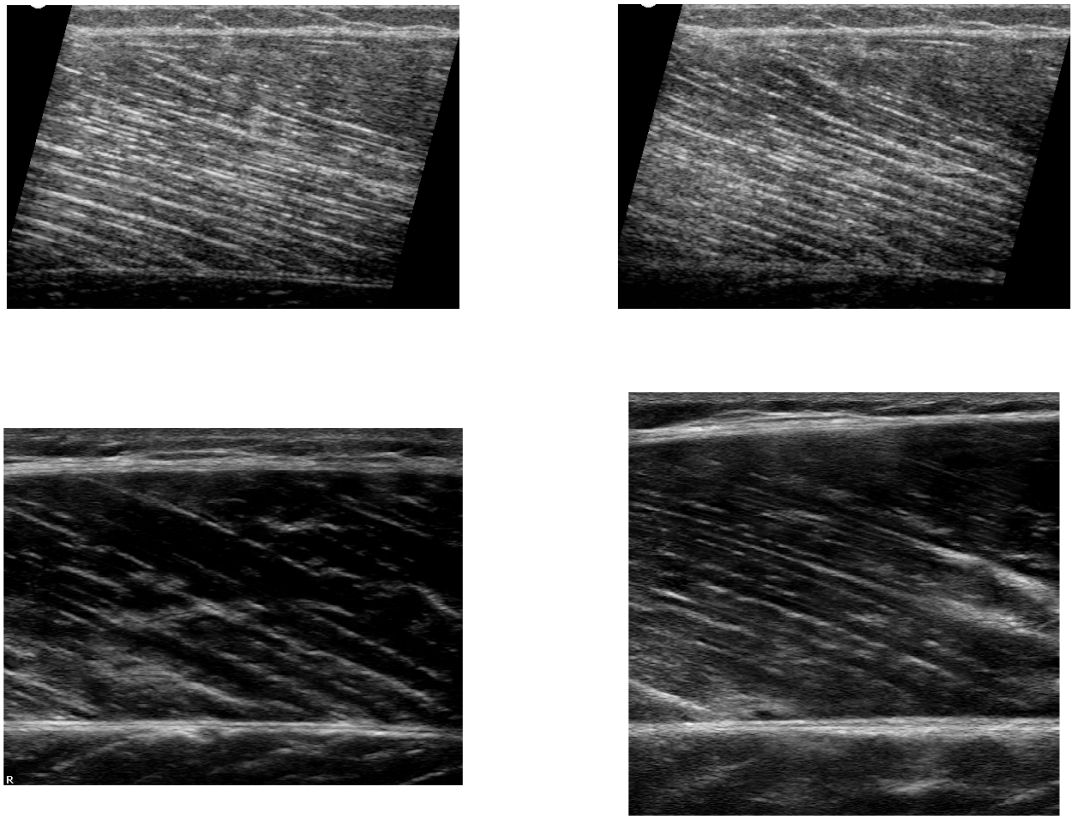
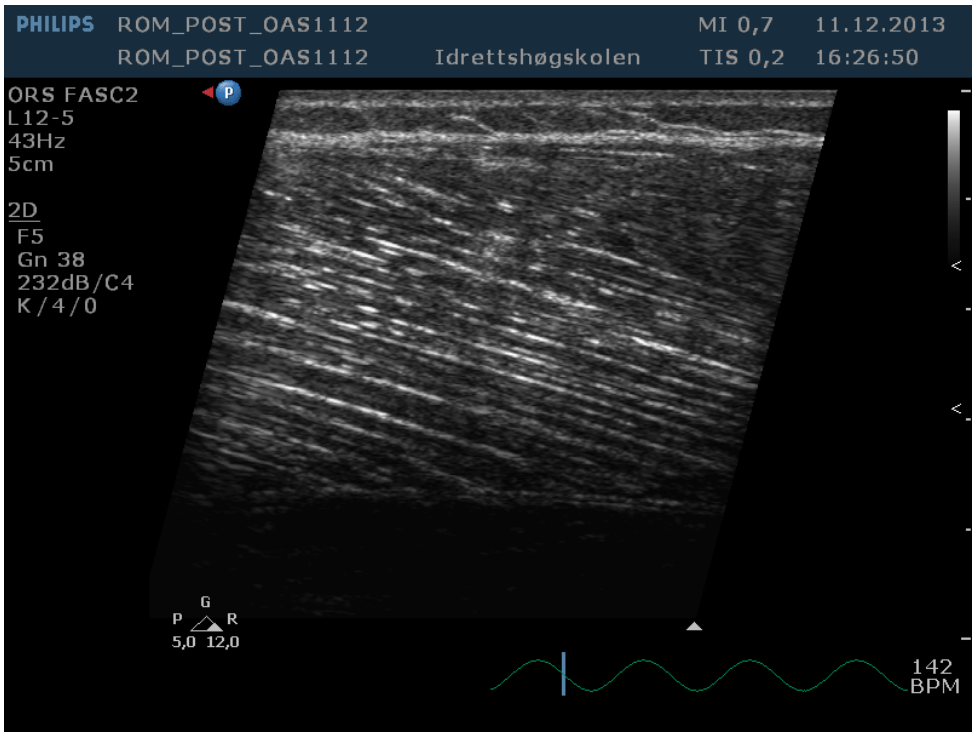


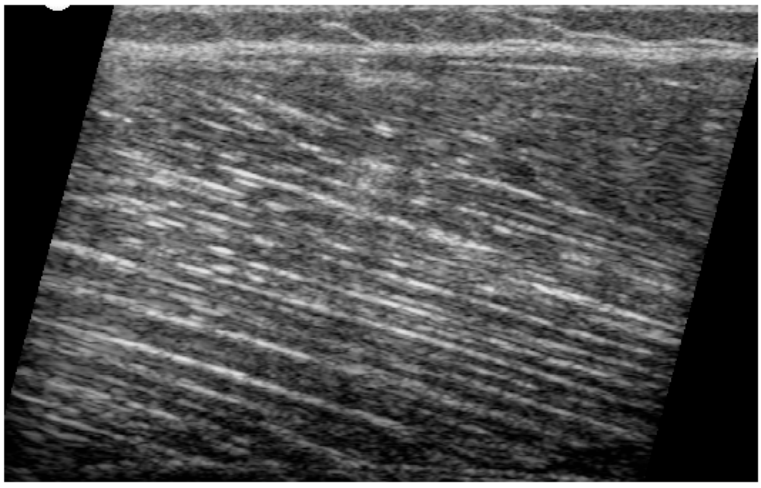
Figure 4.4: The results from using the modified region of interest algorithm on some example images.

to be well functioning in most cases tested.

In a few images in the first dataset (about 10 images or so, out of nearly 400), the image is cropped so that the deep aponeurosis is almost indistinguishable. The reason for this is that directly below the aponeurosis, the image is so dark that the algorithm chooses to crop directly below the aponeurosis. An example of this is shown in Figure 4.5.



(a) The original image, uncropped



(b) Resulting image where the deep aponeurosis is nearly indistinguishable

Figure 4.5: a) The original uncut image, showing that the image is very dark straight beneath the deep aponeurosis, and b) the result where the region of interest algorithm has almost cut away the aponeurosis.

CHAPTER 5

Radon transform

In this thesis, many of the parts of the algorithm are based on the use of the Radon transform. The Radon transform was introduced in 1917 by the Austrian mathematician Johann Radon [Rad17], and was first used in image analysis in 1965 in an article from Princeton [BB65]. Even though it was not called Radon transform in the article, that was essentially the technique the authors used [GHV04].

Since then, the Radon transform has been used in many various occations to detect line structures from images, and has lately been heavily in use when detecting structures from ultrasound images of muscle. See for instance [ZZ11], [RHW09] and [Wan+13].

In this chapter, we define the Radon transform and the normalized Radon transform. We also have a look at what it does in practice, for our usage.

5.1 Definition

In order to detect the aponeuroses and the muscle fasicles in the ultrasound images we make use of a Radon transform.

We define the Radon transform as in [GW06]. To explain the Radon transform we need some prerequisites. First note that, using polar coordinates, we can represent a line as

$$x\cos(\theta) + y\sin(\theta) = \rho. \tag{5.1}$$

We also need the notion of a discrete unit impulse.

Definition 5.1.1. Suppose t is a discrete variable. A *(discrete) unit impulse of* t (located at $t = 0$) is the function $\delta\colon \mathbb{R} \to \{0, 1\}$ defined by

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

We also require that

$$\sum_{t=-\infty}^{t=\infty} \delta(t) = 1. \tag{5.2}$$

We are now ready to define the discrete Radon transform.

Definition 5.1.2. Suppose an image I is represented by an $M \times N$-matrix for two integers M and N. The *(discrete) Radon transform* $g\colon \mathbb{R}^+ \times [0, 2\pi) \to \mathbb{R}^+$ is defined as

$$g(\rho, \theta) = \sum_{x=1}^{M} \sum_{y=1}^{N} I(x, y)\delta(x \cos\theta + y \sin\theta - \rho) \tag{5.3}$$

where ρ and θ are the polar coordinates, and δ is the impulse function.

This means that we sum over all the points in $I(x, y)$ located on the line defined by each pair of (ρ_j, θ_k). This is because the discrete unit impulse becomes zero when looking at a point pair (x, y) not on the line.

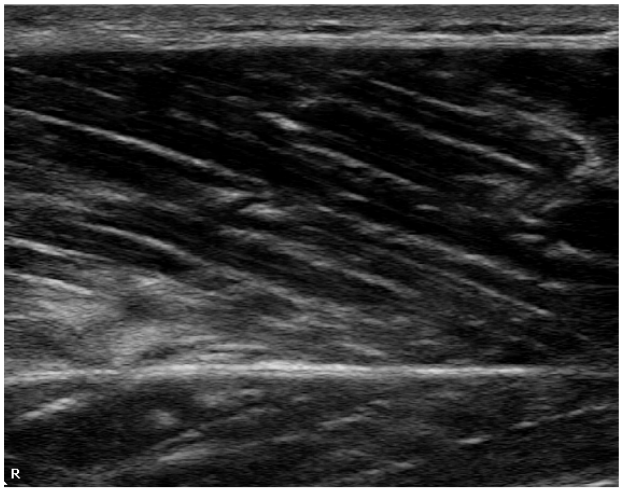
We observe that for lines that correspond to long bright structures in $I(x, y)$, the corresponding value of g will become high. Lines that almost correspond will get a slightly lower value, while the other values will usually stay low.

Figure 5.1b is an example on the Radon transform being used on Figure 5.1a.

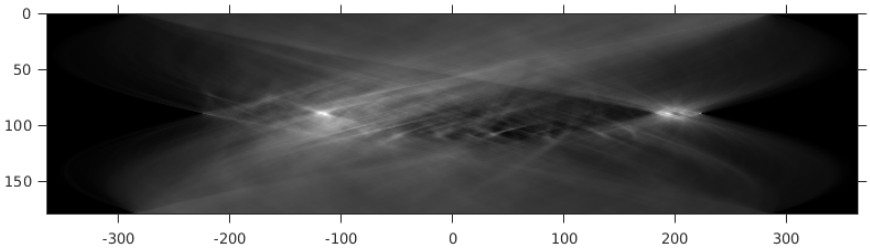
In Figure 5.1b we have two clear peak areas at around 90 degrees, one slightly below -100 and one at about 200 on the x-axis. Observing Figure 5.1a, we can tell that these peaks match the two aponeuroses. Further, we can also notice that the peak at $(\sim 200, \sim 90)$ is quite wide with a small hole in the middle, which represents that area of the original image quite well since the corresponding area is wide with a thin black line inside.

5.2 Normalized Radon transform

In some situations we would like to do a *normalized* Radon transform.



(a) Image before transformation.



(b) The resulting Radon transform.

Figure 5.1: Example of how a Radon transformed image appears. a) Original image, b) the resulting Radon transform. Note that instead of radians, this particular implementation of the algorithm uses degrees. The ρ-values are on the x-axis, and the degrees are on the y-axis.

Definition 5.2.1. Suppose an image I is represented by an $M \times N$-matrix for two integers M and N. The *(discrete) normalized Radon transform* $g\colon \mathbb{R}^+ \times [0, 2\pi) \to \mathbb{R}^+$ is defined as

$$g(\rho, \theta) = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} I(x, y)\delta(x\cos\theta + y\sin\theta - \rho)}{\sum_{x=1}^{M} \sum_{y=1}^{N} \delta(x\cos\theta + y\sin\theta - \rho)} \tag{5.4}$$

where ρ and θ are the polar coordinates, and δ is the impulse function.

In practice, we do a regular Radon transform on an $M \times N$ image, then create a normalization matrix by doing a Radon transform on an $M \times N$-matrix of ones, and divide the transformed image on the normalization matrix element-wise.

The reason this is done is to prevent the resulting transformed image in having larger peaks for longer lines.

CHAPTER 6

Detection of the aponeuroses

6.1 Detecting approximate location

The problem

We want to detect the two aponeuroses in the ultrasound images given as input. There are many possible approaches of doing this, but given the results achieved in [ZZ08], [ZZ15] and [Jal16] with the same type of images, making use of the Radon transform would be a good approach.

Usually, the aponeuroses appear in the images as bright, near horizontal lines. In many cases they are the brightest elements of the images, which would make it very tempting to detect them from image intensity peaks in the Radon transformed image. However, in some cases, especially for the lower aponeurosis, it lies in a dark part of the image. In this case, that aponeurosis might have a smaller image intensity peak in the Radon domain than other parts of the image. What one can make use of then is the fact that it still is mostly brighter than its immediate surroundings. An example of an image where this holds can be seen in the image to the right in Figure 4.1.

Because of this, we choose a strategy where we first find the approximate surroundings of both the aponeuroses, and then perform a Radon transform and search for image intesity peaks in the neighbourhood formed.

The Radon transform integrates over lines in the image, and forms an intensity image with the polar coordinates ρ and θ on the axes. The procedure is described in Chapter 5.

Detection using dominant angles

As in [Jal16] we want to locate the approximate location of the aponeuroses by detecting the dominant angle in each depth of the image, and locate where there are sudden changes in the dominant angle. This part of the algorithm is mostly the same as in that thesis, with a few minor changes.

In order to get a good overview of what the dominant angle is at each depth of the image, a large number of linearly spaced points in the image is selected, and a 30×70-window around each point is used as input to the Radon transform, which is calculated on all angles from 30 to 120 degrees. We choose $\theta \in [30, 120)$ with the assumption that the fascicles and the aponeuroses lie in this angle range, and looking at the training data at hand, this assumption holds. Keep in mind that θ represents the angle in polar coordinates, and not the cartesian system. This means that $\theta = 90$ gives a horisontal line, and the angles in the fasicle plane are at about $\theta \in [60, 80]$.

The maximum value of the resulting transformed image is found, reflecting the line in the image with the highest value when integrated over. Since we wanted to detect both the fascicles when in the fascicle plane, and the aponeuroses, the maximum is conceived to be a good overall measure.

Choosing the maximum value as a measure can in many cases be a risky operation, because it might choose an outlier with a high value instead of an actual peak, but since a sufficiently large number of points are used in the calculation, it is not problematic if a few outliers are picked.

An accumulation matrix is constructed, with the columns representing the angle θ, and the rows representing the corresponding depth in the image. The lines corresponding to the indices of the maximum values is calculated, and the depths at the middle of the detected lines are found. Then, for each line, the element in the accumulation matrix corresponding to the depth and angle of the line is incremented.

The resulting intensity image is formed from the accumulation matrix, and smoothed by a Gaussian filter with a sufficiently large standard deviation. In this case we have used $\sigma = 5$, and the result for the two example images is shown in Figure 6.1. From these images, we can se that there is a definitive change in angle where the aponeuroses lie when we compare to the original images.

The maximum angle for each depth is gathered into an array, again smoothed with a gaussian filter with $\sigma = 3$, and a simple method of numerical differentiation with respect to the angle is used in order to get an indication of where significant changes in angles occur. We found that it was enough to take $array(i + 1) - array(i)$, $\forall i \in [0, m - 1]$ where m is the height of the image in this case. The absolute value of the array is

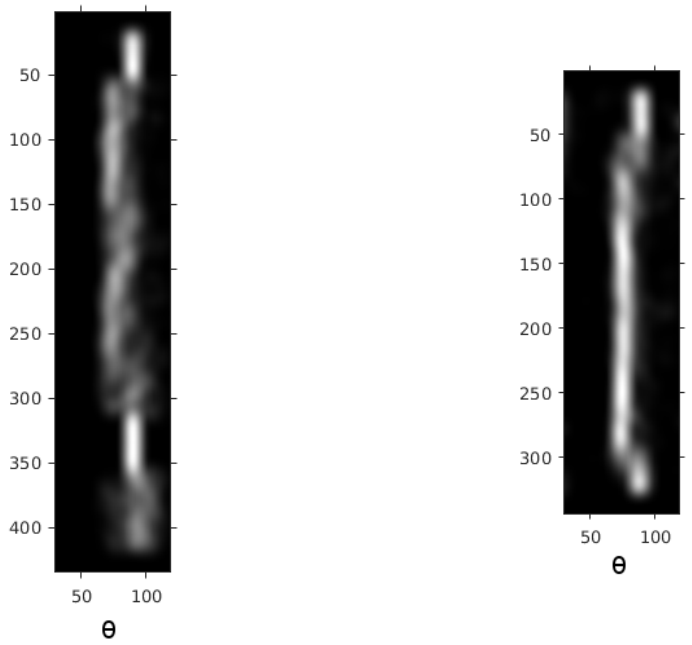


Figure 6.1: The accumulation matrices for the dominant angles of the example images in Figure 4.3. The angles are on the x-axis from 30 to 120 degrees, and the y-axis represents the pixel depth of the image.

taken to simplify further calculations, since the direction of the change is not necessary to take into account in this problem.

When detecting the aproximate area of the aponeuroses, we want to locate the part of the image where there is a change from the aponeurosis to the fasicle plane and vice versa. To do this, we have made two assumptions. The first one is that there is one aponeurosis in the upper half of the image, and one in the bottom half of the image. Since this usually holds, this is a good way of reducing potential errors in the algorithm. The other assumption is that the edge between the aponeuroses and the fascicle plane is not in the top 20 or bottom 20 pixels. This holds for most images, and if the edge is in the top 20 or bottom 20 pixels, it is higly likely that the aponeurosis is not entirely visible in the image. The reason this assumption is made is because when using a window of 30×70-pixels to compute the Radon transform, in order to create the accumulation matrix, the indices that would make parts of the window fall outside the border of the image were ignored. The result of this is that the detected dominant angles in the first 15 values and the last 15 values will all be 30 degrees, and then there will be an abrupt change from the 16'th value as it will get the actually detected

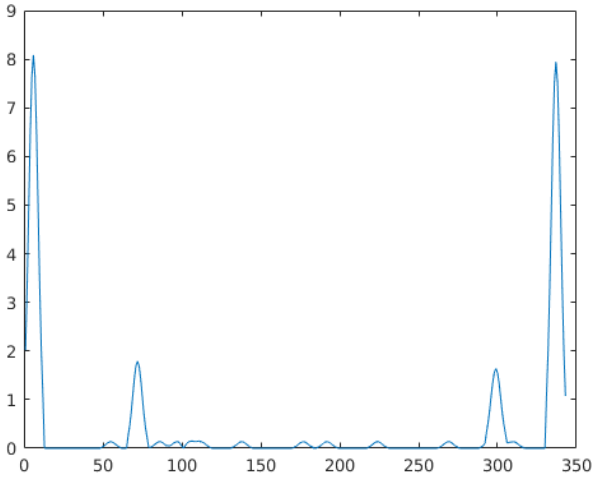


Figure 6.2: The differentiated array from the parallelogram shaped example image in Figure 6.1. The two peaks in the middle represent the edges we are looking for.

degree. To avoid this change, the first and last 20 pixels are ignored.

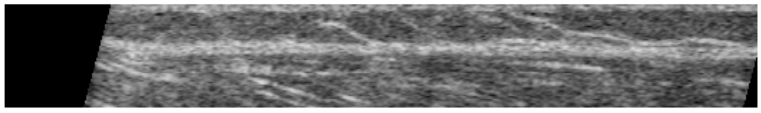
After taking these precautions, we can simply choose the index with the maximum value in the upper half of the differentiated array, and the index with the maximum value in the bottom half of the differentiated array. Usually, there could be small spikes of noise that would disrupt this detection, but since the array has been smoothed by a Gaussian filter, only the wider peaks should be present.

These two indices then represent where we believe the edges between the fascicle plane and the aponeuroses lie. An example plot of the differentiated array can be seen in Figure 6.2.

In this context, it should be mentioned that we are not that worried about choosing the edge on the 'wrong' side of the aponeuroses because upon inspection of the datasets, either the other side does not have a significant change of dominant angle, or the dominant angle falls outside the range of angles looked at in the Radon transform, namely $[30, 120)$.

For the superficial aponeurosis, we estimate it to lie between the value detected, and 100 pixels above it. For the deep aponeurosis, we estimate it to lie between the value detected and 100 pixels below it. We then get two subimages containing the aponeuroses, shown in Figure 6.3.

However, in some cases the image is so noisy, or the structures in the image are in such a way that there are not really any peaks in the differentiated array in the upper and/or lower half of the image. There also



(a) The superficial aponeurosis.

(b) The deep aponeurosis.

Figure 6.3: The resulting subimages from searching for the approximate location of the aponeuroses using the parallelogram shaped example image cropped by the Region of Interest algorithm (as shown in Figure 4.3).

might be several abrupt changes in the detected angle so that there is more than one peak in each half of the image. Thus, we need a different algorithm for the part of the image where this solution does not work. To robustify the algorithm to include these types of images as well, we need to solve two different problems. First, we need to find a set of conditions that are of such character that if they are not met, a different manner of detecting the approximate location is used. Secondly, such an alternative manner of detecting the approximate location must be found.

Detection using Radon transform

Luckily, before implementing the method described above, a different approach was thought of as a solution to the problem of detecting the approximate location. The idea was simply doing a Radon transform on the entire image, locating the two largest peaks in Radon space, and choosing those peaks as the approximate location of the aponeuroses. The background for this is that in many cases, the aponeuroses are the most prominent lines in the image.

The reason this method was rejected, was that in some cases, as mentioned in the beginning of this chapter, the aponeurosis in question is not necessarily one of the two brightest lines in the image. It may lie in a dark

part of the image. Of course, this part has not changed, and it is still a weakness in the algorithm.

However, if the image is noisy so that we cannot analyze the structure of it to decide where the aponeurosis lies, then pixel brightness may be a good second option to search for after all.

The new modification is that if the algorithm fails to locate a peak that after some criterion is believed to be the aponeurosis for one half of the image, a Radon transform for the polar coordinate angles $\theta \in [80, 100)$ is performed on that half of the image instead. Because we only need an approximate estimate of the whereabouts of the aponeurosis, the maximum point in the Radon space is chosen, and the x- and y-coordinates are calculated from the polar coordinates found.

After the x- and y-coordinates are calculated, we estimate the aponeurosis in question to lie between 40 pixels above the highest point on the line detected, and 40 pixels below the lowest point on the line detected.

Determining a criterion

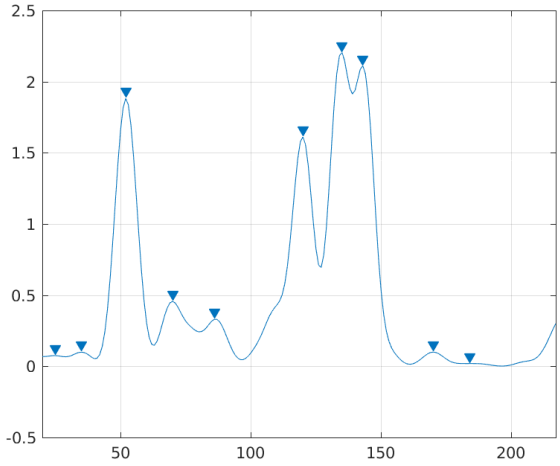
As mentioned previously, we need some type of criterion for when we believe the algorithm where we located changes in the dominant angle is not reliable in locating the aponeuroses.

Clearly, we want to sort out the cases where there is no clear peak, which means that we need to define a threshold where any peak under the threshold is disregarded.

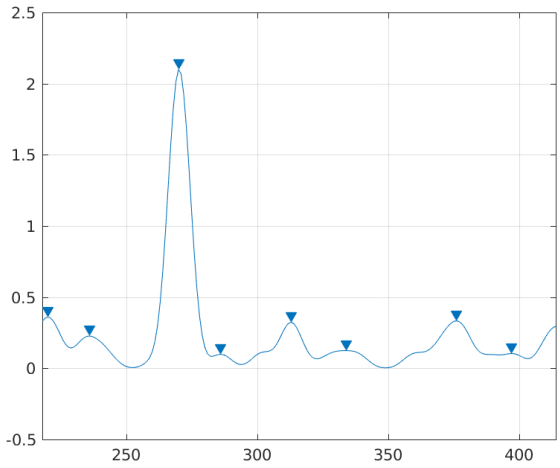
In the example in Figure 6.4 we can see that for the lower half of the image, there is one clear peak that gives us the position of the deep aponeurosis. However, for the upper half of the image, there are several peaks at about the same height, which means it is almost impossible to decide where the actual aponeurosis lies. In this case, and cases like it, we believe it would be best to use a different method of detection. Namely, we use the Radon transform in the upper half of the image to locate the aponeurosis instead. This means we also need to set a threshold for how much taller the largest peak needs to be compared to the other peaks.

In the example in Figure 6.5, we can see that we have only one clearly defined peak, and so in this image we assume it is safe to use that peak to estimate the approximate position of the aponeuroses.

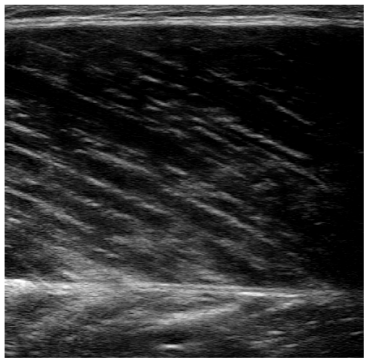
In general, when looking through the data of the differentiated arrays, a threshold of 0.5 between the highest peak and the next to highest peaks seems like it will give us good results. Then we always get one peak that distinguishes itself.



(a)

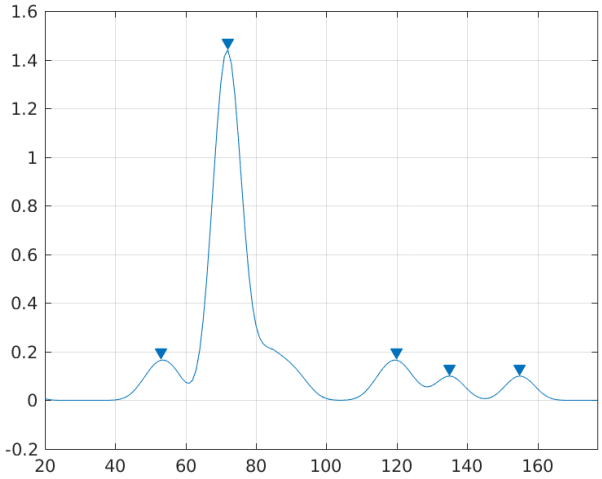


(b)

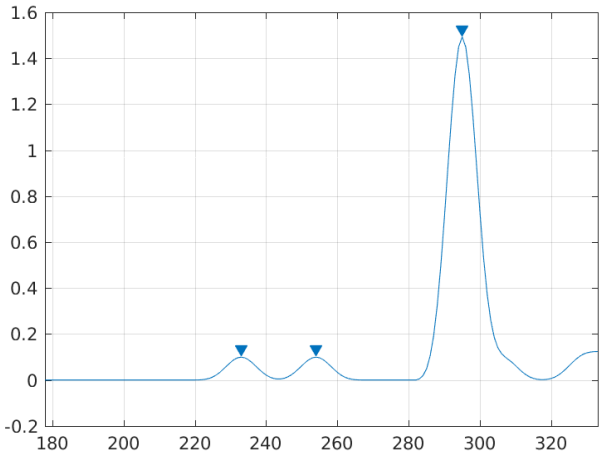


(c)

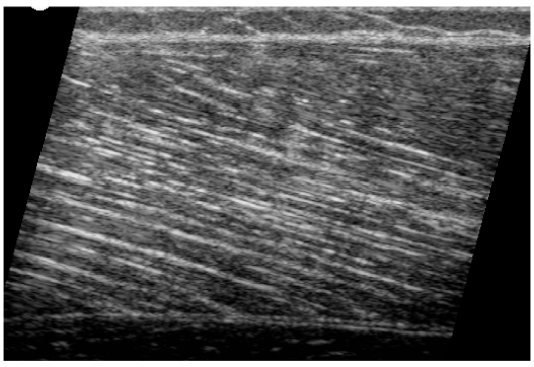
Figure 6.4: a) The differentiated array for the upper half of the image, b) the differentiated array for the bottom half of the image and c) the ultrasound image in question. The x-axis in a) and b) represents the rows in the image, where 1 is the top row.



(a)



(b)



(c)

Figure 6.5: a) The differentiated array for the upper half of the image, b) the differentiated array for the bottom half of the image and c) the ultrasound image in question. The x-axis in a) represents the rows in the image, where 1 is the top row.

6.2 Detecting accurate aponeurosis location

Given that the approximation was able to find the correct location, we should now have two subimages of the original images at hand, one of the superficial aponeuroses, and one of the deep aponeuroses as shown in Figure 6.3.

Now that the search for the aponeuroses is narrowed down to a smaller window, the issue mentioned in Section 6.1, with aponeuroses in darker pars of the image giving smaller peaks in the Radon space than brighter parts of the image, is now diminished. This is because we now evaluate a smaller neighborhood around the aponeuroses.

However, the Radon transform integrates over linear structures, and produce lines. Since the aponeuroses in many cases not necessarily are completely linear we want to divide the subimages into many subimages, perform Radon transform on these and use the data gathered to construct a curve that represents the aponeuroses.

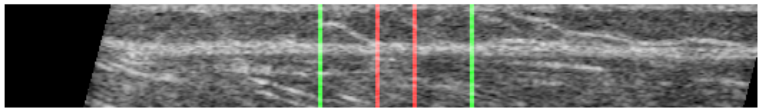


Figure 6.6: One of the pieces the image is divided into. The part of the image between the red lines is the part we want to fit a line to, while the part of the image between the green lines is the part we use as input to the Radon transform.

The subimage is split into a specified number of pieces, in this case 20 pieces, each with some overlap with pieces on each side to get some contextual information. 20 pieces were in this case selected after experimenting with different sizes of overlap and number of line pieces. It was seen as a good compromise between having larger line pieces that might lose some of the curvature in the aponeuroses, and having smaller pieces which might lead to the algorithm detecting other small line fragments close to the aponeurosis.

Of course, that number was selected for the size of these images, a more robust manner to fit all sizes of images would have been to choose the number of line pieces based on the number of pixels in the horizontal direction. This is easily adjusted.

An illustration of this can be seen in Figure 6.6. Then, a Radon transform is performed on each image part with overlap, searching for angles $\theta \in$

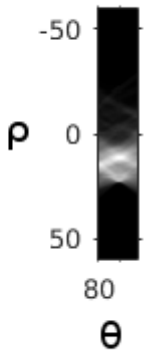


Figure 6.7: The Radon transform of one of the image parts. We want to locate the most prominent peak in this image. θ ranges from 80 to 100 degrees.

