

Speckle Detection in Ultrasonic Images
Using Unsupervised Clustering Techniques

A Thesis

Submitted to the Faculty

of Drexel University

by

Arezou Akbarian Azar

in partial fulfillment of the
requirements for the degree

of

Doctor of Philosophy

January 23, 2014

© Arezou Azar 2014

Arezou A. Azar. All rights reserved

Dedication

I dedicate this thesis to women around the world who pursue higher education in science and engineering regardless of the obstacles in their path.

Acknowledgements

There are too many individuals to thank. My main motivation in pursuing graduate studies was to foster my passion for biomedical sciences. I could not have gotten this far without the love, encouragement, support, and understanding of my lovely family who always offered unconditional love and support to me. I would like to thank my thesis scientific committee members. My special thanks to Dr. Teck-Kah Lim, Dr. N. John DiNardo, Dr. Emad Boctor and Dr. Andres Kriete for making the completion of this process possible.

My dissertation research was conducted as a trainee in Division of Medical Imaging Physics (DMIP) lab in Johns Hopkins University Hospital, Division of Medical Imaging Physics under supervision of Dr. Emad M. Boctor Mikhail. I am grateful to Dr. Benjamin M.W. Tsui, Dr. Emad M. Boctor Mikhail, and thankful to Dr. Elliot R.

McVeigh and Dr. Daniel Herzka for making this great opportunity possible for me. I have had the pleasure of working with Dr. Hassan H. Rivaz and Dr. Arman Rahmim and using their guidance and experience in my publications. I also would like to thanks Dr. Pezhman Foroughi for providing the initial data for my analysis. I made great friends and had great colleagues in Baltimore and Johns Hopkins University that I am grateful to know all of them.

I thank Dr. Kriete for all the guidance and supervision during completion of this research work. I would like to give my special thanks to my scientific committee members Dr. Ahmet Sacan, Dr. David E. Breen, and Dr. Wan Y. Shih. Last but not the least; I would like to thank Dr. Bahrad A. Sokhansanj, Dr. Dov Jaron, and Dr. Margaret A. Wheatley, and Dr. Rahamim (Rami) Seliktar for their guidance during my teaching opportunities.

Table of Contents

List of Tables	8
List of Figures	10
Abstract	17
Chapter 1 Speckle Concept and Characterization	21
1.1. Introduction and History.....	21
1.2. Speckle properties and pattern.....	24
1.3. Physical Properties	25
1.4. Significance of Thesis Work (Thesis Statement)	29
1.5. Dissertation Outline.....	35
Chapter 2 Ultrasound Basis.....	36
2.1 Ultrasound Imaging	36
2.2 Basic Principles and Physics	38
2.3 Ultrasound Modes.....	43
2.4 Interaction of Ultrasound Waves with Tissue	45
2.5 Reflection.....	46
Chapter 3 Envelop Characteristics	50

3.1	Statistical Modeling of Speckle	50
3.2	Statistics of Fully-Developed Speckle (FDS).....	51
3.3	Statistical Models	53
3.3.1	Rayleigh Distribution	55
3.3.2	K-Distribution.....	59
3.4	Features for Speckle Discrimination: SNR, Skewness, and Kurtosis	61
3.4.1	Calculating Statistical Features Based on v	65
Chapter 4	Classification Techniques.....	67
4.1	Background.....	67
4.2	Clustering Techniques	67
4.2.1	K-means and K-medoid.....	68
4.2.2	Fuzzy C-means clustering	69
4.2.3	Gustafson-Kessel clustering algorithm.....	70
4.2.4	Gath-Geva fuzzy Classifier	72
Chapter 5	Data Generation.....	74
5.1	Introduction	74
5.2	Simulations	75
5.2.1	Field II simulation	75
5.2.2	UTool Software: A Novel Ultrasound Simulation GUI for Phantom Generation	78
Chapter 6	Methodology (Speckle Detection).....	81
6.1	Introduction	81
6.2	Feature Selection: Search for Best Statistical Features for Speckle Classification	83
6.3	Classification of extracted statistical features	84
6.3.1	“My Tool” GUI	84
6.4	Method 1: Semi-Automatic and Ensemble Speckle Detection	88
6.4.1	The Overall Approach	89
6.5	Method 2: Automated method (Fully Automatic Speckle Detection (FASD) Algorithm	90
6.5.1	Overall Approach	90
6.5.2	Optimization Search (Exhaustive Search).....	93

Chapter 7 Results and Performance	97
7.1 Phantom results	97
7.1.1 Calculation of B-mode image of Cyst Phantom	101
7.1.2. Calculation of B-mode image of synthetic Kidney	150
7.1.4. Calculation of B-mode image of synthetic Heart	156
7.2 Real data results	165
7.3 Comparative Analysis of Method 1 and Method 2	168
Chapter 8 Conclusion and Outlook	177
8.1 Summary and Conclusion	177
8.2 Future Directions and Applications.....	182
8.2.1 Applications.....	182
8.2.2 Future Directions	183
Appendices.....	185
Appendix A	186
Test Phantom Stat.....	186
Appendix B	189
UTool Source Code	189
Appendix C.....	204
List of Abbreviations	209
References	210
Vita.....	217

List of Tables

Table 2.1: Properties of different media [55].....	42
Table 7.1: Classification results with different number of scatterers.	137
Table 7.2: Patch size effect on K-means classification.....	138
Table 7.3: Patch size effect on K-mediod classification.....	140
Table 7.4: Patch size effect on Fuzzy C-Means classification.....	141
Table 7.5: Patch size effect on GK classification.	143
Table 7.6: Patch size effect on GG classification.	144
Table 7.7: Average performance of the classification methods for speckle detection of simulated left ventricle ultrasound images in three different tissue contrasts and speckle detection results for the Gustafson-Kessel fuzzy classifier.	163
Table 7.8: Average performance of the classification methods for speckle detection of images in Figure 7.76. Total number of 100x100 pixel patches for each image was 96.164	
Table 7.9: List of Voted Order Combinations for this specific test.....	173

Table 8.1: Average speckle detection performance for Method 1.....	181
--	-----

List of Figures

Figure 1.1: Typical Speckle texture.	24
Figure 1.2: RF signals forming envelope echo [32]. (a) and (b) The reflected echo in the transducer is the result of a summation of backscattered waves with different phases. (c) The pulse envelope is approximately Gaussian and it's Gaussian shape is maintained ...	26
Figure 1.3: Diffuse Scattering. Dotted arrow presents the sum.	28
Figure 1.4: Ultrasound beam and resolution cell [29].	28
Figure 1.5: Representation of image patches.	32
Figure 1.6: Static classification of image patches to FDS and non-FDS. Comparative analysis of five different unsupervised clustering techniques with different combinations of statistical features is used as input to the classifiers.	33
Figure 2.1: A typical RF signal obtained from in-vivo human liver. The B-mode image is obtained from the envelope signal [33].	38
Figure 2.2: Medical ultrasound transducer, (a) Linear array, (b) Phased array, and (c) Curved linear array.	39

Figure 2.3: From left to right, (a) Sample of B-Mode image, (b) M-Mode combined with B-Mode image, (c) Sample of Doppler image [http://www.sonoguide.com/physics.html].	44
Figure 2.4: Random reflection on a rough surface.	47
Figure 2.5: Scattering in the tissue volume.	47
Figure 2.6: Scattering of ultrasound wave in interaction with small objects.	49
Figure 2.7: Reflection of an ultrasound wave by a large reflector (scattering between the unhomogeneous borders of two different mediums). In (a) the reflector appears bright and in (b) the reflector appears dark.	48
Figure 3.1: Random walk schematic view (Image courtesy of Vinayak Dutt).	52
Figure 3.2: Statistical models and scattering types [2].	54
Figure 3.3: Probability density function for Rayleigh distribution for different σ values.	56
Figure 3.4: Probability density function K-distribution.	61
Figure 3.5: Pipeline for feature extraction for each patch and Speckle detection (classification) [29].	64
Figure 5.1: Field II GUI implemented in Duke University [http://dukemil.egr.duke.edu/Ultrasound/Simulation/field_ii/field_ii_gui.html].	77
Figure 5.2: Phantom generation process.	79
Figure 5.3: A screen shot of the UTool GUI with the example of a generated cyst phantom. Bench phantoms for cysts and fetal exist, which other worldwide research groups have generated and used, which UTool can be compared to.	80
Figure 6.1: A snapshot of My Toolbox for speckle detection with Cyst phantom example. [red patches = patches with no cyst included]. Patch size is 70x70.	85
Figure 6.2: A snapshot of My Toolbox loaded with the cyst phantom and patch sizes 70x70.	87
Figure 6.3: Static Classification of image patches to FDS and non-FDS. Comparative analysis of five different unsupervised clustering techniques with different combinations of statistical features used as input to the classifiers.	89
Figure 6.4: Adaptive Speckle Classification block diagram.	91

Figure 6.5: Ensemble classification scheme to combine the speckle classification results using majority voting.	92
Figure 6.6: Sensitivity analysis for non-adaptive method (Method 1): A) input image patches; B) Sensitivity of Fuzzy C-Means classifier to orders of statistical features. vK varies from 0 to 4, while other orders, VR and VS set to 1.	94
Figure 6.7: The diagrammatic steps for an optimization search for fully automatic speckle classification for real data with no available ground truth.	95
Figure 7.1: Snapshot of the cyst phantom generation process in Matlab using the UTool GUI Software.	99
Figure 7.2: Comparing overlap and similarity of two regions using the Dice similarity (DS) index. A = image patch labels for ground truth, B = image patch labels obtained by a classifier. DS has always a value between 0 and 1. If A and B are completely different, that means there is no overlap, the Dice similarity is zero, and true positive is zero. If A and B are identical, that means there is full overlap and the Dice similarity is 1	101
Figure 7.3: The cyst phantom generated using GUI as the simulation software: the phantom contains five point targets, 6, 5, 4, 3, and 2 mm diameter water filled cysts, and 6, 5, 4, 3, and 2 mm diameter high scattering regions.	103
Figure 7.4: Cyst phantom simulated by the Field II program, photo courtesy of Duke University.	104
Figure 7.5: Developed UTool Software to generate own phantoms. Snapshot of the ultrasound simulation GUI with steps and examples for the values of the parameters for generation of phantom data using UTool: phantom contains five point targets, 6, 5, 4, 3, and 2 mm diameter water-filled cysts, and 6, 5, 4, 3, and 2 mm diameter high scattering regions.	106
Figure 7.6: Phantom containing 4 square objects.	107
Figure 7.7: Square cyst phantom generation using the UTool GUI.	108
Figure 7.8: Phantom with 6 mm square cysts/objects.	109
Figure 7.9: Field II cyst phantom without the 2 mm cyst.	109
Figure 7.10: Field II cyst phantom generated with two 2 mm and four 3 mm cysts.	110
Figure 7.11: Figure A (left) and Figure B (right), two other Field II cyst phantom generated with different set of sizes and different number of cysts for investigations...	110

Figure 7.12: Cyst phantom with color scale visualization.....	112
Figure 7.13: Probability density function for cyst regions.	113
Figure 7.14: Skewness probability density function.....	114
Figure 7.15: Classification results for the proposed five classifiers: Fuzzy C-Means, K-means, K-mediod, Fuzzy GK and Fuzzy GG, using the generated ground-truth.	116
Figure 7.16: Entropy of the cyst phantom (as the only feature used for classification).	117
Figure 7.17: Heat map of Homogeneity as a feature.	118
Figure 7.18: Heat map of Energy as a feature.	119
Figure 7.19: Color intensity of patch average intensity feature.....	119
Figure 7.20: Color intensity of Skewness as a feature.....	120
Figure 7.21: Color intensity (Heat map) of SNR (order 10) as a feature. False positives exist.	121
Figure 7.22: Color intensity of Skewness (order 10) as a feature.....	121
Figure 7.23: Binary image of the threshold heat map of the Skewness (order 10) as a feature. False positives exist.	122
Figure 7.24: Feature analysis for speckle detection block diagram.....	125
Figure 7.25: B-mode image with a small cyst in a patch at upper right corner of the image.....	126
Figure 7.26: Color intensity of R in the patches.	127
Figure 7.27: Color intensity of S in the patches.....	127
Figure 7.28: Color intensity of K in the patches.	128
Figure 7.29: 100% accuracy in classification.	128
Figure 7.30: Classification results for the $v_R=2.0$, $v_S=1.5$, and $v_K=1.8$ set.	130
Figure 7.31: Statistical analysis of the R (mean to standard deviation ratio feature) values in the image patches. For $v_R=2.0$, both bright and dark cysts were found and general speckle regions are represented by orange/light red colors.	131

Figure 7.32: Statistical analysis of the S (skewness) feature values for each image patches for $v_S=1.5$. In general, the speckle regions are blueish. The darker cysts are presented with darker blue and the brighter cysts are presented with more orange color intensities. The point here is that the image confirms that in non-speckle regions the values of the intensities of the S are far different than in the speckle regions.	131
Figure 7.33: Statistical analysis of the K (kurtosis) feature values in the image patches for $v_K=1.8$	132
Figure 7.34: Classification results for the $v_R=2.0$, $v_S=1.5$, and $v_K=1.8$ set.	132
Figure 7.35: Classification results for the $v_R=1.0$, $v_S=1.8$, and $v_K=0.8$ set.	133
Figure 7.36: Classification results for the $v_R=1.0$, $v_S=0.8$, and $v_K=3.0$ set.	133
Figure 7.37: Classification results for the $v_R=1.0$, $v_S=0.5$, and $v_K=2.5$ set.	134
Figure 7.38: Phantom results with 10000, 50000, 100000, 250000, and 1000000 scatterers (left to right).	135
Figure 7.39: Snapshot of cyst phantom generation line by line using the UTool Software.	136
Figure 7.40: Cyst phantom with dark and bright cysts in color scale.	146
Figure 7.41: Skewness and SNR for cyst phantom with patch size 10x10. Cysts are not obvious in skewness and SNR.	146
Figure 7.42: Skewness and SNR for cyst phantom with patch size 30x30. Cysts are partially obvious in skewness and SNR.	147
Figure 7.43: Skewness and SNR for cyst phantom with patch size 50x50. Cysts are obvious in skewness and SNR.	147
Figure 7.44: Skewness and SNR for cyst phantom with patch size 100x100. Cysts are obvious in skewness and SNR.	148
Figure 7.45: (A) Simulated ultrasound image of a Cyst phantom and the Ground Truth created by expert manual labeling (White (1) color is representing Cyst patches and black (0) is representing fully developed speckle (FDS) patches). (B) Speckle detection results for five different unsupervised classifiers. Total number of 100x100 patches for the phantom image was 24. Patches classified as fully developed speckles (FDS) are shown as black. All methods except GK-fuzzy classifier performed the same. Orders for statistical features respectively were 1,1 and 0.5 [29].	150

Figure 7.46: Artificial kidney scan [105].....	152
Figure 7.47: Artificial fetus in 12th week [101].	153
Figure 7.48: Simulated fetus phantom and patches. Images were segmented into 12x8 image patches, where each image patch had a size of 100x100 pixels [29, 30].....	154
Figure 7.49: Sensitivity analysis for Method 1: (A) Input image patches. (B) Sensitivity of Fuzzy C-Means classifier to orders of statistical features. v_K varies from 0 to 10, while other orders, v_R , v_S are set to 1, (1, 1, at v_K). It can be seen that best results (max Dice similarity) is achieved at $v_K = 1.5$ and best v_K values are in range $0.5 < v_K < 1.5$	155
Figure 7.50: Simulated ultrasound image of a fetus in 12 th week (A) and speckle detection results for 5 classifiers (B.) Patches classified as FDS are shown as black. In this case, FCM, K-means and K-mediod performed the same. GK- and GG-fuzzy classifiers were able to decrease false positives and improve accuracy of the speckle detection. Total number of patches (100x100 pixels) for the phantom image was 96. Orders for features were 10,1 and 0.01 [29, 30].	156
Figure 7.51: Simulation Examples: (A) short-axis echocardiographic image and (B) left ventricle echocardiographic image.	158
Figure 7.52: Simulation examples after image segmentation: (A) short-axis (end diastolic). (B) left ventricle phantoms and patches. Images were segmented into 12x8 image patches, where each image patch had size of 100x100 pixels.	159
Figure 7.53: (A) Simulated ultrasound image of the heart in short axis (end diastolic). (B) Speckle detection results for five different unsupervised classifiers. Patches classified as fully developed speckles (FDS) are shown as black. Total number of patches (100x100 pixels) for the phantom image was 96. Orders for the statistical features respectively were 1,1 and 1.....	160
Figure 7.54: (A) Simulated ultrasound image of left ventricle (B) and speckle detection results for five different unsupervised classifiers. Patches classified as fully developed speckles (FDS) are shown as black. Total number of patches (100x100 pixels) for the phantom image was 96. Orders for statistical features respectively was 1,1 and 1.....	161
Figure 7.55: Simulated left ventricle ultrasound images in three different tissue contrasts and Speckle detection results for the Gustafson-Kessel fuzzy classifier. Patches classified as fully developed speckles (FDS) are shown as black.	162

Figure 7.56: Simulation examples and speckle detection results for the Gustafson-Kessel fuzzy classifier: A) short-axis and B) left ventricle echocardiographic images. Patches classified as fully developed speckles (FDS) are shown as black.	164
Figure 7.57: (Results for Method 1) : (A) Ultrasound image of the right ventricle (B) and speckle detection results for five different unsupervised classifiers. Patches classified as fully developed speckles (FDS) are shown as black (Patches 30x30 pixels). Orders for statistical features respectively was 1,1 and 1.	165
Figure 7.58: Ultrasound image of normal right kidney (Sagital) and segmented image, including the radiologist observations (http://www.sonoguide.com/renal.html).....	166
Figure 7.59: Results for recursive speckle tracking, fully automatic ultrasound image of the kidney (Sagital).	167
Figure 7.60: Results for heart ultrasound image with speckle detection results presented for Method 1 using Fuzzy C-means method and $vR=1$, $vS=0.5$, $vk=0.25$. Blue is FDS and red patches are non-FDS.....	169
Figure 7.61: Results for heart ultrasound image with speckle detection results presented for Method 2 using Fuzzy C-means method. Light blue= FDS; Red=Non-FDS; Orange=Background.	170
Figure 7.62: Heatmap image of the R, S, K values in Method 1 for the heart image.....	172
Figure 7.63: Heatmap image of the R, S, K values in Method 2.	176
Figure 8.1: Average speckle detection performance.....	181
Figure B.1: Running 'Utool' in Matlab.	170
Figure C.1: A sample Kidney bitmap [105]	205

Abstract

Speckle Detection in Ultrasonic Images
Using Unsupervised Clustering Techniques
Arezou A. Azar

Research for the improvement of the quality of clinical ultrasound images has been a topic of interest for researchers and physicians. One of the challenges is the presence of speckle artifacts. This dissertation reviews the speckle phenomena in such images, and develops algorithms to better identify this artifact in sonographic images. Speckle artifact is categorized into two groups: partially developed speckles and fully developed speckles (FDS). This concept has been used, along with the classification techniques, to segment the ultrasound images into patches and classify the patches in the image as FDS or non-FDS. The proposed algorithms and the results of the

experiments have been validated using simulation, phantom and real data that were created for the purposes of this study or taken from other research groups.

