ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

ULTRASOUND IMAGE SEGMENTATION USING THE WATERSHED ALGORITHM

M.Sc. THESIS

Sibel KADIOĞLU

Department of Electronics and Communication Engineering

Telecommunication Engineering Programme

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

ULTRASOUND IMAGE SEGMENTATION USING THE WATERSHED ALGORITHM

M.Sc. THESIS

Sibel KADIOĞLU
(504121331)

Department of Electronics and Communication Engineering

Telecommunication Engineering Programme

Thesis Advisor: Prof. Dr. Mustafa KARAMAN

MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ULTRASON GÖRÜNTÜLERİNİN HAVZA SINIRLAMA YÖNTEMİ KULLANILARAK BÖLÜTLENMESİ

YÜKSEK LİSANS TEZİ

Sibel KADIOĞLU
(504121331)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mustafa KARAMAN

MAYIS 2015

Sibel Kadıoğlu, a **M.Sc.** student of ITU **Graduate School of Electronics and Communication Engineering Department** student ID **504121331**, successfully defended the **thesis** entitled "**ULTRASOUND IMAGE SEGMENTATION USING THE WATERSHED ALGORITHM**", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Mustafa KARAMAN** İstanbul Technical University	.....
Jury Members :	**Prof. Dr. Nizamettin AYDIN** Yıldız Technical University	.....
	Assoc. Prof. Dr. Ender Mete EKŞİOĞLU İstanbul Technical University	.....

Date of Submission : 4 May 2015
Date of Defense : 28 May 2015

In loving memory of my father,
Necdet Kadıoğlu.

FOREWORD

I would like to wholeheartedly thank my advisor Prof. Dr. Mustafa Karaman for his support, guidance, and patience. He not only introduced me to the field of medical image segmentation but also I learnt immensely from his style and work ethics.

I would also like to thank Asst. Prof. Dr. Bülent Bolat and Dr. Jouni Pohjalainen for brainstorming on the subject and my family and friends for their support.

MAY 2015 Sibel KADIOĞLU

TABLE OF CONTENTS

Page

ABBREVIATIONS

dB	: Decibel
PSF	: Point Spread Function
PSNR	: Peak Signal-to-Noise Ratio
RAG	: Region Adjacency Graph
RF	: Radio Frequency
RM	: Region Merging
SKIZ	: Skeleton Influence Zone
SNR	: Signal-to-Noise Ratio
US	: Ultrasound
WM	: Weighted Median
WS	: Watershed

LIST OF TABLES

Page

LIST OF FIGURES

Page

ULTRASOUND IMAGE SEGMENTATION USING THE WATERSHED ALGORITHM

SUMMARY

Ultrasound imaging is an important medical application as it is a safe, non-invasive, and interactive procedure for creating detailed visualization of various structures within the body. Computational methods help improving the quality of ultrasound images as well as they aid in the discovery of important structures. In this line of research, this thesis studies the segmentation problem in general, which aims to identify regions of interest of a given image, and in particular, focuses on its application on ultrasound images using the watershed algorithm.

The thesis consists of three main parts: (i) preprocessing of the ultrasound images, (ii) the watershed algorithm, and (iii) postprocesssing the segmented images.

Despite the technical advances in the clinical field, speckle noise continues to pose a challenge in the interpretation of ultrasound images. The first part of the thesis addresses this issue using preprocessing algorithms, namely; histogram equalization and median-based filters to reduce the speckle noise. This yields an initial improvement in the ultrasound image quality.

Next, in the second part, the watershed algorithm is used on the filtered images to discover important regions. The algorithm works based on an analogy from geography where watershed lines separate catchment basins from each other, hence creating different segments. The algorithm is applied on both intensity values and local statistics which are based on mean and variance values.

A well-known drawback of the watershed algorithm however is that it leads to oversegmentation. Therefore, in the third part, region merging algorithms are studied to overcome this issue. A novel region merging algorithm that incorporates several criteria is proposed. The criteria include the size (the number of pixels in a region), intensity (color values associated with each pixel) and statistical measures (such as variance and mean ratio). These algorithms include three parameters denoted as *Tsize*, *Tincrement* and *Tmerge* which are image dependent.

Extensive tests run on a phantom cyst image and a clinical liver image. Our test results show that the numbers of regions for the cyst and liver images are reduced by about ~40 and ~50 times, respectively. Overall, we achieved a significant reduction in oversegmentation using a mixture of pre- and postprocessing. The results demonstrate the effects of each studied component; median-based filtering, watershed segmentation, and region merging.

ULTRASON GÖRÜNTÜLERİNİN HAVZA SINIRLAMA YÖNTEMİ KULLANILARAK BÖLÜTLENMESİ

ÖZET

Ultrason görüntülemesi tıp alanında yaygın olarak kullanılan bir görüntüleme tekniğidir. Bu teknik, herhangi bir yan etkisinin olmamasına, hareketli organların gerçek zamanda izlenebilmesine olanak sağlaması ve invazif olmaması özelliklerinden dolayı tıbbi amaçlı görüntülemede oldukça önemli bir yer tutmaktadır. Bilgisayar destekli yöntemler hem ultrason görüntülerinin kalitesinin artmasına, hem de önemli bölgelerin bulunmasına yardımcı olmaktadır. İş bu tezde, genel olarak, görüntü bölütleme, yani görüntü içerisindeki farklı bölgelerin çıkarılması problemi üzerinde çalışılmakta, özel olarak ise, havza sınırlama metodu kullanılarak, ultrason görüntülerinin bölütlenmesi amaçlanmıştır.

Tez çalışmasının içeriği üç ana bölümden oluşmaktadır: (i) ultrason görüntülerinde ön işleme, (ii) havza sınırlama metodu, (iii) bölütleme sonrası işlemleri.

Klinik alanındaki teknik gelişmelere rağmen, ultrason görüntülerinde meydana gelen benekli yapılar, görüntülerinin yorumlanmasını güçleştirmektedir. Bu nedenle tezin ilk bölümünde bu beneklerin azaltılması amacıyla ön işlemler uygulanmaktadır. Bu ön işlemler, histogram eşitlemesi ve çeşitli medyan filtreleme teknikleridir. Ön işleme metotları ile ultrason görüntülerinde ilk iyileştirmenin yapılması hedeflenmiştir.

Bir sonraki adımda filtrelenmiş görüntüler üzerinde önemli bölgeleri keşfetmek için bölütleme amacıyla havza sınırlama metodu kullanılmaktadır. Bu metot coğrafi yöntemlerden esinlenilmiş olup, havza alanlarını birbirinden ayırarak farklı bölgeler oluşturma prensibine dayanmaktadır. Yöntem, görüntünün gri seviye değerleri ve bu değerlerden elde edilen lokal istatistik değerleri yani ortalama ve değişinti değerleri üzerinde uygulanmaktadır.

Havza bölütleme metodu, aşırı bölütleme ile sonuçlanmaktadır. Bu sebeple, çalışmanın üçüncü bölümünde aşırı bölütleme probleminin çözümü üzerinde durulmaktadır. Bu amaçla farklı kriterleri değerlendirerek çalışan, özgün bir bölge birleştirme metodu geliştirilmiştir. Kriter olarak bölgelerin büyüklüğü (bölge içindeki piksel sayısı), gri seviye değerleri (her piksele karşılık gelen renk değerleri) ve lokal istatistiksel özellikler (değişinti ve ortalama değerleri) baz alınmaktadır.

Bu yöntemler, fantom kist görüntüsü ve klinik karaciğer görüntüsü üzerinde geniş çaplı testler ile denenmiştir. Test sonuçları, kist görüntüsü için bölge sayısının ~40 kat azaltıldığını, karaciger görüntüsü için ise bölge sayısının ~50 kat azaltıldığını göstermektedir. Sonuç olarak, ön ve son işlemler birlikte kullanılarak, aşırı bölütleme probleminin büyük ölçüde azaltılması başarılmıştır. Bu testler, histogram eşitleme, medyan filtre (ve/veya) çeşitleri, havza sınırlama ve bölge birleştirme metodlarının etkisini göstermektedir.

1. INTRODUCTION

Conceptually, segmentation can be described as partitioning of an image into regions that have similar characteristics such as gray level, texture, statistics, and color [1]. Segmentation is used in many applications such as detection, recognition, identification, and image analysis. In medical imaging, image segmentation helps for various objectives including visualization, volumetric measurement, identification of regions of interests (e.g., locating abnormalities), computer integrated surgery, and treatment planning [2].

This thesis focuses on medical image segmentation, and in particular, considers ultrasound images. Ultrasound offers many medical benefits, as it is non-invasive and provides a painless modality that is not harmful for the patient. It performs interactive visualization in real-time, can analyze moving structures and does not include side effects such as radiation. Moreover, ultrasound machines cost less than other imaging methods [3]. As such, ultrasound imaging has great practical importance, and this is our main motivation to study *segmentation of ultrasound images*. Section 2 provides more information on ultrasound images.

Ultrasound image segmentation is challenging due to several reasons. The main difficulty behind the ultrasound image segmentation is the presence of speckle noise, artifacts, and shadowing [4]. Other challenges include the low level of contrast, namely closeness in gray levels of different tissues as well as the complexity and variability of human body. There are many proposed algorithms to reduce speckle noise and increase image quality, as reviewed in preprocessing methods in Section 3.

1.1 Purpose of Thesis

This thesis focuses on ultrasound image segmentation and it is aimed at, first, improving the quality of ultrasound images, and then, identifying important structures in it.

Our first goal is to suppress the speckle noise as it negatively affects the segmentation. In Section 3, speckle noise and its properties are described and speckle reduction methods in literature are reviewed. In this thesis, for the purpose of preprocessing, *histogram equalization* and *filtering methods* are used. Because of the similarity between speckle noise and salt-and-pepper noise, *median filter* and its variants such as *weighted median filter*, *hybrid median filter* and *adaptive median filter* are utilized.

Next, for image segmentation, the watershed algorithm based on immersion by flooding is used. In particular, the Vincent – Soille algorithm is implemented as described in Section 4. This algorithm, and in fact all watershed algorithms, results in oversegmentation. To solve this issue, this thesis proposes novel region merging algorithms based on different criteria as discussed in Section 5.

The overall structure of the work done in this thesis can be illustrated as follows. The details of each component is later given in **Figure 6.1**.

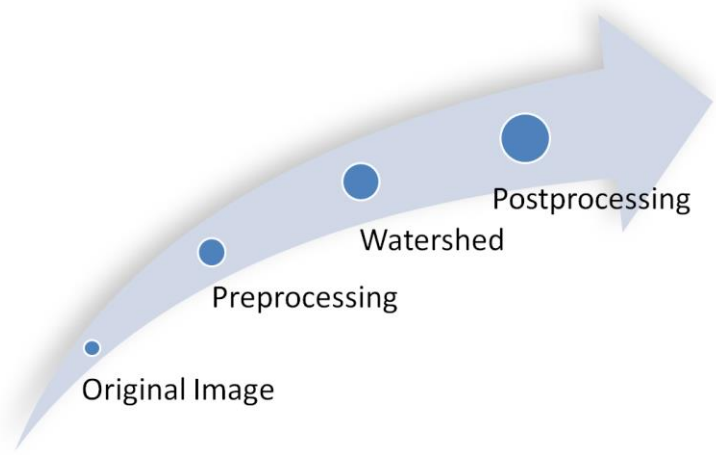


Figure 1.1 : The overall structure of the work done in this thesis.

Based on this schema, each component of our procedure is tested through extensive experiments as summarized in Section 6. Section 7 presents the conclusions and Section 8 lists a number of areas with potential improvement for future work.

1.2 Literature Review

There exist a wide range of ultrasound image segmentation methods in the literature [5]. However, the segmentation of brain tissue has different requirements compared to the segmentation of liver. Therefore, the existing methods can be classified

according to their clinical application such as cardiology, breast cancer, intravascular diseases etc. A comprehensive overview of segmentation algorithms based on this classification can be found in [4]. Segmentation methods are introduced based on detection of similarity or detection of discontinuity criteria. The detection of similarity is region-based according to a predefined criterion and the detection of the discontinuity is based on sudden changes in the intensity levels. Based on the papers [6,7,8] segmentation methods can be grouped as follows.

Edge based segmentation methods aim to detect edges between different regions by using the information that intensity levels change on the edges. Edge detection based methods (e.g., Sobel operator, Canny operator, Laplace operator [1]) cannot always create closed contours. They also require high image quality, which is not possible for ultrasound images. Therefore, these methods are not sufficient.

Random field based segmentation methods can be Markov Random Field (MRF) and Gibbs Random Field models. In these methods, intensity values of pixels are modelled as a random variable, which have a probability distribution function. MRF algorithm consists of neighborhood definition, energy function, which is minimized by gradient descent, simulated annealing etc., and parameter estimation for maximum a posteriori probability.

Partial differential equation based segmentation (deformable models) uses curves that follow smoothness and object boundary knowledge which are ruled by energy functions. Deformable models are divided into two types as parametric deformable models and geometric deformable models. For instance, active contour model, also known as snake model, is a type of parametric deformable models. Snake model uses both regional and border knowledge. Its energy function includes internal and external energy terms. Internal energy is the sum of elastic and bending energy. Elastic energy makes the snake act like membrane and controls the rigidity of the contour while bending energy makes the snake act like thin plate and responsible for shrinking of the contour. External energy is derived from the image and it gets the smaller value at the region of interest. When the sum of energy functions are minimized, that curve represents boundary of the regions. The drawbacks of this method are that it strongly depends on the initial position of the snake and it has high computational complexity [6].

Geometric deformable models are based on level set method and curve evolution theory. Instead of energy function, curves progress according to geometric measures, which do not include parameters. Curves are denoted as a level set of a higher dimensional function.

Machine Learning based algorithms, segmentation problem is treated as classification problem. In this case, clustering algorithms (*K-means*, *fuzzy C means*, *Expectation Maximization*, *Support Vector Machine*s) are used. *Clustering* methods are unsupervised methods, which do not require a training phase. As clustering algorithms neglect the spatial information, they are sensitive to noise. Clustering algorithms also require initialization. Spectral clustering algorithms are based on graph theory using different techniques such as Laplace matrix, eigenvalue decomposition and so on.

Support vector machines with a radial basis function kernel are used to classify different patterns, which are generated by sliding window over entire image as in [9].

Artificial neural networks consist of many parallel nodes, which have weights according to their importance. Artifical neural networks require a training phase where learning weights and connection between nodes are learned.

For machine learning algorithms, *feature extraction* and *training phase* are important as segmentation heavily depends on the selected features. Feature extraction and training also influence the computation time of the algorithm. In supervised learning methods, another issue can occur in the training phase when the training over fits to the data, and then the algorithm cannot classify unseen samples accurately. On the contrary, if the training under fits the data, then it is also not sufficient to label new samples. In the case of ultrasound images, it is hard to find varied training data.

Region based segmentation treats objects as regions and tries to find regions based on predefined criteria, for instance intensities, texture properties etc. A region growing algorithm [8] starts from a seed point and regions are expanded according to the neighbors with similar characteristic. In this method, the difficulty is in choosing the initial seed point and setting a criteria for growth. In region splitting and merging algorithm [8], the image is first partitioned into areas, and then, merging or splitting is applied according to a defined split/merge rule.

Threshold methods are used to separate different gray levels. They can be classified as global threshold, local threshold, and dynamic threshold methods. Threshold methods are susceptible to the noise; hence, they are used together with other methods.

There are several *watershed algorithms* used for ultrasound image segmentation [10, 11], most of which use the immersion-based watershed algorithm. These watershed algorithms are both edge and region based. The watershed algorithm results in over segmentation and the differences between these approaches are due to the different region merging methods that follow the watershed approach.