$[80, 100)$, only searching in that angle range because the structure we want to find is about 90 degrees. The transformed image might look something like in Figure 6.7. In order to locate the largest peak, a measure is needed. In this case, a maximum measure will not suffice. That is because we need the detection for each line piece to be as accurate as possible, and it is not at all guaranteed that the center of the peak we are interested in is the highest value in the image. In this case, a more robust method would be to locate the center of mass. How to calculate this can be found in [Alb16].

Representing the transformed image as a the matrix $I(i, j)$, in order to find the center of mass, the following is calculated.

$$\tilde{x} = \frac{\sum_i \sum_j iI(i,j)}{\sum_i \sum_j I(i,j)} \tag{6.1}$$

$$\tilde{y} = \frac{\sum_i \sum_j jI(i,j)}{\sum_i \sum_j I(i,j)} \tag{6.2}$$

When rounded, $\tilde{x}$ and $\tilde{y}$ are then the indices of the centre of mass.

Using the indices found, we locate the corresponding ρ- and θ values, and convert them into the corresponding x- and y values in the cartesian coordinate system for the correct part of the main image.

Doing this for both the aponeuroses yields the result in Figure 6.8.

There are some issues in the resulting images. The edges are problematic in the case of the parallelogram shaped images where the Radon transform performed in the black edges has little or no data to detect from, so the lines produced by the algorithm are very uncertain.

To overcome this issue, we originally experimenting with using weighs on the data when fitting a curve to it, so that the data found in the middle of the

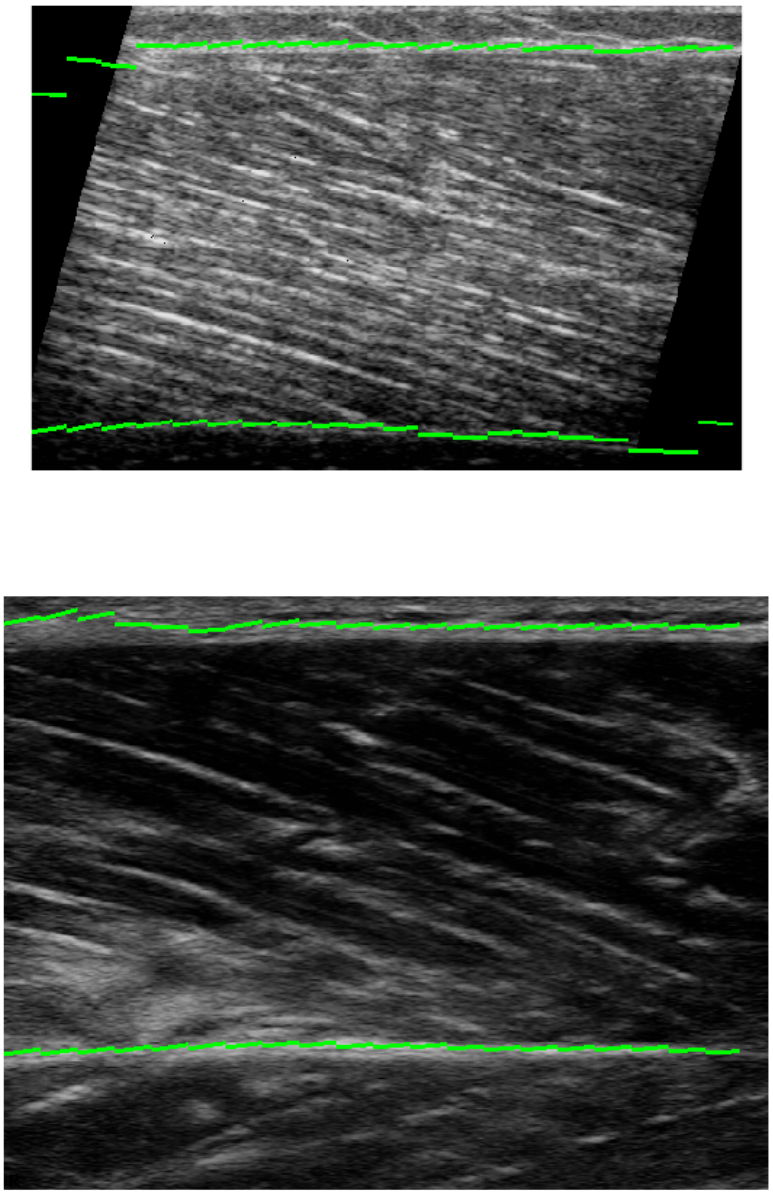


Figure 6.8: The detected lines using Radon transform on 20 subimages of each of the aponeuroses for the example image in Figure 4.3.

image were weighted more than the data near the edges. However, this had the side effect that if the data in the middle had some small perturbations, then these would become prominent in the resulting curve.

Another method would be to replace the 75 data points closest to the border on either side with the next/previous 75 data points. However, this method also had some serious setbacks. An example of that is that if the aponeurosis in question has some kind of angled shape in the edges, then copying the 75 previous data points would straighten the line, rather than follow the curve as it should.

In the end, the conclusion was that the simplest and best method is to create a mesh to detect where the borders between the actual image and the background is located, and then only calculate the spline with the data from the actual image, and not the background. It should also be mentioned that this will not affect the rectangular images, since the border detected will be the actual border of the image.

The final problem is poor detection is some areas. This can especially be seen in the upper left corner of the rectangular image, where the Radon transform detects a different part of the line than the rest because the line is quite wide. This may be a large source of error in further calculations. In this exact case, a solution could be to use some kind of edge detection that would give a maximum or a minimum at the edge, and then do an optimization algorithm to search for this extreme with the line pieces in question. However, since a great deal of the aponeuroses do not have a clear edge, this does not work well in general. An example of this can actually be seen in the same image on the left part of the deep aponeurosis, where there is no clear edge, and the location of the line piece might move further away from the actual aponeurosis.

There are most certainly ways to overcome this issue, one thing we could do is for instance examining the homogenity of the area in question before an eventual search, but searching for a solution to this problem was not prioritized in this thesis.

6.3 Modelling the aponeuroses using spline curves

Even though most of the lines hit the aponeuroses, we want a continous curve with some degree of smoothness since the goal is to do further calculations on it. There are several possibilities in this segment, one could for instance fit a quadratic curve to the data. In this thesis, we have chosen to fit a spline

curve to the data using the least squares method, because of its flexibility and smoothness properties.

We want to fit a spline curve to the line pieces detected with the Radon transform using least squares approximation. The following subsection will discuss this method.

To represent the aponeuroses, we need a curve that will use the information received from the line pieces. However, we do not want the restrictions of linear or quadratic polynomials. A solution is then to use polynomial pieces to represent the curve, with continuity at the joins. It turns out that spline functions fit these criteria.

The construction of splines, the definitions and the least squares theory seen here is to a great extent based on [LM08].

Definition of splines

Splines are piecewise polynomial curves. Their construction is based on repeated averaging and the fact that we can represent a straight line between two points $a, b \in \mathbf{R}$ as

$$f(x) = \frac{t_1 - x}{t_1 - t_0} a + \frac{x - t_0}{t_1 - t_0} b, \qquad x \in [t_0, t_1] \tag{6.3}$$

One of the very nice properties about splines is that it is easy to smoothly join together neighbouring polynomial pieces. In fact, if you have two spline pieces of degree p, the construction automatically gives you $\mathcal{C}^{p-1}$-continuity at the join – or less, if that is something you want.

Splines are defined recursively, which makes the definition quite technical.

Definition 6.3.1. Let p be a nonnegative integer and let $\mathbf{t}=(t_j)$, the *knot vector* or *knot sequence*, be a nondecreasing sequence of real numbers of length at least $p+2$. The jth *B-spline* of degree p with knots $\mathbf{t}$ is defined by

$$B_{j,p,\mathbf{t}}(x) = \frac{x - t_j}{t_{j+p} - t_j} B_{j,p-1,\mathbf{t}}(x) + \frac{t_{j+1+p} - x}{t_{j+p+1} - t_{j+1}} B_{j+1,p-1,\mathbf{t}}(x) \tag{6.4}$$

for all numbers x, with

$$B_{j,0,\mathbf{t}}(x) = \begin{cases} 1 & \text{if } t_j \le x \le t_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

Here, the convention is assumed that $'0/0 = 0'$.

Now, we are ready to define a spline function.

Definition 6.3.2. For $n, p \in \mathbf{N}$ where n is the number of B-splines, and p is the degree, we let $\mathbf{t}= (t_j)_{j=1}^{n+p+1}$ be a knot vector for a total of n B-splines. The linear space of all linear combinations of these B-splines is the *spline space* $\mathbf{S}_{p,\mathbf{t}}$ defined by

$$\mathbf{S}_{p,\mathbf{t}} = \text{span}\{B_{1,p,\mathbf{t}}, \cdots, B_{n,p,\mathbf{t}}\} = \{\sum_{j=1}^{n} c_j B_{j,p,\mathbf{t}}\} \tag{6.6}$$

An element $f = \sum_{j=1}^{n} c_j B_{j,p,\mathbf{t}}$ of $\mathbf{S}_{p,\mathbf{t}}$ is called a *spline function*, or just a spline, of degree p with knots $\mathbf{t}$, and $(c_j)_{j=1}^{n}$ are called the B-spline coefficients of f.

Least squares approximation with splines

In order to create a spline curve, we need to use a method to utilize the data points. Instead of interpolating the line pieces, which would not give a good representation of the aponeuroses considering how the line pieces are scattered, a better solution would be to use an approximation that minimizes the error. Therefore, a least squares approximation is expected to give a better result.

Given some data points, we want a spline of degree p that solves the least squares problem. In this case, we only need $p = 1$, but the approach is the same either way.

First, a knot vector $\mathbf{t}$ needs to be created. These points can be sampled from the data points. From the knot vector, we construct the spline space $\mathbf{S}_{p,\mathbf{t}}$. Now we have all we need to turn this into a minimization problem. Given data $(x_i, y_i)_{i=1}^{m}$ with $x_1 < \cdots < x_m$ for $m \in \mathbf{N}$, and a spline space $\mathbf{S}_{p,\mathbf{t}}$, we want to find a spline $h \in \mathbf{S}_{p,\mathbf{t}}$ which solves the minimization problem

$$\min_{h \in \mathbf{S}_{p,\mathbf{t}}} \sum_{i=1}^{m} (y_i - h(x_i))^2 \tag{6.7}$$

Suppose $\mathbf{c}= (c_1, \cdots, c_n)$ are the B-spline coefficients of some $h \in \mathbf{S}_{p,\mathbf{t}}$. Then

$$\sum_{i=1}^{m} (h(x_i) - y_i)^2 = \sum_{i=1}^{m} (\sum_{j=1}^{n} B_{j,p,\mathbf{t}}(x_i) c_j - y_i)^2 \tag{6.8}$$

If we create an $(m \times n)$-matrix $\mathbf{A} = (a_{i,j}) \in \mathbf{R}^{m \times n}$ and set $a_{i,j} = B_{j,p,\mathbf{t}}(x_i)c_j$ we get

$$\sum_{i=1}^{m}(\sum_{j=1}^{n} B_{j,p,\mathbf{t}}(x_i)c_j - y_i)^2 = \sum_{i=1}^{m}(\sum_{j=1}^{n} a_{i,j} - y_i)^2 \tag{6.9}$$

$$= \|\mathbf{Ac} - \mathbf{y}\|_2^2 \tag{6.10}$$

where $\mathbf{y} = (y_1, \cdots, y_n)$.

Thus we have our problem on a very known form, because the linear least squares problem

$$\min_{\mathbf{c} \in \mathbf{R}^n} \|\mathbf{Ac} - \mathbf{y}\|_2^2 \tag{6.11}$$

always has a solution $\mathbf{c}^*$ which can be found by solving the linear set of equations

$$\mathbf{A}^T\mathbf{Ac}^* = \mathbf{A}^T\mathbf{b} \tag{6.12}$$

Solving this gives the coefficients needed for the spline curve desired.

In order to make the final curve, we start by creating a mesh to locate the borders of the image. Then we use the data gathered from the detection step, excepting the data outside the mesh, as input to the least squares spline approximation. The mesh is created by thresholding the image so that the darkest 5% of the images is turned black, and the rest of the image is turned white. Then, we search for the non-zero instances in the row at the approximate height of the aponeuroses, and select the start point at the first non-zero instance that has consequtive non-zero instances after it. The same is done for both aponeuroses, to find both start points and end points.

The resulting spline curves for the example image is in Figure 6.9. Clearly, the representation of the superficial aponeurosis is characterized by the fact that the line pieces in the left part of the image has found a different part of the aponeurosis than the rest, as discussed at the end of Section 6.2.

6.4 Evaluating the aponeurosis detection algorithm

Summary of algorithm

In this chapter, we have created an algorithm that

- Creates an array depicting the size of change in dominant angle in each depth of the image.

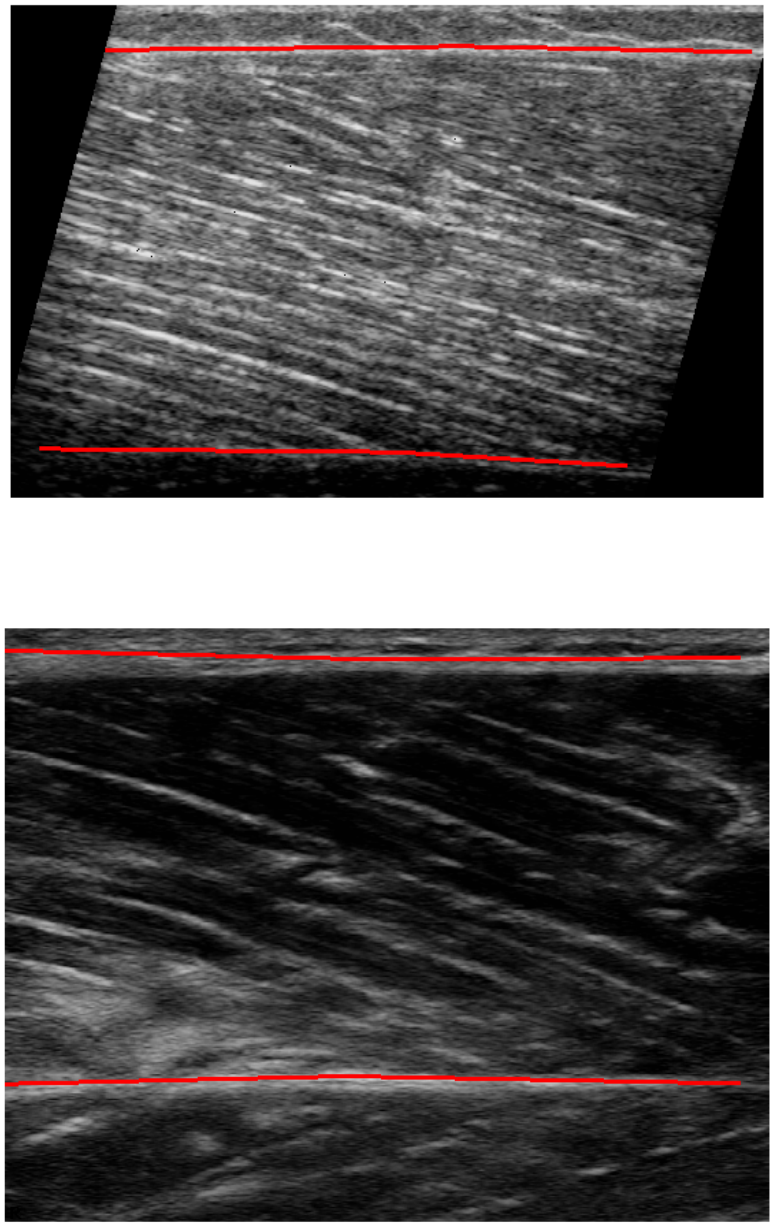


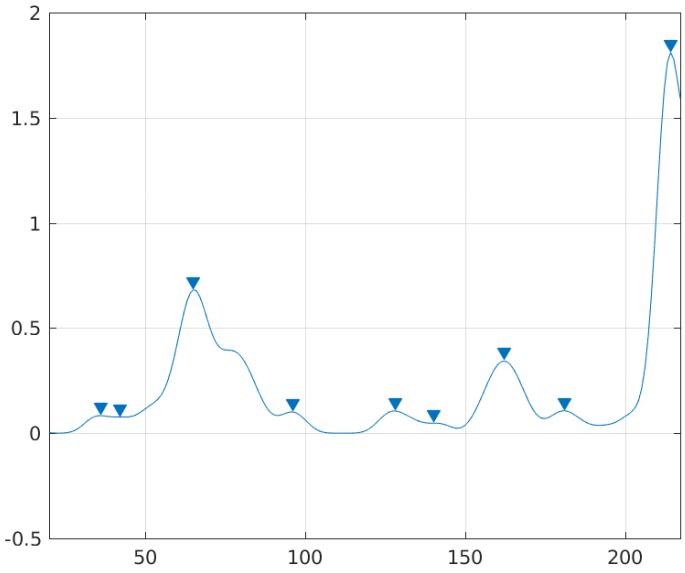
Figure 6.9: The resulting approximation of the aponeuroses using least squares spline curves.

- Checks that it has only one clear peak for each half of the images. Depending if a half as one clear peak or not:
  - If it has one clear peak: The algorithm chooses the largest peak as the approximate area of the aponeurosis.
  - It it does not have one clear peak: The algorithm performs a Radon transform is done on that half of the image, and selects the largest peak in the transformed image as the approximate area.
- Each approximate area is divided in 20 pieces, and a Radon transform is on each piece with overlap on both sides.
- The center of mass in each transformed image piece is located, and the coordinates for the point is transformed into Cartesian coordinates for the corresponding image region.
- The data from the line pieces is used as input to create a least squares spline curve.

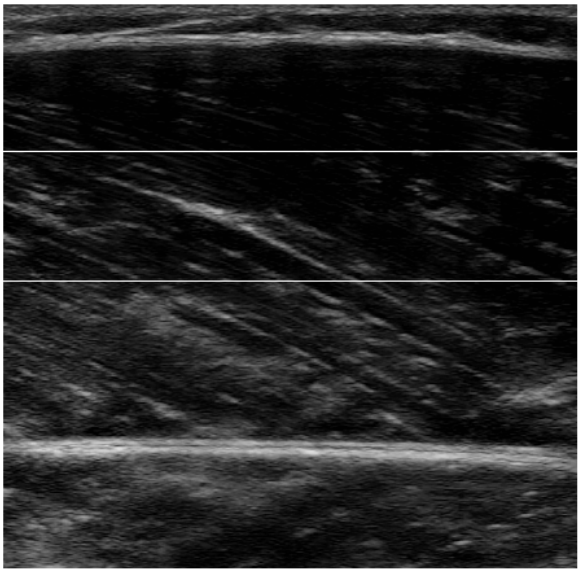
Approximate location

When testing this part of the algorithm on all the images in training set 2, there is only one example where the algorithm has failed to locate the approximate area of where the aponeuroses lies. That example can be seen in Figure 6.10. Looking at the differentiated array we see that there is one peak looming over the rest, and it is not placed where the aponeurosis is placed. When looking at the image, it is clear why. The fascicle plane is very dark with little structure until about 220 pixels down, where the structures suddenly are a lot clearer. This could be avoided by making an assumption that the aponeurosis is not in the middle 100 pixels or so in the image, but this is an assumption we cannot be sure will always hold, and in fear of overfitting the algorithm to the training data, this is therefore not implemented.

On a few images in training set 1, the algorithm failed when locating the deep aponeurosis. In these cases, the aponeurosis was almost/partly outside the image. The approximate area is found, but because the approximate area then is very small, the accurate locating algorithm fails because the Radon transform does not produce good results for a window of just a few pixels height. An example of such an image where this fails is show in Figure 6.11. As we can see, the deep aponeurosis is almost indistinguishable, and partly outside the image. This has to do with the cropping, and has

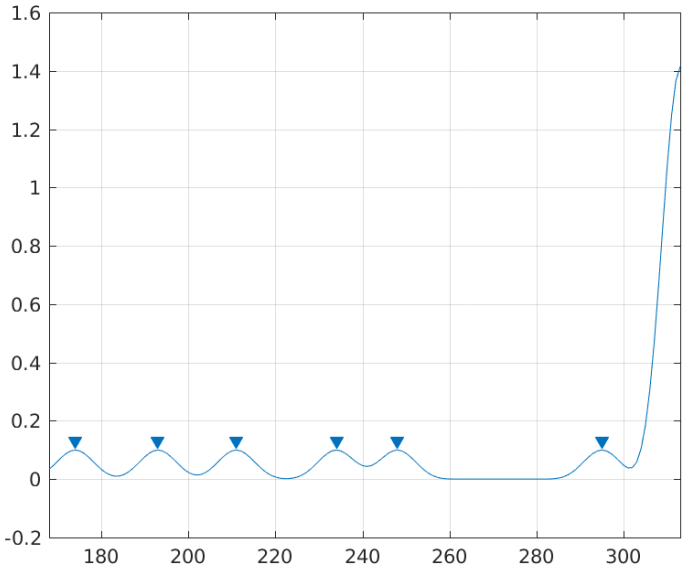


(a)

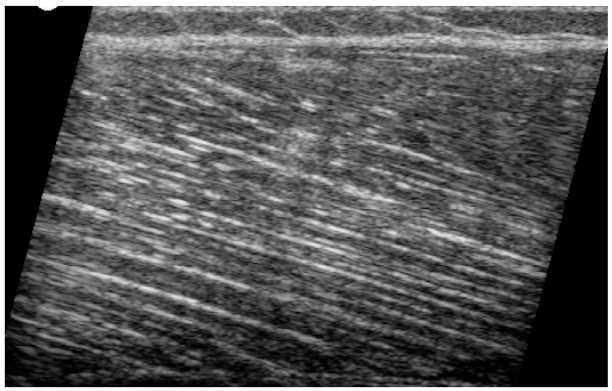


(b)

Figure 6.10: a) The differentiated array for the upper half of the image and b) the ultrasound image in question with the area where the algorithm concludes the superficial aponeurosis is located, marked with white borders. The x-axis in a) represents the rows in the image, where 1 is the top row.



(a)



(b)

Figure 6.11: a) The differentiated array for the lower half of the image and b) the ultrasound image in question.

previously been discussed in Section 4.4, where this exact image has been shown in Figure 4.5.

We recall training set 1 and training set 2 from Chapter 3.

Accurate location

In general, the algorithm appears to give *very good results* when detecting the accurate location. When testing on the two training data sets, excepting the images discussed in the approximate location evaluation, we get *good results for all images in training set 1.* This is including images such as Figure 6.15, which in spite of the slight error in the corner, in general is a good match.

In training set 2 we get good results in 19 out of 24 images, which is not so bad considering the images were picked out because they were noisy.

Picking out three images from each data set, we get some examples of typical results from the algorithm. Some places, we get the same issue as in Figure 6.9, where the points are chosen at slightly different places on the wide aponeurosis, but not as severe. The examples are shown in Figure 6.12.

In general, this is how the results usually appear, and the spline curves seems to *agree very well* with the underlying structure.

Now, let us have a look at some of the cases where the spline curves do not fit well with the underlying structures. We will discuss some examples found when testing the algorithm on all the data sets.

In Figure 6.13, we can see the one of the worst cases of detection in the part of the algorithm dedicated to detect the accurate position of the aponeurosis. In this case, the piecewise Radon transform has completely failed, since it has detected tissue above the aponeurosis instead of the actual aponeurosis on the left side of the image. Clearly, this is a weakness in the method of detection. The Radon transform might detect other bright line structures nearby instead of the actual aponeurosis.

An idea could be to make some kind of extra limitation on the lines. This could for instance be that the lines cannot stray too far from the mean height. However, in this case the detection is so much off the mark, that the mean height would be above the actual aponeurosis. If the line is very skew, this is not a good strategy either.

Perhaps if the image pieces we perform the Radon transform on were wider, it would be less likely to pick up on these small structures. At the same time, one would then get a less accurate approximation for the more curved aponeuroses, so it is not a good solution to the problem.

Finally, a last suggestion would be to combine the two ideas above. An iterative algorithm could try various sizes on the line pieces, and then choose the lines that give the most consistent result. Of course, then one would also have to come up with a clear definition of what a consistent result entails.

The same issue is also presented in Figure 6.14. Notice that the least

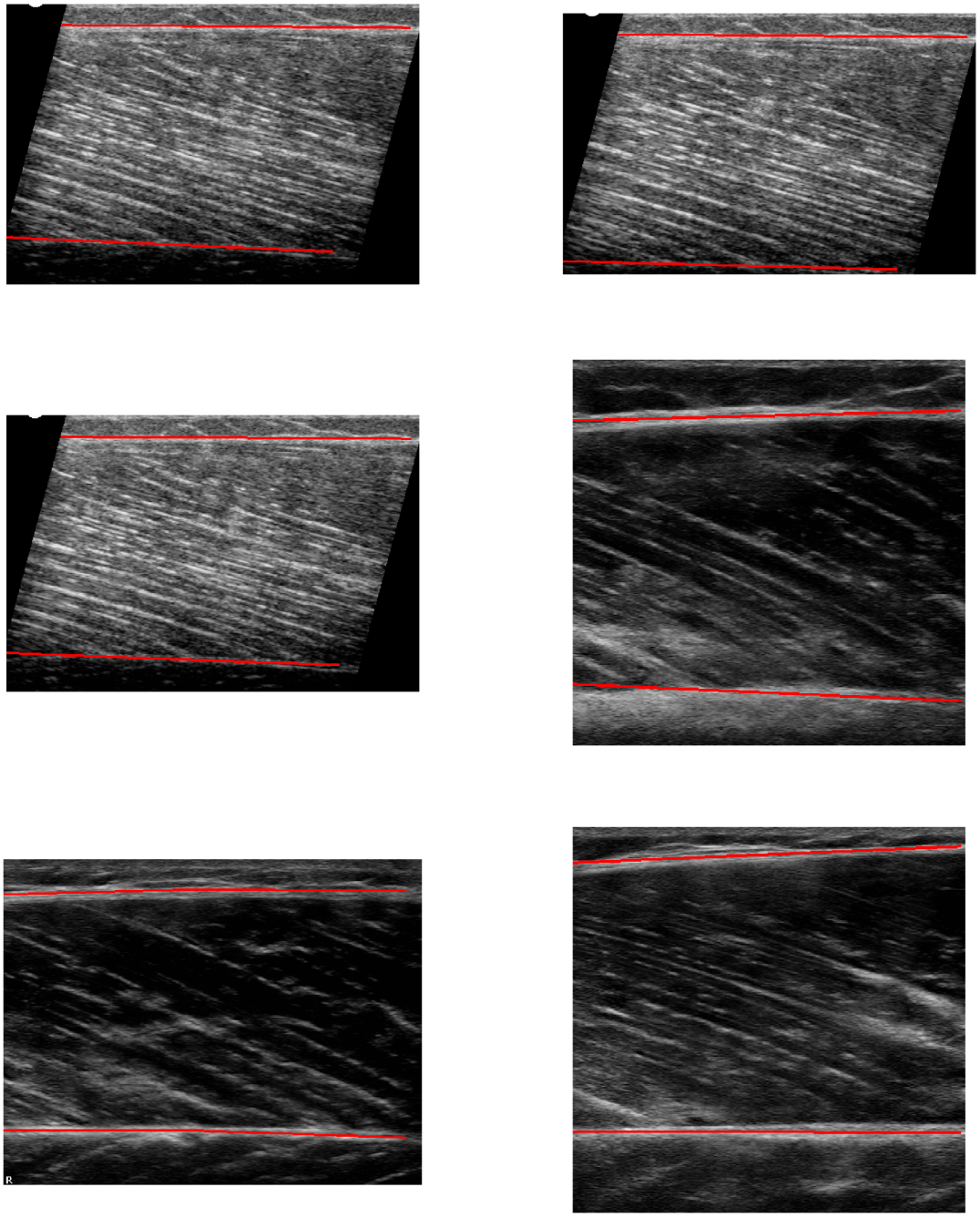
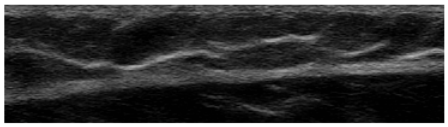
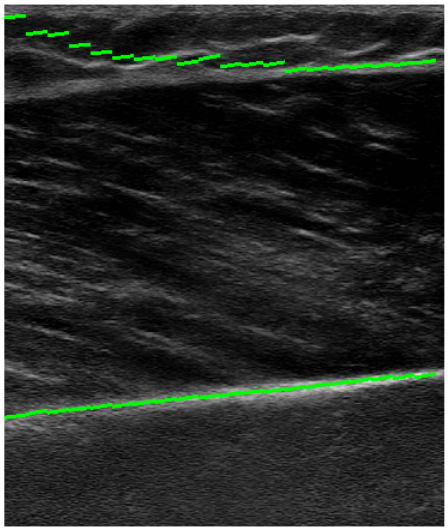


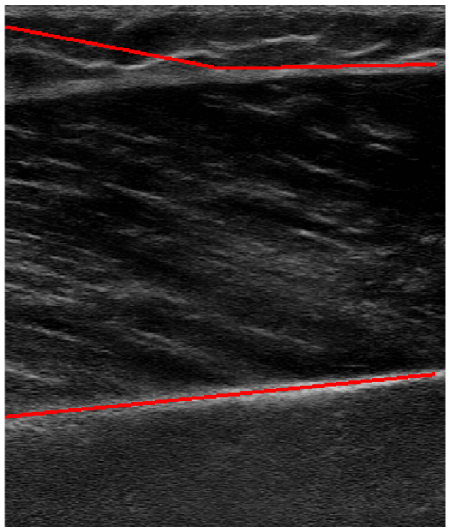
Figure 6.12: The results from detecting the aponeuroses in some example images and modelling them with spline curves.



(a)

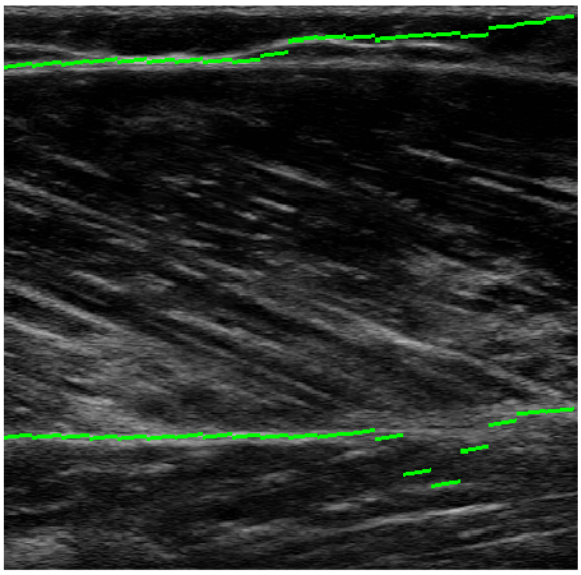


(b)

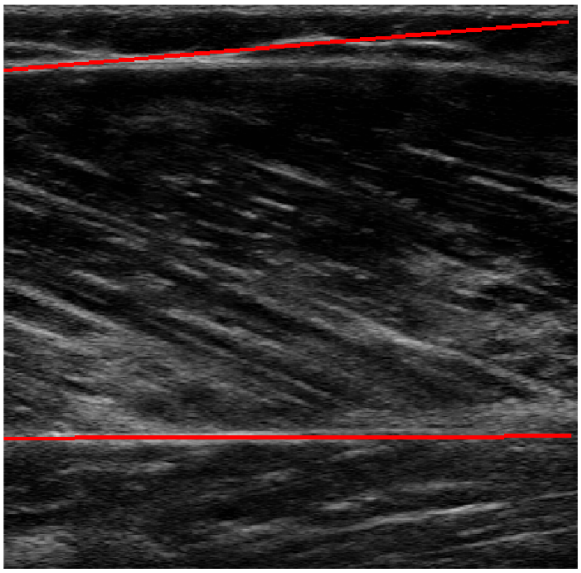


(c)

Figure 6.13: a) The area the aponeurosis was approximated to lie within, b) The line pieces located by the Radon transform and c) The resulting estimated aponeurosis.