Current speckle detection methods do not optimize statistical features and they are not based on machine learning techniques. For the first time this work introduces a novel method for searching and extracting the best features for optimizing speckle detection rate using statistical machine learning and ensemble classification. Potential applications include strain imaging by tracking speckle displacement, elastography, speckle tracking and suppression applications, and needle-tracking applications.

Chapter 1

Speckle Concept and Characterization

1.1. Introduction and History

Artifacts in ultrasound images are very common. These artifacts can manifest themselves as structures that do not actually exist, and improper brightness, shape or size. Artifacts have different causes and various origins. They might be the result of improper equipment operation, improper settings, or related to the nature of the ultrasound modality. The most common artifacts in ultrasound systems include shadow and enhancement artifacts, Doppler artifacts, edge artifacts, mirror artifacts, reverberation artifacts, and speckle artifacts, and are fully described in other work

[34]. The focus of this work is on detecting speckle. Origin of speckle will be fully described in Chapter 2.

Speckle is a common phenomenon in coherent imaging methods. Speckle is considered a type of artifact in sonographic images and has a negative impact on these images; however speckle may provide useful information too depending on the type of under-scan biological tissue as well as the application that studies speckle concept. Speckle has been studied since the early 1970s [1-3] and researchers investigated and described the statistical properties and fundamentals of the speckle. Key areas in speckle research include: tissue characterization, Speckle quantification, Speckle suppression and Speckle detection. Improved speckle detection can help different applications, including segmentation, sensorless 3D sonographic imaging, speckle cancellation and quantitation in biological tissues. In sonographic compounding, for instance, the main objective is to cancel speckle artifact, whereas in sensorless 3D, freehand ultrasound compounding is employed to estimate probe movement. Thus, speckle characterization has numerous applications and the results of this thesis may be utilized and applied in different research areas. However, among all such applications, speckle detection is the main and only interest in this thesis work.

Speckle noise, like any other artifact, lowers the quality of ultrasound B-mode images and is considered the most dominant artifact and the primary factor in limiting the contrast resolution of diagnostic ultrasound imaging [2, 3, 35]. Speckle is difficult to suppress [4-22], and its presence in the image may obscure small structures and

degrade the quality of image, thus reducing the spatial resolution of image [23]. This affects the human interpretation of these images by physicians and non-experts, as well as the accuracy of computer-aided methods. There are techniques used to improve the quality of the sonographic images [25]. The presence of speckle within an image, and the resulting degraded, poor image quality, makes the analysis, feature extraction, pattern recognition and quantitative measurements of such images and thus application of automated computer-aided analysis techniques and algorithms such as edge detection, volume rendering, and 3D display, very problematic and unreliable. Speckle significantly limits the detectability of small, low-contrast lesions [1, 2, 4, 5] by an approximate factor of eight [24].

Therefore, tracking the speckles in the image and employing techniques to despeckle the image is very important, and has compelled additional investigation [10, 12, 19-21, 26-30]. Previous research studied the nature of this noise and applied different filtering techniques in an attempt to suppress speckle noise so that the important features of the image are preserved [4, 7, 31]. While such approaches are useful for reducing speckle artifact, they are limited by the need for multiple images acquired on the same structure. Most recent techniques include texture analysis, expert visual assessment, and image quality evaluation metrics [6], as well as using despeckle filtering techniques such as linear and nonlinear filtering methods, anisotropic diffusion filtering, and wavelet filtering [25].

This thesis work investigates various statistical analysis and pattern recognition techniques, such as using different unsupervised clustering techniques, employed for tracking speckles in ultrasonic images.

1.2. Speckle properties and pattern

Speckle is described as a textural pattern that can vary depending on the properties of the specific biological tissue [25]. Speckle appears as the typical light and dark spots in the image that results from the destructive interference of sonographic waves scattered from different sites of the tissue. Speckle degrades the quality of medical sonographic images and makes the visual assessments very difficult for this modality [1, 2]. Figure 1.1 demonstrates a typical speckle texture. As mentioned, the granular appearance of speckle is the result of the interference of echoes from the distribution of scatterers in tissue.

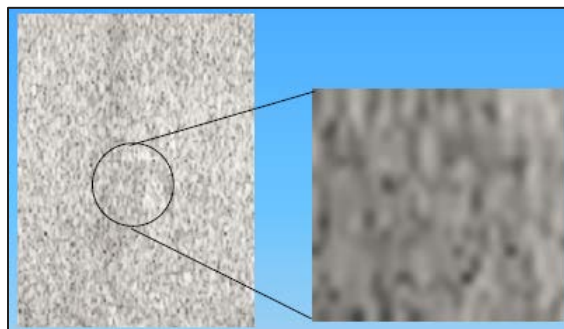


Figure 1.1: Typical Speckle texture.

Speckle is characterized as a random pattern with a deterministic nature that is formed by coherent radiation of a medium that contains many scatterers. In other words, when a fixed rigid object is scanned twice under exactly the same conditions, one obtains identical speckle patterns. Although speckle is of random appearance, speckle is not random in the same sense as electrical noise. However, if the same object is scanned under slightly different conditions, such as a different transducer aperture, pulse length, or transducer angulation, the speckle patterns change [25]. In a medical sonogram, the speckle texture does not relate to the under-scanning structure, but the speckle's local brightness reflects the local echogenicity of the under-scanning scatterer models [2, 3, 28, 32].

1.3. Physical Properties

B-mode ultrasound images display the scattered ultrasound beams or echoes that are returned to the transducer from different organs of interest. The more detailed physics behind this is explained in Chapter 2. These scatterers are due to the inhomogeneities and small structures within a given tissue, where alterations in acoustic impedance occur over a microscopic level within the tissue. Tissue particles that are relatively small in relation to the wavelength (i.e. blood cells, fatty droplets) and particles with differing impedance that lie very close to one another, cause scattering or speckling. Absorption of the ultrasound tissue is an additional factor in scattering and refraction. Figure 1.2 presents the scattering process, consisting of a

homogenous medium through which sonographic waves originate from a transducer, and it also displays four point-like scatterers.

The back-scattered waves make contact with a transducer at slightly different intervals after the transmission of a sonographic pulse. This pulse envelope is approximately Gaussian, as shown in Figure 1.2 (c). This Gaussian shape of the spectrum is maintained while the pulse travels through the medium. When the transducer receives a reflected echo, the radiofrequency is manifested from the transducer as the result of a sum of instantaneous sound pressures originating from backscattered waves. This is shown with four waves in Figure 1.2 (a).

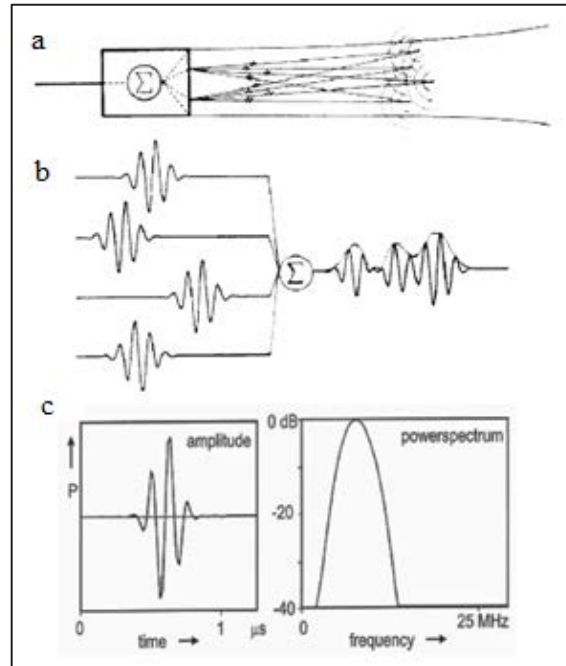


Figure 1.2: RF signals forming envelope echo [32]. (a) and (b) The reflected echo in the transducer is the result of a summation of backscattered waves with different phases. (c) The pulse envelope is approximately Gaussian and it's Gaussian shape is maintained.

The received scatterers in the transducer have depth differences, but these differences are lower in magnitude than the axial size of the resolved volume of the transducer, as reflected by the pulse length. This essentially causes the so-called speckle pattern. Speckle, however, is not a true imaging feature of tissue histology but rather an interference comprising a complex source of noise in a medical sonogram. Speckle pattern is mainly determined by the beam characteristics. Its statistical distribution is non-Gaussian as well as spatially and signal dependent.

The two classifications of scatterings are diffuse and coherent. The diffuse type arises from a large magnitude of scatterers with a random phase within the resolution cells of the US beam, which results in the formation of speckle artifact in the reconstructed ultrasound images. Coherent scattering arises when the scatterers in the resolution cell are in phase, and thus results in dark and bright spots in the sonographic images [29]. As shown in Figure 1.4, each pixel in an ultrasound image is formed by the back-scattered echoes from an approximate ellipsoid called the resolution cell. The Rayleigh distribution model is used as the statistical model used very commonly for the envelope signal. This model assumes that a large number of scatterers exists per resolution cell [29]. Figure 1.3 illustrates the concept of diffuse scattering.

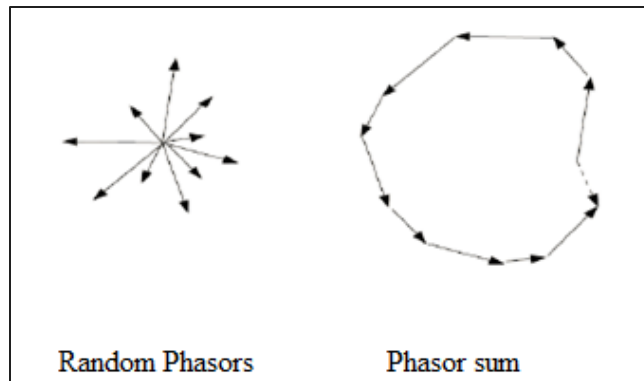


Figure 1.3: Diffuse Scattering. Dotted arrow presents the sum.

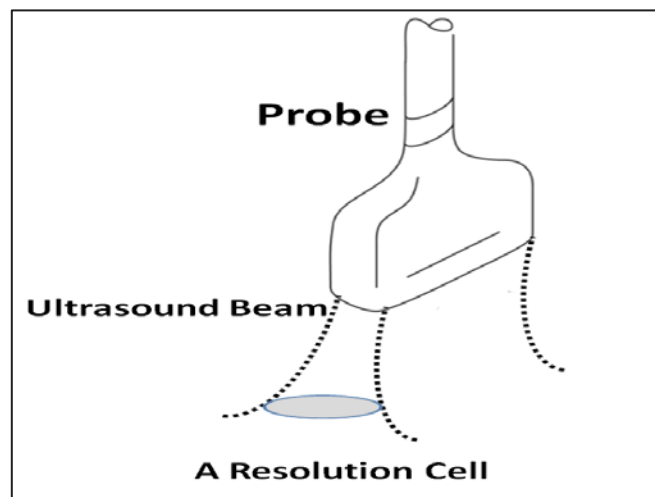


Figure 1.4: Ultrasound beam and resolution cell [29].

If each resolution cell in an image patch has many scatterers, the corresponding patch is called fully developed speckle (FDS). When a large number of scatterers with uniform phase statistics are added, the result can be referred to as

partially developed speckle. In this research we consider both partially developed speckle and non-FDS scenarios as non-FDS.

Raleigh scattering is the one mechanism that enhances echo signal with increased ultrasound frequency. It also refers to an acoustic scattering phenomenon that describes reflection from small objects that have dimensions much smaller than the wavelength of the ultrasound source. This phenomenon is also pertinent in the scanning of solid organs like kidneys and liver.

There is a distinctive difference between the speckle that appears in the image and the speckle received in the RF signal. In tracking from the transducer of the medical imaging system to the display screen, several transformations affect the signal's statistics, most importantly the log compression of the signal, that is used to lower the dynamic range of the input signal to align with the lower dynamic range of the display screen of the imaging system. These settings are usually adjusted manually such as when the overall gain of the machine controls the amplification.

1.4. Significance of Thesis Work (Thesis Statement)

There are several obstacles to study speckle and very limited literature in analyzing the statistical behavior of this phenomenon. Thus one of the key goals of this dissertation was to do a mechanism study and understand the characteristics of both fully developed speckle and partially developed speckle. Another key goal was to try to overcome the limitations of speckle classification and detection. Supervised

clustering techniques offer several obstacles and won't lead us to an automatic detection algorithm. These techniques require sufficient training material and considering the overlap of statistical features of tissues, necessitates a large required number of image patches. Because such large number of image patches is lacking, this work focused on unsupervised clustering techniques.

Supervised methods require manual labeling by an expert. Also these techniques are very application dependent and the results will vary from one application to other application depending on the various tissue types being investigated in each application. Using unsupervised techniques will overcome these obstacles. Thus for the purposes of this thesis unsupervised classification techniques are proposed for speckle classification.

It should also be noted that this speckle research work is focused on speckle detection methods that could be used in possible future works for speckle reduction and suppression purposes.

There is currently no robust algorithm that can automatically detect speckle regions in ultrasonic images, which was the main motivation for current research work. This thesis proposes to develop and test such an algorithm. The results of this speckle detection work can be applied in two key applications of speckle tracking and reduction in medical imaging. Possible benefits of such applications are better quality in sonographic images and potential improve in the accuracy of interventional radiology procedures, such as thermal ablation of tumors, by allowing for improved needle localization relative to the therapeutic target in areas of the image with a large

magnitude of speckle artifact. This in turn would further the effectiveness of ultrasound-guided procedures as opposed to CT-guided procedures that involve radiation exposure. The proposed algorithm can also be used in investigations to quantify the amount of coherent scattering, compensating for the inaccuracies caused by elevational distances between ultrasonic frames in real tissues.

The overall approach this thesis presents overcomes the limitations of previous methods, which are mainly based on supervised techniques and require an expert to manually label a portion of the training data. This thesis presents the state of the art of speckle detection in ultrasonic images. The thesis proposes a novel and radically different methodology that is based on unsupervised clustering techniques and a combination of different statistical models for the fully automatic speckle detection. The proposed methodology can be easily implemented in clinical practice or integrated with any ultrasonic devices.

The overall approach utilized in this work is demonstrated in Figure 1.6. Each ultrasound image was segmented (partitioned) to regions of interest (ROIs), called patches. This partitioning concept is shown in figure 1.5. These patches were then used to study the regional behavior of image pixels. Each given patch includes a variety of backscattering effects: FDS, few scatterers and coherent scatterers [62, 63]. For each image patch A , statistical features are calculated. These features however can overlap for some tissues, so the pattern classification approaches should be utilized to classify tissues based on the extracted statistical features. Over several decades, multiple supervised and unsupervised classification and segmentation

algorithms have been proposed to process medical images. Some of these techniques are listed in [37-39]. Also, the Maximum Likelihood (ML) was computed in estimation of the data A following the Rayleigh distribution [29].

