Universitetet
i Stavanger

Faculty of science and technology

Master thesis

Study program/specialization:
Automation and signal processing

Spring, 2017
Open / confidential: Open

Author:
Christian Marås

Christian Marås
(signature author)

Instructor: Kjersti Engan
Supervisor(s): Kjersti Engan (UiS)

Thesis title:
Automatic segmentation of bone, skin and synovitis in ultrasound images of finger joints
Norwegian title:
Automatisk segmentering av ben, hud og synovitt i ultralydbilder av fingerledd

ECTS: 30

Subject header:
Image processing, USG, rheumatoid arthritis, synovitis, Anisotropic diffusion, Homomorphic filtering Superpixel, Region growing, Active contour, Morphology

Pages: 81
+ Attachment/other: 32+embeded files
Stavanger, 15th of June/2017
Date/year

Automatic segmentation of bone, skin and synovitis in ultrasound images of finger joints

By

Christian Marås

Master of Automation and Signal Processing
Faculty of science and technology

University of Stavanger

15.06.2017, Stavanger

Abstract

Rheumatoid arthritis (RA) is estimated to affect between 0.3 to 1.5 % of the population. It tends to strike individuals between the ages of 35-50, which is their working age, with every third individual diagnosed with RA becoming work disabled, and up to 85% of the individuals who still can work losing almost 40 days per year on average. Therefore, an accurate measurement of disease activity is crucial to provide adequate treatment and care for patients. The first stage in RA is inflammation of the synovial membrane which is called *synovitis*. Using *ultrasonography* has proven to provide useful information regarding the disease activity. The assessment of disease activity has until now been done visually by doctors by grading the synovitis from 0-3 in the ultrasound images. Making a software to automate these assessments in order to reduce the number of human-dependent discrepancies can be advantageous.

Materials given in this thesis came from the Norwegian and Polish collaborative project, MEDUSA. They included ultrasound images of finger joints and manually annotated data which was used for similarity measurement. The objective of this thesis has been to segment the synovitis in the ultrasound images automatically. Since it develops from the joint area towards the skin, it was necessary to segment skin and bones first. Multiple image processing techniques were tested for the proposed system for segmentation of bone skin and synovitis. Novel methods for segmentation and location of these features were also developed. All the proposed methods were implemented using MATLAB.

The similarity measurement was done by computing the modified Hausdorff distance for bone and skin, whereas the Dice coefficient was used for comparing the synovitis with the annotation data. The results show that the proposed system for segmentation of bone and skin functioned well with 80% of the segmented bone and skin features having a distance under 20px to the annotation data. However, one of the two bones had only 55% under 20px, but had a median of 11px. The proposed system for segmentation of the synovitis gave an overall low Dice coefficient, with the best result giving a median and mean Dice of 58 and 54 respectively using Region growing. However, when inspecting the images visually, most of the segmented synovitis seemed descent.

It was concluded that even though the skin and bone segmentation was good, the proposed methods for segmentation of the synovitis did not yield satisfactory results for future grading of it.

Preface

This thesis concludes a two-year master degree in automation and signal processing at the University of Stavanger. The problem of this thesis were formulated by me, but it could not be done without the materials given by the MEDUSA project. I want to thank my contacts from this project, Marcin Fojcik and Marek Kulbacki, for guiding me and acquiring the materials. I would also like to thank my supervisor at the Unversity of Stavanger, Kjersti Engan, for good guidance and input on the structure and style of the report. Also, I appreciate that she took the time to discuss the thesis in depth at each meeting and gave constructive feedback. At last, I would thank Sondre J. C. Kvamme for taking the time for proofreading the thesis.

List of word and abbreviations

Word	Description
Arthritis	Joint inflammation
RA	Rheumatoid arthritis
Synovitis	Inflammation of the synovial membrane
Lesion	Any abnormal damage or changes in tissue
USG	Ultrasonography
US	Ultrasound
PD	Power Doppler
Hypertrophy	Show increase of size
B-mode	Grayscale ultrasound images that only show hypertrophy
PDSUG	Power Doppler ultrasonography
MEDUSA	Name of the project which this thesis was derived from
DMARD	Disease modifying antirheumatic drugs
bDMARD	Biological disease modifying antirheumatic drugs
MRI	Magnetic resonance imaging
MCP	Metacarpal phalangeal joint (first joint connecting hand and finger)
PIP	Proximal intra joint (second finger joint from the hand)
PDE	Partial differential equations
PM filter	Perona-Malik Filter
HE	Histogram equalization
CDF	Cumulative density function
AHE	Adaptive histogram equalization
CLAHE	Contrast limited adaptive histogram equalization
SE	Structural element for morphological image processing
THT	Morphological top-hat transform
BHT	Morphological bottom-hat transform
SLICE	Simple linear iterative clustering
AC	Active contour
SBM	Snake based method (active contour)
LSM	Levels-set method (active contour)
Dice	Similarity measurement used for comparing two surfaces
HD	Hausdorff distance
DHD	Direct Hausdorff distance
MHD	Modified Hausdorff distance
BJP	Bottom Joint point
ROI	Region of interest in this case the area between the skin and bones
STDEV	Standard deviation
Annotation	Is the manually annotated data from MEDUSA
CC	Creative Commons [1]
Bone1	The left bone in the images
Bone2	The right bone in the images

Nomenclatures

Variable	Description
$\mathbf{I}_{description}$	Matrix description of image $\mathbf{I}_{description}$
$I_{description}(x,y)$	Image $I_{description}$ with pixel in position (x,y)
$\mathbf{I}_O(x,y)$	Original image
${I^{bw}}_{description}(x,y)$	Binary image
$I_O(x,y)$	Pixel intensity of the original image in position (x,y)
t	Iteration interpreted as time t
G_σ	Gaussian kernel with standard deviation σ
Γ	Number of diffusion directions
ΔT	Constant applied for stability for anisotropic diffusion
$\nabla I\ _d(x,y)$	Directional derivation in direction d
k	Constant that regulate the preservation of the edges for anisotropic diffusion
$c(x,y)$	Diffusion coefficient for anisotropic diffusion
$m(x,y)$	Multiplicative noise
$a(x,y)$	Additive noise
$H(.)$	Filter function
$I_O(l)$	Pixels in the original image with intensity l
$\mathrm{H}_O(l)$	Histogram of the original input image
L	Number of gray levels
$P_f(l)$	Probability density function
$\boldsymbol{c_f}(l)$	Cumulative distribution function
$I_{\boldsymbol{c_f}}(l)$	The cumulatively distributed pixels
l^{cf}	Gray level in the cumulative distributed histogram
A	Original dataset
B	Dataset for compared with A
a	A specific data point in A
b	A specific data point in B
D	Distance
$SE(x,y)$	Structural element for morphological image processing
$\cap$	Intersection
$\cup$	Union
$\circ$	Morphological opening
$\bullet$	Morphological closing
T	Threshold
$E_{img}(.)$	External energy from the image
$E_{reg}(.)$	Internal energy from the contour
$E_{seg}(.)$	Joint energy of the evolving contour
λ	A constant used for weighting the relative importance of E_{reg} and E_{img}

Content

Abstract 1
Preface 2
List of word and abbreviations 3
Nomenclatures 4
1 Introduction 1
1.1 Objective 2
2 Background 3
2.1 Rheumatoid Arthritis 3
2.2 Epidemiology 4
2.2.1 Treatment 4
2.3 Grading and assessment system 5
2.3.1 Ultrasonography in Rheumatoid Arthritis 5
2.3.2 Noise in ultrasound images 6
2.4 MEDUSA 7
2.4.1 Materials 7
3 Theory 8
3.1 Preprocessing 8
3.1.1 Diffusion-based filtering using Perona-Malik filter 8
3.1.2 Homomorphic filtering 10
3.1.3 Histogram equalization 11
3.2 Morphological operators 13
3.2.1 Morphological operators on binary images 13
3.2.2 Morphological operators on grayscale images 14
3.3 Segmentation 15
3.3.1 Region growing 16
3.3.2 Superpixel 17
3.3.3 Active contour 18
3.4 Similarity and performance measurement 20
3.4.1 Dice 20
3.4.2 Hausdorff distance 20
4 Segmentation of bone and skin 22
4.1 Proposed system for segmentation of bone and skin 23

- 4.2 Image enhancement 24
  - 4.2.1 Conversion from RGB to grayscale 24
  - 4.2.2 Histogram equalization 25
- 4.3 Filtering 26
- 4.4 Segmentation 27
- 4.5 Boundary description and similarity measurement 28
  - 4.5.1 Boundary description 28
  - 4.5.2 Similarity measurement 29
- 4.6 Proposed system for segmentation of bone skin and synovitis 30

5 Segmentation of inflammation region 31
- 5.1 Masking 31
- 5.2 Filtering 32
- 5.3 Segmentations 32
  - 5.3.1 Region growing 32
  - 5.3.2 Active contour 35
  - 5.3.3 Superpixel 36
- 5.4 Similarity measurement 37

6 Implementation 38
- 6.1 Implementation of bone and skin segmentation 38
  - 6.1.1 Internal functions 38
  - 6.1.2 Reading and extraction of annotations and image data 39
  - 6.1.3 External function 39
  - 6.1.4 Perona-Malik filter 39
  - 6.1.5 Homomorphic filtering and frequency domain filters 39
  - 6.1.6 Similarity measurement 40
- 6.2 Implementation of inflammation segmentation 41
  - 6.2.1 Masking 42
  - 6.2.2 Region growing 42
  - 6.2.3 Superpixel 43
  - 6.2.4 Active Contour 43
  - 6.2.5 Similarity measurement Dice 43

7 Experiment 44
- 7.1 Experiment 4.1 Choice of preprocessing method 45
  - 7.1.1 Results Experiment 4.1 46

7.1.2 Discussion 49
7.2 Experiment 4.2 Segmentation of skin and bones 50
7.2.1 Parameter tuning Experiment 4.2 50
7.2.2 Result Experiment 4.2 53
7.3 Experiment 5.1 Parameter tuning for segmenting inflammation region 54
7.4 Experiment 5.2: Segmentation of inflammation region 56
7.4.1 Result pre-experiment 5.2 Region growing 56
7.4.2 Result pre- experiment 5.2 Superpixel 57
7.4.3 Result pre-experiment 5.2 My active contour 58
7.4.4 Result 5.2 inflammation detection 59
8 Discussion 60
8.1 Detection of skin bones 60
8.1.1 Preprocessing 60
8.1.2 Segmenting 61
8.1.3 Boundary description and similarity measurement 61
8.2 Detection of inflammation 62
8.2.1 Masking 62
8.2.2 Preprocessing 62
8.2.3 Segmentation 63
8.2.4 Similarity measurement and materials 64
9 Conclusion 65
9.1 Future work 65
References 1
Appendixes 1
Appendix-A Medical 1
Appendix-A.1 Atlas from of USG MEDUSA 1
Appendix-A.2 MEDUSA presentation 3
Appendix-B Results 9
Appendix-B.1 Additional visual results Experiment 4.2 9
Appendix-B.2 Additional visual results Experiment 5.2 10
Appendix-C Code 13
Appendix-C.1 MATLAB code guide 13
Appendix-C.2 Code 15

1 Introduction

Chronic arthritis is a heterogeneous group of diseases characterized by the long-lasting inflammation of joints, which can influence the general condition of patients. Chronic arthritis is estimated to affect up to 1.5% of the population [2]. *Rheumatoid arthritis* (RA) is the most frequent type of arthritis and has an estimated prevalence of 0.3 to 1.5 % of the population [3] [4]. It tends to strike individuals between the ages of 35-50, which is their working age, with every third individual diagnosed with RA becoming work disabled, and up to 85% of the individuals who still can work losing almost 40 days per year on average [3]. Therefore, an accurate measurement of disease activity is crucial to provide adequate treatment and care for patients.

Using *ultrasonography*(USG) has proven to provide useful information regarding the disease activity. USG or medical ultrasound examination is a method for visualizing the human body structure by using ultrasonic waves (high-frequency sound) directed into the tissues. The first stage in RA is the inflammation of the synovial membrane which is called *synovitis*. Until now, estimates of the inflammation or the *synovitis* activity have been based on a doctor's experience or standardized atlases. The results are registered as a number from 0 to 3, where 0 means no inflammation and 3 represents the highest possible inflammation activity [2] [5]. This method of assessment is unfortunately somewhat subjective since the grading can vary from one doctor to another. Thus, by making a software to automate these assessments to reduce the number of human-dependent discrepancies can be advantageous. Such systems can be used in large clinical trials as well as in everyday clinical practice. Moreover, it can improve the quality of the results from large multicenter studies in which comparability of assessments from different sources is crucial [2].

Until recently, there have only been proposed semi-automatic systems, such as described in [6], where they have used annotation and active contours to detect the inflammation boundary in Ultrasound(US) images. The only published work on automatic detection and grading of inflammation in US images are from the Norwegian and Polish collaborative project, MEDUSA [7], which is further described in Ch.2.4. In the spring of 2014, in the context of recruiting students for manual annotation, this project was presented to me. In the autumn of 2016, MEDUSA was contacted to see if their collected data could be used in this thesis. Hence, the idea for this thesis was derived.

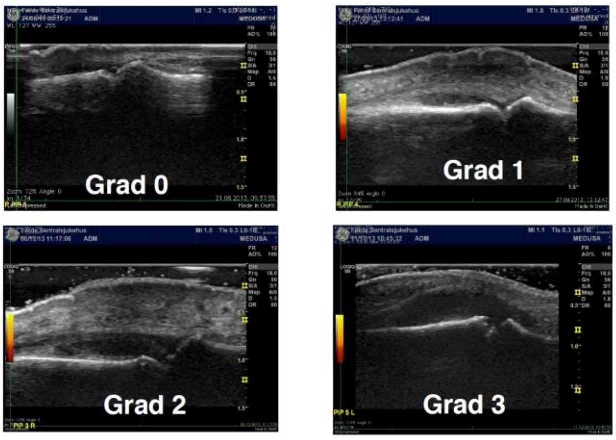


Figure 1: Grading of arthritis which shows how the inflammation develops from the joint (between the bones at the bottom) and extends gradually towards the skin area(top)

1.1 Objective

The objective of this thesis is to automatically segment the inflammation or synovitis region in the US images by using traditional image processing techniques. Since the inflammation develops from the joint area towards the skin as shown in Figure 1, it required to first find a good way to detect the bones, skin and joint locations. This thesis was therefore divided into two main parts:

1. Segmentation of bone and skin
2. Segmentation of inflammation region

The MEDUSA project was concluded in late November 2016, so most of their work was published prior to this thesis. This thesis and the work following it has therefore been independent of that project; MEDUSA was only used as an inspiration.

It was decided that the focus of this thesis should be on the segmentation of bones, skin and inflammation region. Therefore, automatic grading of the synovitis was not addressed. The similarity measurements that were conducted were done by comparing the detected features with the annotation data separately for each part - it will prevent any bad result from the bone and skin segmentation from affecting the inflammation segmentation. This means that only the best results from the bone and skin part were used for inflammation segmentation, and will be further described in Ch.4 and Ch.5.

2 Background

This chapter contains the background information, problems, and earlier work regarding RA. The theory about chronic arthritis is presented first, then some information about the MEDUSA project and the acquired materials.

2.1 Rheumatoid Arthritis

Rheumatoid arthritis is an autoimmune disease that causes chronic inflammation of the joints. Normally the immune system, which is an organization of antibodies and cells, are designed to "seek and destroy" invaders of the body [8] . However, an autoimmune disease such as RA is an illness that occurs when the immune system mistakes the body tissues or cells as foreign objects, inflammation or infection, thus attacking itself [8]. Even though RA is characterized by inflammation of the joint and surrounding tissue, it can also affect other organs such as the skin, eyes and heart. Because of this, RA is referred to as a “systemic illness and is sometimes called a rheumatoid disease” [9]. The symptoms and signs of RA may vary from joint complaints such as pain, stiffness, swelling and functional impairment, to the more constitutional ones such as fatigue and loss of general health [5]. The first stage in RA is inflammation of the synovial membrane that lines the synovial joint, whose medical term is *synovitis*. Figure 2 shows a cross section of a healthy and damaged joint.

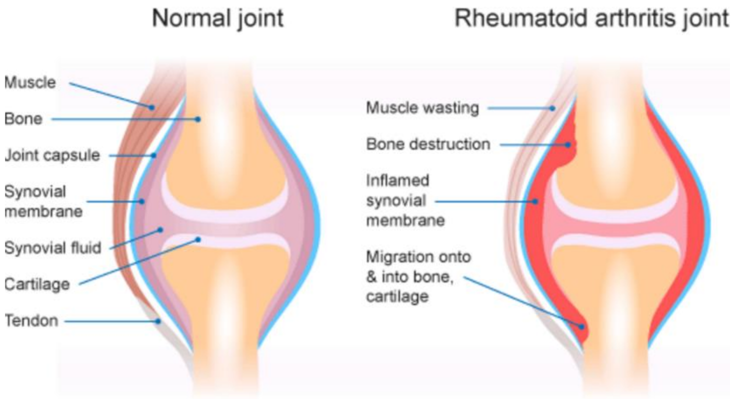


Figure 2: Healthy vs. damage joint [10], [11]. **Reprinted:** Australian Institute of Health and Welfare, under creative commons(CC) CC-BY-3.0 license

2.2 Epidemiology

Worldwide, the annual incidence of RA is approximately 3 cases per 10 000, where women are affected 2-3 times as often as men. However, the sex difference diminishes in the older age groups. RA has a prevalence rate of 0.3-1.5% and increases with age, peaking between the ages 35-50. It is shown that it has some geographic variance, and is uncommon in people under the age of 15. It rises until the age of 80 [3] [12]. The main risk factor for developing RA is genetics which accounts for approximately 50% of the risk; another proven cause can be the use of sex hormones.

2.2.1 Treatment

In addition to training regiments, medicine is used as symptom relievers for small degrees of RA, such as Paracetamol, codeine, and non-steroidal anti-inflammatory drugs. However, in the later stages, stronger medicine is prescribed, such as corticosteroids, disease-modifying antirheumatic drugs (DMARDs) and biologic disease-modifying antirheumatic drugs (bDMARDs) [11]. Since RA develops due to both genetic and environmental factor, som preventive work can be done as well, which includes not smoking, maintaining a healthy diet, and weight control [11].

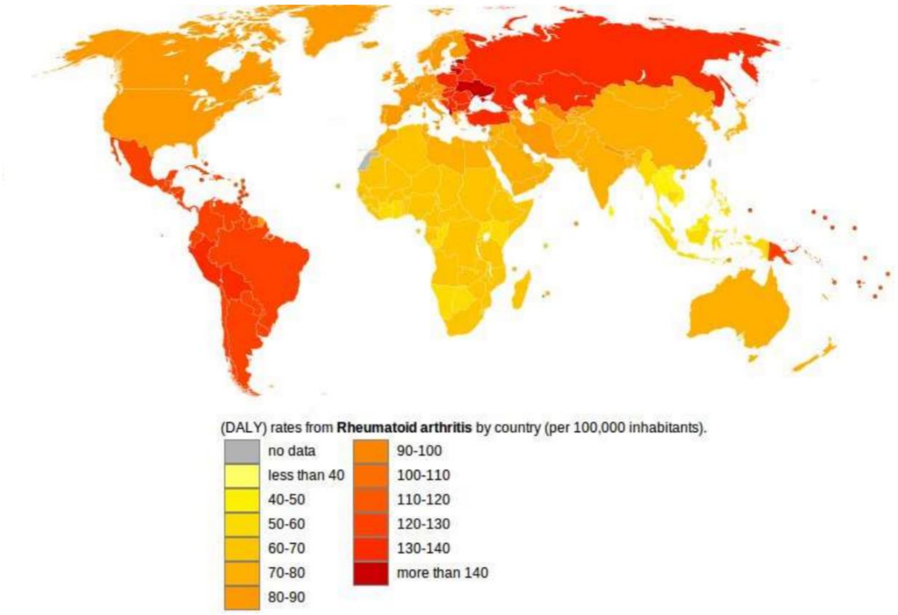


Figure 3: RA rates worldwide.
Reprinted: [13] , Owner: Lokal_Profil under license CC-BY-SA-2.5.

2.3 Grading and assessment system

Currently, there are three main methods of assessment of RA using imaging: X-rays, *Magnetic resonance imaging* (MRI) and USG. X-ray can detect soft tissue swelling, loss of joint space and bone erosion. This method is mainly used in late stages on people with many joints affected, see Figure 4 [12]. In the earlier stages, MRI or USG are used. By using magnetic field and pulses of radio waves, MRI enables more of the soft tissues, such as blood vessels, to show. Also, it can detect bleeding and tumors [14]. In the context of RA, it is used to detect swellings in the bone marrow, thickening of the synovium, and bone erosion, which is shown in Figure 5 [12] .

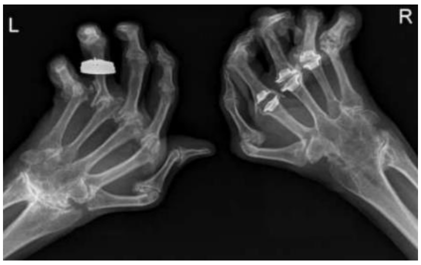


Figure 4 Extensive fusion (ankylosis) at both wrists all of the carpal bones have fused. The patient has had previous joint replacements at the right 2nd, 3rd, and 4th MCP joints, while on the left shows erosions at the MCP joints, with ulnar subluxation. **Reprinted:** © Copyright St Vincent's University Hospital Radiology Department 2017 [15],permission by Prof Eric Heffernan,

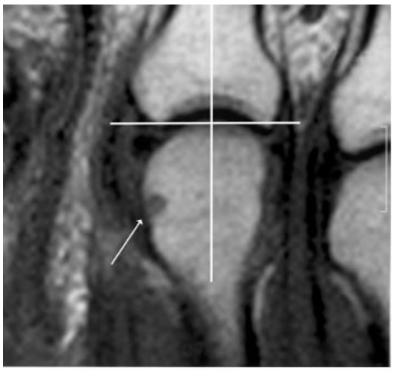


Figure 5: MRI diagnosis of bone erosion in MCP-joint. **Reprinted:** [16] [17], licensed under CC-BY.2.0 .

2.3.1 Ultrasonography in Rheumatoid Arthritis

In the earliest stages of RA, it is shown that use of USG is a valid method of assessment, and it plays an important role in evaluation and monitoring of RA [18] [19]. USG is a method where ultrasonic waves are sent into the body and, by using echo, makes an image of the shape and size of the organs [20]. USG can detect a *lesion* in RA patients, such as joint space widening, a collection of fluid and bone erosion, etc . . . There are two USG methods that are currently being used:

i. "B-mode or gray scale USG: imaging of anatomic structures, which enables visualization of synovial hypertrophy (an increase of size) or effusion which is an increase of fluid in the synovial cavity of a joint" [18].
ii. "Power Doppler USG (PDUSG): blood flow detection, which allows visualization of the movement of blood vessels, therefore detecting increased microvascular blood flow seen in active synovitis" [18]. This is the most used method of the two.

Figure 6 and Figure 7 shows examples of these methods and the grading of the synovitis. Full atlas of the images used in this thesis are depicted in Appendix-A.1.

Although MRI and PDSUG appear to correlate regarding detection of synovitis, there are some differences. PDSUG is not as sensitive when it comes to detection of bone erosion, and is unable to detect bone marrow abnormalities. However, the use of PDSUG is faster, cheaper and easier than MRI, which also requires the use of contrast fluid. PDSUG has also shown to be a useful tool for monitoring changes in a patient under treatment [18].

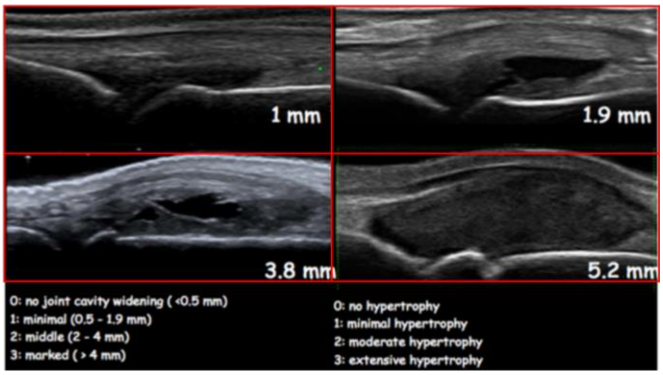


Figure 6: Show B-mode ultrasound images showing only hypertrophy, including different grades of synovitis

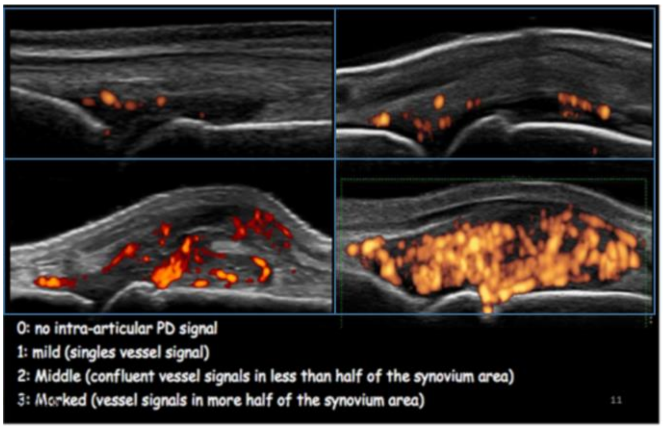

0: no intra-articular PD signal
1: mild (singles vessel signal)
2: Middle (confluent vessel signals in less than half of the synovium area)
3: Marked (vessel signals in more half of the synovium area)
11

Figure 7: Shows ultrasound images with power Doppler that show the blood flow.

2.3.2 Noise in ultrasound images

Ultrasonography is proven to provide useful information regarding disease activity, especially in early stages of RA patients, as described in Ch.2.3.1. Unfortunately, US images are affected by noise because of both internal factors such as moving red blood cells and the different propagation mediums, as well as external factors such as electronic and thermal noise affecting the probe. The resolution of a US image can be better by using higher frequencies, but this limits the depth of the penetration [21]. Either way, reducing the presence of noise in the images to increase the accuracy of both human and computer diagnostics are pivotal.

The main problem is speckle noise which originates from microscopic diffused echoes from the internal organs [22]. Speckle noise Is multiplicative which means that it is locally correlated noise proportional to the signal level. In addition, it is proven to follow a Rayleigh distribution when the scattering density is more than 10 [23] [24]. One of the main issues when trying to find a good despeckling method is to find a method which selectively reduces the noise without altering edges and losing significant features.

2.4 MEDUSA

The MEDUSA project was an international collaboration between i.a the Sogn Og Fjordane University College, Helse Førde, and the Polish-Japanese Institute of Information in Bytom [25]. The project lasted from 15th July 2013 to 30th November 2016. In this period, they made an annotation software which was used to collect the data later used for developing the automatic assessment software. The project also included some semiautomatic approaches, such as one where they looked at segmenting the bone and inflammation region using annotation and image colorization [26] , as well as the development of their methods for segmenting and filtering.

In their automatic assessment approach, they used 403 images from 20 patients, where 38 were used as verification. The best classifier compared to a gold standard gave only a weighted Kappa Choen of 0.38, where 1 is perfect. However, one of the pictures were graded respectively level 0,3 and 1 from three experienced physicians, whilst the 1 by the software. Hence, they concluded that the software gave acceptable results, even though the correlation was low. Figure 8 depict a rough overview of their process of automatically assessing the synovitis region. More about their project can be found on their web page [7] or in the power point presentation in Appendix-A.2.

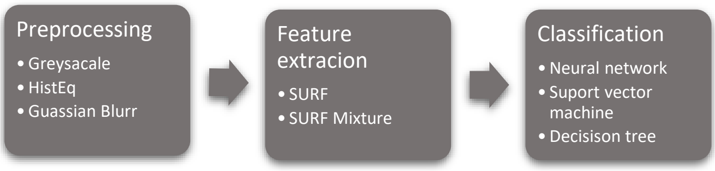


Figure 8: Automatic assessment of synovitis from MEDUSA (rough depiction)

2.4.1 Materials

To reduce bias associated with joint anatomy, the image data acquired is restricted to *metacarpal* (MCP) and *proximal intra-phalangeal joints* (PIP), on finger 2-5 as shown in Figure 9. The annotation data that was given includes annotations of bone, skin, joint and synovitis regions, which is illustrated in Figure 10**.** Images include both power Doppler(PD) and B-mode taken in the medial line of the joint, both of size 960X720. A grading atlas of the images can be found in Appendix-A.1. The images were collected from patients during routine visit at the rheumatology department of Helse Førde. Approval was granted by both the Norwegian Ethics Committee and patients who signed consent forms prior to the collection of the data [2].

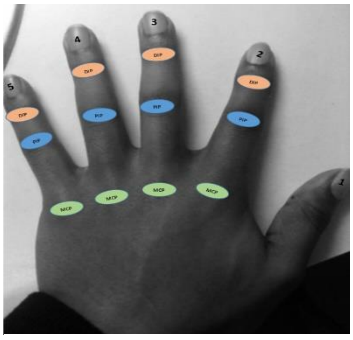


Figure 9: Shows metacarpal (MCP, green) and proximal intra-phalangeal joints (PIP, blue)

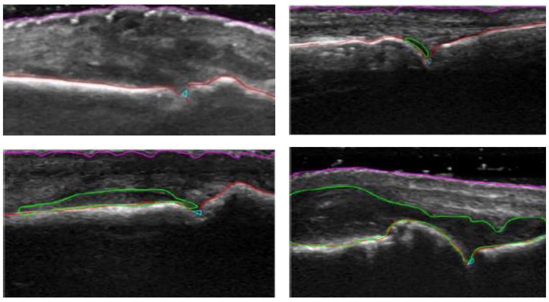


Figure 10**:** Show USG images and annotations with different level of inflammation. From upper left corner level 0, 1, 2 and 3. Bone(red), joint(cyan), skin(magenta), area of inflamation(green)

3 Theory

This chapter outlines the theory behind some of the methods used in this thesis, starting with the theory behind the methods used for preprocessing, followed by segmenting, and finally the similarity measurement.

3.1 Preprocessing

Chapter 3.1 will explain the theory behind some of the preprocessing methods used in this thesis, starting with filtering, followed by histogram equalization.

3.1.1 Diffusion-based filtering using Perona-Malik filter

Diffusing an image is based on partial difference equation (PDEs), and in the case of anisotropic diffusion, the PDEs are typically nonlinear [27]. As described in [27], diffusion *"from chemistry or heat transfer is a process that equilibrates concentration difference without creating or destroying mass."* In the case of images, the mass can be considered as, e.g. pixel intensity or gradient magnitude.

The essence of diffusion-based image processing is to make a multi-scale description of an image. As introduced by Wilkin [27], this can be done by taking the original image and derive it into multiple images. Hence, the original image progresses toward a coarser scale representation. In the process of making the scale space representation of the image, scale generating filters are introduced, traditionally by using a linear filter such as a Gaussian filter. This is obtained by convolving an original image $\mathbf{I}_O$ with a Gaussian kernel $G_\sigma$with standard deviation(scale) of σ, as shown Eq.(3.1.1) [28]. Resulting in the new image $\mathbf{I}_t$ for iterations at time t.

$$\mathbf{I}_t = \mathbf{I}_O * G_\sigma \tag{3.1.1}$$

If $\sigma=\sqrt{t}$, then the Gaussian filter can be achieved through isotropic diffusion by using Eq.(3.1.2). Where $\nabla^2\mathbf{I}_t$ is the Laplacian of $\mathbf{I}_t$.

$$\frac{\partial \mathbf{I}_t}{\partial t} = \nabla^2\mathbf{I}_t \tag{3.1.2}$$

Isotropic diffusion for each pixel is governed by the PDE in Eq.(3.1.3) where $I_t(x,y)$ is the pixel intensity in the image at time t. Even though istotropic diffusion can be used for smoothing and edge detection, it unfortunately results in errors, such as corner rounding and edge localization error.

$$\frac{\partial I_t(x,y)}{\partial t} = \nabla^2 I_t(x,y) \tag{3.1.3}$$

Therefore, an anisotropic diffusion filter can be used instead to reduce noise and minimize the loss of information, including edges as shown in [22] [29] [30]. The main difference is that anisotropic diffusion encourages intra-region smoothing rather than inter-region smoothing. This is done by introducing a diffusion coefficient $c(x,y)$ which varies according to local image gradients in position(x,y) that inhibits smoothing of the edges. Eq.(3.1.4) describe a basic PDE anisotropic.

$$\frac{\partial I_t(x,y)}{\partial t} = \operatorname{div}\{c(x,y)\nabla I_t(x,y)\} \tag{3.1.4}$$

The discreet updating formula for the image intensity using anisotropic diffusion is described in Eq.(3.1.5).

$$[I(x,y)]_{t+1} = \left[I_O(x,y) + (\Delta T)\sum_{d=1}^{\Gamma} c_d(x,y)\,\nabla I_d(x,y)\right]_t \tag{3.1.5}$$

Number of directions is ND, which are usually four in a 2D image, and $\nabla I\ _d(x,y)$ is the directional derivation in direction d at location (x,y). In addition, a constant ΔT is applied for stability, where $\Delta T \leq \frac{1}{4}$ for 2D images with four diffusion directions [27].

Pietro Perona and Jitendra Malik proposed a method that was inexpensive to compute and allowed smoothing while enhancing the edges. This was done by introducing two different diffusion coefficients to implement in the anisotropic diffusion approach [28] [27]. The coefficients are

$$c(x,y) = \exp\left\{-\left[\frac{\nabla I(x,y)}{k}\right]^2\right\} \tag{3.1.6}$$

and

$$c(x,y) = \frac{1}{1+\left[\frac{\nabla I(x,y)}{k}\right]^2} \tag{3.1.7}$$

In both Eq.(3.1.6) and Eq.(3.1.7), k is a constant that regulates the grade of preservation of the edges. The Perona-Malik(PM) filter is proven to be fast to compute and reduce speckle noise in US images which is described in [29] [22] [30].

3.1.2 Homomorphic filtering

As mentioned in Ch.2.3.2, speckle noise is a form of multiplicative noise that follows a probability density function(PDF) with a Rayleigh distribution [31]. Multiplicative noise can be described as:

$$I_O(x,y) = I_{nf}(x,y)m(x,y) + a(x,y) \tag{3.1.8}$$

where $I_O(x,y)$ represents the noisy pixel in the center of a moving window, and $I_{nf}(x,y)$ is the noise free pixel, whereas $m(x,y)$ and $a(x,y)$ represent multiplicative and additive noise respectively. Since the multiplicative part is dominant, Eq.(3.1.8) can be denoted to Eq.(3.1.9) [32].

$$I_O(x,y) = I_{nf}(x,y)m(x,y) \tag{3.1.9}$$

If logarithmic compression is applied to an image with speckle noise, the multiplicative noise is transformed to additive noise [32]. This is because the logarithmic compression changes the statistic of the speckle noise so that the local mean becomes proportional to regional variance, rather than the standard deviation [32], thus affecting the high-intensity tail of the Rayleigh distribution more than the lower one. Hence, the speckle noise becomes closer to white Gaussian noise, see Figure 11.

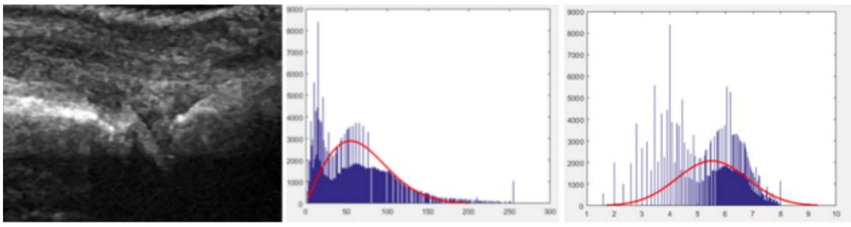


Figure 11:Illustrates a histogram of the original image to the left with a PDF with Rayleigh distribution, follow by logarithmically compressed image with a Gaussian distributed PDF.

The homomorphic filtering process is done by first logarithmically compressing the noisy image, followed by calculating the FFT, and then applying a denoising filter function. Thus, by taking the denoised image and calculating IFFT, it can be decompressed by taking the exponential function of it. Figure 12 illustrates the process of homomorphic filtering where *H(.)* is the filter function applied in the frequency domain, which normally is a type of high-pass filter [32]. A homomorphic filter is typically used to reduce the variation in the illuminance and the sharpening of an image [33].

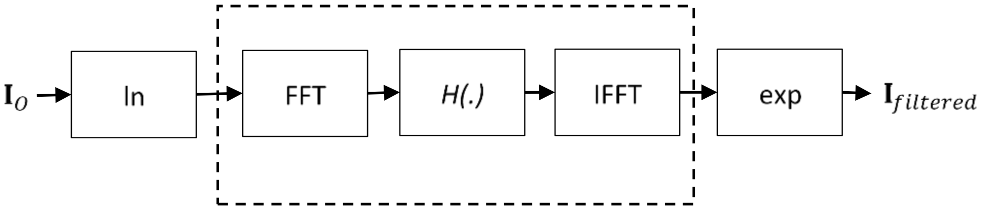


Figure 12: Homomorphic filtering process

3.1.3 Histogram equalization

Histogram equalization (HE) is a popular image enhancement method which increases the contrast of an image. The goal when using this method is to produce an image with a flatter histogram, where all the levels are equiprobable [34]. This is done by simultaneously stretching the histogram and uniformly distributing the pixels in the grayscale range of the image. For a normal 8 bit grayscale image, this is [0:256]. The histogram of an image can be described as $\mathrm{H}_O(l)$,with L number of gray levels and n pixels with gray level l. The normal PDF for the image histogram is defined as:

$$P_f(l) = \frac{\mathrm{H}_O(l)}{MN} = \sum_{l=0}^{L-1} \frac{\mathrm{H}_O(l)}{MN} \tag{3.1.10}$$

where *MN* being the total number of pixels in an image and $P_f(l)$ is the PDF. The cumulative probability distribution(CDF) of the histogram can be described as:

$$\boldsymbol{c_f} = \sum_{l=0}^{L-1} P_{\mathrm{I}}(l) \tag{3.1.11}$$

The CDF $\boldsymbol{c_f}$ is then used as a transfer function when remapping the gray levels of all the pixels in the original image, so that it fills the whole grayscale range [33] [35]. This is done by applying $\boldsymbol{c_f}$ on each pixel:

$$I_{\boldsymbol{c_f}}(l) = \boldsymbol{c_f}[I_O(l)] \tag{3.1.12}$$

and then stretching the cumulative distributed pixels $I_{\boldsymbol{c_f}}(l)$ with gray level l^{cf}, as shown in Eq.(3.1.13).

$$\mathrm{H}_{new} = \frac{\left(I_{\boldsymbol{c_f}}(l) - {l^{cf}}_{min}\right)}{({l^{cf}}_{max} - {l^{cf}}_{min})}(L-1) \tag{3.1.13}$$

However, the histogram does not become fully flat or equalized because it is only a finite number of pixels in the stretch range of an image. In addition, it does not take to account that large peaks in the histogram can be caused by background noise since it does not adapt to local contrast levels [36].

Another approach is using adaptive histogram equalization(AHE), which takes to account the signal level. This is done by computing a map of each pixel by using a different transfer function in each location based on a local neighborhood [37]. However, this requires a more computational power, and it can lead to amplification of noise and contrast in regions with little signal variation. Another method that can reduce these problems is called contrast limited adaptive histogram equalization (CLAHE) [36]. The principal behind this approach is that it uses the knowledge of the slope of the transform computed in the histogram equalization, which is proportional to the histogram of the input image. Hence, it clips and normalizes the input image histogram before computing the transform [37] [38]. Thus, it does not waste histogram bins on background noise. When applying this method, the pixel in the histogram that is above the clip limit is advantageously redistributed equally, rather than being discarded. With low clip factor or contrast factor on the image, the maximum slope of a local histogram will be low – this results in limited contrast enhancement. With a very high clip factor inhibiting the redistribution of the bin values, the result will be equivalent to AHE [36].

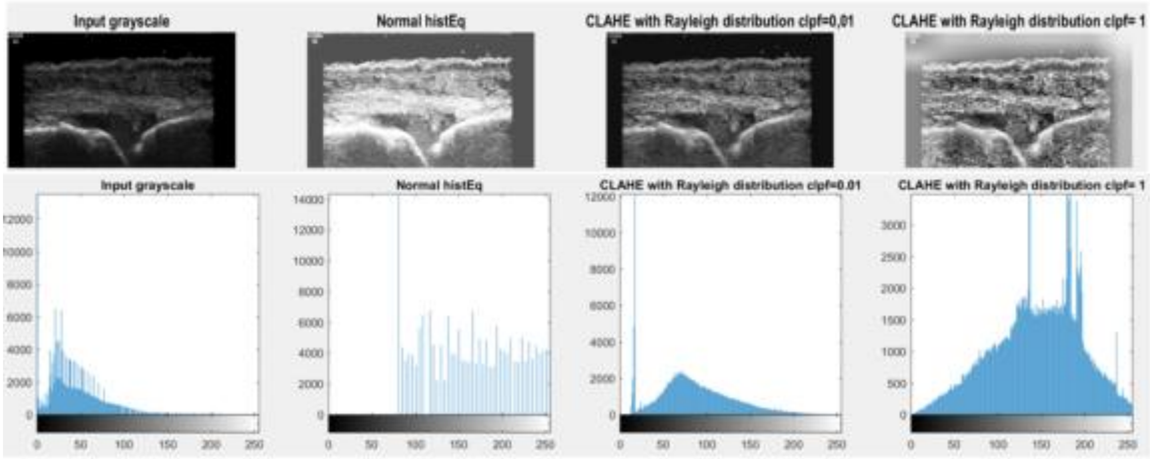


Figure 13: Comparison between normal histogram equalization and CLAHE with low and very high clip factor which show that the contrast increases with the clip factor.