In [10], the size of regions, i.e., the number of pixels in the region, is utilized for region merging. The idea is to merge the small regions together based on some threshold values as stopping criteria.

In [11], three different criteria are used for the region merging algorithm. Their criteria are edge information, gray level information and relationship between neighbor regions.

Besides immersion-based watershed algorithm, marker-controlled watershed algorithms are also proposed to deal with the oversegmentation problem on ultrasound images [12,13].

Finally, it is worth mentioning that image segmentation algorithms are hard to evaluate and compare with each other. One possibility is that algorithms can be evaluated on phantom images, for which the features are already known. Another alternative is to use a ground truth image, which can be segmented manually by experts. However, note also that this is human-biased and time-consuming. Finally, unsupervised evaluation methods such as stand-alone evaluation [14], empirical goodness method, intra- and inter-region variance, performance vector, internal and external contrast do not require a reference image. How to compare the different algorithms in the absence of ground truth is studied in detail in [15].

2. WHAT IS ULTRASOUND?

This section covers some background information regarding ultrasound imaging. It starts with a description of some of the properties of ultrasound waves and then presents the principles and modalities of ultrasound imaging systems. More information on ultrasound imaging can be found in [16].

2.1 Properties of Ultrasound

Ultrasound (US) is essentially sound with a frequency above 20 kHz, which is not audible for human hearing and needs a medium such as tissue to propagate. US waves are longitudinal waves, which means that their motion is parallel to the direction of the energy transfer.

US waves can be explained in terms of *frequency*, *wavelength*, *velocity*, *intensity*, and *power features* [17]. Frequency of the ultrasound waves is proportional to the speed of sound in a tissue while inversely related to wavelength. This relation is shown in the equation (2.1) where f is frequency, c is velocity and λ is wavelength.

$$f = \frac{c}{\lambda} \tag{2.1}$$

It should be noted that propagation speed does not depend on frequency and wavelength. It is solely based on medium properties that the wave propagates through medium. Mathematical description of the propagation speed is given as.

$$c = \sqrt{\frac{\kappa}{\rho}} \tag{2.2}$$

where c is the velocity of sound, ρ is the density in the medium, and κ is the measure of stiffness, which means resistance for not being deformed when wave is compressed. In summary, propagation speed is related to density and compressibility while speed in the medium is related to the frequency and wavelength.

Intensity of the ultrasound waves is related to energy transportation while waves propagate in medium. The average intensity corresponds to the power term. This can be interpreted that intensity of ultrasound waves are proportional to their power. However it is inversely proportional to area of the waves. In acoustics, decibel (dB) scale is used to represent the intensity.

Power in ultrasound denotes the rate of generated and transferred energy by the acoustic wave per time. [18].

2.2 Ultrasound Imaging

Conceptually, ultrasound imaging consists of sending acoustic pulse with high frequency through the body, and then transforming the reflected echoes into electric signals, which are converted into ultrasound images.

An ultrasound system consists of the pulse generator, transducer, amplifier, scan convertor, image memory, image display and recording system [19]. Simplified block diagram of ultrasound imaging system is shown in the **Figure 2.1**.

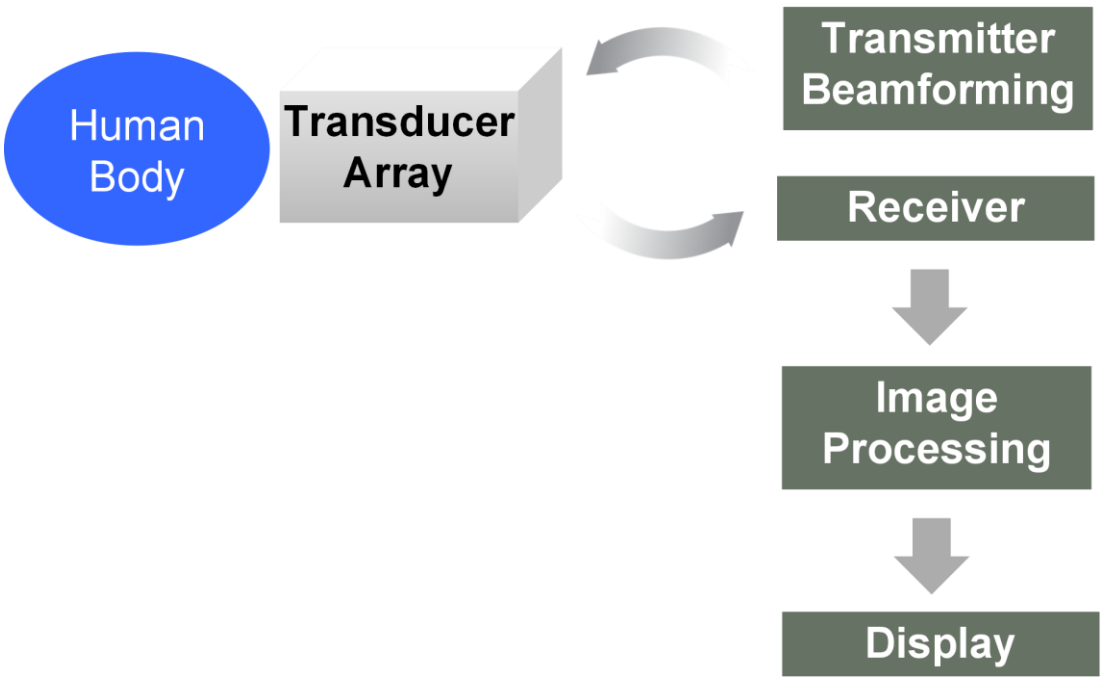


Figure 2.1 : Block diagram of an ultrasound imaging system.

The first requirement of the system is to generate the ultrasonic waves which are transmitted into the body. A transducer based on piezoelectric principle is used for

this purpose. Piezoelectric materials have the property that when mechanical stress is applied, electric field is produced and vice versa. In order to create waves, short duration voltage pulses are applied to the transducer. After the waves are generated, the transducer acts as a receiver. In receiver mode, the transducer collects the waves which are reflected back due to the interaction between waves and tissues. Then a transmitter connects the returning waves (echo) to an electronic signal that is processed and displayed.

Transducers used for medical imaging are arrays. Their commonly used types are linear transducers, curved linear transducers and phased array transducers.

The linear transducers consist of a large number of piezoelectric elements arranged in parallel lines. The parallel positoning of scan lines produces a rectangular image with the same width as the transducers. This kind of transducer is preferred for vascular and some abdominal imaging. The disadvantage of linear transducer is that it only focuses on the image plane which causes image thickness. This has negative impact on spatial resolution.

In the curved linear transducer, piezoelectric elements are located on curve instead of straight lines as in the linear transducer. Image shape of the curved linear transducers is trapezoidal. When the search depth increases namely large tissue structures are examined, for instance in abdominal applications, the curved linear transducers are employed.

The phased array transducer has an array of piezoelectric elements similar to the linear array. Ultrasound pulses are transmitted in straight line which are perpendicular to the transducer unlike linear array transducer. This assembly generates triangular-shaped image. These kind of transducers are designed for cardiac applications. Its disadvantage is that, its procedure is more time-consuming compared to 2-D alternatives because of 3-D properties. These transducer types are shown as example based on [20] in the **Figure 2.2**.

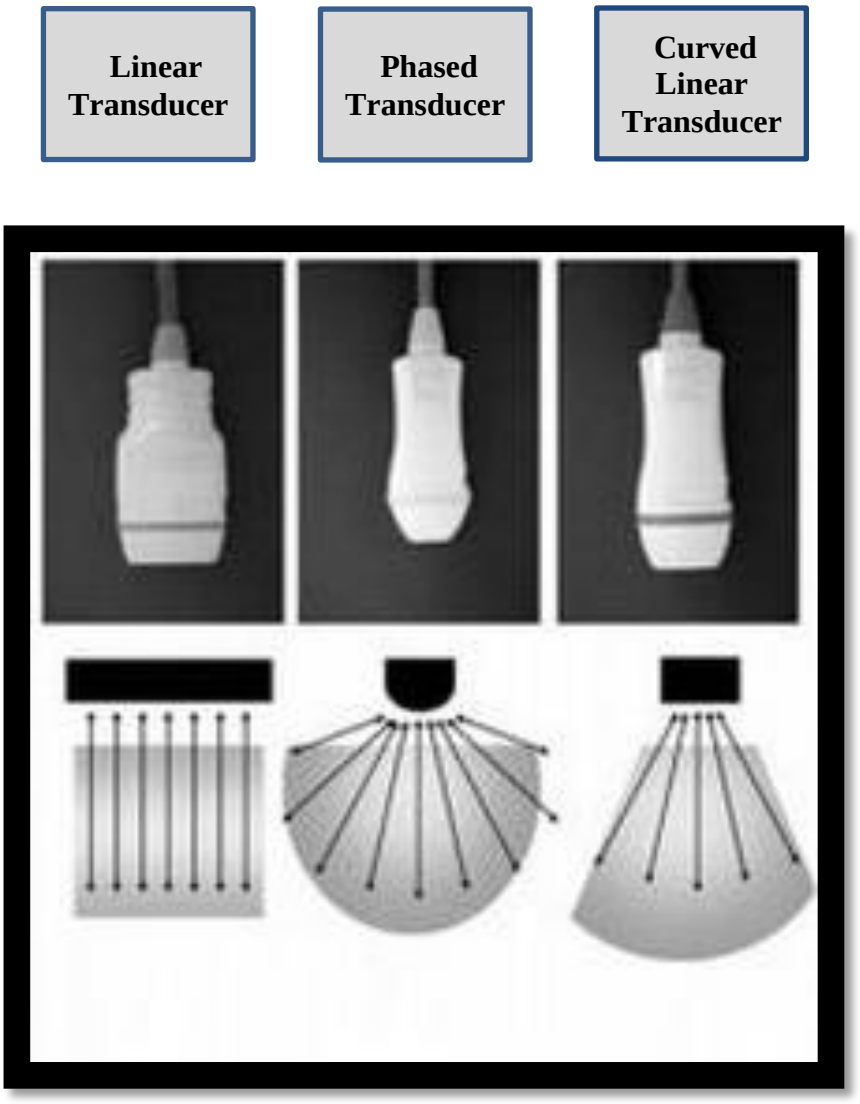


Figure 2.2 : Transducer types based on [20].

The quality of the ultrasound image is related to resolution. Resolution term in ultrasound images can be defined as ability to distinguish different point and has types such as spatial, lateral and temporal resolution. Spatial resolution describes this ability in space and is subcategorized as axial and lateral resolution. Axial resolution represents the resolution property along the ultrasound propagation direction. It is determined by the length of the ultrasound pulse. Lateral resolution denotes the ability of the system to discriminate two points in the direction perpendicular to the direction of propagation. In ultrasound imaging, axial resolution is better than lateral resolution [21]. Temporal resolution is the ability to detect moving objects over time and this can be interpreted as frame rate for medical ultrasound applications.

2.2.1 Ultrasound – tissue interaction

As ultrasound waves are directed into body, it causes different interactions due to the characteristic difference of tissues. These observed interactions are *reflection*, *refraction*, *scattering*, *diffraction* and *interference* [19,21]. Briefly, they can be explained as follows:

Reflection occurs at the boundaries between different tissue types because of acoustic impedance property. Acoustic impedance is based on medium density and propagation speed, and calculated as a product of these terms. Once meeting with a tissue, some waves can be transmitted into other tissue while some are reflected back. The amplitude of the waves is related to acoustic impedance change. Specifically, small acoustic impedance differences produce weak echoes and vice-versa. Another cause of reflection is due to the fact that interfaces are larger compared to the wavelength of ultrasound.

Refraction is defined as direction changes of waves when they hit the boundary of organs. Direction does not change under the condition that speed of sound remains same for both media. Its adverse effect is misleading localization of a structure on an US image. The speed of sound is low in fat and high in soft tissues.

Scattering takes place when US waves interact with tissues that are smaller than ultrasound's wavelength. It is directly proportional to frequency, which means it rapidly increases when frequency of ultrasound increases. Because of the spatial arrangement of the scatterers, there are two types of scattering. If the scatterers produce periodicity in the echo waves, it is called *coherent scattering*. If the scatterers are spatially random distributed, it leads to the *diffuse scattering*. Scattered power is proportional to a tissue size, which is smaller than wavelength, and inversely proportional to the wavelength.

Diffraction occurs in the case of wave spreading through small holes. The amount of diffraction depends on size of this hole and wavelength of waves.

Interference yields phase information when waves align with each other. Main cause for interference is scattering. Interference varies as *constructive* and *destructive*. If waves are in the same phase, it causes constructive interference, which arises from increased amplitude. If waves are out of phase, they are subject to

destructive interference, which leads to a complex wave summation. Focusing of the US beam in real time imaging is based on the principle of wave interference.

Attenuation results in absorption, reflections and destructive interference. Absorption of ultrasound beam is defined as energy conversion process, primarily heat. Attenuation increases proportionally with frequency.

2.2.2 Ultrasound modes

Ultrasound scanning modes mainly varies as *A*- mode and *B*- mode [22]. Other modes such as *M-*, *T-*, *color coded*, and *Doppler* mode are derived from these main modalities. Commonly used modes are outlined below based on [21,22].

A-mode scanning is named based on amplitude, and it is the earliest mode of ultrasound which is no longer used. In this mode, reflected signal is measured and displayed as a continuous signal. The display of the tissue is obtained in one dimension.

B-mode scanning refers to brightness mode. In this mode, reflected waves are demonstrated as a 2D gray level image, which is obtained in many different directions. Amplitude of reflected waves are denoted by gray levels that varies from black to white.

Motion mode, *M–mode*, is similar to A–mode scanning. Results are displayed according to the lines, which are obtained consecutively across time. The vertical axis is the depth of the tissue in meters, while horizontal axis is time in seconds. Unlike B-mode, in M-mode display acoustic axis is fixed.

T-mode display denotes transmission mode scanning. In this mode, there is 180° difference between receiver and transmitter of the ultrasonic wave unlike B-mode. Information about the tissues are obtained by transmission instead of reflection. This mode is used for breast scanning.

Doppler mode provides velocity sampling at different depths and positions. Several types of doppler systems are used in medical diagnosis such as continuous wave Doppler, pulsed wave Doppler, duplex ultrasound and color flow duplex doppler.

3. PREPROCESSING OF ULTRASOUND IMAGES

Preprocessing is an important step of digital image processing. It reduces the artifacts and provides an initial enhancement in the quality of the image. For ultrasound images, the preprocessing method is aimed at *speckle noise reduction*, also known as *despeckling* [22].

Due to coherent energy, speckle noise inherently occurs in ultrasound images. Speckle degrades the image quality in terms of resolution and contrast. This has a negative impact on both diagnostic examination and image processing methods, and hence, needs to be eliminated.

It should be noted that speckle pattern is mostly asserted as noise, which needs to be eliminated, but on the other hand, it also includes important information about tissues [23]. That is, there exists a trade-off between reducing noise and losing information about the image.

Our preprocessing procedure has two components: (i) histogram equalization methods, and (ii) filtering methods. This section first presents speckle noise, its physical properties, and reviews the methods used to reduce it. Then, the details of our preprocessing methods are described.

3.1 Speckle Noise

As previously mentioned, when the beam strikes a tissue, whose size is bigger compared to the wavelength of the beam, it results in scattering. When scatterers are in random phase, coherent summation of backscatterers generate light and dark spots, which is called as *speckle*. Scatterer numbers per region of the object, known as the *scatter number density* (SND), determines the pattern of speckle. This pattern is repeatable, which means that under the same conditions, it is obtained identically in the same way [24].

Speckle noise occurs in many procedures including; coherent radiation, synthetic aperture radar (SAR), laser and medical ultrasound imaging, which is the focus of

this thesis. In ultrasound imaging, speckle noise types are divided into four types based on SND and deterministic components:

- **Fully developed form**, has large number of scatterers without deterministic component. Amplitude of backscatters is modeled as Rayleigh distribution [24,25] and their phases are distributed uniformly between - π and + π.