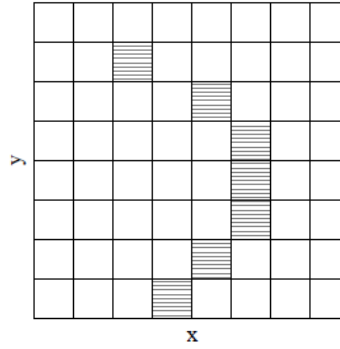


Figure 1.5: Representation of image patches.

This research also investigated the best statistical features (SNR, Skewness, Kurtosis) to be used in speckle detection studies as well as the optimum number of features to be used that may result in the optimum classification results to classify these patches to two classes: fully developed speckle (FDS) and non-FDS patches. Known unsupervised classification methods (K-means, K-medoid, Fuzzy C-means, Gustafson-Kessel and Gath-Geva) were used for classification [37-39].

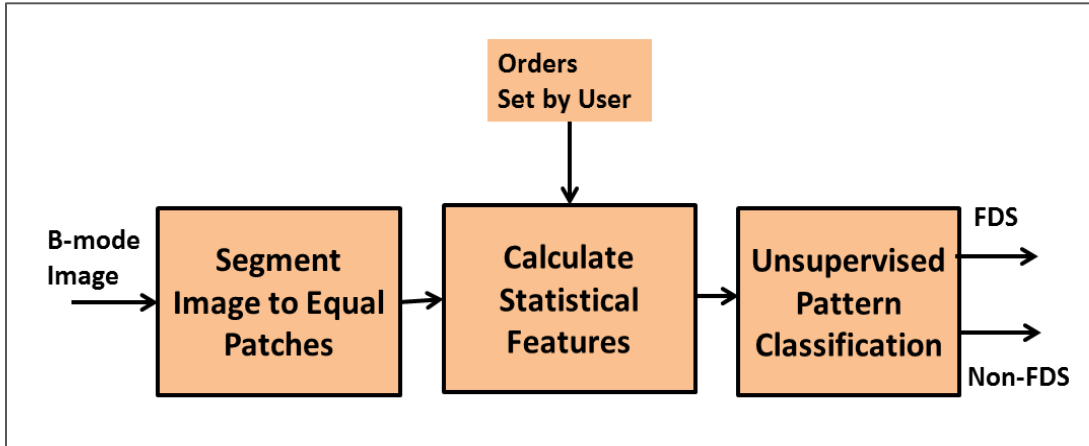


Figure 1.6: Static classification of image patches to FDS and non-FDS. Comparative analysis of five different unsupervised clustering techniques with different combinations of statistical features is used as input to the classifiers.

The proposed ensemble speckle classification method shown in Figure 1.6 consists of three major steps:

- Step 1: Segmentation of the input image into image patches.
- Step 2: Extraction of statistical features for each patch.
- Step 3: Use of one of several unsupervised clustering techniques explained in chapter 3, section 3.2, to classify each image patch as FDS or non-FDS.

Two robust classification algorithms are developed during the study:

- Method 1: A robust unsupervised speckle detection for simulation with available ground truth.

- Method 2: An optimization search and ensemble classification scheme to fully automatic detection of speckle regions for real data with no available ground truth.

Results of this work can also be used in the continuing research to improve ultrasound image quality in the diagnosis and assessment of organ, human blood, and soft tissue structures. Different clinical applications of the ultrasound imaging systems can benefit from the results of this work including: cardiac imaging and vascular occlusion processes, human fetus studies, pediatric malignancy studies, therapeutic and surgical applications, and malignant tumor ablation (e.g. liver cancer, lung cancer). In thermal therapies of localized tumors, sonographic imaging can be used to monitor the extent of ablated tumors and to assess the integrity of important surrounding structures [40].

Improved speckle detection can also help other applications, including segmentation, sensorless 3D manual sonographic imaging, speckle cancellation and quantitation in biological tissues. In sonographic compounding, for instance, the main objective is to cancel speckle artifact, whereas in sensorless 3D, freehand ultrasound compounding is employed to estimate probe movement. Thus, speckle characterization has numerous applications.

1.5. Dissertation Outline

The eight chapters in this thesis discuss speckle detection in ultrasonic images using unsupervised clustering techniques for statistical analysis. Chapter 1 is an introduction to speckle history and characteristics as well as a summary of the state of the art in this thesis work. Chapter 2 is a brief introduction to ultrasound imaging and the background physics and principles of speckle, as addressed within this thesis work. Chapter 3 discusses the statistics of fully developed speckle and lays out the various models and distributions considered for this research. Chapter 4 describes the specific unsupervised clustering techniques and the statistical analysis utilized. Chapter 5 introduces the Field II simulation program, which was used to generate data for this research work, and toolbox that was used for ground truth generation. Chapter 6 discusses the proposed methodology and the optimum statistical features for speckle classification, presenting both a semi-automatic and automated method for detecting speckle. The thesis is concluded by Chapter 7, which presents the results of the phantoms generated by the Field II simulation, and Chapter 8 is a summary that draws conclusions about methods that we developed and how this current research can best be best utilized in different research and applications in the future.

Chapter 2

Ultrasound Basis

2.1 Ultrasound Imaging

Medical imaging technology has been changing dramatically throughout the last decades. The potential for using ultrasound as an imaging modality has been investigated and studied since the late 1940s. Sonographic imaging allows for the rapid, low-cost assessment of multiple clinical conditions including acute cholecystitis, hepatic malignancy, and deep vein thrombosis. Ultrasound (US) also allows for image-guided procedures including biopsies and abscess drainages [8]. Consequently the clinical potential for sonographic imaging has driven research into optimizing the resolution of these images. Another advantage of ultrasound,

compared to computed tomography (CT) and positron emission tomography (PET), is that it does not use ionizing radiation. This noninvasive, real-time and low-cost modality has also been used in clinical studies for cardiac imaging with echocardiography, human fetal imaging, childhood cancer, breast cancer, as well as therapeutic and surgical applications and malignant tumor ablation (e.g. liver cancer, breast cancer), etc.

This chapter will provide a brief overview of ultrasound (US) physics. Diagnostic medical ultrasound imaging is a non-ionizing, real-time, noninvasive and easy to use imaging modality that is widely used in research, diagnostic, clinical and surgical procedures [26, 41-47]. Ultrasound is typically modeled to propagate through tissue at 1540 m/s and typically requires a frequency of 300 MHz to obtain a 5 μm wavelength [48]. Most clinical ultrasound systems utilize frequencies of 1 to 15 MHz. Different frequencies allow for varying depths of penetration in clinical sonography [34]. For example, a focused linear array transducer at 10 MHz found in most sonographic machines has a 5 cm penetration depth [49].

Ultrasound produces a mechanical disturbance that moves through media as a pressure wave. When the medium is human tissue, the wavelike disturbance is the basis for sonographic techniques in diagnostic imaging. Thus, understanding the basic characteristics and behavior of ultrasound waves is essential to understand the utility of ultrasound in diagnostic medicine and thus understanding the concept of speckle and its limitations and impacts on ultrasound imaging.

2.2 Basic Principles and Physics

Ultrasound is an acoustic wave that propagates through an environment with frequencies above 20 KHz. As human auditory capacity ranges from 20 Hz to 20 KHz, humans cannot hear ultrasound waves. Medical ultrasound waves are generated and received by ultrasound transducers that are usually placed on surface of skin, and transport energy. An ultrasound transducer converts electricity into mechanical stress that generates sound waves and transports them to the body media and tissues inside, while the receiver site converts the mechanical vibrations to electrical signals. The received, raw, unprocessed electrical signal in the transducer is usually called the radio-frequency (RF) signal. Figure 2.1 shows a typical RF signal obtained from an ultrasound machine. Later in this chapter, B-mode images that are obtained from the envelope signal shown in this figure will be overviewed.

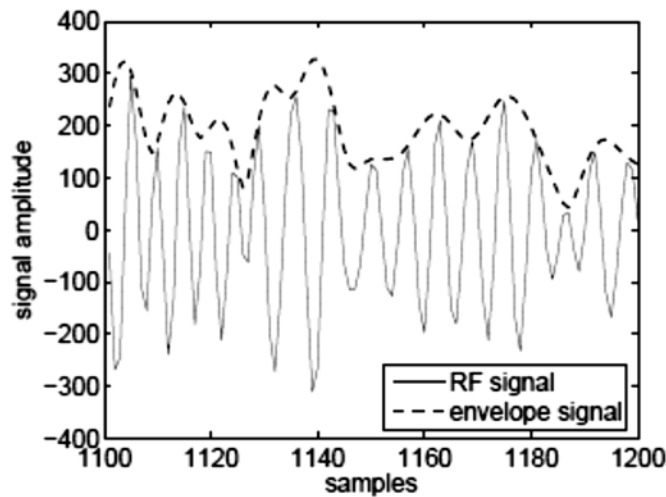


Figure 2.1: A typical RF signal obtained from in-vivo human liver. The B-mode image is obtained from the envelope signal [33].

The ultrasound transducer is made up of arrays of piezoelectric material, known as elements, which are comprised of crystals and ceramic. Sound waves are produced through oscillation (mechanical stress) of these piezoelectric crystals after excitation by electrical pulses and vice versa [50]. Figure 2.2 shows the different types of medical ultrasound transducers used in imaging. Different probe shapes and sizes are used in different applications [51]. The shape of the transducer controls the sound wave formation. Linear array transducers usually have less penetration, excellent resolution and a rectangular shape. Images from a convex array transducer will be wider with increased depth [52].

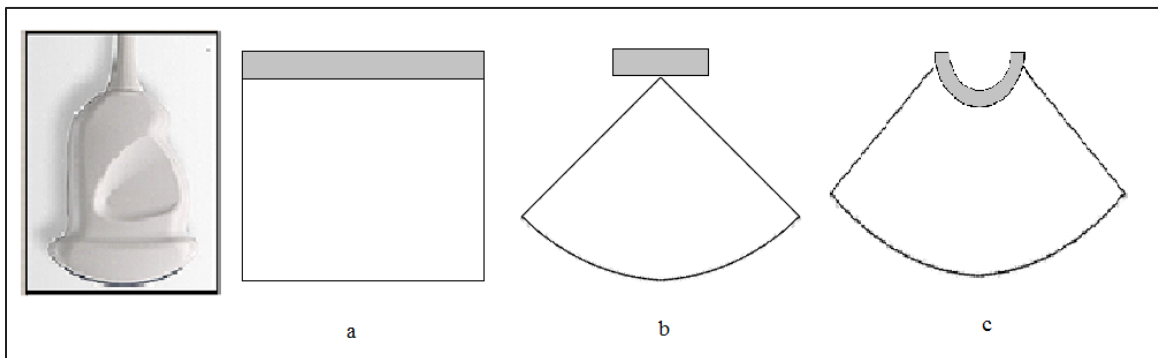


Figure 2.2: Medical ultrasound transducer, (a) Linear array, (b) Phased array, and (c) Curved linear array.

Ultrasound waves propagate and interact with various tissue types. Different tissues have different acoustic properties. Some of this propagated energy is returned from the underlying organs and is detected by the transducer. Medical ultrasound

imaging is primarily based on measuring these back-transmitted echoes, which are digitized and used to produce tissue images, a method known as pulse echo-mode.

Ultrasound waves are usually described through factors including density, direction of wave propagation and displacement of particles.

Wavelength is a key characteristic of a sonographic wave (λ). Wavelength quantifies the distance between two adjunct maximum or minimum values of the wave sine function, with frequency (f) defined as the number of waves per unit of time (t). Wavelength and frequency can subsequently be used for the calculation of velocity of the ultrasound wave (v) as:

$$v = f\lambda$$

(1.1)

And we know:

$$f = 1/T$$

(1.2)

where T is the period that quantifies the time between maximum and minimum values in a given sine wave function. This calculated velocity of the propagation of ultrasound waves can help in measuring the distance of the structures and organs from the ultrasound transducer in the tissue under scan.

As mentioned earlier, medical ultrasound imaging frequencies typically range from 1 MHz to 15 MHz. Larger frequencies are characterized by lower magnitude

wavelengths, and this property can be used to optimize image resolution. Sonographic waves, however, attenuate at higher frequencies. Improved penetration of deeper tissues is therefore obtained at a lower frequency at ranges of 3 MHz to 5 MHz. Such a range is typically used for intra-abdominal organs such as the liver, whereas more superficial structures, such as the scrotum, are optimally approached with higher frequency transducers [53].

The heterogeneity of tissues in the human body can cause the sonographic wave to be potentially scattered with a consequential loss of image resolution, depending on the tissue type. Gas is one example of a tissue in which the ultrasound beam can become defocused [34].

The elapsed time between a sonographic wave being reflected from a given tissue and that wave reaching the ultrasound transducer is also used to calculate the depth of penetration into that tissue. This is known as acoustic impedance (Z), and reflects the resistance ultrasound waves encounter given the density (ρ) of the medium they pass through and the propagation velocity (v). Thus, acoustic impedance can be calculated as the following:

$$Z = \rho v$$

(1.3)

Denser materials reflect more sonographic waves. Fluids such as blood transmit more ultrasound waves than solid materials, and reflect back fewer waves [54]. The characteristics of the returned signal, such as amplitude (A) and phase (θ),

provide information on the nature of the interaction between the ultrasound wave and the tissue, and can thus be used as an indication about the type of the underlying scanned tissue [53].

Table 2.1: Properties of different media [55].

Material Type	Density (kg/m³)	Velocity (m/s)	Acoustic Impedance (kg/m²/s × 10⁻⁶)
Air	1.2	330	0.0004
Water(20° C)	1000	1480	1.48
Soft tissue (Average *)	1060	1540	1.63
Liver	1060	1550	1.64
Muscle	1080	1580	1.70
Fat	952	1459	1.38
Brain	994	1560	1.55
Kidney	1038	1560	1.62
Spleen	1045	1570	1.64
Blood	1057	1575	1.62
Bone	1912	4080	7.8
Lung	400	650	0.26

Ultrasound wave emission is perpendicular to the surface of the transducer. Changes in the frequency of the emitted waves will control the depth of penetration of the waves to the tissue, and thus will affect the resolution of the produced images. Higher frequencies yield better resolution but a reduced depth of penetration. Higher frequencies and longer distances will result in greater attenuation [50]. The received ultrasound signal is usually amplified by increasing the gain. Decreased gain will result in a black image and thus the details will be masked, while increased gain

yields an increasingly white image [56]. The gain factor is usually set so that equally reflective structures are displayed with the same level of brightness, regardless of their depth [50].

2.3 Ultrasound Modes

As previously discussed, signals received in the ultrasound transducer are used to generate the ultrasound image for a given organ of interest, typically displayed in gray scale. Ultrasound imaging devices employ several modes of operation. The two main scanning modes are A (Amplitude)-mode and B (Brightness)-mode imaging. Other modes are M (Moving)-mode, C-mode, duplex ultrasound, color-coded ultrasound, and power Doppler ultrasound [57]. This section will briefly overview A-mode, B-mode, M-mode and Doppler mode but will use only B-mode images for analysis in later chapters.

B-mode is the most basic mode used in two-dimensional (2D) medical ultrasound imaging. B-mode imaging provides 2D structural information of the internal organs under scan by utilizing different shades of brightness on a gray scale depending on the anatomical properties of the tissues under scan [50]. Typically the brighter gray shades present echoes with greater intensity levels.

A-mode images are mainly of historic interest. In this mode a short ultrasound pulse is transmitted into the tissue and the reflected echoes are recorded as electrical voltages and then digitized.

In M-mode, returning echoes are used to generate only one line of the B-mode image and are then displayed over a time axis such that movement of the structures positioned in that line can be visualized.

In Doppler mode imaging, the ultrasound machine considers the frequency change of the sent sound wave with regards to the returning sound wave. This is a well-known physics phenomenon known as ‘Doppler Shift’ in which the changes of frequency or amount of ‘shifts’ generated by sound waves reaching moving particles are correlated with the velocity and direction of particle motion. This mode of imaging therefore uses the information on the direction and speed of the tissue motion and blood flow and presents it in color and spectral displays [54].

A sample of each mode is shown in Figure 2.3.

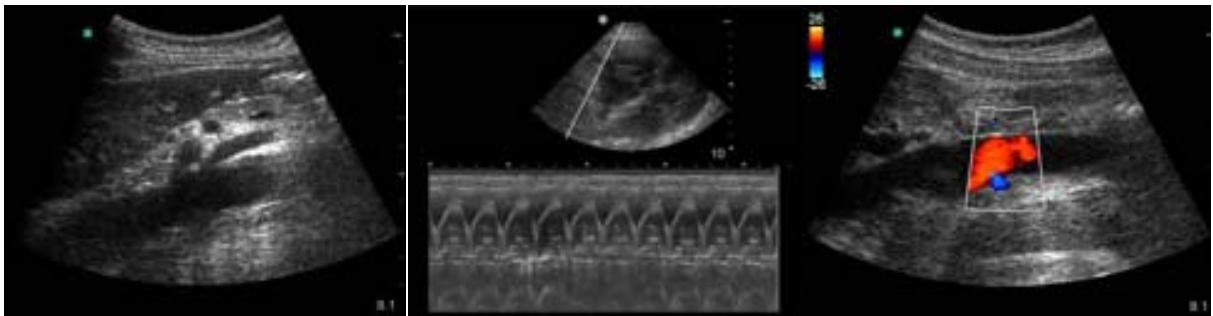


Figure 2.3: From left to right, (a) Sample of B-Mode image, (b) M-Mode combined with B-Mode image, (c) Sample of Doppler image [<http://www.sonoguide.com/physics.html>].