(a)



(b)

Figure 6.14: a) The line pieces located by the Radon transform and b) The resulting estimated aponeurosis.

squares spline curve allows some error in the detection in the deep aponeurosis in this ultrasound image.

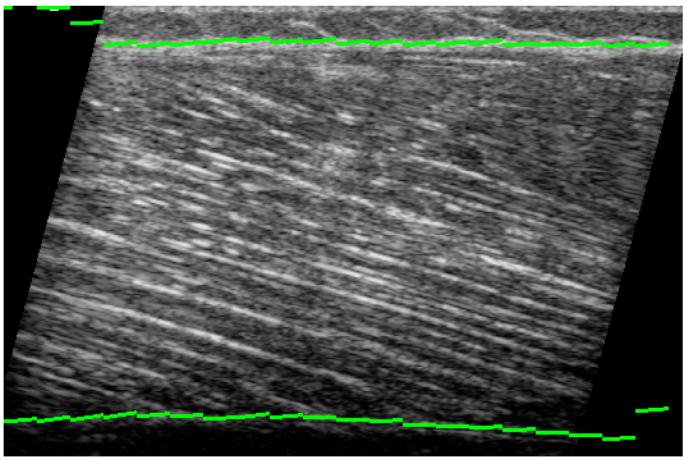
When testing on the first data set, the error is generally very small. An example of what might occur though, is presented in Figure 6.15. Here, we see that the line pieces are curving as they should in the endpoint for the deep aponeurosis to the right, but since information from the line pieces outside the parallelogram is excluded, the spline curve bends slightly less than the aponeurosis actually does. The reason the line pieces outside the image bends the correct way is because the input to the Radon transform is larger than the mere line piece. This means that even though the line piece is outside the image, the input used to calculate the line piece partly contains the edge of the parallelogram. Here, we could potentially experiment with allowing some of the data outside the parallelogram take part in creating the spline curve. However, there is a risk with using a line piece calculated on such little information. It would work in this case, but perhaps not so well in other situations.

The larger errors as in Figure 6.13 and Figure 6.14 only occur in the training set 2. Including images as Figure 6.16, we found five images in all the data sets (426 images, albeit 391 of them are quite similar) that gave errors of this size or larger, six if we count Figure 6.10.

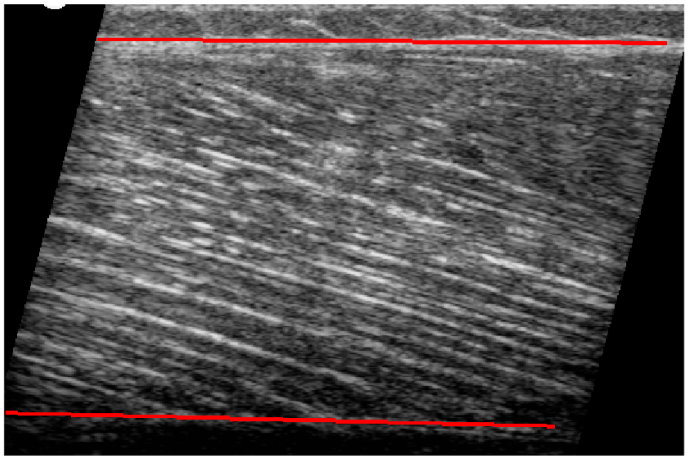
Final thoughts

In general however, we get very good results. The images with the most the largest error is, reasonably enough, from the data set that is particularily noisy. The errors seen are only occasional, and on a few images.

What we take away from this is that in general, the spline curves seems to hit the aponeurosis quite well, but the algorithm would be even better if we were able to make sure that the detected aponeurosis was a bit more precise.



(a)



(b)

Figure 6.15: a) The line pieces located by the Radon transform and b) The resulting estimated aponeurosis.

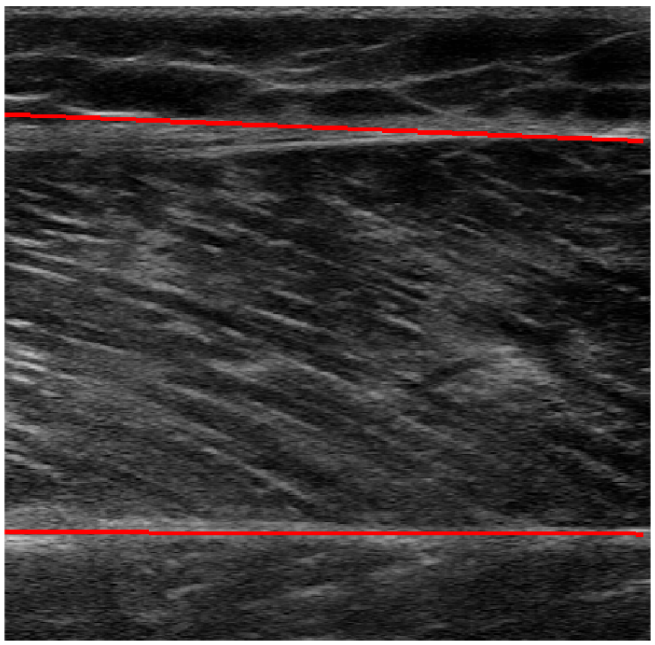


Figure 6.16: Example of a moderately large error in the detection of the superficial aponeurosis.

CHAPTER 7

Fasicle detection

After the aponeuroses are detected, and a spline has been adapted to the lines, we want to detect the dominant angles in different depths of the image in order to construct a reference fasicle. The image is cropped a few pixels below the superficial aponeurosis, and a few pixels above the deep aponeurosis so that only the muscle fascicles are in view. This is done to avoid other elements of the image when detecting the orientation of the fascicles. The area will be referred to as the fascicle plane, and the resulting image for our two main example images can be seen in Figure 7.1.

Rana, Hamarneh and Wakeling [RHW09] noted in their article from 2009 that because the Radon transform projects parallel straight lines across the image, it would be particularly suitable when identifying linear approximations to the muscle fascicles. Since we are aming to construct a reference fascicle in this chapter, we will look into this strategy.

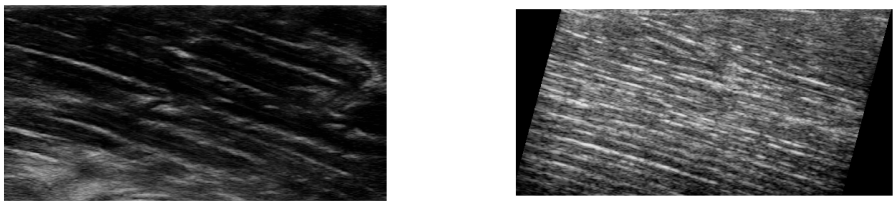


Figure 7.1: The fascicle plane for our two example images.

7.1 Detecting angles in the fascicle plane

There are many ways one can go about this step. Originally, in an attempt to minimize computing, we suggested simply dividing the fascicle plane into a grid, and doing a Radon transform on the grid cells to detect the angle in each grid cell. Then, the median of each row in the grid could be found and used as the angle for the corresponding depth of the image. However, if this strategy was chosen, one would have to have a sufficient size on each grid cell in order to get a proper result from the Radon transform. To get a sufficient size on each grid cell, then one would end up with for instance a 5×4-grid. Thus, each row in the grid would have very little data to calculate the median, and considering possible noise and outliers, it was clear this was not a robust way of proceeding. In addition to very few cells for each row, one would also end up with only about five line pieces with different angles as the end product, and that is very limiting when we want to model the curvature of the fascicle properly.

Since this method had so many limitations, we instead decided to do Radon transforms on windows around uniformly distributed points in the image. That meant we could have up to one measurement for each pixel, and have as many line pieces as wanted to model the fascicle.

Before we describe that method in further detail, we should look at some preprocessing.

Filtering

Originally, to save computation time, we wanted to see if we were able to detect the correct angles in the fascicle plane without using filters. During implementation, it became clear that some type of filtering was needed. Although the angles were mostly correctly detected, in some occasions there were very deviant angles measured that would affect the resulting angle for that part of the image.

[RHW09] used the multiscale vessel enhancement filtering to preprocess the fascicle plane before detecting the fascicle orientation. This gave good results, and we follow their footsteps.

The filter, developed by [Fra+98], is a vessel enchancement filter, used to enhance muscle fascicles. The method enhances the tubular structures in the image. As a part of the computation of the filter, the image is intitally convolved with four Gaussian kernels in which each kernel has a normalized Gaussian distribution centered within the kernel. The standard deviations 1.5, 2, 2.5 and 3 for the Gaussian distributions were used in [RHW09], and since we are looking at many of the same types of images as they did, we

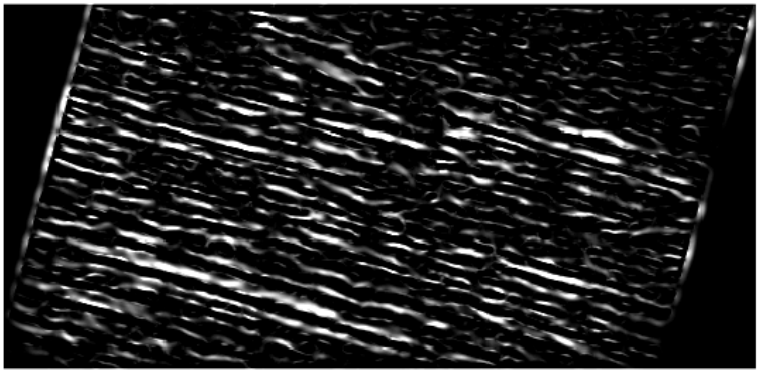


Figure 7.2: Frangis vesselness filter used on one of the images in Figure 7.1. In this image, the brightness/contrast has been adjusted slightly, to make the structures easier to see on printed paper.

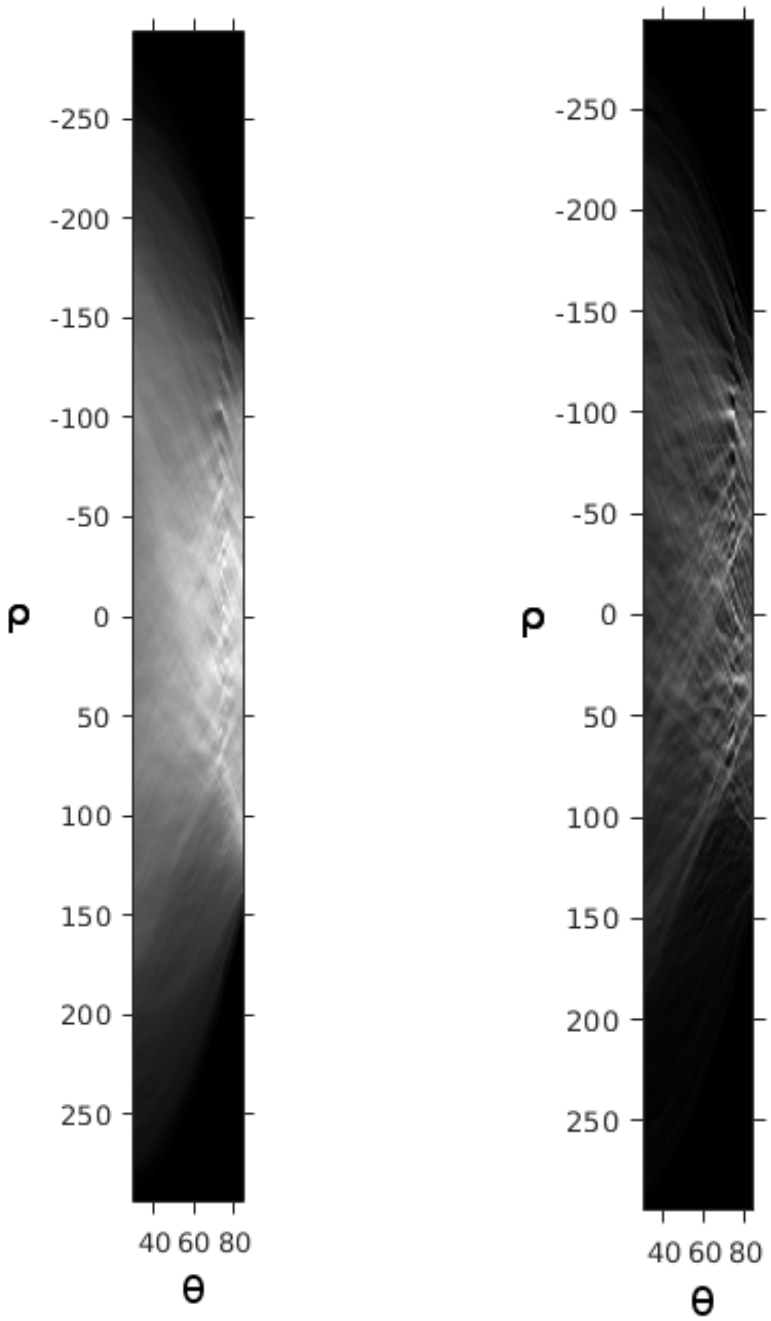
use the same standard deviations when using the filter in our algorithm. The result after using the filter on one of the images in Figure 7.1 is seen in Figure 7.2.

In Figure 7.3 we can see the result of performing a Radon transform on the parallaleogram shaped image in Figure 7.1 and the filtered image Figure 7.2, respectively. The filtered result is sharper and has clearer structures than the unfiltered result, which is a clear advantage during analysis.

Using Radon transform to detect dominant angles

With the filtered image at hand, we follow [RHW09] further, and use the Radon transform to detect angles in the fascicle plane.

We subsample points in the image on a uniformly distributed grid, and a normalized Radon transform is computed in a 30×70 window around each point. In this thesis, 1000 distinct points are used. The number is chosen because it gives enough data to work with, and at the same time it is not too computationally expensive. The transform is performed for angles $\theta \in [30, 85)$ in each window, and the angle corresponding to the column of the transformed image with the maximum variance is chosen as the dominant angle. The maximum variance is chosen as a measure because as shown by [RHW09] the Radon transform shows the greatest variance when θ approaches the orientation within the grid, as they illustrate in



(a) (b)

Figure 7.3: Examples of Radon transforms performed with angles between 35 and 85 degrees of the same image (the parallelogram shaped image in Figure 7.1), a) from unfiltered image and b) from filtered image (with a vesselness filter). ρ and θ are the polar coordinates.

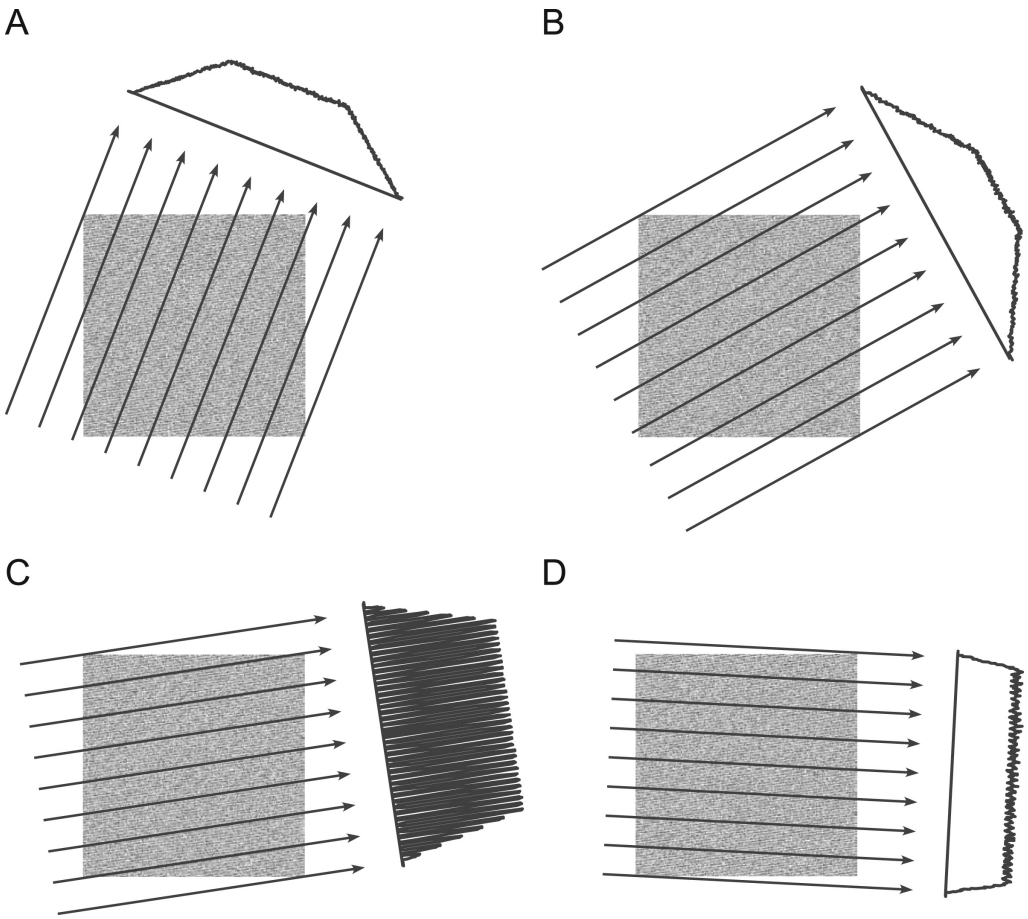


Figure 7.4: Illustration of how variance grows when θ approaches the dominant angle in the image, adapted from [RHW09].

Figure 7.4. The angles $[30, 85)$ are chosen because that is the interval we can expect to detect fascicles, and will not be disrupted by the aponeuroses even if they have not been completely cut out of the image.

To illustrate the results, we have found the median angle in groups of ten angles detected, and chosen the point with the angle closest to the median angle as the midpoint for each line. It should be pointed out that this was simply for illustration purposes, and is not used in the main algorithm. The reason this is done, is to declutter the illustrative image, and get a sense of which observations that are the most prominent. The illustration is shown in Section 7.1.

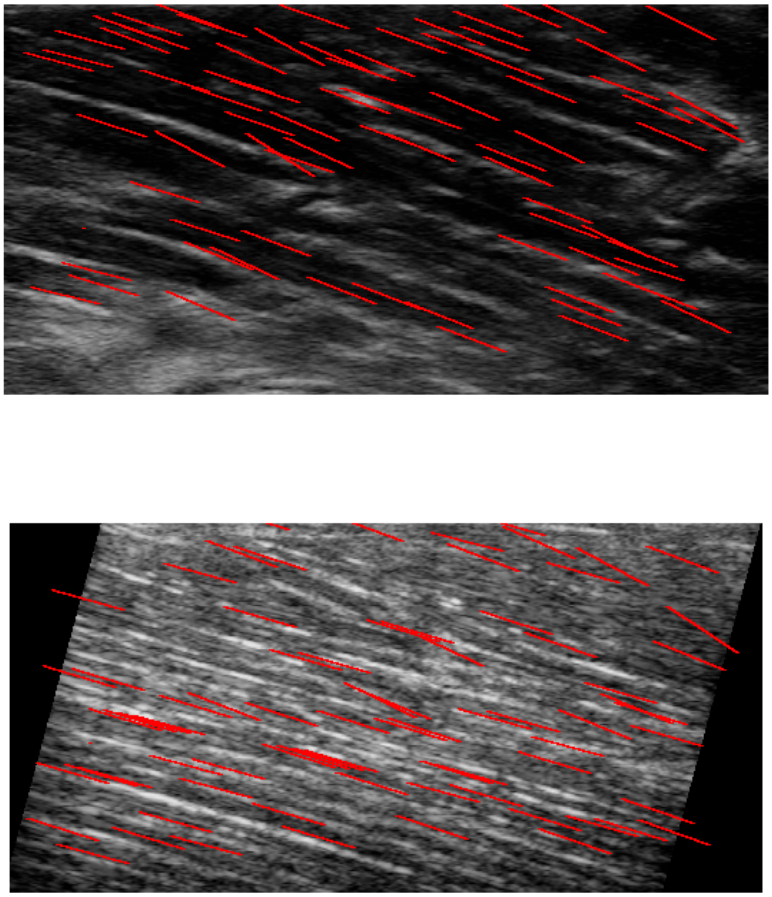


Figure 7.5: Illustration of the angles detected in two example images.

7.2 Construction of the reference fascicle

Estimating the fasicle by subdivision

To construct the reference fascicle, we divide the image into a number of rows, determine the dominant angle from the detections made around the points in each row and create a line piece for each row. In a perfect world, we would let each row be just one pixel high, attain the dominant angle for each of them, and hence construct the best possible model for the fascicle. However, then each row would get very few detections, and we would get a less robust result in the end. In order to avoid this, we partition the image

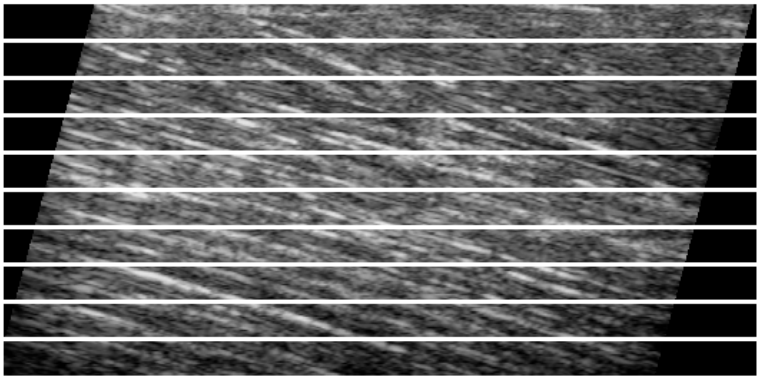


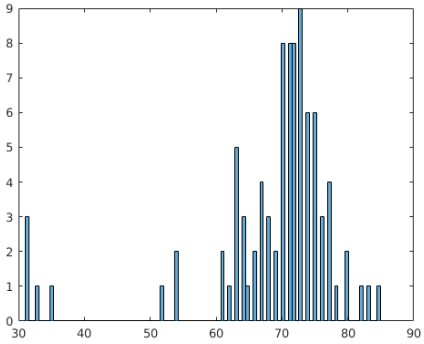
Figure 7.6: Illustration of how the fascicle plane is divided. The detections made in windows around points in an area will be used to calculate the dominant angle for that area.

into rows of greater thickness. Since the fascicles are curved gently, using the same angle over a slightly larger amount of pixels should work without compromising the shape too much. For this thesis, 10 rows have been found to give good results. The fascicle plane is thus divided as seen in Figure 7.6.

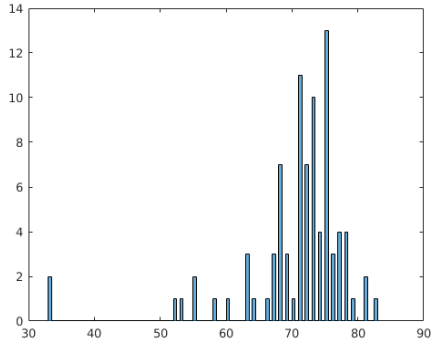
To save time, we didn't automate so that the number of rows are customized to the image size. This is something that one could, and should, automate, as a way of robustifying the algorithm. We have, however, made it so that the top and bottom row is merged into the closest row if there are very few detections in them.

In order to determine what the dominant angle is for each row, we need to decide a method that out of a set of observations chooses the intuitive peak value (which is not necessarily the maximum value). Inspecting the histograms for the observations made in each row for different images, choosing the median to locate the dominant angle should give good results. Examples of histograms for two different rows in two different images can be seen in Figure 7.7.

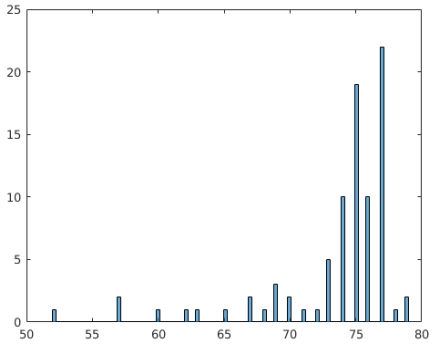
Having determined the dominant angle for each depth of the image, we are now ready to initially estimate a fascicle. The estimated fascicle is made by stitching together line pieces with the correct angle matching their depth. An example of the resulting estimated fascicles for two images is seen in Figure 7.8.



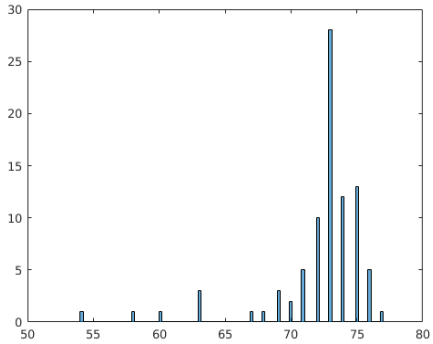
(a) Observations from the second row in the rectangular example image.



(b) Observations from the seventh row in the rectangular example image.



(c) Observations from the fourth row in the parallelogram shaped example image.



(d) Observations from the ninth row in the parallelogram shaped example image.

Figure 7.7: The detected angles within two different parts of the grid in the two example images in Figure 7.1.

Using a Quadratic Polynomial to Model the Reference Fascicle

We can see that even though the data mostly fits very well, there are some small abrupt changes in angle that makes the reference fascicle uneven, which does not fit with how the actual fascicles appear. Thus, we need to smooth out the data to create a good representative fascicle.

Observing the appearance of the fascicles in example photos, we notice they often are shaped like quadratic polynomial curves, and so it is natural to select that as a model.

To minimize the error when fitting the data to a quadratic curve, we use the least squares method to create the final reference fascicle. The resulting

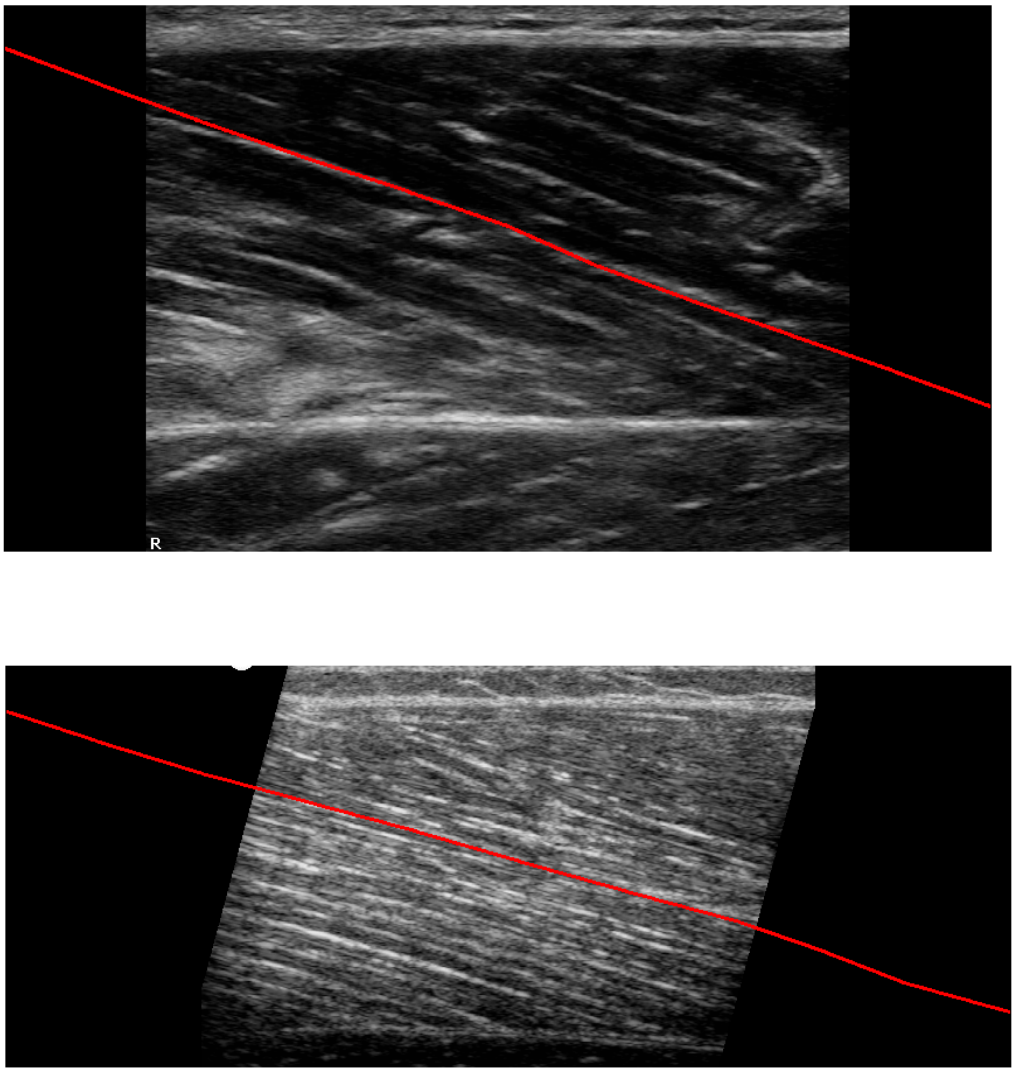


Figure 7.8: Two example images with their estimated fascicles.