- **Fully resolved form**, which has deterministic component in addition to large number of scatterers and modeled as Rice distribution, where echo signal's mean is shifted away from the origin [26].

- **Partially developed form**, which has small number of scatterers unlike fully types with non-existence of deterministic component and it has *K*-distribution model [27,28].

- **Partially resolved form** has features of both small number of scatters and deterministic component. It is represented as *K*-Homodyne distribution [28].

There are many proposed methods for modeling the speckle noise [22]. Fully speckle condition is the most popular in ultrasound images [30], as such; the speckle noise model is defined according to it. The study presented in [30] suggests that the assumptions behind the Rayleigh distribution are better suited to model pixel intensities in homogenous areas compared to other distributions such as *K*-, Nakagami-, and Rician Inverse Gaussian distribution. Rayleigh distribution assumes that the mean value μ is proportional to the standard deviation σ with μ/σ. This implies that speckle noise can be modeled as multiplicative with original image. The variance is related to image intensity in such way that brighter areas include more noise power compared to dark areas. Ultrasound images are displayed on logarithmic scale, and hence the speckle mean μ becomes proportional to variance σ [31-33].

$$g(x,y) = s(x,y) + \sqrt{s(x,y)}\, n(x,y) \quad \textbf{(3.1)}$$

Where $s()$, $g()$ and $n()$ represent the observed signal, noise-free signal, and noise respectively. This model implies that, the image variance is proportional to mean on homogeneous regions where $s()$ can be assumed as constant.

3.2 Speckle Reduction Methods

In the literature, various methods are proposed to reduce speckle noise. Mainly, these methods can be grouped as *linear filtering*, *non-linear filtering*, *diffusion based filtering,* and *wavelet filtering* [21].

Some of the linear filters are used for speckle reduction are Lee [34], Kuan [35] and Frost [36]. The principles behind these filters are based on a sliding window (*kernel*) over the image. Lee filter is based on *minimum mean square error*. In this filter, if the variance in window is low, the central pixel of the window is replaced with the local mean of pixels inside the window. Contrarily, if the variance is high, Kuan filter is used which is a generalized form of Lee filter under non-stationary mean and variance assumption. In Frost filter, the image is convolved with exponential impulse response. This filter performs similar to averaging in low variance areas, and in high variance areas, it instead preserves sharp features.

Regarding nonlinear filters, *homomorphic filters*, *maximum homogeneity over pixel neighborhood filtering* (MHOPNF) can be given as an example.

Homomorphic filters [37] are based on Fourier transform including a logarithmic mapping that converts multiplicative speckle noise form to additive noise. After obtaining image with additive noise, high pass filters in logarithmic domain are applied in order to remove the noise.

In MHOPNF [38], by sliding the window, the center pixel is updated based on the most homogenous neighborhood around which is calculated for every pixel in the image. This filter does not require any parameters or threshold to be tuned. Diffusion filtering varies as *anisotropic diffusion filtering*, *speckle-reducing anisotropic diffusion* (SRAD) and *coherent diffusion*.

Anisotropic diffusion filter [39] uses nonlinear partial differential equation (PDE) to remove speckle. This method is capable of reducing speckle and at the same time strengthening the edge information with edge detection function.

SRAD [40] utilizes instantaneous coefficient of variation instead of gradient-based edge detector in PDE of the anisotropic diffusion filter. This method is applied on image, which does not have logarithmic compression.

Coherent diffusion [41] is proposed for coherent enhancement. It can be treated as a hybrid method that combines isotropic diffusion, anisotropic diffusion and mean curvature motion.

In wavelet filtering [42], the image is transformed into wavelet domain and noise is removed based on wavelet coefficients.

As noted in [22], most of the proposed speckle reduction methods have their limitations. First of all, they are sensitively dependent on the window size and the shape. That means, the choice of window size is critical. In case, the window is too large, it smooths overly, as a result edge information is lost due to blurring. On the other hand, if the window is too small, then it does not sufficiently smooth the image so as to reduce speckle noise. Secondly, some methods are based on a threshold value which should be set properly according to the image at hand. When it is not chosen correctly, it leads to noisy boundaries. Both quantitative metrics (SNR, PSNR etc.) and visual assessments by experts are used for evaluation but differences between such measurements prevent objective comparison about speckle filtering methods.

In this thesis, speckle reduction methods are used as preprocessing which consist of two parts. First, in order to increase dynamic range of pixel values, histogram equalization and its adaptive version are used. Then, since speckle noise can be considered as a salt and pepper noise, the median filters are applied as a second step. In particular, *median filter* and its variants such as *hybrid median filter*, *weighted median filter* and *adaptive median filter* are used. The next section describes these methods.

3.3 Preprocessing – I: Histogram Equalization

Firstly, histogram of an image denotes the frequency of occurrence of the pixel values. Mathematically, it represents probability distribution of the pixel based on divided value of number of specific k^{th} pixels to total number of pixels.

Histogram of the image is interpreted in a way that if image has a narrow histogram, its contrast is low, namely the differences between maximum and minimum intensities is small. For visibility and quality of the image, a large histogram is

demanded [43]. Another knowledge is obtained form histogram is number of peaks denotes the number of regions which can be used to analyze the image.

Histogram equalization represents the histogram as a cumulative distribution which is summation of the pixel distributions. This accomplishes to make pixels uniformly distributed. In the equations below, former one denotes the calculation of the histogram and latter one represents the transformation of the pixels distribution namely histogram equalization is shown.

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \dots, L-1 \tag{3.2}$$

$$s_k = \sum_{j=0}^{k} p_r\left(r_j\right) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, 2, \dots, L-1 \tag{3.3}$$

Where r_k is k[th] gray level, $p_r(r_k)$ is its probability density function. In next equation, s_k is the output image when each pixel with level r_k is mapped. For both equation n_k is number of k[th] pixels, n is total number of pixels.

Histogram equalization enhances the image globally but if local enhancement is required then its extension, namely adaptive histogram equalization, is used. In adaptive histogram equalization, histogram equalization is calculated on small regions (tiles) rather than entire image. To avoid noise amplification, contrast expansion is limited. In the end, histogram equalized tiles are combined by using bilinear interpolation.

3.4 Preprocessing – II: Filtering Methods

3.4.1 Median filter

Median filter [40] is non-linear and categorized as order statistic filter. It is non-linear because it does not satisfy the following superposition property:

$$Median[A(x) + B(x)] \neq Median\, A(x) + Median\, B(x) \tag{3.4}$$

where $A(x)$ and $B(x)$ represent the images. This fact should be taken into account if summing of filtered images is considered.

Median filter is applied in the spatial domain by shifting a window through the image, where the center pixel of input image is replaced with the median value of ascendingly sorted pixel values inside the window. This step is repeated for all pixels, yielding the filtered image. For pixels at the edge, the window size exceeds image, therefore zero padding or symmetrical padding is used. Since the median value is used, the window size is mostly chosen as an odd number. Otherwise, after sorting the pixels, the average of two middle values has to be calculated.

With this method, outliers, which have high intensities in the image, are suppressed, and variances of intensities are reduced. Therefore, its effect on image can be interpreted as smoothing. However, it should be noted that smoothing causes blurring which is an undesired effect.

The performance of the median filter depends on the window size and the type of noise. In terms of window size, the median filter can remove noise correctly if the number of noisy pixels in the window is less than $(n^2+1)/2$ where the window size is $n \times n$ [45]. In terms of the noise type, in [44] it is shown that median filters are capable of dealing with multiplicative and impulsive noise.

There are various variants of median filtering techniques including directional median, iterative median, recursive median, switching median, sequential median filters so on. Apart from the median filter, in this study, hybrid, weighted and adaptive versions are used for preprocessing, as outlined next.

3.4.2 Weighted median filter

While in median filter, all pixels have equal weights, in weighted median (WM) filter [46], the window is weighted using integer/non-integer coefficients for specific filter positions. WM filter can be defined in two different, but equivalent ways, based on the choice for selecting the weights, as shown in [46]:

Definition I: weights are assumed as non-negative integer values. The first pixel inside the filter window is replicated *n* times where *n* corresponds to weights in filter window. Then, the replicated array is sorted in ascending order, and lastly, the median value of array is chosen. This process can be described as:

$$WM(X_1, X_2, \dots, X_N; a) = MED[a_1 \otimes X_1, a_2 \otimes X_2, \dots, a_N \otimes X_N] \quad \textbf{(3.5)}$$

Where $(a = a_1, a_2, \dots, a_N)$ corresponds to weights in the filter window, $x = (X_1,\ X_2, \dots, X_N)$ is pixel values and $\otimes$ symbolizes duplication operation. The result of this equation is new pixel value in the filtered image.

Definition II: weights are assumed to be positive non-integer. Let the weighted median value of X set of pixels in window be β, which minimizes the following equation.

$$L(\beta) = \sum_{i=1}^{N} W_i |X_i - \beta| \qquad \textbf{(3.6)}$$

Piecewise linear and convex properties of $L(\beta)$ provides that β is certainly one of the samples of X_i.

Calculation WM of pixels is carried out as follows: sort the pixel values in the window in ascending order, which also changes the order of weights. After sorting pixels and reordering, weights are summed up starting from first element. This iterative summation stops when the sum of the weights considered so far, exceeds half of the total sum of weights. The output value for WM is the pixel corresponding to the last weight added.

In this study, Definition – I, is used for the WM filtering. The weights of the filter are chosen in such way that pixels further from center contribute less.

3.4.3 Hybrid median filter

Hybrid median filter [43] is a multiple ranking filter, specially based on applying median filter several times with different positioning of the filter.

In this method, center pixel value within window of filtered image is assigned as follows: First, median value is chosen among the horizontal and the vertical neighbors of the center pixel. Second, the diagonal neighbors' median value is found. Median values, obtained in first and second step, are ranked with the central pixel and final median value is determined from these three median values.

For example, given a window of size 3x3 where C is the center pixel and its diagonal neighbors are denoted by D, its horizontal neigbors are denoted by H and its vertical neighbors are denoted by V.

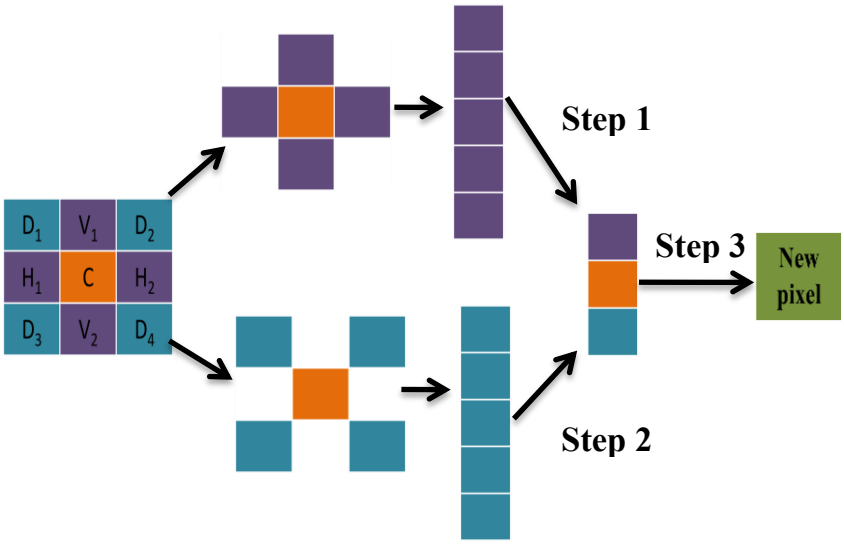


Figure 3.1 : Illustration for computation of hybrid median filter, adapted from [43].

Based on the figure above, the hybrid median filter is calculated as follows.

Step 1: medianN= findMedian (V_1 V_2 C H_1 H_2)

Step 2: medianD = findMedian (D_1 D_2 C D_3 D_4)

Step 3: hybrid median = findMedian(medianN, medianD, C)

Compare to conventional median filter, advantages of hybrid median filter can be considered as less smoothing, better preservation for lines and edges, and faster due to the less computation [1].

3.4.4 Adaptive median filter

In adaptive median filter [1], the window size changes during the filtering process based on the local features within window. The goal of this filter is to remove impulsive noise while other noise types are suppressed. Based on the description given in [1], the following notation and algorithm is used to describe the steps of this filtering.

Notation:

Wxy = Window at (x,y) where (x,y) is window center
W_max = Maximum window size
I_min = Minimum gray level value in Wxy
I_max = Maximum gray level value in Wxy
I_med = Median gray level value in Wxy
Ixy = Gray level value at (x,y) coordinate

LEVEL – I:

```
if      ( (I_med – I_min) > 0 && (I_med – I_max) < 0) )
          Go to LEVEL – II
 else
          Wxy = Wxy  + 2
          if       Wxy < W_max
                    Repeat  LEVEL – I
          else
                    Center pixel =  Ixy
          end
end
```

Figure 3.2 : LEVEL – I of adaptive median filter.

LEVEL – II:

```
if   ((Ixy – I_min) > 0 && (Ixy – I_max) < 0)
       Centerpixel = Ixy
  else
       Centerpixed  = I_med
  end
```

Figure 3.3 : LEVEL – II of adaptive median filter.

In this algorithm, Level – I tries to find out whether *I_med* is impulsive noise. In case, it is not impulsive noise, the window size is increased and this level is repeated until the window size reaches its maximum. When *I_med* is impulsive noise, *Ixy* value is checked in terms of having impulsive noise or not. When *Ixy* is not an impulsive, then *Ixy* is returned as output, and otherwise, *I_med* is returned as output.

4. THE WATERSHED ALGORITHM

The watershed concept for image processing is used as segmentation method which is region based. The algorithm is based on an analogy from geography where the watersheds form the lines that separate drainage areas, which are called *catchment basin*, of different rivers [47].

The input image is treated as a topographical surface where its gray level values considered as altitude because of the intensity differences. The catchment basins represent regions of segmented objects, while watershed lines are boundaries between different catchment basins.

There exists a number of different watershed algorithms. This thesis employs the immersion-based algorithm. An overview of existing watershed algorithms is provided in the next section. A more detailed survey on watershed algorithm can be found in [47].

4.1 An Overview of Watershed Algorithms

The idea of using the watershed method is first proposed by Digabel and Lantuéjou [48] by using binary image for contour detection. Later, Beucher and Lantuéjou [49] proposed extended watershed method to be used in gray scale images. There are many definitions for the watershed algorithms in literature. The watershed algorithms for the discrete case can be classified mainly in five groups based on [50].

The first definition is based on *flooding*, which simulates the immersion of the topographical surfaces with water. It is a recursive process where regions expand based on skeleton by influence zone. The algorithm used in this thesis follows the flooding definition, hence, this is explained in more detail in the next section.

The second definition is based on *topological distance* watershed where topographical distance between any two pixel is the minimum path among all topographical distances. The important point here is that the paths between pixels should be the steepest descent. The watershed based on topographical distance is

described over a cost function. The cost function between any two pixel is computed as summation of geodesic distances between middle pixels. This function determines that pixels belong to unique minimum, hence, catchment basins. If pixels belong to more than one basin, it is labeled as watershed.

The third definition is based on *path-cost minimization* algorithms. In this case, pixels are grouped as catchment basin when topographical distance is precisely small to the regional minimum. In image foresting transform, the image is described as a weighted graph. Then, a forest of minimum path-trees is created by path-cost minimization. Trees in this forest correspond to basins.

The fourth definition is based on *local condition* watershed algorithms that assign the label of minimum points to each pixel instead of building watershed lines. It is derived from topographical distance which does not have unique label restriction anymore. Locality means that image are subdivided into blocks and labelling is performed over evey block separately in order to get globally good results.