2.4 Interaction of Ultrasound Waves with Tissue

In a typical sonographic exam the waves pass through various tissue media, encountering different body organs with various properties and densities before being reflected back to the transducer. The portion of the echo that gets reflected back to the transducer forms the ultrasound image [58]. This interaction of the ultrasound wave with the tissue is described by the following physics concepts: attenuation, reflection, scattering, refraction and diffraction.

Various body tissues have different appearances and different brightness levels within the image due to the specific characteristics of tissue density. For example, skin is a highly reflective tissue and appears as a smooth and bright texture in the ultrasound image. Muscles usually appear dark and tendons typically appear bright. Nerves are not normally seen but their appearance is similar to tendons. Bones usually appear as bright lines due to the dramatic difference in acoustic impedance between bone and soft tissue. Fluids such as blood, effusion or cyst are generally shown as black or dark. Fatty tissue can appear as both bright and dark in the image, but the subcutaneous fat tissue layer typically appears dark.

From all of the examples mentioned, one concept that emerges is that tissues that are more solid and have higher densities such as bones and calculi produce more 'white' or brighter images, while less solid tissues or more fluid tissues appear 'black' on the image [59]. This displayed black and white image is generally referred to as a

grey scale image. A large difference in the acoustic impedance in an ultrasound image is usually referred to as acoustic impedance mismatch. A greater acoustic mismatch translates to a larger percentage of ultrasound waves that are reflected and less that are ultimately transmitted. Examples include soft tissue to bone and soft tissue to air interfaces. Indeed, the acoustic impedance of gas or air media is such that they form a virtually impenetrable barrier to ultrasound waves.

2.5 Reflection

Specular reflection and scattering are two different patterns that are important for ultrasound image formation.

Reflection is defined as a sonographic wave making contact with a distant surface of a size proportionately larger than its wavelength. Some of the ultrasound waves travel into the tissue and some are reflected back. This process is analogous to light passing from air to water, with some components of light traveling through water and others reflected back to the surface. This usually occurs in fibrous tissues such as tendons and in tissue boundaries such as those between bone and muscle, and is responsible for the bright appearance of such tissue in the ultrasound image. The reflected amount at these interfaces depends upon the acoustic properties, or acoustic impedance, of the tissues traversed by the ultrasound echo, as well as properties such as the density and compressibility of the tissue. Figure 2.4 presents the random reflection of a sound wave on a rough surface.

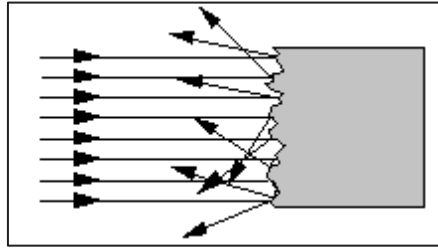


Figure 2.4: Random reflection on a rough surface.

Scattering happens at small (relative to wave length), subtle boundaries and small structures, such as tissue components, cells, and tiny fat droplets, that exist within the under scan tissue. These tissue particles subsequently absorb and re-transmit energy, effectively acting as an origin point. Sonographic scattering can be identified on an ultrasound picture by its typical texture appearing within soft tissue. Figure 2.5 shows this reflection concept, or refraction or diffraction, on impurities in the volume.

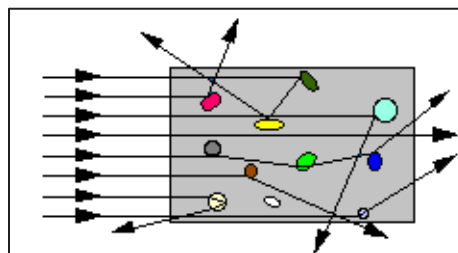


Figure 2.5: Scattering in the tissue volume.

Increasingly reflective surfaces generate a powerful echogenic signal that manifests visually on a sonographic image as a bright spot, and on weakly reflective interfaces as relatively hypoechoic dark spots. Areas lacking acoustic interfaces, such as vessel lumens or other liquid containing structures (blood, ascites, bile, or urine,) provide no reflection or spot on the screen or black space on the sonographic monitor. Sonographic waves that make contact with gaseous or solid densities have their acoustic energies reflected, making it difficult to visualize surrounding structures. Figures 2.6 and 2.7 present reflection and scattering concepts in ultrasound systems.

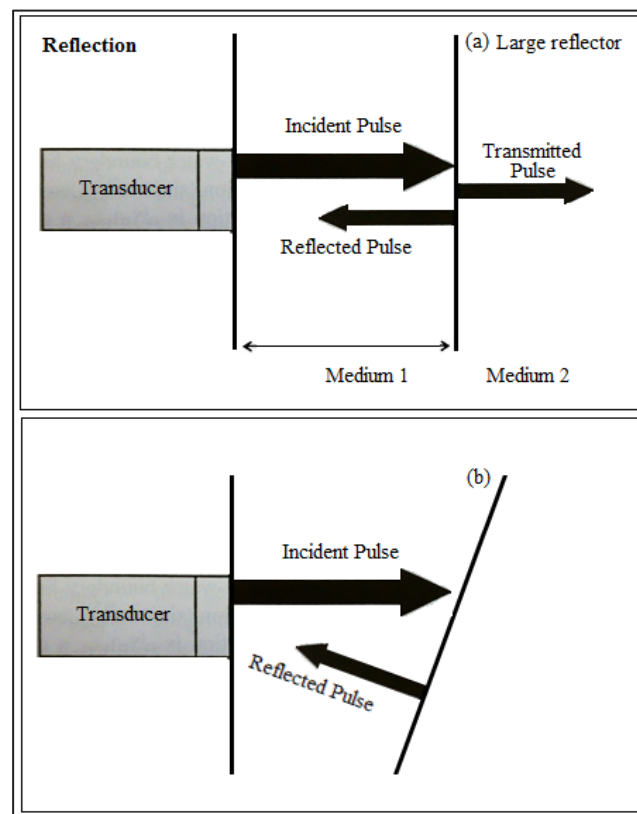


Figure 2.6: Reflection of an ultrasound wave by a large reflector (scattering between the unhomogeneous borders of two different mediums). In (a) the reflector appears bright and in (b) the reflector appears dark.

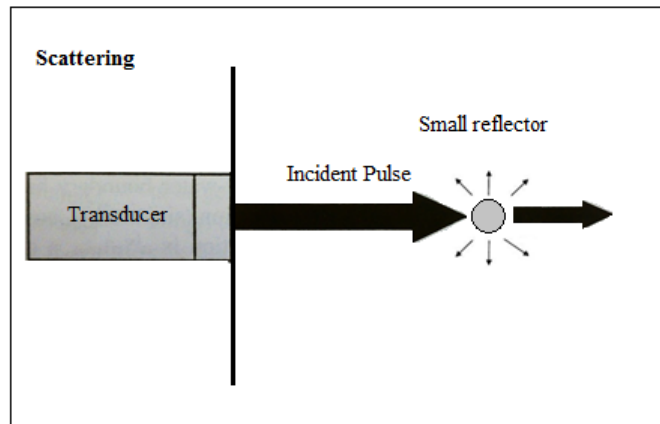


Figure 2.7: Scattering of ultrasound wave in interaction with small objects.

Chapter 3

Envelop Characteristics

3.1 Statistical Modeling of Speckle

As the first step in speckle research the goal was to do a mechanism study and better understand the characteristics of both fully developed speckle and partially developed speckle. Wagner et al. [2] showed that the resolution cell histogram amplitudes of the envelope-detected RF signal backscattered from a uniform area with an adequately high scatterer density possesses a Rayleigh distribution with mean m proportional to the standard deviation σ (with $\mu/\sigma = 1.91$). Mathematical statistical modeling for speckle in ultrasound images may be approximated as multiplicative, taking into consideration the log compression of the ultrasound image. Specific consideration has to be given to the effect of the additive noise, which is

considerably smaller as compared to multiplicative noise. Some other research work modeled tissue as a sound absorbing medium containing scatterers, which scatter the sound waves [25, 28, 32, 60].

When the number of scatterers per resolution increases, it yields Gaussian statistics for the radio-frequency (RF) but in the case of partially developed speckle, the statistics for the RF signal do not follow Gaussian distribution. In the setting of a partially developed speckle artifact however, the distribution has a higher likelihood of being non-linear. Therefore, in order to model the statistical behavior of the RF data to cover detection of partially developed speckles K distribution framework is used while the Raleigh distribution doesn't cover partially developed speckle behaviors and is used in case of fully developed speckles , as described [36]. Therefore in addition to Raleigh distribution, as described in [28], K distribution should also be considered.

3.2 Statistics of Fully-Developed Speckle (FDS)

As mentioned earlier, speckle noise has a stochastic nature, thus when making any conclusion about the ultrasound images, one needs to understand and describe the speckle pattern statistically. For this purpose the same statistics that have been used in previous researches and speckle literature have been utilized in this work [33, 61].

Chapter 1 and 2 described a random deterministic speckle pattern formed due to the presence of a large number of scatterers present in the medium (tissue) being

scanned. Diffuse and coherent scatterings were introduced. Diffuse scatterings result from a large quantity of scatterers with random phase in the resolution cell of the sonographic beam that leads to speckle artifact in the reconstructed image. Coherent scattering was discussed when the scatterers in the resolution cell are in phase leading to light or dark spots on a sonographic image. .

Each of the diffuse scatterers under scanned tissue contributes to a portion of the echo signal in a sum that is commonly known as a ‘random walk’. This two-dimensional (2D) random walk, also known as ‘drunkard's walk’, is in fact the mathematical formalization of a path that consists of a succession of random steps in the complex plane. Random walk has an important value within the statistical analysis of speckle and optics in general. Figure 3.1 demonstrates the random walk schematically.

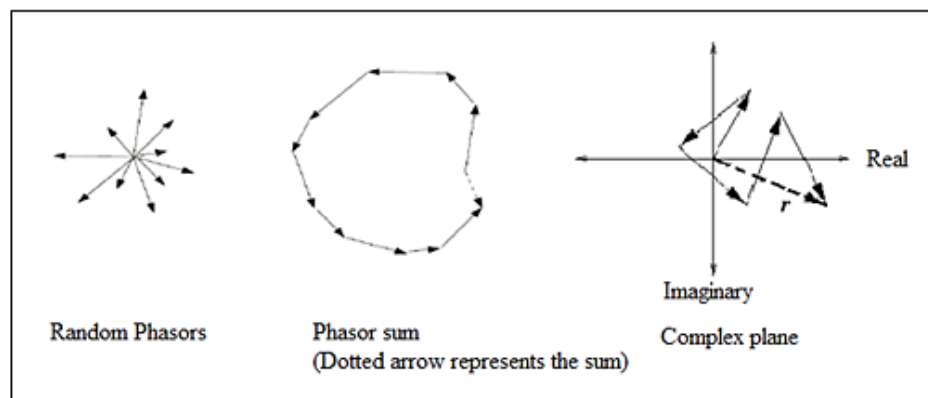


Figure 3.1: Random walk schematic view (Image courtesy of Vinayak Dutt).

This figure shows diffuse scatterers in a random walk that comprises the consequently received complex echo r . The values of the magnitude of r for these scatterer groups follow the Rayleigh PDF.

Speckle brightness increases with fewer, longer steps in the random walk, compared to many shorter steps. Increase in scatterer density will also result in the speckle brightness. The random walk problem is explained in detail and modeled in Goodman's book [61].

By designating each step of the walk as an independent random variable, it is possible, over numerous walks, to apply the Central Limit Theorem to their sum. According to this theorem, and if one considers the number of steps in the sum to be N and $N \rightarrow \infty$ (assuming that N is very large) then N is Gaussian. Therefore, in fully developed speckle (FDS) from diffuse scatterers alone, this complex RF echo signal (presented as r) has a mean equal to zero and a 2D Gaussian PDF in the complex plane. Envelope detection removes the phase component, and thus creates a signal with a Rayleigh amplitude PDF [61] that is described in detail in section 3.2.1.

3.3 Statistical Models

Previous research proposed a number of models and developed different distributions for the envelope signal in medical ultrasound applications [2]. Some of these distributions include the Rayleigh distribution and the K distribution. Given the inherent positive and negative aspects of each model, it is necessary to consider the

utilization of both statistical approaches, Rayleigh distribution and the K distribution, to better characterize statistical behavior of the RF signal [29].

Figure 3.2 classifies the different statistical models that are developed for the envelope signal used in medical ultrasound imaging applications [5], including Raleigh distribution and K-distribution, and different scattering types. This figure demonstrates that the mechanism is different, based on number of speckles. For fully developed speckles it follows Raleigh distribution, but for partially developed speckles, when there are fewer scatterers in each resolution cell, K-distribution is followed. Based on this figure, it can be seen that partially developed speckles follow K-distribution, while when the number of scatterers is high, the patch has a Raleigh/Rician distribution.

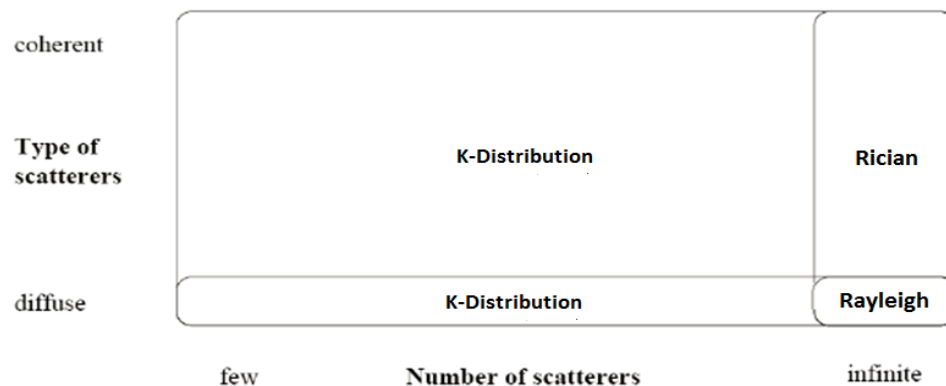


Figure 3.2: Statistical models and scattering types [5].

Because the derivations of these distributions have been covered extensively in the existing literature [28, 64], only a brief review of Rayleigh and K-distributions is presented in this chapter.

3.3.1 Rayleigh Distribution

The Rayleigh distribution is observed when a high magnitude of nearly identical and randomly distributed scatterers influence the echo signal [65]. Assuming FDS, the envelope RF image patch, $Y = \{ y_{i,j} \}$, is modeled using Rayleigh statistics, where the PDF is provided by [29]:

$$p(y_{i,j}) = \frac{y_{i,j}}{\sigma_{i,j}^2} e^{-\frac{y_{i,j}^2}{2\sigma_{i,j}^2}} \quad (3.1)$$

with $y_{i,j}$ representing the envelop amplitude. The parameter σ is the standard deviation and σ^2 is the variance of the complex Gaussian distributed in-phase and quadrature components of the complex echo envelope [33, 66] (i.e. both real and imaginary portions.)

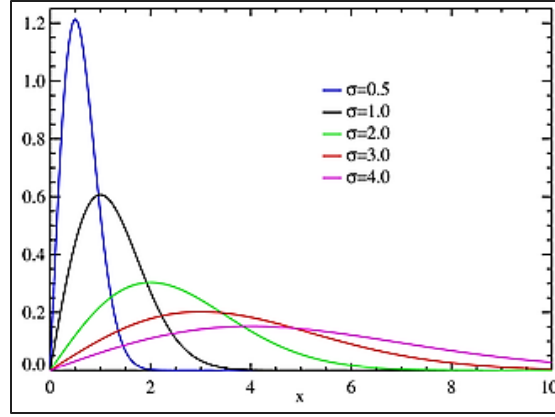


Figure 3.3: Probability density function for Rayleigh distribution for different σ values.