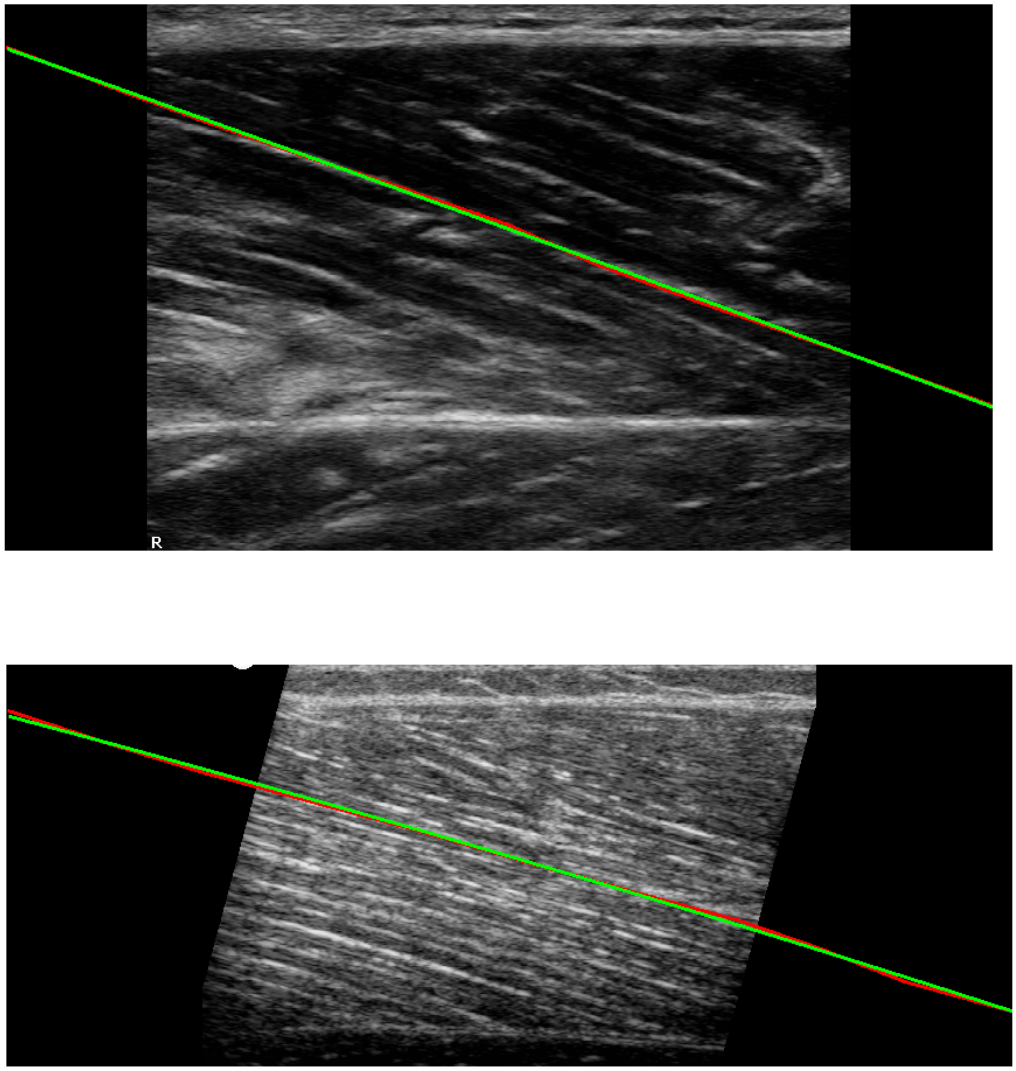


Figure 7.9: The resulting reference fascicles constructed in two example images. The green curve is the quadratic polynomial, while the underlying red curve is the data the polynomial is based on.

reference fasicle for the example images can be seen in Figure 7.9.

7.3 Evaluation of fascicle detection and modelling

Summary of algorithm

- The image is cropped above the deep aponeurosis and below the superficial aponeurosis, this is the fascicle plane.
- The image is filtered using Frangis vesselness filter ([Fra+98]).
- The fascicle orientation is detected in 1000 uniformly distributed points by using the Radon transform. Note: In this step, the pennation angle is also detected – see Chapter 8.
- The fascicle plane is divided into 10 rows.
- The median angle for each row is found from the detected angles in the points belonging to the row.
- An initial reference fascicle is created by stitching together line pieces with the median angle matching their row.
- The data from the initial reference fascicle is fit to a quadratic polynomial using the least squares approximation.

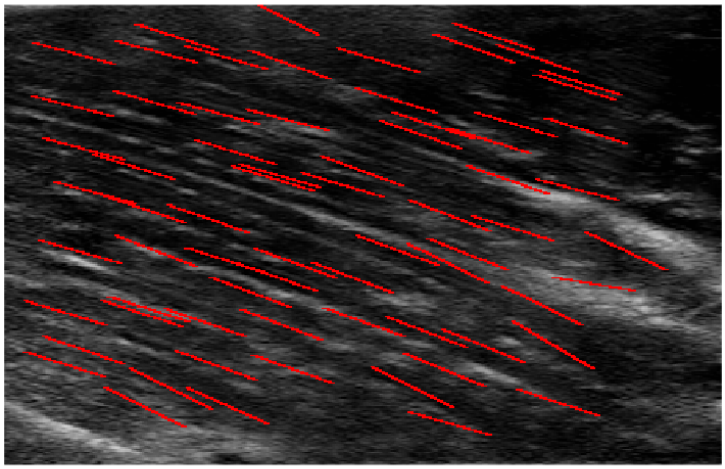
Detection of angles

When testing this part of the algorithm on the data set, we get angles that represent the angle of the underlying structure, almost without exceptions. This part of the algorithm seems to function very well, and even the images with the largest amount of erronous detections we could findas, the results still has the potential of producing a good reference fascicle.

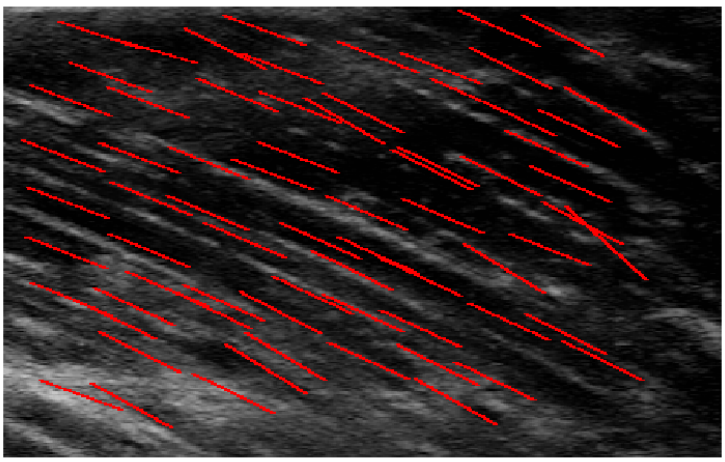
As mentioned previously, we have found the median angle in groups of ten angles detected, and chosen the point with the observed angle closest to the median angle as the midpoint for each line as a way of illustration.

Some typical results is shown in Figure 7.10 and Figure 7.11.

Now, let us have a look at some images where there are a few misdetected angles, and discuss these situations. From our data sets, the images shown in Figure 7.12 were the result of attempting to pick out some under-par results. However, even they provide very decent results, and might give us

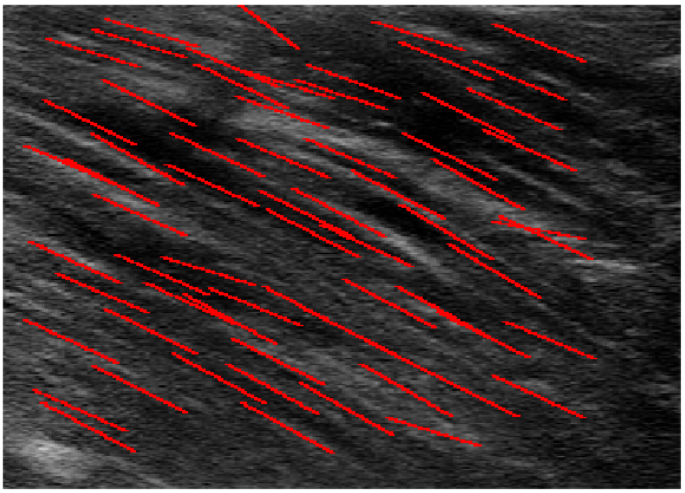


(a)

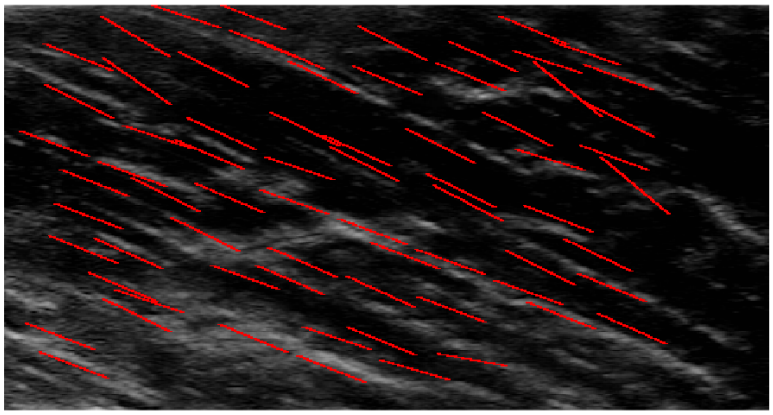


(b)

Figure 7.10: Images describing the median angles located in two example images, each line depicted based on 10 detected angles.



(a)



(b)

Figure 7.11: Images describing the median angles located in two example images, each line depicted based on 10 detected angles.

a good reference angle because the angles in general are good. This again points to this part of the algorithm being well-functioning.

Not surprisingly, we can see that the median angles detected in homogenous areas, noisy areas, and areas without clear structure, sometimes become off. Keeping in mind that these observations are based on the median angle of ten observed angles, we realize that the erronous angles in this image is based on many underlying erronous observations.

We take a look at some example images from the other data set in Figure 7.13. These images have clearer structures, but as previously, we notice that the edges of these images may be problematic some places, since the detected angles are based on just small pieces of data. However, as we see here, this is not a very big problem. Only very few of the lines on the edges are bothered by this problem, and it is doubtful that this will produce problems when creating a reference fascicle.

In all images tested from this data set, we get the same type of result as in Figure 7.13, or better.

As mentioned, in areas with a lot of noise, or lack of structure, the detections may be erronous, but since there usually are good detections in the rest of the image, it seems this part of the algorithm works very well. The only potential problem is if a large part of the fascicle plane in the same depth get erronous detections, but this does not appear to be a problem when considering these images.

Modelling

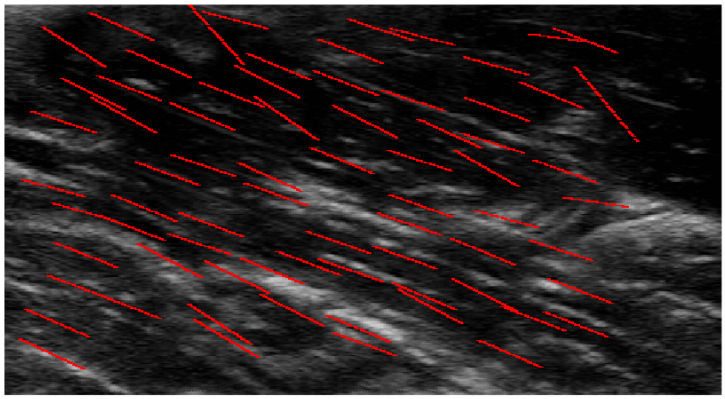
First, let us have a look at the resulting reference fascicles constructed from the angles detected in Figure 7.12. They can be seen in Figure 7.14. The curves appears to represent the curvature of the fascicles quite well in most of the image, but we can see that there is some disagreement with the intuitive angle near the deep aponeurosis.

In all images in this subsection, the green curve is the quadratic least squares approximation to the red stitched lines underneath it.

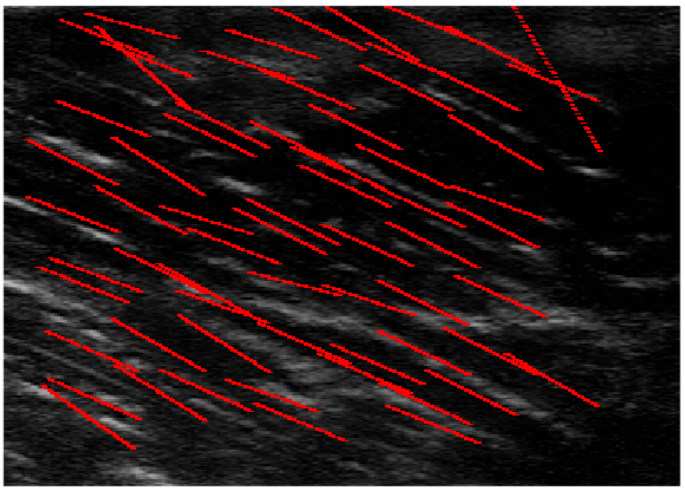
It turns out that inaccuracy close to the superficial and the deep aponeurosis is a problem in this dataset, and in Figure 7.15 we can see one examples of this. The angles close to the superficial and deep aponeurosis are in such a state that the quadratic curve is convex rather than concave. We can spot tendencies of this in other images as well, but not as severe as in this image.

Mostly, the curve agrees with what one would expect though, and examples of good reference fascicle curves can be seen in Figure 7.16.

On the first data set, the areas close to the aponeuroses still have clear structure, and as a direct result, the reference fascicles match very well with

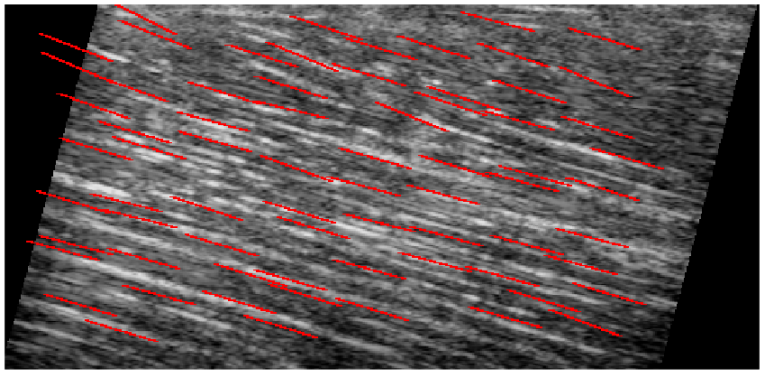


(a)

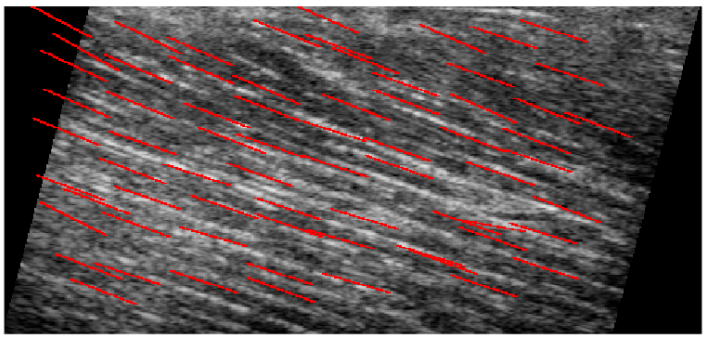


(b)

Figure 7.12: Images describing the median angles located in two example images, each line depicted based on 10 detected angles. Manually inspecting the images, these were the two images with the worst results found.

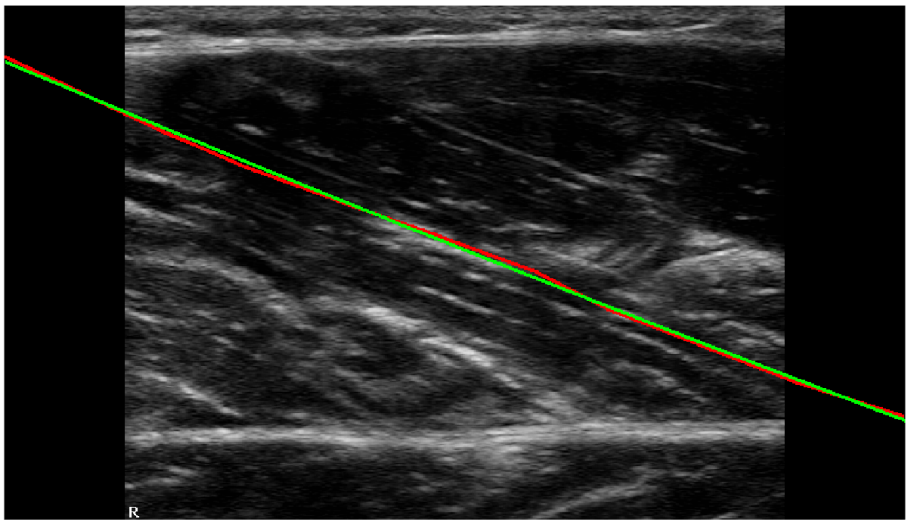


(a)

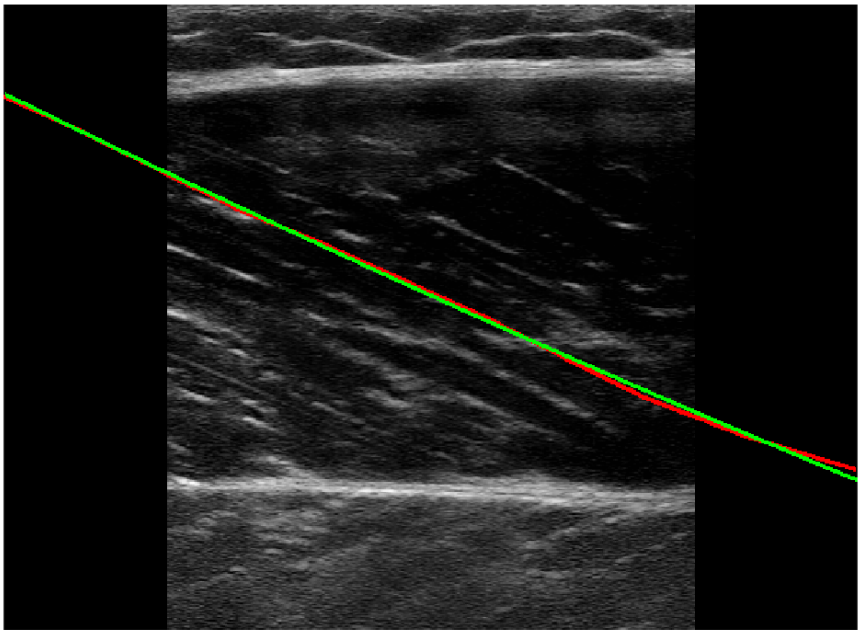


(b)

Figure 7.13: Images describing the median angles located in two example images, each line depicted based on 10 detected angles.



(a)



(b)

Figure 7.14: The resulting reference fascicles constructed from the fascicle plane shown in Figure 7.12. The green curve is the quadratic polynomial, while the underlying red curve is the data the polynomial is based on.

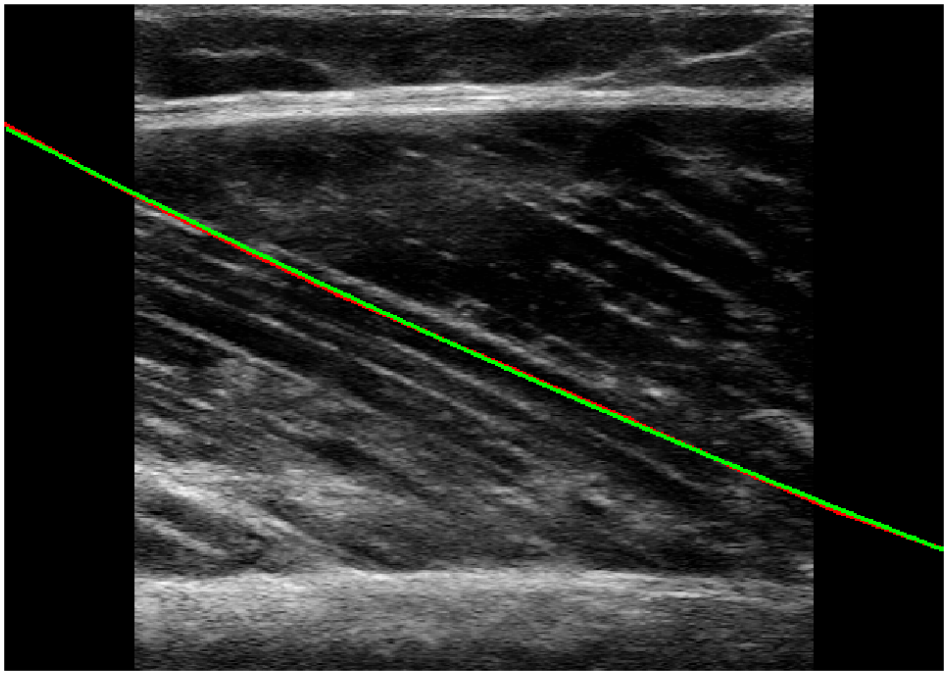


Figure 7.15: The resulting reference fascicles constructed from the fascicle plane shown in Figure 7.12 b).

the curvature in the image. No bad results were found in this data set. Examples of results from this data set is shown in Figure 7.17.

If one or more of the detected aponeuroses are wrongfully detected, then there are two possible scenarios for this step. One scenario is that we get an area including one or more aponeuroses, which can cause the algorithm to detect angles closer to the angle of the aponeurosis, and which will cause issues with the constructed reference fascicle. Another scenario is that we get an area much smaller than it should be as the fascicle plane. In those cases we have two problems. One problem is that each partition of 10 partition the fascicle plane is divided into will be smaller, and as a result we might get less representative angles in those areas. A second issue is that we do not get data for the areas that are cropped away, and the quadratic polynomial representing the reference fascicle might not give a proper representation. Finally, if the aponeuroses are wrongfully detected, clearly we will not get a correct length estimation of the reference fascicle, which is the main point of creating one.

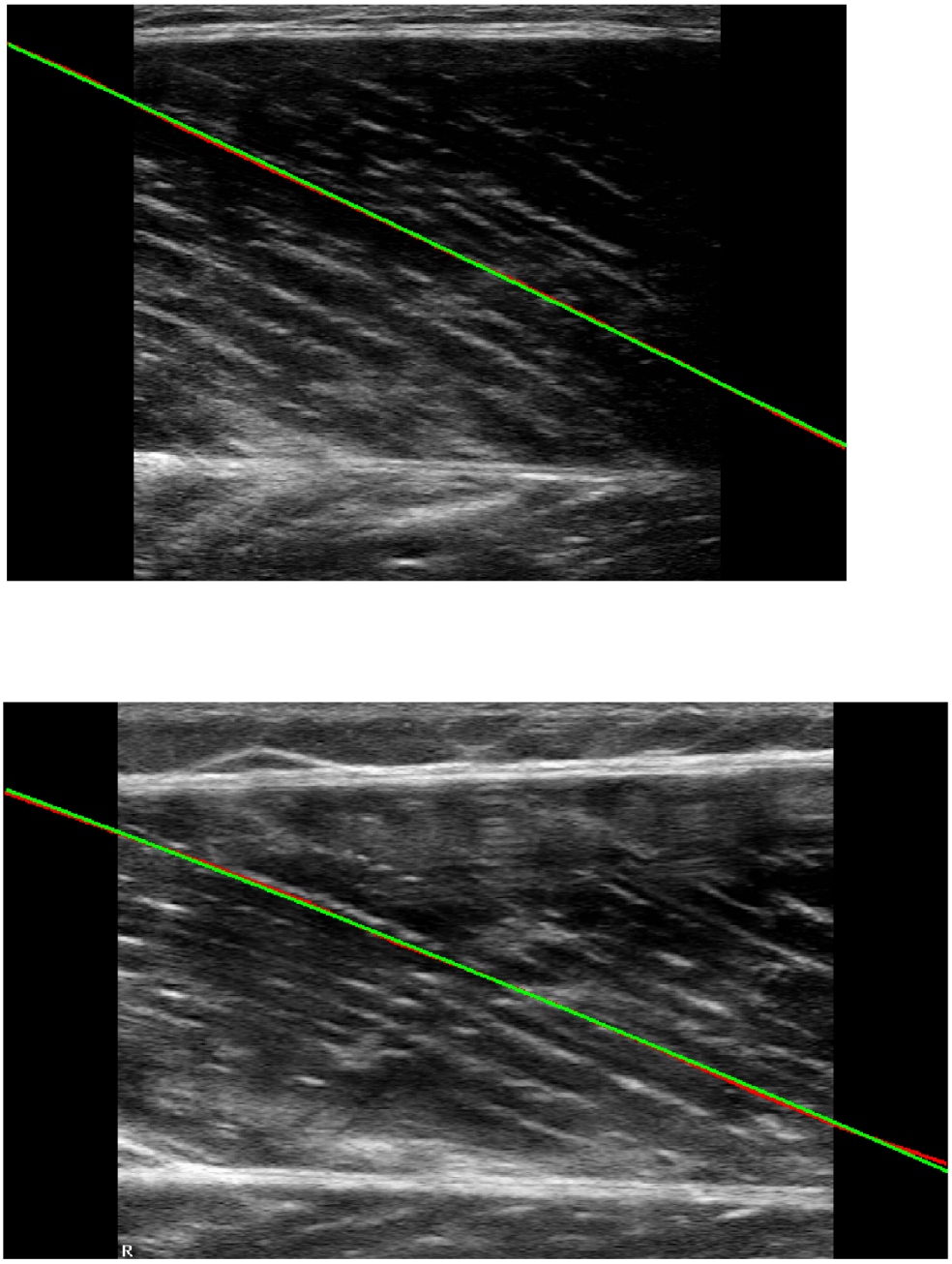


Figure 7.16: Examples of images where the reference fascicles constructed matches well with our intuitive understanding of the curvature of the fascicles. The green curve is the quadratic polynomial, while the underlying red curve is the data the polynomial is based on.

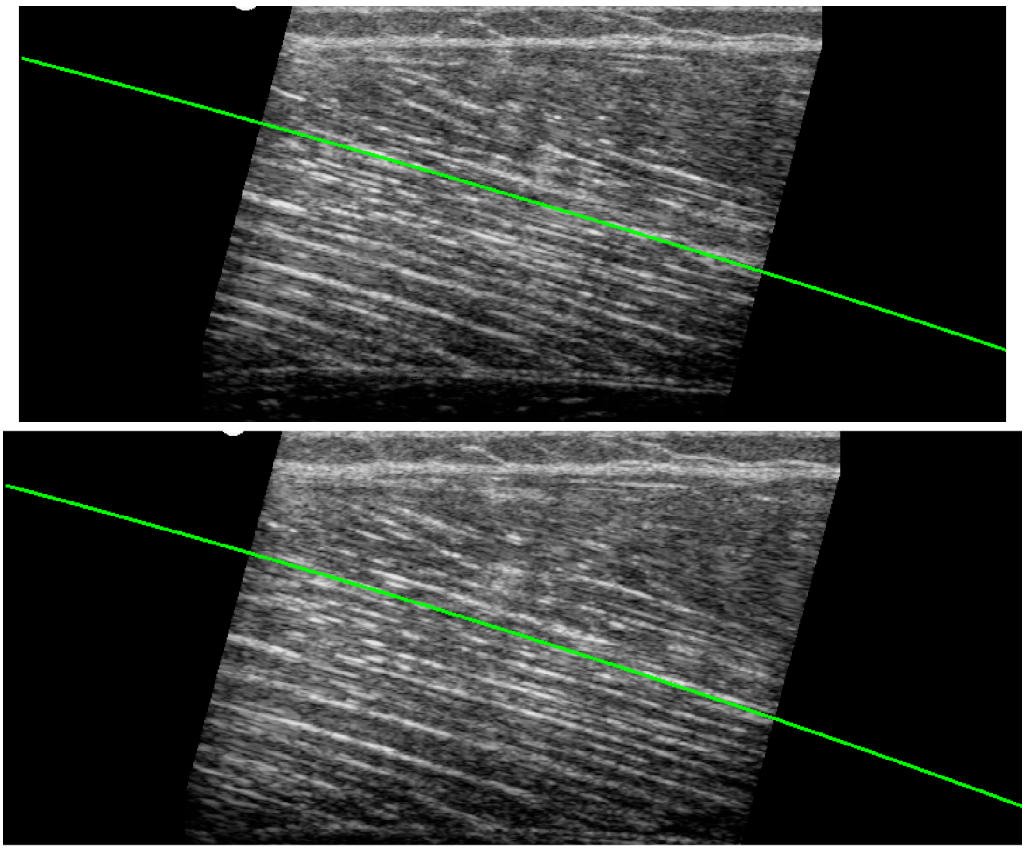


Figure 7.17: Examples of images where the reference fascicles constructed matches well with our intuitive understanding of the curvature of the fascicles.

CHAPTER 8

Fascicle length and pennation angle estimation

8.1 Estimating the length of the reference fascicle

The fascicle length is one of the crucial muscle architecture parameters for understanding the contraction mechanics and pathological conditions of muscles ([ZZ15]). Thus, the natural next step in the algorithm now that we have created a reference fascicle, is to find a method to measure the length.

Since we have a quadratic polynomial to represent the fascicle curvature, the natural course of action is to use this representation to estimate the fascicle length.

Arc length

From Calculus ([Lin06]), we know we can calculate the arc length of any $\mathbf{C}^1$ function $f : [x_1, x_2] \to \mathbf{R}$. The length is estimated as

$$L = \int_{x_1}^{x_2} \sqrt{1 + f'(x)^2} dx \tag{8.1}$$

In this case, we have a quadratic polynomial $f(x) = ax^2 + bx + c$, so $f'(x) = 2ax + b$. Thus, we get the arc length of the curve to be

$$L = \int_a^b \sqrt{1 + (2ax + b)^2} dx \tag{8.2}$$

To use this formula, we also need to determine the endpoints x_1 and x_2.

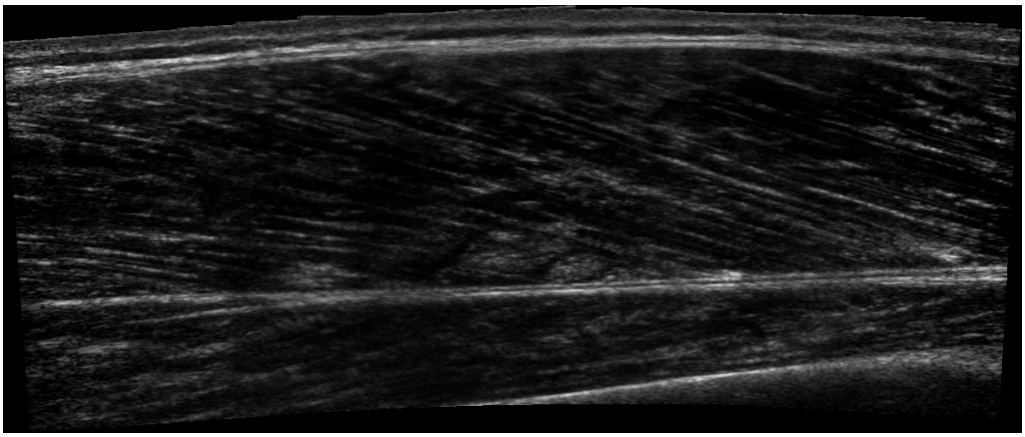


Figure 8.1: Ultrasound images stitched together to give an image of a larger part of the muscle.

Extrapolating the aponeuroses

Since most fascicles are not entirely in the view of one ultrasound image, we need to make some assumptions. First, we need to assume that an extrapolation of the reference fascicle will mimic how the muscle appears outside the bounds of the image. Secondly, we need to assume that an extrapolation of the aponeuroses will mimic how the aponeurosis appears outside the bounds of the image.

Given that the algorithm has estimated a reference fascicle that fits well with the structures in the image we can observe, and using what we know of muscle architecture, it is not probable that the shape of the fascicle will suddenly change outside the imaged area. Thus, the first assumption is deemed to be rather safe.

The second assumption might be slightly more controversial. Spline curves, while having many good qualities, are not good for extrapolation in this case. The extrapolation would not give us a satisfactory result when approximating the continuation of the aponeuroses. Instead, we have chosen to extrapolate the data we already have from the spline curves, with a quadratic polynomial. This is the approximation we can think of that will give the most accurate results when finding endpoints for the reference fascicle.

In Figure 8.1, we can see ultrasound images stitched together, which gives us an idea of how the architecture of a larger piece of muscle appears.