The fifth one is based on *minimum spanning forest*, the watershed is constructed as a weighted graph where its nodes are catchment basins corresponding to regional minima.

Besides these definitions, another source of difference is due to the strategies used to scan the pixels. These strategies are *immersion, hierarchical queue*, *shortest path, hill climbing, sequential scanning, connected components, union-find, chain code, tie-zone, order invariant toboggan and immersion* and *watershed cut* [51].

Noise and local inhomogenity have negative impact over the watershed algorithms resulting in oversegmentation. Additional knowledge such as gradient and color might be used to overcome this issue, which can be embedded in *marker-controlled* watershed algorithm [1]. These markers are connected components that contain a priori knowledge for segmentation. They can be classified as *internal* and *external* markers. Internal markers are applied to get information about region of interest, which includes the objects, that is, the foreground. External markers are used to gain an understanding of the outside of the object, that is, the background. In addition, markers can be assigned manually or automatically.

Before describing the technical details of the immersion-based flooding algorithm, namely; the Vincent-Soille algorithm, some definitions are given in the next section.

4.2 Definitions

First of all, description of a gray scale image is a triple $G = (D, E, f)$ where (D, E) is a graph $f : D \rightarrow \mathbb{N}$ is a function which assigns an integer to each pixel p, $p \in D$. In view of this definition, $f(p)$ represents the gray level value or altitude in watershed algorithms.

Neighborhood: The neighborhood can be defined in three ways, 4-, diagonal and 8-neighborhood. For a given pixel *p(x, y)*;

The 4-neighbors are:

$$f : \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\} \quad \textbf{(4.1)}$$

The 4-diagonal neighbors are:

$$f : \{(x+1, y+1), (x-1, y-1), (x-1, y+1), (x-1, y-1)\} \quad \textbf{(4.2)}$$

The 8-neighborhood consists of both 4- and 4-diagonal neighbours.

Adjaceny: If two pixels are neigbors and their gray level values satify same similarity criterion, then these pixels are assumed to be adjacent.

Path: A path is a sequence of pixels. For example, the path from pixel *p* to pixel *q* with length *l* is such that; $(p_0, p_1, \ldots, p_{l-1}, p_l)$ where $p_0 = p$, $p_l = q$, and pixels p_i and p_{i-1} are adjacent for $1 \leq i \leq l$. If pixel values do not increase in the path, the path is called a *descending path*. The opposite condition is *ascending path* where pixels values do not decrease.

Connected Component: A connected component is a subset of pixels in an image where there is a path between any two pixel.

Level Component: A connected component in which the pixels have same value at specific level *h* is known as a level component. The level *h* is known as altitude.

Regional Minimum: A connected component of pixels under the condition that there is no pixel with a lower value than *h.*

Plateau: A plateau is a connected component in which pixels have constant gray value.

Threshold: Threshold is set of values which are lower than *h.* Formally;

$$T_h = \{\, p \in D \mid f(p) \leq h \,\} \tag{4.3}$$

in which f symbolizes gray level values in pixels p.

Geodesic Distance: The geodesic distance $d_A(x,y)$ is defined for two points *x* and *y* in *A* where $A \subseteq \mathbb{Z}^d$, and it represents the length of the shortest path among all paths from *x* and *y* within *A.*

Geodesic Influence Zone: Let *B* be a subset of *A* where *B* is composed of *n* connected components B_i, $i = 1,\ldots,n$. Then the geodesic influence zone of B_i in A is defined as;

$$iz_A\,(B_i) = p \in A \mid \forall j \in [1 \ldots n] \setminus \{i\} : d_A\,(p, B_i) < d_A\,(p, B_j) \tag{4.4}$$

This equation can be interpreted as the geodesic influence zone of B_i is the set of points in *A* that their distance is the closest to *A* among all B_i. The union of the geodesic influence zone of the all connected component is;

$$IZ_A(B) = \bigcup_{i=1}^{n} iz_A\,(B_i) \tag{4.5}$$

Skeleton by Influence Zones (SKIZ): The skeleton influence zone is defined as the complement set of the geodesic influence zone of all components. Based on the equation (4.5);

$$SKIZ_A(B) = A \setminus IZ_A(B) \tag{4.6}$$

That means geodesic SKIZ is the boundaries of all geodesic influence zones.

Catchment Basins: The catchment basins are regions in *S* surface, which contain only one regional minimum. $CB(m_i)$ of minimum m_i among minimum set of m_k where $k = 1, \ldots, n$. This means points in $CB(m_i)$ is closer to m_i than any other regional minimum m_k.

$$CB(m_i) = \{x \in D | \forall j \neq i : f(m_i) + T_f(x, m_i) < f(m_j) + T_f(x, m_j)\} \quad \textbf{(4.7)}$$

In the equation above, f denotes gray level values in D, T_f represents distance between minima.

Watershed: A watershed is the set of points which do not belong to any catchment basin.

$$watershed(f) = D \cap \{\cup_{i \in I} CB(m_i)\}^c \quad \textbf{(4.8)}$$

4.3 The Vincent – Soille Algorithm

The Vincent – Soille algorithm [52] is an immersion-based watershed algorithm. The analogy behind this algorithm is shown in **Figure 4.1** which is adapted from [53].

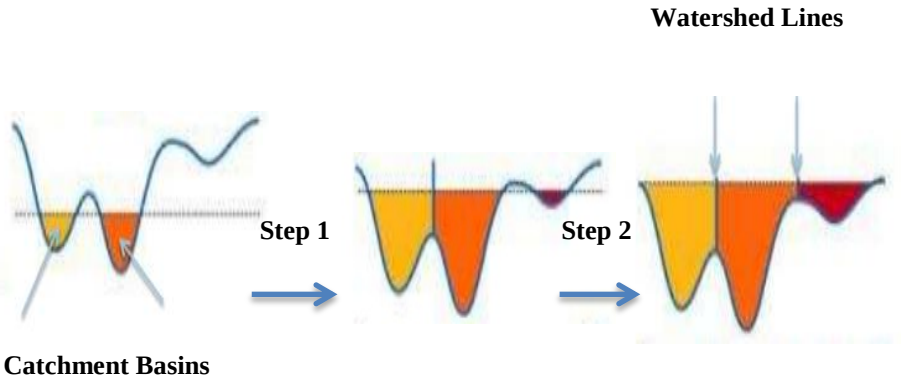


Figure 4.1 : Illustration for the immersion-based watershed algorithm.

The basins are filled with water starting from the local minimum in every basin. When the water sources different basins meet, dams are built so as to prevent merging. Flooding is stopped when the water level reaches the highest peak. In this analogy, catchment basins symbolize different objects and dams correspond to watershed lines which separates the catchment basins.

Algorithmic definition of immersion can be shown with the following recursion [47];

$$X_{h_{min}} = \{\, p \in D \mid f(p) = h_{min} \,\} \quad \textbf{(4.9a)}$$

$$X_{h+1} = X_h \cup MIN_{h+1} \cup [\, IZT_{h+1}(X_h) \setminus T_h], \quad h \in [h_{min},\ h_{max}) \quad \textbf{(4.9b)}$$

The definitions of the terms used in equation (4.9a) and (4.9b) are:

f: $D \rightarrow \mathbb{N}$ where f is digital gray level image whose minimum and maximum intensity values are shown by h_{min}, h_{max}.

X_h: represents computed combination of basins at level *h*

T_{h+1}: is threshold set for level *h+1*. This threshold can be interpreted as a new minimum or expansion of a basin in X_h.

MIN_h: denotes the combination all all regional minima at level *h*

$IZT_{h+1}(X_h)$: is computation of the geodesic influence zone of X_h within T_{h+1} and it is a updating criteria for X_{h+1}.

$$Watershed(f) = \mathrm{D} \setminus X_{h_{max}} \quad \textbf{(4.10)}$$

4.4 The Implementation of the Vincent – Soille Algorithm

The actual implementation of the recursive algorithm in practice can be summarized in two steps, *the sorting step* and *the flooding step*.

In the first step, pixel values are sorted in an ascending order which is later used to access pixels directly for specific level *h*. The flooding step consists of fast breadth-first scanning of all pixels from h_{min} to h_{max} and progress for every gray level by using a First-In-First-Out (FIFO) data structure called *queue*. All pixels which have neighbours with lower than current level *h* are added in the queue. These pixels are labelled with the minimum label of their neighbor. When the flooding is completed until level *h*, this implies that pixels which have equal or less gray level to *h* are assigned to a unique label.

In the next step, pixels in *h+1* have two possibilities either being the catchment basin of *h* or a new local minimum at level *h+1*. Some of the pixels in *h+1* level that are neighbours of *h* level are added into queue. The pixels from the queue are processed sequentially. Pixels in the queue that are not assigned to any existing label, generate new catchment basins. All catchment basins have unique label representing different regions in the image. When two catchment basins come to the point where they can join, a watershed line is generated which a special value. If the point labelled as watershed, then it is not considered again in the next iteration.

The watershed algorithm works well if the objects are in local minima. However, if there are many minima in the image unrelated to regions of the object, then oversegmentation, which is the main issue of watershed algorithms in general, occurs. Besides that, noise in the image might also cause oversegmentation [54].

In this thesis, to address the oversegmentation problem, region merging algorithms based on various criteria, which are described in the next section, are proposed as a postprocessing step.

5. POSTPROCESSING OF ULTRASOUND IMAGES

The watershed algorithms suffer from oversegmentation leading to insufficient segmentation results. To deal with oversegmentation problem, a region merging algorithm is based on the following criteria is introduced.

1. The size of the regions
2. The difference in intensity of regions
3. Statistical information about the regions

Before explaining the details of the region merging algorithm, let us first review how the regions are obtained.

5.1 Extracting the Regions

Once the watershed algorithm is executed on a given ultrasound image, a matrix, which is of the same size as the input image is produced it also includes *labels*. This matrix is called the *label matrix.* Labels are integer values from 1 to *n*, where *n* denotes the maximum value of labels. A group of pixels that have the same label defines a *region*. That is, the number of labels also represents the number of regions. Besides the labels, the matrix also includes zero entries which are interpreted as the watershed lines that separate different regions from each other.

In order to define the association of the regions precisely, a graph theoretic data structure called *region adjaceny graph (RAG)* [55] is employed. RAG creates a graph with nodes $N = \{ N_1, N_2, ..., N_k \}$ where N_i corresponds to region R_i. If there exists a watershed line between regions R_i and R_j, then they are adjacent to each other, which is represented by creating an edge between the corresponding N_i and N_j nodes. Adjacency between nodes can be described in different ways such as 4-neighborhood, 8-neighborhood and etc.

In this study, regions are considered as adjacent if they are separated by watershed lines (by zeros in the label matrix) in the horizontal or vertical direction. Connectivity between pixels is determined according to 8-neighborhood.

5.2 The Region Merging Algorithm

The region merging algorithm which is denoted by Algorithm – RM provides the general outline of the region merging procedures proposed in this thesis. It makes use of the adjacency information between regions is obtained by *RAG* and the label matrix *M*, the output of the watershed algorithm.

The algorithm is based on three threshold parameters *Tsize*, *Tincrement*, *Tmerge*, and a particular merging criterion *C*.

Algorithm – RM

IN: Label matrix *M*, region adjacency graph *RAG*

IN: Threshold values for *Tsize*, *Tincrement* and *Tmerge*

IN: Merging criterion, *C*

1. Select a region *Ri* $\in$ *RAG* who is the smallest in size among all regions that are smaller than *Tsize*.
    a. If no such *Ri* exists, then increase *Tsize;*

$$Tsize = Tsize + Tincrement$$

2. Among the neighbours of *Ri* whose distance to *Ri* according to criterion *C* is smaller than *Tmerge*, select the one with the smallest distance, *Rj*.
    a. If no such *Rj* exists, go to (1) and choose the next smallest *Ri*.
3. Merge *Ri* and *Rj*:
    a. Add the neighbors of *Rj* into *Ri*.
    b. Replace Rj labels in *M* with *Ri*.

OUT: Return the updated label matrix *M'*

Figure 5.1 : The general outline of the region merging procedure.

The parameter values for the threshold *Tsize*, *Tincrement* and *Tmerge* need to be chosen experimentally since they are image-dependent. We show how to analyse the label matrix *M* and region adjacency *RAG* in order to obtain reasonable threshold ranges for tests.

The parameters *Tsize*, *Tmerge* and *C* determine which regions to merge with each other. To choose first region based on size, *Tsize* parameter is used and only regions which are smaller than Tsize are considered. When *Tsize* increases, the number of regions that are merged also increase. That is, the higher the *Tsize* value the higher the number regions merged. If R_i and R_j are too different than each other in terms of criterion *C*, then they should not merge. In order to control it, *Tmerge* parameter is used.

To sum up the idea behind the algorithm RM is that regions smaller than *Tsize* are merged with their neighboring regions that are smaller in distance than *Tmerge* according to the criterion *C*.

Three different criteria is evaluated leading to three different algorithms; RM-based on size, RM-based on intensity, and RM-based on local statistics are described next.

5.3 Region Merging based on Size

In this algorithm, the criterion to merge is solely based on region size, which means number of pixels in the region. Regions that are smaller than *Tsize* form a candidate set for merging. In Step 1, the smallest such region is selected. Then, in Step 2, among the adjacent neighbors of that region whose size is smaller than *Tmerge* is considered, and the smallest one is selected. Finally in Step 3, these two regions are merged.

The assumption of Algorithm – I is if the region is too small, it can belong to another adjacent region instead of existing singlehandedly. However this assumption might not produce good results for segmentation purpose of medical images. If we consider, for example small cyst region in an ultrasound image, it would merge into one of its larger neighbors. So important information for diagnosis would be lost. Therefore, another criterion is needed to prevent such unwanted results. For this purpose, the difference between intensity values is proposed as an additional criterion.

5.4 Region Merging based on Intensity

Intensity information is an important feature to distinguish the regions, hence is useful for region merging. For merging, there must be a kind of proximity between intensity values of regions.

Given two adjacent regions R_k and R_l with N_k and N_l number of pixels and μ_k and μ_l, mean intensity values of regions respectively, the intensity distance $d_{k,l}$ is computed as follows [56].

$$d_{k,l} = \frac{N_k}{N_{new}}|\mu_k - \mu_{new}|^2 + \frac{N_l}{N_{new}}|\mu_l - \mu_{new}|^2 \quad \textbf{(5.1a)}$$

$$N_{new} = N_k + N_l \quad \textbf{(5.1b)}$$

$$\mu_{new} = \frac{N_k\mu_k + N_l\mu_l}{N_{new}} \quad \textbf{(5.1c)}$$

Both RM-based on size and RM-based on intensity algorithms scan the regions in increasing size in Step 1. The main difference between the two is in Step 2. While the former selects the closest adjacent region *in size* bounded with the parameter *Tmerge*, the latter selects the closest adjacent region *in intensity distance* again bounded with *Tmerge*.

While the size information is not explicitly considered in Step 2 of RM-based on intensity, notice how it is implicitly considered as the neighboring regions are scanned in the smallest first order. However the fact that two regions are small is not enough for merging in this case. Their intensity levels must also be close to each other.

Another criterion that can be used for region merging is related to local statistics of the image as shown next.

5.5 Region Merging based on Local Statistics

Local statististics of a given image are calculated according to the definitions given in [31]. The first step is to find a proper window size such that it resembles speckle

statistics explicitly. To achieve this, local mean and local variance on speckle areas are calculated for different window sizes W using the equations below.

$$\mu_{i,j} = \frac{1}{W^2} \sum_{m=-W/2}^{W/2} \sum_{n=W/2}^{W/2} x_{(i-m,j-n)} \quad \textbf{(5.2)}$$

$$\sigma_{i,j}^2 = \frac{1}{W^2} \sum_{m=-W/2}^{W/2} \sum_{n=W/2}^{W/2} (x_{(i-m,j-n)} - \mu_{i,j})^2 \quad \textbf{(5.3)}$$

where $x_{i,j}$ is the pixel value at the location (i,j).