The echogenicity represents a confluence of the foundational parameter of the Rayleigh distribution, $\Sigma = \{\sigma_{i,j}\}$, related to pixel intensity of the RF image $y_{i,j}$ and this latter term is associated to the acoustic properties at the corresponding location (i, j) [29]. Figure 3.3 shows a typical PDF for the Rayleigh distribution. Let $Z = \{z_{i,j}\}$ be a $N \times M$ B-mode image corrupted by speckle where each pixel is generated according to the following Log-Compression law [29]:

$$z_{i,j} = a \log(y_{i,j} + 1) + b \quad (3.2)$$

where (a, b) are unknown parameters that are respectively related to contrast and brightness. Given the equation (3.1), the distribution of the observed pixels z ,

$$p(z_{i,j}) = \left\{ \left| \frac{dy}{dx} \right| p(y) \right\}_{z=z_{i,j}} = \frac{y_{i,j}(y_{i,j}+1)}{a\sigma_{i,j}^2} e^{-\frac{y_{i,j}^2}{2\sigma_{i,j}^2}} \quad (3.3)$$

where $y = e^{\frac{z-a}{b}} - 1$. As pointed out in Seabra [67], equation (3.3) defines a double exponential distribution with known standard deviation (SD) analytical expression [68], yielding an estimate for a :

$$\hat{a}_{i,j} = \sqrt{24} \frac{\sigma_z(i,j)}{\pi} \quad (3.4)$$

where $\sigma_z(i,j)$ is the SD of the observations inside the window w , centered at the (i,j) th pixel.

To estimate the parameter b , it is first necessary to consider the minimum of the observed pixels $z_{i,j}$ given by:

$$\begin{aligned} s = \min\{i, j\} &= a \log(\min\{y_{i,j}\} + 1) + b \\ &= a \log(t + 1) + b \end{aligned} \quad (3.5)$$

which means that:

$$b = s(Z) - a \log(t(\sigma, L) + 1) \quad (3.6)$$

with $Z = \{z_{i,j}\}$. The distribution of b , derived in Sanches [69], is:

$$p(b | s(Z), \sigma) = \frac{L}{a\sigma^2} t(t+1) e^{-\frac{L}{2\sigma^2} t^2} \quad (3.7)$$

where $t = e^{\frac{s-b}{a}} - 1$. An estimator of b is found by computing the expected value of $b_{i,j}$ using a numerical approach, such that [29]:

$$\hat{b}_{i,j} = \sum_{k=1}^L b_{i,j}(k) p(b_{i,j}(k) | s, \beta_{i,j}) \quad (3.8)$$

where $b_{i,j}(k) = k s / (L - 1)$ and $k = 0, 1, \dots, L - 1$ are L uniformly distributed values in the interval $[0, s]$, since $b \geq 0$ and from equation (3.6), $b \leq s$. In equation (3.8),

$$\beta_{i,j} = \sqrt{\frac{1}{2nm} \Omega_{k,l} q_{k,l}^2} \quad (3.9)$$

is the Maximum Likelihood (ML) estimation of $\sigma_{i,j}$ from the pixels inside the window $[w, q_{k,l}]$ [29].

The estimators of a and b , considered constant across the image, are obtained by averaging the estimates $\hat{a}_{i,j}$ and $\hat{b}_{i,j}$, such that: $\hat{a} = \frac{1}{NM} \sum_i \sum_j \hat{a}_{i,j}$ and $\hat{b} = \frac{1}{NM} \sum_i \sum_j \hat{b}_{i,j}$.

These parameters (\hat{a}, \hat{b}) are used to retrieve the envelope RF image according to:

$$y_{i,j} = e^{\frac{z_{i,j} - b}{a}} - 1 \quad (3.10)$$

In some sonographic subspecialties, such as echocardiography, the Rayleigh distribution fits to account for properties of t model reflections from blood, but this assumption does not hold with complex structures [29, 70].

The K distribution has been proposed in prior work as another approach to estimate various tissue types in sonographic envelope imaging [29]. The K distribution is a more general model that approaches the Rayleigh distribution in the limit $\mu \rightarrow \infty$ [71] where μ is a measure of the effective number of scatterers per resolution cell. In order to interpret the parameter μ correctly, the resolution cell volume must be known.

3.3.2 K-Distribution

Originally Shankar et al. developed the K-distribution for the envelope signal to describe a model for scattering from tissues [72]. The interest in using the K-distribution is based upon its potential for modeling both fully developed speckle (FDS) (e.g., blood pool) and partially developed speckle (non-FDS) (e.g., tissue area) [29]. This section will overview this distribution and the corresponding PDF for the RF signal [36, 73].

The individual components of each scatterer embedded in the resolution cell leads to the backscattered sonographic wave, and can be quantitatively modeled as a random walk in a complex plane as described previously [29].

The analytic signal from this mathematical model can be conceived as a random process reliant on multiple scatterers present inside the resolution cell and their relative position and contribution. Consequently, a joint density function of the envelope and phase can be obtained in terms of the statistical properties of the phase and scatter amplitude. An important quantitative outcome of this assumption is a K-

distribution in which the scatterers phase assumes a uniform distribution with an amplitude that is itself modeled as a K-distribution [29, 74].

The RF signal corresponds to the real part of the analytic signal. Thus as a result, the RF signal PDF relates to a K-distribution obtained by integrating the PDF as described briefly below and fully in prior works [71, 73]:

$$pdf^{RF}(x) = \frac{b}{\sqrt{\pi}\Gamma(\nu)} \left(\frac{b|x|}{2}\right)^{\nu-0.5} K_{\nu-0.5}(b|x|) \quad (3.11)$$

In this formula, Γ is the Gamma function and $K_{\nu-0.5}$ represents the modified Bessel function of the second kind of order $\nu - 0.5$. This mathematical relationship is defined by its two parameters ν and b , such that ν (the order) controls the shape and b the scale of the PDF. Parameter b can be expressed and calculated [71]. The corresponding distribution to this formula for the sonographic images is termed the K_{RF} . The K_{RF} provides the basis for segmentation of sonographic images when there is a partially developed speckle. The K_{RF} distribution, though, has the several drawbacks based upon prior research [70, 73]. Numerical simulations in prior literature show that estimation bias increases substantially as ν increases, which consequently results in unreliable estimates in blood regions (i.e. $\nu \gg 1$). An additional disadvantage is that Equation (3.11) suggests a repeated Bessel function. This increases the computational cost of the algorithm. Figure 3.4 shows an example of a PDF K-distribution.

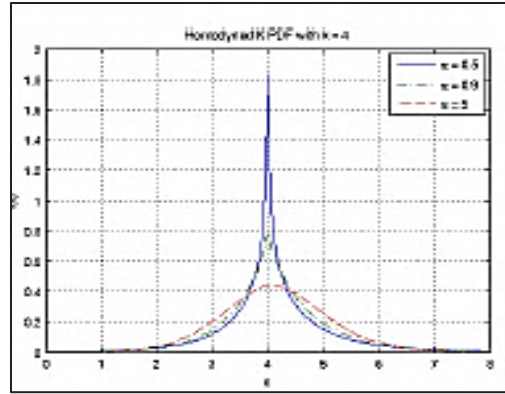


Figure 3.4: Probability density function K-distribution.

3.4 Features for Speckle Discrimination: SNR, Skewness, and Kurtosis

After the mechanism study an extensive review of literature was done to identify the best possible statistical features that describe speckle for classification purposes. Also, application of different features such as Entropy, Homogeneity, Energy, Average Intensity, SNR, Skewness, Kurtosis, and Raleigh likelihood was investigated to identify the optimum combination of these features that would lead to better classification results.

Prior approaches used various methods for echo envelope signal processing, speckle classification and speckle suppression. These statistical features were obtained by fitting a statistical model to data to describe the underlying material (tissue) properties (speckle vs. non-speckle).

Dutt [75] studied and used *SNR* to estimate the scatterer number density using K-distribution. Prager et al. used *SNR* and skewness as statistical features in their speckle study investigations [76]. Wear et al. [77] used the square of the *SNR* to estimate scatterer amount per resolution cell. However, the square of the *SNR* was found to be sensitive only in the range of small scatterer number densities, i.e., one or two scatterers per resolution cell [77].

As mentioned earlier, the K-distribution has been developed for the envelope signal [35]. Prior work [71, 79] only used *SNR* (signal-to-noise ratio or *R*) and Skewness (*S*) features, and suggest that these features are among the most effective for classification purposes. This thesis suggests using Kurtosis as a new statistical feature and also investigated a feature describing Raleigh distribution (Raleigh Likelihood estimation).

Informed by this literature, this current work proposes a minimum of 3 features is sufficient to best describe speckle and improve previous results. Also optimum patch separability was achieved by choosing three classifiers - *SNR*, Skewness (*S*), and Kurtosis (*K*) - for classification purposes. Previous work has also studied the use of *SNR* and the skewness of samples of the echo envelope raised to the positive powers [33]. Therefore, in this thesis work, sample kurtosis has been applied as a novel statistical approach for the purpose of speckle classification, and data that demonstrates improvement over previous approaches is described.

The *SNR* measures the mean divided by the standard deviation, the skewness quantifies the asymmetry of the distribution, and the kurtosis characterizes the

peakedness of the distribution. In particular, these classifiers can be used to describe the distribution of envelope samples.

Based on [69, 81], the following statistics on arbitrary powers ν of the image patch A were calculated where [80]:

$$\begin{aligned}
 R &= \frac{\text{mean}}{\text{Std}} = \frac{\langle A^\nu \rangle}{\sqrt{\langle A^{2\nu} \rangle - \langle A^\nu \rangle^2}} \quad (\text{SNR}) \\
 \text{Skewness} &= \frac{(A^\nu - \langle A^\nu \rangle)^3}{(\langle A^{2\nu} \rangle - \langle A^\nu \rangle^2)^{3/2}} \\
 \text{Kurtosis} &= \frac{(A^\nu - \langle A^\nu \rangle)^4}{(\langle A^{2\nu} \rangle - \langle A^\nu \rangle^2)^{4/2}}
 \end{aligned} \tag{3.12}$$

where R is the signal-to-noise ratio (SNR), A is the amplitude of the ultrasound signal data or the RF envelope, and ν equals signal power. Std is the standard deviation, and $\langle \dots \rangle$ represents mean [80]. Various values of ν (powers of A) can be used to calculate the R , S and K [5, 81-83] for better discrimination of these features. Previous research has employed extensive number of computer simulations to investigate the optimum values for R and S [81, 82]. The terms in the denominators of equations are guaranteed to be positive, ensuring that these functions are well defined because division by zero does not occur for any valid combination of model parameters.

For each image patch A , the proposed statistical features in the equation (3.12) and the Maximum Likelihood (ML) $\beta_{i,j}$ in equation (3.9) were calculated to estimate

the each patch data (A) according to the Rayleigh distribution. Extremely large numbers of computer simulations have been done in this work to investigate the optimum values for v_R , v_S and v_K . Following feature extraction for each image patch A, the classification scheme shown in Figure 3.5 was used to classify each image patch to FDS or non-FDS [29].

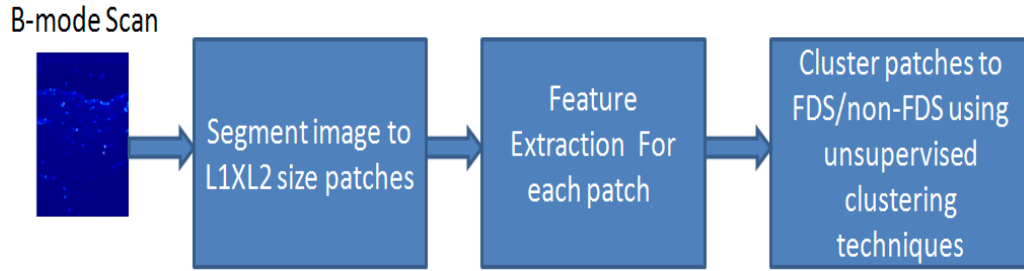


Figure 3.5: Pipeline for feature extraction for each patch and Speckle detection (classification) [29].

Additionally patch size selection effect on classification results was investigated and is reported in the results section in Chapter 7.

To characterize RF signal behavior the effective number of scatterers per resolution cell (μ) > 10 can classify FDS [80]. Thus, using statistical features one can classify each patch to speckle (FDS) and non-speckle (non-FDS) patches by using clustering techniques that will be described in Chapter 4.

Also, classification results is dependent on choosing the optimal values of v_R , v_S and v_K that substitute v in equation (3.12). As mentioned earlier thesis proposes using Kurtosis as a new statistical feature and thus investigated the optimum value set of v_R , v_S and v_K to substitute in equation (3.12) which is discussed in next section. However it must be noted that these values of v are application dependent and will change depending on the criteria.

3.4.1 Calculating Statistical Features Based on v

Optimum speckle classification depends on how well one can discriminate the desired statistical features. Discrimination of these features are all dependent of the values of v_R , v_S and v_K .

Ad hoc values for moment orders in literature had been reported. Given the results from prior work [80], values of v more than one ($v > 1$) are suggested to perform well [29]. Depending on the data correlation, thousands of sample data are necessary to accurately calculate R and S [75, 80] and K . In Dutt [75] and Prager [83], values of v that reduce this sample size were obtained. Useful variability of clusters of sample data with different μ and k values were maximized in [83] to find the optimal v , a method that is scrutinized by [84]. Since R and S are different order moments of sample data, optimal values for v in R and S are not necessarily identical [80].

Most parameter estimation algorithms involve the use of moments, a choice that has been studied and investigated [28]. The use of integer moments, due to the analytical convenience, [28] and fractional moment orders, due to more robust

estimates [85, 86], have been reported. Prager et al. [83] found a fractional moment order (~ 1.8) to be optimal for speckle detection. Other studies have reported that finding an optimal value for the moment order may not be justifiable and that a simple optimum moment order for statistical features does not exist [84]. Based on the lack of consensus in the literature for what constitutes the optimal moment orders, it would be desirable to employ an estimation algorithm capable of using arbitrary moment orders.

This research proposes a more systematic way to analyze and find the optimum value set of v_R , v_S and v_K , depending on the criteria and application which is discussed in the results section. (Appendix A: **PatchStat Toolbox**). Extremely large numbers of computer simulations, with thousands of sample data, have been done to investigate the optimum values for optimum values for v_R , v_S and v_K .

Chapter 4

Classification Techniques

4.1 Background

This chapter will overview some previously known methods and algorithms, and discuss how such algorithms were applied to this research work and the data set. This chapter will review some of the very known classification techniques that has been employed in this research for speckle classification purposes.

4.2 Clustering Techniques

Data clustering means dividing data into crisp (hard) or fuzzy subsets. Hard clustering for a given data set X means partitioning the data into a specified pre-defined number of subsets (clusters) of X based on each object membership within a

specific cluster. In such a model, K equals the number of clusters [29]. While hard partitioning has the advantage of representing a simple approach for data clustering, it lacks reliability and precision. However, the second method of fuzzy data clustering, allows objects to simultaneously belong to multiple clusters with varying magnitudes of membership. This makes fuzzy clustering more advantageous compared to the hard clustering techniques because objects that may exist on the boundaries between several classes are not required to fully belong to one of the clusters [29]. The resulting hypothesis is therefore that in our data set, fuzzy clustering is a much better method than hard clustering techniques. This assertion was tested utilizing both hard (K-means and K-mediod) [87] and fuzzy partitioning (Fuzzy C-Means, Gustafson-Kessel and Gath-Geva) techniques for competitive speckle detection [88, 89].

4.2.1 K-means and K-medoid

Hard partitioning methods are useful for their simplicity but lack numerical precision, and thus not reliable. From an NxN dimensional data set, K-means and K-medoid algorithms [87] allocate each data point to one of the K clusters. For this work two clusters were utilized (FDS and non-FDS), to minimize the within-cluster sum of squares (distance norm) [29]:

$$\sum_{i=1}^K \sum_{t \in A_j} \|X_i - V_j\|$$

(4.1) where A_j is a set of objects (data points) in the j-th cluster. In K-means, V_j is the mean for those points over cluster j which are called the cluster prototypes or centers. In K-medoid clustering the cluster centers are the nearest objects to the mean of the

data in one cluster. K-mediod is particularly useful when there is a lack of data space continuity [29].

4.2.2 Fuzzy C-means clustering

The Fuzzy C-Means (FCM) clustering algorithm is based on the minimization of an objective function called C-Means functional [29]. In FCM, any one piece of data can belong to two or more clusters. FCM is defined by Dunn as:

$$J = \sum_{j=1}^C \sum_{i=1}^N (\mu_{ji})^m \|X_i - V_j\|^2 \quad (4.2)$$

where μ is membership degree, m is order, and, V_j is the center of the cluster j , and all these parameters need to be determined. The minimization of the C-Means functional J (the cost function) represents a nonlinear optimization problem that can be solved variously using different methods (e.g., grouped coordinate minimization, simulated annealing, or genetic algorithms) [29]. However, the most common method is using a Picard iteration [90] through the 1st order conditions for stationary points of J , that is known as the Fuzzy C-Means algorithm [29].

The stationary points of function J can be calculated by adjoining the following constraint for the fuzzy partitions $U = [\mu_{ij}]$:

$$\sum_{j=1}^C \mu_{ji} = 1, \quad 1 \leq i \leq N \quad (4.3)$$

If $\|X_j - V_j\|^2 > 0$ and $m > 1$, then (U, V) may minimize J only if

$$\mu_{ji} = \frac{1}{\sum_{k=1}^c \left(\frac{\|X_i - V_j\|}{\|X_i - V_k\|} \right)^{\frac{2}{m-1}}}, \quad 1 \leq j \leq c, \quad 1 \leq i \leq N \quad (4.4)$$

$$V_j = \frac{\sum_{i=1}^N \mu_{ji}^m X_i}{\sum_{i=1}^N \mu_{ji}^m}, \quad 1 \leq j \leq c,$$

The resulting solution fulfills the constraints on fuzzy partitions. Also it should be noted that the equation for the cluster centers gives V_j as the weighted mean of the data items that belong to a cluster, where these weights are the degrees of membership [29], and thus as a result, this algorithm is designated as "C-Means". The FCM algorithm computes with the standard Euclidean distance norm, which induces hyperspherical clusters [29], and therefore it only detects clusters with the same shape and orientation, because the common choice of norm matrix is the identical matrix.