Finally, having polynomials $p1(x)$ and $p2(x)$ approximating the aponeuroses, and the polynomial $r(x)$ as the reference fascicle, we can finally locate the endpoints by solving the equations

$$p1(x) - r(x) = 0 \tag{8.3}$$
$$p2(x) - r(x) = 0 \tag{8.4}$$

These are simple quadratic equations, and we find the endpoints by finding the roots of the equations.

Because we know the polynomials are very gently curved, the root we want for each equation is the one closest to zero. Thus, for each equation, we find the end or start point by choosing the root with the smallest absolute value.

8.2 Estimating the pennation angle

Since the pennation angle may vary in different parts of the image, we need to determine which part of the fascicle plane we should estimate the pennation angle from. In addition, we also need to correct for the angle of the aponeurosis. In this section, we will first discuss how to calculate the pennation angle without accounting for the deep aponeurosis, and then calculate the angle of the deep aponeurosis to find the final pennation angle.

The pennation angle with respect to the horizontal line

The manual measurements acquired from the Norwegian School of Sport Sciences, were measured within the lower third of the muscle thickness. In order to make the results as comparable to the manual measurements as possible, the pennation angle should also be estimated within this area. In addition, because the angles might be more uncertain near the edges, the area should not extend all the way to the edges of the image.

To fulfill these requirements, we choose to estimate the pennation angle from the angles measured in a window consisting of all points located in the middle third in the horisontal direction, and the lower third of the fascicle plane in the vertical direction. In addition, the lower border of the window is moved $\frac{1}{12}$ of the fascicle plane upwards, while the upper border of the window is moved by the same amount downwards. The last step is performed since the angles are calculated in an area of 30×70 around the points inside the window. An example of how such a window appears, is shown in Figure 8.2.

This part of the algorithm is done while measuring the angles in the rest of the image as well, as a part of determining the dominant angles in the fascicle detection described in Chapter 7. That means that in the same

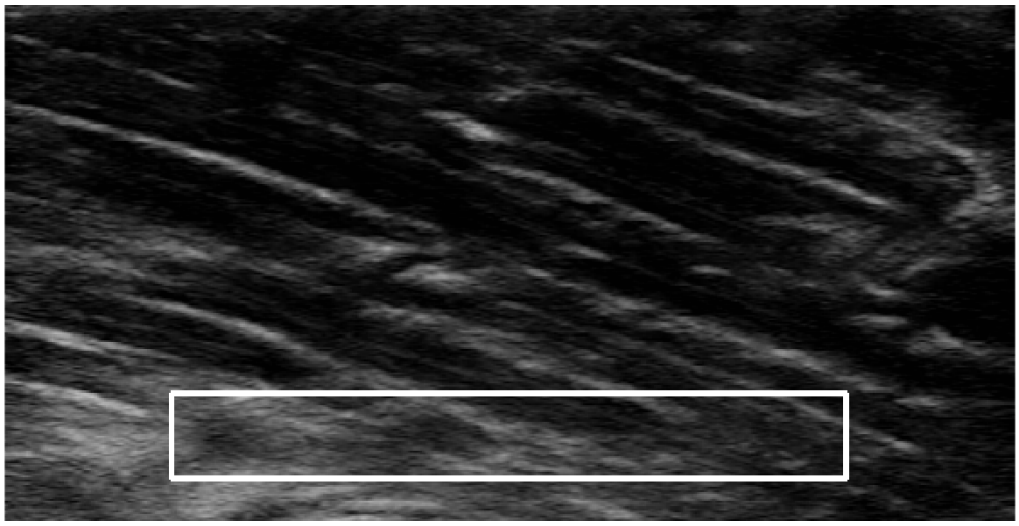


Figure 8.2: Example of where the window we search for the pennation angle in is located.

manner as in that chapter, the image is filtered with the vesselness filter, and each measurement is calculated in a 30×70 window around the points in question, and the normalized Radon transform is performed to locate an angle in each 30×70 window.

That procedure gives us about 100 measurements to determine the pennation angle from. As previously, we use the median to find the peak value in the area, and use the resulting value as the pennation angle.

However, this pennation angle is only valid if the deep aponeurosis is completely horizontal. It rarely is, so we need to correct the pennation angle with respect to the angle of the aponeurosis.

The pennation angle with respect to the deep aponeurosis

Since we have chosen to model the aponeurosis as a quadratic curve, we need to decide on a method to determine the angle of the aponeurosis with regards to the pennation angle.

To do this, two approaches were used. One approach was to simply fit a line to the quadratic curve, only using the middle 90% of the data, and calculate the angle of this line compared to the horizontal line. This is the method they use at the Norwegian School of Sport Sciences. The second approach was to only fit a line to the part of the quadratic curve below the window defined above. Both these approaches seemed to give

plausible results, and in many images there was little difference between the two variations. In the end, we chose to perform the measurements using the approach they used at NIH, in order to have comparable data.

Finally, the pennation angle was found by calculating the angle between the line determined from the window, and the line fitted to the deep aponeurosis.

CHAPTER 9

Results

In this chapter, we try to evaluate the algorithm as a whole on test set 1 and 2. They have not been used to develop the algorithm.

9.1 How to evaluate automatic estimates

These types of automatic measurements are typically very difficult to evaluate directly against manual measurements. This is because there are large differences in measurements from person to person when measuring manually. The reason for this is that there are very small margins here, and during a manual measurement, there may be little difference to the naked eye between for instance 14° and 15° when adapting a line to the fascicle orientation. Because of this, direct comparisons against manual measurements cannot be seen as the answer to whether or not the method produces good results.

But because of the big variation when measuring manually, automatic measurements are important since they only give one result on one objective.

As a first step of evaluating the results, we have performed a visual analysis of the results in test set 1. Secondly, we have used the algorithm on test set 2, and attempted to draw some conclusions from the results by comparing with the manual analysis from the set. Finally, we have tested the detection of fascicle orientation on two ultrasound images of hair, where we know the exact angle.

9.2 Visual analysis of results

Because of the difficulty of evaluating the results, visual analysis of the image is a part of the evaluation of the method. In the following section, we will

Deep Apo.:	How well the spline curve fits the deep aponeurosis
Superficial Apo.:	How well the spline curve fits the superficial aponeurosis.
Penn. Angle:	How well the line created by the pennation angle detected fits with the orientation of the muscle fascicles in the surrounding area.
Ref. Fasc:	How well the reference fascicle represents the structure of he muscle fascicles.

Table 9.1: Explanations for the columns in Table 9.2.

Image	Deep Apo.	Superficial Apo.	Penn. Angle	Ref. Fascicle
1	GOOD	GOOD	GOOD	OK
2	GOOD	GOOD	GOOD	GOOD
3	GOOD	OK	GOOD	GOOD
4	GOOD	GOOD	GOOD	GOOD
5	GOOD	GOOD	GOOD	GOOD
6	GOOD	GOOD	BAD	GOOD
7	GOOD	GOOD	OK	GOOD
8	GOOD	GOOD	OK	OK
9	GOOD	GOOD	BAD	GOOD
10	GOOD	GOOD	GOOD	OK
11	GOOD	GOOD	GOOD	OK

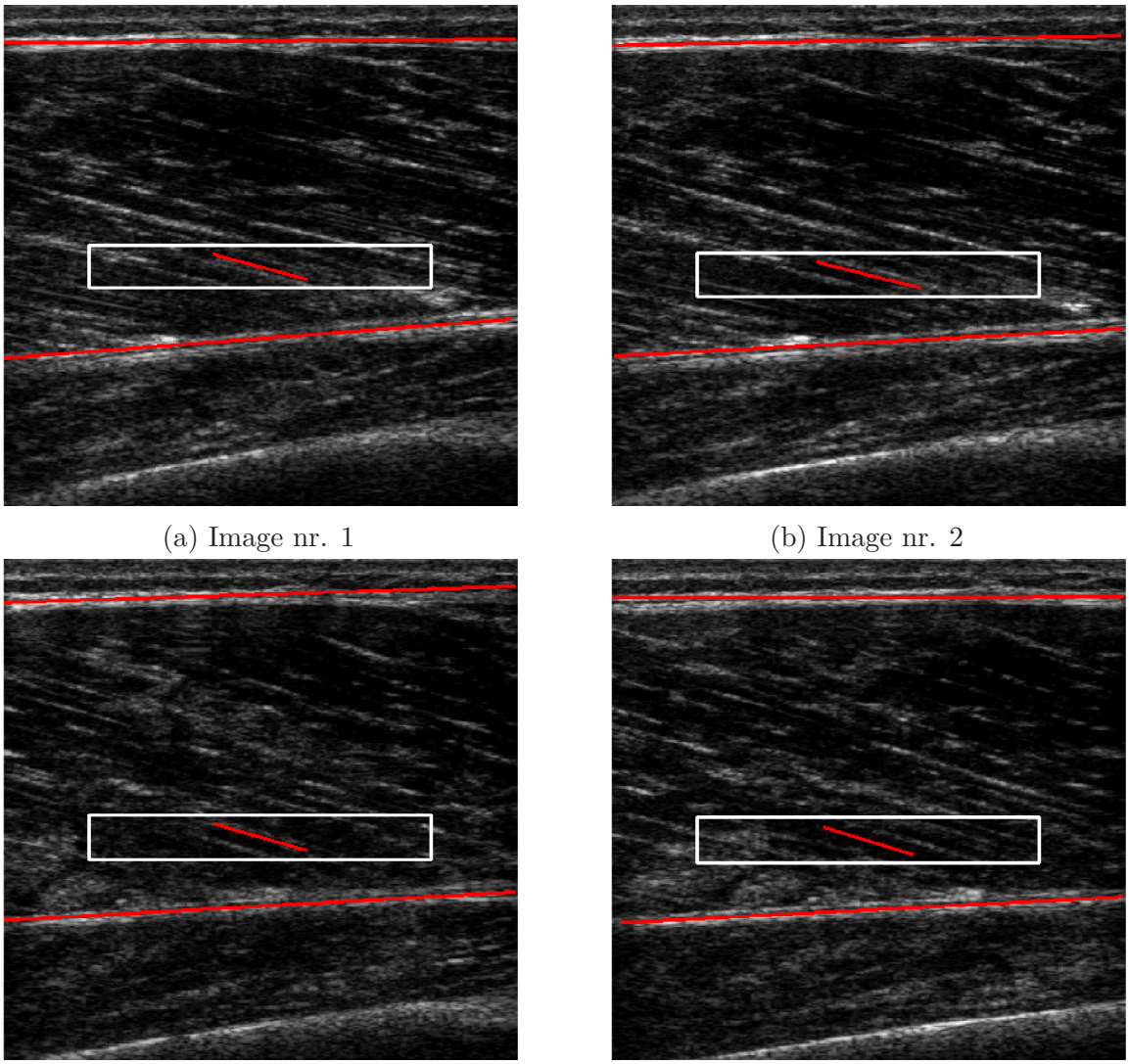
Table 9.2: Visual analysis of results in image nr. 1-11 in the test set.

present results where we have analyzed the algorithm by visual inspection on test set 1.

We present Table 9.2, where four aspects are evaluated on a test set of 11 images. Explanation for the four aspects in the table are in Table 9.1.

The categories are evaluated in terms of GOOD, OK and BAD. GOOD is a complete/almost complete match, OK is mostly a match with some issues, and BAD is an unusable result.

The images with the results for the aponeuroses and pennation angle is shown in Figure 9.1 (nr. 1–4), Figure 9.2 (nr. 5–8) and Figure 9.3 (nr. 9–11). The images with the images with the constructed reference fascicles is shown in Figure 9.4 (nr. 1–3), Figure 9.5 (nr. 4–6), Figure 9.6 (nr. 7–9), and Figure 9.7 (nr. 10–11).



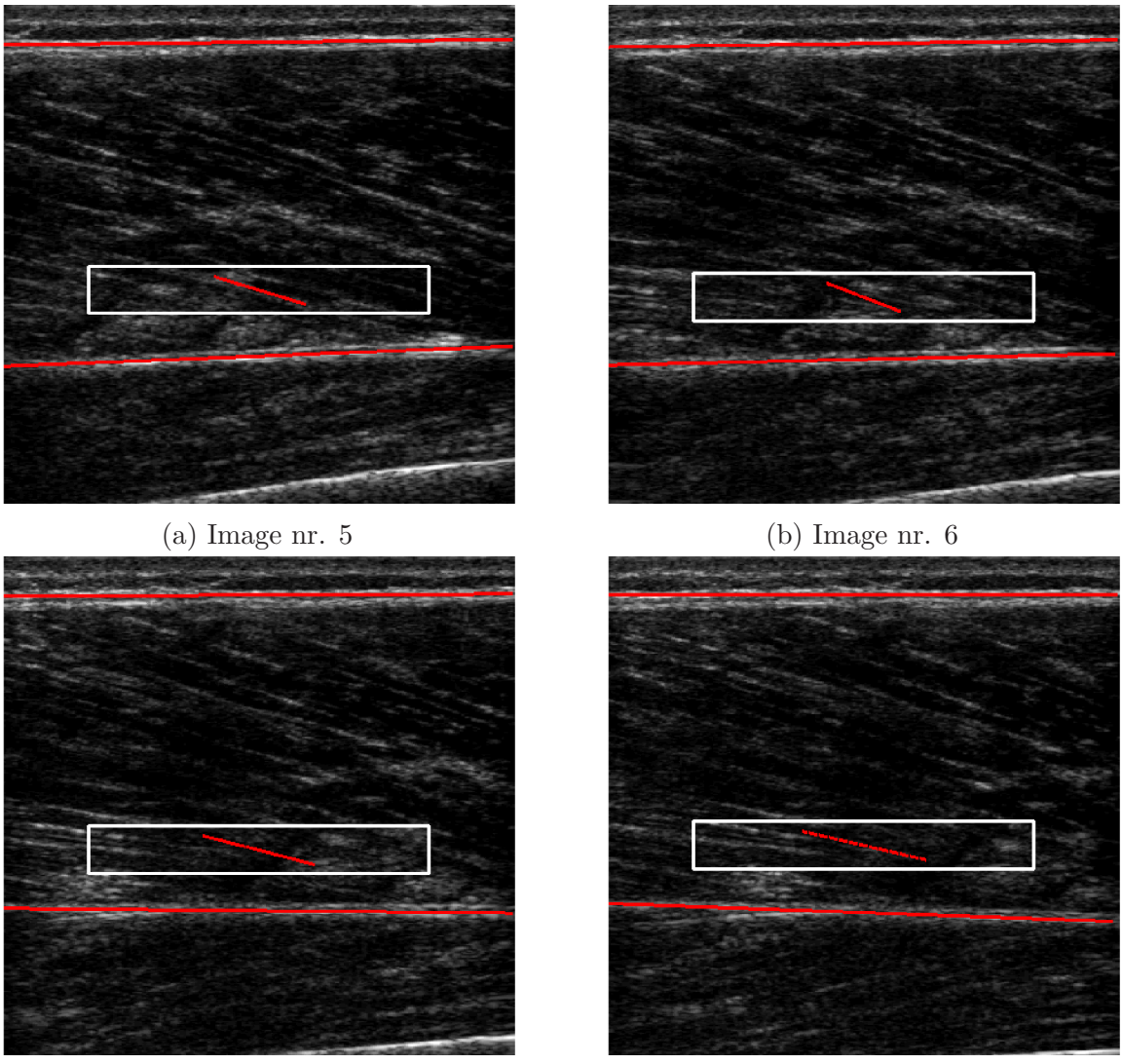
(a) Image nr. 1

(b) Image nr. 2

(c) Image nr. 3

(d) Image nr. 4

Figure 9.1: Detected aponeuroses and observed pennation angle in images nr. 1-4.



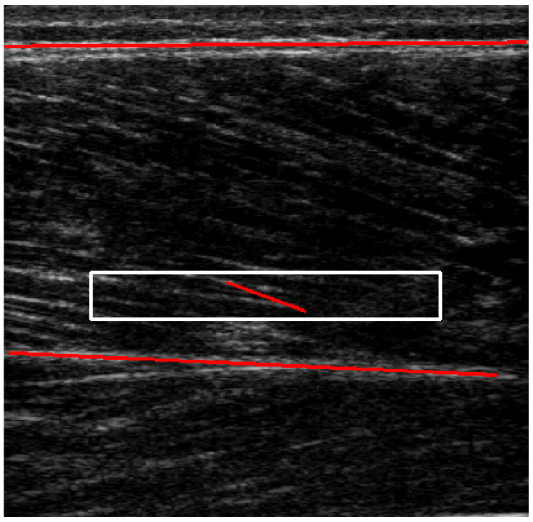
(a) Image nr. 5

(b) Image nr. 6

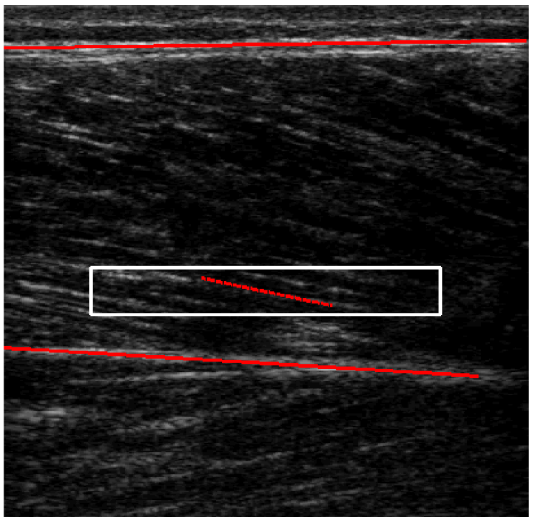
(c) Image nr. 7

(d) Image nr. 8

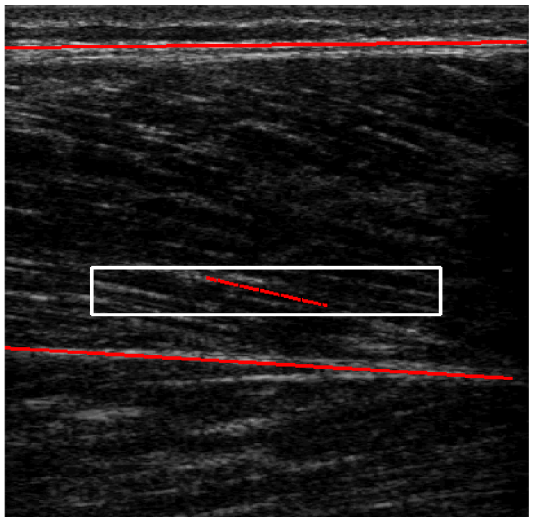
Figure 9.2: Detected aponeuroses and observed pennation angle in images nr. 5-8.



(a) Image nr. 9

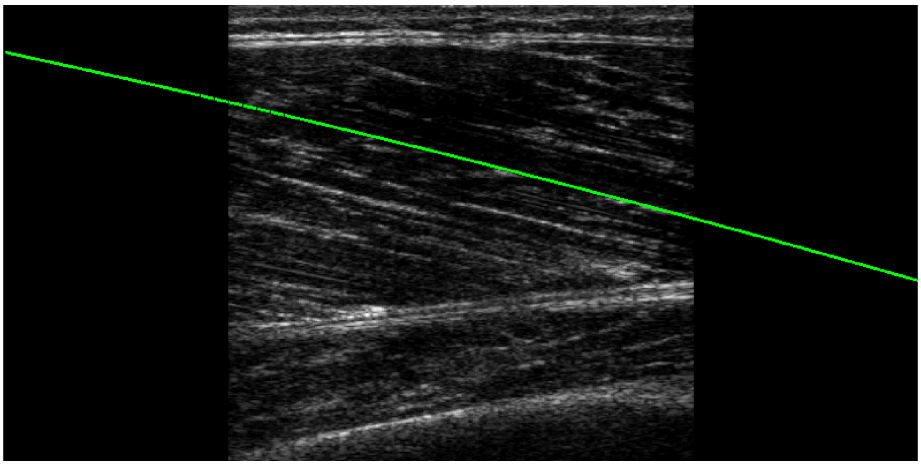


(b) Image nr. 10

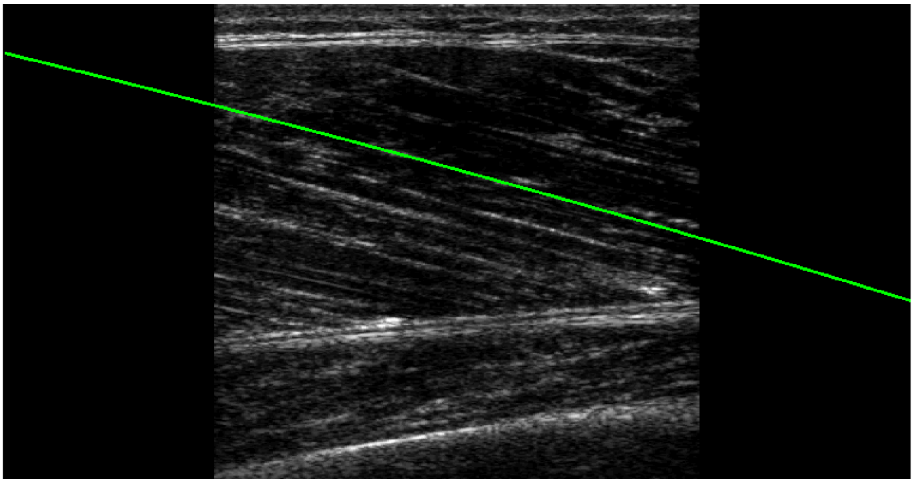


(c) Image nr. 11

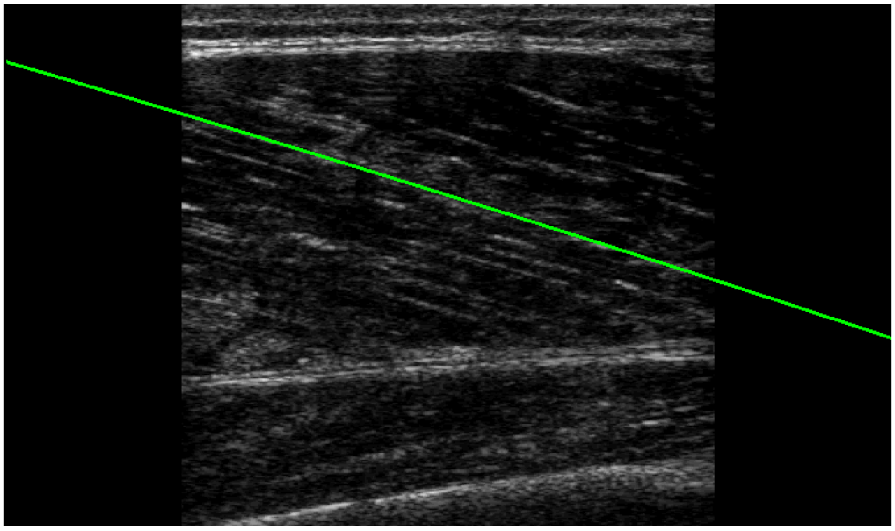
Figure 9.3: Detected aponeuroses and observed pennation angle in images nr. 9-11.



(a) Image nr. 1

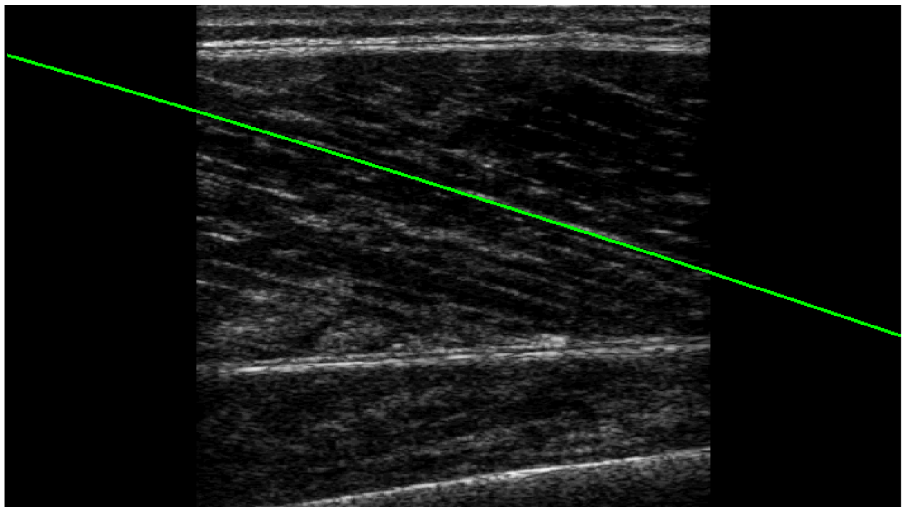


(b) Image nr. 2

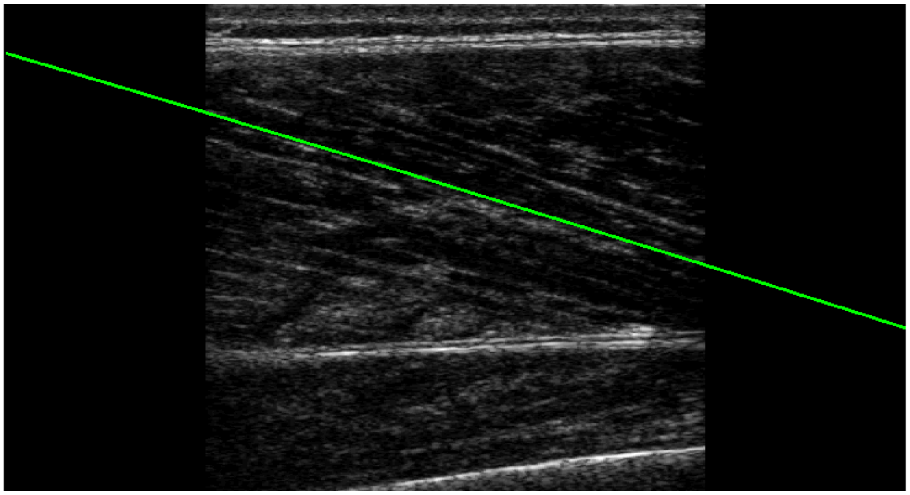


(c) Image nr. 3

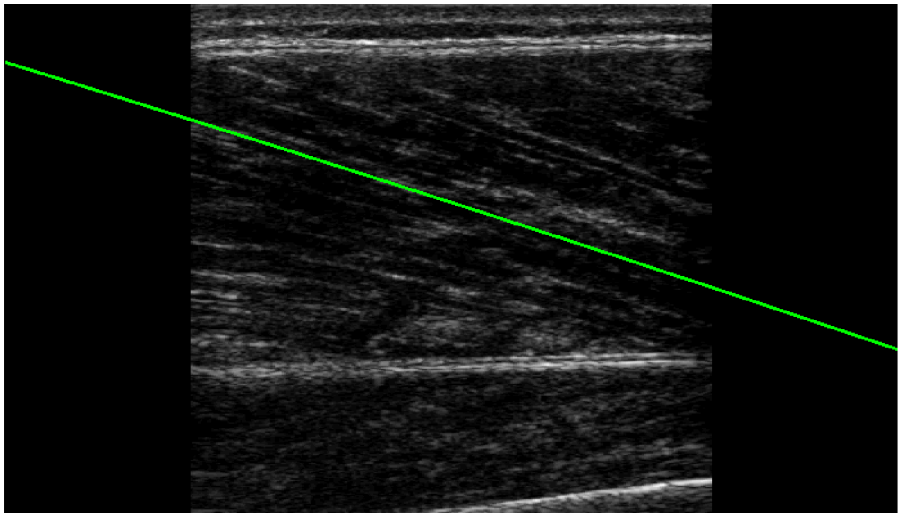
Figure 9.4: Constructed reference fascicles in images nr. 1-3.



(a) Image nr. 4

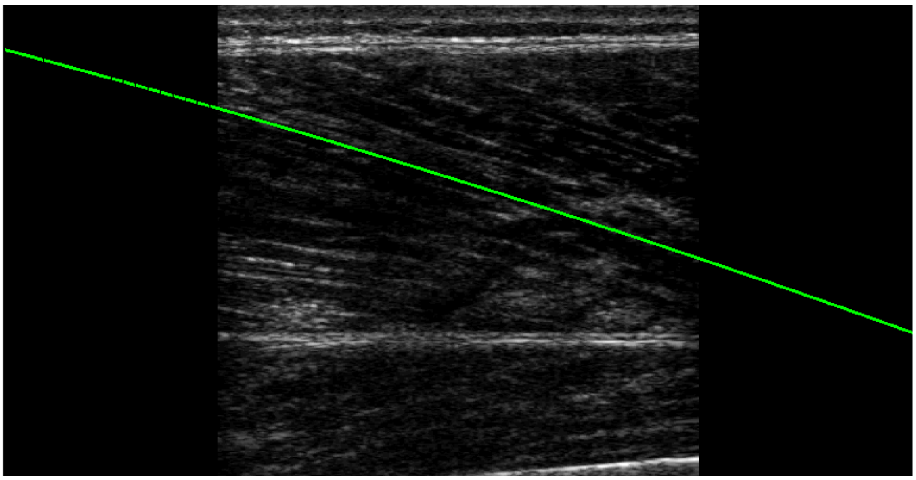


(b) Image nr. 5

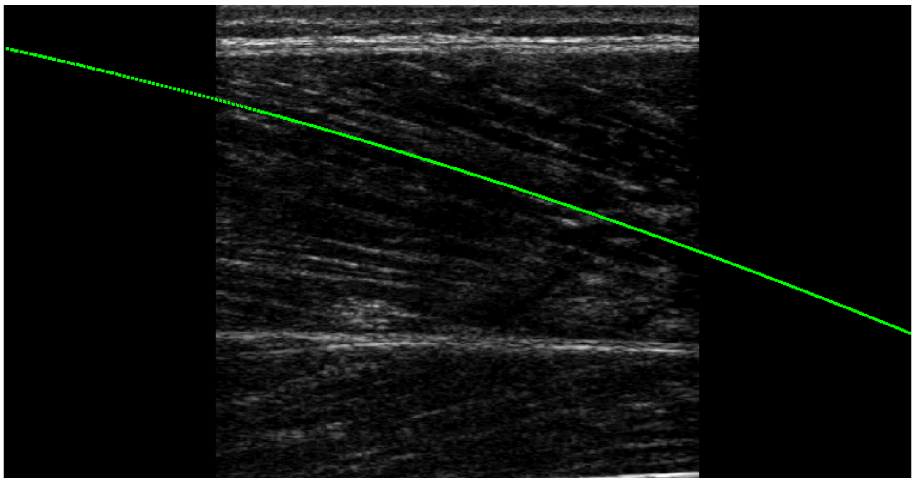


(c) Image nr. 6

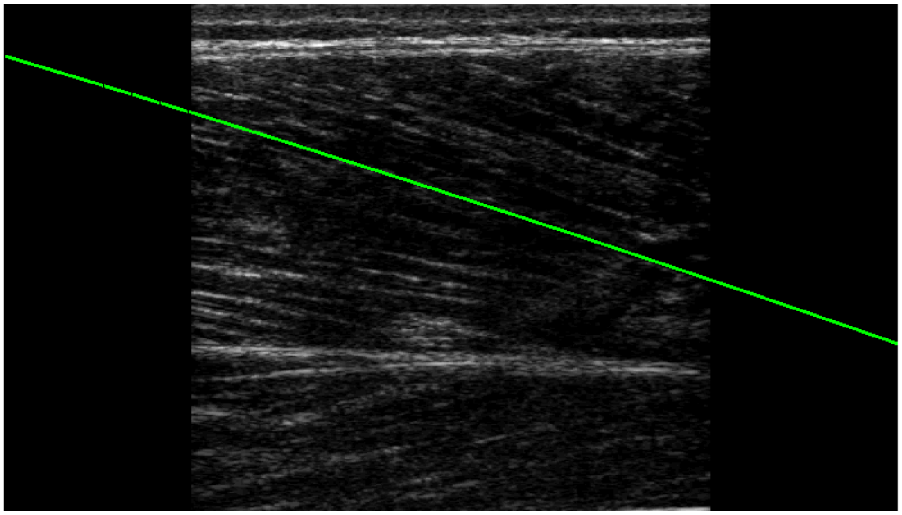
Figure 9.5: Constructed reference fascicles in images nr. 4-6.