Based on these equations, local mean and local variance values for different (i,j) locations from speckle regions are obtained for different window sizes. Then, for every window size, averaged mean and variance values are calculated. In principle, the ratio between the variance and the mean, denoted by *alpha*, increases with window size up to a certain point and then stabilizes. The all window sizes where this ratio saturates are the proper window size for the image at hand.

Once the proper window size on the speckle areas only is found, alpha values for the entire image using this window size can be calculated. Alpha values are found for each pixel location (i,j) can be used for the purposes of region merging.

The idea of RM-based on local statistics is similar to that of RM-based on intensity except that alpha values of each pixel are used instead of the intensity values when calculating the distance between two regions.

6. TEST RESULTS

Section 3 outlined the preprocessing algorithms to improve the image quality, Section 4 gave an overview of the watershed algorithm, and Section 5 proposed novel region merging algorithms for postprocessing.

This section combines all of these methods together and provides a comprehensive test study using reference ultrasound images. These images were selected prior to the design, implementation and the experimentation of these methods.

The complete set of procedures used in this thesis is illustrated in **Figure 6.1**.

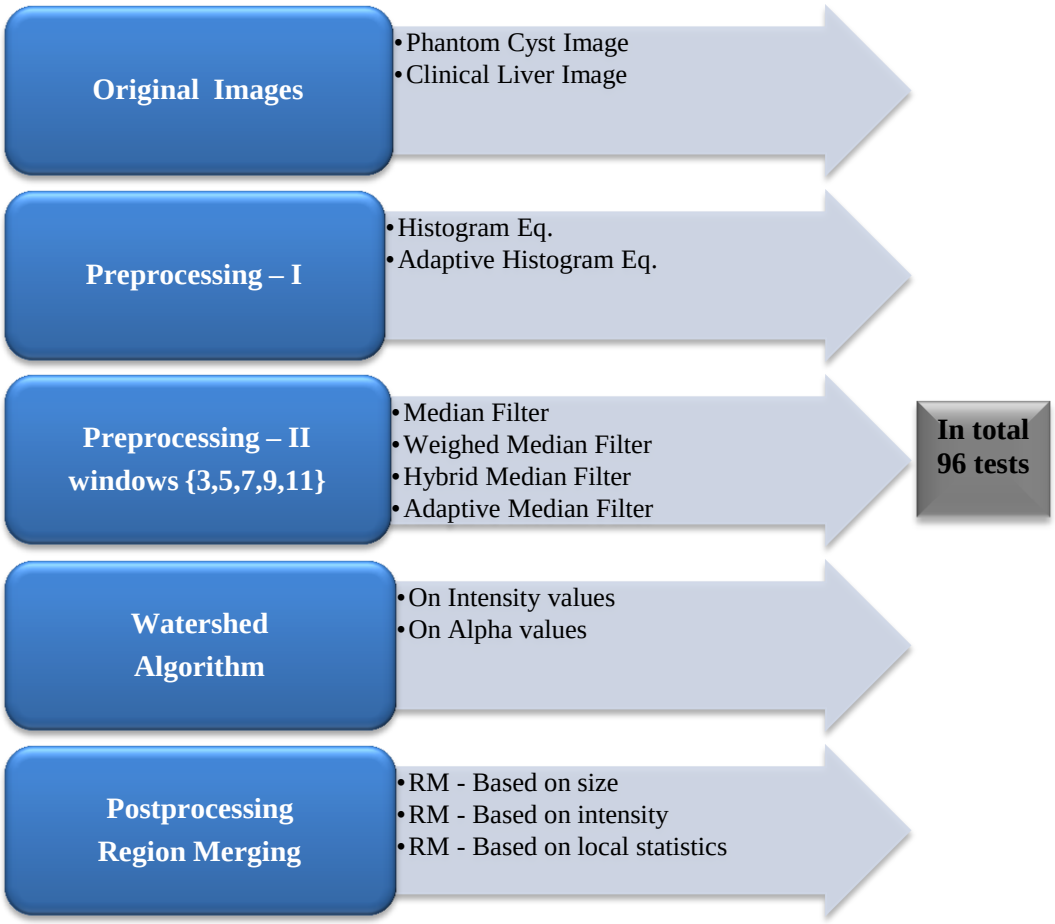


Figure 6.1 : The overall structure of the algorithms used in this thesis.

The existence of various choices in the procedure leads to 96 different image segmentation tests as it is denoted in figure. This is without counting the impact of parameter tuning for the window size in preprocessing and the selection of threshold values for postprocessing. In combination, the total number of experiments conducted amounts more than a thousand different tests. All experiments are run using Matlab.

The goal of the experiments are twofold:

1. To evaluate the effect of each component in terms of how important and useful they are for image segmentation, and
2. To find the best segmentation for the ultrasound image at hand that helps identifying important regions.

The tests are classified according to the input that the watershed algorithm is executed with which are

i. the intensity values of the original images,

ii. the intensity values of the preprocessed images, and

iii. the local statistical values are computed from the original images.

Two indicators are used to evaluate the results of the different segmentation algorithms. The first is the number of regions, denoted by *numR.* Ideally, we would like to obtain only the regions that are of importance in reality. Therefore, the smaller the number of regions, the better the segmentation is. The number of region alone is not enough, but the regions should also corrrespond to these locations in the image with special structure. The best way to assess this criterion is to visually inspect the images which is our second indicator of quality.

6.1 The Ultrasound Test Images

In this study, all tests were run on the two ultrasound images. These test images were originally generated and used by M. Karaman (advisor of this thesis) in [31] and [57]. The cyst image was reconstructed from raw RF data acquired from a custom phantom using 128-element transducer array. The liver image was captured by M. Karaman from his liver using a commercial digital ultrasound scanner.

As shown in **Figure 6.2**, the cyst image exhibits 4 local regions with special regions referenced as cyst1, cyst2, csyt3 and cyst4 from left to right. Specifically, cyst4, the region at the bottom right, is the most blurry, hence it is the most indicative of the quality of the segmentation algorithms whether they can retain this region of the original image.

The liver image includes branches and the outer boundaries. In particular, difficulty of segmenting this image is to dealing with thin lines.

These images and their contents are shown in figure below.

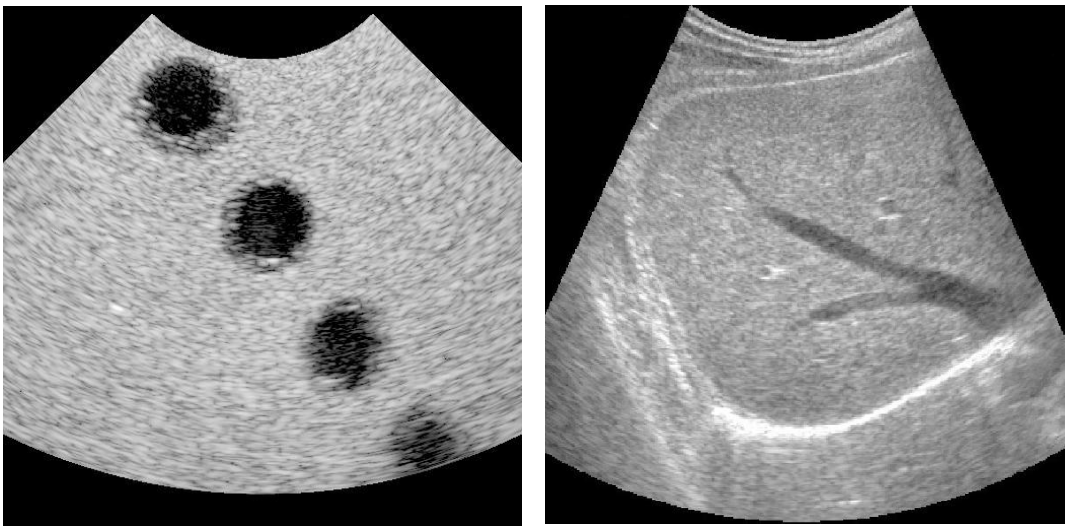


Figure 6.2 : (Left) Cyst phantom image [57] and (Right) Clinical liver image [31].

6.2 The Results of the Watershed Algorithm

6.2.1 The watershed algorithm on the original images

The watershed algorithm (WS) is applied directly on the original images without any preprocessing. These tests result in 7132 different regions for the cyst image and 27440 different regions for the liver image which form our baselines for comparison. As shown in **Figure 6.3**, there are thousands of regions. This is a case of oversegmentation.

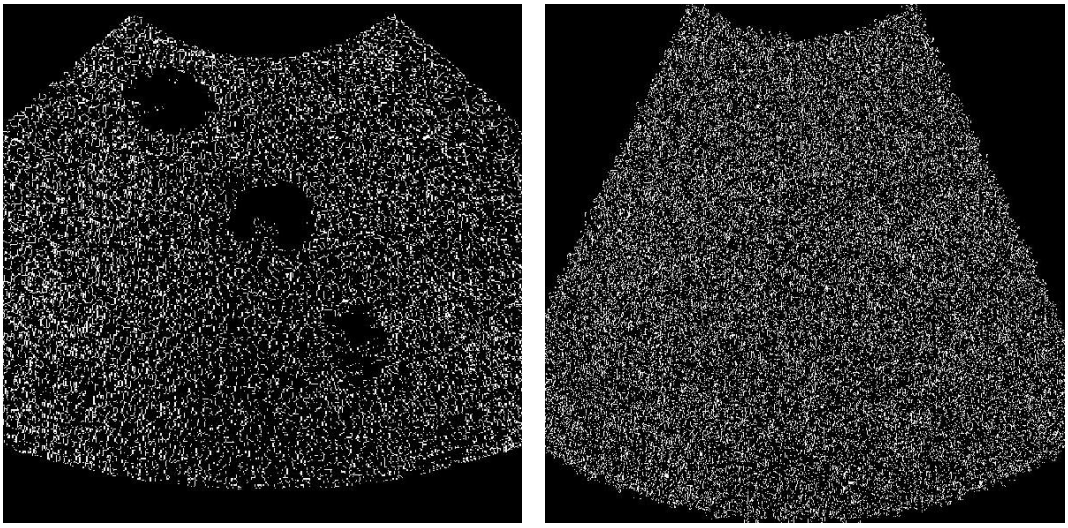


Figure 6.3 : The results of the watershed algorithm without preprocessing methods (left) the cyst and (right) the liver images.

For the cyst image, it can be noticed that the information about cyst4 is almost lost. For the liver image, there are no segmented regions at all, all details are lost.

The question is whether preprocessing algorithms can help the segmentation by reducing the noise in the original images and boost the performance of the watershed algorithm, which is studied next.

6.2.2 The watershed algorithm on the preprocessed images

There are two components in preprocessing (i) *histogram equalization*, and (ii) *filtering methods*.

For histogram equalization, the standard histogram equalization and adaptive histogram equalization are tested.

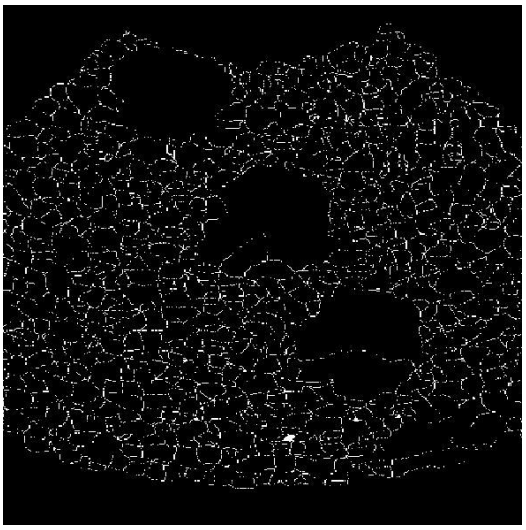
For filtering method, median filter, hybrid median filter, weighted median filter and adaptive median filter are tested.

Results of the filtering methods are tested with and without histogram equalization methods. This results in 2 x 4 = 8 combinations. Plus, the filtering methods have a parameter for setting the window size. We tested different alternatives using window size, W = {3, 5, 7, 9, 11} for each of the filtering method. In total, 40 different tests are conducted. We found out that filtering methods yield better results when they are applied with standard histogram equalization. On the other hand, adaptive histogram equalization distorts the filtered images. Therefore, standard histogram equalization is used by default for the rest of all tests. The results for adaptive median filtering are

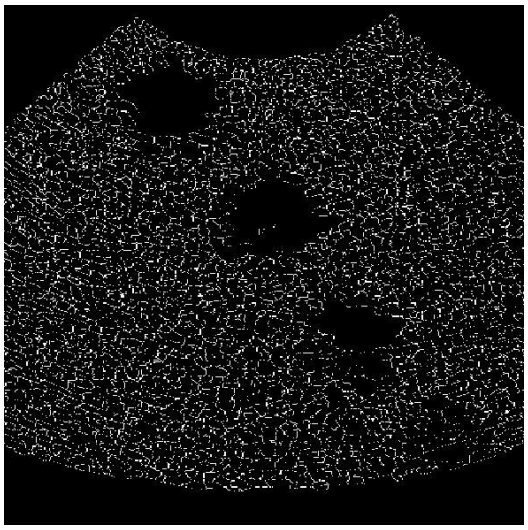
omitted because, interestingly, it resulted in *more* number of regions than using watershed with no preprocessing.

Our evaluation is based on both reducing the number of regions from 7132 and 27440 and at the same time, keeping visual quality means the preservation of the details of the images.

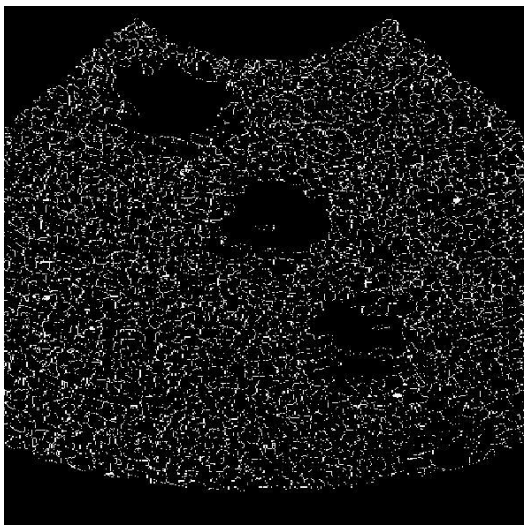
We present the results for the best settings for each of the filtering method with different window sizes for the cyst and the liver image respectively.