4.2.3 Gustafson-Kessel clustering algorithm

Gustafson-Kessel (GK) clustering algorithm represents an extended version of the standard fuzzy C-Means algorithm and utilizes an adaptive distance norm for detecting clusters of different geometrical shapes in one data set [89]. Each cluster has its own norm-inducing matrix A_i :

$$D_{ikA}^2 = (x_k - v_i)^T A_i (x_k - v_i), \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \quad (4.5)$$

The matrices A_i are employed as optimization variables in C-Means functional and therefore permitting each cluster to adapt the distance norm to the local topological structure of the data [29]. The cost function for the GK algorithm is:

$$J = \sum_{i=1}^C \sum_{k=1}^N \mu_{ik}^m D_{ikA_i}^2 \quad (4.6)$$

where the $N \times C$ matrix $U=[\mu_{ik}]$ represents the fuzzy partitions. D is the distance as defined in (4.5). To optimize the cluster's shape, it is proven that A_i should be as following [29]:

$$A_i = [\rho_i \det(F_i)]^{1/n} F_i^{-1} \quad (4.7)$$

where ρ_i is a fixed value for each cluster and $\det()$ means determinant of a matrix and F is the fuzzy covariance matrix defined below as:

$$F_i = \frac{\sum_{k=1}^N \mu_{ik}^m (X_k - V_i)(X_k - V_i)^T}{\sum_{k=1}^N \mu_{ik}^m} \quad (4.8)$$

A_i is a squared Mahalanobis distance norm [29, 91] between point X_K and the cluster mean V_i , where the covariance is weighted by the membership degrees in $U=[\mu_{ik}]$. The Mahalanobis distance is used for identifying and analyzing similarity of a data set samples or a pattern to a known one. This distance is different from the Euclidean distance as it is scale-invariant (not dependent on the scale of measurements) and also considers the correlations of the data set.

4.2.4 Gath-Geva fuzzy Classifier

The Gath-Geva (GG) fuzzy clustering algorithm is indeed an extension of Gustafson-Kessel algorithm but considers the size and the density of the clusters as well. GG algorithm utilizes a distance norm based on the fuzzy maximum likelihood estimates (FMLE), proposed by Bezdek and Dunn [29, 92]:

$$D_{ik}(X_k, V_i) = \frac{\sqrt{\det(F_{wi})}}{\alpha_i} \exp(0.5(X_k - V_i)^T F_{wi}^{-1} (X_k - V_i)) \quad (4.9)$$

In contrast to the GK algorithm, this distance norm involves an exponential term and consequently decreases faster. F_{wi} is the fuzzy covariance matrix of the i -the cluster, given by [29]:

$$F_{wi} = \frac{\sum_{k=1}^N \mu_{ik}^w (X_k - V_i)(X_k - V_i)^T}{\sum_{k=1}^N \mu_{ik}^w}, \quad 1 \leq i \leq c \quad (4.10)$$

If $w = 1$, F_{wi} is identical to the original FMLE algorithm, but if $w = 2$, the partition becomes more fuzzy to compensate the exponential term of the distance norm [29]. The difference between the matrix F_i in GK algorithm and the F_{wi} in (4.10) is that the latter does not involve the weighting exponent m , instead consisting of $w = 1$. This is due to two weighted covariance matrices arising as generalizations of the classical covariance from two distinct concepts. The α_i is the prior probability of selecting the cluster, given by:

$$\alpha_i = \frac{1}{N} \sum_{k=1}^N \mu_{ik} \quad (4.11)$$

The membership degrees μ_{ik} are interpreted as the posterior probabilities of selecting the i -th cluster given the data point x_k . Gath and Geva [93] reported that the fuzzy maximum likelihood estimates clustering algorithm is able to detect clusters of varying shapes, sizes and densities [29]. The cluster covariance matrix is used in conjunction with an "exponential" distance, and the clusters are not constrained in volume. However, this algorithm is less robust because it requires good initialization, due to the exponential distance norm; it converges to a near local optimum [29].

Chapter 5

Data Generation

5.1 Introduction

In this work, two types of data were used for the analysis: simulation ultrasound images and real ultrasound images. In order to test the proposed developed algorithm, ground truth images are optimal. As this is not possible with real data, there is a need for a simulation program and phantom data in which the parameters and locations of every object and their exact properties in the image are known. Real sonographic images were also required later on for analysis and will be discussed in subsequent sections.

5.2 Simulations

To assess the proposed speckle approach and analyze the performance of each proposed unsupervised classification method, we used simulated B-mode images [29] of different tissues. The Field II simulation program was used to obtain simulation data [94]. Generated simulations were used as ground truth for testing and evaluating the proposed algorithms. The following sections briefly provide an introduction to this simulation program and will represent and analyze the produced images in the results section.

5.2.1 Field II simulation

Field II is a worldwide known program for simulating all types of advanced ultrasound transducer fields and ultrasound images, for different types of transducers. The Field II program can simulate linear imaging and realistic images of human tissue [29].

The Field II program is freeware and runs under Matlab on different operating systems including Windows, Linux, Mac, SUN, IBM, and others. In the analysis presented in this work, Matlab was used on the Windows operating system (OS). More information about the Field II program, including the background, basics and parameters of the system, such as frequency, sampling (recommended value of 100 GHz), transducer settings, etc. are described on the Field II program website (<http://field-ii.dk>). The website also describes in detail how to set different values for

the system parameters in the user's manual of the program (http://field-ii.dk/documents/users_guide.pdf) [95-98].

Researchers at Duke University created a Graphical User Interface (GUI) so that the user can easily set the parameters and run the program based on the application and phantom they need, in order to generate the simulation image they desire. Figure 5.1 is a screenshot of this program. Informed by this GUI and based on the Field II simulator, this thesis work resulted in the creation of a new GUI for phantom generation and analysis purposes, called the 'Ultrasound Tool' or 'UTool'. This resulted in simulated cyst, kidney, fetus and heart phantoms, created using the parameters and values on the Field II program website and re-generated using code generated by this thesis, which will be discussed later. The utility functions of the Field II are freely available through the Field II simulator application interface (API) (http://field-ii.dk/?./downloading_7_12.html). It should be mentioned that the Field II simulation is not integrated software and it only offers utility functions and specifically implemented mathematical models for RF data generation and phantom simulation.

Summing the field from an array of point scatterers allows for the imaging of artificial phantoms. A single RF line in a sonogram is computed by adding the response from an array of scatterers, in which the scattering strength is ascertained by the density and speed of sound disruptions in a given tissue (http://field-ii.dk/examples/cyst_phantom/cyst_phantom.html). Homogeneous tissue is consequently comprised of a collection of randomly placed scatterers that have a

scattering strength with a Gaussian distribution, where the variance of the distribution is determined by the backscattering cross-section of the particular tissue [99]. Such phantoms usually consist of 100,000 or more scatterers, and simulating 50 to 128 RF lines can take several days depending on the computer used [99].

The screenshot shows the 'Linear Array' GUI with the following parameters:

Transducer Parameters		General Settings	
Number of Elements	128	Sampling Frequency (MHz)	100
Kerf (mm)	0.02	Sound Speed (m/s)	1540
Width (mm)	0.2	Target Type:	Structured&Diffuse
Height (mm)	5	Number of Axial Points	1
Focal Pt. [lat elev axial] (mm)	0 0 30	Dist. b/t Axial Points (mm)	0.5
Center Frequency (MHz)	7.5	Axial Dist. from Focus (mm)	0
Fractional Bandwidth (%)	50	Number of Lateral Points	1
		Dist. b/t Lateral Points (mm)	0.5
		Lat. Dist. from Focus (mm)	0
		Diffuse Parameters:	
		Diffuse ROI Size (mm)	1
		Diameter of Cyst (mm)	0
		Diffuse Scatterer Density	10
		Amplitude Ratio: struc/diff	20

Image Size		Excitation Pulse:	
Lateral ROI Range (mm)	2	Center Frequency (MHz)	7.5
Lateral Increment (mm)	0.02	Number of Cycles	5

Calculate:		Plot to save:	
Plot:		Save filename	testing

Figure 5.1: Field II GUI implemented in Duke University
[\[http://dukemil.egr.duke.edu/Ultrasound/Simulation/field_ii/field_ii_gui.html\]](http://dukemil.egr.duke.edu/Ultrasound/Simulation/field_ii/field_ii_gui.html).

The code simulates the image line by line. Simulating the lines takes roughly 40-45 minutes on a 3.40 GHz CPU PC. But if clusters are used the image can be generated in around 10 minutes.

For the one million scatterers the creation of one RF line takes approximately 70 minutes for a 3 GHz CPU to generate one RF line a rate of about 270 point scatterers per second.

5.2.2 UTool Software: A Novel Ultrasound Simulation GUI for Phantom Generation

As discussed in previous sections, Field II simulation software is a well-characterized, MATLAB-based freeware that can generate phantom data with a customized number of scatterings and tissues or organs. This work expands upon this freeware with the development of a novel simulation software, based on utility functions freely available on the Field II simulator application interface (API,) which can be used to generate simulated phantoms for algorithm testing. This new GUI, called ‘UTool’ or ‘Ultrasound Tool’, was used to model simulated objects/organs to include four different types of phantoms: cyst, kidney, fetus and heart.

The UTool software is a simple to use GUI that runs in Matlab. Through this GUI, one can set the physical characteristics of a transducer. The parameters that can be set are the number of scatterings, number of physical and active elements, transducer center, frequency and element height.

Once the phantom parameters have been defined, the actual phantom can be created using the Field II code. Next, the Field II code can be run by calling `simulate_image.m`. For the simulation, it is assumed that the imaging system has a linear, space-invariant point spread function (PSF) and the transducer is linear. In this

analysis, specifically, a linear array of 192 elements was used, each being 5 mm in height, centered at 3.5 MHz. Using the developed GUI one can set or change these ultrasound sensor variables. The output from this stage is the raw ultrasound RF data. A top-level block diagram of the phantom generation is shown in Figure 5.2 below.

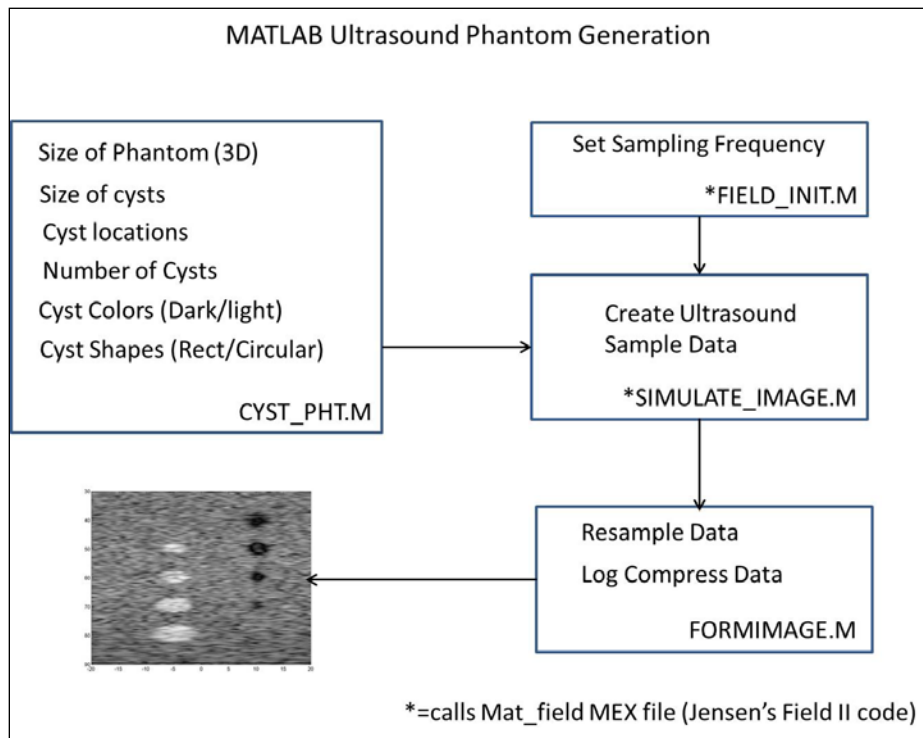


Figure 5.2: Phantom generation process.

The code for generating phantoms is represented in Appendices.

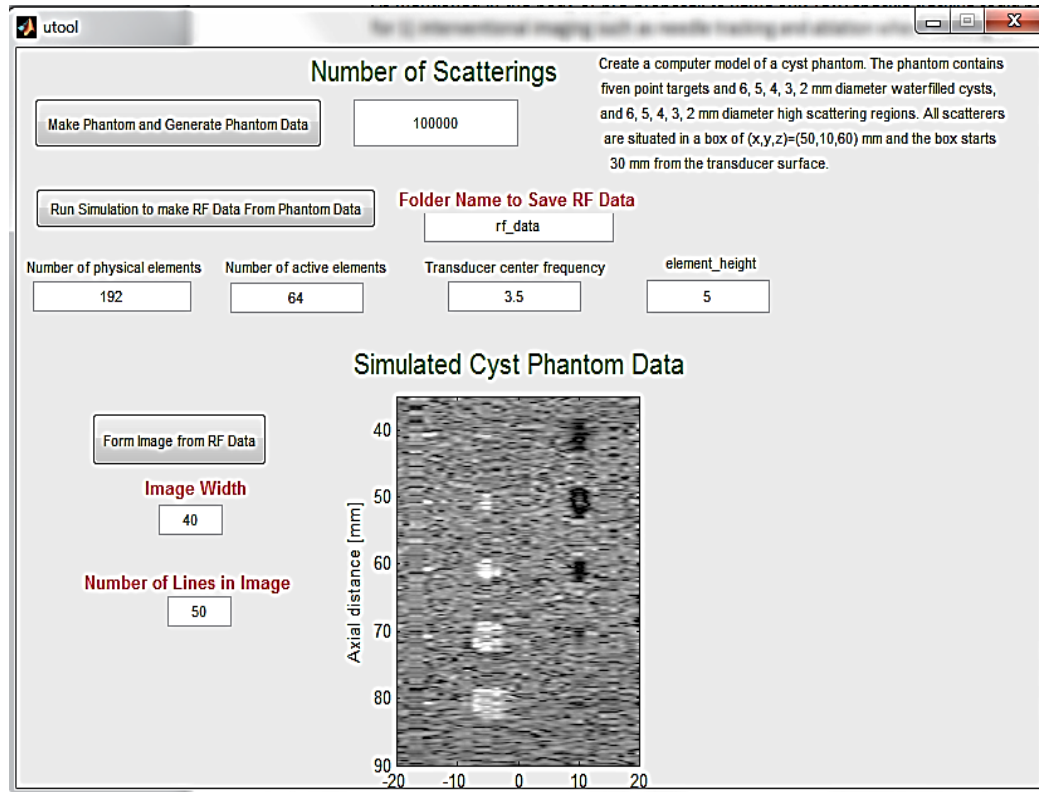


Figure 5.3: A screen shot of the UTool GUI with the example of a generated cyst phantom. Bench phantoms for cysts and fetal exist, which other worldwide research groups have generated and used, which UTool can be compared to.

Chapter 6

Methodology (Speckle Detection)

6.1 Introduction

This section provides an overview of the two robust methods developed in this research work for speckle detection purposes: Semi-Automatic and Fully-Automatic classification. After creating ground truth data, each patch of the simulated phantom images was labeled as either FDS or non-FDS, using expert opinion. The same simulated images were then used to validate the results of the proposed methods. In the case of real data, using expert opinion would not be a possibility, thus the proposed fully automatic method, which uses voting for speckle (FDS) vs no-speckle (non-FDS) patch classification, can be used without user intervention.

- ✓ Method 1. **Develop a robust unsupervised speckle detection for simulation with available ground truth**
- ✓ Method 2. Develop an **optimization search and ensemble classification** scheme to **fully automatic detection** of speckle regions **for real data with no available ground truth.**

Method 1 is a semi-automatic classification. The aim is to introduce a fast robust unsupervised speckle detection algorithm (with available ground truth) in ultrasound images (simulated ultrasound images). In Method 1 the user needs to initialize values of the moment orders manually and choose the optimal orders through experience and visual assessment of the generated results. Thus, in this method, the results will depend upon the level of expertise of the user, as the user initially defines the values of v_R , v_S and v_K parameters. This could create some drop-outs (false positives) in some conditions. This is a fast method because it only requires one iteration.