(a) Image nr. 7

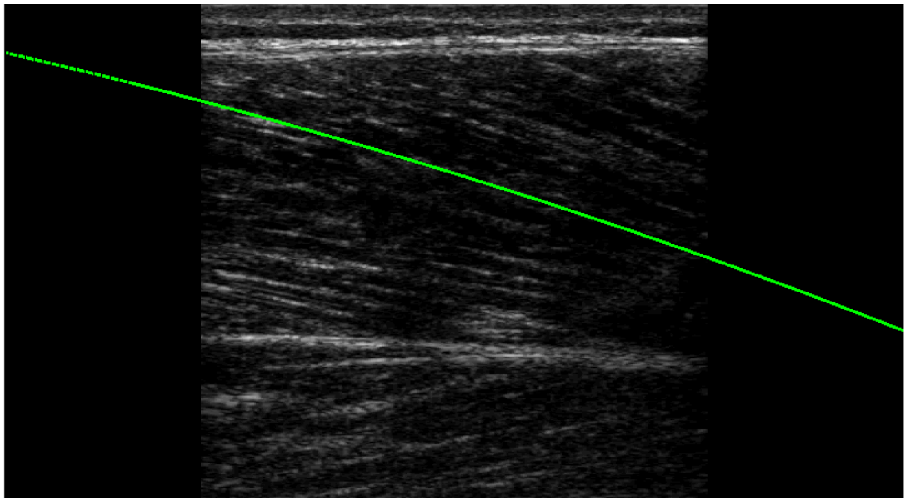


(b) Image nr. 8

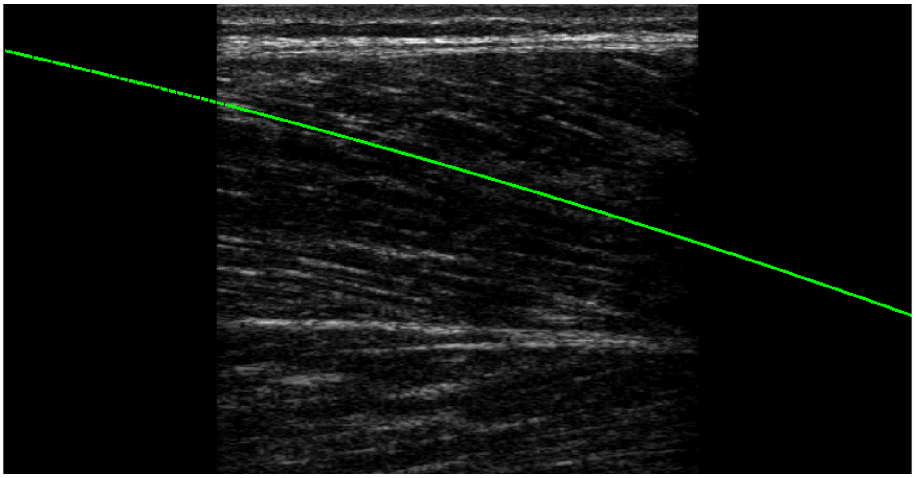


(c) Image nr. 9

Figure 9.6: Constructed reference fascicles in images nr. 7-9.



(a) Image nr. 10



(b) Image nr. 11

Figure 9.7: Constructed reference fascicles in images nr. 10 and 11.

Discussion of the visual analysis

Test set 1 worked particularily well with the algorithm, considering it located all the aponeuroses very well. That can be a bottleneck in the algorithm, because if that part of the algorithm fails, we will not receive accurate measurements for the pennation angle and the fascicle length. This gives us a good opportunity to evaluate how the part of the algorithm measuring the pennation angle and constructing the reference fascicle works.

However, there are some difficulties with a visual inspection as well. In most cases, we can clearly see if the aponeuroses match as they should, and that was fairly straight forward in this set of images. The pennation angle and the reference fascicles are not necessarily as easy to evaluate.

Difficulties with the analysis

The pennation angle is determined by the angle of the deep aponeuroses, and the detected angles in the window we have created. In some cases, there may be a lot of noise, or several different orientations in the structures in the window. In those cases, it is very difficult to evaluate if the algorithm has matched the underlying structures when inspecting the image. Some examples of large amounts of noise in the window can be seen in image 6 and image 7 (Figure 9.2b, Figure 9.2c). In these cases, the visual inspection has consisted more of evaluating how the detected angle fits with the clearer structures close to the window, not necessarily in the window itself.

In image 7 (Figure 9.2c), this gives rise to a conflict. In the left-most part of the window, there is some clear structure, with a very steep angle, and in most of the rest of the image there is much noise. However, close above the window from the middle and to the right there is more structure with a less steep orientation. In these cases, the evaluation is more determined by the overall understanding of the angle in the area, and it is not as certain as in the less noisy cases. It is worth mentioning that the same will occur when doing a manual measurement in images with that type of problems.

Evaluating the constructed reference fascicles by way of visual analysis is even more unreliable, there are few clear cut cases. One has to follow the curvature of the other fascicles in the image, and in most cases even 'extrapolate' the curves outside the field of view in the image intuitively. If the fascicles are mostly linear, the evaluation is simpler, but this does not hold for any of the images in this test set.

Because of these difficulties, it is with some uncertainty that the results for the pennation angle and the reference fascicle are classified in the three categories, and this should be taken into account when evaluating the results.

However, since humans are very good at pattern recognision, this is one of the most efficient tests we have to see if the algorithm is well functioning.

Evaluation of results

When locating the aponeuroses in this data set, the algorithm worked very well. The reasons for this is probably that there are not that many other bright structures in the same area as the aponeuroses, and that there is a very clear change of dominant direction in the image around the aponeuroses. Only one image did not receive a "GOOD", and that is the superficial aponeurosis in image 3 (Figure 9.1c). In this case, we can see that the aponeurosis goes up towards the skin layer towards the right edge, and appears to move slightly away from the aponeurosis itself. This could in theory affect the calculation of the fascicle length, because it could skew the extrapolation of the aponeurosis on the left side, but in these images, the fascicles are almost completely in view, and the curving of the aponeurosis is so slight, that the calculated length will probably not be far off. We can conclude that the aponeuroses detection worked very well for this data set.

The pennation angles detected are also mostly good detections, but there are two detections that clearly does not represent the angle in the area well. Particularily image nr. 9 (Figure 9.3a) presents an angle that differs a lot from the perceived angle in the area. A theory of why it does not detect the angle in the window is that the actual angle in the area relative to the horizontal line is around 5° (or smaller). As a way of ignoring the aponeuroses if they happened to come slightly into view in the fascicle plane (in the case of misdetection of the aponeuroses), the algorithm only searches for angles larger than 5°, and in this case that might be the reason why the pennation angle is wrongfully detected.

As an improvement to the algorithm, one might consider expanding the search the Radon transform performs, because the length estimation of the reference fascicle will not be valid when the aponeurosis is misdetected anyway.

There is also one detection that has been evaluated to "OK", image nr. 8 (Figure 9.2d). This is because there were a lot of noise, and more than one orientation in the window. The detected angle seemed to fit decently with the perceived structure, but it is not completely convincing.

The other eight images have very convincing and good results for the pennation angle in this data set.

The reference fascicles appear to fit very well with curvature of the muscle fascicles. In some images, in particular image nr. 8 and 10 (Figure 9.6b, Figure 9.7a), the curve on the top towards the superficial aponeurosis seems

a bit too steep. Other than that, these are very good results. All in all, the reference fascicles in these images fits very well with the perceived fascicle curvature.

That is an interesting fact in itself, because there are some cases we find the pennation angle not fitting the structure, while the reference fascicle still fits the image. This happens in image nr. 6 and 9. The reason the reference fascicle receives a nicer result for the angle in these two images is probably due to the fact that it is constructed by smoothing out the angeled line pieces. The question is whether or not it would be more robust to use the angle of the reference fascicle in the same height of the image as the pennation angle. This is something that might be worth looking into.

9.3 Reliability analysis

In this section, we would like to test how reliable the algorithm is. To do this, we have tested the algorithm on test set 2, and compared with the manual analysis done on the same set.

As mentioned in chapter 3, test set 2 was received a week before the deadline for this thesis. Upon creating this data set, the operator has tried to get as good view of the fascicles as possible, with the drawback that the aponeuroses become less prominent.

The set contained ultrasound images of the gastrocnemius muscle (GM) and the vastus lateralis muscle (VL) from 10 different subjects. Each of the subjects had three ultrasound scans on the same region of each muscle done, but with the probe shifted in a slightly different position in each of the three images. Each image comes with three manual measurements for the pennation angle, and three manual measurements for the fascicle length. The intention was that this could be used to do a reliability analysis, by comparing the typical error on the manual measurements with the typical error on the automatic measurements.

However, the algorithm failed to locate the aponeuroses in many of the images of test set 2, so a discussion on that matter is required before we can compare with the manual measurements.

Aponeuroses detection

In test set 2 there were some major issues with determining the approximate location of the aponeuroses. In Table 9.3 and Table 9.4 we present the results of the aponeurosis detection.

Subject	Muscle	Image nr.	Deep Aponeurosis	Superficial Aponeurosis
1	VL	1	-	GOOD
		2	GOOD	GOOD
		3	GOOD	GOOD
	GM	1	GOOD	APPROX
		2	GOOD	APPROX
		3	-	APPROX
2	VL	1	-	GOOD
		2	GOOD	GOOD
		3	-	GOOD
	GM	1	GOOD	APPROX
		2	GOOD	APPROX
		3	-	PARTIAL
3	VL	1	GOOD	GOOD
		2	GOOD	GOOD
		3	GOOD	GOOD
	GM	1	-	GOOD
		2	APPROX	GOOD
		3	APPROX	GOOD
4	VL	1	GOOD	GOOD
		2	-	GOOD
		3	-	GOOD
	GM	1	-	GOOD
		2	GOOD	GOOD
		3	GOOD	PARTIAL
5	VL	1	GOOD	GOOD
		2	-	GOOD
		3	-	GOOD
	GM	1	GOOD	PARTIAL
		2	GOOD	GOOD
		3	GOOD	PARTIAL

Table 9.3: Tabel with evaluation of the aponeurosis detection for 5 subjects on two different muscles with three images on each muscle. "GOOD" means the model fits well with the aponeurosis, "PARTIAL" means it almost fits, "APPROX" means it has located the area but not much else, and "-" means it did not find the approximate location.

Subject	Muscle	Image nr.	Deep Aponeurosis	Superficial Aponeurosis
6	VL	1	-	GOOD
		2	-	GOOD
		3	GOOD	GOOD
	GM	1	GOOD	GOOD
		2	GOOD	APPROX
		3	-	APPROX
7	VL	1	-	GOOD
		2	PARTIAL	GOOD
		3	-	GOOD
	GM	1	GOOD	GOOD
		2	APPROX	PARTIAL
		3	APPROX	GOOD
8	VL	1	GOOD	PARTIAL
		2	GOOD	PARTIAL
		3	APPROX	-
	GM	1	GOOD	GOOD
		2	GOOD	PARTIAL
		3	GOOD	-
9	VL	1	PARTIAL	GOOD
		2	PARTIAL	GOOD
		3	PARTIAL	GOOD
	GM	1	GOOD	GOOD
		2	GOOD	GOOD
		3	GOOD	GOOD
10	VL	1	-	APPROX
		2	GOOD	APPROX
		3	-	GOOD
	GM	1	GOOD	APPROX
		2	GOOD	APPROX
		3	GOOD	PARTIAL

Table 9.4: Tabel with evaluation of the aponeurosis detection for 5 subjects on two different muscles with three images on each muscle. "GOOD" means the model fits well with the aponeurosis, "PARTIAL" means it almost fits, "APPROX" means it has located the area but not much else, and "-" means it did not find the approximate location.

Approximate location

When inspecting where the approximate aponeuroses detection works and where it does not work, there are very subtle differences. An example of this is in images nr. 1 and 2 for the VL for subject nr. 1, shown in Figure 9.8. The largest difference here is that there is slightly more noise below the deep aponeurosis in image nr. 1. If we have a look at the differentiated arrays for the bottom half of these two images in Figure 9.9, we can see why the deep aponeurosis has been misdetected for the first image. There are peaks reaching as high about 1 for both images in the (presumed) correct position, however, there is a very large peak at the end of image 1 that overshadow the correct peak completely. Looking at the generated image showing the angles detected in different areas of the two images in Figure 9.10 can give us an answer on what is happening in this area. Even though there are no clear peak in each row in neither of the images beneath the position of the actual deep aponeurosis, b) is less noisy, and we believe that the reason the aponeurosis is misdetected in a) is that the image locates the 'wrong' median in several rows, which means the smoothing we have performed will not have any effect in this scenario.

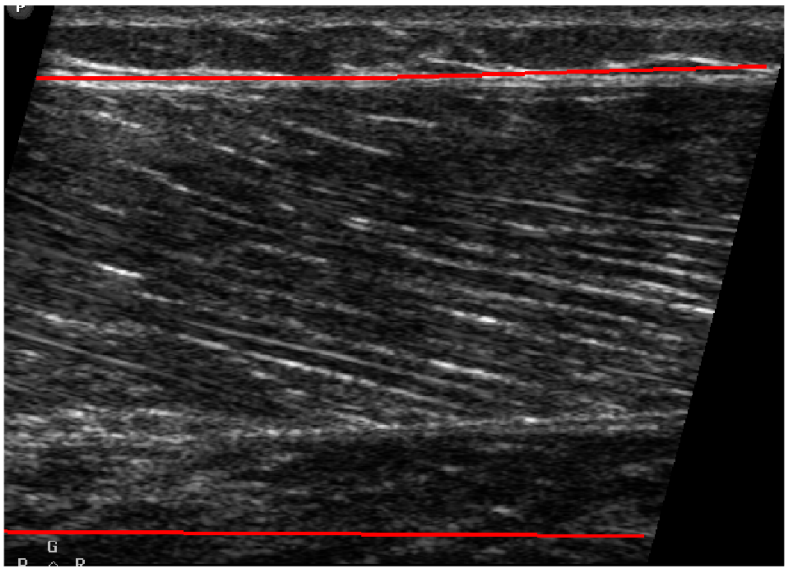
Searching through the data set, the differentiated arrays and the angles detected in each height in the image, this appears to be the cause of all(!) the cases where the approximate locating of the deep aponeurosis has failed, which is 18 images in this data set. That means that if we can preprocess the generated images in Figure 9.10 in some way to remove noise or create clearer peaks, the problem would be fixed. Other examples of these images where the deep aponeuroses is misdetected because of this problem is shown in Figure 9.11.

In the training data sets used to create the algorithm, we could also see a tendency to the detected angles in the images being more spread out beneath the deep aponeurosis, but it never became an issue because the smoothing of the differentiated array was enough to take care of the problem. Because of that, not a lot of work was done to robustify that part of the algorithm further, but clearly this needs to be handled.

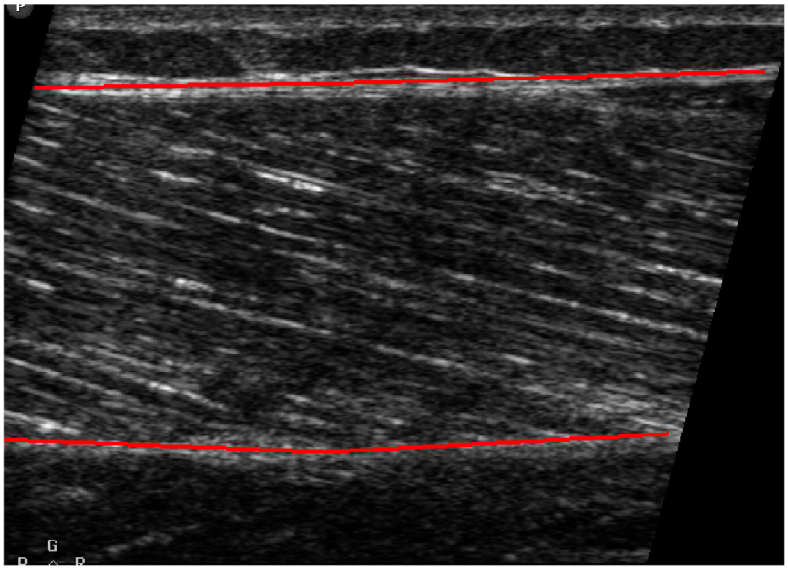
Exact detection

In the rest of the cases where the aponeuroses detection fail, the accurate location failed. This means that the algorithm has located a window around the aponeurosis, but it has not located the aponeurosis in that window.

In most of these cases, there is just a part of the aponeurosis that is misdetected, and the rest is correct. This happened to a certain degree

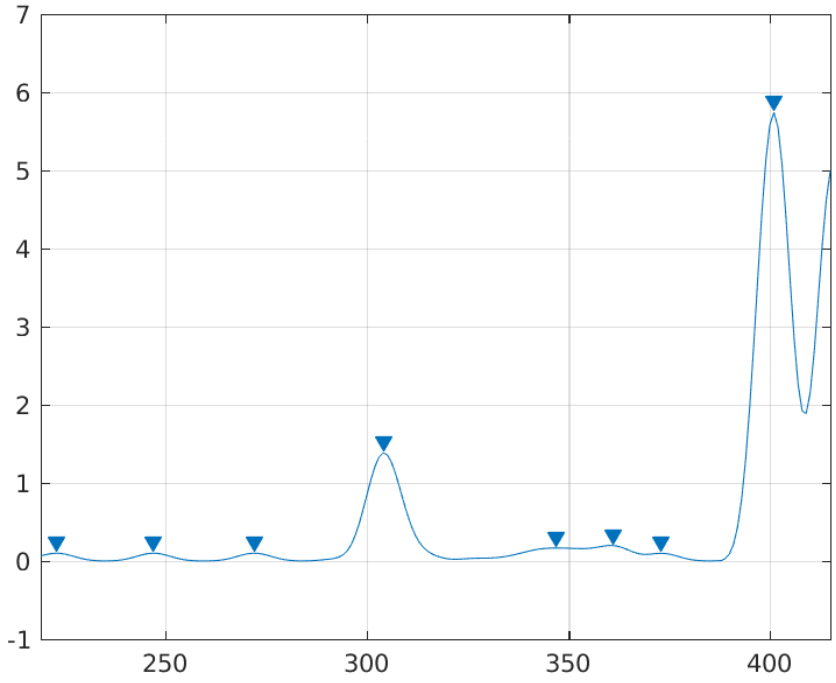


(a)

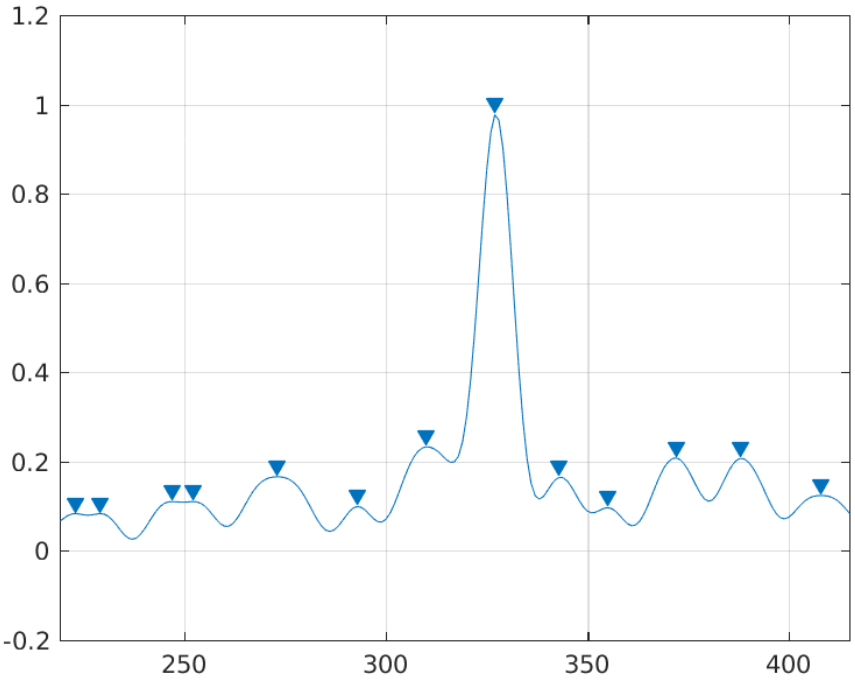


(b)

Figure 9.8: Detected aponeuroses and for the Vastus Lateralis on subject nr. 1 for a) Image nr. 1 and b) Image nr. 2.

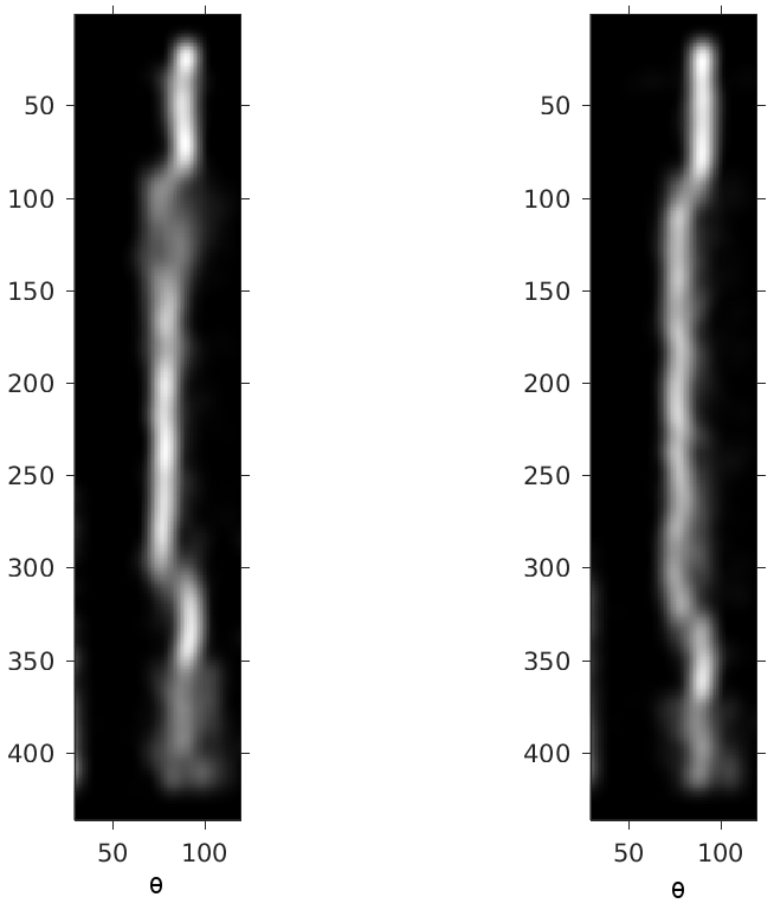


(a)



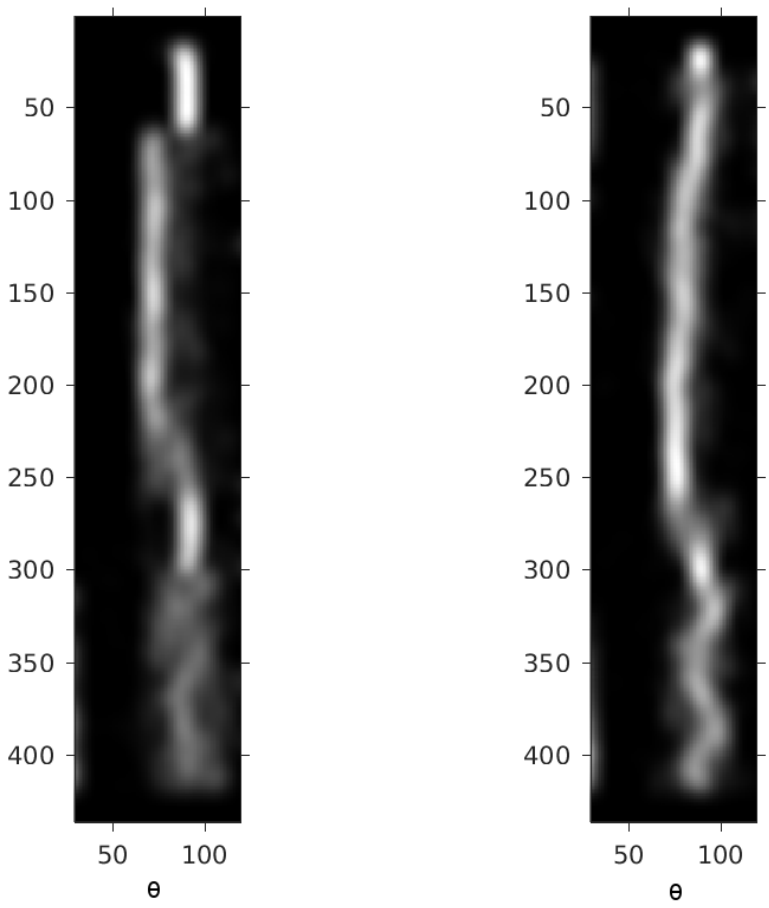
(b)

Figure 9.9: The arrays depicting change in dominant angle on Vastus Lateralis on subject nr. 1 for a) Image nr. 1 and b) Image nr. 2.



(a) (b)

Figure 9.10: The images generated depicting the angles detected in the different heights in the image on Vastus Lateralis on subject nr. 1 for a) Image nr. 1 and b) Image nr. 2. The rows are the pixel rows in the image, and the columns are the angle θ.



(a) (b)

Figure 9.11: Examples where the images generated depicting the angles detected in the different heights in the image have been too noisy beneath the deep aponeurosis. a) GM image nr. 1 on subject nr. 4 b) VL image nr. 3 on subject nr. 10. The rows are the pixel rows in the image, and the columns are the angle θ.

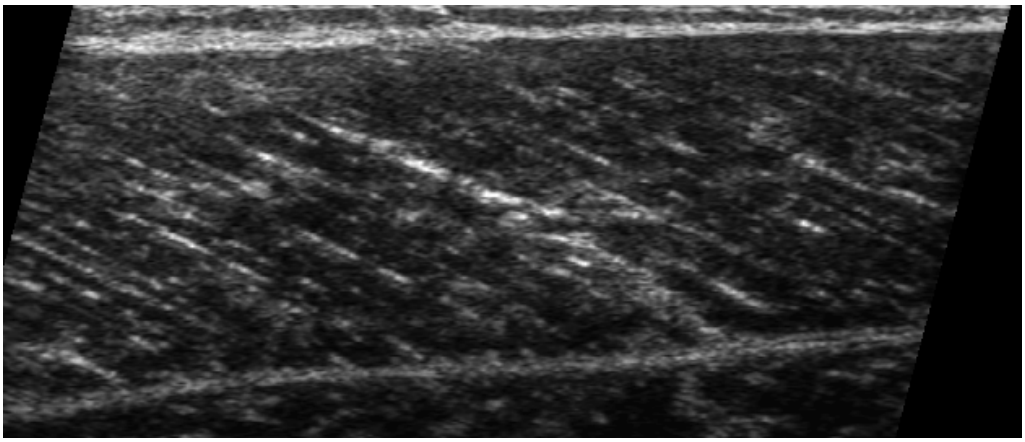


Figure 9.12: Example of how the data was cropped before repeating the algorithm.

in the data sets when we developed the algorithm as well, and possible solutions to this has been discussed in Section 6.4.

Further analysis of results from testing with test set 2

In order to get comparable data with the manual measurements, we decided to crop the data manually in the images where the aponeurosis detection had failed/was poor. The images were cropped so that only the aponeuroses and the fascicle plane was in view. An example of this is shown in Figure 9.12. Then the algorithm was performed again on these images. The results were mostly good, and we got acceptable results for the aponeurosis detection on all images except two images where the accurate detection was deemed to poor to use. The cropping was not only efficient on the images where the approximate detection had failed, but also on many of the images where the accurate detection failed. In Figure 9.13 we show examples of before and after for the gastrocnemius muscle on subject 3.

We present the results including the new data when necessary in Table 9.5 and Table 9.6 for the pennation angle, and Table 9.7 and Table 9.8 for the fascicle length. The results for the images that was cropped to get usable results are marked with a '*'.

In order to get a better overview of the results, we will first discuss and present some tabels for the results for the pennation angle in the vastus lateralis and the gastrocnemius muscle, and then do the same for the fascicle length. For each muscle, we present two tabels. One where we look at the mean over each subject, and one where we look at the standard deviation

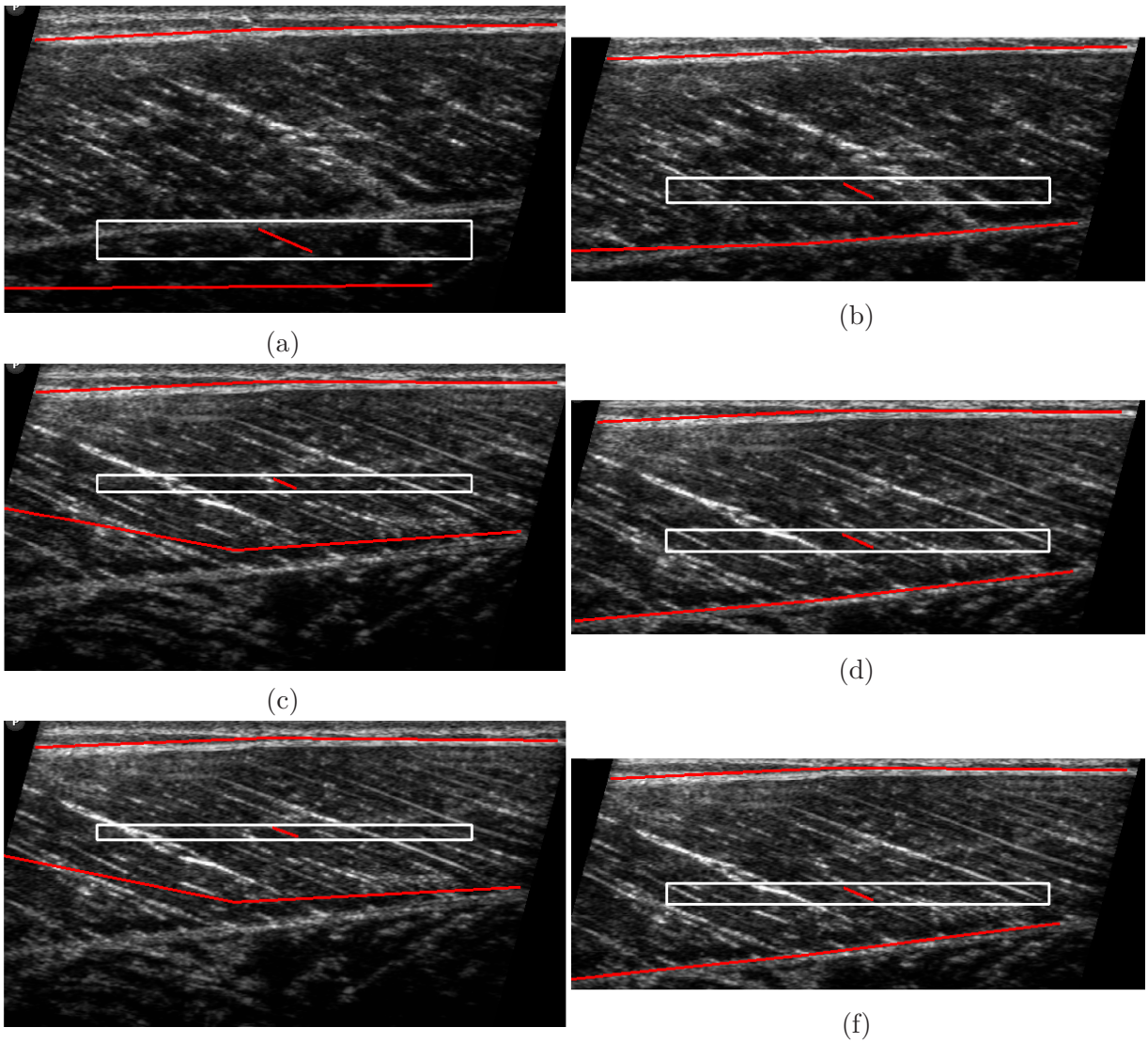


Figure 9.13: The detected aponeuroses before and after cropping of the ultrasound images of the gastrocnemius muscle for subject 3. a) and b) are of image nr. 1, c) and d) image nr. 2 and e) and f) are image nr. 3.