Median Filter – W = 9
numR = 907



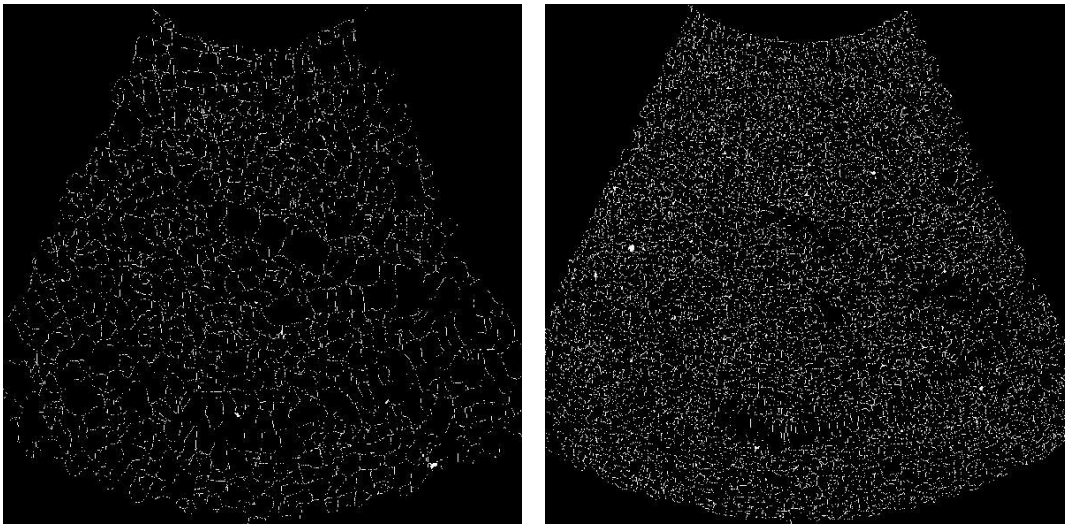
Weighted Median Filter – W = 3
numR = 2764



Hybrid Median Filter – W = 5
numR = 2945

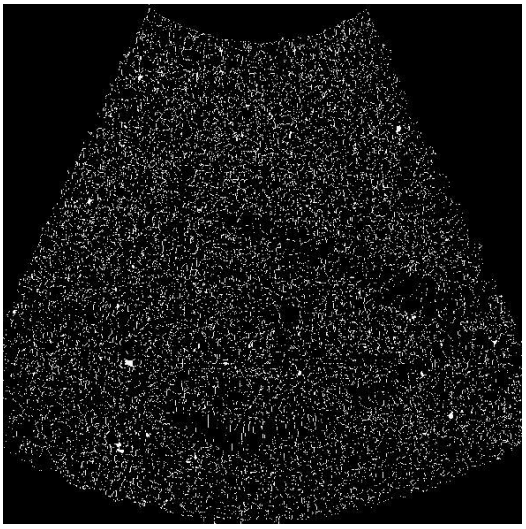
Figure 6.4 : The results of the watershed algorithm on the preprocessed cyst image using different filtering methods with different window sizes.

For the cyst image, median filter with window size 9 gives the best result in terms of the number of regions among all other filtering methods with different window sizes. Number of regions, which was 7132 for the original cyst image, is now reduced to 907 using preprocessing. More importantly, notice how, cyst4 region becomes distinctive after preprocessing which was lost in the watershed algorithm on the original image.



Median Filter – W = 11
numR = 1099

Weighted Median Filter – W = 5
numR = 5136



Hybrid Median Filter – W = 5
numR = 6626

Figure 6.5 : The results of the watershed algorithm on preprocessed liver image using different filtering methods and window sizes.

For the liver image, median filter gives the best result again in terms of number of regions. With this filtering method, number of regions which is 27440 for original

liver image is reduced to 1099. With any filtering methods shown in **Figure 6.5**, details of the liver image become clear compared to the watershed algorithm on the original image (**Figure 6.3**).

Consequently, notice that after preprocessing methods, number of regions decreased. This shows preprocessing methods might help dealing with oversegmentation.

6.2.3 The watershed algorithm on the local statistics

Another alternative is to run the watershed algorithm not on the intensity values of the (preprocessed) image itself, but on the local statistics computed from intensity values. In Section 5.5 we described how to obtain such local statistics. Note that these computations are done on only speckle areas not on the whole image.

As mentioned, a proper window size needs to be determined. For that purpose, we tried different window sizes ranging from 3 to 25 using an increment of 2. The ratio of variance to mean is plotted against different window sizes for the cyst and the liver image in **Figure 6.6**.

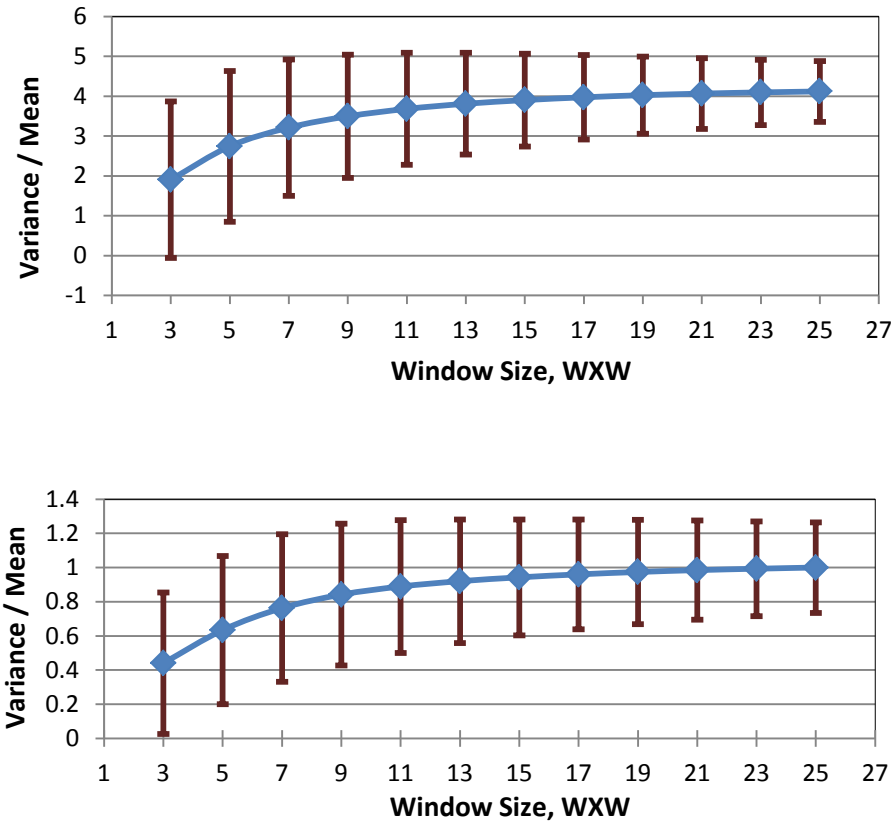
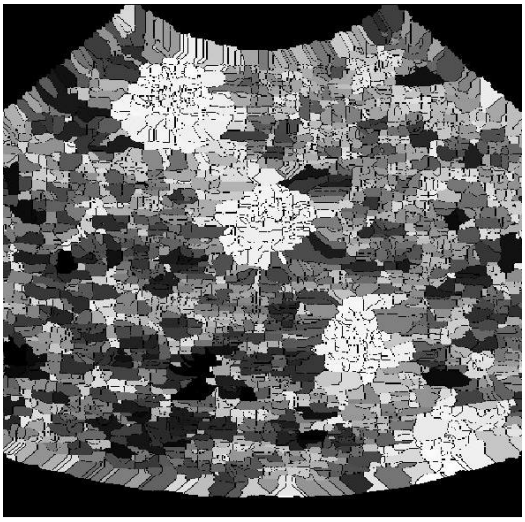


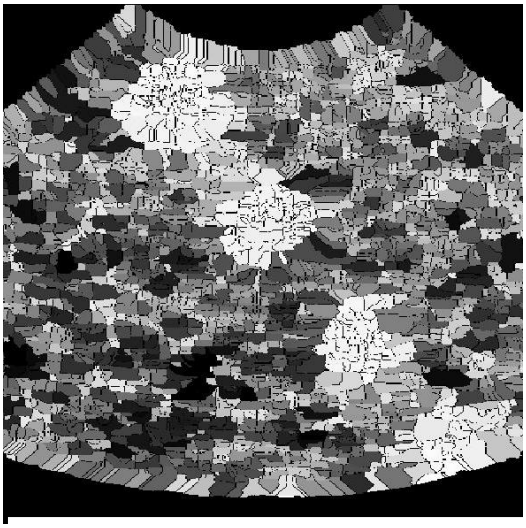
Figure 6.6 : The local statistics computed for different window sizes (top) the cyst and (bottom) the liver images.

All window sizes after variance/mean ratio stabilizes, are proper windows for that image. For the cyst image window size 15 and 17 are chosen whereas window size 13 and 15 are selected for the liver image.

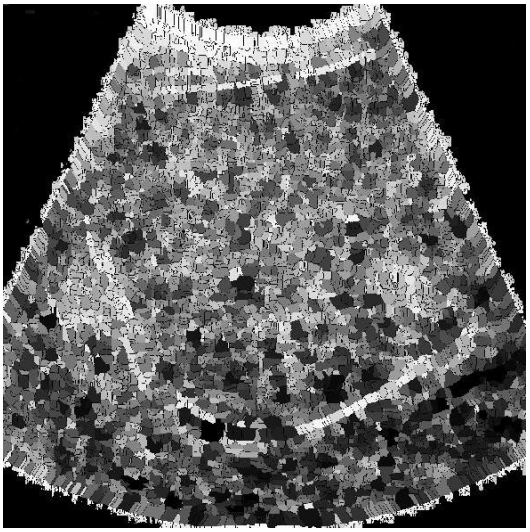
Next, local statistics for the entire images are obtained using these window sizes. Then the watershed algorithm is run on these alpha values (variance/mean) found for each pixel. **Figure 6.7** shows the results for the cyst and liver images.



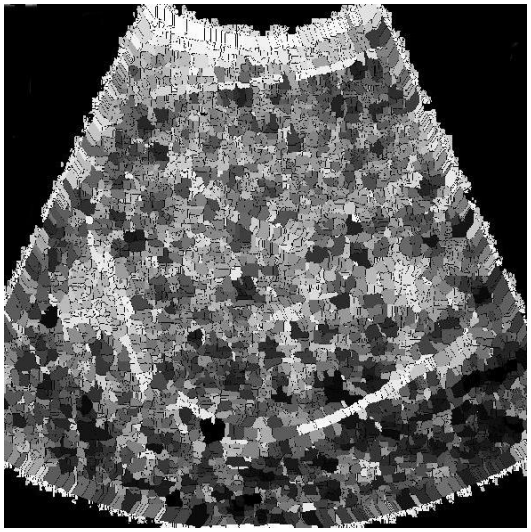
Window Size = 15
numR = 2818



Window Size = 17
numR = 2724



Window Size = 13
numR = 7551



Window Size = 15
numR = 6882

Figure 6.7 : The watershed algorithm on the the local statistics (top) the cyst and (bottom) the liver images.

From these results, it can be seen that watershed algorithm on alpha values have less number of regions compared to the watershed algorithm on intensity values. Moreover, number of regions inversely related to window size for both image. However, preserving region features, which is another important criterion for successful segmentation is not satisfied for this case.

6.3 The Results of the Region Merging Algorithm

Based on the outline of the RM algorithm given in Section 5.2, different RM variants are tested in terms of their effectiveness in reducing the number of regions and preserving the localities.

Both the label matrices produced by the intensity based watershed algorithm with preprocessing (results given in Section 6.2.2), and the alpha value based watershed algorithm (results given in Section 6.2.3) are used as an input for the region merging process.

There are 3 different RM variants; RM-based on size, RM-based on intensity, RM-based on local statistics applied on 2 types of inputs; intensity and alpha values for both the cyst and liver image.

6.3.1 Parameter selection for the region merging

RM algorithm depends on three parameters; *Tsize, Tincrement* and *Tmerge*. As these parameters are image dependent, analysis is necessary to set their values. Consider, for example, the analysis of region sizes for Median Filter – W = 9 is used for the cyst image.

Table 6.1: Statistics on the number of pixels in regions.

Median Filter W = 9	
Total Number of Regions	907
Total Number of Pixels	512 x 512
Minimum region size	1
Maximum region size	37166
Mean region size	277.46
Median region size	136
Std. Deviation region size	1662

Notice that difference between the median region size and the mean region size relatively large. This is due to the exterior of the image as well as the watershed claimed large region, which skew the average size.

The statistics for other configurations and for the liver image are similar. Therefore, *Tsize* is chosen as

$$Tsize = \{100, 200, 400, 800, 1600\} \quad \textbf{(6.1)}$$

The parameter *Tincrement* determines the number of iteration for merging based on *Tsize*. Its value is set [1..10] such that

$$Tincrement = \{1, 4, 10\} \quad \textbf{(6.2)}$$

The other parameter *Tmerge* is used to decide whether to merge two regions or not. This threshold is bounded with the maximum distance that is possible according to each criteria.

For the size criterion, the maximum distance is:

$$maxDist = Size_{max} - Size_{min} \quad \textbf{(6.3)}$$

$Size_{max}$: the maximum sized region, excluding exterior regions of the images and the watershed lines.

$Size_{min}$: the minimum sized region (this was almost always 1 in our tests)

For the intensity distance, the maximum value:

$$maxDist = distance(I_{max} - I_{min}) \quad \textbf{(6.4)}$$

I_{max}: the maximum intensity of the image.

I_{min}: the minimum intensity of the image.

distance: based on the distance function given in Equation 5.1a, 5.1b and 5.1c.

For the alpha distance, the maximum value is

$$maxDist = distance(A_{max} - A_{min}) \quad \textbf{(6.5)}$$

A_{max}: the maximum alpha value of the image.

A_{min}: the minimum alpha value of the image.

distance: based on the distance function given in Equation 5.1a, 5.1b and 5.1c.

Based on these equations, the *Tmerge* parameter is adapted for each input image as

$$Tmerge = \lambda * maxDist \quad \textbf{(6.6)}$$

λ: the bounding coefficient which varies between [0..1]. The tests use,

$$\lambda = \{0.1, 0.2, 0.4, 0.8\} \quad \textbf{(6.7)}$$

In addition to previous configurations, 5 different values for *Tsize*, 3 different values for *Tincrement* and 4 different values for *Tmerge* are considered, yielding extensive test cases. Only the best results of both images are presented. It is interesting to note that, all of the best results shown in the following use *Tincrement* equal to 1.

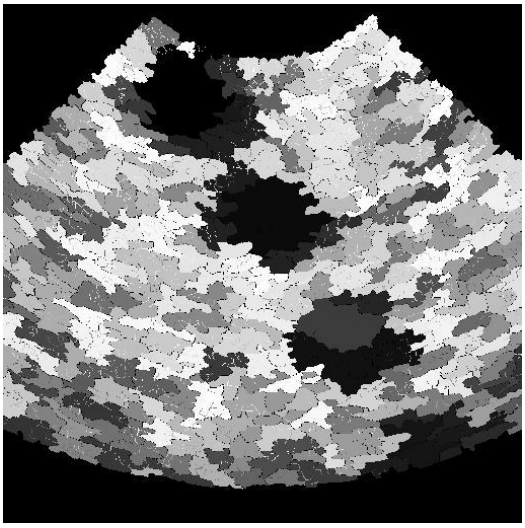
6.3.2 Region merging based on size on the preprocessed images

RM-based on size uses the size criterion for merging as described in Section 5.3.

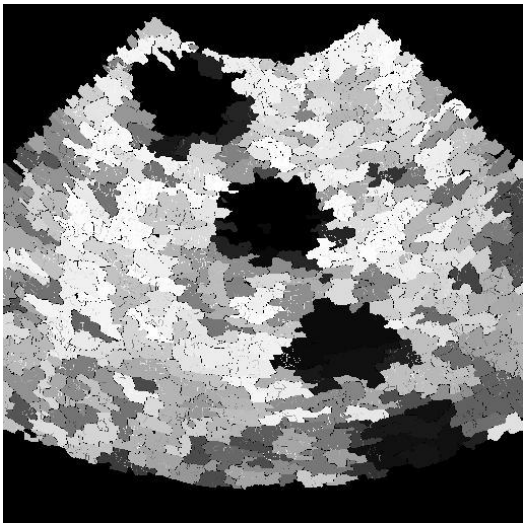
For the cyst image, weighted median filter with window size 3 gives the best result with 303 number of regions which was 2764 for the preprocessed image. Moreover, hybrid median filter with window size 7 gives second best result by decreasing the number of the regions from 3488 to 338. Interesting case for this result is for hybrid median filter, the best result is obtained with window size 5 in terms of number of regions (**Figure 6.4**), but with postprocessing its result becomes beyond hybrid median filter, window size 7.