3.2 Morphological operators

The word morphology is term originating from biology and geography which refers to any scientific study of form or structures [39]. The mathematical basis of morphology is set theory such as union and (U), intersection (∩) and complements [40]. For binary images, morphology can be used to shape and form the structures or objects in the image since it only contains ones and zeroes. Morphological image processing can also be used directly on grayscale images to reduce noise, brighten the image, or object detection [41]. In both binary and grayscale images, the essence of morphological operation is to change the state of pixels dependent on the neighborhood pixels using a structural element(SE) and a transition function [39], and then to extract topological and geometrical information in the image. Morphological image processing is used in a wide range of applications today such as inspection, medical imaging, pattern recognition and robot vision [41].

3.2.1 Morphological operators on binary images

Performing an intersection between a data set A and B can be governed by Eq(3.2.1).

$$C = A \cap B \tag{3.2.1}$$

If this set theory operator is applied directly on a binary image, A stands for the binary image data ${I^{bw}}_{\boldsymbol{O}}(x,y)$ and B as the structural element $SE(x,y)$. In Eq.(3.2.2), ${I^{bw}}_{new}(x,y)$ is the output image after moving the structural element over the image. Hence, by deriving the location of where the structural elements fit within a binary object, the information of the image structure is extracted [42].

$${I^{bw}}_{new}(x,y) = \begin{cases} 1\,, & if\ {I^{bw}}_{O}(x,y) \quad and\ SE(x,y) = 1 \\ 0\,, & otherwise \end{cases} \tag{3.2.2}$$

Based on this principal, all the binary morphological operators are spun out of, where the two most common ones are dilation and erosion. Erosion and dilation can also be used in sequence to produce operators such as opening and closing, an example of the effect of these morphological operators are illustrated in Figure 14.

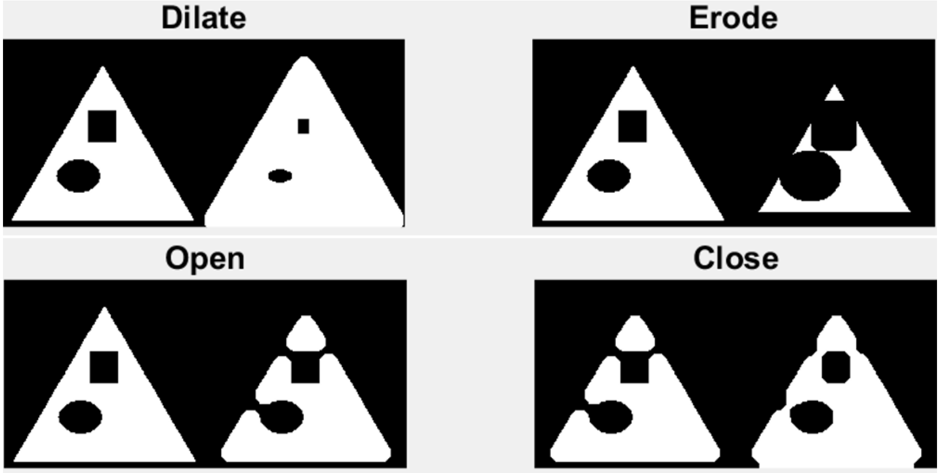


Figure 14: Example of a morphological operator on a binary image. All the images have the original to the left except for the closed example which has the opened to the left. SE is a circular structure with a neighborhood(*NB*) of 15px.

3.2.2 Morphological operators on grayscale images

When morphology is used on grayscale images it can be described as a binary image in a 3D-space, where the third dimension is the brightness of the image [39]. This is done by looking at the maximum and minimum gray level of the image with a moving structural element. Figure 15 illustrate how opening and closing works if SE is a ball structure, and the curve represent the signal surface. In the case of opening, the signal brightness changes because the SE only keeps the maximum level of which it can be translated while still fitting beneath the surface. Because of this, it removes bright objects (pixel cluster) that are small. Complementary, closing will remove small dark pixel clusters [39].

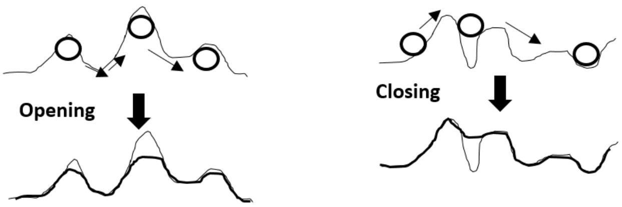


Figure 15: Illustrates the opening and closing on a grayscale image using a ball structure under and over the image signal level

Morphological top and bottom hat transform (THT, BHT), also known as open and closed top hat transform, is a way to detect key markers in an image. THB is done by subtracting the opened image from the original, whereas BHT is done by subtracting the original from the closed image, as shown in Eq.(3.2.3) and Eq.(3.2.4) [41].

$$I_{THT}(x,y) = I_O(x,y) - (I_O(x,y) \circ SE(x,y)) \tag{3.2.3}$$

$$I_{BHT}(x,y) = I_O(x,y) \bullet SE(x,y) - I_O(x,y) \tag{3.2.4}$$

THT can be used as a preprocessing technique on grayscale images, e.g. to adjust uneven illumination and enhance bright objects. On the opposite, BTH is used to enhance dark objects, both THT and BTH are illustrated in Figure 16 [42].

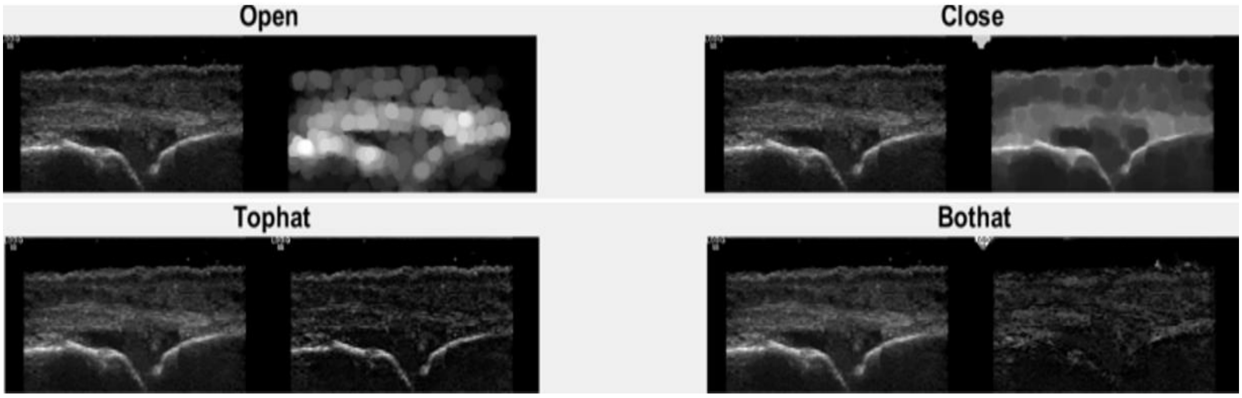


Figure 16: The figure illustrates what effect the opening, closing, THT and BHT on a grayscale image with a circular SE and *NB=15px*. As illustrated, THT can be used to adjust the illumination so that the bone and skin area becomes more distinct.

3.3 Segmentation

Segmentation is a way of separating selected objects or structures from the background in an image. This is done by either identification of pixels belonging inside each object, or by identifying its boundary [43].The most basic form of image segmentation is thresholding, where some are based on image histograms and others on local properties such as mean, standard deviation or gradient. Thresholding can be achieved by using a global threshold on the image, which separates the background and desired structure from each other by a given intensity threshold as showed in Eq.(3.3.1) and Figure 17. This chapter will take a close look at three alternative methods, first region growing, then superpixel, and finally the use of active couture.

$$ {I^{bw}}_{new}(x,y)\begin{cases}1\ if\ I_O(x,y) > T\\ 0\ if\ I_O(x,y) \leq T\end{cases} \tag{3.3.1} $$

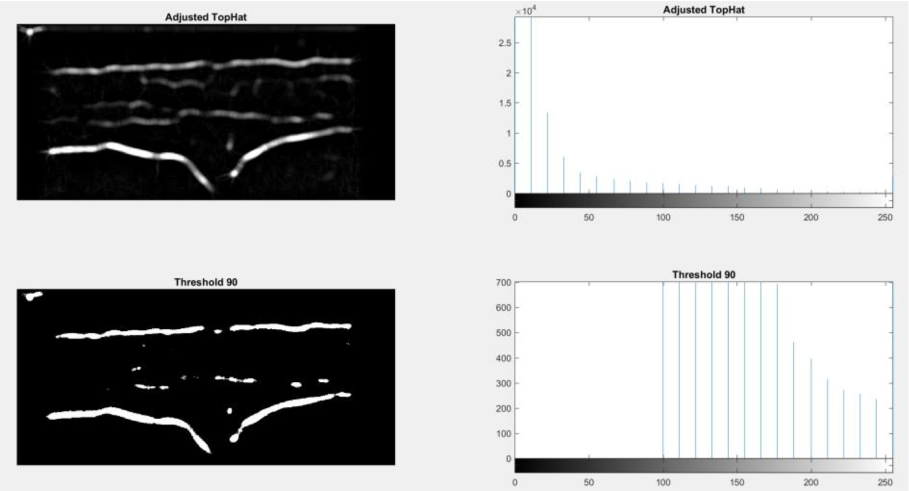


Figure 17: Examples of use of global threshold. The intensity threshold is 1 for I (x, y)>90

3.3.1 Region growing

Whereas conventional thresholding is focused on the difference in intensities using region growing, the focus is on a group of pixels with similar intensities [43]. Region growing is done by finding a pixel or pixels within the structure of interest called seeds. Choice of seed point(s) can be made either manually or an automatically seed location algorithm. The next step is examining the neighboring pixels, one at the time, to decide whether to add it to the region or not. This is done by a uniformity test, e.g. looking at the difference in intensity of each pixel compared to the mean of the region. Hence, if the pixel difference is less than a predefined value, the pixel is included in the region; otherwise, it is defined as an edge pixel. If this predefined value is not chosen correctly, the region growing will leak out of the desired region as illustrated in Figure 18. Therefore, the choice of uniformity criterion is crucial to how the outcome of this method will be [43]. The examining of the neighboring pixel is done until no more pixels are excepted during the region growing procedure. An example of this is showed in Table 1.

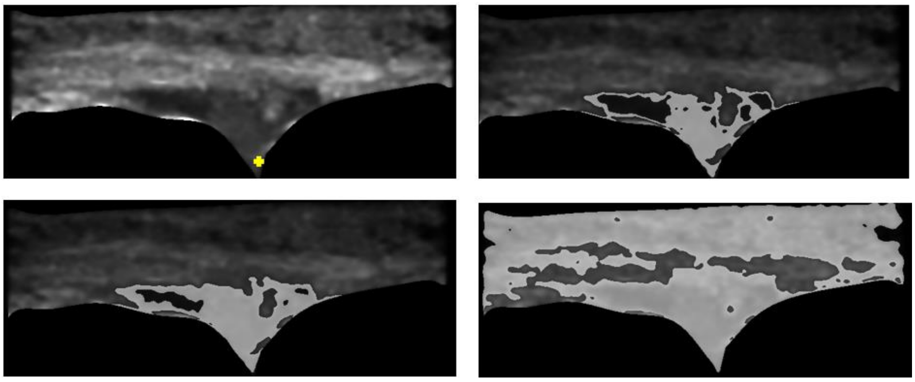


Figure 18: Illustrates what happens when increasing the distance when using this distance to the mean for the uniformity test. The first image shows the seed point in yellow and the region of interest being the dark region which the seed is inside of. Then the increasingly growing region is the white binary part blended with the three last images.

Table 1: Region growing with max difference to mean equal 2 and neighborhood equal 4. Seed pixel is 6 with a start mean value of 6, continuing the previous mean as shown.

Image							Iteration	Mean
1	5	8	5	5	0	7	0(seed)	6
5	3	4	5	2	4	8	1	6
9	1	3	**6**	0	1	2	2	2.5
1	8	4	3	9	6	4	3	5
3	1	2	4	0	5	2	4	4.25
4	6	1	7	1	9	5	5(stops growing)	4.27

3.3.2 Superpixel

The basic principal behind superpixel algorithms is to reduce the image complexity by grouping image pixel into meaningful regions. Following this, one can use the more primitive description of the image for segmentation and feature extraction etc. [44]. There are many different superpixel algorithms to date which can be grouped into two main groups: graph-based and gradient ascent based methods [44]. The former treats each pixel as a node in a graph where the edge-weights between two nodes are equal to the similarity of the neighboring pixels. Hence, it forms the superpixels by minimizing a cost function over the graph. The latter method uses gradient ascent iteratively until a predefined convergence criterion is met, starting from an initial clustering of the pixels. In [44], another method is proposed on how to generate superpixels called *simple linear iterative clustering*(SLICE), which is based on *k*-means clustering. The benefit of SLICE is that it is fast and easy to implement and use. In addition, it only has one input which is the desired number of superpixels. Figure 19 shows an example of SLICE with 200 and 800 superpixels.

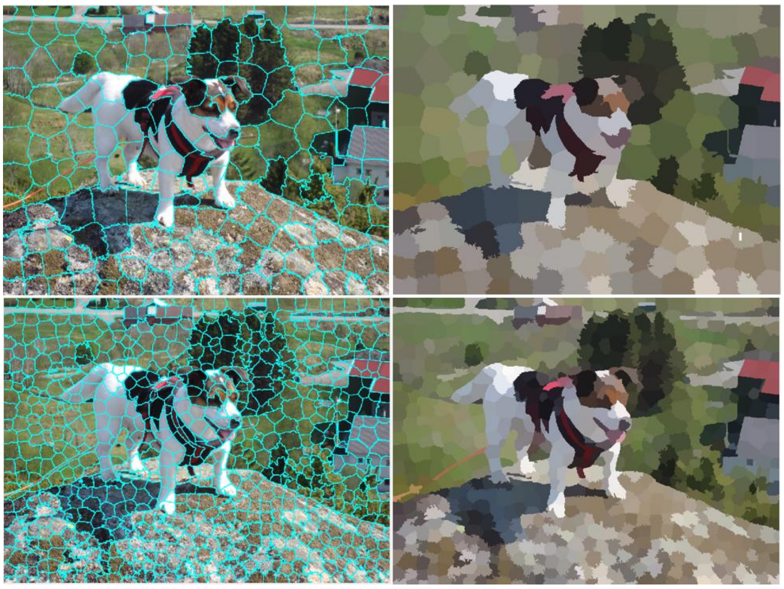


Figure 19: Example of a 960x720-image segmented into 200 and 800 superpixels, then put together based on the mean in each superpixel resulting in a more primitive description of the image. Notice how the superpixel edges adapt to the pixel regions and how the choice of the number of superpixel plays in the resulting segmentation.

3.3.3 Active contour

Active contour(AC) is a popular technique that is used particularly for image segmentation and tracking due to it advantageously being less sensitive to blurred edges and broken contour lines compared to other methods, such as thresholding and edge detection. The basis of AC is to deform and initial contour towards the boundary of the desired object or structure. This is done by using an energy minimization function, normally containing two components: the smoothness of the curve, and one for pulling the curve closer to the boundary [45]. In Eq.(3.3.2), a general example of an energy minimization function is shown.

$$E_{seg}(.) = E_{img}(.) + \lambda E_{reg}(.) \tag{3.3.2}$$

E_{img} is the image based energy that drives the energy closer to the segmentation, based on image features which can be edge based, region based or both [46]. The edge based features utilizes the distance between the contoure and the nearby edges in the image , whilst region based features uses the information of the inhomogenity of the image inside the boundary of the contour. E_{reg} is the internal energy of the evolving contour, which regulates the contour so that the segmenting does not become unstable. This term can be anything from a first-order, second-order smoothnes constraint to shape constraint, etc. To weigh relative importance of E_{img} and E_{reg}, a factor λ is included. Hence, the resulting segmented areas are the results of where the joint energy E_{seg} approaches local mimimum, which is reached at the boundary of the object [45].

There are two main approaches to active contours: parametric and nonparametric. The former originates from the snake based method(SBM) by Kass et al., whereas the latter is based on the level set method (LSM) proposed by Osher and Sethian [45]. The main difference between the original LSM and SBM is the representation of the contour. Using the SBM, the contour is represented by parameterized points, whereas LSM represents the contour as a whole.

The SBM requires that the initial contour, which is the connected form of the input "snake points," is placed close to the desired structure [45]. This helps avoid sources of error such as noise, which can attract the AC, since it is deemed a local minimum [47]. The initial contour using the LSM can be a curve or a surface which works as a boundary that separates one image region from another. This contour is represented as a zero-crossing of a level set function Φ of a higher dimension. Consequently, Φ is an independent variable in $E_{seg}(\Phi) = E_{img}(\Phi) + \lambda E_{reg}(\Phi)$, which is mimimized. Hence, the optimal segmentated contour is extracted by finding the zero-crossing of the function Φ [45].

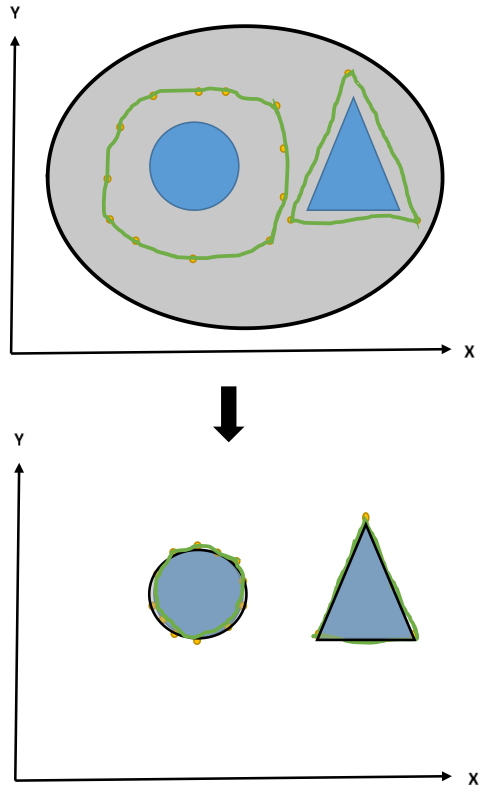


Figure 20: Showing an example of active contour where the desired object is in blue, SBM are represented with the initial points(yellow) connected by the green curve, and LSM is the gray surface with black boundary. As the figure shows, the LSM "snake points" must be more specific and it requires information about the objects that are being segmented. The LSM contour illustrates that it can split into multiple segmented areas.

The advantages of using a nonparametric approach over parametric is that is extremely flexible to the topology of the objects being segmented. Also, the "snake" cannot split into multiple boundaries or merge from multiple initial contours such as the LSM. In addition, LSM can handle multiple objects with different shapes in an image without using complex mathematical methods or numerical scheme [46].

Over the years, extension and changes have been made to both SBM and LSM such as the balloon and distance snake method [47], and the Chan-Vese method [48]. The last one evolved from the Mumford-Shah model and LSM, and uses region properties instead of edge properties when minimizing the energy function. Thus, it does not do edge detection [45].

3.4 Similarity and performance measurement

In this chapter, two methods for measuring similarities between two sets of data will be presented.

3.4.1 Dice

Dice similarity coefficients can be used to compare the similarity between two sets of binary data. Thus, it can be used to compare the similarity between two binary objects in an image. The Dice coefficient is a result of the shared member's times two divided by the total number of both samples [49]. This method does not require the size of the data to be the same, and it results in a similarity number between 0-100 %. In the Eq.(3.4.1), A is true data and B is the measured data

$$Dice(A,B) = \frac{2 * |A \cap B|}{|A| + |B|} \tag{3.4.1}$$

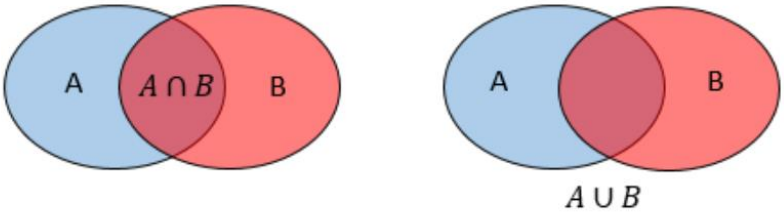


Figure 21: Illustration of intersection and union

3.4.2 Hausdorff distance

In the case where you have two datasets which differ in size and no one-to-one correspondence exists, distance measurements like Euclidian distance do not suffice. In this case, the use of Hausdorff (HD) distance can be used instead [50]. The HD finds the maximum distance between two sets of data by first calculating the shortest distance between each data point in both directions, and then takes the maximum of the distances. Thus, the result of the HD will be the most miss-matching point, which is the farthest point of A to B and vice versa [51]. Eq.(3.4.3) describes the HD and Eq.(3.4.2) shows the direct Hausdorff distance DHD from A to B.

$$D(A,B) = \max_{a \in A} \min_{b \in B} ||a - b|| \tag{3.4.2}$$

Using the given norm $||a - b||$ (e.g. Euclidian), $D(A,B)$ identifies the point $a \in A$ that is farthest from any point in B and measures the distance to its closest neighbor in B. Because of this, it uses the lowest upper bound as the distance.

$$HD(A,B) = \max\{D(A,B), D(B,A)\} \tag{3.4.3}$$

However, this method is very sensitive to outliers in the dataset. Therefore, HD will end up being high even though the datasets or objects are similar. Another approach is using a modified version of the HD (MHD), described in [52] [50]. This method takes the average of the distance to the closest neighbor in both directions and uses the maximum of these as distance measurement [52].

$$D_{mhd}(A,B) = \left\{ mean\big(min(||a-b||) \big) \right\} \tag{3.4.4}$$

$$MHD(A,B) = \max\{D_{mhd}(A,B), D_{mhd}(B,A)\ \} \tag{3.4.5}$$

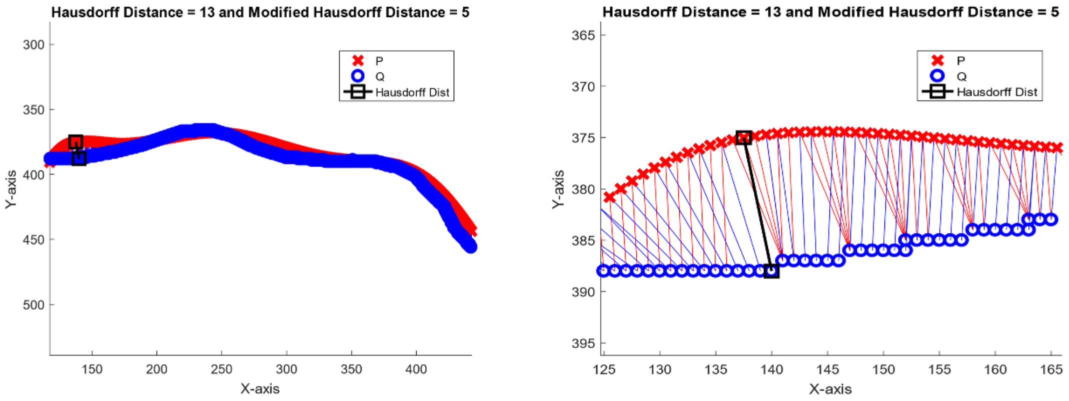


Figure 22: Examples of HD(black) and the different results it gives compare to MHD. The image to the right shows that one point in data **P** can be the closest neighbor to multiple points in data **Q,** and vice versa.

4 Segmentation of bone and skin

As stated earlier in Ch.1.1, the inflammation region develops from the joint area, which is between the bones, towards the skin. Hence, the first step towards the segmentation of the synovitis is to find the skin and bones which can be used later as reference points to locate the synovitis region. In addition, they can be used to mask the image. Thus, only the area between the bones and skin is processed in the inflammation segmentation part. Figure 23 shows an example of annotated bones, skin, and synovitis region, whereas an illustration of the whole proposed system for segmentation of skin, bone and synovitis can be found in Figure 33 in Ch.4.6. This chapter outlines the methods tested for the skin and bones segmentation as depicted in the dashed box in Figure 24.

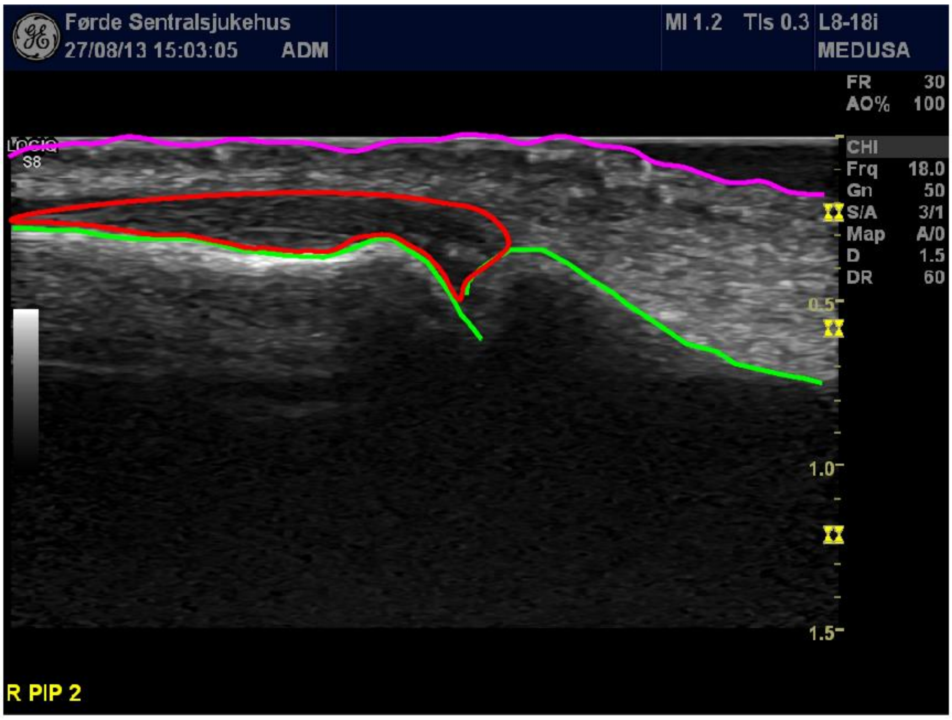


Figure 23: Bone(red), skin (magenta) and synovitis region (green)

4.1 Proposed system for segmentation of bone and skin

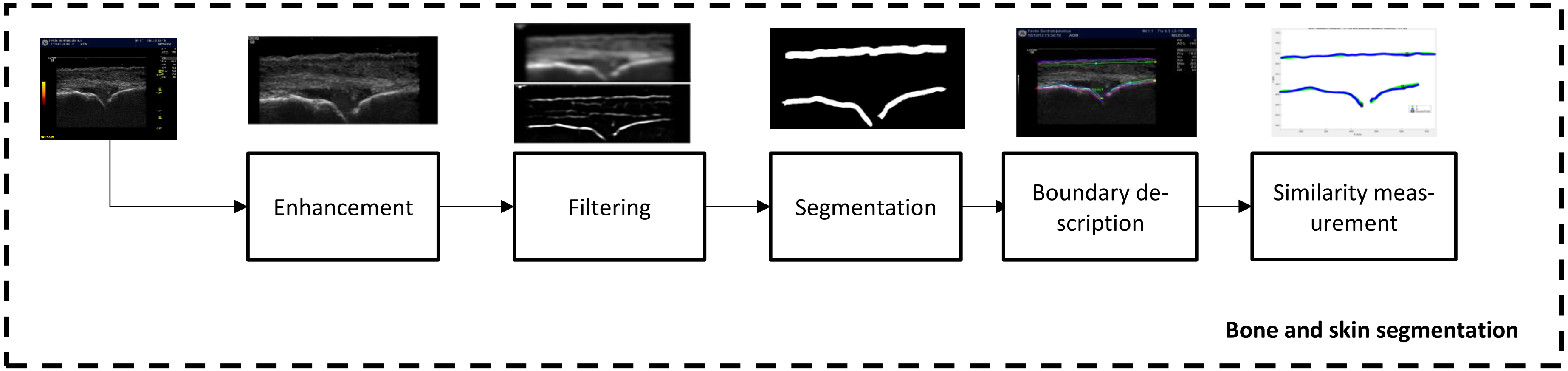


Figure 24: Proposed system for bone and skin segmentation

4.2 Image enhancement

The US images used in this thesis had high variation both regarding the illuminance and the distinctiveness of the bone and skin area. In addition, the dataset contained two types of images, PD and B-mode. For this reason, the first step in the proposed system, box 1 in Figure 24, is an enhancement step. The different image enhancement techniques that were tested will be outlined in this sub-section. Figure 25 illustrate some of the variations in the images.

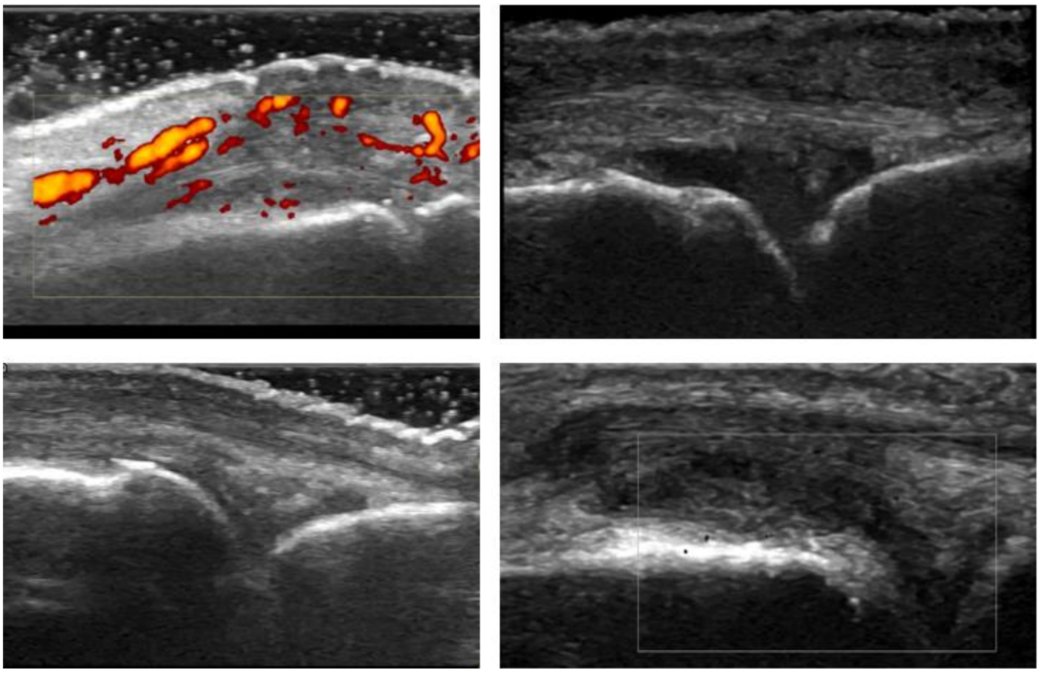


Figure 25: This figure shows an example of some variation in the images.
All input images are RGB, which is also the case for all these images, except for the lower right one, which is grayscale. The last image is an example of where the left bone can be seen easily but not the skin and the right bone. From upper right: PDSUG and high intensity, inflammation region easily seen and the bottom left all features can be easily seen.

4.2.1 Conversion from RGB to grayscale

Before starting the image enhancement, it was conducted an image normalization test by checking if all the images covered the whole grayscale range. The purpose was to find out whether this could be the reason for the illuminance difference, which proved not to be the case. After doing the normalization test, the image was cropped from 960x720 to 756x352 to exclude the margins. Then all the images were converted directly to grayscale. However, during experiment 4.2, it was discovered that the yellow box in the PD images was erroneously segmented as bone, see Figure 26 – therefore, grayscale conversion was used onwards, after removing the yellow channel in the CMYK-converted RGB image.

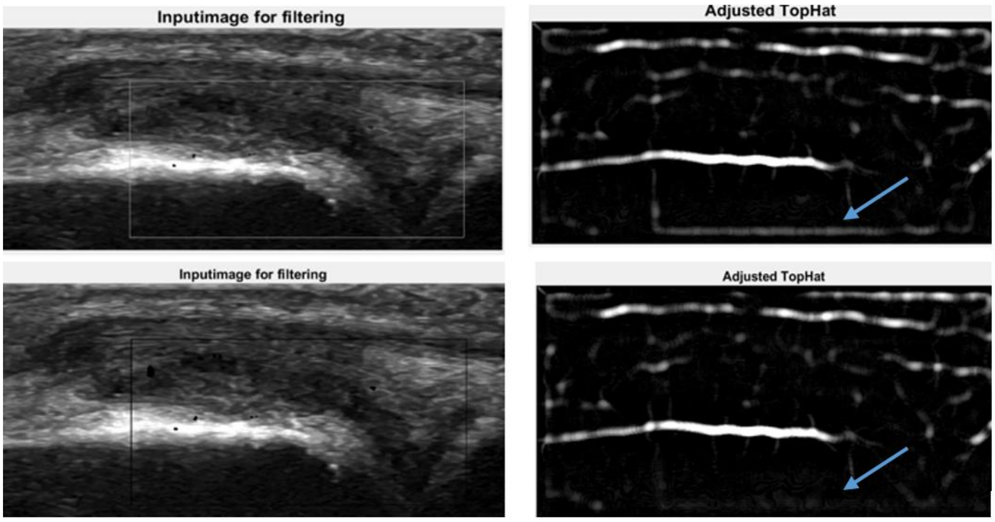


Figure 26: Illustrate the problems regarding bone detection due to the yellow box in PD images. As the upper row show, the yellow box becomes white after grayscale conversion. Thus, when filtering it can be confused with the bones. The lower row shows that by removing the yellow channel, the lower white line in the upper right image disappears.

4.2.2 Histogram equalization

As described in Ch.3.1.3, HE is used for obtaining a flatter histogram where all levels are equiprobable, thus increasing the contrast. Three different HE methods were tested:

1. HE based on the histogram of a reference image
2. CLAHE with Rayleigh distribution
3. CLAHE with uniform distribution

The first equalization method was tested to see if using the histogram of an image where the bone and skin area were distinct could help enhance these features in other images. The reference image that was used is the lower left image in Figure 25, whilst Figure 27 shows an example of the use of this method.

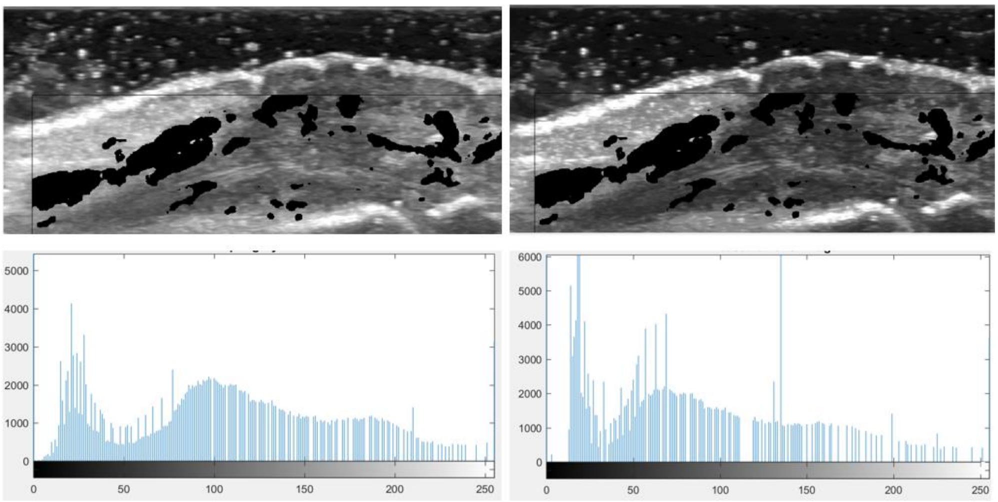


Figure 27: The left colon shows a grayscale image without yellow channel with high intensity.
The right shows that by using the histogram of a "good" image, the bones and the illuminance can be enhanced.

4.3 Filtering

After the image enhancement step, a filtering step is sketched in Figure 24. Since the US images are affected by multiplicative speckle noise, which can be seen in Figure 25, three different filtering methods, which are proven to reduce speckle noise, were tested. The three filtering method were:

1. Perona-Malik filter
2. Homomorphic filtering
3. Gaussian (lowpass) smoothing filter

All the filtering method was done iteratively followed by a morphological top-hat filter to enhance illumination and the bone and skin features.

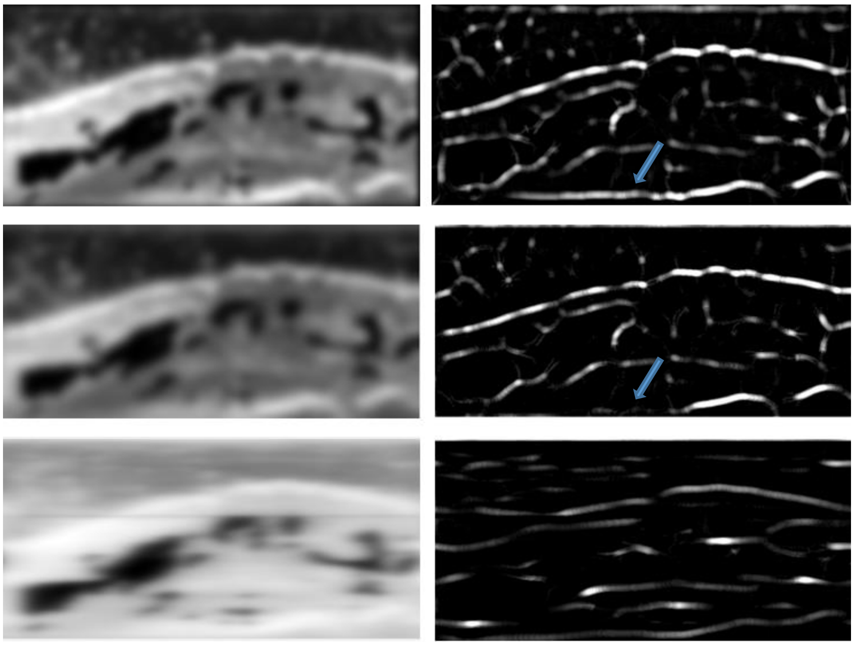


Figure 28: First-row is with the PM-filter, second-row the Gaussian smoothing filter, and the last row is homomorphic high-pass filtering. The images in the left column are the result after doing the top-hat filtering and some illumination adjustment. Left column shows that the PM filter keeps more of the features than the Gaussian filter.

4.4 Segmentation

This subsection describes the segmentation part done after the filtering is finished, corresponding to box 3 in Figure 24. To extract the bone and skin features, two segmenting algorithms were developed; one for the bones and one for the skin. Both algorithms use a global threshold of the filtered image before using a combination of morphological binary operators such as dilation, opening, and closing. Then the bones and skin are found separately before they are put back together. Both algorithms are illustrated in Figure 29, which also shows that both algorithms use the same top-hat filtered input image and global threshold.

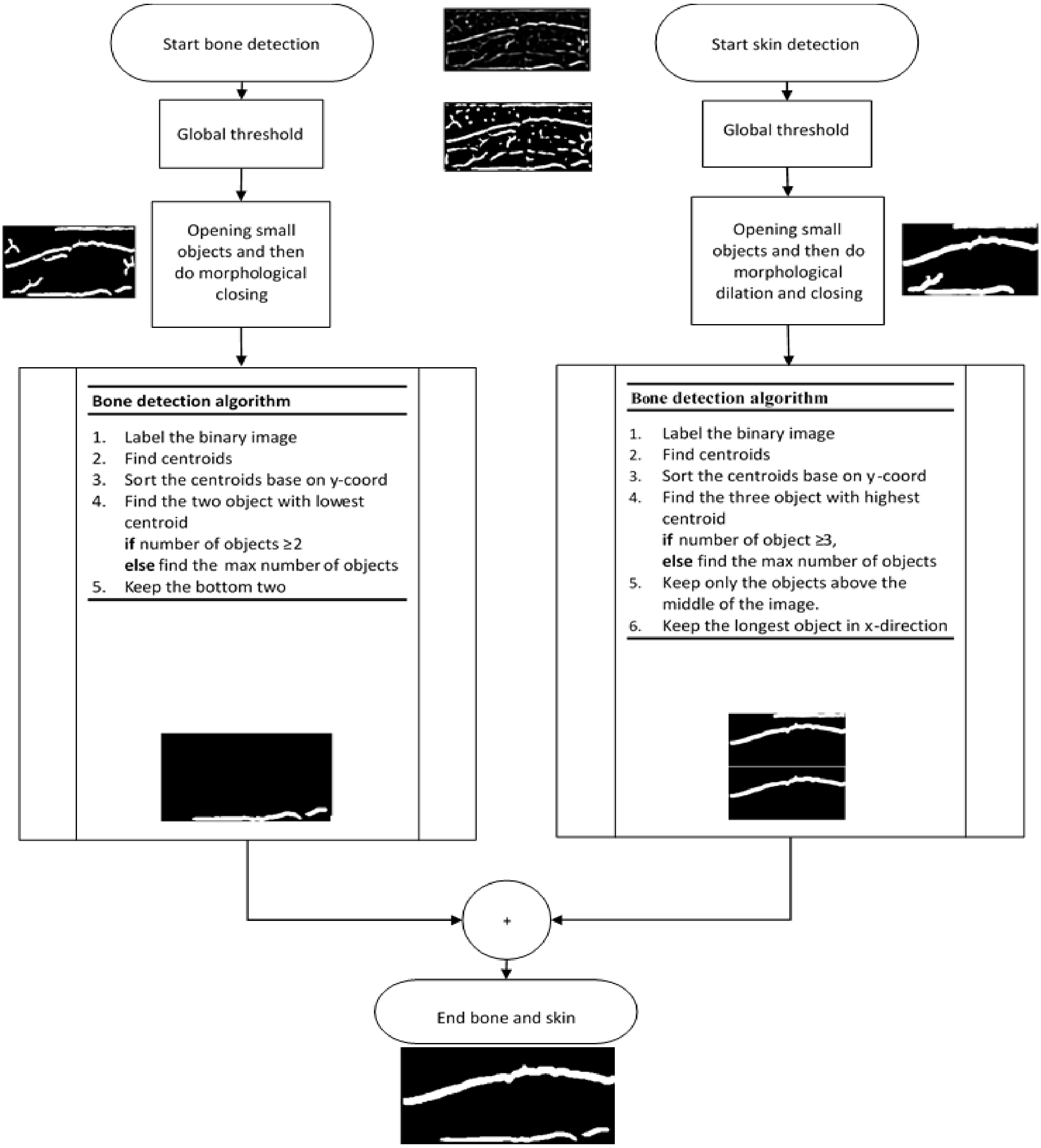

Start bone detection
Global threshold
Opening small objects and then do morphological closing
Bone detection algorithm
1. Label the binary image
2. Find centroids
3. Sort the centroids base on y-coord
4. Find the two object with lowest centroid
if number of objects ≥2
else find the max number of objects
5. Keep the bottom two
Start skin detection
Global threshold
Opening small objects and then do morphological dilation and closing
Bone detection algorithm
1. Label the binary image
2. Find centroids
3. Sort the centroids base on y-coord
4. Find the three object with highest centroid
if number of object ≥3,
else find the max number of objects
5. Keep only the objects above the middle of the image.
6. Keep the longest object in x-direction
+
End bone and skin

Figure 29: Bone and skin segmentation algorithm where the input images is the adjusted top-hat filtered image

4.5 Boundary description and similarity measurement

When the skin and bones are segmented, the boundary description is done before comparing them to the annotation data. In Figure 24, this is depicted in the two last boxes. This subsection outlines how the boundary description of the segmented bone and skin objects was done. Then the similarity measurement of the detected skin and bones, compared to the annotation data, is explained.