Subject	Muscle	Image nr.	Aut. Pennation Angle	Man. Pennation Angle
1	VL	1	14.5*	16
		2	15.5	14.5
		3	16	16.5
	GM	1	18*	17.5
		2	18*	18
		3	20.50*	17
2	VL	1	23*	18.5
		2	21	19.5
		3	22*	21
	GM	1	25.5*	25.5
		2	25.5*	22
		3	23*	24
3	VL	1	21.5	22
		2	21	21.5
		3	20.5	20.5
	GM	1	29*	31
		2	27.5*	27
		3	28.5*	27.5
4	VL	1	16.5	18
		2	15.5*	17
		3	15*	17
	GM	1	22.5*	24.5
		2	24.5	24
		3	22.5	21.5
5	VL	1	20	17.5
		2	18*	19
		3	19.5*	18
	GM	1	21.5	22
		2	23	21
		3	25	23.5

Table 9.5: Tabel with automatic and manual measurements of pennation angle for 5 subjects on two different muscles with three images on each muscle. The manual measurement is the mean of three measurements on each image. '*' means the image was cropped before detection.

Subject	Muscle	Image nr.	Aut. Pennation Angle	Man. Pennation Angle
6	VL	1	20*	20
		2	20*	20
		3	17.5	18
	GM	1	28.5	25.5
		2	27.5	28
		3	26.5*	25.5
7	VL	1	19*	17.5
		2	17*	18.5
		3	17.5*	19.5
	GM	1	21.5	18.5
		2	-	20
		3	21*	20
8	VL	1	15.5	16.5
		2	17	16.5
		3	16.5*	16.5
	GM	1	24.5	24
		2	24.5	26.5
		3	27*	28
9	VL	1	14*	15.5
		2	16*	16.5
		3	16*	14.5
	GM	1	25.5	24
		2	25.5	24
		3	23	26
10	VL	1	14.5*	14.5
		2	18*	18
		3	-	18
	GM	1	23	24
		2	24	22.5
		3	23	24

Table 9.6: Tabel with automatic and manual measurements of pennation angle for 5 subjects on two different muscles with three images on each muscle. The manual measurement is the mean of three measurements on each image. '*' means the image was cropped before detection.

Subject	Muscle	Image nr.	Aut. Fasicle Length	Man. Fascicle Length
1	VL	1	845*	1040
		2	800	1095
		3	845	1090
	GM	1	540*	645
		2	530*	640
		3	560*	625
2	VL	1	735*	895
		2	745	860
		3	695	785
	GM	1	390*	455
		2	380*	500
		3	395*	480
3	VL	1	645	690
		2	615	695
		3	645	770
	GM	1	360*	405
		2	395*	445
		3	405*	440
4	VL	1	765	875
		2	800*	905
		3	695*	920
	GM	1	540*	570
		2	540	600
		3	545	650
5	VL	1	625	895
		2	1020*	860
		3	925*	990
	GM	1	685	690
		2	620	700
		3	730	700

Table 9.7: Tabel with automatic and manual measurements of fascicle length for 5 subjects on two different muscles with three images on each muscle. The manual measurement is the mean of three measurements on each image. '*' means the image was cropped before detection.

Subject	Muscle	Image nr.	Aut. Fascicle Length	Man. Fascicle Length
6	VL	1	1125*	875
		2	1045*	870
		3	935	880
	GM	1	520	655
		2	550	595
		3	485*	620
7	VL	1	830*	1015
		2	830*	995
		3	725*	975
	GM	1	515	630
		2	-	665
		3	540*	635
8	VL	1	750*	735
		2	740*	785
		3	815*	805
	GM	1	475	495
		2	480	455
		3	440*	445
9	VL	1	1015*	975
		2	16*	950
		3	16*	1025
	GM	1	540	615
		2	540	595
		3	580	600
10	VL	1	795*	945
		2	630*	750
		3	-	745
	GM	1	440	535
		2	430	560
		3	435	560

Table 9.8: Tabel with automatic and manual measurements of fascicle length for 5 subjects on two different muscles with three images on each muscle. The manual measurement is the mean of three measurements on each image. '*' means the image was cropped before detection.

over each subject, for both the automatic measurements and the manual measurements.

In the last row in the tabels over the mean, we have found the mean over all automatic measurements, the mean over all manual measurements, and the mean over all the differences. In the last row in the tabels over the standard deviations, we have found the standard deviation over all the automatic measurements and the standard deviation over all the manual measurements (3 measurements per image). We have not found the standard deviation for the difference in every measurement, since this was not found necessary for the discussion.

Pennation angle

The results for the pennation angle in the vastus lateralis can be seen in Table 9.9 and Table 9.10.

There seems to be a good correspondance with the measured mean angle in most cases, with two very notable exceptions. Especially in subject 2, where there is a difference of more than 2 degrees, these images with detected angle and aponeuroses are shown in Figure 9.14. Upon visual inspection, we can see that both the aponeuroses and the line indicating the detected angle looks good, and we believe that the automatic algorithm is accurate in this case. This was included as an example because it shows that we cannot take the manual measurements as the correct estimates as discussed previously, because they are also subject to error.

In general, the standard deviation is slightly higher in the automatic measurements than the manual measurements, which is a bit disappointing since it indicates that the algorithm in general gives a higher variability in the results than the manual measurements. To examine this we look at the subject with the highest standard deviation, subject 10. This consists only of two images, because the algorithm failed to detect the deep aponeurosis of the third image correctly, and they are shown in Figure 9.15. It is rather clear that the problem lies in the second image here. Probably due to noise in the area where the pennation angle is detected, the resulting angle clearly does not match the underlying structure. It is not far off, but the margins are small. If there are more of these types of errors, then it quickly explains why we get a higher standard deviation for the algorithm than the manual measurements.

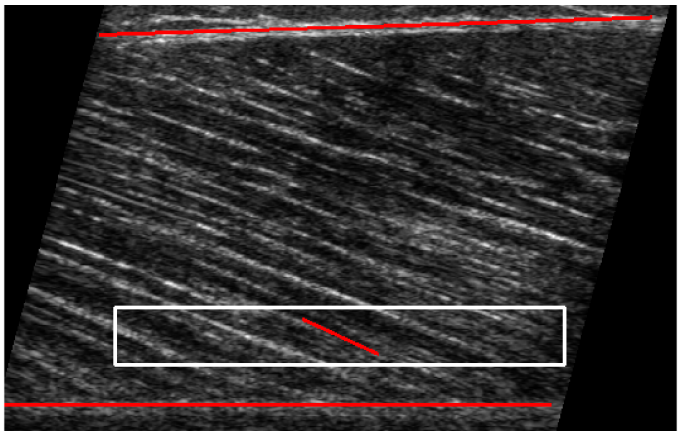
We also have a look at the results for the gastrocnemius muscle. The mean and the standard deviation is shown in Table 9.11 and Table 9.12, respectively. The largest mean difference we find in subject nr. 7. This set also has one image with a misdetected deep aponeurosis which we have not

Vastus lateralis – pennation angle			
Subject	Aut. Mean	Man. Mean	Diff. Mean
1	15.24	15.60	0.36
2	21.97	19.76	2.21
3	21.04	21.30	0.26
4	15.55	17.43	1.88
5	19.30	18.12	1.17
6	19.29	19.34	0.04
7	17.95	18.46	0.51
8	16.18	16.39	0.22
9	15.44	15.38	0.06
10	16.34	16.76	0.43
Total	17.83	17.85	0.71

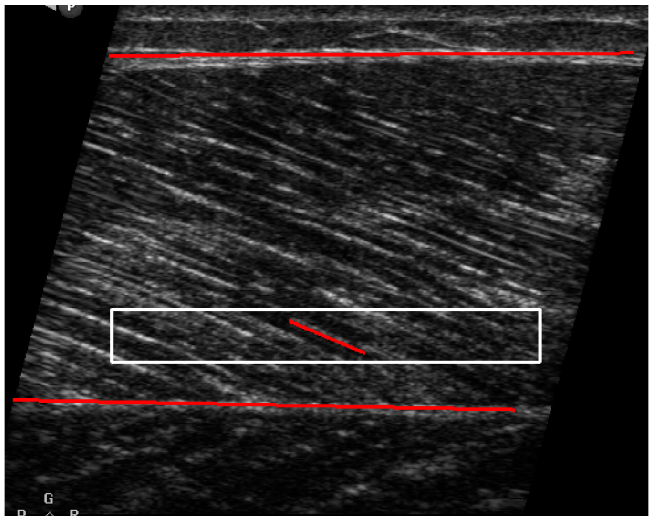
Table 9.9: Mean for the pennation angle in the vastus lateralis muscle over all three images for all ten subjects, and the total mean, shown for the automatic measurements, the manual measurements and the difference between the two.

Vastus lateralis – pennation angle			
Subject	Aut. SD	Man. SD	Diff. SD
1	0.86	1.08	0.22
2	1.07	1.47	0.40
3	0.59	1.09	0.49
4	0.80	0.70	0.11
5	1.07	0.61	0.46
6	1.36	1.13	0.22
7	1.08	1.11	0.04
8	0.75	0.60	0.15
9	1.05	0.92	0.13
10	2.48	1.74	0.74
Total	2.56	2.10	-

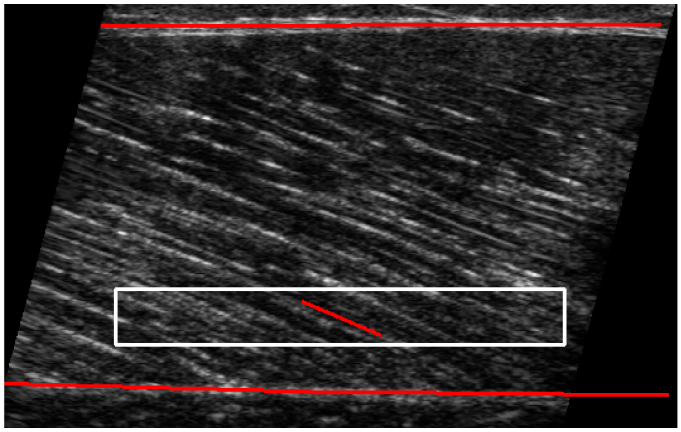
Table 9.10: Standard deviation for the pennation angle in the vastus lateralis muscle over all three images for all ten subjects, and the total standard deviation for all images, shown for the automatic measurements, the manual measurements and the difference between the two. The standard deviation for the manual measurements is calculated over all 9 measurements per subject (3 per image).



(a)

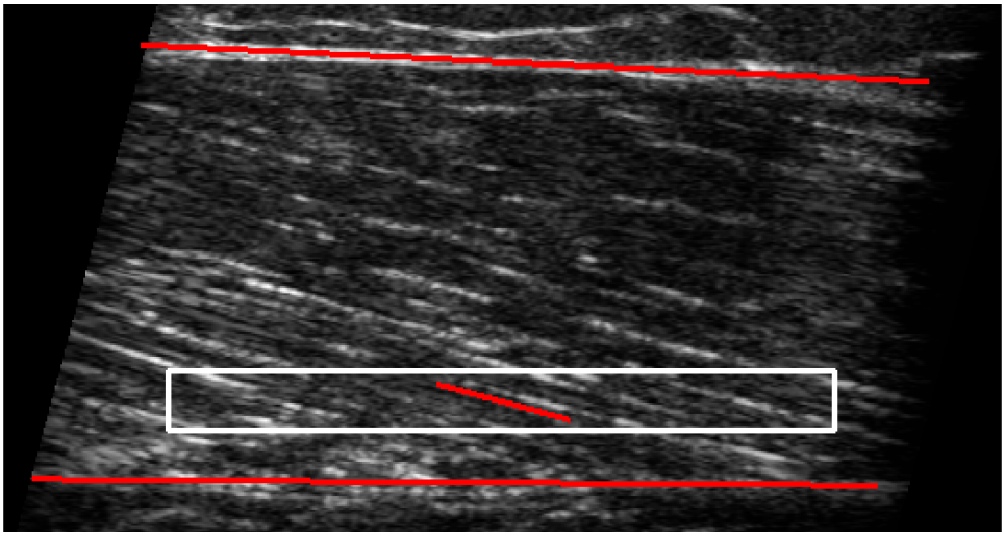


(b)

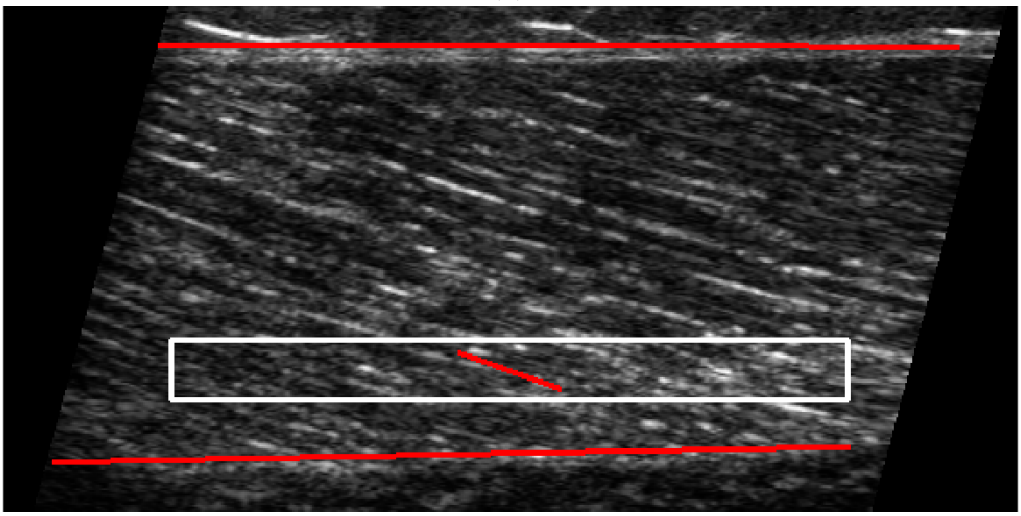


(c)

Figure 9.14: The detected aponeuroses and pennation angle on the vastus lateralis for subject nr. 2.



(a)



(b)

Figure 9.15: The detected aponeuroses and pennation angle on the vastus lateralis for subject nr. 10.

included in the data set. The images can be seen in Figure 9.16. Here we see a slight misdetection of the deep aponeurosis for b). Interestingly enough, it is the first image of these two that has the highest difference between the manually measured and the automatically measured angle (as can be seen in Table 9.6).

Having a look at the standard deviation, we see that while the standard deviation is in general higher for the gastrocnemius muscle than the vastus lateralis, it is actually lower for the automatic estimations than the manual estimations. In subject 3, we can see a relatively low standard deviation for the automatic estimations, while there is a higher standard deviation for the manual estimation. The images for subject 3 is shown in Figure 9.17. What we see is that in image a), the deep aponeurosis is slightly misdetected, which as a result gives a smaller angle than it should have been. A result of this is actually that the standard deviation is artificially small for the automatic estimations. Interestingly enough, the mean angle detected is almost the same.

We see that there are small errors on the automatic algorithm in some images, most of it is due to a slightly erronous deep aponeurosis detection, some of it is due to a noisy fascicle plane in the area we estimate the pennation angle from. When looking at the statistics and the detections in general however, we see that it indicates mainly good results.

Fascicle length

The results for the fascicle length in the vastus lateralis can be seen in Table 9.13 and Table 9.14. When inspecting the reference fascicle illustrations, keep in mind that the bottom height of the fascicle is the top height of the deep aponeurosis, and the top height is the bottom height of the superficial aponeurosis. When estimating the length, we extrapolate the polynomials and estimate the length from the point the extrapolated fascicle hits the extrapolated deep aponeurosis, to the point the extrapolated fascicle hits the extrapolated superficial aponeurosis. Thus, the illustrated polynomial does not have the same length as the polynomial we use to calculate the length, but it has the same shape.

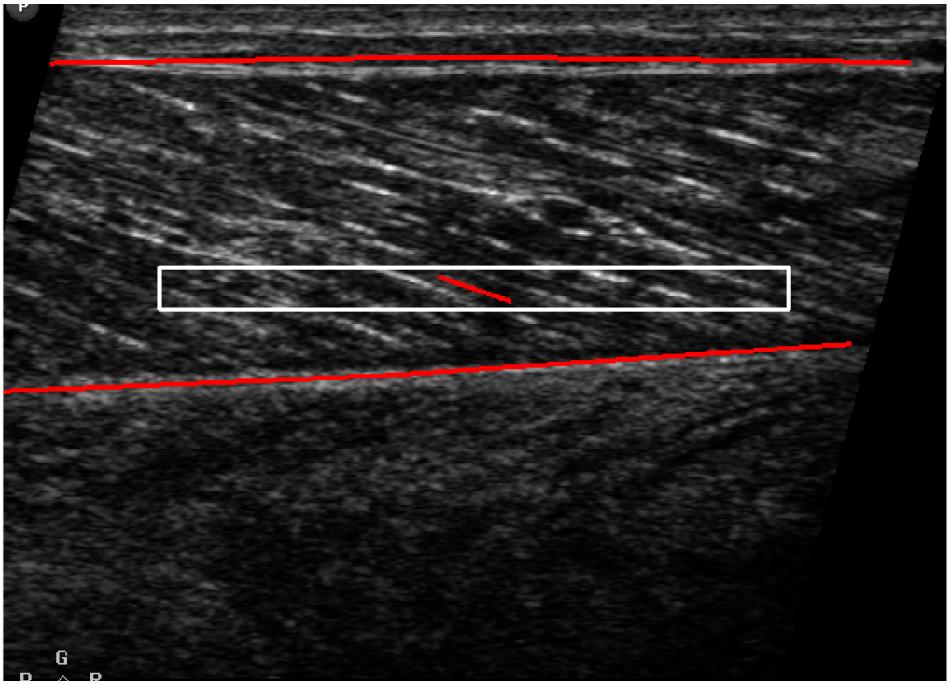
We see that the average difference in the fascicle length is quite large, and it seems that the algorithm in general reports a lower fascicle length than the manual estimation does. We have a look at subject nr. 1, who has the largest mean difference estimated, with a whopping 243 pixel difference. In Figure 9.18, we see the estimated aponeuroses (and pennation angle), and in Figure 9.19, we see the estimated reference fascicle.

Gastrocnemius muscle - pennation angle			
Subject	Aut. Mean	Man. Mean	Diff. Mean
1	18.86	17.50	1.36
2	24.57	23.69	0.87
3	28.49	28.39	0.10
4	23.20	23.24	0.04
5	23.25	22.09	1.15
6	27.65	26.18	1.47
7	21.13	19.65	1.48
8	26.07	26.15	0.09
9	24.58	24.77	0.19
10	23.29	23.54	0.25
Total	24.11	23.52	0.70

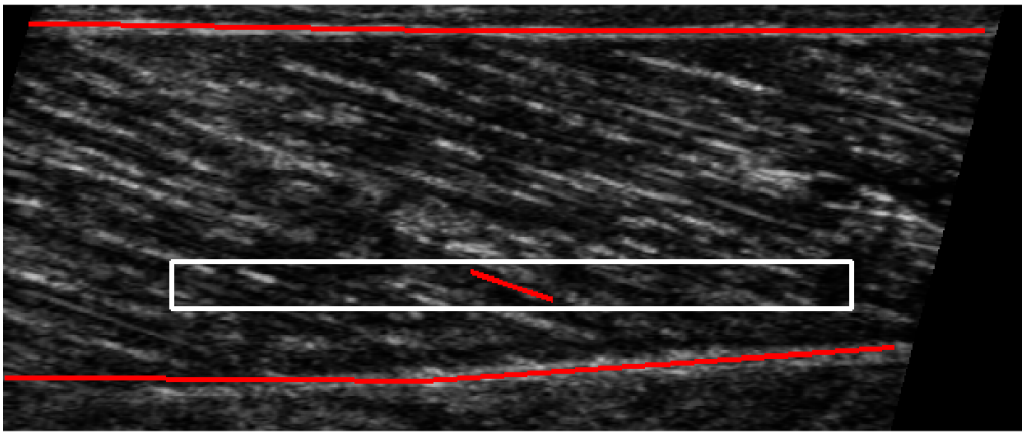
Table 9.11: Mean for the pennation angle in the gastrocnemius muscle over all three images for all ten subjects, and the total mean, shown for the automatic measurements, the manual measurements and the difference between the two.

Gastrocnemius muscle - pennation angle			
Subject	Aut. SD	Man. SD	Diff. SD
1	1.45	0.34	1.11
2	1.58	1.80	0.22
3	0.76	2.15	1.39
4	1.15	1.60	0.46
5	1.85	1.34	0.51
6	0.99	1.45	0.46
7	0.18	0.97	0.79
8	2.65	1.83	0.83
9	1.24	1.10	0.14
10	0.61	0.84	0.23
Total	3.01	3.35	-

Table 9.12: Standard deviation for the pennation angle in the gastrocnemius muscle over all three images for all ten subjects, and the total standard deviation for all images, shown for the automatic measurements, the manual measurements and the difference between the two. The standard deviation for the manual measurements is calculated over all 9 measurements per subject (3 per image).

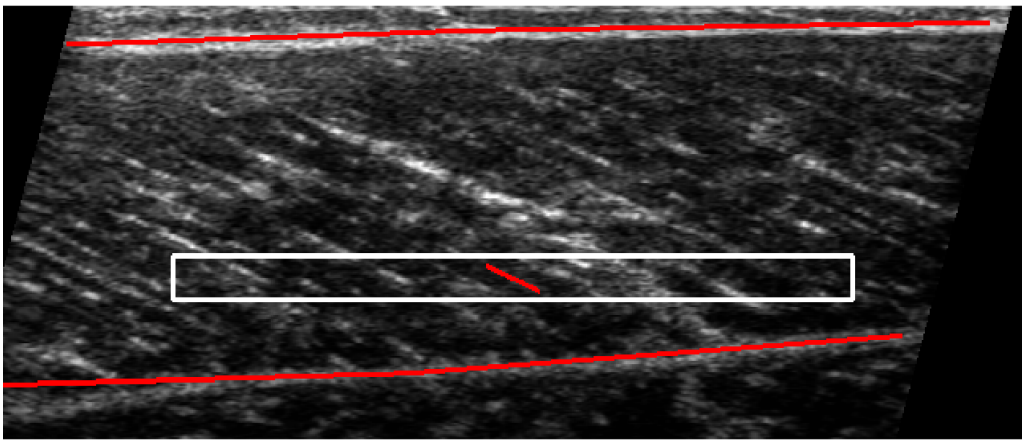


(a)

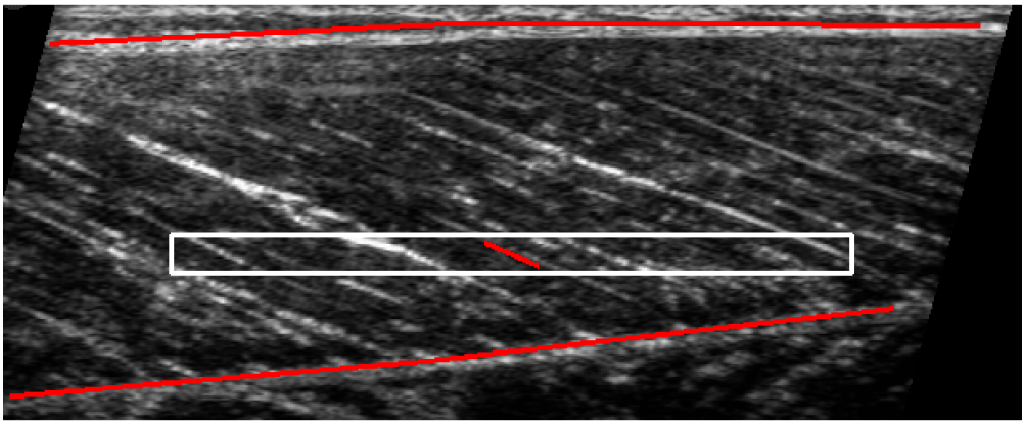


(b)

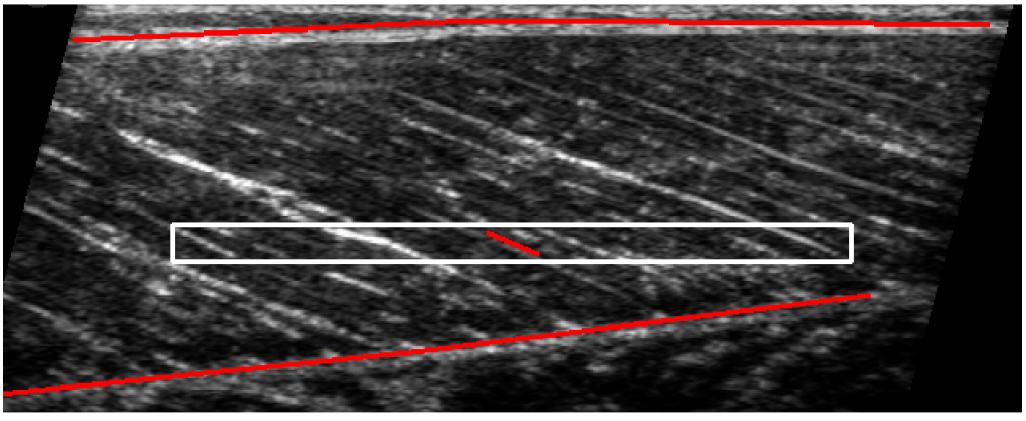
Figure 9.16: The detected aponeuroses and pennation angle on the gastrocnemius muscle for subject nr. 7.



(a)



(b)



(c)

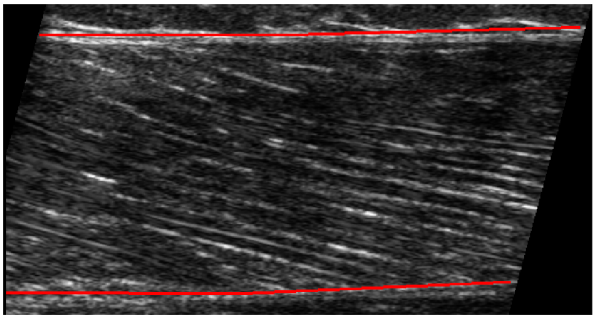
Figure 9.17: The detected aponeuroses and pennation angle on the gastrocnemius muscle for subject nr. 3.

Vastus lateralis – fascicle length			
Subject	Aut. Mean	Man. Mean	Diff. Mean
1	831.53	1074.59	243.05
2	724.90	846.10	121.20
3	634.13	718.20	84.06
4	752.70	900.59	147.89
5	857.87	915.90	58.03
6	1035.00	874.09	160.91
7	795.83	995.40	199.57
8	768.87	775.27	6.40
9	898.53	984.22	85.69
10	713.95	748.85	34.90
Total	801.33	883.32	114.17

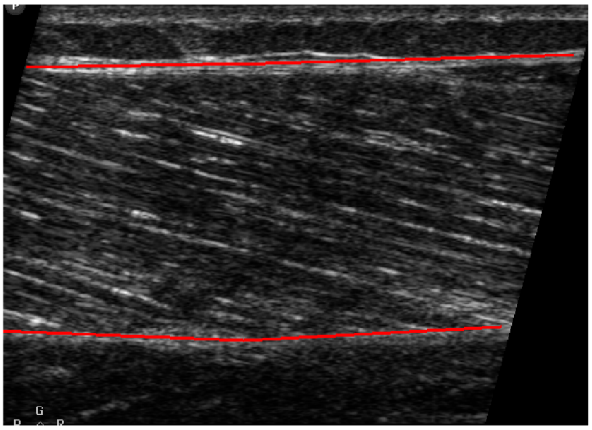
Table 9.13: Mean for the fascicle length in the vastus lateralis muscle over all three images for all ten subjects, and the total mean, shown for the automatic measurements, the manual measurements and the difference between the two.

Vastus lateralis - fascicle length			
Subject	Aut. SD	Man. SD	Diff. SD
1	25.66	36.27	10.61
2	26.17	65.85	39.68
3	16.15	49.86	33.71
4	53.32	25.70	27.62
5	205.22	61.90	143.32
6	95.18	39.12	56.06
7	60.57	31.29	29.29
8	41.77	39.19	2.58
9	114.30	40.93	73.37
10	116.32	100.78	15.54
Total	132.69	114.96	-

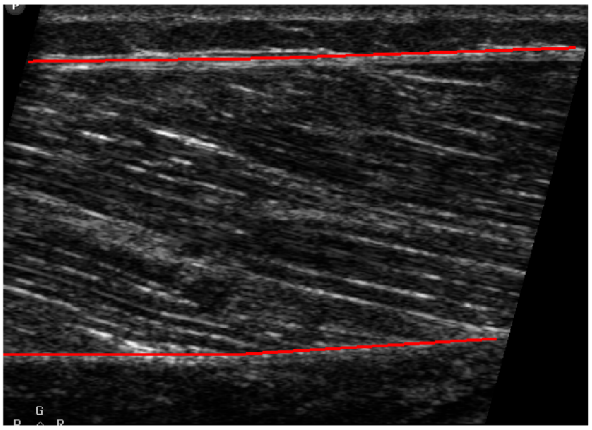
Table 9.14: Standard deviation for the fascicle length in the vastus lateralis muscle over all three images for all ten subjects, and the total standard deviation for all images, shown for the automatic measurements, the manual measurements and the difference between the two. The standard deviation for the manual measurements is calculated over all 9 measurements per subject (3 per image).