For the liver image, the best result is obtained with median filter with window size 11 where the number of regions is reduced from 1099 to 515. Also weighted median filter with window size 5 is able to to reduce the number of regions from 6626 to 1540 which is the second best result for this case.

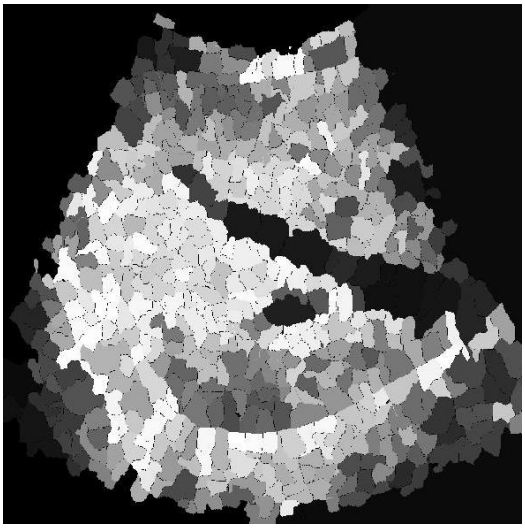
Overall, these mentioned results are shown in the following figure.



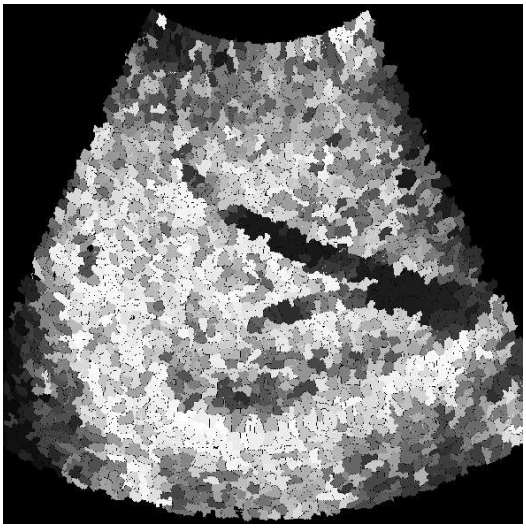
Weighted Median Filter – W = 3
Tsize = 400 λ = 0.4
numR = 303



Hybrid Median Filter – W = 7
Tsize = 400 λ = 0.4
numR = 338



Median Filter – W = 11
Tsize = 400 λ = 0.1
numR = 515



Weighted Median Filter – W = 5
Tsize = 400 λ = 0.2
numR = 1540

Figure 6.8 : The results of the region merging on intensity values based on size criterion (top) the cyst and (bottom) the liver images.

6.3.3 Region merging based on size on the local statistics

RM using alpha values resulted in degraded performance. Example cases are shown in the following figure where regions are mixed together. All other results have equally poor quality. Therefore, we omit the results of other local statistics at here.

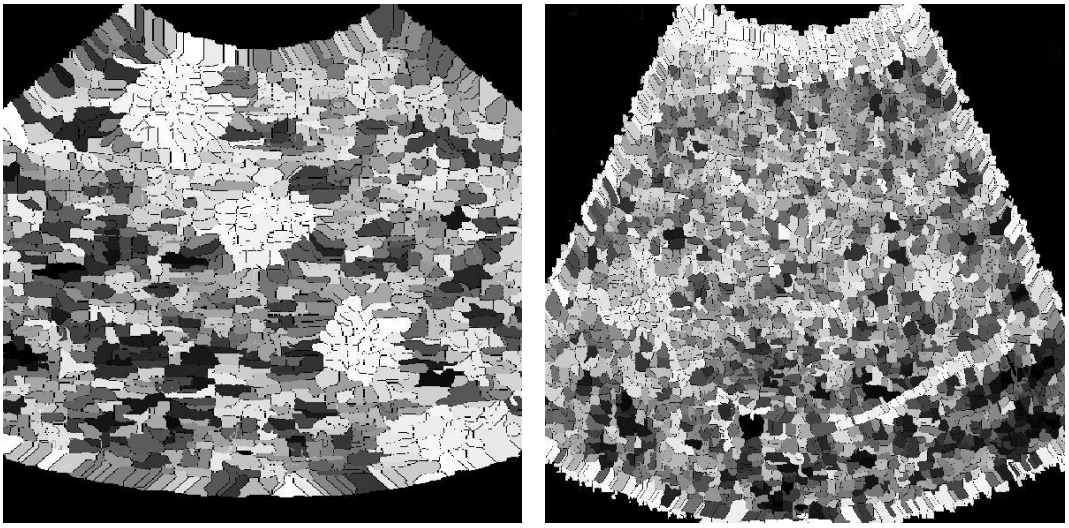
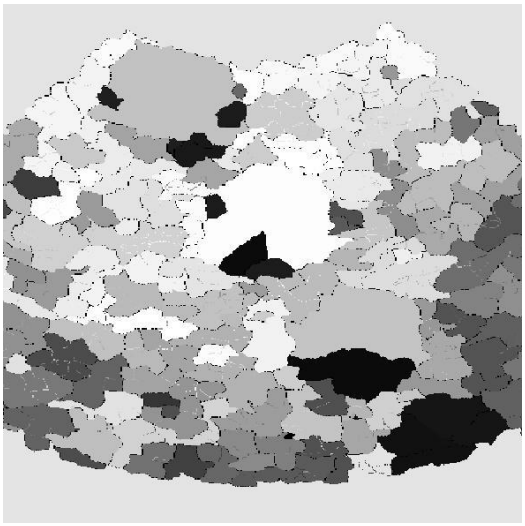


Figure 6.9 : The results of the region merging based on size criterion on local statistics, (top) the cyst and (bottom) the liver images.

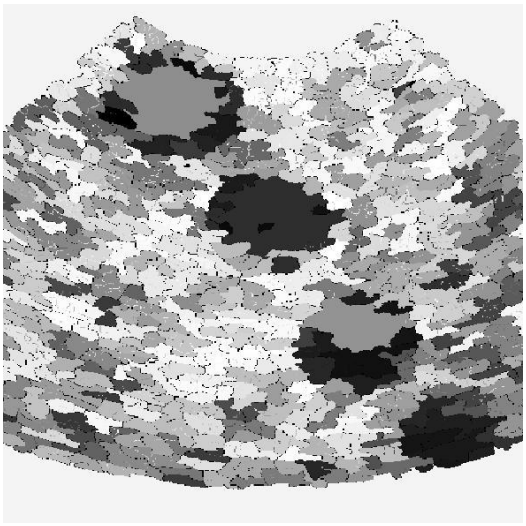
6.3.4 Region merging based on intensity on the preprocessed images

RM-based on intensity uses intensity values as the merging criterion, which is described in Section 5.4. In these experiments, median filter gives the best results for both the cyst and the liver image. These mentioned results are shown in **Figure 6.10**.

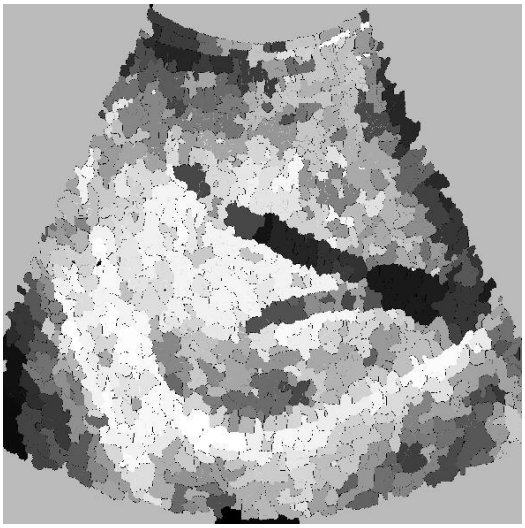
In this case, the number of regions for cyst image is around ~160 whereas it is around ~550 for the liver image. It should be noted that for the cyst image, the number of regions for that filter without postprocessing are 907. For the liver image, this baseline result in terms of the number of regions is 1540.



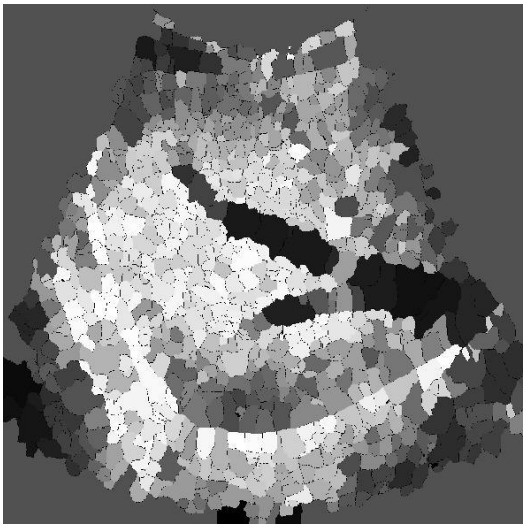
Median Filter – W = 9
Tsize = 800 λ = 0.8
numR = 164



Weighted Median Filter – W = 5
Tsize = 200 λ = 0.1
numR = 506



Median Filter – W = 5
Tsize = 1600 λ = 0.2
numR = 556



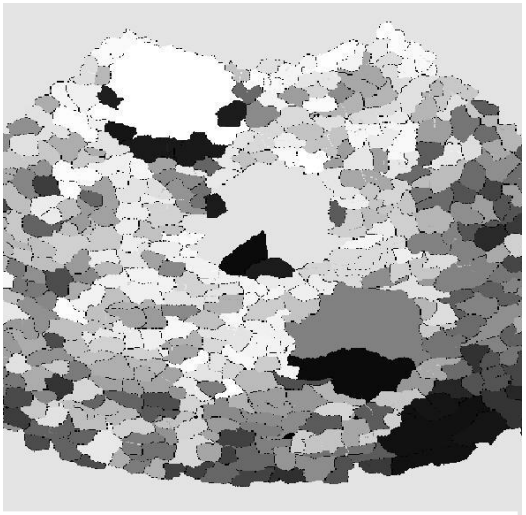
Median Filter – W = 11
Tsize = 100 λ = 0.4
numR = 862

Figure 6.10 : The results of the region merging based on intensity criterion on intensity values, (top) the cyst and (bottom) the liver images.

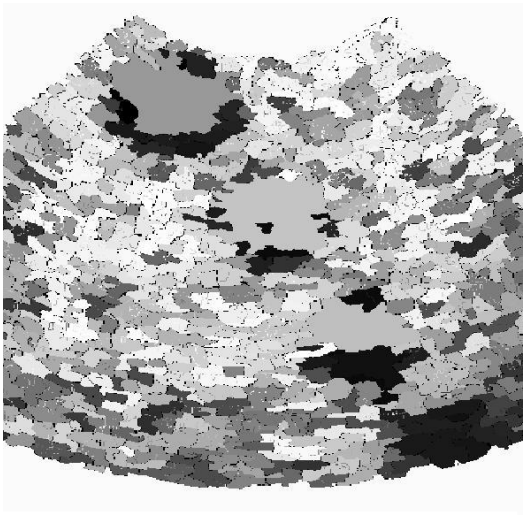
6.3.5 Region merging based on local statistics on the preprocessed images

As described in Section 5.5, RM-based on local statistics uses the alpha values as the merging criterion. For the cyst image, number of the regions is reduced from 907 to 397 with median filter with the window size 9. For the liver image, the best result is

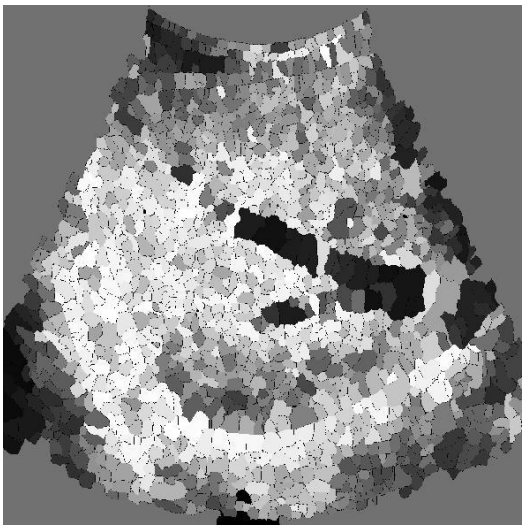
obtained with median filter, window size 11. For this case, number of regions are reduced from 1099 to 856.



Median Filter – W = 9
Tsize = 200 λ = 0.8
numR = 397



Hybrid Median Filter – W = 9
Tsize = 200 λ = 0.1
numR = 557



Median Filter – W = 7
Tsize = 100 λ = 0.1
numR = 1379

Median Filter – W = 11
Tsize = 1600 λ = 0.4
numR = 856

Figure 6.11 : The results of the region merging based on local statistics criterion on intensity values, (top) the cyst and (bottom) the liver images.

7. CONCLUSION

Section 3 outlined the preprocessing algorithms to improve the image quality, Section 4 gave an overview of the watershed algorithm, and Section 5 proposed novel region merging algorithms for postprocessing.

This thesis studied the segmentation of ultrasound images. The overall approach consists of three main components: preprocessing, the watershed algorithm for segmentation, and postprocessing for region merging.

Preprocessing is used to deal with speckle noise in ultrasound images. Two types of preprocessing methods, and their combination, are considered. For the former, we tried both standard histogram equalization and adaptive histogram equalization. For the latter, four different filtering methods are considered; median filter, weighted median filter, hybrid median filter and adaptive median filter.

Next, for segmentation of the image, the watershed algorithm, in particular, the Vincent–Soille algorithm is implemented. An inherent drawback of this approach was the oversegmentation issue which we also observed on the test images. The initial numbers of regions were ~7000 for the phantom cyst image and ~27000 for the clinical liver which were rather high.

In postprocessing, the goal was twofold; reducing the number of regions while preserving the segmentation of local regions. We proposed a general region merging algorithm, RM, that can be specialized into three different variants. Each variant focused on specific properties related to the image; the size of the regions, the intensity values, and the local statistical information. As the input matrix for the region merging phase, two alternatives are tested. First, the intensity values of the preprocessed image, and second, the alpha values of the original images.

Overall, our procedure manifested 96 different tests (excluding the parameter tuning of each). Extensive experimental evaluation using more than a thousand tests are conducted using on both the cyst and liver images.

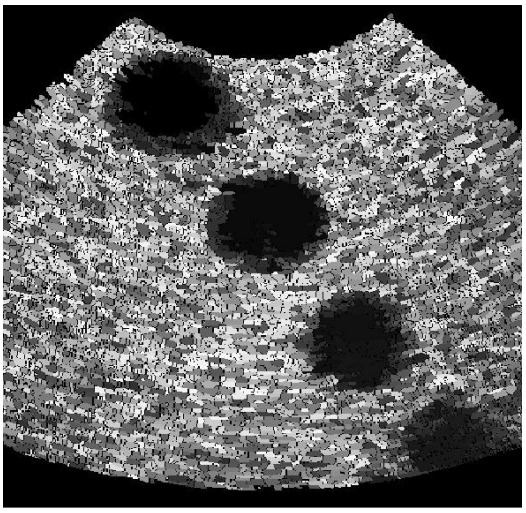
Our first finding is that histogram equalization helps improving the quality of the ultrasound images, hence it is used by default in most experiments as part of the preprocessing. On the contrary, adaptive histogram equalization performed poorly.

Combining this with filtering methods yielded a further improvement with the exception of adaptive median filtering. It is interesting to note that adaptive approaches such as adaptive histograms, and adaptive filtering produced poor results for the selected ultrasound images.