To evaluate the performance of the speckle detection algorithm, Dice similarity between patch labels detected by this method and ground truth labels were used. Dice similarity is explained in section 7.1. Based on trial and error, base optimum global values for moment orders of R , S , K were subsequently investigated and found.

To avoid human errors in Method 1, another fully automatic method is proposed, in which the machine chooses the best values of v_R , v_S and v_K that lead to

better or best classification results, based on employing a voting system between the five classifiers employed in Method 1. The Fully Automatic Speckle Detection (FASD) algorithm in Method 2 searches for the best values that lead to best results based on optimization and exhaustive search.

Both methods are explained in the following sub-sections.

6.2 Feature Selection: Search for Best Statistical Features for Speckle Classification

As described previously in Chapter 3, Statistical Modeling of Speckle, in the feature selection and extraction method, each image is initially segmented into patches. Then the statistical features for each image patch are calculated using an optimization routine. These calculated statistics are based on arbitrary powers ν of the image patch A . In the proposed technique, for each image patch A , statistical features in equation (6.1) were computed. For further analysis of the values of ν_R , ν_S and ν_K , a new method was proposed which will be discussed in the results section, and the code is attached to Appendix A: PatchStat Toolbox.

This loads the data file of the B-mode image, extracts its properties and calls a function `get_features()`, which will extract the features and, before passing it for classification, visualizes the HeatMap of the features in each patch for user analysis.

6.3 Classification of extracted statistical features

In this study both hard (K-means and K-mediod) [87] and fuzzy partitioning (Fuzzy C-Means, Gustafson-Kessel and Gath-Geva) techniques were used [88, 89] for speckle detection in a competitive as well as ensemble manner. Each of five methods classifies the image patches to FDS and non-FDS with different aspects. So, by comparing the classification results of the five methods mentioned above, it is possible to decrease false positives and improve classification performance.

The images processed by the algorithm are the ones generated and described in Chapter 7, which talks about generating B-mode images.

Running the `speckle_detection.m` file will load the data file of the B-mode image, extract its properties and call a function `get_features()`, which will extract the features and pass to `classify()`. This function has separate calls to five unsupervised classification techniques, which return their results. The `make_grid()` and `plot_all_results()` functions will plot the B-mode image and results of five classification techniques (K-means, K-medoid, Fuzzy C-Means, Gustafson-Kessel, Gath-Geva fuzzy classifiers.)

6.3.1 “My Tool” GUI

To extract statistical features and classify patches to speckle (FDS) and non-speckle (non-FDS,) I have developed a toolbox which can detect speckles using both

Method 1 and Method 2 that will be described in next section. Both methods are embedded in this GUI.

Figure 6.1 shows a snapshot of My Toolbox for speckle detection with the cyst phantom example.

One must note that there is no relation between the colors of the classified patch results when that patch is FDS or non-FDS. These colors are assigned randomly with Matlab and the user must confirm it visually.

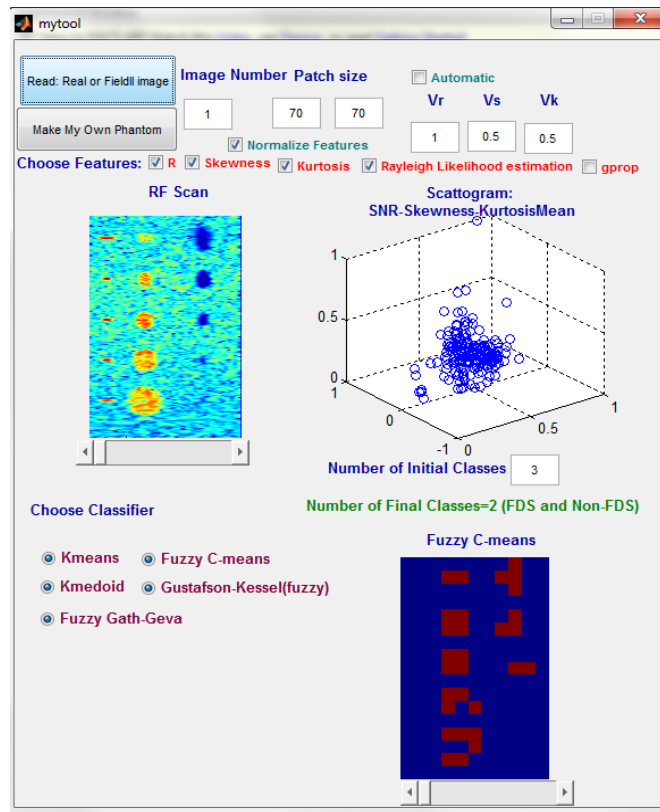


Figure 6.1: A snapshot of My Toolbox for speckle detection with Cyst phantom example. [red patches = patches with no cyst included]. Patch size is 70x70.

6.3.1.1 Description

Example: Classify cyst phantom to FDS and Non-FDS patches

Figure 6.2 shows a phantom which contains five point targets, 6, 5, 4, 3, 2 mm diameter water filled cysts, and 6, 5, 4, 3, 2 mm diameter high scattering regions. To analyze this phantom image, one just needs to load the image into the toolbox:

- By pressing (1) (**Read: Real or Field II image**) in Figure 6.2. B-Scans (ultrasound images) can be loaded in '.mat' or '.jpg' or any other image formats.
- By initially loading the image and setting the parameters with the condition that the checkbox in (10) (**Automatic**) is not checked (Automatic mode is off) Method 1 will run for analysis and classification. Checking this box will choose Method 2 as classification method.
- Pressing (2), calls another toolbox called 'UTool' to make phantom data, and it loads generated phantom images into 'MyTool'.
- When B-mode images are loaded, the first B-mode image is displayed in the plot labeled as (3). By scrolling to the right user can visualize other frames (B-mode images).
- User also can enter their desired frame number in the edit box in (4).

In (5), since the input image was segmented into patches for speckle detection, these two edit boxes are to specify horizontal and vertical patch sizes.

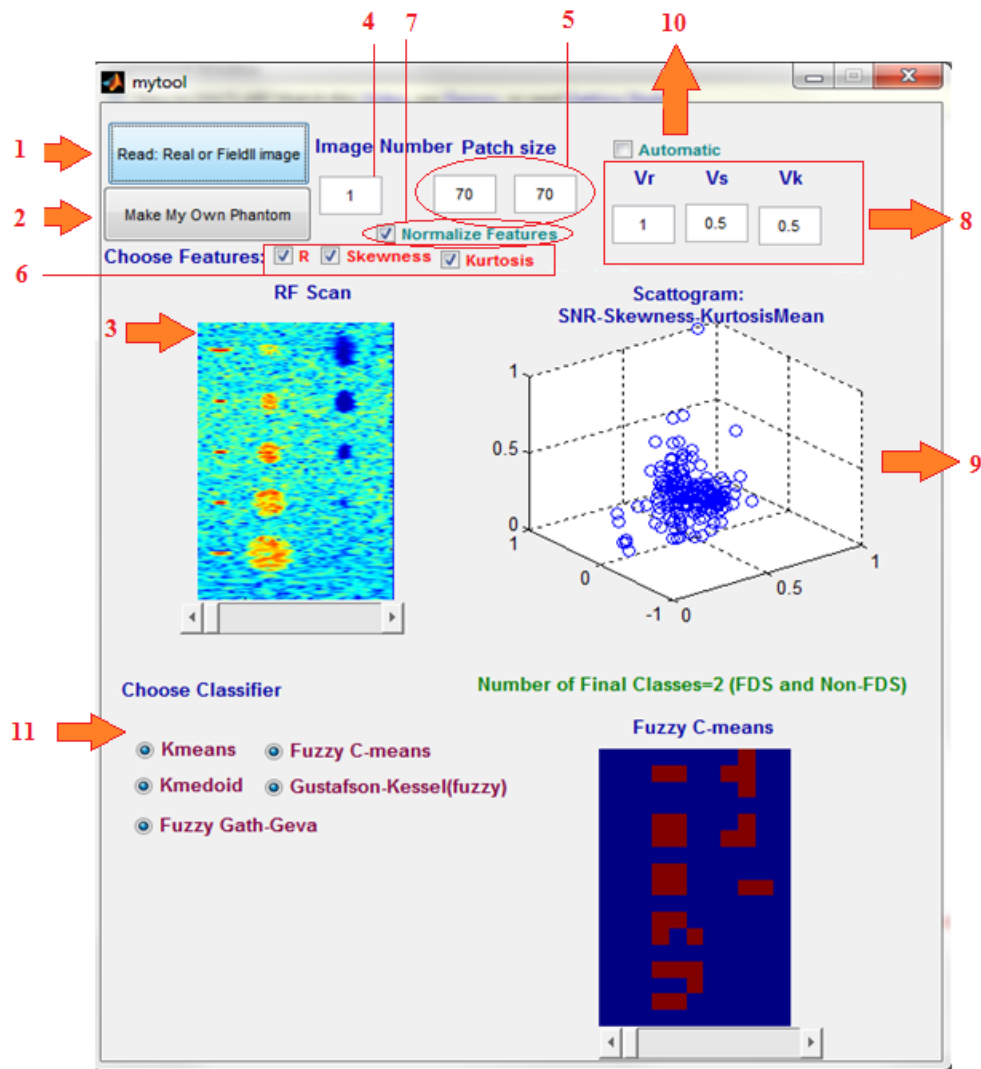


Figure 6.2: A snapshot of My Toolbox loaded with the cyst phantom and patch sizes 70x70.

- By checking the checkboxes in (6), the user can choose statistical features to be used, (R=SNR).
- Checking checkbox (7) will normalize statistical features for each patch.

- By changing the values of the boxes in (8), the user can set values of v_R , v_S and v_K (moment orders for R , S and K) in Method 1 and run experiments to visually assess the results. In Method 2, these boxes are deactivated.
- Graph (9) shows the statistical features R , S and K for each patch in a 3D plot. Each 3D point shows the values for statistical features of R , S , and K for one patch.
- If the user checks checkbox (10), Method 2 (FASD) will be applied and the moment orders in (8) will be disabled. FASD will change the moment orders in an exhaustive search manner. If (10) is disabled, Method 1 will be applied and Method 1 will use the moment orders in (8).
- By checking the clustering methods in (11), only chosen clustering methods will be applied.

6.4 Method 1: Semi-Automatic and Ensemble Speckle Detection

This method consists of a comparative analysis and fusion of five different unsupervised clustering techniques (Kmeans, K-medoid, Fuzzy C-Means, Fuzzy Gath-Geva, and Gustafson-Kessel clustering) with different combinations of statistical features used as inputs to the classifiers. Fusion means that after each

algorithm completes the classification, the results of all five classifiers are used to choose the best ones.

6.4.1 The Overall Approach

The proposed semi-supervised speckle detection Method 1 (the semi-supervised and ensemble speckle detection method) is shown in Figure 6.3, which consists of three major steps:

- Step 1: Segmentation of the input image into equal size image patches.
- Step 2: Extraction of statistical features for each patch.
- Step 3: Use of one of several unsupervised clustering techniques explained in chapter 3, section 3.2, to classify each image patch as FDS or non-FDS.

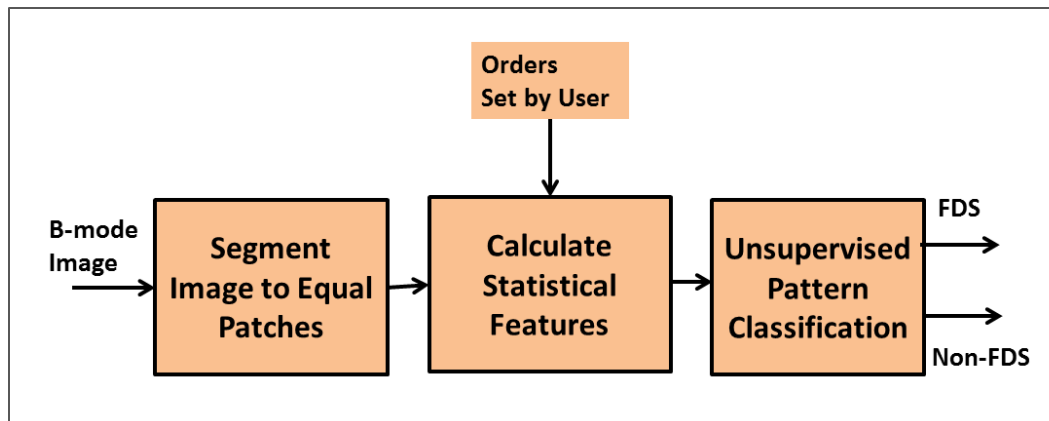


Figure 6.3: Static Classification of image patches to FDS and non-FDS. Comparative analysis of five different unsupervised clustering techniques with different combinations of statistical features used as input to the classifiers.

6.5 Method 2: Automated method (Fully Automatic Speckle Detection (FASD) Algorithm)

As discussed in the Introduction section, an optimization search and ensemble classification scheme was proposed for a fully automatic detection of speckle regions. This method aims to avoid human errors that might be present in Method 1. A novel method called the Fully Automatic Speckle Detection (FASD) Algorithm was developed, which searches for the best values of v_R , v_S and v_K that lead to the best classification results (decided by voting,) based on optimization and an exhaustive search. The main goal is to introduce a fast algorithm for speckle detection in ultrasound images. As mentioned, Method 2 calculates the best results based on an optimization search and voting. Fusion means that after each algorithm completes the classification, using voting, the results of all five classifiers are used to choose the best ones.

6.5.1 Overall Approach

The proposed FASD method (Method 2) is shown in Figure 6.4, which consists of four major steps:

- Step 1: Segmentation of the input image into equal patches.
- Step 2: Extraction of specific statistical features for each patch. The same statistical features as in Method 1 were used.

- Step 3: After extracting features for each image patch A, use the classification scheme shown in Figure 6.4. This figure illustrates the proposed exhaustive search by varying moment orders for R , S and K features from 0 to L to determine the occurrence of two classes, FDS and non-FDS. If a class happens more than 50 percent of the total iterations, the corresponding patch is classified to that class. Suppose $L=2$ and step size is 0.1, therefore there will be $20^3=8000$ iterations.
- Step 4: Use the proposed ensemble classification schema shown in Figure 6.5 to combine classification results and get a final single classification result. Suppose $L=2$ and step size is 0.1, then number of iterations (N in Figure 6.5) will be $20^3=8000$. With 8000 iterations there will be $N=8000$ classification results and by use of schema at Figure 6.5, we can do majority voting to combine classification results and get final results. As an example suppose for a typical patch A, out of 8000, 5620 times classifier says it is FDS. $5620/8000=70\%$ which is bigger than $50\%+1$ (threshold for majority vote), therefore patch A has a final classification result as FDS.

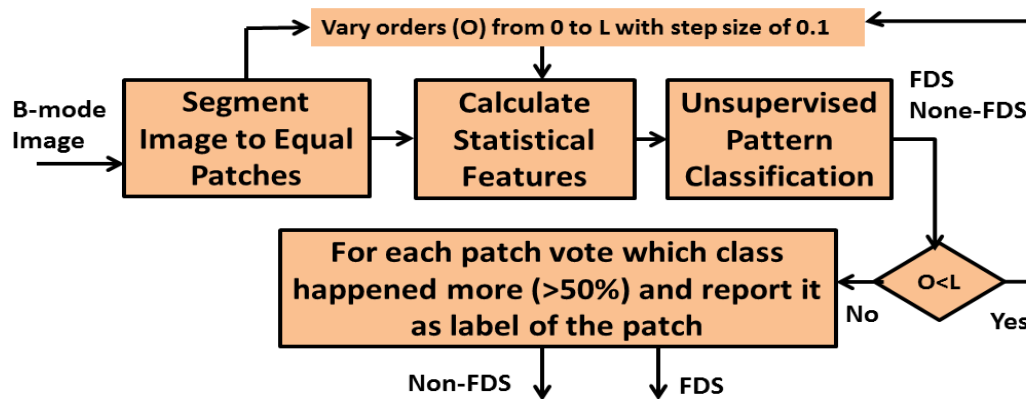


Figure 6.4: Adaptive Speckle Classification block diagram.