(a)

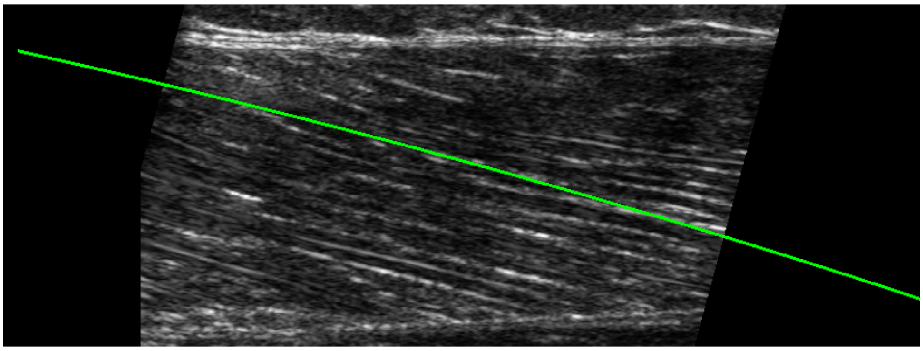


(b)

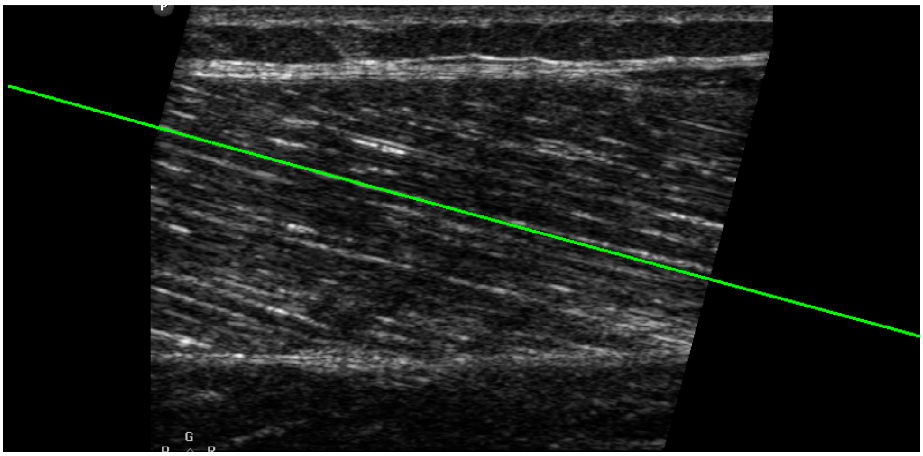


(c)

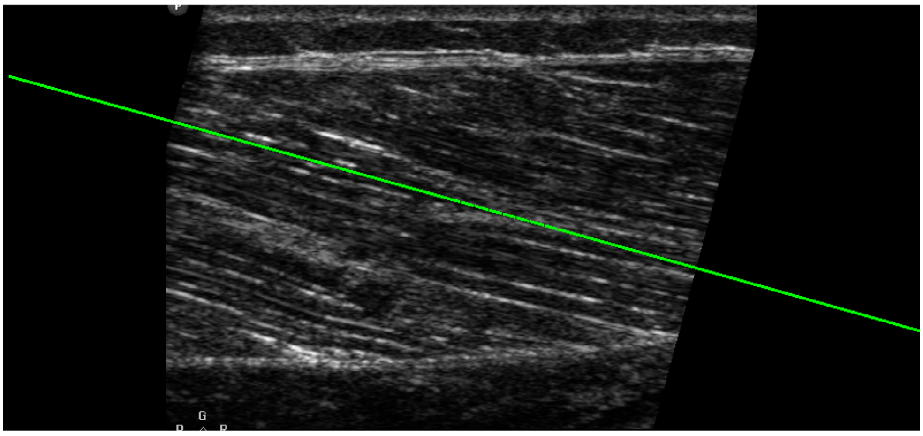
Figure 9.18: The detected aponeuroses on the vastus lateralis for subject nr. 1.



(a)



(b)



(c)

Figure 9.19: The constructed reference fascicle on the vastus lateralis for subject nr. 1.

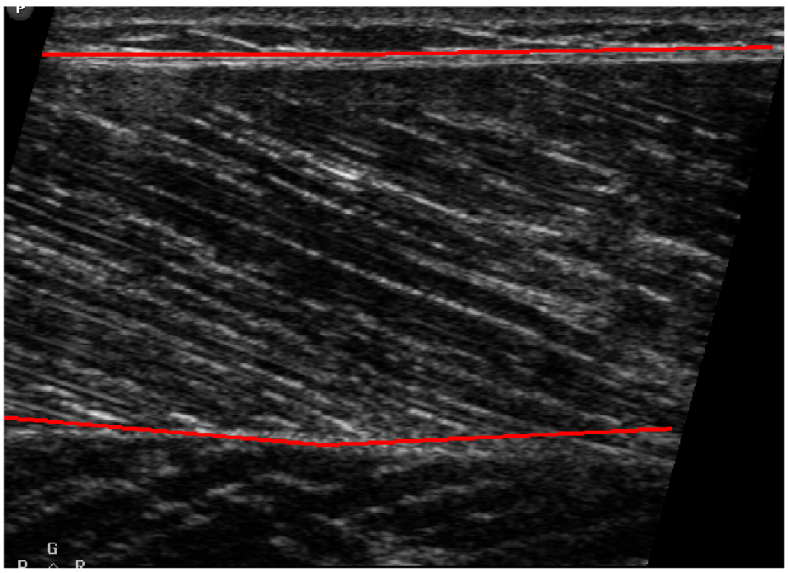


Figure 9.20: The estimated aponeuroses for the first image from subject 5.

The aponeuroses seems to be well detected in all three images. In the first image, the reference fascicle is a mismatched concave curve, but in the other two images the reference fascicle looks more correct. As we can see in Table 9.7, the fascicle length of the first image is the same as the fascicle length of the third image, so that does not explain the reason why the mean difference is so different from the manual measurements. Since the manual measurements of the fascicle length relies on linearly extrapolated aponeuroses, one theory is that the fact that we extrapolate the aponeuroses as quadratic polynomials gives us start and/or end points that are closer than if the aponeuroses are extrapolated linearly.

The total standard deviation is slightly larger for the automatic measurements than the manual ones, and especially in subject 5. A quick look in Table 9.7 reveals that the estimated fascicle length of image 1 is a lot smaller than the other two, and a look at the aponeurosis detection reveals why, in Figure 9.20. Because the deep aponeurosis is not estimated correctly, the start point will be a lot higher than it should be, especially because of the extrapolation. This shows us that what we perceive as small errors in the aponeurosis detection, can generate large errors in the results.

Finally, we have a look at the fascicle length for the gastrocnemius muscle. The data is shown in Table 9.15 and Table 9.16. Since more of each fascicle

is in view, it should in theory give smaller errors because there is less need for extrapolation. This seems to hold, because the standard deviation for the automatic measurements is at about the same rate as for the manual measurements.

In subject 5, we find the highest standard deviation for the automatic measurements. When we look at the curvation of the reference fascicle in Figure 9.21, we see that this seems to fit well with what we would expect it to. However, looking at the detected aponeuroses, we understand why there is such a high standard deviation. In Figure 9.22, we see that for two of the images, the superficial aponeurosis is erronously detected, and curved in such a way that it is clear that the extrapolation of these will give a much longer fascicle length than it is supposed to.

What we can see is that in most cases, the reference fascicle follows the curvature of the fascicles, but clearly the detection of the aponeuroses is a bottleneck in the algorithm.

As mentioned, since the manual measurements of the fascicle length relies on linearly extrapolated aponeuroses, the way we extrapolate the aponeuroses might give us a shorter fascicle length – but it might not necessarily be physically wrong, since we know the aponeuroses does curve outside the field of view in many cases. However, if the aponeurosis detection is slightly off, then this will produce possibly large errors when extrapolating them.

9.4 Test of validity

We also wanted to test the fascicle orientation part of the algorithm on images we knew the angle of, as a test of validity. We got access to two ultrasound images of hair. These images have structures that can be compared to fascicles. These two images can be seen in Figure 3.6, and when calculating the pennation angle for both of the images, the algorithm detected 16° and 22°, relative to the horizontal plane. The images have angles of 16.6° and 22.4°. Since MATLABs implementation of Radon transform only outputs whole degrees, this should be considered a good result, although technically, one could argue that it should have been 17° instead of 16°.

However, the pennation angle is calculated in the bottom third of the image, and it is a lot less noisy in those areas than otherwise, so we have a look at the reference fascicles in the middle of the images as well. These are shown in Figure 9.23.

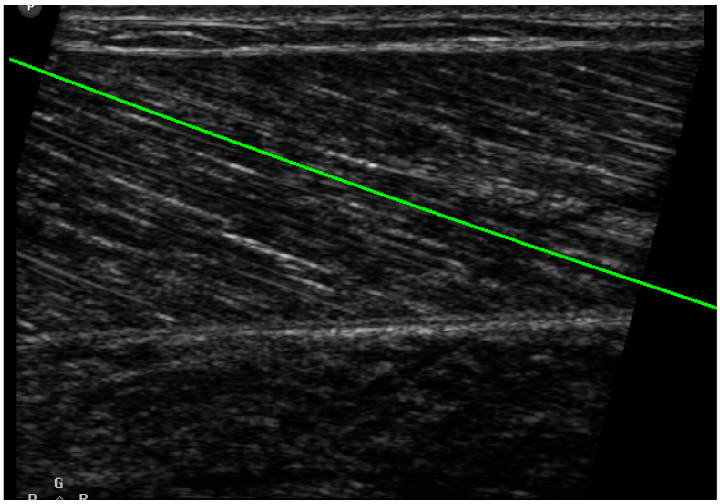
Here, we find the first reference fascicle to be 16°, but in the second image, we find the angle of the reference fascicle to be 21°, which means it

Gastrocnemius muscle - fascicle length			
Subject	Aut. Mean	Man. Mean	Diff. Mean
1	544.20	636.66	92.46
2	387.43	478.66	91.22
3	386.27	429.83	43.56
4	540.47	606.56	66.10
5	678.33	697.35	19.01
6	518.53	622.88	104.35
7	529.45	644.31	114.86
8	465.43	466.57	1.13
9	553.17	603.30	50.14
10	433.27	551.14	117.87
Total	503.66	573.73	70.07

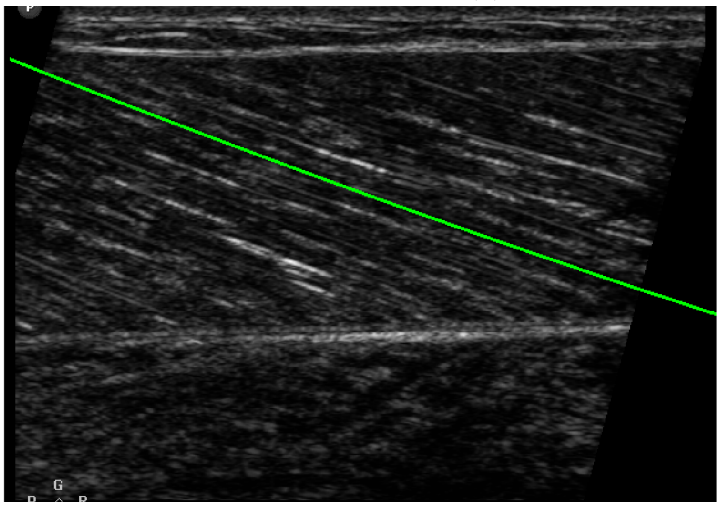
Table 9.15: Mean for the fascicle length in the gastrocnemius muscle over all three images for all ten subjects, and the total mean, shown for the automatic measurements, the manual measurements and the difference between the two.

Gastrocnemius muscle - fascicle length			
Subject	Aut. SD	Man. SD	Diff. SD
1	15.50	13.35	2.15
2	6.53	28.37	21.84
3	23.84	19.95	3.89
4	3.96	38.74	34.79
5	56.42	19.96	36.46
6	31.14	29.77	1.37
7	17.32	25.12	7.80
8	24.24	27.13	2.89
9	21.08	17.02	4.07
10	4.95	20.11	15.15
Total	89.33	87.59	-

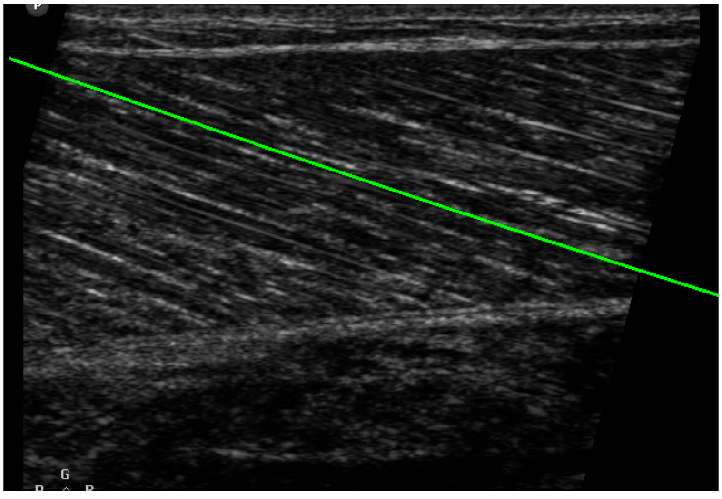
Table 9.16: Standard deviation for the fascicle length in the gastrocnemius muscle over all three images for all ten subjects, and the total standard deviation for all images, shown for the automatic measurements, the manual measurements and the difference between the two. The standard deviation for the manual measurements is calculated over all 9 measurements per subject (3 per image).



(a)

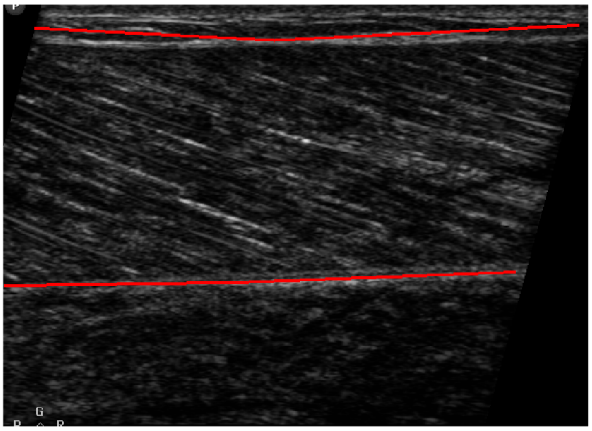


(b)

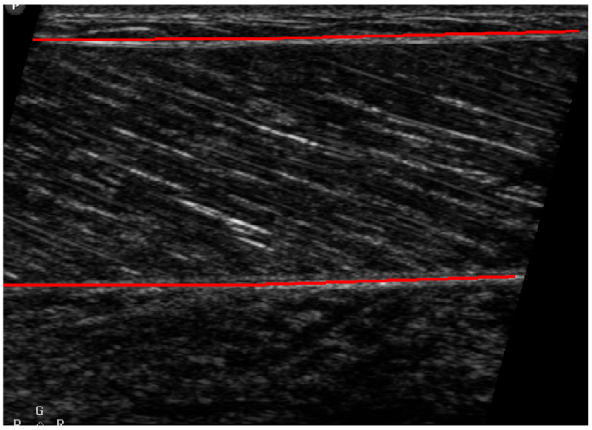


(c)

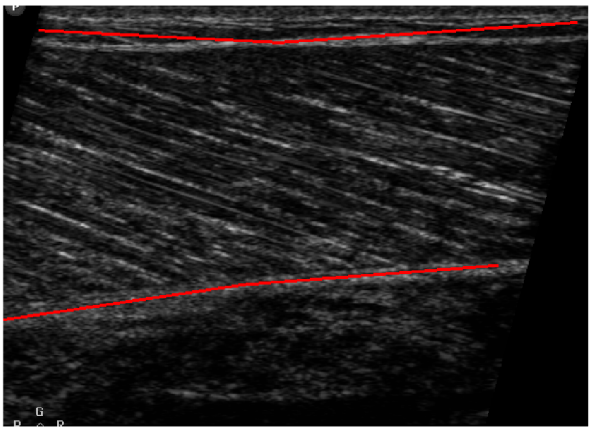
Figure 9.21: The constructed reference fascicle for the gastrocnemius muscle for subject nr. 5.



(a)

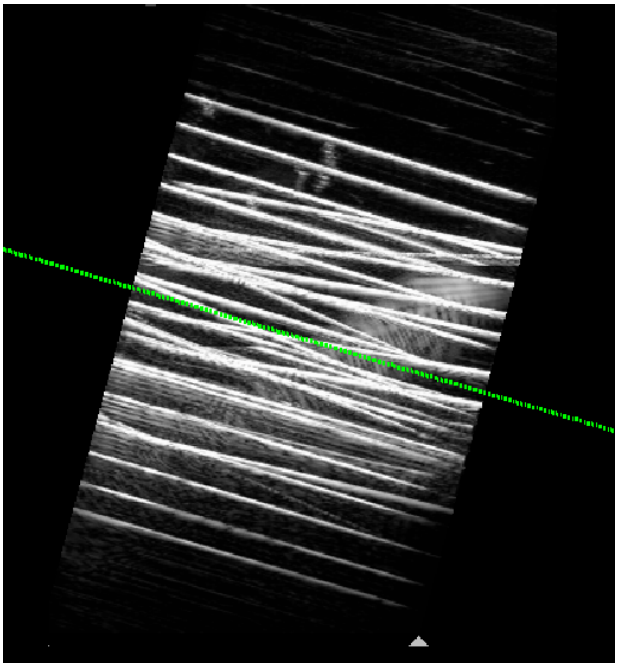


(b)

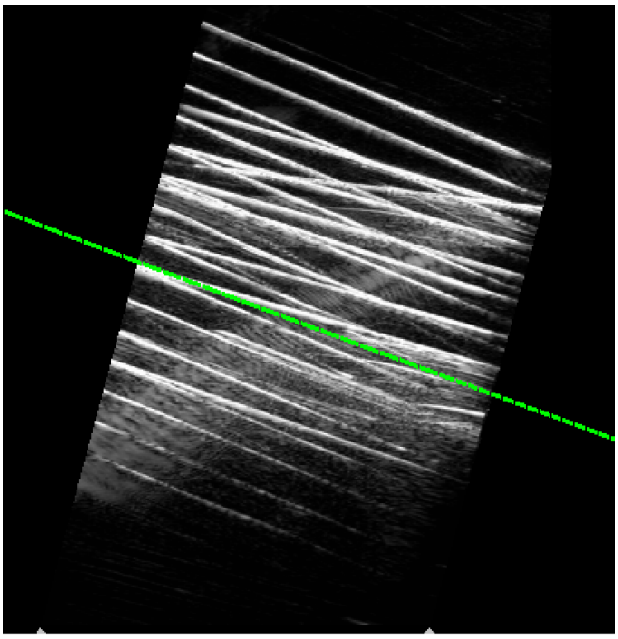


(c)

Figure 9.22: The detected aponeuroses on the gastrocnemius muscle for subject nr. 5.



(a) Image with underlying structure of 16.6°.



(b) Image with underlying structure of 22.4°.

Figure 9.23: The constructed reference fascicle over the images of hair.

has been affected by some of the other structures in the same depth.

The problem with this test is that even though we have the typical variance of the white structures with the black parts in between, as we can see in ultrasound images of actual fascicles, the images cannot be completely compared. Here we have several lines going in different directions in the image, and even though this can happen in ultrasound images of fascicles as well, it usually does not happen to this extent.

Thus, when choosing the median as the representative angle for each depth, we will get something in between the conflicting orientations. Because of that, a construction of a reference fascicle for such a data set should be done in a different way.

To get a good test of validity we need images where we know the exact angle, that more closely resembles a fascicle plane, and preferably a series of images from high to low quality, so we could see exactly where the algorithm stops being well-functioning.

9.5 Discussion

We see that the algorithm gives good, and what we believe to be accurate results in many occasions. However, the manner of how the images are generated has a lot to say. In test set 1, with clear prominent aponeuroses, the algorithm detects the aponeuroses very well. In these images, the fascicles are less prominent, but both the pennation angle and the reference fascicle are still mostly well detected.

When testing on test set 2, where the aponeuroses are less prominent, and the fascicles more prominent, the algorithm has more problems. A rather big issue discovered in this data set was that the noise below the deep aponeuroses made the detected dominant angle fluctuate in such a manner that it gave a great impact on the array indicating a change of dominant direction. This was not an issue in any of the training sets, and so it was not originally believed to be a problem. However, it should not be too difficult to implement a fix to this. If we have a threshold supressing the smallest values when choosing the median angle for each row in the accumulation matrix, a lot of noise will be taken care of, and in addition we could add a check to see how the value of the point picked measures up to the other values in the same row. If the value is too low, then we will supress the peak it creates in the array indicating a change of dominant direction.

After cropping away the noise in the images this caused problems for, we saw that we were mostly able to detect a pennation angle close to what we belive is accurate. It would be interesting to compare the detected pennation

angle relative to the horizontal instead of the deep aponeurosis, to see how much of an impact the detection of the deep aponeurosis has, and to see if any larger adjustments need to be made on the pennation angle detection itself.

With regards to the fascicle length, we see that the curvature of the fascicles mostly fit the underlying structures. Although, in that segment there is also more work to be done, because in some cases the fascicle takes on unwanted shapes. Perhaps some more restrictions when detecting the angles in each depth, or after they have been detected, would make sure this does not happen. This is especially problematic close to the superficial aponeurosis, because it is usually more noise in that area. Mostly however, the problem is that extrapolating a slightly erronous aponeurosis detection, gives large errors.

Clearly, the accurate detection of the aponeuroses needs more work. Perhaps more restrictions on the shape of the aponeuroses would give better results. We also made the choice of picking the center of mass in the transformed image pieces when detecting the accurate location. It might be worth looking into whether this should be replaced by a better measure.

CHAPTER 10

Conclusion

10.1 Summary of problem, main finding and discussion

In this thesis, we aimed to create an algorithm that would robustly detect the pennation angle and the fascicle length of the vastus lateralis and the gastrocnemius muscle from B-mode ultrasound images.

When finding the pennation angle, it was important to detect the dominant orientation of the muscle fascicles accurately, as well as robust detection and good representation of the aponeuroses. In order to calculate the fascicle length accurately, a good representation of the fascicle curvature was needed, especially when extrapolating the fascicles outside the field of view is a necessity. In addition, the calculation of the fascicle length is dependent on the accurate detection and representation of both the superficial and deep aponeurosis.

The resulting algorithm includes detection of the region of interest, detection and modelling of the deep and superficial aponeuroses, filtering of the fascicle plane, fascicle orientation detection, modelling a representative fascicle, and calculation of the pennation angle and fascicle length based on the detected features.

The results are mixed, with mostly good results when the aponeuroses are detected correctly, but there were some difficulties with the aponeurosis detection for the images where the aponeuroses were less prominent. This affected the results a lot, seeing as both the pennation angle and the fascicle length is dependent on a correct aponeurosis detection.

The region of interest algorithm works very well, and there has been no issues with it while using the algorithm on any of the data sets.

Further, we believe that when the issue with the detection of the approxi-

mate location of the aponeuroses is fixed (as discussed in Section 9.5), this will provide good detection results. We experienced this when testing on the two training sets and test set 1, where we in total only got 1(!) error for the approximate detection.

The detection of the fascicle orientations seems to work quite well. Small changes to how we gather the pennation angle or construct the reference fascicle might be worth looking in to, but there are no large problems with this part of the algorithm. More testing and comparisons with manual detections with the detected pennation angle relative to the horizontal line instead of the aponeurosis is required.

Furthermore, modelling the aponeurosis non-linearly is definitely worth looking further into. Either by using splines as in this thesis, or simply using quadratic polynomials might also give decent results. This should be done because it will give more accurate results in the field of view when the aponeuroses are slightly curved, and if it is detected correctly, it should also give more accurate results when extrapolating outside the field of view.

However, more work definitely needs to be done with the detection of the accurate aponeurosis. Even though we mostly get good detections, there are many images where there are small or large errors that will affect both the pennation angle, and the fascicle length. This is the clear bottleneck of the algorithm.

As the algorithm stands today, it definitely get best results when working with images where the probe has been perpendicular to the aponeuroses when being generated. In those cases, there is a lot less error in the accurate aponeurosis detection, and even though the fascicles are not as prominent in these images (training set 2 and test set 1), the results are still mostly good for the detection of fascicle orientation.

10.2 Comparison of past research and known methods

In this thesis, we attempted to build upon much of the research done already, and in particular, we continued to develop many of the ideas from [Jal16]. We have improved the region of interest algorithm to receive both rectangular and parallelogram shaped ultrasound images in DICOM-format, and we have not discovered any problems with this part of the algorithm.

Furthermore, we have used the idea of detecting the change of dominant direction to detect the approximate location of the aponeurosis, and robustified it by exchanging it with the Radon transform in cases where there is

doubt.

In the background material we looked at, all the aponeuroses were represented as lines. We have attempted to represent the aponeuroses as curves. Of course, this presents new issues. In cases where a straight line would have represented an aponeurosis nearly perfect, our curved aponeurosis might weir off and prevent large errors. However, that is something we can continue to develop, and it is something we believe would be beneficial from being looked further into. The more accurate the aponeuroses are represented, the better results the algorithm will get. The aponeuroses will not be accurately represented with a straight line in a lot of cases.

10.3 Further work

There are several aspects of the algorithm that could, and should, be improved upon. Some of them are small tasks, like making sure that noisy areas in the image does not disrupt the approximate location of the aponeuroses, or dividing into grids that takes the image size into account by increasing or decreasing the amount of pieces we divide the image into. Some are larger tasks, like robustifying the accurate detection of the aponeuroses, as discussed above.

Further, it would be very interesting to attempt to use the algorithm as a basis for tracking. For instance, in [Cro+11], they implemented an efficient automatic fascicle tracking method based on the Lucas-Kanade optical flow extension, but needed to manually select the region of interest and the fascicle endpoints in the first ultrasound image of each video sequence. Since our method automatically detects these features, looking at the possibility of combining it with the Lucas-Kanade optical flow extension or other tracking methods could be interesting to pursue.

Bibliography

[Alb16] Fritz Albregtsen. "Object Description - Feature Extraction". 2016. URL: http://www.uio.no/studier/emner/matnat/ifi/INF4300/h16/undervisningsmateriale/inf4300-2016-f05-description.pdf.

[BB65] MJ Bazin and JW Benoit. "Off-line global approach to pattern recognition for bubble chamber pictures". In: *IEEE Transactions on Nuclear Science* 12.4 (1965), pp. 291–293.

[Cro+11] Neil J. Cronin et al. "Automatic tracking of medial gastrocnemius fascicle length during human locomotion". In: *Journal of Applied Physiology* 111.5 (2011), pp. 1491–1496. ISSN: 8750-7587. DOI: 10.1152/japplphysiol.00530.2011. eprint: http://jap.physiology.org/content/111/5/1491.full.pdf. URL: http://jap.physiology.org/content/111/5/1491.

[Fra+98] Alejandro F. Frangi et al. "Multiscale vessel enhancement filtering". In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI'98: First International Conference Cambridge, MA, USA, October 11–13, 1998 Proceedings*. Ed. by William M. Wells, Alan Colchester, and Scott Delp. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 130–137. ISBN: 978-3-540-49563-5. DOI: 10.1007/BFb0056195. URL: http://dx.doi.org/10.1007/BFb0056195.

[Fuk+97] Tetsuo Fukunaga et al. "Determination of fascicle length and pennation in a contracting human muscle in vivo". In: *Journal of Applied Physiology* 82.1 (1997), pp. 354–358.

[GD87] Carl Gans and Frits De Vree. "Functional bases of fiber length and angulation in muscle". In: *Journal of Morphology* 192.1 (1987), pp. 63–85.

[GHV04] Michael van Ginkel, CL Luengo Hendriks, and Lucas J van Vliet. "A short introduction to the Radon and Hough transforms and how they relate to each other". In: *Quantitative Imaging Group, Imaging Science & Technology Department, TU Delft* (2004).

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN: 013168728X.

[HMW96] Olaf Hellwich, Helmut Mayer, and Gerhard Winkler. *Detection of Lines in Synthetic Aperture Radar (SAR) Scenes*. 1996.

[Jal16] Frida Elén Jalborg. *Automatic detection of skeletal muscle architecture features*. 2016.

[Lin06] Tom Lindstrøm. *Kalkulus 3. utgave*. Postboks 508 Sentrum 0105 Oslo: Universitetsforlaget AS, 2006, pp. 403–405. ISBN: 978-82-15-00977-3.

[LM08] Tom Lyche and Knut Morken. "Spline methods draft". In: *Department of Informatics, Center of Mathematics for Applications, University of Oslo, Oslo* (2008).

[Rad17] Johann Radon. "Uber die Bestimmung von Funktionen durch ihre Integralwerte langs gewissez Mannigfaltigheiten, Ber". In: *Verh. Sachs. Akad. Wiss. Leipzig, Math. Phys. Klass* 69 (1917).

[RHW09] Manku Rana, Ghassan Hamarneh, and James M Wakeling. "Automated tracking of muscle fascicle orientation in B-mode ultrasound images". In: *Journal of biomechanics* 42.13 (2009), pp. 2068–2073.

[Teaa] Healthline Editorial Team. *Gastrocnemius*. URL: http://www.healthline.com/human-body-maps/gastrocnemius-muscle (visited on 05/24/2017).

[Teab] Healthline Editorial Team. *Vastus Lateralis*. URL: http://www.healthline.com/human-body-maps/vastus-lateralis-muscle (visited on 05/24/2017).

[Wan+13] Michael Wang et al. "Imaging transverse isotropic properties of muscle by monitoring acoustic radiation force induced shear waves using a 2-D matrix ultrasound array". In: *IEEE transactions on medical imaging* 32.9 (2013), pp. 1671–1684.

[WF16] Q. R. Wei and D. Z. Feng. "Extracting Line Features in SAR Images Through Image Edge Fields". In: *IEEE Geoscience and Remote Sensing Letters* 13.4 (Apr. 2016), pp. 540–544. ISSN: 1545-598X. DOI: 10.1109/LGRS.2016.2523560.

[Zho+12] Yongjin Zhou et al. “Dynamic measurement of pennation angle of gastrocnemius muscles during contractions based on ultrasound imaging”. In: *BioMedical Engineering OnLine* 11.1 (2012), p. 63. ISSN: 1475-925X. DOI: 10.1186/1475-925X-11-63. URL: http://dx.doi.org/10.1186/1475-925X-11-63.

[ZZ08] Yongjin Zhou and Yong-Ping Zheng. “Estimation of muscle fiber orientation in ultrasound images using revoting hough transform (RVHT)”. In: *Ultrasound in medicine & biology* 34.9 (2008), pp. 1474–1481.

[ZZ11] Heng Zhao and Li-Qun Zhang. “Automatic tracking of muscle fascicles in ultrasound images using localized radon transform”. In: *IEEE transactions on biomedical engineering* 58.7 (2011), pp. 2094–2101.

[ZZ15] Guang-Quan Zhou and Yong-Ping Zheng. “Automatic fascicle length estimation on muscle ultrasound images with an orientation-sensitive segmentation”. In: *IEEE Transactions on Biomedical Engineering* 62.12 (2015), pp. 2828–2836.