4.5.1 Boundary description

After segmenting the bone and skin in the last stage, the resulting binary image was labeled, and then the object boundary was found. To get the representation of boundaries of the detected features as similar to the annotations as possible, the boundary was first divided into the upper and lower layers of the skin, and the bone curves. The skin and bone curves were then transformed by representing them as a 7^{th}-degree polynomial, thus smoothing the boundary of the skin and bones. Figure 30 show the annotation in blue and the compared bone and skin in green.

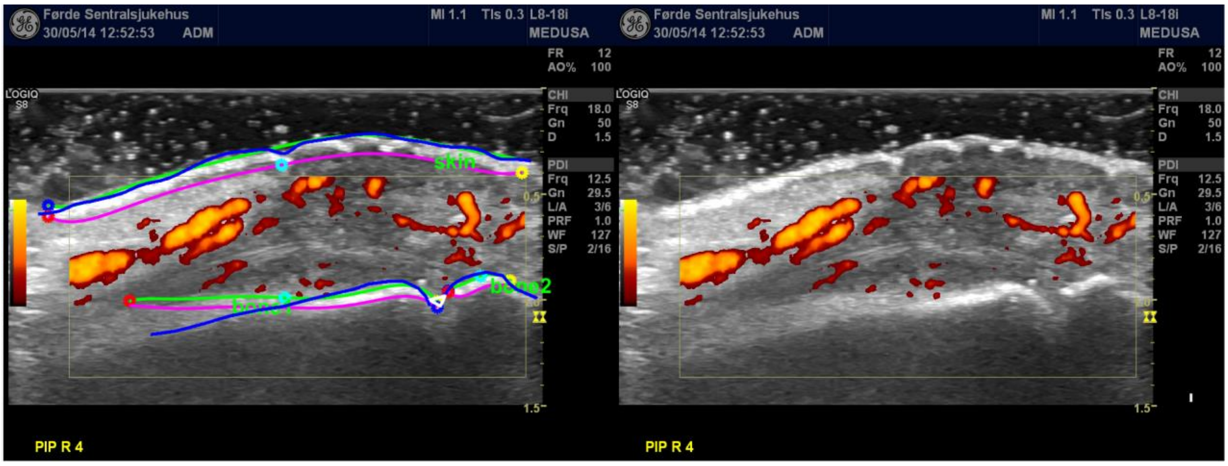


Figure 30: Segmented skin and bones VS annotation data. The annotation data is in blue, and the found bone and skin that was compared is in green. In magenta is the rest of the segmented bone and skin boundary. The lower skin was later used for masking.

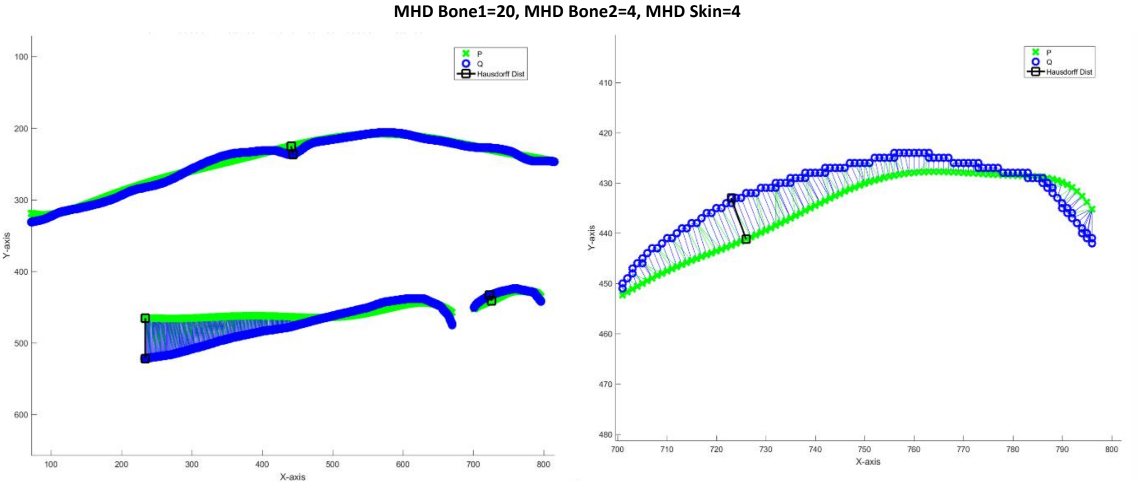

MHD Bone1=20, MHD Bone2=4, MHD Skin=4

Figure 31: Similarity measurement on the features found in Figure 30,
where the annotation is still in blue and the segmented features are in green.
To the right is a closer look at the right bone(Bone2).

4.5.2 Similarity measurement

Since the location of the bone and skin are of paramount importance, and not so much the shape of them, it was decided to restrict the area that was compared. This was done by only comparing the points on the annotated and the segmented features that were between the points with the maximum and minimum x- coordinates. Figure 32 shows an example of this restriction when having a data set **P**(blue) and **Q**(red) with the restriction given in Eq. (4.5.1).

$$\max\left(\min\big((\boldsymbol{P}(x,:),\boldsymbol{Q}(x,:))\big)\right) < restricted\ area < min\left(\max\big((\boldsymbol{P}(x,:),\boldsymbol{Q}(x,:))\big)\right) \tag{4.5.1}$$

The methods used for similarity measurement was MHD which was described in Ch.3.4.2. An example of both MHD and HD used for measuring the similarity between segmented features and the annotation in Figure 30 can be found in Figure 31.

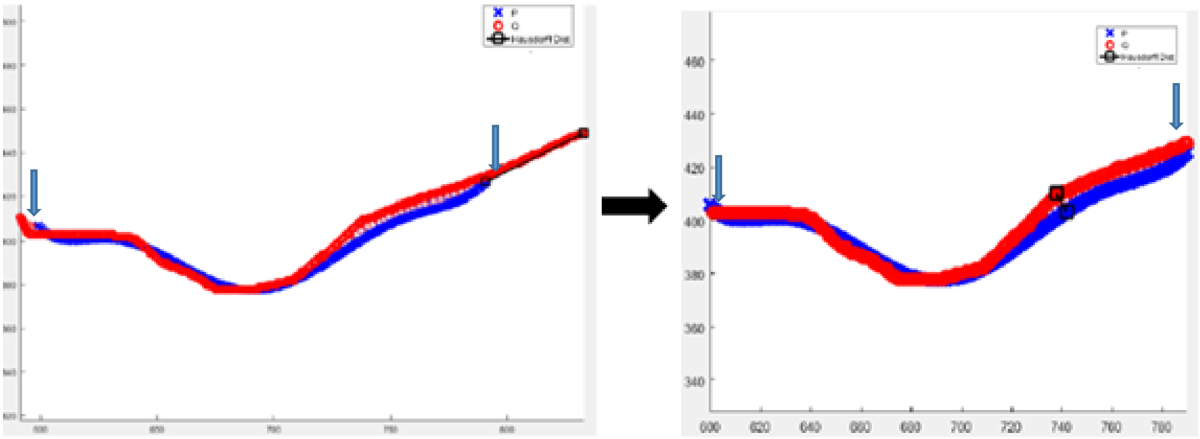


Figure 32: Illustration of how the comparison of a bone was done, using only the data point that had the same vicinity in the x direction.

4.6 Proposed system for segmentation of bone skin and synovitis

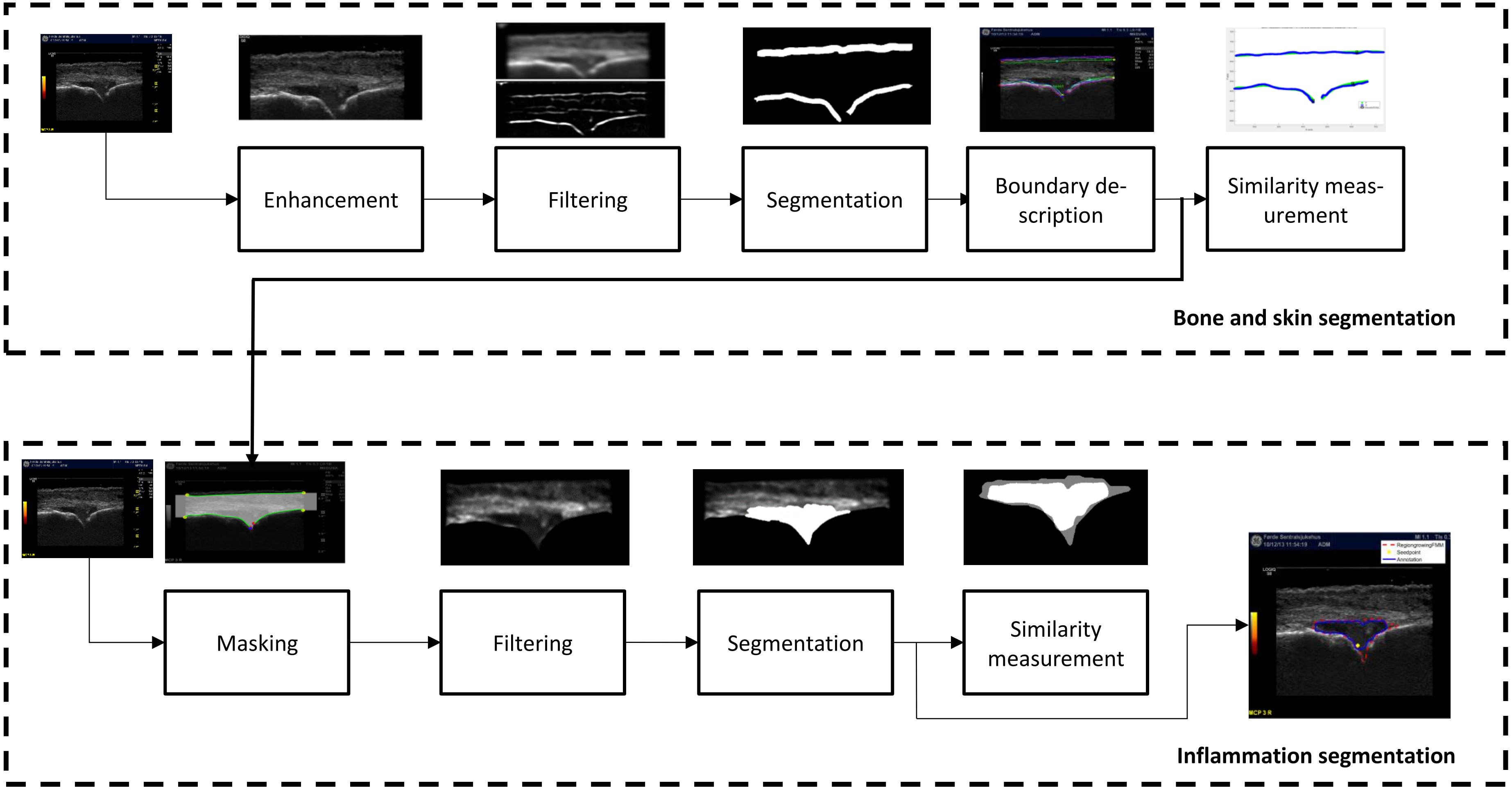


Figure 33: Proposed system for segmentation of bone, skin and synovitis

5 Segmentation of inflammation region

Using the methods from Ch.4, the inflammation regions can now be located more easily by using the segmented bone and skin features. In this chapter, the lower box depicted in Figure 33 in Ch.4.6 will be explained. The masking is explained first, then the filtering, followed by the segmentations of the inflammation region. Finally, the similarity measurement will be explained.

5.1 Masking

As mentioned in Ch.4, the bone and skin features can be used to restrict the potential area of the synovitis. This was done by using the upper bones and the lower skin features as outer limits for a mask which excludes the area outside the bones and skin. However, the bone and skin features did not always go all the way in the x-direction, which leads to the mask being too small. Therefore, it was necessary to define these limits. The size of the images that were used as stated earlier was 920x720, but to avoid the information in the margins, the *x*-upper and lower limit was set to 810 and 60 respectively. In addition, the lowest point in the mask, i.e. the bottom joint point(BJP) was defined, using Eq.(5.1.1). The BJP was defined as the displaced middle point between the left and right bone, with a displacement of 50px in Y-direction. Figure 34 shows the resulting outer points in the x-direction using y-coordinates from the bones and skin in yellow, and BJP in red.

$$BJP(x, y+50px) = \frac{\max(\text{Leftbone}(\text{x},\text{y})) + \min(Rightbone(x,y))}{2} \tag{5.1.1}$$

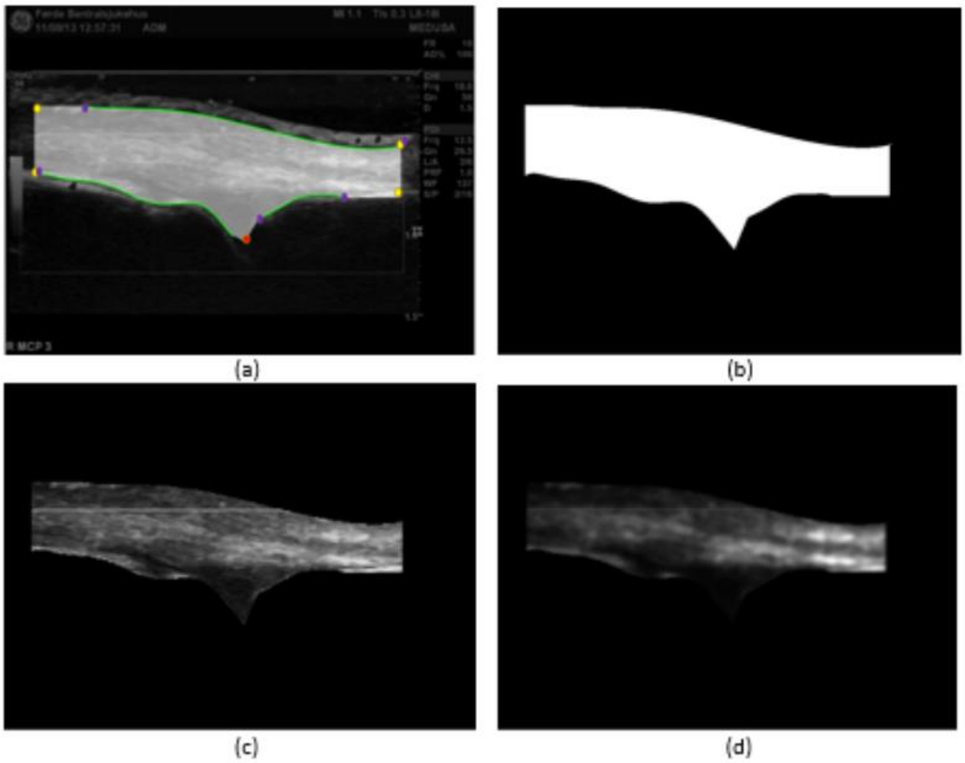

(a) (b) (c) (d)

Figure 34: Shows the steps before the segmenting, where (a) illustrates the masked boundaries and points on the original image. The result of the x-upper and lower limit is the yellow points and the BJP is in red. Image (b) shows the resulting mask, (c) is the resulting image when applying the mask, and (d) is the filtered image.

5.2 Filtering

After masking the area outside the skin and bones, there was a filtering step corresponding to the second box in the lower dashed area in Figure 33. In the bone and skin location part, the focus was to find a filtering method which reduced the speckle noise while preserving the features - this was also the case for this part. However, while filtering the image for bones and skin detection, the focuses were to preserve the small high-intensity regions. Thus, hard filtering was necessary. In this case, enhancing the low-intensity region was the main goal. After doing the experiment described in Ch.7.1, it was discovered that this could be acquired using a homomorphic Butterworth high-boost filter, see Eq.(5.2.1). In addition to this, a light PM-filter with 20 iterations was used beforehand for despeckling and smoothing.

$$\mathbf{I}_{Higboost} = \mathbf{I}_O + \mathbf{I}_{Highpass} \tag{5.2.1}$$

5.3 Segmentations

This subsection will explain segmenting which can be seen in the second-to-last box in the lower dashed area in Figure 33. To segment and extract the inflammation region from the masked area, 3 methods were developed, based on three adaptive segmentation techniques:

1. Region growing
2. Active contour
3. Superpixel

These techniques will be explained further in the following sub-subsections.

5.3.1 Region growing

Region growing, wich was explained in Ch.3.3.1, can be used to segment image regions based on pixels with similar intensities. This is done by initiating the growth with a seed pixel, and then defining the intensity distance that is meant to be included in the growing region. Because of this, the choice of the location of the initial seed point, or pixel, is important. Therefore, it was decided to develop 4 different methods for placing the seed, which will be explained now. All of these seed location methods used the same threshold to minimize the locations of possible seed point. Algorithm 1 shows this method, where the number 10 is just a factor derived from testing to increase the lower threshold, so that it does not include pixels with too low intensities, and thus excludes pixels with intensity values equal zero along the mask boundary or near the bones. The number 30 is the upper average threshold which fitted most of the images. Hence, by adding this average to the *Threshfactor*, the upper threshold will increase if the minimum intensity of the image increases.

Algorithm 1: Thresholding

```
Input: FilteredImage, ROIMASK
    1. Calculating the area of the masked region
    2. Calculating max and min intensity of masked region
    3. Threshfactor= minpixel*10/maxpixel
    4. Threshold= 30+(30* Threshfactor)
    5. LowerBound= 1% of the MASK size
    6. BinaryImage=(minIntensity< FilteredImage <threshold )        (Objectsize> LowerBound)
Output: BinaryImage,Threshold, LowerBound
```

Algorithm 2: Seed 1 and 2 - Centroid and weighted centroid

Input: *FilteredImage*(weighted centroid), *BinaryImage, Leftbone features*

1. Potential seedpoints= all the centroids from the labeled *BinaryImage*
2. Calculate the Euclidian distance between the point on the left bone, which is closest to the joint area, to the centroids of the binary objects in *BinaryImage*.
3. Seedpoint(x,y) = Centroid with minimum Euclidean distance to the point on the left bone.

Output: *Seedpoint(x,y*

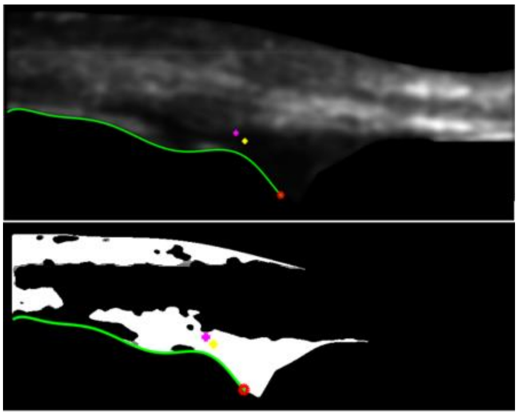


Figure 35: Filtered image followed by the *BinaryImage* after the thresholding algorithm. Weighted centroid(magenta), unweighted centroid(yellow) and the left bone reference point (red)

Algorithm 3: Seed 3- Lowest point

Input: *BinaryImage, Leftbone features*

1. Potential seedpoints= all the centroids from the labeled *BinaryImage*
2. Calculate the Euclidian distance between the point on the left bone, which is closest to the joint area, to the centroids of the binary objects in *BinaryImage*.
3. MinEuclедianDist = minimum Euclidean distance to the point on the left bone.
4. Index= Index of object with unwighted centroid with MinEuclедianDist
5. SeedObject= labeled *BinaryImage(*Index)
6. Seedpixel(x,y)= SeedObject(x,(min(y)))
7. Seedpoint(x,y)= Seedpixel(x,y-displacement)

Output: *Seedpoint(x,y), SeedObject*

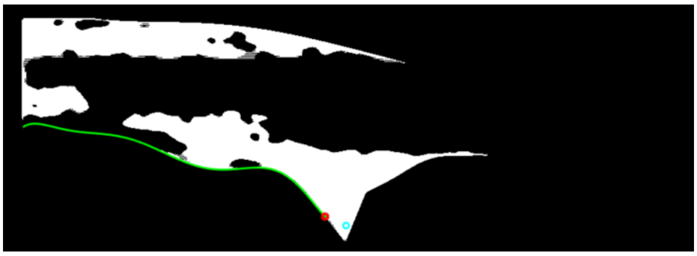


Figure 36: Example of using the displaced lowest point of the object with the closest centroid to the point on the left bone as seed point(cyan).

Algorithm 4: Seed 4- Elliptic mask

```
Input: Filtered image, Threshold, Left bone features
    1. Drawing an ellipse defined by the left bone features(with some displacement) as the center and defined radius in x and y-direction
    2. ElipticMask=Binarisation of the ellipse
    3. 0.5<Potential seed point inside the ElipticMask<Threshold
    4. Calculate the Euclidian distance between the point on the left bone, which is closest to the joint area, to the potential seed points inside the ElipticMask.
    5. Seedpoint(x,y) = Potential seed points with minimum Euclidean distance to the point on the left bone.
Output: Seedpoint(x,y), ElipticMask
```

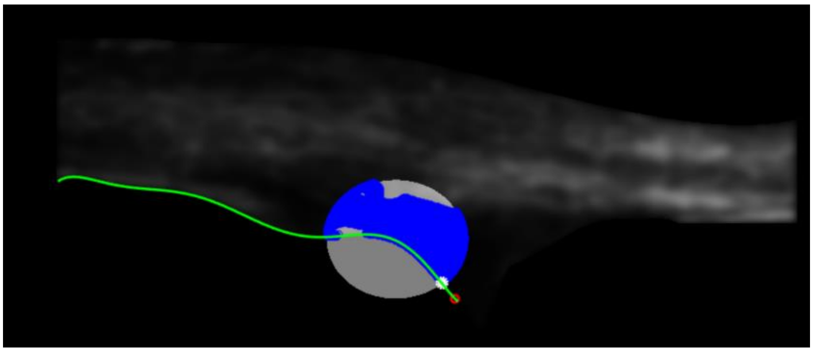


Figure 37: Illustrates using an elliptic mask to delimitate the potential seed point areas (transparent white), The potential seed points are in blue, and the resulting seed point is the white point.

For all the algorithms for seed point location, the left bones were used as a reference, because this feature gave the most constantly good results from the experiments explained in Ch.7.2. To exclude low-intensity pixel that could be detected too close to the bone, as well as dark areas below the joint area, it was added displacement to the seed points in Algorithm 3 and the center of the ellipse in Algorithm 4.

5.3.2 Active contour

Active contour which was explained in Ch.3.3.3, and is a method which can be used for segmentation, given an initial contour. Hence, deforming this contour towards the boundary of the desired object or structure, a segmented area is derived. The initial contours that were tested was an elliptic mask which was explained in Algorithm 4 and the *InitialConour* from Algorithm 5. The last algorithm was developed after discovering the weaknesses using the elliptic mask in Experiment 5.1, see Ch. 7.3. Algorithm 5 still uses the elliptic mask, but only to locate the most likely object in the *BinaryImage* to be the inflammation region. This was done by finding the object that yielded the biggest Dice coefficient when compared to the *ElipticMask*. In addition to using another initial contour, Algorithm 5 used histogram equalization based on a reference image before doing the filtering. Figure 38 and Figure 39 show an example of active contour using an ellipse and the resulting *InitialConour* from Algorithm 5 as the initial contour respectively.

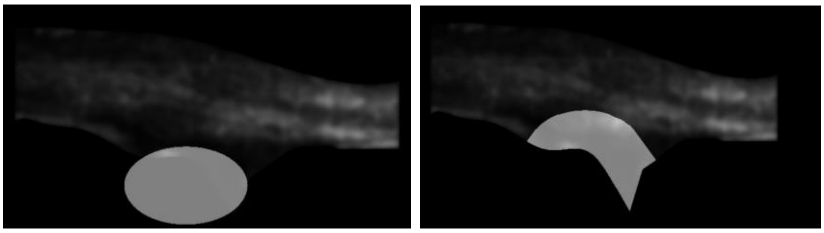
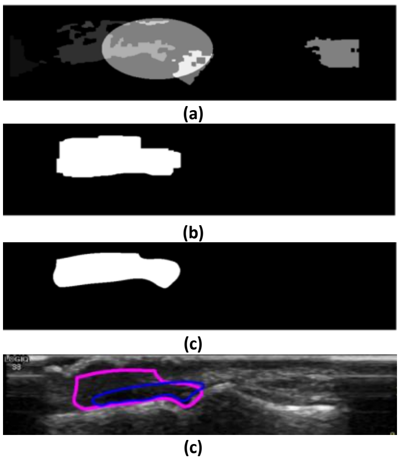


Figure 38: Active contour using an ellipse as the initial contour after 50 iterations of evolving

Algorithm 5: My active contour

```
Input: Grayscale image
    1. Histogram equalization based on a reference image,followed by filtering
    2. Using Algorithm 1and getting  output : BinaryImage
    3. Using Algorithm 4 getting the output : ElipticMask
    4. Calculating the Dice coefficient between all the objects in BinaryImage and ElipticMask  (a)
    5. InitialConour = max(Dice)
    6. Doing dilation and erosion of InitialConour    (b)
    7. Using Chan-Vese AC  with the dilated and eroded  InitialConour  as initial contour, resulting in
       inflamationregion.  (C)
Output: InitialConour, inflamationregion
```



(a)

(b)

(c)

(c)

Figure 39: Visualization of step 4, 6 and 7 in Algorithm 5

5.3.3 Superpixel

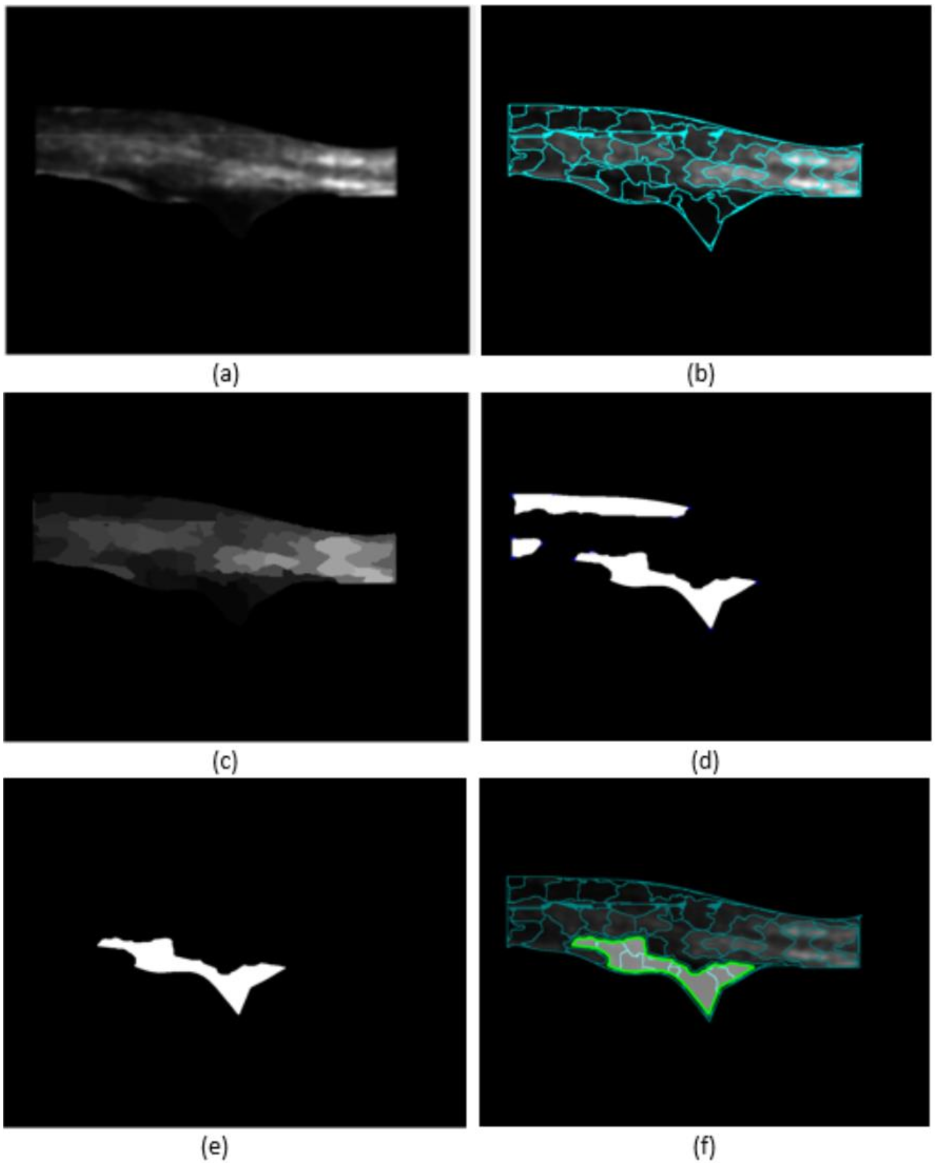


Figure 40: Shows how the segmenting method using Superpixel (a) input image, (b) 150 superpixel, (c)after setting each superpixel to its median, (d) using a threshold, (e) found object with bottom right extrema point with the shortest Euclidian distance to the most right point of the left bone. (f) all put together

Figure 40 illustrate how the segmenting method using Superpixel works. For this method, the Euclidian distance was calculated between the left bone point to the bottom right extrema point. This was because after doing step (c) and (d) it could result in segmented regions below the bones. It turned out, after some testing, that by using the bottom right extrema point this could be avoided.

5.4 Similarity measurement

Corresponding to the last box in Figure 33, the last step after segmenting the synovitis or inflammation region was to measure its similarity to the annotation data. The similarity measurement was done using Dice coefficient described in Ch.3.4.1. To do these similarity measurements, it is required to first binarize the annotation data. Hence, the coefficient could be derived by comparing the binary data, which was logical '1' in both the annotation and the segmented regions. Figure 42 illustrates the union and intersection between the annotated inflammation region and segmenting using region growing. Figure 41 shows a comparison of the different methods and the annotation data.

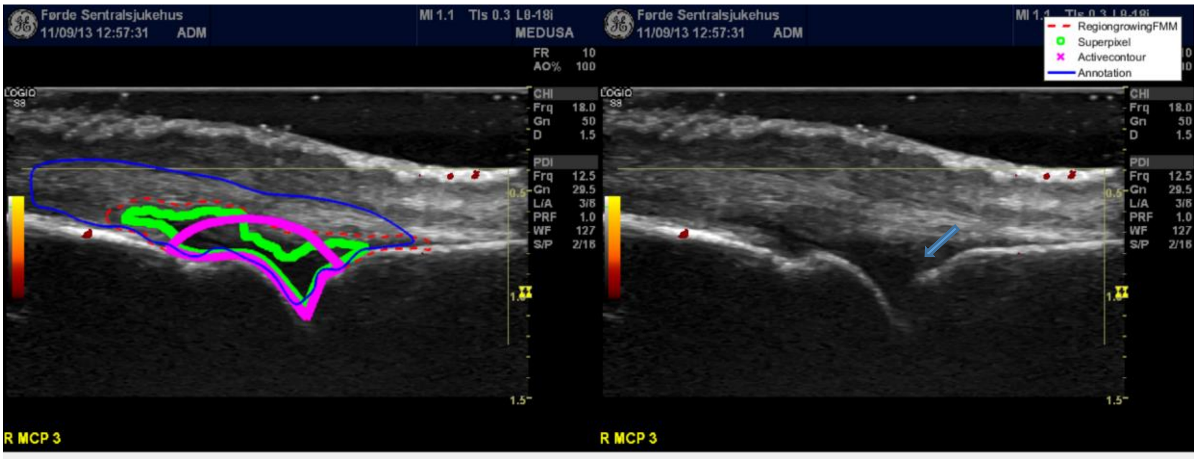


Figure 41: Example of using the region growing(red) and AC(magenta) using the elliptic mask, Superpixel in green and the annotated data in blue. As the image to the left shows, the algorithms find the dark are growing from the joint area(arrow) toward the skin. However, the annotation is often quite generous with the markings which lead to low Dice even though the detection is relatively good.

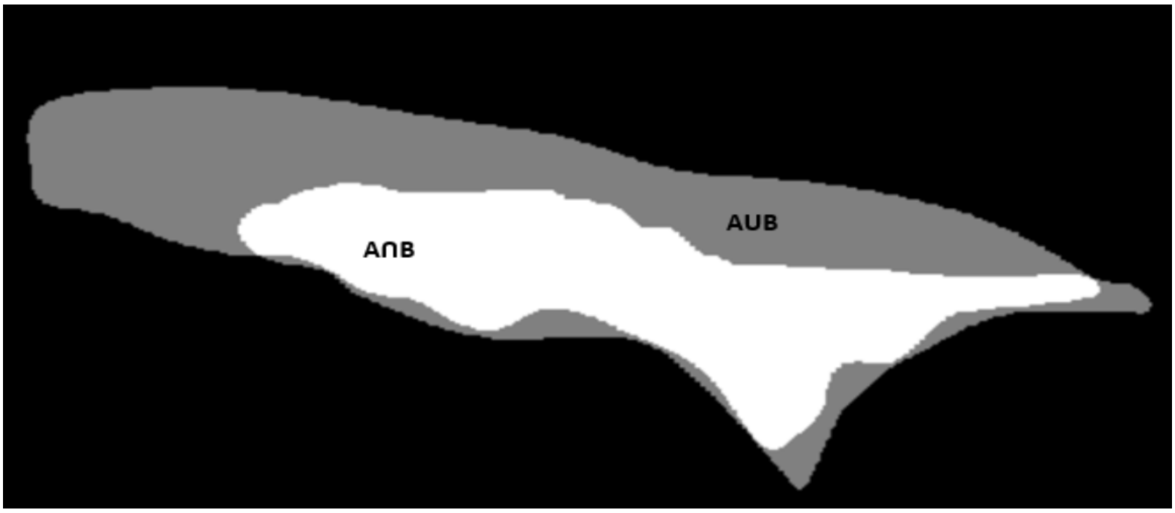


Figure 42: Example of union and intersection between the annotated inflammation region and segmenting using region growing. Resulting in Dice= 62

6 Implementation

All the methods and algorithms were developed and tested in MATLAB, using a combination of the internal functions of the image processing toolbox, external functions from Mathworks, and novel functions [53]. In Table 2 and Table 4, an overview of the most essential functions and where they were collected from can be seen, whereas the codes can be found in Appendix-C.

6.1 Implementation of bone and skin segmentation

This chapter will explain the implementation of the bone and skin detection, where the internal functions will be explained first, followed by the external and novel functions.

Table 2: An overview of the most essential functions used in the
MATLAB implementation of the bone and skin detection methods.
The name of the internal MATLAB functions is in cursive behind the functions.
Where there is checked in two places, it means that some alteration were done to the original function

Methods		**Internal**	**External**	**Novel**
Reading and extraction of annotations and image data				**X**
RGB→CMYK			**X**	
Grayscale conversion (*rgb2gray*)		**X**		
Cropping (*imcrop*)		**X**		
HE and CLAHE (*imhist, histeq and adapthisteq*)		**X**		
Homomorphic filtering and frequency domain filters			**X**	**X**
Gaussian smoothing filter (*imgaussfilt*)		**X**		
PM filter			**X**	**X**
Morphological operators (*imopen, imclose, imtophat, bwmorph*)		**X**		
Illumination adjustment (*imadjust*)		**X**		
Labeling (*bwlabel*)		**X**		
Boundary description (*regionprops, bwboundaries, polyval and polyfit*)		**X**		
Similarity measurement			**X**	**X**

6.1.1 Internal functions

The internal functions listed in Table 2 are in cursive under each method respectively, e.g. *histeq* in combination with *imhist* and *rgb2gray* was used to do histogram equalization based on the reference image histogram.

```
ImgGrayEqRef=histeq(ImageOriginalGray,imhist(rgb2gray(ReferenceImage)));
```

CLAHE was implemented using the *adapthist* function which is based on [36]. This function allows one to choose distribution and adjust the clip limit

```
AdaptHistCroppedUniform=adapthisteq(ImageOriginalGray,'clipLimit',0.01,...
    'Distribution','uniform');
```

6.1.2 Reading and extraction of annotations and image data

Before making the algorithm used for reading and extraction of annotation data, the annotation was structured and sorted into two classes, C1(with inflammation) and C2(without inflammation). Then all the image names in the annotation data were indexed to find the number of annotations complementing each image. The annotations were normally structured as showed in Table 3, where the number of annotations complementing each image is 5.

Table 3: Normal annotation structure and names

	Number of	X(n)	Y(n)
Image name	5		
Bone	341		
Bone	202		
Joint	305		
Region of inflammatory synovitis degree of inflammation/hyperplasia	804		
Skin	390		

The following numbers are the sum of the number of points in each direction for each annotation. In multiple cases, the number of and the order of the annotations varied. Therefore, it was necessary to check if each annotation existed, which was done by comparing the annotation names with a string with the normal annotation names depicted in Table 3. If some annotations are missing, the program jumps over that iteration and goes to the next index. String comparison was also used to check if the corresponding annotations had inflammation data when applying the algorithms for reading and extracting annotations, as seen Figure 43. The string comparison was done by using the internal MATLAB function `strncmp`.

6.1.3 External function

As mentioned in Ch.4.2.1, a RGB to CMYK conversion was performed to remove the yellow channel in the image. Since the image processing toolboxes did not have this function, it was acquired from [54]. The other external functions were implemented for the same reason. All the external functions, except for the RGB to CMYCK conversion, were altered for this implementation – these functions will be explained separately in the following sub-subsections.

6.1.4 Perona-Malik filter

PM-filter was implemented using the 2D anisotropic functions from [55], which uses the algorithms described in Ch.3.1.1 and the diffusion coefficient proposed by Perona and Malik. The only difference in this Implementation is that instead of four (N, S, W and E) diffusion directions, it uses 8 (N, S, W, E, NE, NW, SE and SW). Table 2 shows that the novel section is also crossed out for this method; this is because alterations were done regarding the convolution part. The change that was done was the replacement of `imfilter(I,h,'conv')` with `conv2(I,h,'same')`, thus increasing the filtering speed with an average of 1 second for each image.

6.1.5 Homomorphic filtering and frequency domain filters

The homomorphic filtering function was based on [56], which uses the same pipeline shown in Figure 12. Some changes were done to open up for the use of multiple frequency domain filters and make them iterative. The frequency domain filters implemented was the Gaussian and Butterworth high-pass and high-boost filter based on [57].

6.1.6 Similarity measurement

The similarity measurement was implemented using a combination of HD and MHD-based functions made by [58] and [59] respectively. However, some changes were done: the restriction described in Ch.4.5.2, and; enabling both HD and MHD to be extracted from one function. As mentioned in Ch.4.5.2, the similarity measurement was only done by computing the MHD, so the HD implementation [58] was mainly used for plotting and restricting the data.

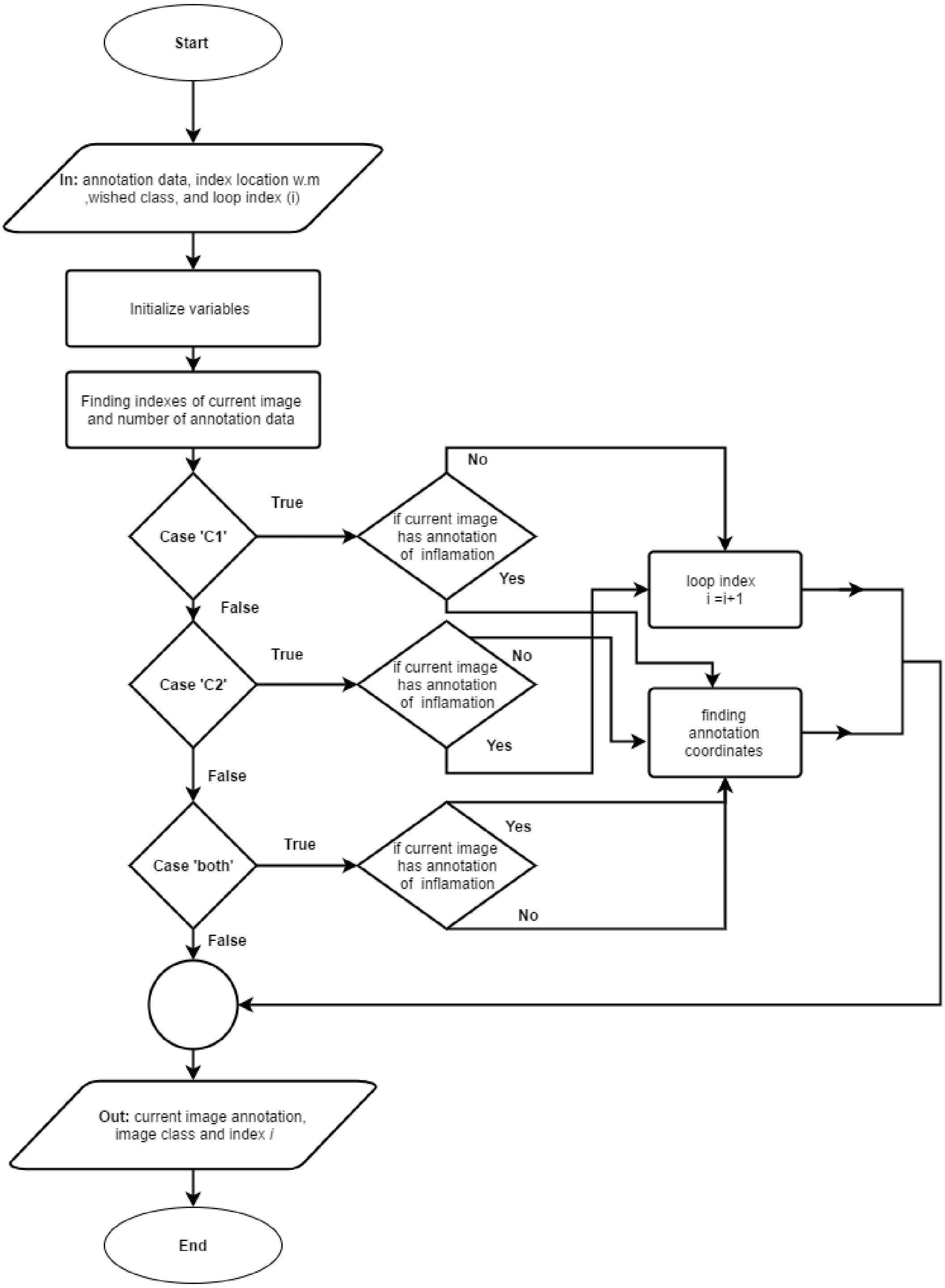

Start
In: annotation data, index location w.m ,wished class, and loop index (i)
Initialize variables
Finding indexes of current image and number of annotation data
Case 'C1'
True
False
if current image has annotation of inflamation
No
Yes
Case 'C2'
True
False
if current image has annotation of inflamation
No
Yes
Case 'both'
True
False
if current image has annotation of inflamation
Yes
No
loop index i =i+1
finding annotation coordinates
Out: current image annotation, image class and index i
End

Figure 43: Algorithm for extracting annotation data

6.2 Implementation of inflammation segmentation

After doing the experiments regarding bone and skin detection, the images and annotation data were saved in a table with the segmented bone and skin features. Thus, it was not required to use the algorithm for reading and extracting annotation and image data described Ch.6.1. In addition, since the yellow box was not a problem in the PD images for the segmentation of the inflammation region, as mentioned in Ch.4.2.1, CMYK conversion was not required either. In the following sub section, some of the MATLAB functions used for inflammation detection will be explained.