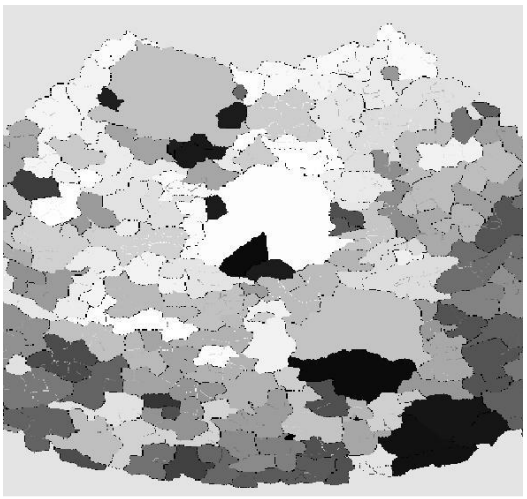
Regarding the watershed approach, our implementation of the Vincent-Soille algorithm performed well. For the tested images, it produced the exact same number of regions compared to the default watershed implementation in Matlab. This is also interesting since the default Matlab function uses a different reference algorithm.

In region merging, three criteria are used such as size, intensity values and local statistics. In terms of run time, postprocessing tests take around ~30 minutes. The best result, for the cyst image are obtained using the region merging algorithm based on intensity. The initial number of regions is reduced from 7132 down to 164 regions while local regions were preserved visually. This is a 40 fold reduction in size. For the liver image, the best result is obtained with size-based region merging. It reduces number of regions from 27740 to 515 regions. The final results are shown in next page in **Figure 7.1**.

Overall, we acknowledge the oversegmentation problem of the watershed algorithm and mitigate its effect using multiple criteria for merging. It is apparent from our results that segmentation problem goes hand in hand with preprocessing and postprocessing both of which considerably improved the results. Therefore, we strongly suggest the researchers to consider these components together. In combination, our procedure can be considered as a sound approach for ultrasound image segmentation as supported by our computational results.

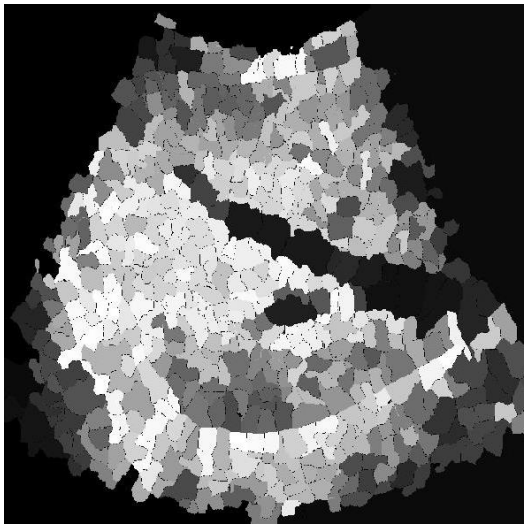


Without pre- and postprocessing
numR = 7132



Median Filter – W = 9
RM-based on Intensity
numR = 164

Without pre- and postprocesing
numR = 27440



Median Filter – W = 11
RM-based on Size
num R = 515

Figure 7.1 : The best resultant images with pre- and postprocessing.

8. FUTURE WORK

Our approach, as illustrated in **Figure 6.1**, lays the groundwork for ultrasound images segmentation using the watershed algorithm based on its Vincent-Soille implementation. While it is in complete working state with its several components, it also leaves many room for potential future research.

First of all, we showed the critical importance of the speckle reduction in segmentation. This is achieved using histogram equalization and filtering. There exists several other speckle suppression methods, even more complex ones than the filtering methods. Further work is needed to expand on this front.

Secondly, the Vincent–Soille algorithm has some drawbacks. One of them is difficulty of labelling pixels in a plateau, which results in incomplete and thick boundaries and the other one is discrete or disconnect watershed lines. Different watershed algorithms, which are mentioned in Section 4, can be used for a comparison.

Finally, for region merging, several interesting criteria remain to be explored in postprocessing. Some of these criteria are; the uniformity of regions boundaries, the connectedness of regions, the perimeter of regions and etc. In all variants of the region merging algorithms, the initial selection of the regions were fixed according to size criterion. Other criteria such as intensity or local statistics can be chosen for first region which might produce different (better) results.

Moreover, region merging algorithm has a number of parameters such as *Tsize, Tincrement* and *Tmerge* which totally depend on the image. These parameters greatly influence the results found in practice. However, parameter tuning requires conducting various tests and is time consuming.

REFERENCE

[1] **Gonzalez**, **R. C., Woods, R. E.** (2008). *Digital Image Processing*. 3rd Edition Peardon Prentice Hall, Upper Saddle River, MA.

[2] **O'Donnel, L**. (2001). Semi-Automatic Medical Image Segmentation. *PhD Thesis,* MIT, Cambridge, MA.

http://people.csail.mit.edu/lauren/publications/odonnell_ms_thesis.pdf

[3] **Jacobson, J. A.** (2007). *Fundamentals of Musculoskeletal Ultrasound*. Elsevier Health Sciences.

[4] **Noble, J. A., Boukerroui, D.** (2006). Ultrasound Image Segmentation: A Survey. *IEEE Trans. On Medical Imaging, Vol. 25, No.8, pp. 987-1010.*

[5] **Burger, M., Modersitzki, J., Tenbrinck, D.** (2014). Mathematical Methods in Biomedical Imaging. *GAMM-Mitteilungen Vol. 37, No. 2, pp. 154-183*.

[6] **Dass, R., Swapna, D.** (2012). Image Segmentation Techniques. *International Journal of Electronics & Communication Technology, Vol. 3, Issue 1, pp. 66-74.*

[7] **Tamiselvi, P. R.** (2014). Image Segmentation Technique for Medical Ultrasound Images. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences (IJIRMPS), Vol. 2, Issue 3.*

[8] **Pham, D. L., Xu, C., Prince, J. L.** (1998). A Survey of Current Methods in Medical Image Segmentation. *Annual Review of Biomedical Engineering.*

[9] **Kontropoulos, C., Pitas, I**. (2003). Segmentation of Ultrasonic Images using Support Vector Machines. *Pattern Recognition Letters-Special issue: Ultrasonic Image Processing and Analysis, Vol. 24, Issue 4-5, pp.715-727.*

[10] **Deka, B., Ghosh, D.** (2006). Watershed Segmentation for Medical Ultrasound Images. *IEEE International Conference on Systems, Man. and Cybernetics, Vol. 4, pp. 3186-3191.*

[11] **Li, L., Fu, Y., Bai, P., Mao, W.** (2009). Medical Ultrasound Image Segmentation based on Improved Watershed Scheme. *Proceedings of the 3rd International Conference on Bioinformatics and Biomedical Engineering, ICBBE, pp.1-4.*

[12] **Huang, Y., Chen, L.** (2004). Watershed Segmentation for Breast Tumor in 2-D Sonography. *Ultrasound in Medicine & Biology Vol. 30, pp.625-632.*

[13] **Gomez, W., Leija, L., Alvarenga, A., Infantosi, A., Pereira, W.** (2010). Computerized Lesion Segmentation of Breast Ultrasound based on Marker-Controlled Watershed Transformation. *Medical Physics Vol. 37, pp. 82-95.*

[14] **Zhang, H., Fritts, J., E., Goldman, S. A.** (2008). Image Segmentation Evaluation: A Survey of Unsupervised Methods. *Computer Vision and Image Understanding (CVIU), Vol. 110, No. 2, pp. 260-280.*

[15] **Sikka, K., Deserno, T. M.** (2010). Comparison of Algorithms for Ultrasound Image Segmentation without Ground truth. *Proceedings of SPIE, Vol. 7627.*

[16] **Szabo, T. L.** (2002). *Diagnostic Ultrasound Imaging*. Elsevier Academic Press.

[17] **William, R., Hendee, R.** (2002). *Medical Imaging Physic*. Wiley-Liss 4th Edition.

[18] **Gibss, V., Cole, D., Sassano, A.** (2011). *Ultrasound Physics and Technology: How, Why and When.* Elsevier Health Sciences, pp. 92-94.

[19] **Bhargava, S.** (2002). *Principles and Practise of Ultrasonography*. Jaypee Brothers Medical Publishers.

[20] **Valdueza, J. M., Scheiber, S. J., Roehl, J. E., Klingebiel, R.** (2008). *Neurosonology and Neuroimaging of Stroke*. Thieme Publisher 1st Edition.

[21] **Tole, N. M., Ostensen, H.** (2005). *Basic Physics of Ultrasonographic Imaging.* World Health Organization.

[22] **Loizou, C., Pattichis C.** (2008). *Despeckle Filtering Algorithms And Software for Ultrasound Imaging.* Morgan & Claypool Publishers.

[23] **Burckhardt, C. B**. (1978). Speckle in Ultrasound B-mode Scans. *IEEE Trans. On Sonics and Ultrasonics, Vol. 25, pp.1-6.*

[24] **Abbot, J., Thurstone, F.** (1979). Acoustic Speckle: Theory and Experimental Analysis. *Ultrasound Imaging, Vol. 4, pp. 303-324.*

[25] **Eltoft, T.** (2006)**.** Modelling the Amplitude Statistics of Ultrasonic Images. *IEEE Transactions on Medical Imaging, Vol. 25, pp. 229-240.*

[26] **Narayanan, V., Shankar, P. M., Reid, J. M.** (1994). Non-Rayleigh Statistics of Ultrasonic Backscattered Signals. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on, Vol. 41, pp. 845–852.*

[27] **Prager, R. W., Gee, A. H., Treece, G. M., Berman, L. H.** (2003). Decompression and Speckle Detection for Ultrasound Images using the Homodyned *K*-Distribution. *Pattern Recognition Letters, Vol. 24, pp.705–713.*

[28] **Jakeman, E., Tough, R. J. A.** (1987). Generalized *K*- Distribution: A Statistical Model for Weak Scattering. *Journal of the Optical Society of America, Vol. 4, No. 9, pp. 1764–1772.*

[29] **Joel, T., Sivakumar, R.** (2013). Despeckling of Ultrasoud Medical Images: A Survey. *Journal of Image and Graphics, Vol. 1, No. 3, pp. 161-165.*

[30] **Sanches, J. M., Andrew, F**. (2011). *Ultrasound Imaging Advances and Applications.* Springer-Verlang, New York.

[31] **Karaman, M., Kutay, M. A., Bozdagi, G.** (1995). An Adaptive Speckle Suppression Filter for Medical Ultrasonic Imaging. *IEEE Trans. on Medical Imaging, Vol. 14, No. 2, pp. 283-292.*

[32] **Loupas, T., McDicken, W. N., Allan, P. L.** (1987). Noise Reduction in Ultrasonic Images by Digital Filtering. *Brit. J. Radiol., Vol. 60, pp. 389-392*.

[33] **Loupas, T., McDicken, W. N., Allan, P. L.** (1989). An Adaptive Weighted Median Filter for Speckle Suppression in Medical Ultrasonic Images. *IEEE Trans. Circuits Syst., Vol. 36, No. 1, pp. 129-135.*

[34] **Lee, J. S.** (1981). Speckle Analysis and Smoothing of Synthetic Aperture Radar Images. *Computer Graphics Image Process., Vol. 17, pp. 24–32*.

[35] **Kuan, D. T., Sawchuk, A., Strand, T. C., Chavel, P.** (1987). Adaptive Restoration of Images with Speckle. *IEEE Trans. Acoustic, Vol. ASSP-35, pp. 373–383.*

[36] **Frost, V. S.** (1982). A Model for Radar Images and Its Application for Adaptive Digital Filtering of Multiplicative Noise. *IEEE Trans. Pattern Analysis, Machine Intelligence, Vol. 4, No. 2, pp. 157–165.*

[37] **Hadhoud, M. M.** (1999). Image Contrast Enhancement using Homomorphic Processing and Adaptive Filters. *Radio Science Conference, pp.89-92.*

[38] **Ali, S. M., Burge, R.** (1988). New Automatic Techniques for Smoothing and Segmenting SAR Images. *Signal Processing, Vol.14, pp. 335–346.*

[39] **Perona, R., Malik, J.** (1990). Scale-space and Edge Detection using Anisotropic Diffusion, *IEEE Trans. Pattern Analysis, Machine Intelligence, Vol. 12, No. 7, pp. 629–639.*

[40] **Yongjian, Y., Acton, S.T.** (2002). Speckle Reducing Anisotropic Diffusion. *IEEE Trans. Image Process., Vol. 11, No. 11, pp. 1260–1270.*

[41] **Abd-Elmoniem, K. Z., Youssef, A. B., Kadah, Y. M.** (2002). Real-time Speckle Reduction and Coherence Enhancement in Ultrasound Imaging via Nonlinear Anisotropic Diffusion. *IEEE Transactions on Biomedical Engineering Vol. 49, pp. 997-1014.*

[42] **Gupta, S., Kaur, L., Chauhan, R. C., Sexana, S. C.** (2004). A Wavelet-based Statistical Approach for Speckle Reduction in Medical Ultrasound Images. *Medical and Biological Engineering and Computing, Vol. 42, pp.189–192.*

[43] **Russ., J.** (2011). *The Image Processing Handbook.* 6th Edition, Taylor & Francis Group.

[44] **Jain, A.** (1989). *Fundamentals of Digital Image Processing*. Prentice Hall Inc. Upper Saddle Rivers NJ, USA.

[45] **Cornelius, T**. (1998). *Image Processing and Pattern Recognition.* Neural Network Systems Techniques and Applications Academic Press.

[46] **Yin, L., Yang, R., Neuvo, Y.** (1996). Weighted Median Filters: A Tutorial. *IEEE Trans. On Circuits and Systems Analog and Digital Signal Processing Vol. 43 No.3, pp. 157-192.*

[47] **Roerdink, M., Meijster, A.** (2000). The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamental Informatice, Vol. 41, pp. 187-228.*

[48] **Digabel, H., Lantuejoul, C.** (1978). Iterative Algorithms. *In Proceedings of 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science, pp. 85–99.*

[49] **Beucher, S., Lantuejoul, C.** (1979). Use of Watersheds in Contour Detection. *International Workshop on Image Processing, CCETT/IRISA.*

[50] **Audigier, R., Lotufo, R. A.** (2007). Watershed by Image Foresting Transform, Tie-zone, and Theoretical Relationships with Other Watershed Definitions. *ISMM'2007 Proceedings, Vol. 1, Universidade de São Paulo (USP).*

[51] **Korbes, A., Lotufo, R.** (2009). Analysis of the Watershed Algorithms Based on the Breadth-first and Depth- first Exploring Methods. *IEEE Computer Society in SIBGRAPI, pp. 133-140*.

[52] **Vincent, L., Soille, P.** (1991). Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 6, pp. 583-598.*

[53] **Scientific Volume Imaging**. *http://www.svi.nl/watershed*, date retrieved: 10.04.2015

[54] **Jayaraman, S., Esakkirajan, S., Veerakumar, T.** (2009). *Digital Image Processing.* Tata McGraw Hill Education.

[55] **Kong, T. Y., Rosenfeld, A.** (1996). *Topological Algorithms for Digital Image Processing.* Elsevier Press.

[56] *http://engineering.purdue.edu/~bouman/ece637/notes/Segmentation/pdf*, date retrieved: 20.04.2015.

[57] **Karaman, M., Li, P. C., O'Donnell, M.** (1995). Synthetic Aperture Imaging for Small Scale Systems. *IEEE Trans. Ultrason., Ferroelect., Freq. Contr., Vol. 42, No. 3, pp. 429-442*.

CURRICULUM VITAE

Name Surname: Sibel Kadıoğlu

Place and Date of Birth: 23.09.1988

E-Mail: sibela.kadioglu@gmail.com

EDUCATION

B.Sc.: Electronics and Communication Engineering
Yıldız Technical University

Electrics Engineering, Double Major
Yıldız Technical University

PROFESSIONAL EXPERIENCE AND REWARDS:

Fenac Encoder, summer internship	2014
Teratek, summer internship	2012
NCR, summer internship	2011
Türk Telekom, summer internship	2010
Brown University, USA, research internship	2009
Erasmus exchange scholarship	
Aalto University, Finland	2010/11