Table 4: Overview of the most essential functions used for detection of the inflammation region.
When two places are checked, it means that some alterations were performed on the original function(s)

Methods		**Internal**	**External**	**Novel**
Grayscale conversion (*rgb2gray*)		X		
Masking (*roipoly* and *roifilt2*)				X
Region growing (*graydiffweight* and *imsegfmm*)		X		X
Active contour (*activecontour*)		X		X
Superpixel (*superpixels*)		X		X
Making ellipse			X	
Eucledian distance				X
Similarity measurement Jaccard and Dice				X
PM filter			X	X
Homomorphic filtering and frequency domain filters			X	X
Morphological operators (*imopen, imclose, imtophat, bwmorph*)		X		
Labeling (*bwlabel*)		X		
Boundary description (*regionprops, bwboundaries, polyval and polyfit*)		X		

6.2.1 Masking

To only execute the function on the region of interest(ROI) after masking, it was necessary to implement each of the methods in an anonymous function. “These type of functions is not stored in a program file such as normal MATLAB functions but is associated with a variable whose data type is a function handle. An anonymous function can accept inputs and return outputs, just as standard functions do. However, they can contain only a single executable statement” [60]. All methods from the filtering to segmentation were implemented using anonymous functions. Following is a general example of how the anonymous function was used.

```
%Creating the Mask using XCoord,YCoord for the bundary defined by the bones and skin features
MASK = roipoly(ImageOriginalGray,XCoord,Ycoord);
% Anonymous function that uses the external homomorphic filter fuction is defined
HomomoficFilterFunction=@(x) homofilt(x,cutoffrequens,Iterations,FilterOrder,Filtertype);
% Using roifilt2(image,MASK,func) to only do the filtering on ROI.
HomoFilterdImage=roifilt2(I,MASK HomomoficFilterFunction)
```

The external homomorphic function is named *homofilt* where the input image(I) is the variable (x) in the anonymous function. Hence, the anonymous functions can be used on the ROI by using the internal MATLAB function *roifilt2*.

6.2.2 Region growing

The region growing method was implemented using a combination of *graydiffweight* and *imsegfmm,* which are internal MATLAB functions. The former was used to calculate the pixel weight of the ROI using the seed pixel as the average. This weight was used as an input in the region growing function *imsegfmm.* The region growing method uses a fast-marching method based on the differences in grayscale intensities compared with the seed pixel [61]. Following is an example of how it was implemented.

```
% Making anonymous function and compute the pixel weight in ROI. x1= FilteredImage
% The GrayDifferenceCutof threshold for the intensity difference
GrayWeightFunc = @(x1) graydiffweight(x1,SeedX,SeedY,'GrayDifferenceCutoff',25/255);
GrayWeight=roifilt2(FilteredImage,MASK,GrayWeightFunc);
% Making anonymous function and compute the region growing. x2= GrayWeight
ImsegFMMFunc = @(x2)imsegfmm(x2, SeedX,SeedY, IntDist);
BWRegionGrow=roifilt2(GrayWeight,MASK,ImsegFMMFunc);
```

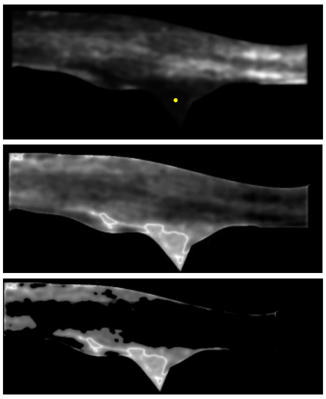


Figure 44:Illustration of the effect of doing *graydiffweight*. The brightest area (the highest weighted) is where the difference to the seed pixel is the smallest. It shows first the filtered input image, followed by a logarithmic representation of the image where the *GrayDifferenceCutoff* is high and low respectively.

6.2.3 Superpixel

Superpixel was implemented using a SLICE based method as described in Ch.3.3.2, which only needed one input. The image processing toolbox provided by MATLAB included a function based on this method. Using *superpixels (I, number of superpixels),* the image could be divided into the number of section that was defined by the number of superpixel pixels, see Figure 40 (b). Based on this, the median of each superpixel was calculated and set as the superpixel value followed by the threshold-based segmentation.

6.2.4 Active Contour

Two different active contours were tested: one edge-based and one based on the Chan-Vese method, both of which MATLAB provided using the function *activecontour (I,MASK,n,'method ')* , where *n* is the number of iteration before the evolution of the contour stops. In addition, whether the contour should evolve in or outwards, could be regulated by adding an initial input.

6.2.5 Similarity measurement Dice

Since the Matlab did not provide functions for calculating Dice coefficients, this was developed using a novel function based on Eq.(3.4.1) and logical operators. Algorithm 6 show a pseudocode of how this was implemented in MATLAB. As it shows, it was not necessary to use an anonymous function for similarity measurement, due to that it was done by comparing two sets of binary data.

Algorithm 6: Dice

```
Input:  BWInflammationOriginal, BWInflammationSegmented
    IntersectionAB = BWInflammationOriginal & BWInflammationSegmented
    UnionAB= BWInflammationOriginal | BWInflammationSegmented
    A=Sum(BWInflammationOriginal=1)
    B=Sum(BWInflammationSegmented =1)
    Dice= (IntersectionAB*2)/(A+B)  *100%
Output: Dice
```

7 Experiment

In this chapter, the experiments complementing Ch.4 are found in Ch.7.1 and Ch.7.2, whereas the experiments complementing Ch.5 are found in Ch.7.3 and Ch.7.4. Since the end goal was to locate the inflammation region, only the data, D2, containing inflammation annotations were kept. Additional visual results can be found in Appendix-B.

When conducting the experiments, it was discovered that in multiple of the images the algorithms did not detect any bone or skin features. In most of the cases, it was due to the images being distorted, destroyed by artifacts, or having no distinct features. In addition, some of the images had no complementing bone, skin or inflammation annotations. Figure 45Figure 44 show two examples of the former problem. To avoid doing similarity measurement on these images, they were located and removed. This was first done visually, then automatically resulting in the dataset D3.

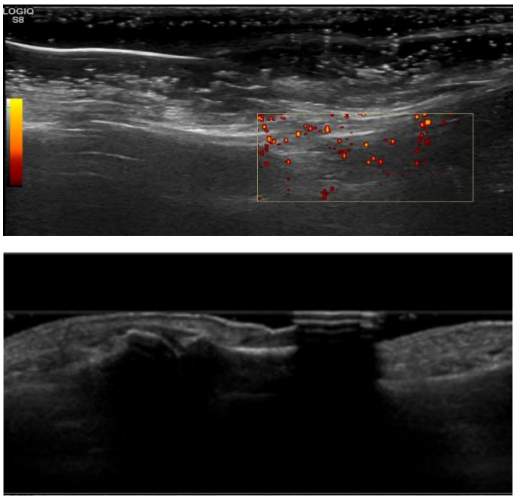


Figure 45: Show two example of images that lead to errors.

In Experiment 4.1 and 4.2, the dataset D2 was separated into T1, T2, and T3. T1 and T2 were used to conduct parameter tuning for bone and skin segmentation. Hence, T3 were the remaining images used to test the methods fully. When conducting the experiments for inflammation segmentation, only the data where the MHD for the bone and skin features were less than 20px were kept. Thus, the performance of the inflammation detection methods could be measured without the results from the previous experiments influencing it. Table 5 shows an overview of the data that were used in the experiments.

Table 5: Explanation of the data used for the experiments

DATA	EXPLANATION
D1	All 1935 images
D2	The 1714 images with Inflammation annotations
D3	The 1628 images in D2 without errors which were used in Experiment 4 and 5
D4	The 605 images from T3 where all annotation was found adequately, which was used for similarity measurement in Experiment 5
T1	The 5 images in D2 used for visual parameter tuning
T2	The 50 images in D2 used for calculating similarity when tuning
T3	D3-T2-T1 = 1573 images used for testing the tuned algorithms in experiment 4
T4D4	T1 +T2=55 where all annotation was found adequately (31/55), used for parameter tuning for before testing on D4 which was used in Experiment 5

7.1 Experiment 4.1 Choice of preprocessing method

This experiment was done to see which histogram equalization method was the best in conjunction with the three filtering methods described in Ch.4.3. In addition, a parameter optimization for both filtering and equalization methods was conducted before testing the different combinations on T2. The parameter optimization in this experiment was done visually on T1, where the distinctiveness of bone and skin was the main deciding factor. Table 6-Table 8 shows which parameters that were adjusted. Similarity measurement was done by calculating the modified Hausdorff distance and the detected features. The best overall similarity was decided mainly based on the relationship between the mean, median and the maximum MHD. In all the tables, Bone1 and Bone2 correspond to left and right bone respectively.

Table 6: Parameter test Perona-Malik filter

Parameter	Range	Step
Input image type	[Grayscale, HERefImage, CLAHE w. Rayleigh or Uniform distribution]	-
Clipfactor CLAHE	0:1	0.01
Distribution factor CLAHE Rayleigh	0:1	0.1
Filter iterations	10:100	5
Integration constant	0:(1/4)	1 (Denominator)
Kappa	30:200	10
Diffusion coefficient	1:2	1
Top-hat SE(disk)	2:30	5
Threshold	30:120	10
Opening	1000:6000	300

Table 7: Parameter test Gaussian smoothing-filter

Parameter	Range	Step
Input image type	[Grayscale, HERefImage, CLAHE w. Rayleigh or Uniform distribution]	-
Clip factor CLAHE	0:1	0.01
Distribution factor CLAHE Rayleigh	0:1	0.1
Filter iterations	5:50	5
Kernel Sigma(std)	0.1:5	0.1
Top-hat SE(disk)	2:30	5
Threshold	30:120	10
Opening	1000:6000	300

Table 8: Parameter test Homomorphic filter. The 4 filetypes that were tested were Butterworth high-boost and high-pass filter as well as Gaussian high-boost and high-pass filter

Parameter	Range	Step
Input image type	[Grayscale, HERefImage, CLAHE w. Rayleigh or Uniform distribution]	-
Clip factor CLAHE	0:1	0.01
Distribution factor CLAHE Rayleigh	0:1	0.1
Filter type	1:4	1
Filter iterations	0:1000	10
Cutoff frequency	0:255	10
Top-hat SE(disk)	2:30	5
Threshold	30:120	10
Opening	1000:6000	300

7.1.1 Results Experiment 4.1

Perona-Malik filter

Table 9: Resulting parameters used in Experiment 4.1 for Perona-Malik filter

Parameter	Optimal			
	Grayscale	HistEqRefImage,	CLAHE w.Rayleigh	CLAHE w.Uniform
Clipfactor CLAHE	-	-	0.01	0.01
Distribution factor CLAHE Rayleigh	-	-	0.4	-
Croppmethode	1	1	1	1
Filter iterations	70	70	150	150
Integration constant	(1/4)	(1/4)	(1/4)	(1/4)
Kappa	90	90	90	90
Diffusion coefficient	1	1	1	1
Tophat SE(disk)	15	15	15	15
Threshold (bone/skin)	60/50	60/50	60/50	60/50
Opening (bone/skin)	2000/3500	200/3500	2000/3500	2000/3500

Table 10: Results Experiment 4.1 Perona-Malik filter on T2. The distance is in pixels, where MHD= 0 is perfect. The best performing method is highlighted in dark gray color.

	ONLY GRAYSCALE			HE EQ BASED ON REFIMAGE			CLAHE-R			CLAHE-U		
	Bone1	*Bone2*	*Skin*	*Bone1*	*Bone2*	*Skin*	*Bone1*	*Bone2*	*Skin*	*Bone1*	*Bone2*	*Bon2*
MHD												
STDEV	55,39	94,23	62,46	55,43	108,17	66,78	57,86	112,32	53,49	58,44	103,76	51,66
MEAN	15,92	49,46	63,99	16,08	57,53	58,12	19,62	50,52	56,66	18,30	48,43	45,60
MEDIAN	5,72	4,50	28,27	5,82	4,71	12,12	6,27	6,65	29,46	6,29	6,84	23,94
MAX	397,12	385,25	197,05	397,32	476,34	218,03	399,91	482,89	185,70	411,44	395,47	188,37
MIN	2,50	1,17	2,26	2,67	1,27	2,14	2,37	1,90	3,48	2,11	2,40	3,13

Figure 46 and Figure 47 show examples of how some of the visual adjustments of the parameters were conducted, including an image with the parameter from Table 9. The best overall similarity based on the MHD depicted in Table 10, was done by first comparing the mean and median with the maximum distance. As described in Ch.7.1, only using the mean as the deciding factor for the similarity was not accurate, which the difference in maximum and minimum distance shows. The best overall similarity for the bones was found using the grayscale input image, whilst for the skin was achived using HE based on a ref image.

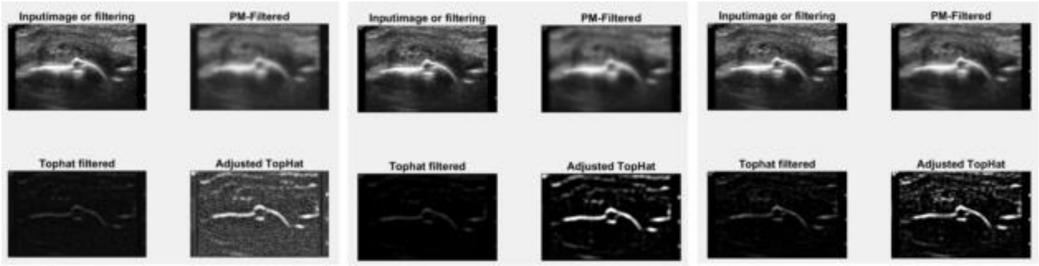


Figure 46: Shows the effects of different integration factors.: first 1/3(over max) followed 1/4(max) at last 1/20(small).

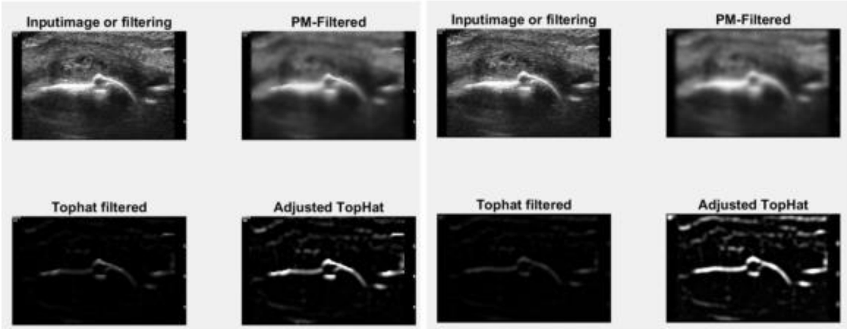


Figure 47: Examples of the effects of the Kappa constant, first kappa= 30 followed kappa= 90.

Gaussian smoothing filter

Table 11: Resulting parameters used in Experiment 4.1 for Gaussian smooth-filter

Parameter	Optimal			
	Grayscale	HistEqRefImage,	CLAHE w.Rayleigh	CLAHE w.Uniform
Clip factor CLAHE	-	-	0.01	0.01
Distribution factor CLAHE Rayleigh	-	-	0.4	-
Crop method	2	2	2	2
Filter iterations	35	35	20	30
Kernel Sigma(std)	2	2	2	2
Tophat SE(disk)	15	10	8	8
Threshold (bone/skin)	60/50	70/60	70/50	70/50
Opening (bone/skin)	1500/3500	1500/3500	1500/3500	1200/3500

Table 12: Results Experiment 4.1 Gaussian smoothing filter on T2. The distance is in pixels, where MHD= 0 is perfect. The best performing method is highlighted in dark gray color.

	ONLY GRAYSCALE			**HE BASED ON REFIMAGE**			**CLAHE-R**			***CLAHE-U***		
	Bone1	***Bone2***	***Skin***	***Bone1***	***Bone2***	***Skin***	***Bone1***	***Bone2***	***Skin***	***Bone2***	***Bon2***	***Skin***
MHD												
STDEV	66,24	70,39	67,24	68,17	121,01	67,79	74,29	94,49	64,00	90,20	133,50	60,71
MEAN	24,33	42,07	67,39	27,51	70,79	67,77	33,49	61,44	58,46	51,72	79,80	50,91
MEDIAN	6,97	6,25	32,78	7,07	6,43	32,63	9,49	8,09	24,98	11,00	8,88	8,31
MAX	390,51	227,35	227,09	369,44	479,02	216,16	393,13	330,78	201,29	392,24	495,85	203,76
MIN	1,98	1,33	1,85	2,11	1,41	1,66	1,57	2,32	2,99	1,88	1,65	1,52

The result using Gaussian smoothing show that the best overall smallest MHD for the bones were obtained by only using the grayscale input image. The best result regarding skin was achieved by using CHLAHE with a uniform distribution. As shown in Table 11, the input image that was used was changed using a slightly more cropped image, because this method wrongly found a skin in the top of the images.

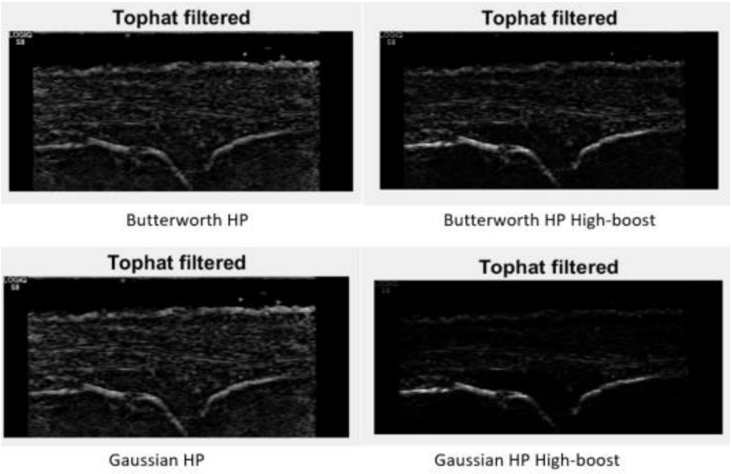
Homomorphic filter

Table 13: Resulting parameters used in Experiment 4.1 for the Homomorphic filter. Filter type 3 and 2 correspond to Butterworth high-boost and Gaussian high-pass filter.

Parameter	Optimal			
	Grayscale	HistEqRefImage,	CLAHE w.Rayleigh	CLAHE w.Uniform
Clip factor CLAHE	-	-	0.01	0.01
Distribution factor CLAHE Rayleigh	-	-	0.4	-
Crop method	2	2	2	2
Filter type	3 and 2	3 and 2	3 and 2	3 and 2
Filter iterations (3/2)	1/70	1/80	1/60	1/80
Cutoff frequency	255	255	255	255
Tophat SE(disk)	15	20	15	12
Threshold (bone/skin)	100/90	80/90	110/100	110/100
Opening(bone/skin)	1500/3500	1500/3500	1500/3500	1500/3500

Table 14: Results Experiment 4.1 Homomorphic filter on T2. The distance is in pixels where MHD= 0 is perfect. The best performing method is highlighted in dark gray color.

	ONLY GRAYSCALE			HE BASED ON REFIMAGE			CLAHE-R			*CLAHE-U*		
	Bone1	***Bone2***	***Skin***	***Bone1***	***Bone2***	***Skin***	***Bone1***	***Bone2***	***Skin***	***Bone2***	***Bon2***	***Skin***
MHD												
STDEV	96,64	117,41	145,84	92,23	112,02	122,50	114,84	144,96	42,81	100,42	119,15	145,55
MEAN	211,91	241,69	263,23	210,84	260,46	302,80	115,42	198,95	28,20	186,87	225,15	99,44
MEDIAN	242,22	254,87	358,86	239,19	244,09	358,40	60,23	261,60	7,94	233,32	266,32	10,38
MAX	579,72	480,74	366,17	590,03	536,89	368,38	418,55	452,15	187,44	418,67	437,71	367,49
MIN	3,42	2,92	3,10	3,51	2,07	3,25	2,75	2,21	2,17	2,55	5,79	1,90



Butterworth HP

Butterworth HP High-boost

Gaussian HP

Gaussian HP High-boost

Figure 48: Shows examples of the different filtering methods with 1 iteration and top-hat-filtering.

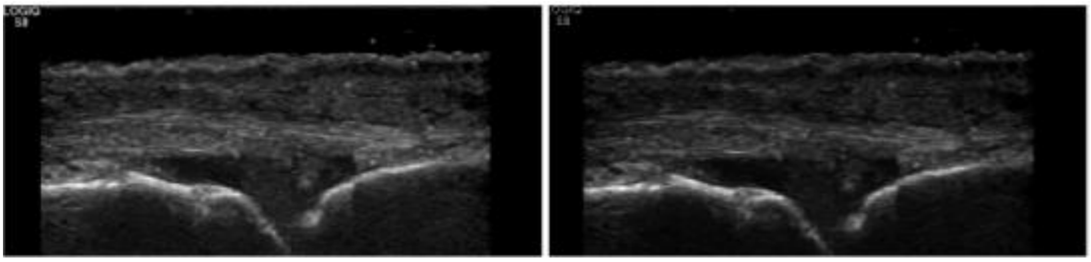


Figure 49: Shows how that the high-boost Butterworth filter(right) can be used later for light-filtering to increase the distinctiveness of the inflammation region. To the left is the unfiltered grayscale image.

Before optimization and testing of the preprocessing methods, the effects of the 4-different frequency-domain filters were tested. Figure 48 shows the various filters with 1 iteration after using Top-hat filter. It was decided to use 1 iteration of Butterworth high-boost filter combined with multiple iterations of Gaussian high-pass filter for blurring forward. The result from Table 14 shows that homomorphic filtering gave an overall weak result. However, skin segmentation using CLAHE with Rayleigh distribution yielded good results.

7.1.2 Discussion

The best overall similarity was based on the MHD by comparing the mean and median with the maximum distance. Only using the mean as the deciding factor for the similarity was not accurate, which the difference in maximum and minimum distance showed. The best results for bone detection were obtained by only using the grayscale image for both Perona-Malik and a Gaussian smoothing filter. Both had a median MHD well beneath 10px. Homomorphic filtering resulted in the overall worst similarity as well as speed performance which is shown in Table 16. However, combining homomorphic filter and CLAHE with Rayleigh distribution gave the best results regarding the skin. Since the overall result and performance using homomorphic filter was low, it was decided to discard this filter for bone and skin segmentation forward. Figure 49 show that homomorphic filtering could be used later for light filtering before localizing the inflammation region. The result summary for this experiment can be found in Table 15.

Table 15: Result summary Experiment 4.1

METHODE	BONE1	BONE2	SKIN
PERONA-MALIK	Grayscale	Grayscale	HE based on a ref image
GAUSSIAN SMOOTH	Grayscale	Grayscale	CLAHE-U
HOMOMORPHIC	CLAHE-R	CLAHE-R	CLAHE-R

Table 16: Speed performance for detection of skin and bones on T2

Method	Time in min on the 50 images in T2
Homomorphic 80 iterations	8.87
Gaussian 35 iterations	1.05
PM filter 70 iterations	2.3

7.2 Experiment 4.2 Segmentation of skin and bones

After doing Experiment 4.1, it was decided to continue with Gaussian smoothing filter and PM-filter both using a grayscale input image. This experiment was focused on parameter optimization of the filtering and segmenting methods, as well as finding the best overall method. To do this, the impact of the parameter adjustment was first visually inspected on T1, followed by analyzation of the results derived by calculating the MHD for the images in T2. From the information gathered in Experiment 4.1 it was decided to add one more measurement to the result table, percentage of MHD under 20px, to give an indication of the overall accuracy. An example where segmented Bone1 yielded a MHD of 20px is shown in Figure 30. After finding the optimally tuned parameter for each filtering method, they were tested on all the images in T3.

7.2.1 Parameter tuning Experiment 4.2

Table 17: Parameters for tuning on T2 using PM-filter

Parameter	**Optimal**			
	Grayscale(Ref)	Grayscale 1	Grayscale2	Grayscale3 (excluding yellow PD-box)
Crop method	1	1	1	1
Filter iterations	70	70	70	70
Integration constant	(1/4)	(1/4)	(1/4)	(1/4)
Kappa	90	90	100	60
Diffusion coefficient	1	1	2	2
Top-hat SE(disk)	15	10	9	9
Threshold (bone/skin)	60/50	40/50	40/40	40/40
Opening (bone/skin)	2000/3500	1200/2000	1200/2000	1200/2000
Dilate (bone/skin)	-	1/8	1/8	1/6

Table 18: Parameters for tuning on T2 using Gaussian smoothing-filter, all excluding yellow PD-box

Parameter	**Optimal**			
	Grayscale(Ref)	Grayscale 1	Grayscale 2	Grayscale 3
Crop method	2	1	1	1
Filter iterations	35	35	35	30
Kernel Sigma(std)	2	2	1.8	1.8
Top-hat SE(disk)	15	8	12	20
Threshold (bone/skin)	60/50	40/40	40/40	40/40
Opening (bone/skin)	1500/3500	1200/2000	1200/2000	1200/2000
Dilate (bone/skin)	-	1/8	1/6	1/6

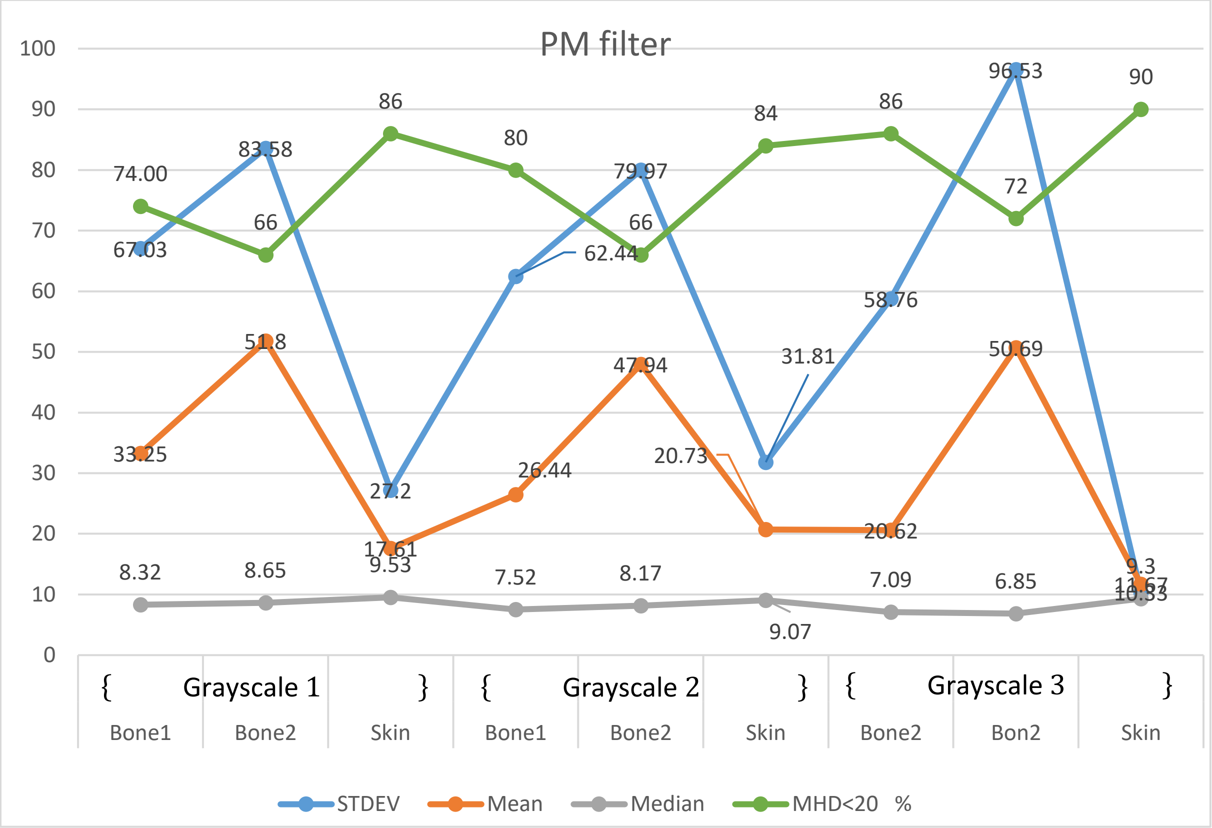


Figure 50: Results using PM-filter on T2, with the different parameter in Table 17

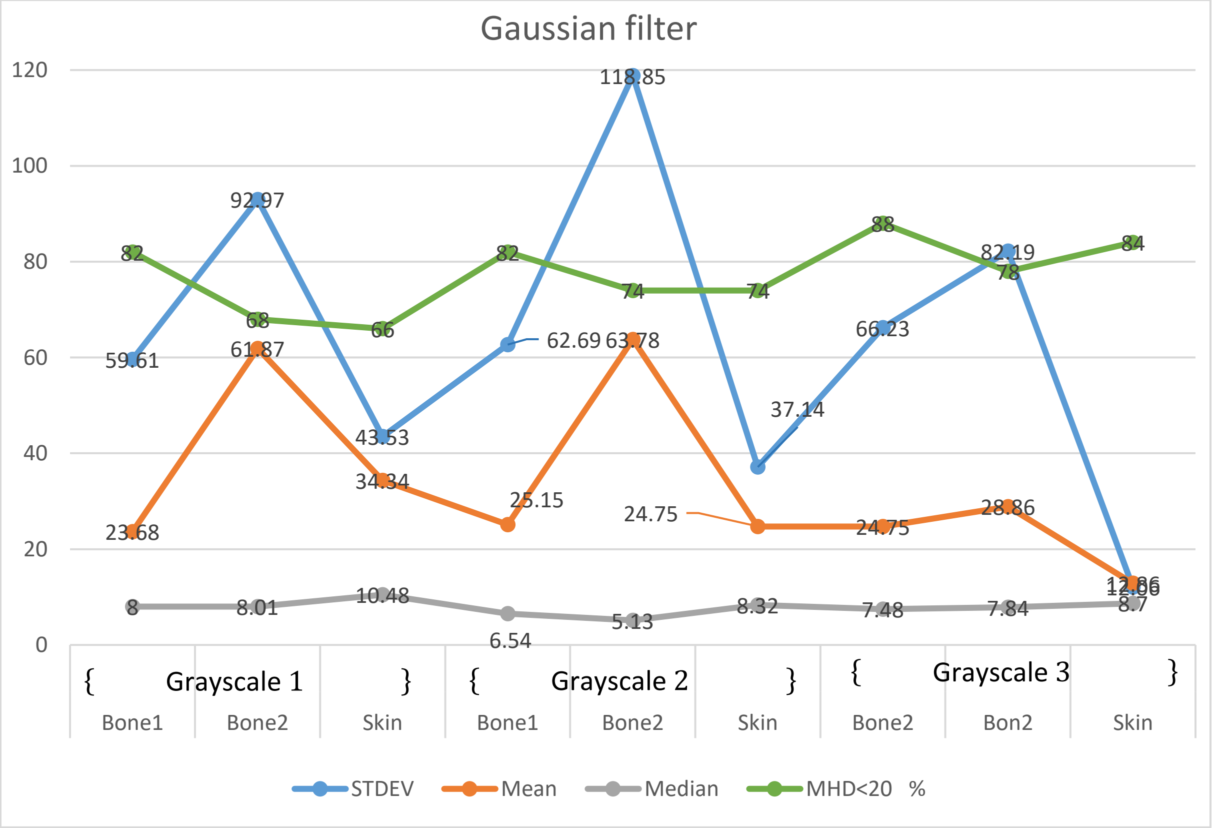


Figure 51: Results using Gaussian smoothing filter on T2, with the different parameter in Table 18.

Table 19: Best results from Experiment 4.2 using parameters named Grayscale 3 in Table 17 and Table 18, on T2

	BEST PM-FILTER			BEST GAUSSIAN		
	Bone1	*Bone2*	*Skin*	*Bone2*	*Bon2*	*Skin*
MHD						
STDEV	58,76	96,53	10,33	66,23	82,19	12,06
MEAN	20,62	50,69	11,67	24,75	28,86	12,86
MEDIAN	7,09	6,85	9,30	7,48	7,84	8,70
MAX	389,00	375,72	59,09	347,45	482,52	60,63
MIN	1,99	1,69	2,77	2,20	1,93	2,60
MHD<20	**86%**	**72%**	**90%**	**88%**	**78%**	**84%**

For both filtering methods, the three best results from the parameter optimization were noted which is shown in Figure 50 and Figure 51. Table 19 shows an overview of the results using the best parameters on the 50 images in T2. The result shows that the skin segmentation was significantly increased from the last experiment, this due to two changes in the segmenting method for skin detection. The first change that was done was slightly thickening the objects, after opening and thresholding, by using morphological dilation. The second change was to the skin localization algorithm, by limiting the possible skin centroid location to be over the middle of the image in the y-direction. From the best skin segmentation in Experiment 4.1, the mean MHD was reduced from 64ps to 12px, and 67px to 13px, for PM-filter and Gaussian filter respectively.

Segmentation of Bone1 seems to be relatively consistent regarding mean and median MHD for both methods. However, there was a slight change in MHD<20 when using the Gaussian filter. As Figure 51 shows, MHD<20 for Bone1 went from 82% to 88%, and using PM-filter went from 74% to 86%. Bone2 gave the overall worst results, but there was an increase of roughly 10% MHD<20 for both filtering methods, where using Gaussian gave the biggest decrease in mean MHD. Moving forward it was decided to use the last method, "Grayscale 3", to test on all images to see which filtering method that was the best.

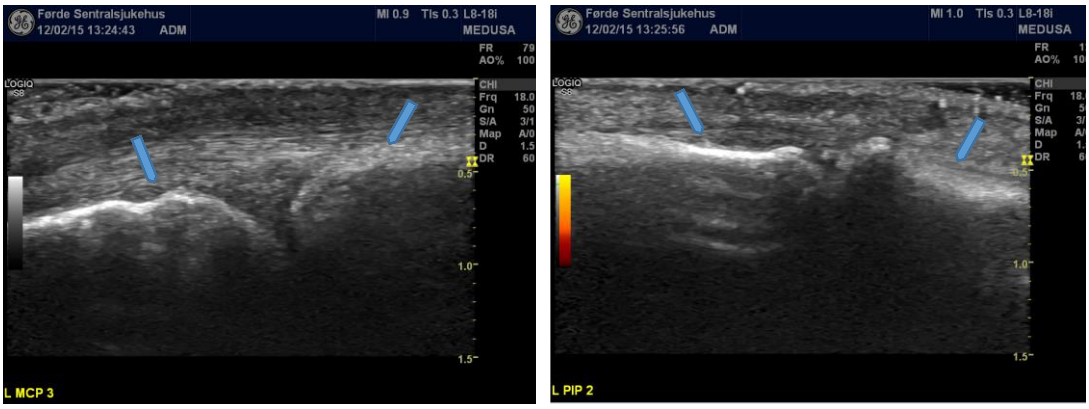


Figure 52: Shows the issue with the distinctiveness of the right bone(Bone2), compared to the left bone (Bone1).

7.2.2 Result Experiment 4.2

Table 20: Results from T3 for Experiment 4.2, where the distance is in pixels and MHD= 0 is perfect.

	PM-FILTER GRAYSCALE 3			**GAUSSIAN GRAYSCALE 3**		
	Bone1	***Bone2***	***Skin***	***Bone1***	***Bone2***	***Skin***
MHD						
STDEV	52,34	128,01	31,01	55,36	112,79	31,97
MEAN	22,42	87,68	20,00	25,23	56,61	31,01
MEDIAN	5,95	11,30	6,28	9,14	10,99	14,13
MAX	495,82	849,36	152,49	472,53	835,19	132,27
MIN	1,12	0,76	1,38	2,11	1,38	1,36
MHD<20	**84%**	**55%**	**79%**	**84%**	**64%**	**54%**

Running the best tuned parameters, which were called Grayscale 3 in Table 17 and Table 18, on all the images in T3, the most consistent feature was Bone1. With MHD<20 equal 84% for both methods, it gave just a decrease of 2% and 4% compared to the results in Table 19 using Gaussian and PM respectively. The mean MHD was slightly better using PM filter, but the median was almost twice as low compared to the MHD using a Gaussian filter.

For Bone2, the MHD<20 decreased 14% and 17% using Gaussian and PM respectively compared to running it on the 50 images in T2. Using Gaussian gave almost 10% higher MHD<20 and 30px lower mean than PM-filter did. However, the median of the MHD was almost the same. Regarding the skin detection, mean MHD was 10px lower and median was 8px lower using PM compared to using Gaussian. MHD<20 was also 25% higher using PM.

In addition to looking at the MHD, the speed performance of each method was calculated. The fastest method was Gaussian which took 31 minutes, whereas PM took 52 minutes. However, the Gaussian filter only had 30 iterations compared to PM-filter which had 70 iterations as shown in Table 17 and Table 18. Summarized, PM-filter gave the overall best results. Therefore, it was decided to use PM-filter as the filtering method for bone and skin segmentation continuing forward. Figure 53 shows examples of both good and bad segmentation of the skin and bones features using PM filter.

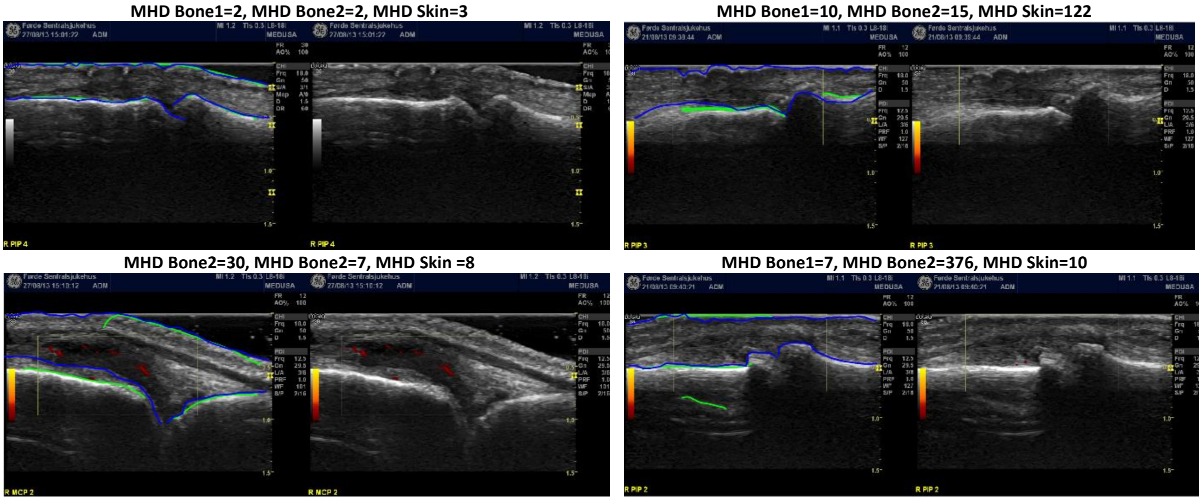

MHD Bone1=2, MHD Bone2=2, MHD Skin=3
MHD Bone1=10, MHD Bone2=15, MHD Skin=122
MHD Bone2=30, MHD Bone2=7, MHD Skin =8
MHD Bone1=7, MHD Bone2=376, MHD Skin=10

Figure 53: Shows the annotations in blue and segmented bone and skin features in green.
From the top left: good segmentation of all features followed by an image where skin is erroneously detected as the left bone. From bottom left: MHD just above 20 for the left bone (Bone1) followed by an image where the right bone (Bone2) is detected beneath Bone1. It also shows that the annotation can go further in both x-direction than the segmented features and still have high similarity due to the restriction described in Ch. 4.5.2.

7.3 Experiment 5.1 Parameter tuning for segmenting inflammation region

This experiment was conducted to test problems and weaknesses in the algorithms developed for segmentation of the inflammation region. This was done by running each method on the 31 images in T4D4 and visually look for errors in the images with low Dice coefficient, and then deriving the best parameters for each method. For the region growing method, some parameters were locked since the main goal was to find the best method for locating the seed point.

Table 21: Parameter used for the choice of seed where all other parameters were locked.
C means centroid, WC is weighted centroid, LP lowest point and EM elliptic mask.

Parameter	Optimal
Input image	Filtered Grayscale
Intensity distance (C/WC/LP/EM)	0.015/0.01/0.025/0.025

Table 22: Results from testing of seed point for region growing on T4D5

METODE	**SEED POINT METHOD**			
	Centroid	***Weighted centroid***	***Lowest point***	***Elliptic Mask***
DICE				
STDEV	27,96	28,15	25,32	19,35
MEAN	38,93	38,45	35,53	49,37
MEDIAN	42,53	39,92	33,75	54,86
MAX	85,96	87,06	86,03	82,37
MIN	0,00	0,00	0,00	11,41

Table 23: Parameter range Active contour.

Parameter	Range	Step
Initial contour	[Threshold object, Ellipse]	
Type	[Chan-Vese, Edge]	-
Contrast bias	-20:10	1
Smooth Factor	-10:10	1
Iterations	30:300	20

Table 24: Resulting parameters for Experiment 5.1 using Active contour. Since the contrast bias is negative, it is restricted to grow outwards

Parameter	Optimal Chan-Vese	Optimal Edge
Initial contour	Ellipse	Ellipse
Contrast bias	-15	-15
Smooth Factor	0	1
Iterations	50	50

Table 25: Parameter range Superpixel.

Parameter	Range	Step
Number of superpixels	50:500	40
Segmenting Threshold	30:50	1
Erosion	2:20	1
Majority	2:30	1

Table 26: Resulting parameters for Experiment 5.1 using Superpixel.

Parameter	Optimal
Number of superpixels	150
Segmenting Threshold	My adaptive
Erosion	10
Majority	25

Table 27: Best results from each method on T4D4 in experiment 5.1 where 100 is the best.

BEST RESULT FROM TEST ON T4D5

	(AC)Chan-Vese	*(AC) Edge*	*Super-pixel*	*Elliptic Mask*
DICE				
STDEV	17,29	17,03	22,79	19,35
MEAN	53,28	53,48	47,36	49,37
MEDIAN	53,57	54,18	56,61	54,86
MAX	82,39	82,43	80,89	82,37
MIN	13,78	14,07	0,00	11,41

Using the elliptic mask described in Algorithm 4 to locate the seed point turned out to yield the best results, both visually and by looking at the Dice coefficient. However, the region growing compared with the other methods gave overall the second worst results. When inspecting the images visually, it shows that this method can be improved by changing the form of the elliptic mask and add more restriction on the intensity distance.

Both active contour methods gave a decent result, but the edge-based method was slightly better with the parameter in Table 24. The main difference was the speed of each method. The edge-based method took 2.20 minutes, whilst Chan-Vese method only took 1.05 min to run through all 31 images in T4D4. The main problem visually for each method was that the resulting inflammation area did not grow outwards enough when the annotated inflammation area was large. This could be improved by letting the active contour evolve for a longer period, or by using the binary image that Algorithm 1 yielded as the initial contour.

Superpixel gave overall decent result when choosing superpixels between 150 and 500, but as Table 26 shows, the optimal number of pixel was 150. The paramount issue was that in some cases the merging of the superpixel classified as potential inflammation region was too generous, thus resulting in large areas. However, this method gave the best median Dice coefficient. Overall the edge based AC method gave the best result with no zero values, highest mean and max value, as well as the lowest STEDEV.

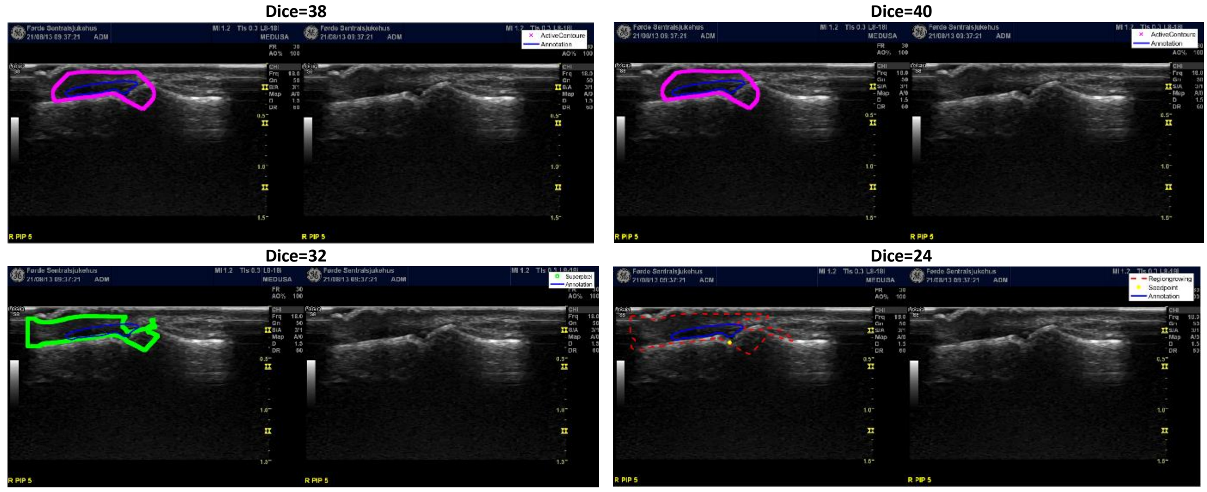


Figure 54: From upper left: shows example of Chan-Vese and Edge based active contour in row one using an ellipse as the initial contour. From left in row two: Superpixel and Region growing. Annotated inflammation region is in blue for all the images. It shows that some adjustment can still be made going into experiment 5, and that the annotation is a little bit conservative, considering that the dark area stretches well outside the blue boundary. However, if it followed the grading atlas in Appendix-A.1., this image should not have any inflammation annotation since it is graded as 0.

7.4 Experiment 5.2: Segmentation of inflammation region

Before running the different segmenting algorithms on all 605 images in D4, it was necessary to do even more parameter adjustments, in an attempt to increase the Dice coefficients. Therefore, a pre-experiment was performed on the best methods from Experiment 5.1 on T4D4. Using the knowledge from Experiment 5.1, a method was also developed that combined the algorithms used previously with the active contour as showed in Algorithm 5. In this subsection, result from the pre-experiment for each of the methods will be presented first, followed by the final result when running through D4.

7.4.1 Result pre-experiment 5.2 Region growing

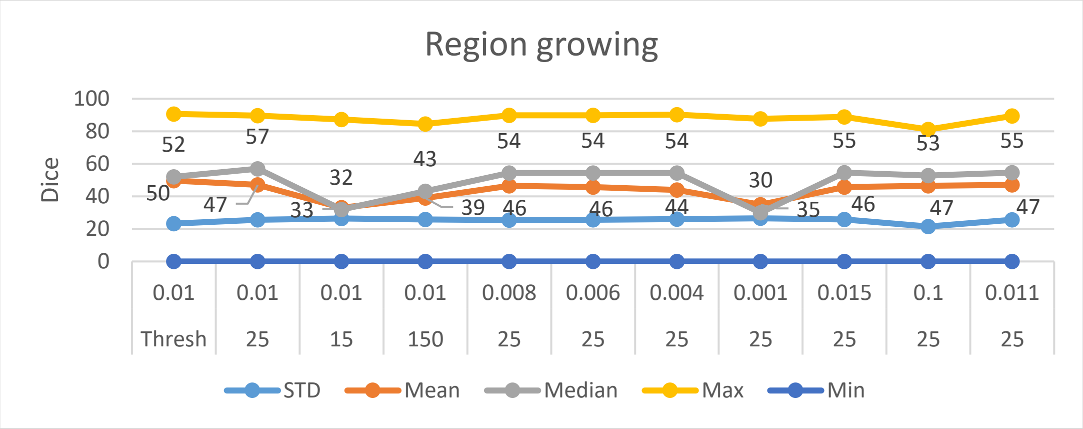


Figure 55: Shows that changing the intensity distance (first row) and graydiffcutoff (second row) does not change the result for T4D4 radically.

Table 28: resulting parameters for region growing in Experiment 5.2, used on T4D4 and D4 where my adaptive is the *Threshold* from Algorithm 1.

Parameter	Optimal
Threshold	0.5<T< My adaptive
Intensity distance	0.01
Graydiff cutoff	25
Ellips displace x/y	-70/-80
Ellips Radius x/y	80/50

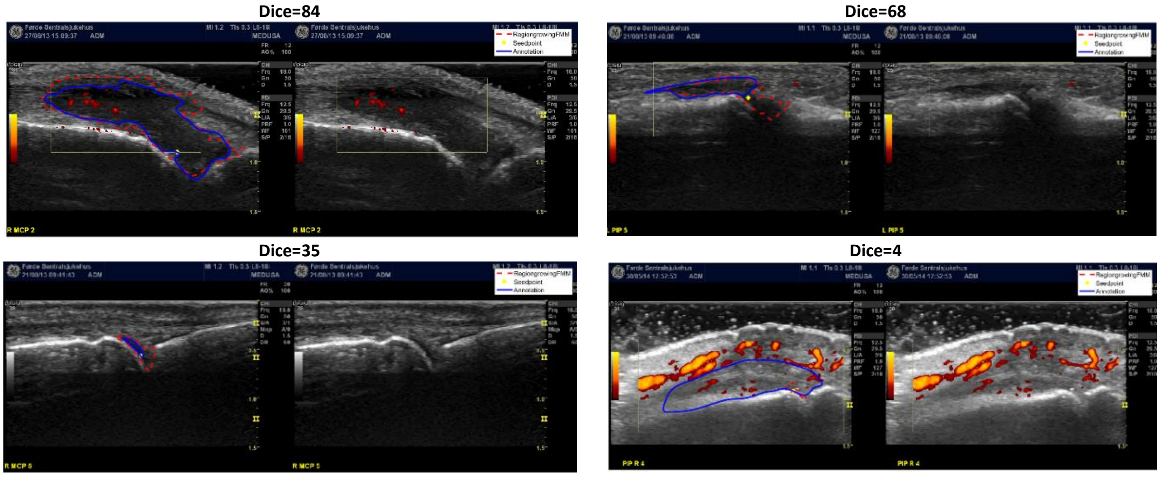


Figure 56: Examples from T4D4 using Region growing. It shows that this method is not that good if the illumination is high, such as in the bottom right image. Also, it shows that the Dice coefficient is low even though it finds the region relatively well, as showed in the bottom left image where it is only 35.

7.4.2 Result pre- experiment 5.2 Superpixel

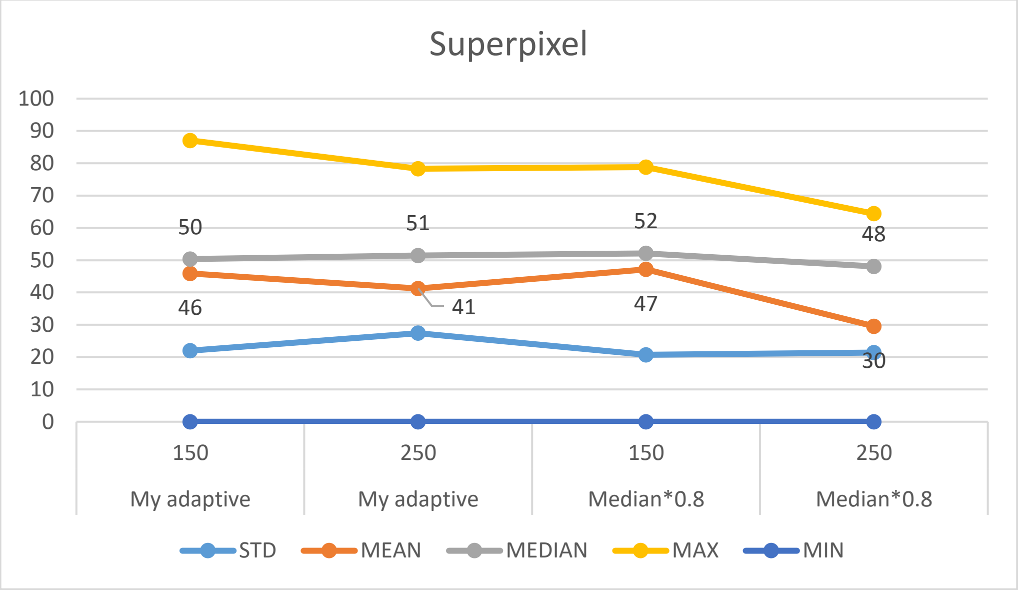


Figure 57: Shows what happened when changing the threshold of the superpixel to include in the segmentation. The number of superpixels are in row one and the threshold in row two.

Table 29: Resulting parameters for Superpixel in Experiment 5.2, used on T4D4 and D4. 80% of median means the median of the resulting superpixel image, see step **(c)** in Figure 40.

Parameter	Optimal new	Optimal Exp.5 see Table 26
Number of superpixels	150	150
Segmenting Threshold	80% of median intensity	My adaptive
Erosion	10	10
Majority	25	25

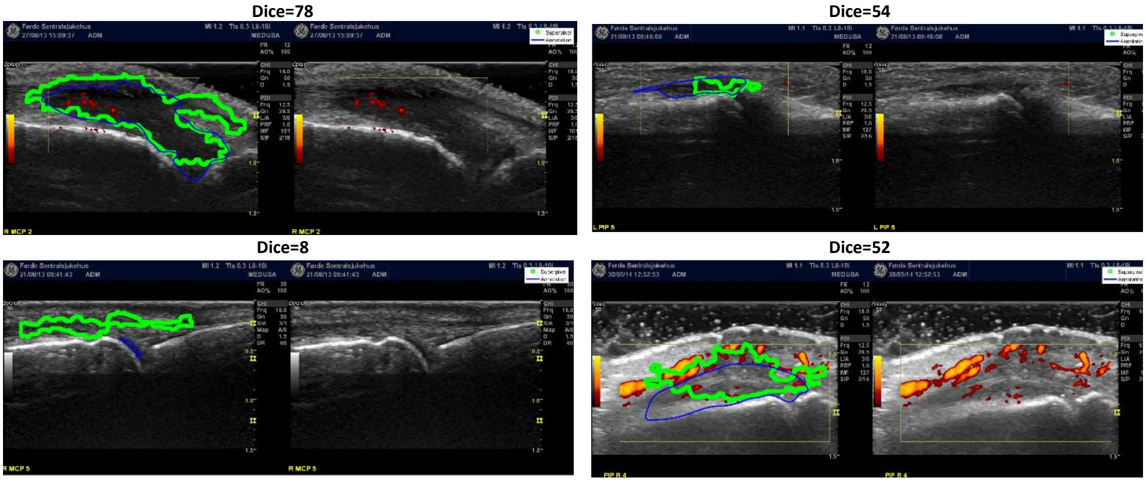


Figure 58 Examples from T4D4 using Superpixel. As seen in the bottom left image, the superpixel method sometimes misses the small inflammation region. This is because the superpixels is a result of the median of a region.

7.4.3 Result pre-experiment 5.2 My active contour

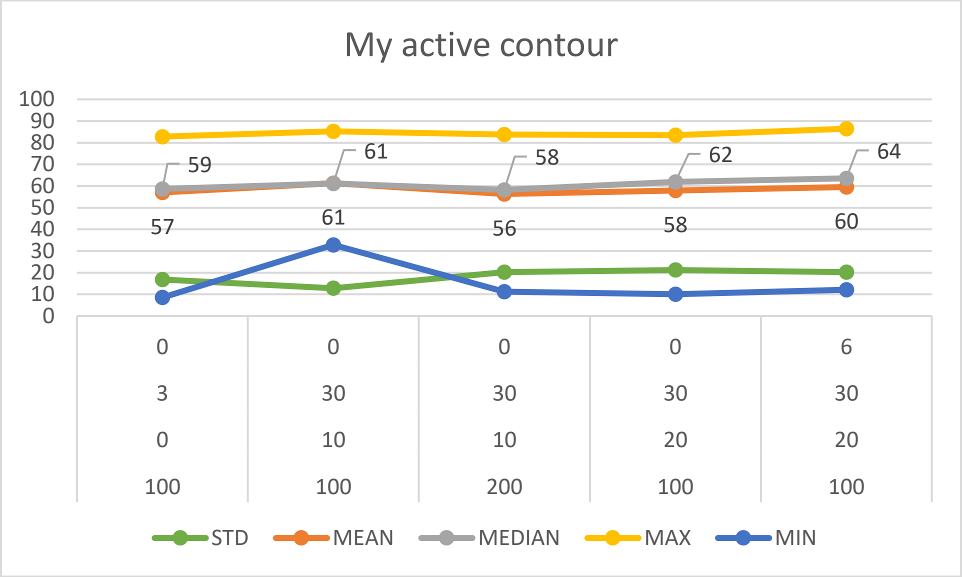
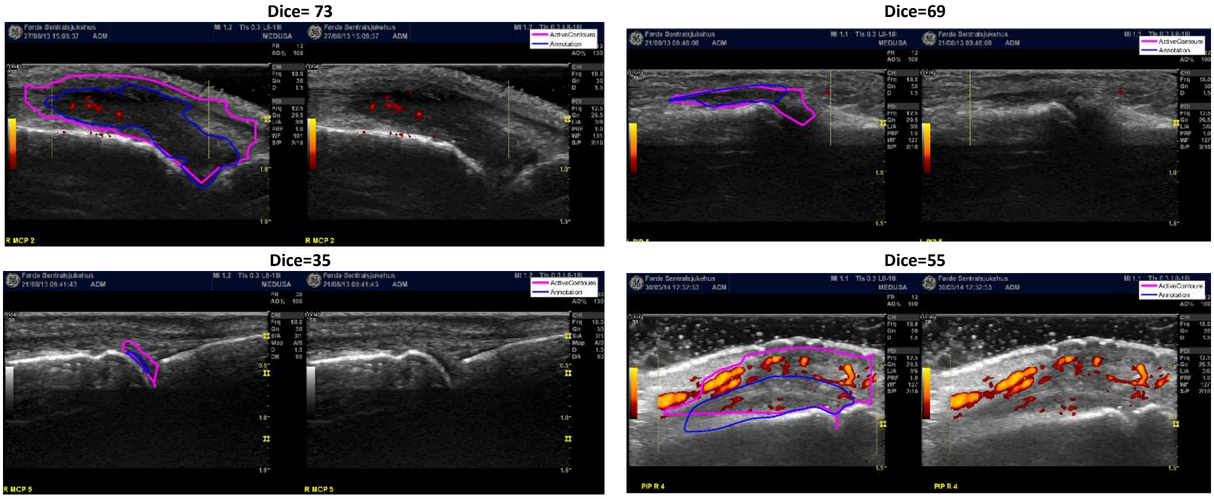


Figure 59: Shows changing the size of the structural element of the morphological operators: thin, erosion and dilation respectively from the top row. The last row shows the change in the evolution time(iterations) of the contour. The name Col2 and Col5 used in the following comes from parameter combination in the respective column of this figure.

Table 30: Resulting parameters for my active contour in Experiment 5.2 used on T4D4 and D4.

Parameter	Optimal Col2	Optimal Col5
Method	Chan-Vese	Chan-Vese
Threshold	My adaptive with 26 as best fit	My adaptive with 26 as best fit
Initial contour	Smallest Dice Threshold object	Smallest Dice Threshold object
Erosion/Dilation/thin	30/10/0	30/20/6
Contrast bias	0	0
Smooth Factor	20	20
Iterations	100	100


Dice= 73

Dice=69

Dice=35

Dice=55

Figure 60: Example from T4D4 using my active contour method. As the images shows, this method finds the inflammation region reasonably well in all the images. In addition, this method results in smoother boundaries. However, since dilation is performed on the segmented object, thin dark “arms” will be concatenated with the “main” part of the segmented object as seen in the upper left image, and thus yield a more generous boundary.

7.4.4 Result 5.2 inflammation detection

Table 31: The best results for each of the segmenting methods when testing on T4D4

BEST RESULT FROM TEST ON T4D5

	(ACMY)Chan-Vese Col5	*(ACMY)Chan-Vese Col2*	*Superpixel new*	*Elliptic Mask*
DICE				
STDEV	20,30	12,82	20,72	25,68
MEAN	59,59	61,34	47,15	47,00
MEDIAN	63,54	61,20	52,09	56,99
MAX	86,46	85,21	78,84	89,65
MIN	12,07	32,86	0,00	0,00

Table 32: Results for each of the segmenting methods when running through all 605 images in D4

BEST RESULT FROM TEST ON D4

	(ACMY)Chan-Vese Col5	*(ACMY)Chan-Vese Col2*	*Superpixel new*	*Superpixel Exp5.1*	*Elliptic Mask*
DICE					
STDEV	22,88	21,27	24,72	21,98	22,26
MEAN	49,49	46,11	48,15	45,90	53,95
MEDIAN	51,36	48,82	53,99	50,31	58,11
MAX	92,90	91,71	86,84	87,06	91,10
MIN	0,00	0,00	0,00	0,00	0,00

Table 33: Time consumption for each of the segmenting methods on D4

Method	**Time**
AC	34.5 min
Region growing	15.6 min
Superpixel	26.08 min

The results using active contour shows that the parameters called Col2 gave the most concise results for T4D4, with highest minimum, lowest STDEV, and almost equal mean and median. Although Col2 gave the best result on T4D4, it gave the lowest results when running through D4. Therefore, the Col5 parameters, which gave the highest median on T4D4, were also tested on D4. This led to an overall higher similarity.

When trying to improve the results using superpixel on T4D4, it did improve visually but not the Dice coefficient. Therefore, the parameters from Experiment 5.1 were also used on D4 as Table 29 shows. The results shows that even though the parameter from experiment 5.1 were best on the test images T4D5, it was not the case on D4. When running both parameter settings on D4, all results except the STDEV were higher using the new parameter settings.

Even though region growing was improved regarding median and mean from Experiment 5.1, it now had zero-minimum values. Also, it had the second lowest similarity on T4D4 as Table 31 shows. However, it resulted in the best overall similarity, as well as being the fastest method, as respectively Table 32 and Table 33 shows.

All the methods were overall improved from Experiment 5.1, except for the superpixel method on T4D4, where only the STDEV was improved, as seen when comparing Table 27 and Table 31.

8 Discussion

The objective of this thesis was to develop a method which could automatically segment the synovitis or inflammation region in US images. The synovitis develops from the joint, which is between the bones, towards the skin. Hence, methods for locating and segmenting the skin and bones were developed to use later as reference points for segmentation of the inflammation region. This thesis was therefore divided into two main parts: segmentation of skin and bone, and segmentation of inflammation region. Multiple image processing techniques were tested for both parts, which will be discussed further in the following subsections.

8.1 Detection of skin bones

In this subsection, the methods and results regarding the segmentation of the skin and bones will be discussed.

8.1.1 Preprocessing

The main issue when working with US images is the presence of multiplicative noise, in the form of speckles. I therefore experimented with multiple enhancements and filtering methods to enhance the bone and skin features as well as removing the presence of noise.

Three enhancements techniques were tested, including HE based on a reference image, CLAHE with Rayleigh and uniform distribution. The thought behind the first method was that using the intensity distribution of an image where the skin and bone features were distinct could enhance these features in other images. CLAHE was tested with Rayleigh distribution, in addition to uniform distribution because the speckle noise is proven to follow a Rayleigh distribution [23] [24]. As the results show in Ch. 7.1, the overall best results came when not using any equalization on the images. However, when using homomorphic filtering for skin segmentation, equalization yielded decent results. Although equalization gave sometimes decent results, this was not significant enough, and was therefore not used onwards. The only enhancement that was done was the removal of the yellow box in the PD images before performing the grayscale conversion as explained in Ch.4.2.1.

When looking for good despeckling filters, it is important that they can remove the speckle noise without losing or altering the image features. Hence, three filtering techniques were tested: a PM filter, a homomorphic filter, and a Gaussian smoothing filter. All the filters were used iteratively, resulting in smoothing of the images. Following the smoothing, morphological THT was utilized to adjust the illumination. After doing Experiment 4.1 in Ch.7.1, it was discovered that the use of homomorphic filter yielded overall bad results, as well as being significantly slower than the two other filters. Moving forward in Experiment 4.2, only the Gaussian and PM filters were tested. The results from this experiment show that the PM-filter gave the best results when looking at MHD. Another reason for the PM-filter yielding the best result is that it visually seems to preserve more of the features than the Gaussian as shown in Figure 28.

8.1.2 Segmenting

A novel method was developed for segmenting the bone and skin features, based on a global threshold of the top-hat transformed image and morphological operators. This was done using the prior knowledge about the position of the skin and skin and bones. As stated earlier, it was decided to move forward with no equalization, even though in some cases it yielded better results regarding skin segmentation than only using grayscale before filtering. The reason for this was that after visual inspection, it was discovered that these results could be improved by adjusting the segmenting algorithms and some of the filtering parameters. The main change to the segmenting algorithm was the dilation of the binary objects after doing global thresholding and restricting possible skin objects to be over the middle of the image.

As the results from running the proposed system for bone and skin segmentation on all the images in T3 show, the skin and the left bone (Bone1) were detected quite well. With a MHD<20 of 84% and 79%, they were significantly better detected than the right bone (Bone2), with MHD<20 of only 55%. One of the main reasons Bone2 was not detected as well as the other features, was that in the images it was often not as distinct as the others, which Figure 52 shows. In addition to this, some improvements to the algorithm for bone segmentation could be made, especially to avoid such errors as shown in the bottom right image of Figure 53. This could be done by separating the segmentation algorithm for bone detection in two. Hence, using Bone1 as a reference point when locating Bone2 could be advantageous since Bone1 was segmented consistently good.

8.1.3 Boundary description and similarity measurement

Since the annotations described the bone and skin features as a curve, it was necessary to change the resulting closed boundary from the segmentation to a curve. This was done by splitting the boundary in two: an upper and a lower curve. The curves were represented as a 7th degree polynomial of the points in the upper and lower curve. Hence, after visual inspection of the annotation data, it was decided to compare the upper skin and bone curves with the annotations.

Similarity measurements were done by calculating the MHD between the segmented bone and skin features, and the annotations. Before doing this, restrictions were made regarding the area of the features and annotations. It was decided to focus on measuring the similarity of the position of the skin and bones, rather than how similar the contour of the detected features was compared to the annotations. Therefore, similarity measurements were restricted to the points which had the same vicinity in the x-direction as explained in Ch. 4.5.2, thus avoiding low similarity due to a curvature of the features or annotations, as well as if either of them were longer in the x-direction. An example of where the resulting MHD was low even though the distance is long between multiple of points, due to the restriction, can be seen in Figure 53. One can argue that this restriction can produce an inadequate representation of the skin and bones, and that it could affect the masking done in the inflammation detection part. As Figure 53 shows, the location of the skin and bones was good even though they were not exactly equally long in each x-direction as the annotations.

To get a better picture of the overall similarity when calculating the MHD for all the images, aditional measurments were computed. Including mean, median, STDEV, maximum and minimum. Since the outliers had a big effect on the mean, the focus was first on looking for small medians and then comparing them to the other computations. After doing Experiment 4.1 , it was decided to add another measurement to get an indication of the accuracy of the skin and bone segmentation. Hence, the percentage of the features with MHD beneath 20px was calculated. The threshold of 20px came after a visual inspection of the images. Examples of MHD equal and above 20px can be seen in Figure 31 and Figure 53 respectively.

8.2 Detection of inflammation

In this subsection, the methods and results regarding the inflammation detection and the materials acquired for this thesis will be discussed.

8.2.1 Masking

Regarding the masking problem mentioned in Ch.8.1.3, this was compensated by defining fixing the maximum and minimum x-coordinates, as shown in Ch. 5.1. By combining the y-coordinates from the endpoints of the detected skin and bones with the fixed coordinates, this problem was avoided.

The masking was performed in order to restrict the area of the potential inflammation to be between the skin and bones. This would help avoiding the detection of dark regions below the bones or above the skin to be confused as potential inflammation regions when doing the segmentation. In addition, masking the image lead to the image processing going faster since it only needed to run on the ROI.

Detection and similarity measurement of the joint was not included in this thesis since the annotations were not of consistent shape or size. Thus, the similarity would be low and not give useful information. However, since the joint is between the bones, it could easily be defined; this was done by displacing the middle point between the bones, as depicted in Eq.(5.1.1). This BJP was used in the masking to include as much as possible of the joint area. Overall the masking algorithm worked well and did what it was intended to do.

8.2.2 Preprocessing

Since the input images for inflammation segmentation was the same as for the segmentation of skin and bone, the problem with speckle noise was still present. It was therefore necessary to apply a despeckling filter in the inflammation detection part to. Unlike for bone and skin, the goal was to remove the speckle while preserving the dark regions that potentially could be the synovitis. Therefore, a lighter filtering with fewer iterations was required.

After doing Experiment 4.1, it was discovered that a homomorphic Butterworth high-boost filter managed to increase the distinctiveness of the inflammation region, thus avoiding the loss of small inflammation regions when applying smoothing on the images. The homomorphic filter was used in combination with a PM filter, which was used for light-despeckling and smoothing of the image. Visually, this filtering method seems to work as intended.

In the bone and skin segmentation part, it was necessary to remove the yellow channel of the images prior to the grayscale conversion and filtering, since the yellow box in the PD images led to problems. After visual inspection, it was concluded that the yellow box did not affect the segmenting of the inflammation much and was therefore not removed.

8.2.3 Segmentation

Three segmenting methods were tested: Region growing, Superpixel and Active contour. In addition, some variations of the algorithms were tested. However, in the context of the experiments, these variations, e.g. Chan-Vese and edge based active contour, were considered as parameters. Each of the segmenting methods were used in combination with morphological operators and novel algorithms used to restrict location of the synovitis and do initial thresholding. Locating the inflammation region was done by using Bone1 as a reference point. Thus, the low results of Bone2 did not affect the localization of the inflammation region.

The main issue when using a Region growing method is the placement of the initial seed point and the choice of intensity distance. It was therefore developed three algorithms, designed to automatically locate potential seed points and eventually place the initial seed point. Experiment 5.1 show that using an elliptic mask based on Algorithm 4 gave the best results. Therefore, only this seed point algorithm was used in Experiment 5.2. Some adjustments were made to the shape and location of the ellipse, as well as the intensity distance, to optimize the region growing in the last experiment. This proved to improve the results, both visually and regarding the Dice coefficient, compared to the results from Experiment 5.1. Also, it gave the overall best results on D4 in Experiment 5.2 with a mean and median Dice of 54 and 58 respectively. However, it now yielded zero-minimum values, i.e. some of the segmented inflammation regions fully missed the annotated regions.

Experiment 5.1 showed that Superpixel gave the highest median, but the overall lowest results on T4D4. The main issue was the threshold used to segment the image after computing the median of each superpixel. Often, it would result in large inflammation regions because it concatenated too many of the superpixels. One of the reasons for this was that the number of superpixels was fixed, which lead to images where the ROI was smaller consequently had smaller and finer superpixels than the bigger ones. Some of the thresholding problems were improved by using a percentage of the median as the threshold, instead of the threshold depicted in Algorithm 1. Even though the results for the tuning images in T4D4 were not improved, the resulting Dice coefficient for the images in D4 were increased as Table 32 shows. However, this method could be improved even more, by making an algorithm that adjusted the number of superpixel according to the size of the ROI, and one for smoothing the resulting boundary of the segmented inflammation region.

When the first parameter adjustment was made to the Active contour method, the resulting ellipse from Algorithm 4 was used as the initial contour. Both the Chan-Vese's and the edge-based method's initial contour evolved for 50 iterations, with the restriction that it should grow outwards. Overall, the best results from Experiment 5.1 came by using the edge-based method. However, the shape of the segmented region still looked much like the initial elliptic mask and not like the inflammation region. In the first attempt to solve this problem, the initial contour was allowed to evolve for a longer period of time and the restrictions on the evolution were remove. This proved to cause even more problems, since it led to the contour splitting into multiple segments. This lead to the development of Algorithm 5, which used the most probable object in the resulting binary image, derived from the initial threshold in Algorithm 1 as the initial contour. In addition, the algorithm dilated the segmented areas to accommodate the sometimes-generous annotations. Then the Chan-Vese based active contour with 100 iterations and no restriction was applied. The reason for the use of Chan-Vese instead of Edge base, even though the latter gave the best results in the previous experiment, was that the former was twice as fast and yielded almost the same results. Using the new initial contour resulted in a mean and median Dice of 60 and 64 respectively, which was the highest measurement on T4D4. Also, the detected inflammation regions looked the best on T4D4. However, when running this algorithm on D4, it gave overall the third lowest results, with only a mean and median of 50 and 51 respectively. This means that the parameters used in Algorithm 5 were overfitted to the images in T4D4.

8.2.4 Similarity measurement and materials

Similarity measurement was done by computing the Dice coefficient between the binarized annotation and segmented inflammation region. The goal was that median dice should be at least 70, as well as having a minimum value above 0. As the results show, the overall Dice coefficients were low, with the highest measured median of 64 using Algorithm 5 on T4D4. However, when inspecting the images visually, it revealed that the Dice often did not reflect the accuracy of the segmented inflammation region. Sometimes it was due to flaws in the segmenting algorithms and sometimes because the annotations were too generous.

For the tuning data in T4D4, the Dice was above 0 for all the methods, except when using Superpixel. However, since all the resulting minimum Dice's for D4 were 0, it means that the segmenting methods that were tested did not work on all the images. After visual inspection, this turned out to be due to that the inflammation was detected in the wrong place, which was mainly because the inflammation regions were small in the images. An attempt was made to adjust the parameters for these images, but that lead to the results getting worse in other images, and the overall results got worse or stayed idle. Since there was a big variation in the images, a generalization of the algorithms became difficult. Therefore, the main focus was to get the highest possible median and mean as well as low STDEV.

The goal was to segment the dark inflammation region which develops from the joint area between the bones. Visual inspection of the segmented inflammation region shows that in most cases where the inflammation is correctly located, the segmenting algorithms work as intended. Still, the resulting Dice coefficient became low. This was due to that the manual annotations were often quite generous, meaning that it covered more than the dark region defining the inflammation. Figure 41 shows an example of the problem due to the generosity problem of the manually annotated data. Another source which reduced the Dice coefficient was small inflammation regions, because when comparing small surfaces, it requires more accuracy since a small difference amounts to a bigger percentage than with large surfaces, hence the low Dice in images such as the lower left image in Figure 56. Therefore, it could be argued that either the similarity of the segmented inflammation region should be based on a doctor's review, or that the annotations should be controlled again. The latter could also help in removing bad images and missing annotations. In addition, it could avoid confusions such as described in Figure 54. The difference in the images was also high, so a better standardization of the acquiring of the images could help improve the accuracy of all the tested methods.

9 Conclusion

The goal of this project was to develop a method for automatically segmentation of bone, skin and synovitis in US images of finger joints. Multiple well-known image processing techniques were tested in order to obtain this goal, including multiple filtering and segmenting algorithms. In addition, it was developed novel algorithms for localization and segmentation of skin, bones and synovitis.

Results show that the proposed method for segmentation of skin and bones was satisfying using PM-filter in combination with the novel bone and skin segmentation algorithm. However, some improvements to the segmentation of the right bone could be made. Regarding the segmentation of the synovitis or inflammation region, light filtering and segmentation using region growing yielded the best results. Even if the best results measured a low Dice coefficient, most of the detected synovitis was decent when visually inspected.

Even though the skin and bone segmentation was good, the proposed methods for segmentation of the inflammation region did not yield good enough results for future grading of the synovitis.

9.1 Future work

Future work would include improvement of the proposed methods, additional review of the materials, automatic labeling using machine learning, and finally automatic grading of the synovitis region.

Although the detection of skin and bones gave overall good results, it could be improved especially regarding Bone2. This could be done by separating the segmentation algorithm for bone detection in two. Using Bone1 as a reference point when locating Bone2 could be advantageous, since Bone1 was detected consistently good. Regarding the synovitis detection, all the algorithms could be improved. Both to remove localization error of the synovitis region as well as increasing the overall similarity of the segmented synovitis.

Additional review of the materials would also be advantageous, both to remove bad images and verification of the annotations accuracy. Alternative segmenting methods could also be considered, e.g. the use of machine learning to label the bone and skin features and locate the synovitis region. Finally, when the synovitis is adequately detected and segmented, grading it would be the next step.

References

[1] University of Oslo, "www.uio.no/," [Online]. Available: https://www.uio.no/for-ansatte/arbeidsstotte/profil/nettarbeid/veiledninger/bilde-lyd-video/bilder/creative-commons.html. [Accessed 03 June 2017].

[2] K. W. &. B. S. &. M. F. wm., "A machine-learning approach to the automated," Springer International Publishing, 2016.

[3] Department of Social Protection, "Rheumatoid Arthritis," [Online]. Available: http://www.welfare.ie/en/downloads/protocol14.pdf. [Accessed 13 February 2017].

[4] World health organisation , "who.int," WHO, [Online]. Available: http://www.who.int/chp/topics/rheumatic/en/. [Accessed 13 February 2017].

[5] P. L. C. M. v. R. &. H. R. S. Jr, "How does one assess early rheumatoid," in *Best Practice & Research Clinical Rheumatology Vol. 15*, Nijmegen, University Medical Center Nijmegen, Department of Rheumatology, 2001, pp. 67-76.

[6] R. S. A. C. B. R. V. B. E. S. A. C. L. C. U. F. E. G. Elisa Veronesea, "www.sciencedirect.com/," *mproved detection of synovial boundaries in ultrasound examination by using a cascade of active-contours,* pp. 188-194, February 2013.

[7] MEDUSA, "medusa.aei.polsl.pl," [Online]. Available: http://www.medusa.aei.polsl.pl/index.php/about-medusa. [Accessed 26 October 2016].

[8] medicinenet.com, "medicinenet.com," [Online]. Available: http://www.medicinenet.com/rheumatoid_arthritis/article.htm. [Accessed 08 March 2017].

[9] C. B. Driver, "MedicineNet.com," [Online]. Available: http://www.medicinenet.com/rheumatoid_arthritis/article.htm. [Accessed 13 February 2017].

[10] aihw.gov.au, "http://www.aihw.gov.au," [Online]. Available: http://www.aihw.gov.au/uploadedImages/Subjects/Arthritis_and_musculoskeletal_conditions/rheumatoid-arthritis-diagram.png. [Accessed 06 March 2017].

[11] AIHW, "Rheumatoid arthritis: musculoskeletal fact sheet," AIHW, 2015. [Online]. Available: http://www.aihw.gov.au/publication-detail/?id=60129551057. [Accessed 21 March 2017].

[12] L. B. L. M. O. B. N. L. N. M. M. Yabluchansky, "http://dspace.univer.kharkov.ua," [Online]. Available: http://dspace.univer.kharkov.ua/bitstream/123456789/11992/2/Lecture_Rheumatoid_Arthritis.pdf. [Accessed 06 March 17].

[13] M. Johnson, "http://buyprovailencheap.org," 17 Octrober 2013. [Online]. Available: http://buyprovailencheap.org/rheumatoid-arthritis-osteoarthritis-septic-arthritis-juvenile-arthritis/. [Accessed 21 March 2017].

[14] Webmd, "webmd.com," [Online]. Available: http://www.webmd.com/a-to-z-guides/magnetic-resonance-imaging-mri#1. [Accessed 08 March 2017].

[15] Radiology St.Vincent's University Hospital, "Radiology St.Vincent's University Hospital," [Online]. Available: http://www.svuhradiology.ie/case-study/rheumatoid-arthritis-hands/. [Accessed 21 March 2017].

[16] M. K. E. N. M. C.-P. C. S. K. E. J. H. S. T. a. M. Ø. Marcin Szkudlarek, "arthritis-research.biomedcentral.com," 6 March 2006. [Online]. Available: http://arthritis-research.biomedcentral.com/articles/10.1186/ar1904. [Accessed 22 March 2017].

[17] Wikimedia, "commons.wikimedia.org," [Online]. Available: https://commons.wikimedia.org/wiki/File:Rheumatoid_arthritis_ultrasound_MRI_MCP_joint_ar1904-2.gif. [Accessed 22 March 2017].

[18] S. b. &. P. P.Cheung, "The Role of Power Doppler Ultrasonography as Disease Activity," Hindawi Publishing Corporation, Singapore, 2015.

[19] M. Ø. &. M. Szudlarek, "Ultrasonography: A Valid Method for Assessing Rheumatoid Arthritis?," American college of Rheumatology , 2005.

[20] Webmd, "webmd.com," [Online]. Available: http://www.webmd.com/a-to-z-guides/what-is-an-ultrasound#1. [Accessed 08 March 2017].

[21] H. A. Faouzi Benzarti, "Speckle Noise Reduction in Medical Ultrasound Images," 2013. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1305/1305.1344.pdf. [Accessed 14 February 2017].

[22] D. C. C. K. Karthikeyan, "Speckle Noise Reduction of Medical Ultrasound Images," *International Journal of computer Applications,* vol. 22, no. 9, pp. 8-14, 9 May 2011.

[23] Z. J. &. X. Yang, "A Variational Model to Remove the Multiplicative Noise in Ultrasound Images," *Journal of Mathematical Imaging and Vision,* vol. 39, no. 1, pp. 62-74, 28 Semptember 2010.

[24] Oleg V. Michailovich & Allen Tannenbaum, "www.ncbi.nlm.nih.gov," 29 April 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3639001/. [Accessed 25 March 17].

[25] Eeagrants., "http://eeagrants.org," [Online]. Available: http://eeagrants.org/project-portal/project/PL12-0015. [Accessed 21 February 2017].

[26] B. S. Adam Popowicz, "Isoline Based Image Colorization," IEEE Digital Library, Gliwice, 2014.

[27] A. C.Bovik, "Diffusion Partial Differential Equations for edge detection," in *The Essential Guide to Image Processing, 2nd Edit*, Elsevier Science, 2009, pp. 525-550 .

[28] T. S. &. J. M. Petro Perona, "Geometry-Driven Diffusion in Computer Vision," in *Geometry-Driven Diffusion in Computer Vision*, Springer Netherlands, 1994, pp. 73-92.

[29] Y. Y. &. S. Acton, "Speckle reducing anisotropic diffusion," in *IEEE Transactions on Image Processing* , IEEE, 2002, pp. 1260-1270.

[30] Y. T. &. M. R. Hanung Adi Nugroho, "Performance analysis of filtering techniques for speckle reduction on breast ultrasound images," IEEE, 2017.

[31] S. W. S. J. M. S. H. L. Robert f. Wagner, "Statistics of Speckle in Ultrasound B-Scans," IEE, 1983.

[32] C. R.-F. J. S. Kathuria Chirinjeev, "Despecling filters," in *Advances in Diagnostic and Therapeutic Ultrasound Imaging*, Artech House, 2008, pp. 40-47.

[33] M. P. &. C. Petrou, Image Processing. The fundamentals, 2nd edit, West Sussex: John Wiley & Sons, 2010.

[34] A. A. &. Nixton, "Basic image processing operators," in *Feature Extraction and Image Processing for Computer Vision*, Elsevier Science, 2012, pp. 72-80.

[35] A. C. Bovik, "Basic Gray Level Image Processing," in *The Essential Guide to Image Processing, 2nd Edit*, Elsevier Science, 2008, pp. 44-69.

[36] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," in *Graphics Gems IV*, San Diego, Academic press. Inc, 1994, pp. 474-485.

[37] U. B. &. F. Diekmann, Digital Mammography, Berlin: Springer Berlin Heidelberg, 2010.

[38] M. D. &. M. I. K. Smriti Sahu, "Liver Ultrasound Image Analysis using Enhancement Techniques," *International Journal of Advanced Computer Research ,* vol. 2, no. 6, 2012.

[39] S. Frank Y, Image processing and Marhematical Morphology, New York: CRC Press , 2009.

[40] O. Marques, Practical Image and Video Processing Using MATLAB, Wileay, 2011.

[41] E. R. A. R. A. Lotufo, "Hands-on Morphological Image Processing," Washington, Spie Press, 2003, pp. 79-159.

[42] F. M. &. M. W. Kenneth Castleman, "Morphological Image Processing," in *Microscope Image Processing*, Elsevier Science, 2010, pp. 113-157.

[43] I. Bankman, "Handbook of Medical Image Processing and Analysis," in *Semengtation*, Elsevier Science, 2008, pp. 70-89.

[44] A. S. K. S. L. F. &. S. S. Radhakrishna Achanta, "SLIC Superpixels Compared State-of-the-Art Superpixel Methods," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,,* vol. 34, no. 11, pp. 2274-2281, 2012.

[45] C. Chen, "Active Contours," in *Computer Vision in Medical Imaging*, World Scientific Publishing Company, 2013, pp. 6-12.

[46] S.-H. O. a. L. J.-H. Wei Xiong, "Segmentation and Object Detection," in *Wiley Series in Biomedical Engineering and Multi-Disciplinary Integrated Systems*, John Wiley & Sons, Incorporated, 2015, pp. 78-101.

[47] Z. P. P. W. W. L. W. W. G. W. Lei He, "A comparative study of deformable contour methods," *Image and Vision Computing,* vol. 26, no. 2, pp. 141-163, 2008.

[48] T. F. C. &. L. A. Vese, "IEEE Active Contours Without Edges," *TRANSACTIONS ON IMAGE PROCESSING,* vol. 10, no. 2, pp. 266-277, 2001.

[49] "Sørensen–Dice coefficient," [Online]. Available: http://en.wikipedia.org/wiki/Sørensen–Dice_coefficient. [Accessed 03 April 2017].

[50] E. Vezzetti, "Hausdorff Distance for Face Recognition," in *Similarity Measures for Face Recognition*, Bentham Science Publishers, 2015, pp. 39-46.

[51] G. A. K. &. W. J. R. Daniel P. Huttenlocher, "Comparing Images Using the Hausdorff Distance," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 15, no. 9, pp. 850-863, 1993.

[52] M.-P. D. a. A. E. Jain, A Modified Hausdorff Distance for Object Matching, IEEE, 1994.

[53] Matworks, "se.mathworks.com," [Online]. Available: https://se.mathworks.com/help/. [Accessed 04 May 2017].

[54] Ch.Begler, "Ch.Begler," mooring.ucsd.edu, 09 May 2009. [Online]. Available: http://mooring.ucsd.edu/software/matlab/doc/toolbox/graphics/color/rgb2cmyk.html. [Accessed 04 May 2017].

[55] D. Lopes, "se.mathworks.com," [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/14995-anisotropic-diffusion--perona---malik-. [Accessed 04 May 2017].

[56] Q. Ahmed, "se.mathworks.com," [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/21357-homomorphic-filtering. [Accessed 05 May 2017].

[57] M. Athiq, "se.mathworks.com/," [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/40579-frequency-domain-filtering-for-grayscale-images. [Accessed 05 May 2017].

[58] Z. Danziger, "se.mathworks.com," [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/26738-hausdorff-distance. [Accessed 05 May 2017].

[59] Sasikanth, "se.mathworks.com," [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/29968-modified-hausdorff-distance. [Accessed 05 May 2017].

[60] MatWorks, "se.mathworks.c," [Online]. Available: https://se.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html. [Accessed 08 May 2017].

[61] J. Sethian, Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Cambridge: Cambridge University Press, 1999.

[62] statisticshowto.com, "www.statisticshowto.com," [Online]. Available: http://www.statisticshowto.com/jaccard-index/. [Accessed 03 April 2017].

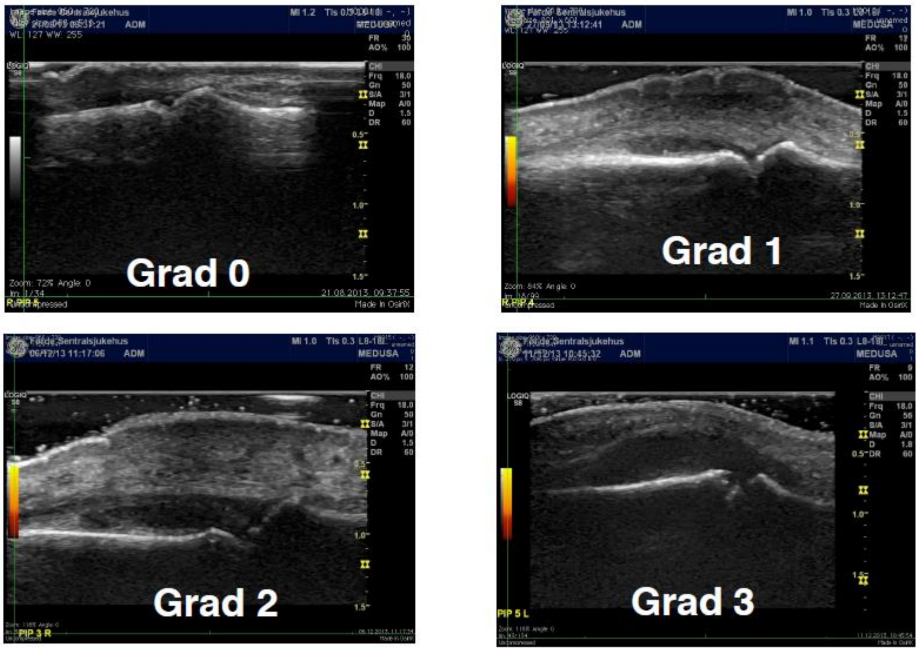
[63] Kildekompasset.no , "kildekompasset.no," [Online]. Available: http://kildekompasset.no/referansestiler/ieee.aspx. [Accessed 24 March 2017].

Appendixes

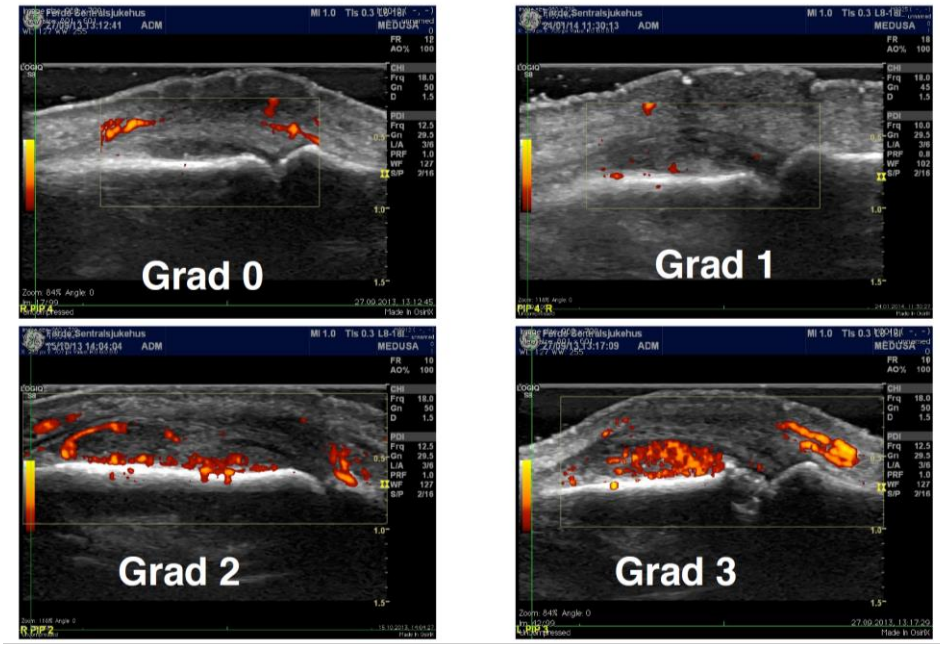
Appendix-A Medical

Appendix-A.1 Atlas from of USG MEDUSA

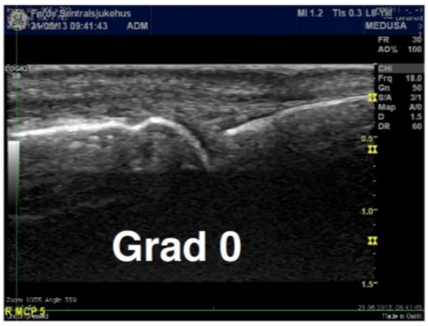
PIP, B-mode Hypertrophy

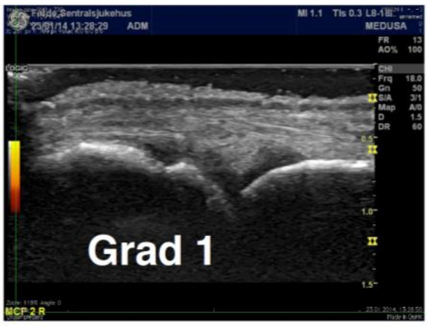


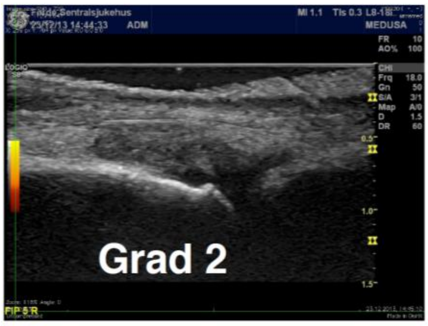
PIP, Doppler

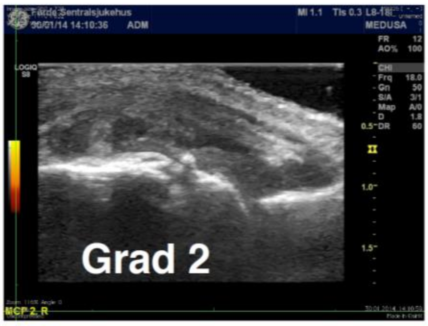


MCP, B-mode Hypertrophy

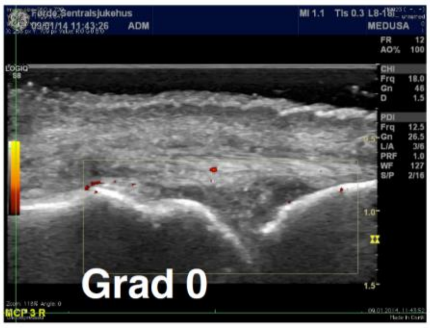


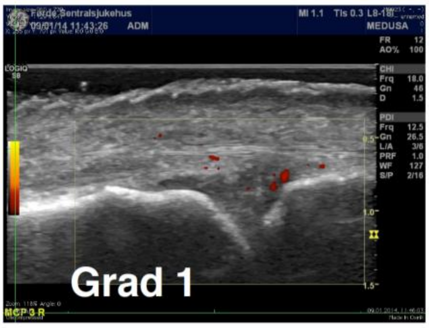


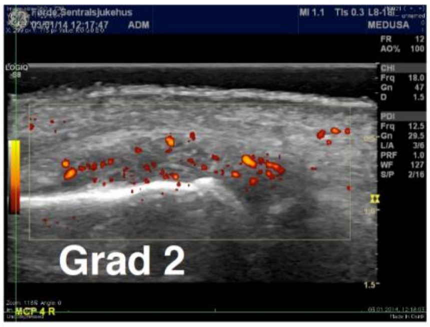


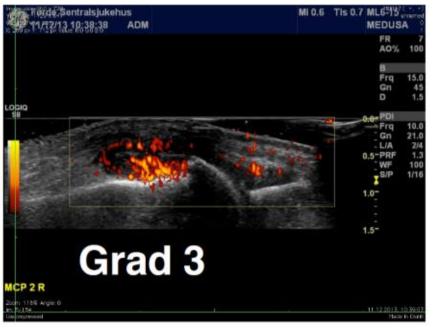


MCP, Doppler

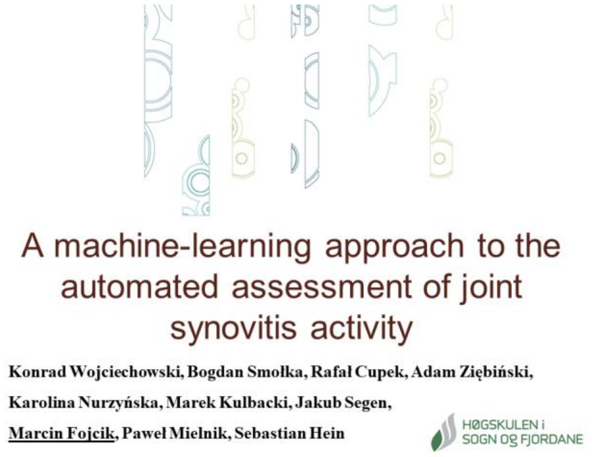






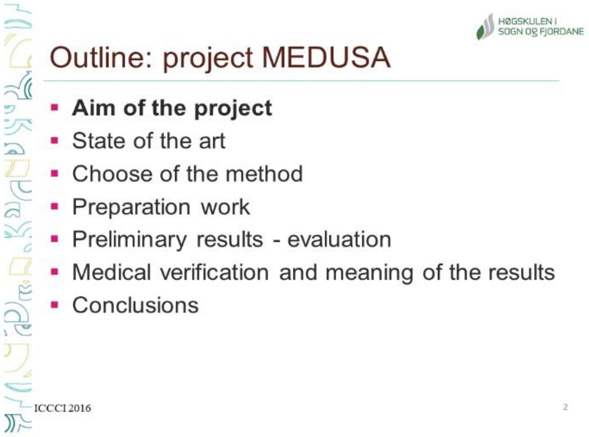


Appendix-A.2 **MEDUSA presentation**



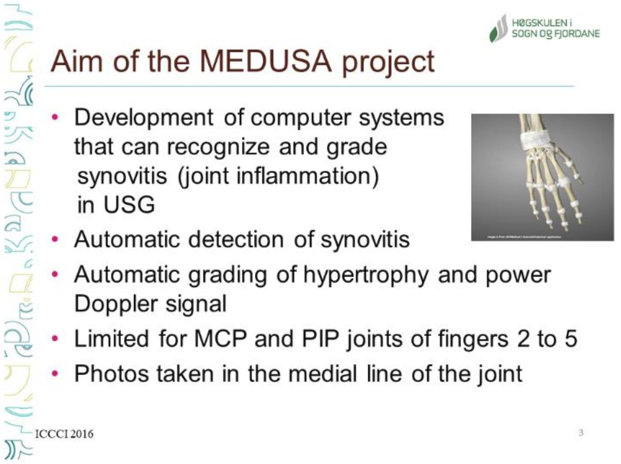
A machine-learning approach to the automated assessment of joint synovitis activity

Konrad Wojciechowski, Bogdan Smołka, Rafał Cupek, Adam Ziębiński, Karolina Nurzyńska, Marek Kulbacki, Jakub Segen, Marcin Fojcik, Paweł Mielnik, Sebastian Hein



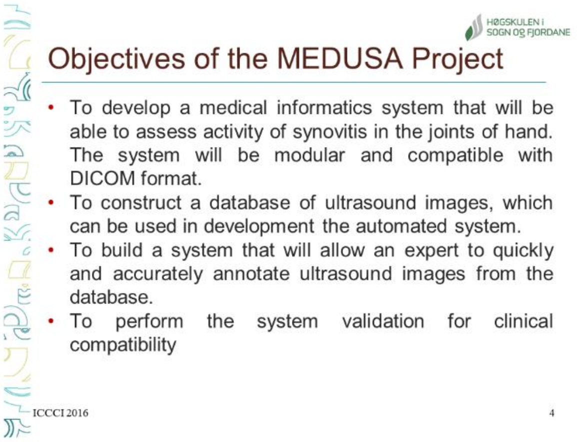
Outline: project MEDUSA

- **Aim of the project**
- State of the art
- Choose of the method
- Preparation work
- Preliminary results - evaluation
- Medical verification and meaning of the results
- Conclusions



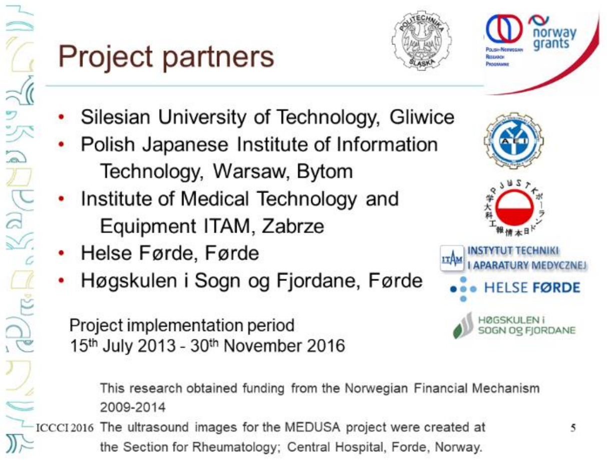
Aim of the MEDUSA project

- Development of computer systems that can recognize and grade synovitis (joint inflammation) in USG
- Automatic detection of synovitis
- Automatic grading of hypertrophy and power Doppler signal
- Limited for MCP and PIP joints of fingers 2 to 5
- Photos taken in the medial line of the joint



Objectives of the MEDUSA Project

- To develop a medical informatics system that will be able to assess activity of synovitis in the joints of hand. The system will be modular and compatible with DICOM format.
- To construct a database of ultrasound images, which can be used in development the automated system.
- To build a system that will allow an expert to quickly and accurately annotate ultrasound images from the database.
- To perform the system validation for clinical compatibility



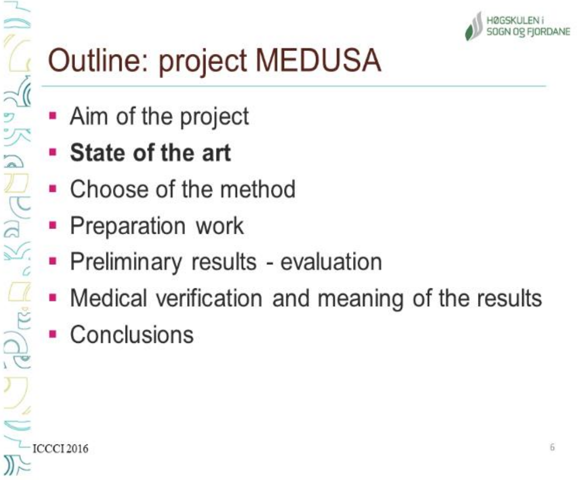
Project partners

- Silesian University of Technology, Gliwice
- Polish Japanese Institute of Information Technology, Warsaw, Bytom
- Institute of Medical Technology and Equipment ITAM, Zabrze
- Helse Førde, Førde
- Høgskulen i Sogn og Fjordane, Førde

Project implementation period
15th July 2013 - 30th November 2016

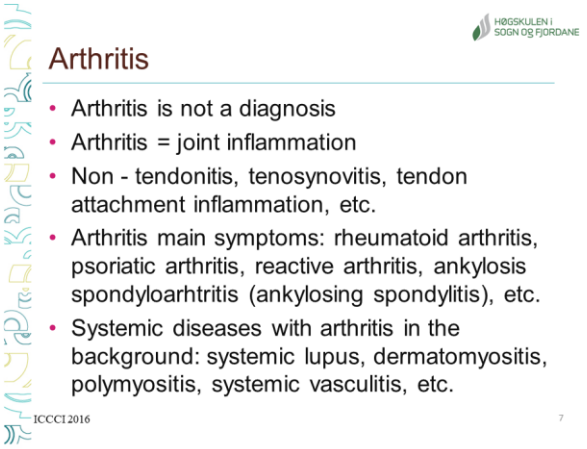
This research obtained funding from the Norwegian Financial Mechanism 2009-2014

The ultrasound images for the MEDUSA project were created at the Section for Rheumatology; Central Hospital, Forde, Norway.



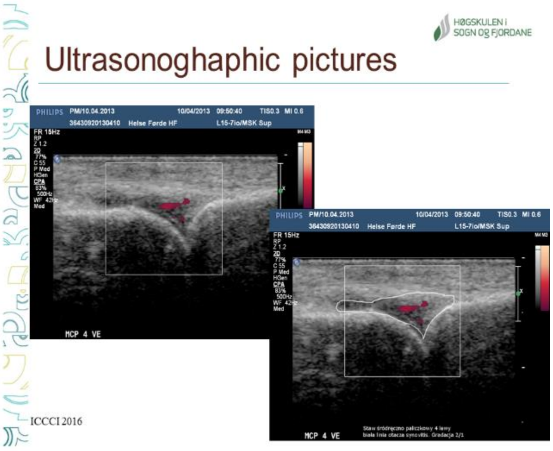
Outline: project MEDUSA

- Aim of the project
- **State of the art**
- Choose of the method
- Preparation work
- Preliminary results - evaluation
- Medical verification and meaning of the results
- Conclusions



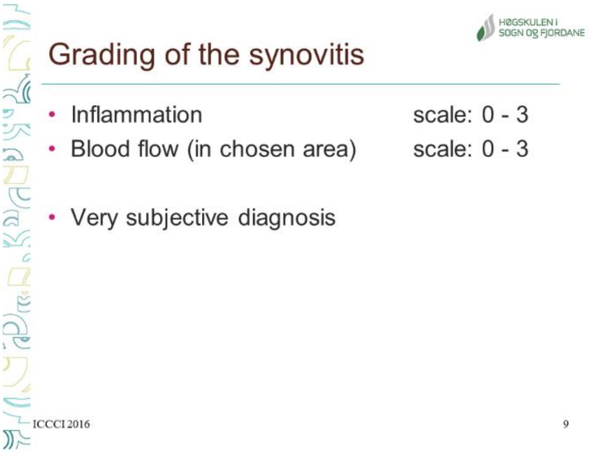
Arthritis

- Arthritis is not a diagnosis
- Arthritis = joint inflammation
- Non - tendonitis, tenosynovitis, tendon attachment inflammation, etc.
- Arthritis main symptoms: rheumatoid arthritis, psoriatic arthritis, reactive arthritis, ankylosis spondyloarhtritis (ankylosing spondylitis), etc.
- Systemic diseases with arthritis in the background: systemic lupus, dermatomyositis, polymyositis, systemic vasculitis, etc.



Ultrasonoghaphic pictures

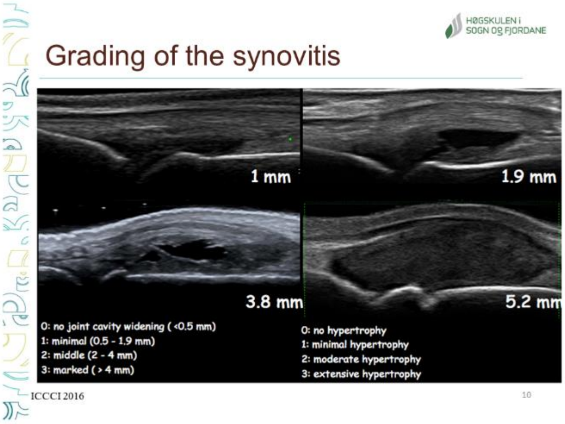
Staw śródręczno paliczkowy 4 lewy biała linia otacza synovitis. Gradacja 2/1



Grading of the synovitis

• Inflammation	scale: 0 - 3
• Blood flow (in chosen area)	scale: 0 - 3

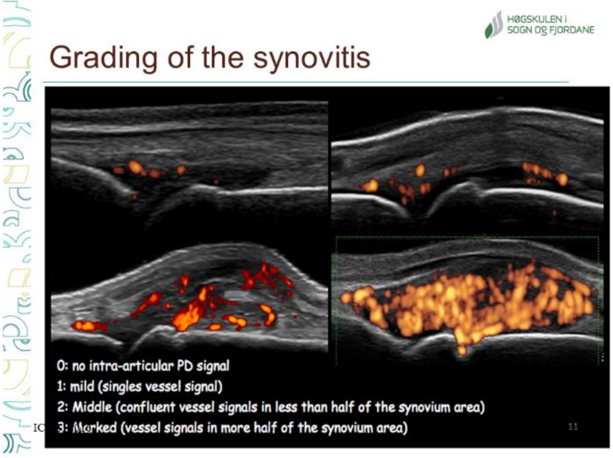
- Very subjective diagnosis



Grading of the synovitis

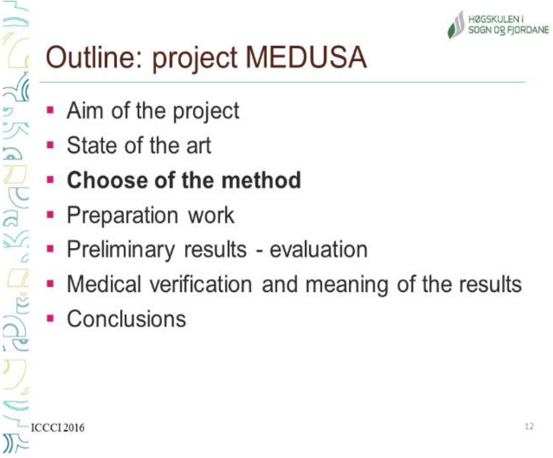
0: no joint cavity widening (<0.5 mm)
1: minimal (0.5 - 1.9 mm)
2: middle (2 - 4 mm)
3: marked (> 4 mm)

0: no hypertrophy
1: minimal hypertrophy
2: moderate hypertrophy
3: extensive hypertrophy



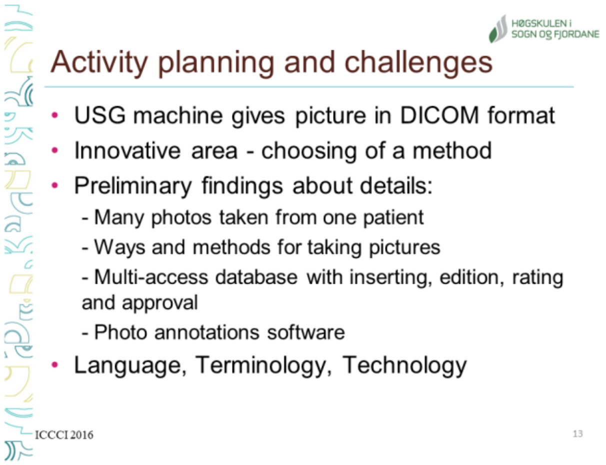
Grading of the synovitis

0: no intra-articular PD signal
1: mild (singles vessel signal)
2: Middle (confluent vessel signals in less than half of the synovium area)
3: Marked (vessel signals in more half of the synovium area)



Outline: project MEDUSA

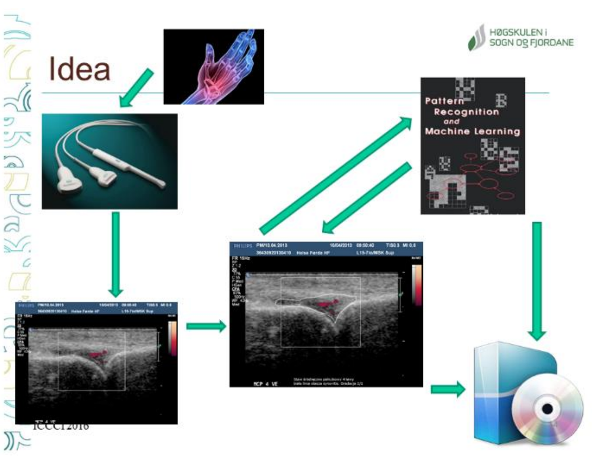
- Aim of the project
- State of the art
- **Choose of the method**
- Preparation work
- Preliminary results - evaluation
- Medical verification and meaning of the results
- Conclusions



Activity planning and challenges

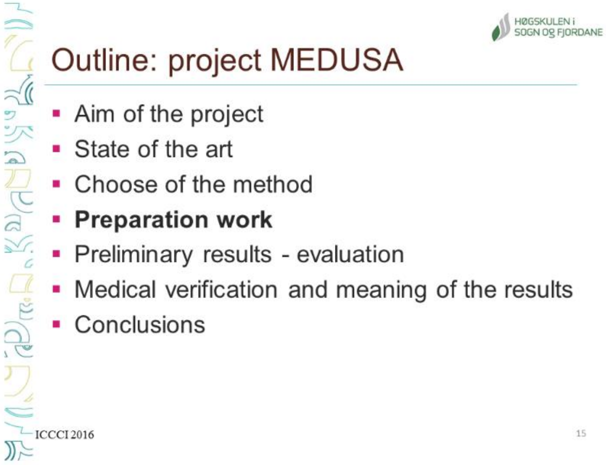
- USG machine gives picture in DICOM format
- Innovative area - choosing of a method
- Preliminary findings about details:
  - Many photos taken from one patient
  - Ways and methods for taking pictures
  - Multi-access database with inserting, edition, rating and approval
  - Photo annotations software
- Language, Terminology, Technology

ICCCI 2016 13



Idea

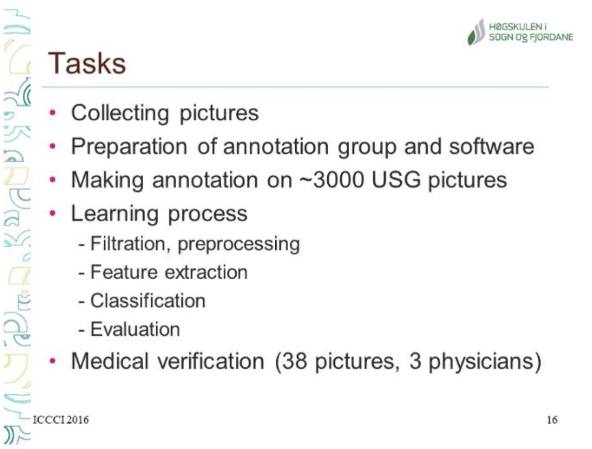
ICCCI 2016



Outline: project MEDUSA

- Aim of the project
- State of the art
- Choose of the method
- **Preparation work**
- Preliminary results - evaluation
- Medical verification and meaning of the results
- Conclusions

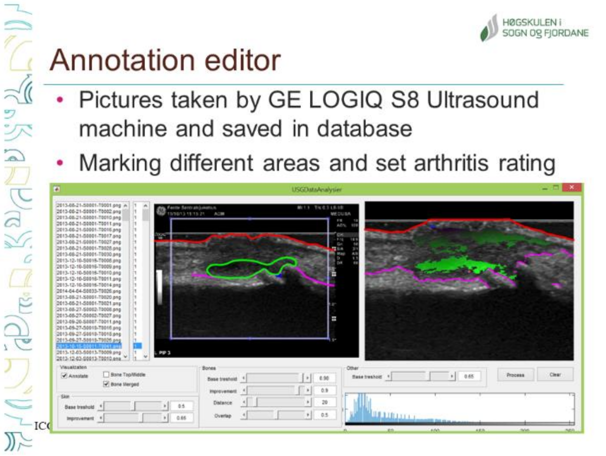
ICCCI 2016 15



Tasks

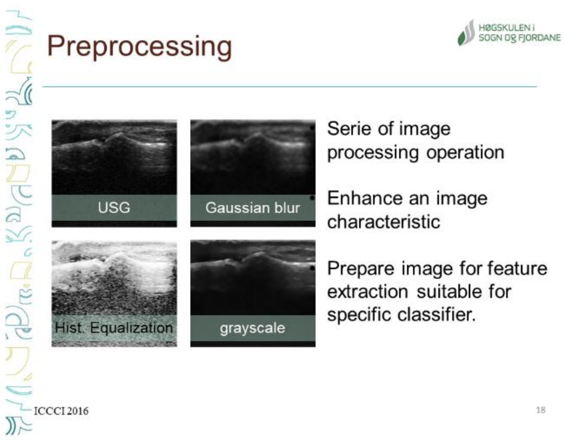
- Collecting pictures
- Preparation of annotation group and software
- Making annotation on ~3000 USG pictures
- Learning process
  - Filtration, preprocessing
  - Feature extraction
  - Classification
  - Evaluation
- Medical verification (38 pictures, 3 physicians)

ICCCI 2016 16



Annotation editor

- Pictures taken by GE LOGIQ S8 Ultrasound machine and saved in database
- Marking different areas and set arthritis rating



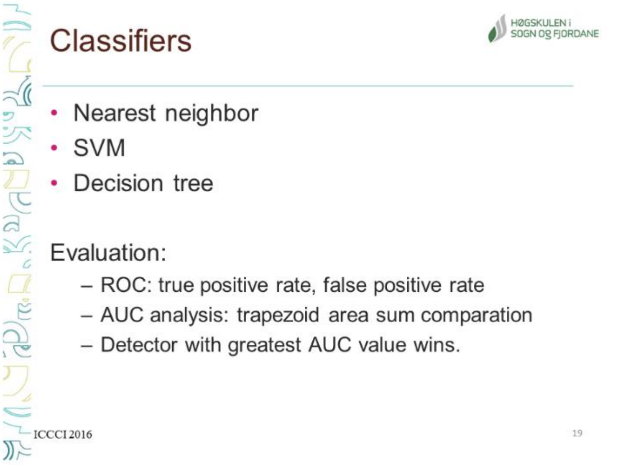
Preprocessing

Serie of image processing operation

Enhance an image characteristic

Prepare image for feature extraction suitable for specific classifier.

ICCCI 2016 18



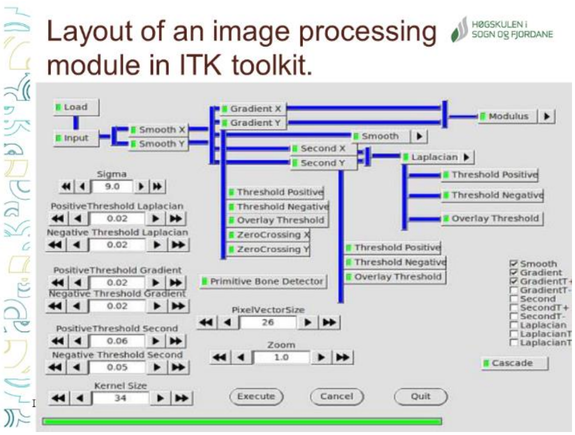
Classifiers

- Nearest neighbor
- SVM
- Decision tree

Evaluation:

- ROC: true positive rate, false positive rate
- AUC analysis: trapezoid area sum comparation
- Detector with greatest AUC value wins.

ICCCI 2016 19

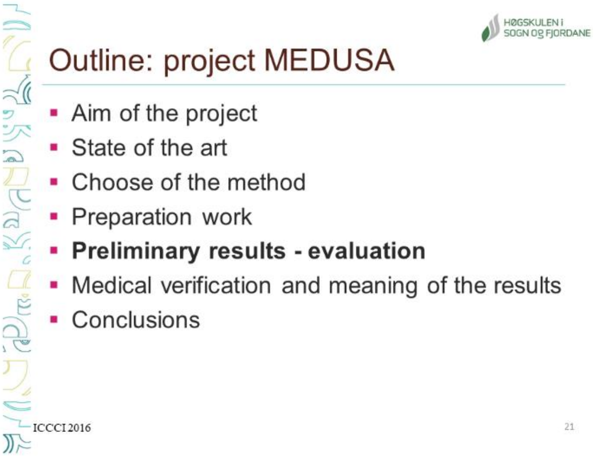


Layout of an image processing module in ITK toolkit.

Sigma 9.0
PositiveThreshold Laplacian 0.02
Negative Threshold Laplacian 0.02
PositiveThreshold Gradient 0.02
Negative Threshold Gradient 0.02
PositiveThreshold Second 0.06
Negative Threshold Second 0.05
Kernel Size 34
PixelVectorSize 26
Zoom 1.0

☑ Smooth
☑ Gradient
☑ GradientT+
☐ GradientT-
☐ Second
☐ SecondT+
☐ SecondT-
☐ Laplacian
☐ LaplacianT
☐ LaplacianT

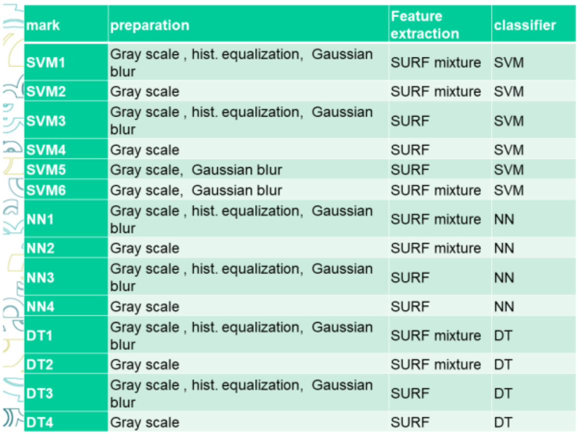
Load
Input
Smooth X
Smooth Y
Gradient X
Gradient Y
Modulus
Smooth
Second X
Second Y
Laplacian
Threshold Positive
Threshold Negative
Overlay Threshold
ZeroCrossing X
ZeroCrossing Y
Threshold Positive
Threshold Negative
Overlay Threshold
Threshold Positive
Threshold Negative
Overlay Threshold
Primitive Bone Detector
Cascade
Execute
Cancel
Quit



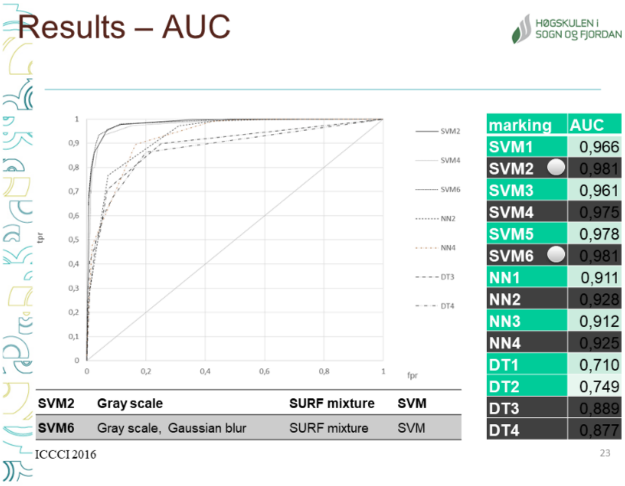
Outline: project MEDUSA

- Aim of the project
- State of the art
- Choose of the method
- Preparation work
- **Preliminary results - evaluation**
- Medical verification and meaning of the results
- Conclusions

ICCCI 2016 21



mark	preparation	Feature extraction	classifier
SVM1	Gray scale , hist. equalization, Gaussian blur	SURF mixture	SVM
SVM2	Gray scale	SURF mixture	SVM
SVM3	Gray scale , hist. equalization, Gaussian blur	SURF	SVM
SVM4	Gray scale	SURF	SVM
SVM5	Gray scale, Gaussian blur	SURF	SVM
SVM6	Gray scale, Gaussian blur	SURF mixture	SVM
NN1	Gray scale , hist. equalization, Gaussian blur	SURF mixture	NN
NN2	Gray scale	SURF mixture	NN
NN3	Gray scale , hist. equalization, Gaussian blur	SURF	NN
NN4	Gray scale	SURF	NN
DT1	Gray scale , hist. equalization, Gaussian blur	SURF mixture	DT
DT2	Gray scale	SURF mixture	DT
DT3	Gray scale , hist. equalization, Gaussian blur	SURF	DT
DT4	Gray scale	SURF	DT

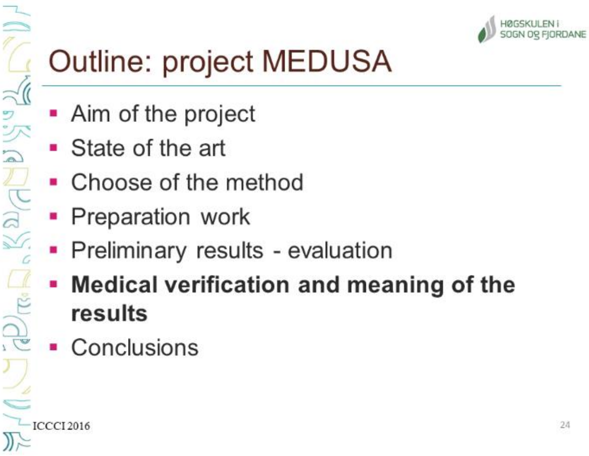


Results – AUC

SVM2	**Gray scale**	**SURF mixture**	**SVM**
SVM6	Gray scale, Gaussian blur	SURF mixture	SVM

marking	AUC
SVM1	0,966
SVM2 ●	0,981
SVM3	0,961
SVM4	0,975
SVM5	0,978
SVM6 ●	0,981
NN1	0,911
NN2	0,928
NN3	0,912
NN4	0,925
DT1	0,710
DT2	0,749
DT3	0,889
DT4	0,877

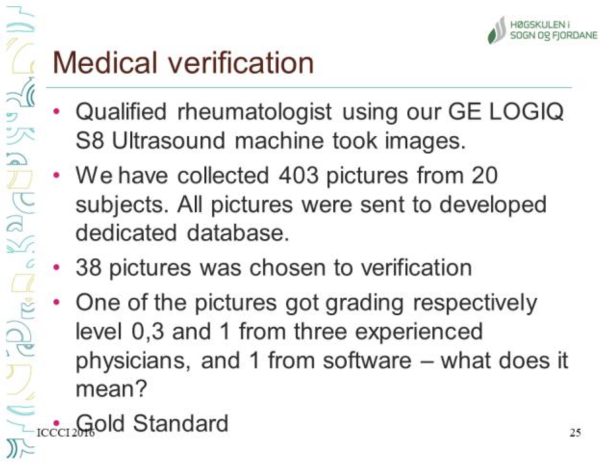
ICCCI 2016 23



Outline: project MEDUSA

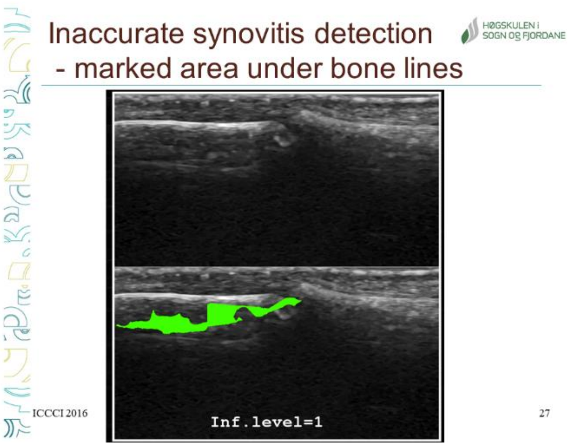
- Aim of the project
- State of the art
- Choose of the method
- Preparation work
- Preliminary results - evaluation
- **Medical verification and meaning of the results**
- Conclusions

ICCCI 2016 24

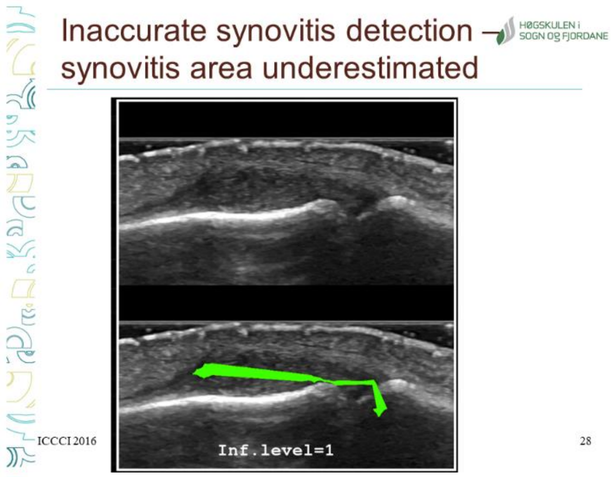


Medical verification

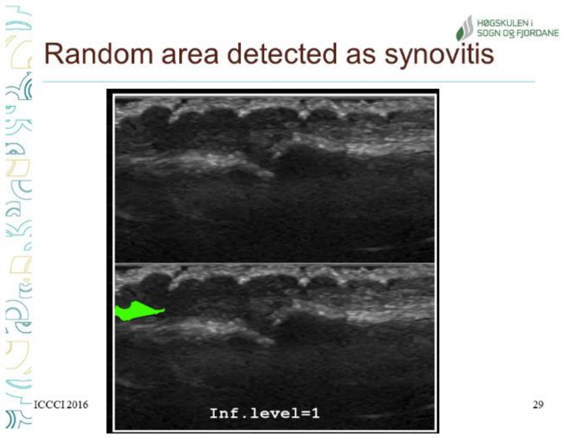
- Qualified rheumatologist using our GE LOGIQ S8 Ultrasound machine took images.
- We have collected 403 pictures from 20 subjects. All pictures were sent to developed dedicated database.
- 38 pictures was chosen to verification
- One of the pictures got grading respectively level 0,3 and 1 from three experienced physicians, and 1 from software – what does it mean?
- Gold Standard



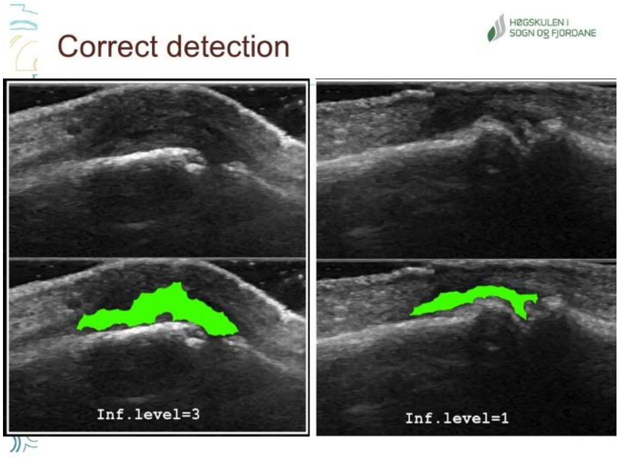
Inaccurate synovitis detection - marked area under bone lines



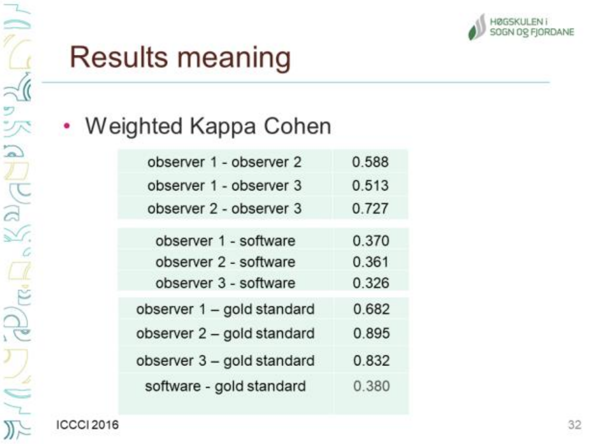
Inaccurate synovitis detection – synovitis area underestimated



Random area detected as synovitis



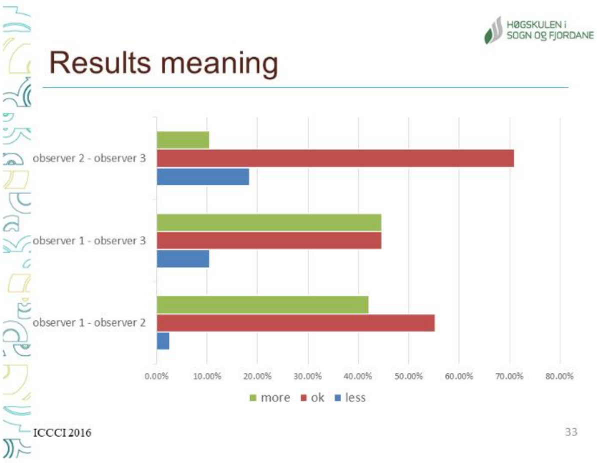
Correct detection



Results meaning

- Weighted Kappa Cohen

observer 1 - observer 2	0.588
observer 1 - observer 3	0.513
observer 2 - observer 3	0.727
observer 1 - software	0.370
observer 2 - software	0.361
observer 3 - software	0.326
observer 1 – gold standard	0.682
observer 2 – gold standard	0.895
observer 3 – gold standard	0.832
software - gold standard	0.380

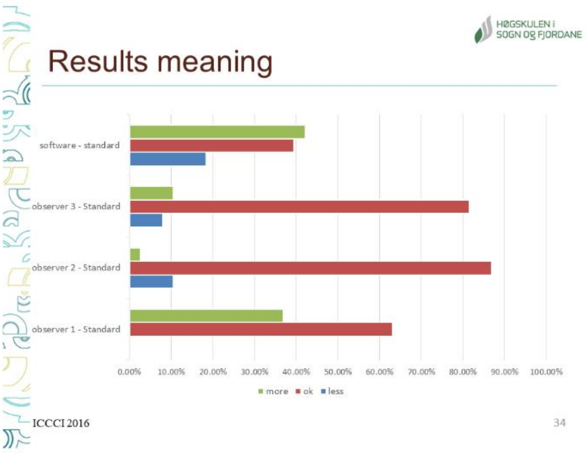


HØGSKULEN I SOGN OG FJORDANE

Results meaning

ICCCI 2016

33

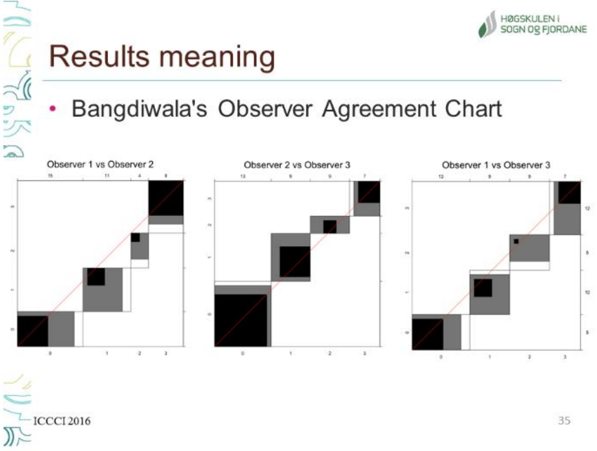


HØGSKULEN I SOGN OG FJORDANE

Results meaning

ICCCI 2016

34



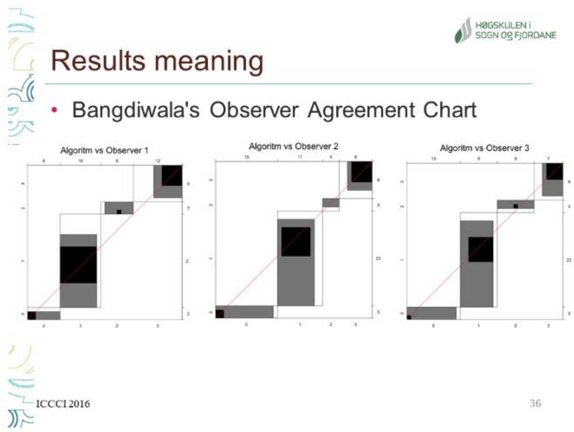
HØGSKULEN I SOGN OG FJORDANE

Results meaning

- Bangdiwala's Observer Agreement Chart

ICCCI 2016

35



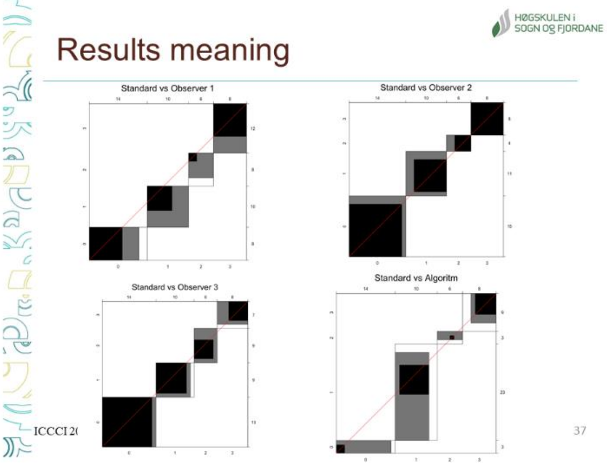
HØGSKULEN I SOGN OG FJORDANE

Results meaning

- Bangdiwala's Observer Agreement Chart

ICCCI 2016

36

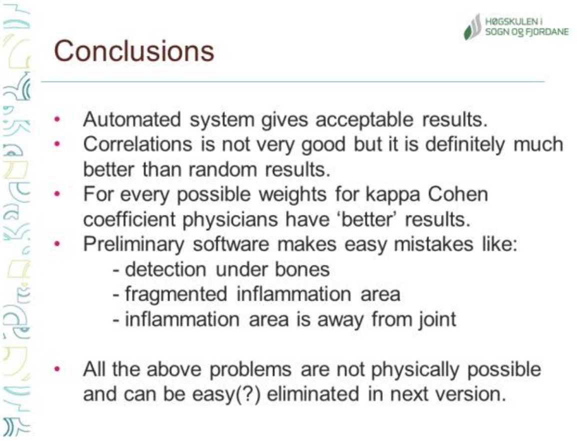


HØGSKULEN I SOGN OG FJORDANE

Results meaning

ICCCI 2

37



HØGSKULEN I SOGN OG FJORDANE

Conclusions

- Automated system gives acceptable results.
- Correlations is not very good but it is definitely much better than random results.
- For every possible weights for kappa Cohen coefficient physicians have 'better' results.
- Preliminary software makes easy mistakes like:
  - detection under bones
  - fragmented inflammation area
  - inflammation area is away from joint

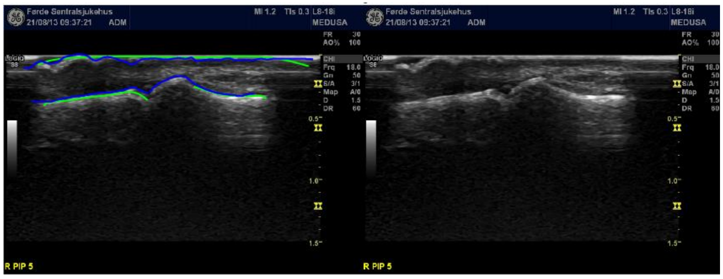
- All the above problems are not physically possible and can be easy(?) eliminated in next version.

Appendix-B Results

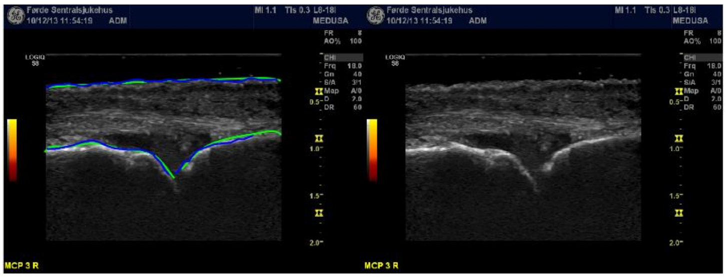
Appendix-B.1 Additional visual results Experiment 4.2

Inn all images the annotations are in blue and the segmented bone and skin features are in green.

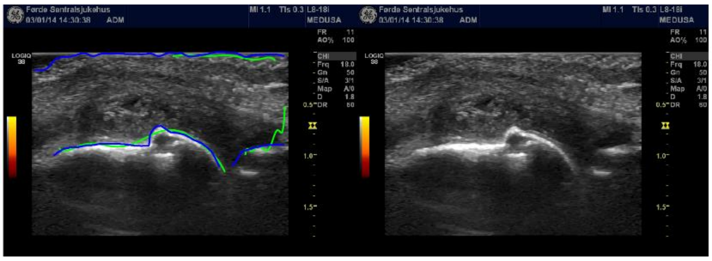
MHD Bone1=7, MHD Bone2=3, MHD Skin=7



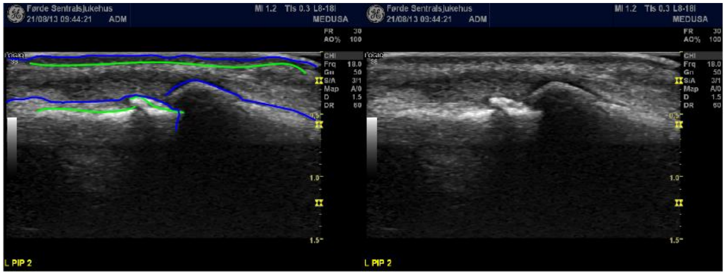
MHD Bone1=3, MHD Bone2=6, MHD Skin=3



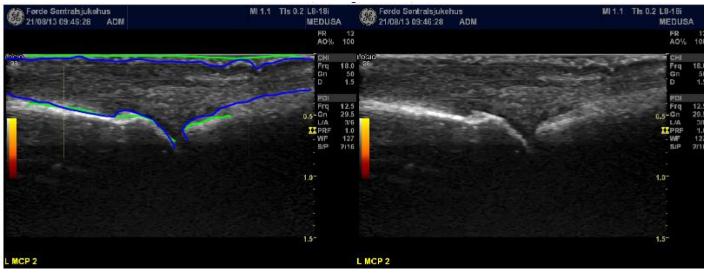
MHD Bone1=6, MHD Bone2=25, MHD Skin=7



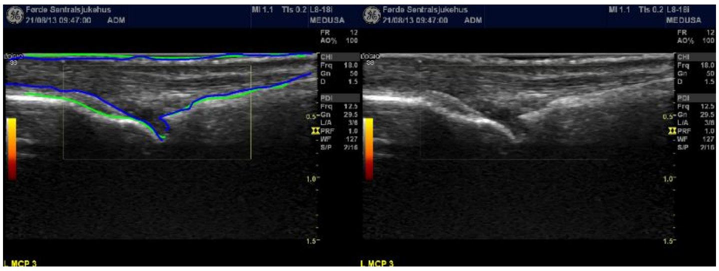
MHD Bone1=26, MHD Bone2=60, MHD Skin=19



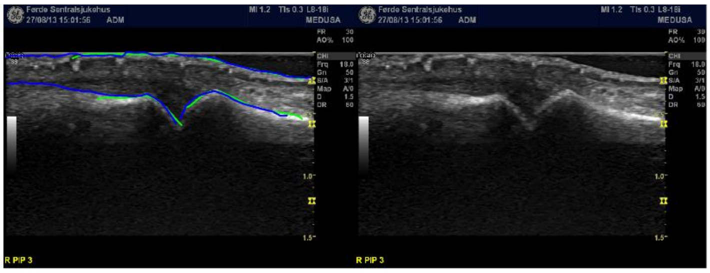
MHD Bone1=5, MHD Bone2=17, MHD Skin=13



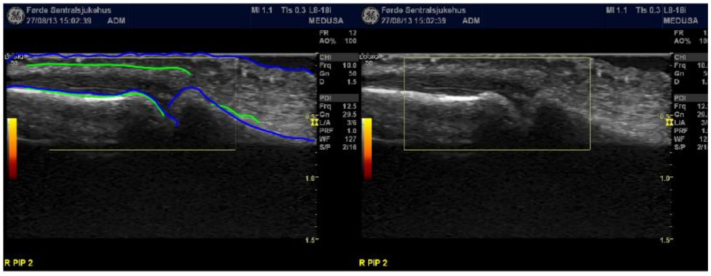
MHD Bone1=20, MHD Bone2=7, MHD Skin=5



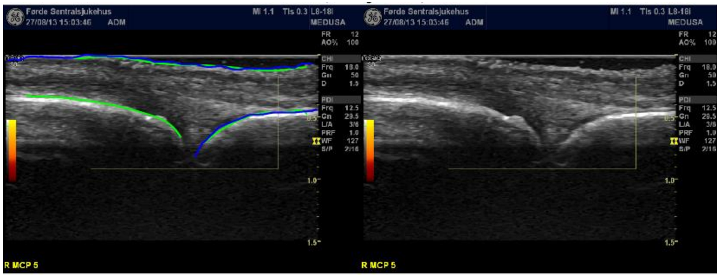
MHD Bone1=4, MHD Bone2=3, MHD Skin=3



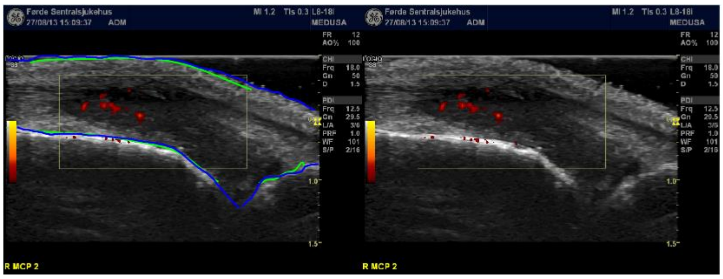
MHD Bone1=6, MHD Bone2=12, MHD Skin=30



MHD Bone1=389, MHD Bone2=3, MHD Skin=4



MHD Bone1=3, MHD Bone2=7, MHD Skin=7

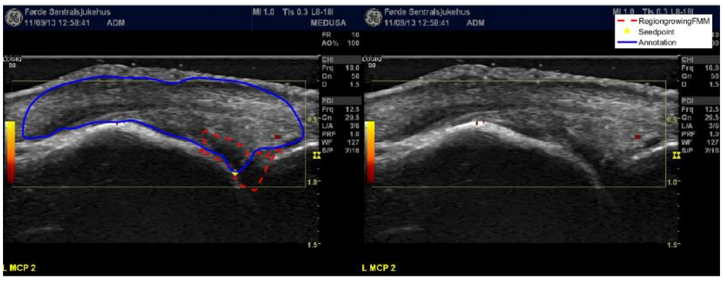


Appendix-B.2 **Additional visual results Experiment 5.2**

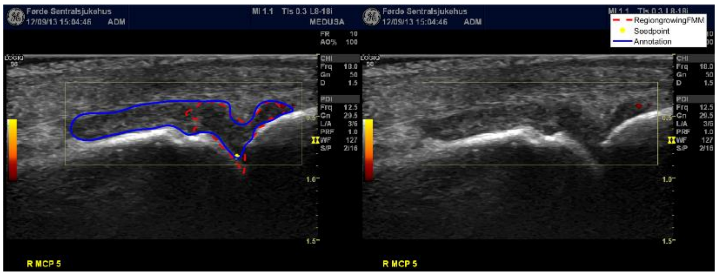
For all the methods, the six first images are from D4 and the last four are from T4D4

Region growing: showing segmented inflammation area in red, seed point in yellow, and annotation in blue.

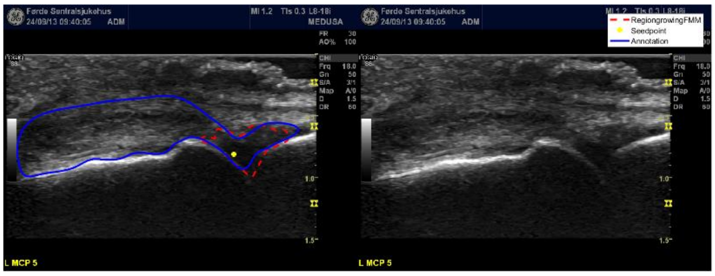
Dice=12



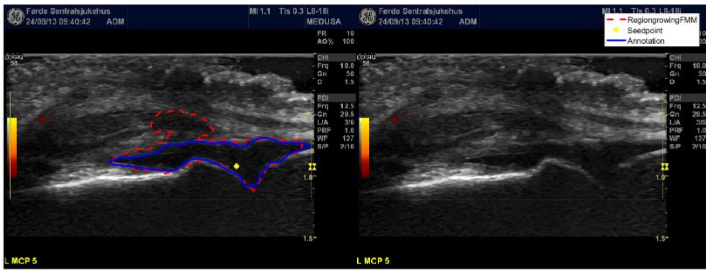
Dice=56



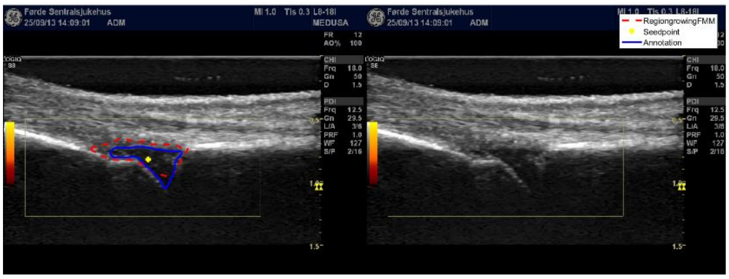
Dice=24



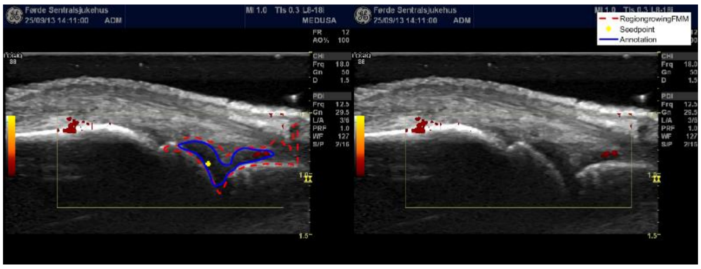
Dice=73



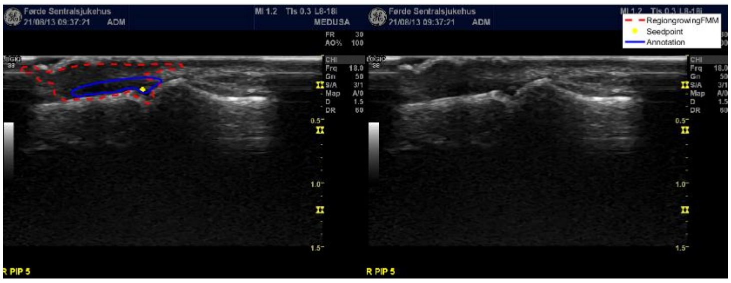
Dice=69



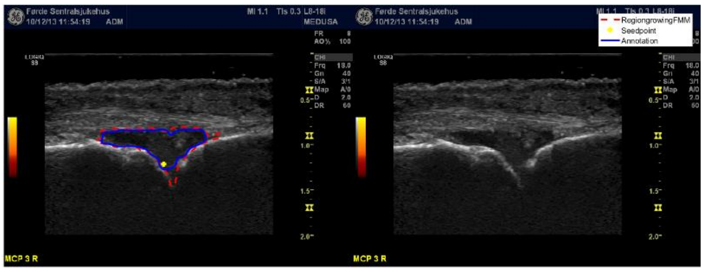
Dice=60



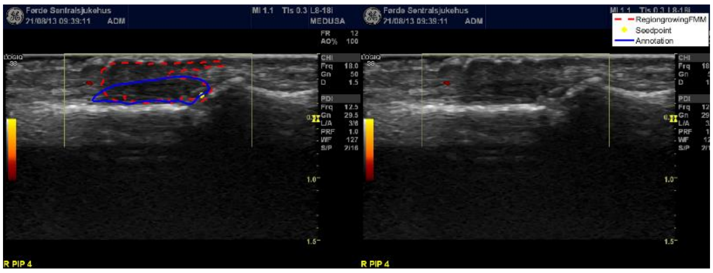
Dice=37



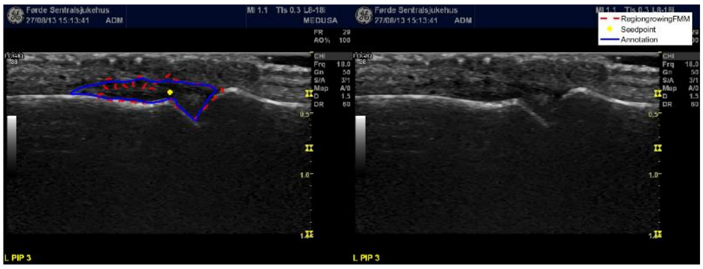
Dice=85



Dice=61

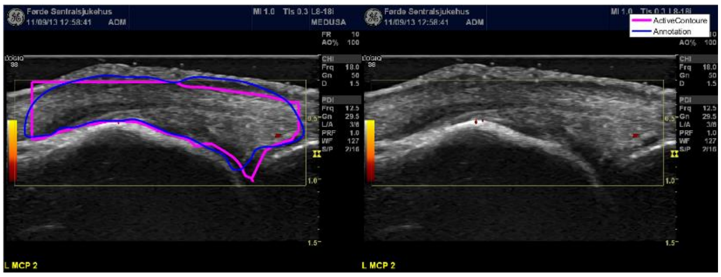


Dice=86

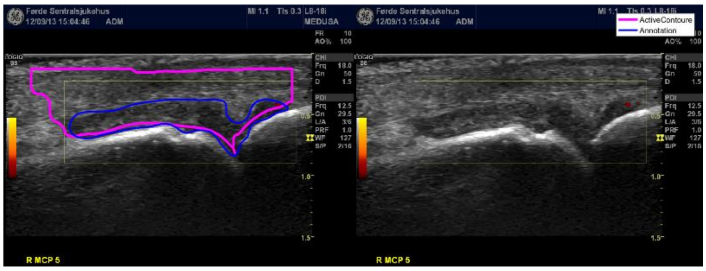


My active contour: showing segmented inflammation area in magenta, and annotation in blue. The “Col5” parameters were used in the examples below.

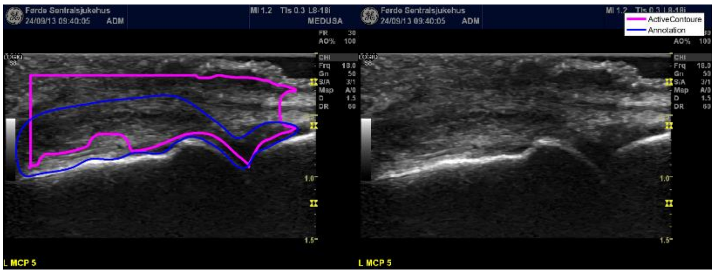
Dice=90



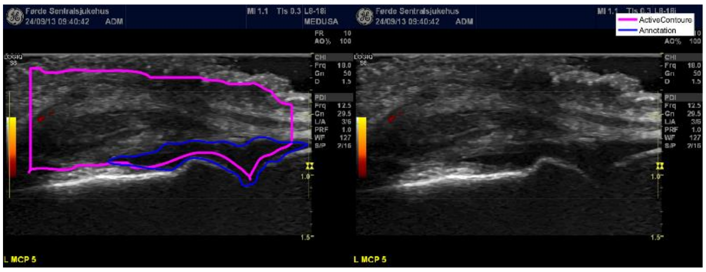
Dice=45



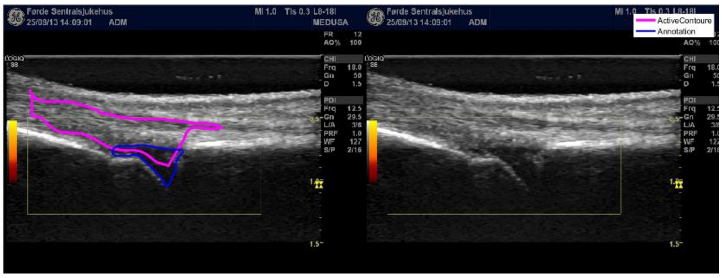
Dice=59



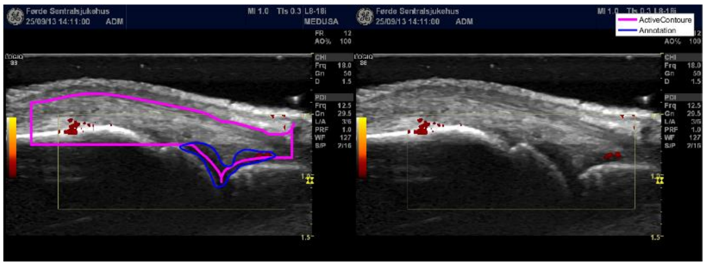
Dice=20



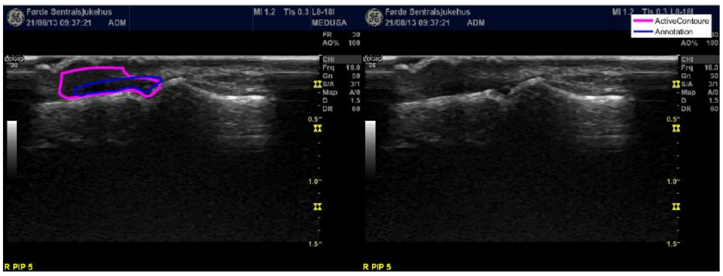
Dice=17



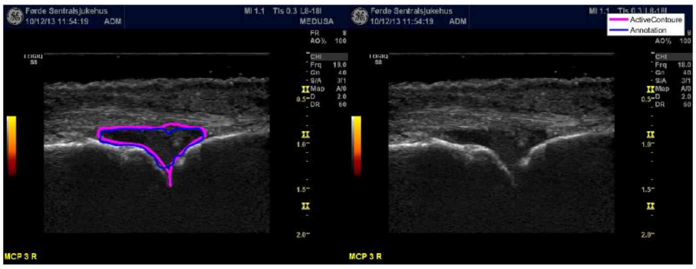
Dice=12



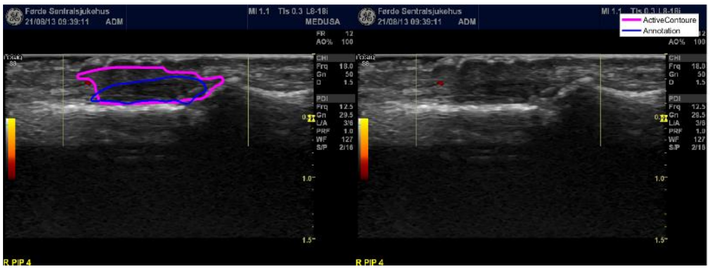
Dice=54



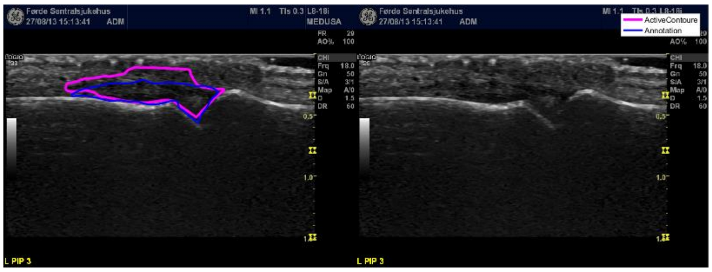
Dice=86



Dice=68

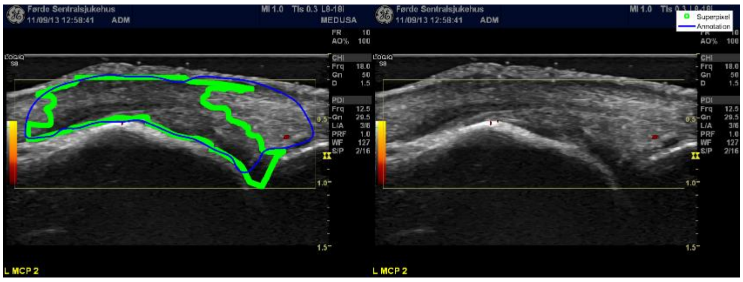


Dice=71

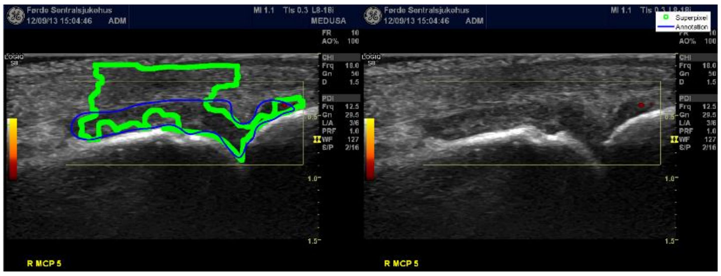


Superpixel: showing segmented inflammation area in green, and annotation in blue. The “new” parameters were used in the examples below.

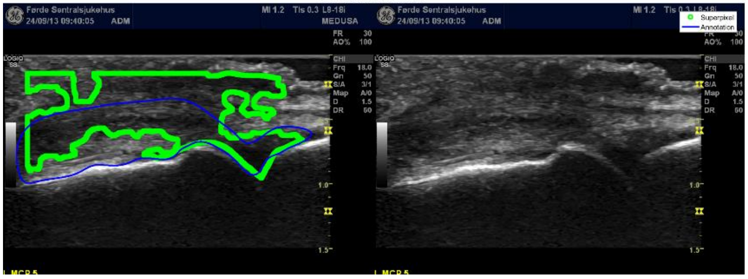
Dice=12



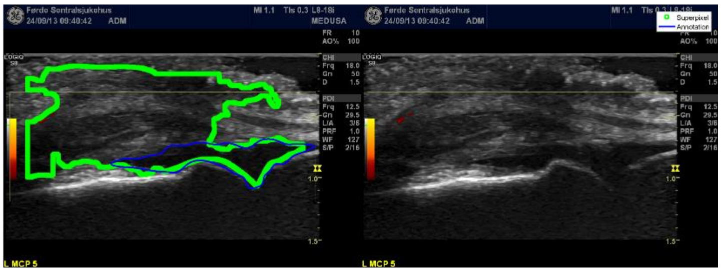
Dice=53



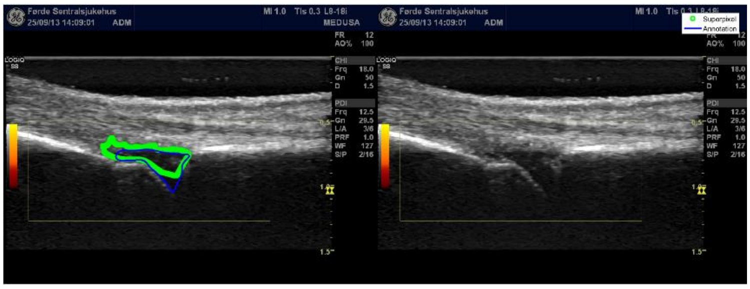
Dice=57



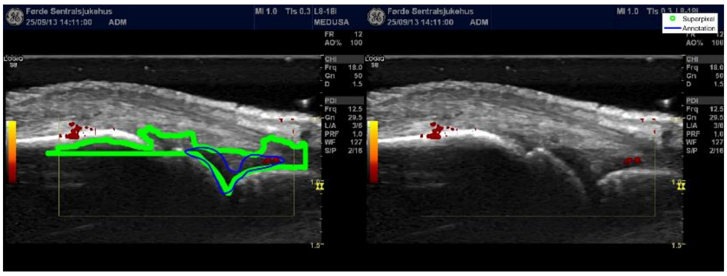
Dice=29



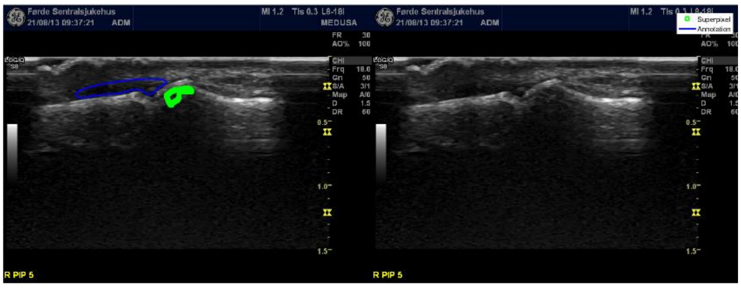
Dice=71



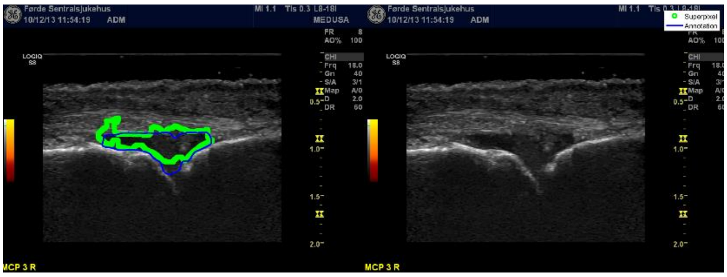
Dice=43



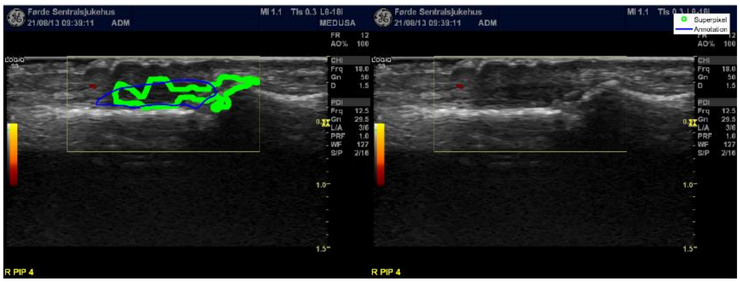
Dice=0



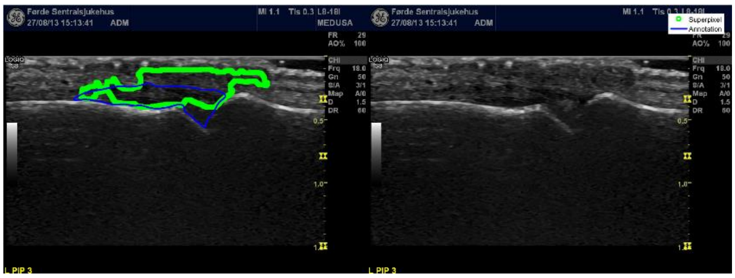
Dice=78



Dice=63



Dice=58



Appendix-C Code

The attached code cannot be runned since I was not allowed to distribute the materials, including the images, from the MEDUSA project.

Appendix-C.1 MATLAB code guide

Code used for Experiment 5: Segmentation of inflammation region

Filename	Description
ResultExp4.2	A folder that containing detected bone and skin features from experiment 4.2
anisodiff2D.m	Anisotropic filter: Used to apply PM-filter
bhp.m	Frequency domain filter: Butterworth High pass
blpf.m	Frequency domain filter: Butterworth Low pass
curvplot.m	Used to make ellipse
EllipsMASK.m	Used to create Elliptic mask using **curvplot.m** as well as locating seed point for region growing
EllipsMASK2.m	Same as above, but only used when running **MainInflamExp52MYAcativeContour.m**
eucleadiandistance.m	Function for calculating Euclidian distance
findShortesEucledianDist.m	Function for calculating min Euclidian distance to extrema points of binary objects using when applying superpixel
ghp.m	Frequency domain filter: Gaussian high pass
hbb.m	Frequency domain filter: High boost Butterworth
hbg.m	Frequency domain filter: High boost Gaussian
homofil.m	Homomorphic filter
MainInflamExp51Activecontour.m	Main run file for Experiment 5.1 using Active contour
MainInflamExp51Regiongrowing.m	Main run file for Experiment 5.1 using Region growing
MainInflamExp52MYAcativeContour.m	Main run file for Experiment 5.2 using Active contour
MainInflamExp52Regiongrowing.m	Main run file for Experiment 5.2 using Region growing
MainInflamExp512Superpixel.m	Main run file for Experiment 5.1 and 5.2
MyRegionGrowMethod.m	Function for applying Region growing on ROI and plotting boundary of the segmented are
MySuperpixelMethod.m	Function for applying Superpixel on ROI and plotting boundary of the segmented area
normalize.m	Function for normalization of the images
progressbar.m	Function for progress bar to see remaining time of the program
SimilarityMeasurementJDI.m	Function for calculating Dice; this function also returns Jaccard

Full code Experiment 5 can be found in the following ZIP-file:

Code used for Experiment 4: Segmentation of skin and bone

Filename	Description
anisodiff2D.m	Anisotropic filter: Used to apply PM-filter
bhp.m	Frequency domain filter: Butterworth High pass
blpf.m	Frequency domain filter: Butterworth Low pass
CSV2Cord.m	For converting annotations to x, and y- coordinates in vector form; used inside **ImportAnnotationDataFunc.m**
findNObjectsBoneOrder.m	Used to locate bones
findNSkinObjectsOrder.m	Used to locate skin
ghp.m	Frequency domain filter: Gaussian high pass
HausdorffDistV2.m	Computing MHD with mentioned restrictions as well as possibility of plotting the measured MHD
hbb.m	Frequency domain filter: High boost Butterworth
hbg.m	Frequency domain filter: High boost Gaussian
homofil.m	Homomorphic filter
ImageNameIndexAndNumberOfAnnotation.mat	Table: All image names, index and number of annotation. This was used as input in **ImportAnnotationDataFunc.m**
ImportAnnotationDataFunc.m	Function for extract annotation data
IndexALLAndType.mat	Table: Indexes of all images with labeled type of image PD(“PD”) or B-mode(“HT”)
IndexC1WInfAType.mat	Table: Indexes of all images with inflammation annotations and labeled type of image PD(“PD”) or B-mode(“HT”)
IndexC2WoInfAType.mat	Table: Indexes of all images without inflammation annotations and labeled type of image PD(“PD”) or B-mode(“HT”)
indexClass1.mat	Table: Indexes of all images with inflammation annotations
indexClass2.mat	Table: Indexes of all images without inflammation annotations
MainBoneSkinExperiment41.m	Main run file for Experiment 4.1
MainBoneSkinExperiment42.m	Main run file for Experiment 4.2
ModHausdorffDist.m	Function for MHD
normalize.m	Function for normalization of the images
progressbar.m	Function for progress bar to see remaining time of the program
rgb2cmyk.m	Function for converting RGB to CMYK
segmentBone.m	Function for segmenting bones using **findNObjectsBoneOrder.m** to locate bones
segmentSkin.m	Function for segmenting skin using **findNSkinObjectsOrder.m** to locate bones

Full code Experiment 4 can be found in the following ZIP-file:

Appendix-C.2 **Code**

Following are links to videos that shows the basics when running

Experiment 4.2 with PM-filter Ca. 6 min

Experiment 5.2 using Region growing Ca.6 min

Some example of the code shown in the video can be found below

ImportAnnotationDataFunc.m

```
function [i,bone1,bone2,joint,inflamation,skin,currentfilename,Class] =
iportDataFromCSV(AnnotationDATA,ImageNameIndexAndNumberOfAnnotations,GetClass,i)

% String to compare
InflamString= 'region inflammatory synovitis degree of inflammation / hyperplasia';
BoneString= 'bone';
JointString= 'joint';
SkinString= 'skin';
bone1=[];
bone2=[];
joint=[];
inflamation=[];
skin=[];
currentfilename={};%  Choosing which class and checks if  inflamation data are present
Class=[];


ImageIndex=table2array(ImageNameIndexAndNumberOfAnnotations(1,2));
NumberOfAnnot=table2array(ImageNameIndexAndNumberOfAnnotations(1,3));
AnnotationDataControl=AnnotationDATA(ImageIndex:ImageIndex+NumberOfAnnot,1); % for string
comparing
YesInflam =
strncmp(InflamString,table2array(AnnotationDataControl(:,1)),length(InflamString));
i;
switch GetClass
    case 'C1' % whith inflamamtion

        if max(YesInflam)~=1
            i=i+1;
        else
            Class=1;

            AnnotationData=AnnotationDATA(ImageIndex:ImageIndex+NumberOfAnnot,:);
            % comparing string
            IndexOfInflamationData = strmatch(InflamString,table2array(AnnotationData(:,1)));
            IndexOfBoneData = strmatch(BoneString,table2array(AnnotationData(:,1)));
            IndexOfSkinData = strmatch(SkinString,table2array(AnnotationData(:,1)));
            IndexOfJointData = strmatch(JointString,table2array(AnnotationData(:,1)));
            if length(IndexOfBoneData)<1 ||length(IndexOfJointData)<1
               ||length(IndexOfSkinData)<1 || length(IndexOfInflamationData)<1
             i=i+1 % i gets returned and utdated so it jumps over current interation if som
                   type of error
            else
                 % converting  CSV coordinates to xy-coordinates
                if length(IndexOfBoneData)>1
```

```
                    bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                    bone2 = CSV2Cord(AnnotationData(IndexOfBoneData(2),3:end));
                else
                    bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                    bone2=[];
                end
                joint = CSV2Cord((AnnotationData(max(IndexOfJointData),3:end)));
                inflamation = CSV2Cord(AnnotationData(max(IndexOfInflamationData),3:end));
                skin = CSV2Cord(AnnotationData(max(IndexOfSkinData),3:end));
                currentfilename=AnnotationData(1,1);% getting imagename
                i=i;
            end
        end

    case 'C2' % without inflamamtion
        if max(YesInflam)~=1
            Class=2;

            AnnotationData=AnnotationDATA(ImageIndex:ImageIndex+NumberOfAnnot,:);

            IndexOfInflamationData = strmatch(InflamString,table2array(AnnotationData(:,1)));
            IndexOfBoneData = strmatch(BoneString,table2array(AnnotationData(:,1)));
            IndexOfSkinData = strmatch(SkinString,table2array(AnnotationData(:,1)));
            IndexOfJointData = strmatch(JointString,table2array(AnnotationData(:,1)));

            if length(IndexOfBoneData)>1
                bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                bone2 = CSV2Cord(AnnotationData(IndexOfBoneData(2),3:end));
            else
                bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                bone2=[];
            end
            joint = CSV2Cord((AnnotationData(max(IndexOfJointData),3:end)));
            inflamation = CSV2Cord(AnnotationData(IndexOfInflamationData,3:end));
            skin = CSV2Cord(AnnotationData(IndexOfSkinData,3:end));
            currentfilename=AnnotationData(1,1);
        else
            i=i+1;
        end
    case 'both'

        if max(YesInflam)~=1
            Class=2;
            AnnotationData=AnnotationDATA(ImageIndex:ImageIndex+NumberOfAnnot,:);
            IndexOfInflamationData = strmatch(InflamString,table2array(AnnotationData(:,1)))
            IndexOfBoneData = strmatch(BoneString,table2array(AnnotationData(:,1)));
            IndexOfSkinData = strmatch(SkinString,table2array(AnnotationData(:,1)));
            IndexOfJointData = strmatch(JointString,table2array(AnnotationData(:,1)));

            if length(IndexOfBoneData)>1
                bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                bone2 = CSV2Cord(AnnotationData(IndexOfBoneData(2),3:end));
            else
                bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                bone2=[];
            end
            joint = CSV2Cord((AnnotationData(max(IndexOfJointData),3:end)));
            inflamation = CSV2Cord(AnnotationData(max(IndexOfInflamationData),3:end));
            skin = CSV2Cord(AnnotationData(IndexOfSkinData,3:end));
            currentfilename=AnnotationData(1,1);

        else
            Class=1;
            AnnotationData=AnnotationDATA(ImageIndex:ImageIndex+NumberOfAnnot,:);
            IndexOfInflamationData = strmatch(InflamString,table2array(AnnotationData(:,1)));
            IndexOfBoneData = strmatch(BoneString,table2array(AnnotationData(:,1)));
            IndexOfSkinData = strmatch(SkinString,table2array(AnnotationData(:,1)));
            IndexOfJointData = strmatch(JointString,table2array(AnnotationData(:,1)));

            if length(IndexOfBoneData)>1
                bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                bone2 = CSV2Cord(AnnotationData(IndexOfBoneData(2),3:end));
            else
                bone1 = CSV2Cord(AnnotationData(IndexOfBoneData(1),3:end));
                bone2=[];
            end
            joint = CSV2Cord((AnnotationData(max(IndexOfJointData),3:end)));
            inflamation = CSV2Cord(AnnotationData(max(IndexOfInflamationData),3:end));
            skin = CSV2Cord(AnnotationData(IndexOfSkinData,3:end));
            currentfilename=AnnotationData(1,1);

        end

end
end
```

Published with MATLAB® R2016b

MainBoneSkinExperiment42.m

```
% Load data
ReferenceImage=imread('2014-05-30-S0038-T0044.png');
load('AnnotationData.mat');
load('ImageNameIndexAndNumberOfAnnotation.mat'); data from csv-file
load('indexClass1.mat') % index of images with inflammation annotations
load('indexClass2.mat') % index of images without inflammation annotations
WInflamation=IndexOfClass1AndNumberOfAnnotations;
NoInflamation=IndexOfClass2AndNumberOfAnnotations;
AllImageIndex=ImageNameIndexAndNumberOfAnnotations;
%----------------------------------------------------------------
% Indexes with  Power doppler and B-mode
% % load('IndexALLAndType.mat');
% load('IndexC1WInfAType.mat')
% % load('IndexC2WoInfAType.mat')
% WInflamation=IndexC1WInfAType;
% WIflHT= strmatch('HT',table2array(WInflamation(:,4)));
% WIflPD= strmatch('PD',table2array(WInflamation(:,4)));

%  NoInflamation=IndexC2WoInfAType;
%  AllImageIndex=IndexALLAndType;

% Extracting  Power doppler or only B-mode

% IndexHypTropNoinflam=strmatch('HT',table2array((NoInflamation(:,4))));
% IndexPowerDopplerNoinflam=strmatch('PD',table2array((NoInflamation(:,4))));
%
% IndexHypTropWInlfam=strmatch('HT',table2array((WInflamation(:,4))));
% IndexPowerDopplerWInlfam=strmatch('PD',table2array((WInflamation(:,4))));
%
% IndexHypTropALL=strmatch('HT',table2array((AllImageIndex(:,4))));
% IndexPowerDopplerALL=strmatch('PD',table2array((AllImageIndex(:,4))));

PartOfImage= {}';
ModifiedHusdorfDistance=[]';
HausdorfDistance = []';
HausdorfDistanceTowardsJoint=[]';
IndexofImage=[]';
MyBone1={}';
MyBone2={}';
MySkin={}';

AnnoBone1={}';
AnnoBone2={}';
AnnoSkin={}';
AnnoInflam={}';
MYANNDATA=table(IndexofImage,ImageName,MyBone1,MyBone2,MySkin,AnnoBone1,AnnoBone2,AnnoSkin,Ann
oInflam);
%
HusdorfTable=
table(PartOfImage,HausdorfDistance,ModifiedHusdorfDistance,HausdorfDistanceTowardsJoint);
ErrorTableImage=table(ImageName,IndexofImage);

% Parameter for filtering and cases

filterMetode=1; %[PM,HomoButterworth,directGausiansmoth]
```

```
EqualizedMethod=1; %[none,hisEqBasedonRef,CLAHER,CLAHEU]
GetClass='C1'; %['C1';'C2';'both'] % C1 = with inflamamtion
num_iter =70;
delta_t = 1/4;
kappa =60;
% opotion : which of the diffusion coefficent that should be used
option = 2; % if 1 privileges high-contrast edges over low-contrast ones.
% if 2  privileges wide regions over smaller ones.
```

```
% Starting loop for program

% inputdata can be: AllImageIndex, NoInflamation or  WInflamation
% InnputData=WInflamation(IndexPowerDopplerWInlfam,:);
InnputData=WInflamation;

LOOPLENGHT=2;
testimages=[1,466,1201,1701,33]; % Visual tuning
index=testimages;
timerVal = tic;

for i= 2:2
    try
        % Using  novel method for extracting  annotation data
        [iout,bone1,bone2,joint,inflamation,skin,currentfilename,class] =...
            ImportAnnotationDataFunc(AnnotationDATA,InnputData(testimages(i),:),GetClass,i);

    catch ME
        ErrorTableImage=table(ImageName);
        continue
    end
    if i==iout

        i

        %  progressbar
           progressbar(i/LOOPLENGHT)  % end
```

```
       % different cropping of the image
        ROIposBone2=[35.51 141.510 795.980 495.980];
        ROIPOS3=[52.51 144.51 766.98 371.98];
        ROIpositionBone =[61.00 137.00 755.00 351.00];

        ROIposBone2=ROIpositionBone;
        currentfilename1=table2array(currentfilename{1,1});
        try
            ImageOriginal=imread(strcat('C:\Users\Christian\Google Drive\MASTER\all-
2017\Data\',currentfilename1,'.png'));
        catch ME
            continue
        end
```

```
  %  Starting enhancement

        %Removing yellowchannel

        % RGB--> CMY-->Grayscale
        % figure
        % imshow(ImageOriginal(:,:,1));
        CMYK = rgb2cmyk(ImageOriginal);
        % Black = CMYK(:,:,4);
        % Cyan = CMYK(:,:,1);
        % Magenta =  CMYK(:,:,2);
        Yellow =  CMYK(:,:,3);
        % imshow(ImageOriginal)

        NewUYOriginal=ImageOriginal.*(1-Yellow); %removing the yellow box


        ImageOriginal=NewUYOriginal;
        ImageOriginalGray=rgb2gray(ImageOriginal);

        ImgGrayEqRef=histeq(ImageOriginalGray,imhist(rgb2gray(ReferenceImage)));



        ImageGrayCroped= imcrop(ImageOriginalGray,ROIposBone2);
        ImageGrayCropedEqRef= uint8(255*mat2gray((imcrop(ImgGrayEqRef,ROIposBone2))));

        % CLAHE
AdaptHistCropped=adapthisteq(ImageGrayCroped,'clipLimit',0.01,'Distribution','rayleigh','Alpha
',0.4);

AdaptHistCropped2=adapthisteq(ImageGrayCroped,'clipLimit',0.01,'Distribution','uniform');

% Choice of Equlazing method
        switch EqualizedMethod
            case  1
                CroppedImage= ImageGrayCroped;
            case 2
                CroppedImage= ImageGrayCropedEqRef;
            case 3
                CroppedImage= AdaptHistCropped;
            case 4
                CroppedImage= AdaptHistCropped2;
        end

        % Preprossering:  Filtreing
        switch filterMetode
            case 1

                % PM-filter
          ImGrayAniDiffFIlter=anisodiff2D(gather(CroppedImage),num_iter,delta_t,kappa,option);
                seToHt = strel('disk',9);
                II=gather(ImGrayAniDiffFIlter);
                tophatFiltered = imtophat(II,seToHt);
                ImadjustTopH=imadjust(tophatFiltered);
            case 2
```

```
         case 3
             %Guasian smoothing filter

             seToHt2 = strel('disk',20);

             CroppedImage;

             sigma=1.8;
             Gus=imgaussfilt(CroppedImage,sigma);
             SmothGaushigpass=(Gus);
             for ii=1:30
                 Gus=imgaussfilt(SmothGaushigpass,sigma);
                 SmothGaushigpass=(Gus);

             end

             tophatFiltered2 = imtophat(SmothGaushigpass,seToHt2);
             ImadjustTopH=imadjust(tophatFiltered2);


     end
```

```
% Segmenting

     try
        %  using segmenting my segmenting and locating method for skin and bone
         wishedObjectBone= segmentBone(ImadjustTopH,0,40);
         wishedObjectSkin= segmentSkin(ImadjustTopH,0,40);
     catch ME
         ErrorTableImage=table(ImageName);
         continue
     end

     SkinandBone=(gather(gpuArray(wishedObjectSkin+wishedObjectBone)));
     SkinandBone= imfill(SkinandBone,'holes');
     LSuper = bwlabel(SkinandBone);
     statS = regionprops('table',LSuper,'Centroid');
     SkinandBone = bwareaopen(SkinandBone,500); %
     if length(table2array(statS))~=3
         SkinandBone=bwmorph(SkinandBone,'erode',1); % 4, 2
         SkinandBone=bwmorph(SkinandBone,'thin',1);
         SkinandBone=bwmorph(SkinandBone,'majority',2);
     end
```

```
     % Boundary description bone and skin
     warning('off', 'MATLAB:polyfit:RepeatedPointsOrRescale')
     figure('Name','Comparison with annotations ','visible','on');
     imshow(ImageOriginal);hold on
     Lable={'skin','bone1','bone2','','','','','',''};
     colors=['b' 'g' 'r' 'c' 'm' 'y'];
     Boundarypixellocation={};
     [BoundarypixellocationB,LabelMatrix1,NumberofObjects1,A1] =
bwboundaries(wishedObjectBone,'noholes');
        [BoundarypixellocationS,LabelMatrix2,NumberofObjects2,A2] =
bwboundaries(wishedObjectSkin,'noholes');

Boundarypixellocation={BoundarypixellocationS{1};BoundarypixellocationB{1};Boundarypixellocati
onB{2}};
        for j=1:length(Boundarypixellocation)
            boundary = Boundarypixellocation{j};
            sort(boundary(:,2),'ascend'); % sorting in ascending order based on y-coord
            lenght=1:round(length(boundary)/2);
            lenght2=round(length(boundary)/2):round(length(boundary));

            poltFUpper=polyfit(boundary(lenght,2),boundary(lenght,1),7); % polyfiting upper
            poltFLower=polyfit(boundary(lenght2,2),boundary(lenght2,1),7); % polyfiting lower
            cidx = mod(j,length(colors))+1;
            if j==1
                B = boundary((1:round(length(boundary)/2)),:); % minimum half of boundary
                B2= boundary(round(length(boundary)/2):round(length(boundary)),:);

                %Upper skin
                UpperSkin=[B(:,2)+ROIposBone2(1),polyval(poltFUpper,B(:,2))+ROIposBone2(2)];
                plot(UpperSkin(:,1),UpperSkin(:,2),'m','LineWidth',2);

                %Finds end point

               UpperConnectionLineSkin=[UpperSkin(1,1),UpperSkin(1,2)];

               plot(UpperConnectionLineSkin(1),UpperConnectionLineSkin(2),'bo','LineWidth',3);


                %%Lower skin
                LowerSkin=[B2(:,2)+ROIposBone2(1),polyval(poltFLower,B2(:,2))+ROIposBone2(2)];
                plot(LowerSkin(:,1),LowerSkin(:,2),'g','LineWidth',2);


                 %Finds end point
                LowerConnectionLineSkinLeft=[LowerSkin(end,1),LowerSkin(end,2)];
       plot(LowerConnectionLineSkinLeft(1),LowerConnectionLineSkinLeft(2),'ro','LineWidth',3);

                LowerConnectionLineSkinRight=[LowerSkin(1,1),LowerSkin(1,2)];

plot(LowerConnectionLineSkinRight(1),LowerConnectionLineSkinRight(2),'yo','LineWidth',3);

                MidlelegthSkin=round(length(LowerSkin)/2);

LowerConnectionLineSkinMidle=[max(LowerSkin(MidlelegthSkin,1)),LowerSkin(MidlelegthSkin,2)];

plot(LowerConnectionLineSkinMidle(1),LowerConnectionLineSkinMidle(2),'co','LineWidth',3);


            elseif j==2
                B = boundary((1:round(length(boundary)/2)),:);
                B2= boundary(round(length(boundary)/2):round(length(boundary)),:);
                %         Upper bone
                UpperBone1=[B(:,2)+ROIposBone2(1),polyval(poltFUpper,B(:,2))+ROIposBone2(2)];
                        plot(UpperBone1(:,1),UpperBone1(:,2),'g','LineWidth',2);
                UpperBoneT1=[B(:,2)+ROIposBone2(1),B(:,1)+ROIposBone2(2)];


                %Finds end points
                UpperConnectionLineBone1Left=[UpperBone1(1,1),UpperBone1(1,2)];

plot(UpperConnectionLineBone1Left(1),UpperConnectionLineBone1Left(2),'ro','LineWidth',3);


                UpperConnectionLineBone1Right=[UpperBone1(end,1),UpperBone1(end,2)];

plot(UpperConnectionLineBone1Right(1),UpperConnectionLineBone1Right(2),'yo','LineWidth',3);

                MidlelegthUpperbone1=round(length(UpperBone1)/2);

UpperConnectionLineBone1Midle=[max(UpperBone1(MidlelegthUpperbone1,1)),UpperBone1(MidlelegthUp
perbone1,2)];

plot(UpperConnectionLineBone1Midle(1),UpperConnectionLineBone1Midle(2),'co','LineWidth',3);

                %Bon down side
                LowerBone1 =
[B2(:,2)+ROIposBone2(1),polyval(poltFLower,B2(:,2))+ROIposBone2(2)];
                        plot(LowerBone1(:,1),LowerBone1(:,2),'m','LineWidth',2);
                %Finds end point
                LowerConnectionLineBone1=[LowerBone1(1,1),LowerBone1(1,2)];

plot(LowerConnectionLineBone1(1),LowerConnectionLineBone1(2),'bo','LineWidth',3);

            elseif j==3
                B = boundary((1:round(length(boundary)/2)),:); % minste halvded av boundary
                 % basert på Y
                B2= boundary(round(length(boundary)/2):round(length(boundary)),:);

                % upper bone
                UpperBone2=[B(:,2)+ROIposBone2(1),polyval(poltFUpper,B(:,2))+ROIposBone2(2)];
                        plot(UpperBone2(:,1),UpperBone2(:,2),'g','LineWidth',2);


                 %Finds end points

                UpperConnectionLineBone2Left=[UpperBone2(1,1),UpperBone2(1,2)];

plot(UpperConnectionLineBone2Left(1),UpperConnectionLineBone2Left(2),'bo','LineWidth',3);

                UpperConnectionLineBone2Right=[UpperBone2(end,1),UpperBone2(end,2)];

plot(UpperConnectionLineBone2Right(1),UpperConnectionLineBone2Right(2),'yo','LineWidth',3);

                MidlelegthUpperbone2=round(length(UpperBone2)/2);

UpperConnectionLineBone2Midle=[max(UpperBone2(MidlelegthUpperbone2,1)),UpperBone2(MidlelegthUp
perbone2,2)];

plot(UpperConnectionLineBone2Midle(1),UpperConnectionLineBone2Midle(2),'co','LineWidth',3);

                %Bone lower bone
                LowerBone2 = [B2(:,2)+ROIposBone2(1),
polyval(poltFLower,B2(:,2))+ROIposBone2(2)];
                        plot(LowerBone2(:,1),LowerBone2(:,2),'m','LineWidth',2);

                %Finds end point
                LowerConnectionLineBone2=[LowerBone2(end,1),LowerBone2(end,2)];

plot(LowerConnectionLineBone2(1),LowerConnectionLineBone2(2),'ro','LineWidth',3);




            end
                rndRow = ceil(length(boundary)/(mod(rand*j,7)+1));
                col = boundary(rndRow,2)+ROIposBone2(1); row =
boundary(rndRow,1)+ROIposBone2(2);
                h = text(col+1, row-1, Lable{j});
                set(h,'Color','g','FontSize',14,'FontWeight','bold');

        end

        %  plotting annotation data if it exist
        if length(bone1(:,1))==2
            plot(bone1(1,:),bone1(2,:),...
                'b','LineWidth',2);
        end
        if length(bone2(:,1))==2
            plot(bone2(1,:),bone2(2,:),...
                'b','LineWidth',2);
        end
        if length(skin(:,1))==2
            plot(skin(1,:),skin(2,:),...
                'b','LineWidth',2);
        end
        if length(joint(:,1))==2
            plot(joint(1,:),joint(2,:),...
                'w','LineWidth',2);
        end
        if length(inflamation(:,1))==2
            plot(inflamation(1,:),inflamation(2,:),...
                'r','LineWidth',2);
        end
```

```
        % addding annotations ro table for use in experiment 5
        AnnoBone1(end+1,:)={bone1};
        AnnoBone2(end+1,:)={bone2};
        AnnoSkin(end+1,:)={skin};
        AnnoInflam(end+1,:)={inflamation};


        MyBone1(end+1,:)=[{UpperBone1},{LowerBone1}];
        MyBone2(end+1,:)=[{UpperBone2},{LowerBone2}];
        MySkin(end+1,:)=[{UpperSkin},{LowerSkin}];
        ImageName(end+1,:)= [{currentfilename1}]';
        IndexofImage(end+1,:)= [i]';
        %Similarity measurement of Bones and skin

figure('Name','Similarity measurment bone and skin','visible','off');
        try
            [HausdorfDistanceBone1, D1 ModHausdorfDistanceBone1] =
HausdorffDistV2(UpperBone1,bone1,[],'vis');
            [HausdorfDistanceBone2 D2 ModHausdorfDistanceBone2 ] =
HausdorffDistV2(UpperBone2,bone2,[],'vis');
            [HausdorfDistanceSkin D3 ModHausdorfDistanceSkin ] =
HausdorffDistV2(UpperSkin,skin,[],'vis');
        catch ME
            ErrorTableImage=table(ImageName);
             % continue
        end


       title({['Bone1: Hausdorff Distance = ' num2str(HausdorfDistanceBone1),' and Modified
Hausdorff Distance = ',num2str(ModHausdorfDistanceBone1)]...
            ['Bone2: Hausdorff Distance = ' num2str(HausdorfDistanceBone2),' and Modified
Hausdorff Distance = ',num2str(ModHausdorfDistanceBone2)]...
            ['Skin: Hausdorff Distance = ' num2str(HausdorfDistanceSkin),' and Modified
Hausdorff Distance = ',num2str(ModHausdorfDistanceSkin)]});
        hold off



HausdorfDistance(end+1,:)=[HausdorfDistanceBone1,HausdorfDistanceBone2,HausdorfDistanceSkin]';


ModifiedHusdorfDistance(end+1,:)=[ModHausdorfDistanceBone1,ModHausdorfDistanceBone2,ModHausdor
fDistanceSkin]';
        PartOfImage(end+1,:)={'Bone1','Bone2','Skin'}';




    else
        continue
        ErrorTableImage=table(ImageName);
    end
end

```

```
% loop is done and additional results is computed


timlapsed=toc(timerVal)/60
progressbar(1)

MYANNDATA=table(IndexofImage,ImageName,MyBone1,MyBone2,MySkin,AnnoBone1,AnnoBone2,AnnoSkin,Ann
oInflam);
HusdorfTable= table(PartOfImage,HausdorfDistance,ModifiedHusdorfDistance);

STD=[std(table2array(HusdorfTable(:,2:end)))]';
MAX=[max(table2array(HusdorfTable(:,2:end)))]';
MIN=[min(table2array(HusdorfTable(:,2:end)))]';
MEAN=[mean(table2array(HusdorfTable(:,2:end)))]';
MEDIAN=[median(table2array(HusdorfTable(:,2:end)))]';
```

```
ArraModHaus=(table2array(HusdorfTable(:,3)));
PERCENU20=[];
for ii=1:3
    CountU20=find((ArraModHaus(:,ii))<20);
    sizeCountU20=size(CountU20);
    PERCENU20(ii,:)=(sizeCountU20(1)/length((ArraModHaus(:,2))))*100;
end
PERCENU20=[[0 0 0]';PERCENU20];
ResulltTableHausdorf=table(STD,MEAN,MEDIAN,MAX,MIN,PERCENU20);
% writetable(ResulltTableHausdorf,'ResultExp4.2\PM\ResulltTableHausdorfFinal.xlsx')
% save('ResultExp4.2\PM\PMFinal.mat','HusdorfTable');
% save('ResultExp4.2\PM\AnnotationMyPMALL.mat','MYANNDATA');
```

Published with MATLAB® R2016b

MainInflamExp52Regiongrowing.m

```
% Testdata T4D4-D6
TESTDATAHausExp4=load('ResultExp4.2\PM\Exp4Testdata\PMTestdata.mat');
MYannotationExp4=load('ResultExp4.2\PM\Exp4Testdata\AnnotationMyTestdata.mat');
% D4
% TESTDATAHausExp4=load('ResultExp4.2\PM\FINAL42\PMFinal2.mat');
% MYannotationExp4=load('ResultExp4.2\PM\FINAL42\AnnotationMyPMALL2.mat');
%
TESTDATAHausExp4Haus= TESTDATAHausExp4.HusdorfTable;
GOODMYANNOArray=table2array(TESTDATAHausExp4Haus(:,3));
% Extracting only bone and skin features where all have MHD<20
IDXAllGoodLess20=find(GOODMYANNOArray(:,1)<20&GOODMYANNOArray(:,2)<20&GOODMYANNOArray(:,3)<20)
;


% Table for similarity measurment
MethodNumber=[];Methode={};DiceSimCoeff=[];JaccardSimCoeff=[];
JDIRegiongrowing= table(MethodNumber,Methode,DiceSimCoeff,JaccardSimCoeff);


timerVal = tic;
LOOPLENGHT=numel(IDXAllGoodLess20);

for i= 2:2
    %     progressbar((i)/LOOPLENGHT)  % end
    i
    IDXFIND= IDXAllGoodLess20(i);
    % extracting  annotations and  bone and skin features from Experiment 4 using IDXFIND
    Mybone1UP=MYannotationExp4.MYannotationExp4.MYANNDATA.MyBone1{IDXFIND,1};
    Mybone2UP=MYannotationExp4.MYannotationExp4.MYANNDATA.MyBone2{IDXFIND,1};
    MyskinUP=MYannotationExp4.MYannotationExp4.MYANNDATA.MySkin{IDXFIND,1};
    Mybone1LOW=MYannotationExp4.MYannotationExp4.MYANNDATA.MyBone1{IDXFIND,2};
    Mybone2LOW=MYannotationExp4.MYannotationExp4.MYANNDATA.MyBone2{IDXFIND,2};
    MyskinLOW=MYannotationExp4.MYannotationExp4.MYANNDATA.MySkin{IDXFIND,2};

    TBone1=MYannotationExp4.MYannotationExp4.MYANNDATA.AnnoBone1{IDXFIND,1}';
    TBone2=MYannotationExp4.MYannotationExp4.MYANNDATA.AnnoBone2{IDXFIND,1}';
    TSkin=MYannotationExp4.MYannotationExp4.MYANNDATA.AnnoSkin{IDXFIND,1}';
    TInflamation=MYannotationExp4.MYannotationExp4.MYANNDATA.AnnoInflam{IDXFIND,1}';

    % Input image
    currentfilename1=MYannotationExp4.MYannotationExp4.MYANNDATA.ImageName{IDXFIND};
    ImageOriginal=imread(strcat('C:\Users\Christian\Google Drive\MASTER\all-
2017\Data\',currentfilename1,'.png'));

    ImageOriginalGray=rgb2gray(ImageOriginal);
    %  Defining  names for the  segmented bone and skin features


    %  bone1

    UpperConnectionLineBone1Left=[Mybone1UP(1,1),Mybone1UP(1,2)];

    UpperConnectionLineBone1Right=[Mybone1UP(end,1),Mybone1UP(end,2)];
    LowerConnectionLineBone1Right=[Mybone1LOW(1,1),Mybone1LOW(1,2)];

    % bone2

    UpperConnectionLineBone2Left=[Mybone2UP(1,1),Mybone2UP(1,2)];

    LowerConnectionLineBone2Left=[Mybone2LOW(end,1),Mybone2LOW(end,2)];

    UpperConnectionLineBone2Right=[Mybone2UP(end,1),Mybone2UP(end,2)];

    % % skin
    LowerConnectionLineSkinLeft=[MyskinLOW(end,1),MyskinLOW(end,2)];

    LowerConnectionLineSkinRight=[MyskinLOW(1,1),MyskinLOW(1,2)];
```

```
    % Masking for region growing ): Finding ROI between bone and skin
    dicplaceBottom=50;
    displacementBone2M2=60;
    heightside=80;
    CmaXRight=810; %% max l
    CmaXLeft=60;
    % finds a point between the bones and displace it to get a bottomjoint point
    XBottom= (UpperConnectionLineBone1Right(1)+UpperConnectionLineBone2Left(1))/2;

YBottom=UpperConnectionLineBone1Right(2)+UpperConnectionLineBone2Left(2))/2+dicplaceBottom;
    BottomJoint=[XBottom,YBottom]; % BJP point
    ImageOriginalGray1=rgb2gray(ImageOriginal);
```

```
    % Making  the  polygon coordinates for the mask
    % y-coordinates
RowY1=[UpperConnectionLineBone1Left(2),UpperConnectionLineBone1Left(2),Mybone1UP(:,2)',UpperConnectionLineBone1Right(2),BottomJoint(2)...
UpperConnectionLineBone2Left(2),Mybone2UP(:,2)',UpperConnectionLineBone2Right(2),UpperConnectionLineBone2Right(2)...
LowerConnectionLineSkinRight(2),LowerConnectionLineSkinRight(2),MyskinLOW(:,2)',LowerConnectionLineSkinLeft(2),LowerConnectionLineSkinLeft(2)...
        UpperConnectionLineBone1Left(2)];

   % x-coordinates
CollumnX1=[CmaXLeft,UpperConnectionLineBone1Left(1),Mybone1UP(:,1)',UpperConnectionLineBone1Right(1),BottomJoint(1)...
UpperConnectionLineBone2Left(1),Mybone2UP(:,1)',UpperConnectionLineBone2Right(1),CmaXRight...
CmaXRight,LowerConnectionLineSkinRight(1),MyskinLOW(:,1)',LowerConnectionLineSkinLeft(1),CmaXLeft,CmaXLeft];

    % Creating the MASK using RowY1 og CollumnX1
    InflamImageMASK = roipoly(ImageOriginalGray,CollumnX1,RowY1);
```

```
    % Preprocessing of masked region
    imageForClassifying=ImageOriginalGray;
    newimage=double(imageForClassifying).*double(InflamImageMASK);
    newimage= uint8(255 * mat2gray(newimage)); % for showing unfiltered region

    % Filtering
    num_iter2 =20;
    delta_t2 = 1/4;
    kappa2 = 1000;
    option2 = 1; % if 1 privileges high-contrast edges over low-contrast ones.
    cutofrequens=50;
    FilterOrder= 1;
    typeF=4;

    AnisDif = @(x1) anisodiff2D(x1,num_iter2,delta_t2,kappa2,option2);
    newFilteredImage = roifilt2(newimage,InflamImageMASK,AnisDif);
    %                     figure;imshow(newFilteredImage);title('PM-filter')

    HomomoficFilteredHigboost=@(x2) homofil(x2,cutofrequens,0,FilterOrder,typeF); %Butterworth

  HomButtshigpass=roifilt2(double(newFilteredImage),InflamImageMASK,HomomoficFilteredHigboost);
    NorMIMG=uint8(normalize(HomButtshigpass,255,0));
    newFilteredImage=NorMIMG;

    FilteredROIPixelVector=(newFilteredImage(newFilteredImage>1));
    FilteredROIPixelVectorDouble=double(FilteredROIPixelVector);
    ImageDoubleFilt = double(newFilteredImage)/255;
```

```
    % My thresholding method Algorithm 4
    LabeledMASK= bwlabel(InflamImageMASK);
    AMask = regionprops(LabeledMASK,'Area');
    AreaMASK = max(cat(1, AMask.Area));
    AreaMASKM=AreaMASK(1,1); % finding the size of the mask
    LowerBound = double(round(0.01*AreaMASKM)); % LowerBound 1% of Area of MASK

    MinPixelvalue=min([FilteredROIPixelVectorDouble(:,1)]');
    MaxPixelvalue=max([FilteredROIPixelVectorDouble(:,1)]');
    thresfactor= (MinPixelvalue*10/MaxPixelvalue);

    ThresForInflam=30+(30*thresfactor); % Definding the threshold
    % Binary image or Threshimage
    thrshImag=(newFilteredImage<ThresForInflam & newFilteredImage>MinPixelvalue);

    thrshImag=bwareaopen(thrshImag,LowerBound);
    thrshImagL= bwlabel(thrshImag) ; % labeling of image for extractring regionpors


    try
   % Finding the seed point using creating the eliptic Mask
 [EMasK,SeedpointT]=EllipsMASK(UpperConnectionLineBone1Right(1),UpperConnectionLineBone1Right(2
 ),80,40,ImageDoubleFilt,ThresForInflam/255,(0),1,'');
        SeedX=SeedpointT(1);
        SeedY=SeedpointT(2);
    catch
        continue
    end
      Try
```

```
  % Using graydiffwiight and imsegfmm to do region growing
    IntensityDistance = 0.01; % threst=0.005
```

```
        [InflamRegiongrow,BWInternRG]= MyRegionGrowMethod((ImageDoubleFilt),...
            InflamImageMASK,SeedX,SeedY,IntensityDistance,25,'');
        hold off
```

```
    catch
        continue
    end
    % Similarity measurement for inflamation region

    % Making the annotate region a binary object
    InflamRegionOrg=roipoly(ImageDoubleFilt,TInflamation(:,1),TInflamation(:, 2));

    SE6 = strel('disk',50);

    % InflamRegionExtern=imclose(BWExternRG,SE6);
    InflamRegionRG=imclose(BWInternRG,SE6);

JDIRegiongrowing(end+1,:)=similarityMeasurementJDI(InflamRegionOrg,InflamRegionRG,{'Methode'}'
,2);

    % Just a way to plot the inflamation boundary and image
    if i<30
        warning('off', 'Images:initSize:adjustingMag')
        h2=figure('Name','Region growing FMM','visible','off');
        JDIString = sprintf('Dice= %d',round(JDIRegiongrowing.DiceSimCoeff(end));
        imshowpair(ImageOriginal,ImageOriginal,'montage');title(JDIString);
        hold on
        plot(InflamRegiongrow(:,2),InflamRegiongrow(:,1),'--r','LineWidth',2);
        plot(SeedX,SeedY,'y+','LineWidth',3);

        plot(TInflamation(:,1),TInflamation(:, 2),'b','LineWidth',2);
        legend('RegiongrowingFMM','Seedpoint','Annotation')
        hold off

        baseFileName1 = sprintf('PlotBonandSkin%d.png',i);
        fullFileName1 = fullfile('IMPLOT', baseFileName1);
        saveas(h2,fullFileName1,'png')
    end

end
timlapsed=toc(timerVal)/60
% Similarity measurement
JDITable=[JDIRegiongrowing];

STD=[std(table2array(JDITable(:,3:end)))]';
MAX=[max(table2array(JDITable(:,3:end)))]';
MIN=[min(table2array(JDITable(:,3:end)))]';
MEAN=[mean(table2array(JDITable(:,3:end)))]';
MEDIAN=[median(table2array(JDITable(:,3:end)))]';

ResulltJDITable=table(STD,MEAN,MEDIAN,MAX,MIN);
```

Published with MATLAB® R2016b

MyRegionGrowMethod.m

```
function [InflamIntern,BW33]= MyRegionGrowMethod(Im,MASK,SX,SY,IntDist,Cutoff,ShowFigureOrNot)
% Fuction used  for region growing in Experiment 5.1 and 5.2
% Im:   the input image
% MASK:   the ROI mask
% SX,SY:   the seedpoint location
% IntDist:   the intensitydistance

%-----------------------------------------------------------------------------
GrayWeightFunc = @(x4)
graydiffweight(x4,round(SX),round(SY),'GrayDifferenceCutoff',Cutoff/255);  %intensity distance
0.08
Weight=roifilt2(Im,MASK,GrayWeightFunc);
ImsegFMMFunc = @(x5)imsegfmm(x5, round(SX),round(SY), IntDist);
BW33=roifilt2(Weight,MASK,ImsegFMMFunc);

BW33=bwmorph(BW33,'thicken',5);
BW33=bwmorph(BW33,'close',100);
boundaries33 = bwboundaries(BW33,'noholes');
for j=1:length(boundaries33)
    InflamIntern = boundaries33{j};

end
if ShowFigureOrNot=='v'
 figure('Name','Region growing using Internal functions');
subplot 221;imshow(Im);title('ROI Filtered');hold on
plot(SX,SY,'y+','LineWidth',3); hold off

subplot 222;imshow(BW33);title('Segmentdet area');
subplot 223;imshow(Im+BW33);title('Region growing');
subplot 224;imshow(Im);title('Inflamation boundary');hold on

plot(InflamIntern(:,2),InflamIntern(:,1),'g','LineWidth',2);
hold off;

end

end
```

Published with MATLAB® R2016b

EllipsMASK.m

```
function   [EMasK,ECord]=EllipsMASK(xx,yy,Rx,Ry,I,Threshold,displcM,RgOrAc,vis)
% xx and yy is  the  referencepoint
% Rx and Ry  is the radius in y and x direction
```

```
% curveplot is the external function which make the  Ellips
cr=curvplot(xx-70,yy-80,Rx,Ry); % for regiongrowing

% cr=curvplot(xx-70,yy-20,Rx,Ry); % for ctive contour
% making the  elliptic mask
crX=cr.XData;
crY=cr.YData;
PotSedArData=[crX;crY]';
PotSeedPointAreaBW=roipoly(I,PotSedArData(:,1),PotSedArData(:, 2));
nImgMaskPot=I.*(PotSeedPointAreaBW);
```

```
if RgOrAc==1 %if region growing RgOrAC=1,  else Active contoure
```

```
   % extracting the pixel between this range
    [iy,ix]=find((Threshold-5/255)<nImgMaskPot& nImgMaskPot<Threshold);
    ecd=eucleadiandistance([xx,yy],[ix,iy]);
    [Eucld,idxecd]=min(ecd);
    ECord=[ix(idxecd),iy(idxecd)+displcM]; %% min eucledian distance to bone1
    EMasK=PotSeedPointAreaBW;
    if vis=='v'


        figure;imshowpair(I,PotSeedPointAreaBW,'blend');hold on
        plot(ix,iy+displcM,'bo');% hold off

        plot(ix(idxecd),iy(idxecd)+displcM,'wo','LineWidth',4)
        plot(xx,yy,'ro','LineWidth',2); %hold off

    end
```

```
else
    M1=nImgMaskPot==1/255;
    EMasK=PotSeedPointAreaBW.*M1;
    ECord=[]; %% min eucledian distance to
    if vis=='v'
        figure;imshowpair(I,EMasK,'blend');hold on
        %     plot(ix(idxecd),iy(idxecd)+displcM,'ro'); hold off
    end


end
```

```
end
```

Published with MATLAB® R2016b