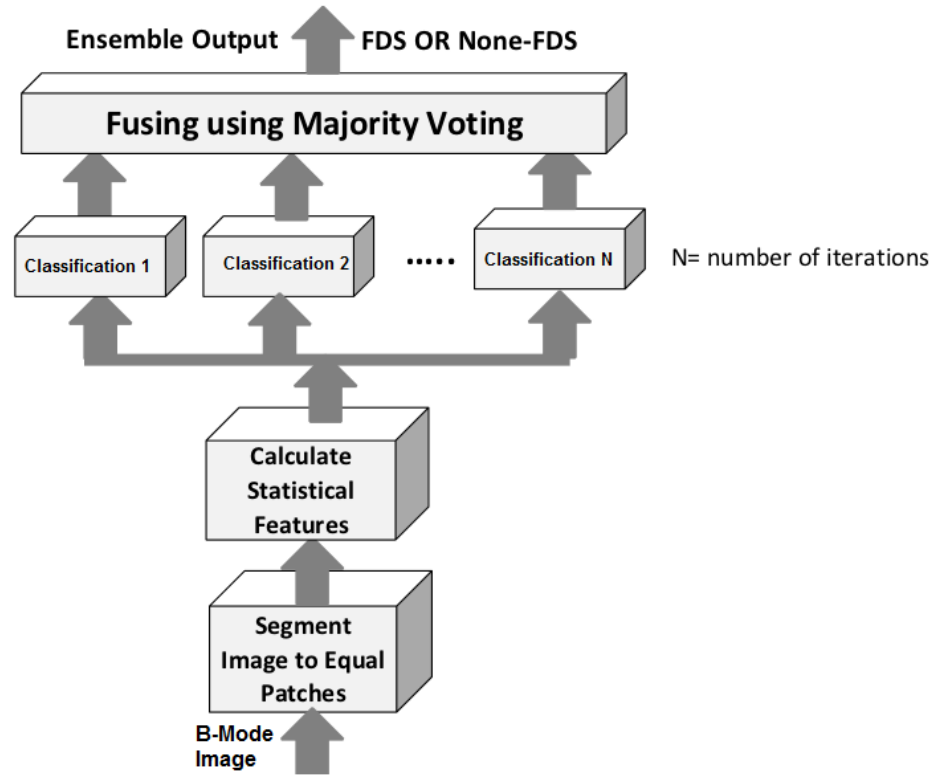


Figure 6.5: Ensemble classification scheme to combine the speckle classification results using majority voting.

It should be mentioned that all the steps above should be repeated for each of the proposed classification methods. For example in case of Fuzzy C-Means, one needs to follow all steps above. Then, by comparing the classification results of the five methods, it is possible to decrease false positives and improve the classification performance. As shown in Figure 6.5, a popular majority voting system can be used for this combination.

6.5.2 Optimization Search (Exhaustive Search)

In real data there is a lack of ground truth and the manual labeling of speckle patches is time-consuming and subjective. Therefore, we propose Method 2, which is based on an exhaustive search, for best class identification by varying moment orders. Method 1, the proposed semi-supervised speckle detection based on the ensemble scheme shown in Figure 6.5, is very sensitive to orders of statistical features. Method 1 is based on trial and error, and performance is sensitive to moment orders. In Method 2 we automated the process by varying moment orders to find the optimum class for each patch. Optimum class occurrence is more than 50 percent.

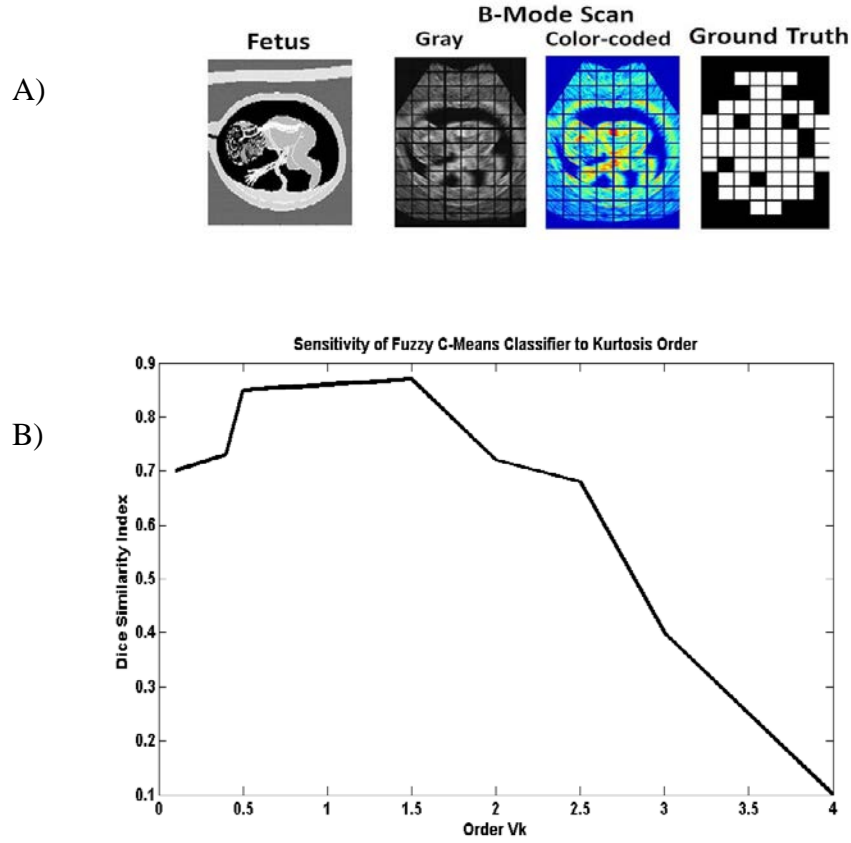


Figure 6.6: Sensitivity analysis for non-adaptive method (Method 1): A) input image patches; B) Sensitivity of Fuzzy C-Means classifier to orders of statistical features. v_K varies from 0 to 4, while other orders, VR and VS set to 1.

Figure 6.6 shows the sensitivity of a Fuzzy C-Means classifier to a variation of kurtosis order. So, feature selection will need to empirically and manually tune the statistical orders to achieve the best classification results. However, searching for best features and best orders manually could be frustrating and time-consuming. Therefore Method 2 proposes to use an optimization search to look for the best statistical features and best orders for them.

Figure 6.7 shows the diagrammatic steps for such an optimization search for fully automatic speckle classification for real data with no available ground truth.

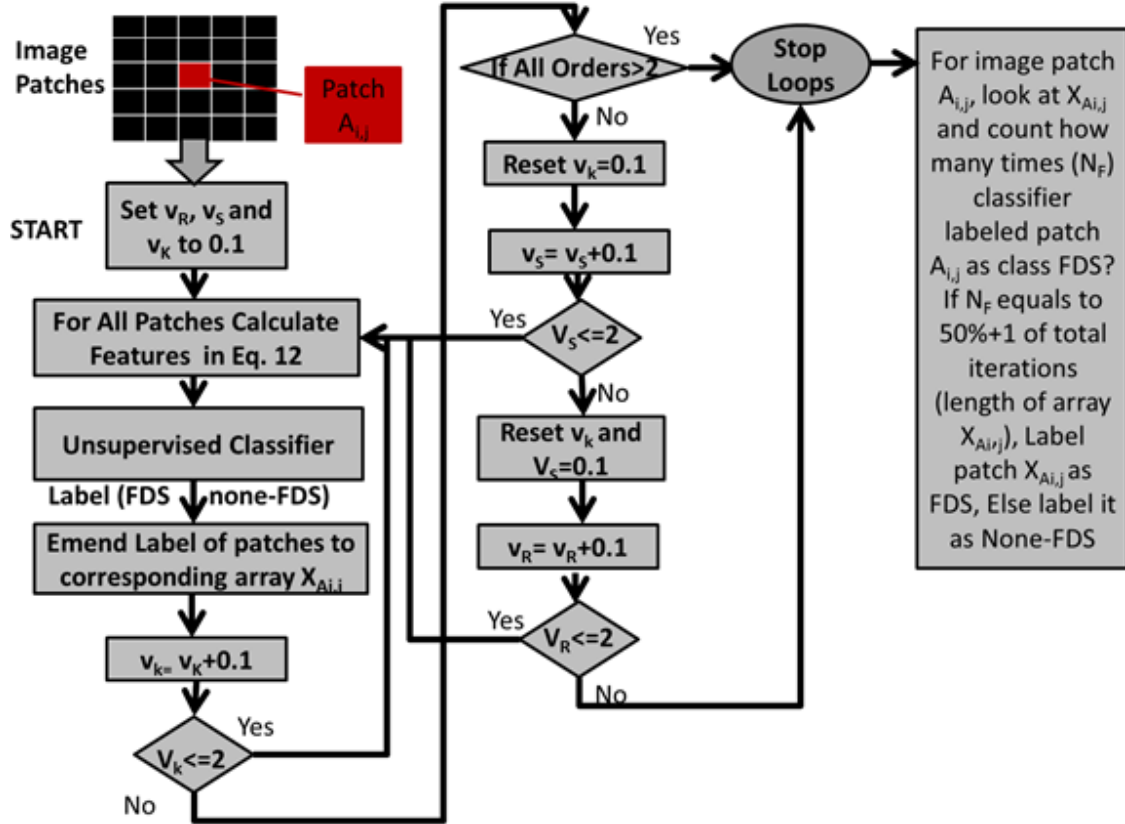


Figure 6.7: The diagrammatic steps for an optimization search for fully automatic speckle classification for real data with no available ground truth.

Algorithm and steps for the optimization search:

- 1- Segment Input B-mode Image to $L_1 \times L_2$ image patches. Patches have the same size of $M \times N$.
- 2- Set orders of Features R, Skewness (S) and Kurtosis (K), respectively v_R, v_S and v_K in Eq. 12 to 0.1.

- 3- Calculate Feature R for all image patches.
- 4- Calculate Feature S for all image patches.
- 5- Calculate Feature K for all image patches.
- 6- Feed Classifier with statistical features of image patches obtained in steps 1-3.
- 7- Classify image patches to FDS and Non-FDS classes and store class label for each image patch in a separate array $X_{Ai,j}$, where i and j refer to the location of image patch $A_{i,j}$.
- 8- Increment v_k by step of 0.1. $v_k = v_k + 0.1$;
- 9- If $v_k \leq 2$, go to step 5 to re-run steps 3 - 7 with updated features for all patches. Else if $v_k > 2$ and v_s or $v_r < 2$, reset v_k to 0.1 and go to step 10. If all v_k, v_s and v_r are ≥ 2 , stop optimization search and go to step 14.
- 10- Increment v_s by step of 0.1. $v_s = v_s + 0.1$;
- 11- If $v_s \leq 2$, go to step 5 to re-run steps 3 - 7 with updated features for all patches. Else if $v_s > 2$ and v_k or $v_r < 2$, reset v_s to 0.1 and go to step 10. If all v_k, v_s and v_r are ≥ 2 , stop optimization search and go to step 14.
- 12- Increment v_r by step of 0.1. $v_r = v_r + 0.1$;
- 13- If $v_r \leq 2$, go to step 5 to re-run steps 3 - 7 with updated features for all patches. Else if $v_r > 2$ and v_s or $v_k < 2$, reset v_r to 0.1 and go to step 10. If all v_k, v_s and v_r are ≥ 2 , stop optimization search and go to step 14.
- 14- For image patch $A_{i,j}$, look at $X_{Ai,j}$ and count how many times (N_F) classifier labeled patch $A_{i,j}$ as class FDS. If N_F equals 50%+1 of total iterations (length of array $X_{Ai,j}$), label patch $X_{Ai,j}$ as FDS, or else label it as Non-FDS.
- 15- Repeat steps 1 - 14 for all five classifiers.
- 16- Fuse the results of all classifiers to get an ensemble output result for each image patch.

Chapter 7

Results and Performance

7.1 Phantom results

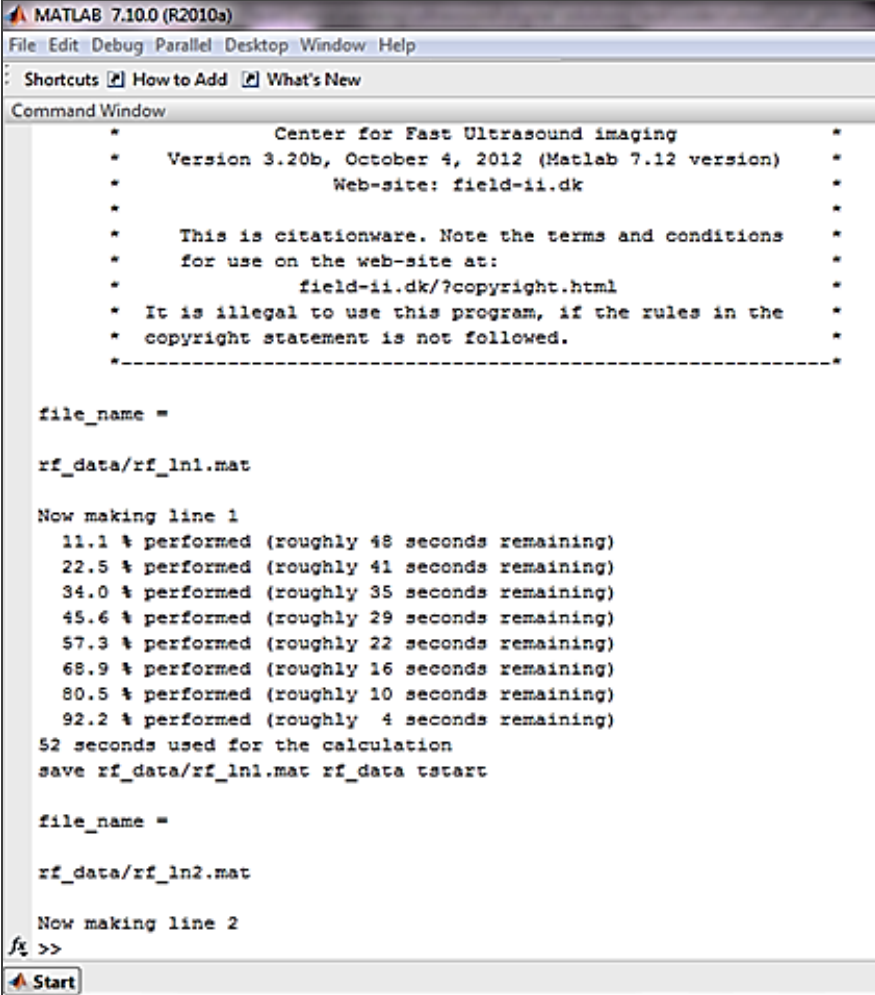
As discussed in previous chapters, a basic simulation software and GUI have been developed for B-mode simulated image generation based on utility functions that are freely available through the Field II simulator application interface (API). The newly developed simulation software has been used to generate the desired B-mode ultrasound images for the investigations. These simulated objects/organs or the so-called phantoms include cyst, kidney, fetus, prostate, and heart.

Imaging of these synthetic organs is done by summing the field from a collection of point scatterers. Scattering strength depends on density and speed of

sound in the target tissue, and has a Gaussian distribution. Simulation happens line by line for each image. The number of scatterers in each phantom are set to be around 100,000 or more, so simulating 50 to 128 RF lines (e.g. 50 for cyst and 128 for kidney) can take several days depending on the computer used. If clusters are used, however, the image can be generated faster, in around ten minutes. Simulating the lines takes roughly 40 to 45 minutes (e.g. for 50 lines) for 100,000 scatterers, and one hour and ten minutes for one million scatterers on a 3.40 GHz CPU PC. The process of line-by-line phantom image generation is shown in Figure 7.1, which is a snapshot of the cyst phantom generation process done in Matlab using the UTool GUI Software developed for the purpose of this research.

To evaluate the proposed speckle classification scheme and calculate the performance of each unsupervised classification technique, as the ground truth, B-mode images were simulated and used by ultrasound simulation programs with 100,000 scatterers. After generation of the images, each image was resized and segmented to image patches, with the desired image patch size (e.g. 100x100 pixels). Then the statistical features for the image patch were calculated (R, Skewness and Kurtosis). After calculating statistical features for each image patch, the resulting three-dimensional features for each image patch were used to classify each patch as FDS or non-FDS by using the unsupervised clustering techniques discussed in previous chapters. For this purpose, five pattern classification techniques were employed: K-means, K-medoid, Fuzzy C-Means, Gustafson-Kessel fuzzy classifier

and Gath-Geva fuzzy classifiers. The performance of the classification of the two proposed methods for speckle detection was then evaluated.



```

MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Shortcuts How to Add What's New
Command Window

*           Center for Fast Ultrasound imaging           *
*   Version 3.20b, October 4, 2012 (Matlab 7.12 version)   *
*           Web-site: field-ii.dk                       *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*   This is citationware. Note the terms and conditions   *
*   for use on the web-site at:                           *
*           field-ii.dk/?copyright.html                   *
*   It is illegal to use this program, if the rules in the *
*   copyright statement is not followed.                  *
*-----*

file_name =

rf_data/rf_ln1.mat

Now making line 1
11.1 % performed (roughly 48 seconds remaining)
22.5 % performed (roughly 41 seconds remaining)
34.0 % performed (roughly 35 seconds remaining)
45.6 % performed (roughly 29 seconds remaining)
57.3 % performed (roughly 22 seconds remaining)
68.9 % performed (roughly 16 seconds remaining)
80.5 % performed (roughly 10 seconds remaining)
92.2 % performed (roughly 4 seconds remaining)
52 seconds used for the calculation
save rf_data/rf_ln1.mat rf_data tstart

file_name =

rf_data/rf_ln2.mat

Now making line 2
fx >>
Start

```

Figure 7.1: Snapshot of the cyst phantom generation process in Matlab using the UTool GUI Software.

The corresponding B-Mode image is divided into a number of ROIs to image differences in the underlying material properties. Furthermore, statistical analysis can be used to quantify the variability of the estimates throughout the image.

For a statistical analysis of the results, and in order to validate the proposed estimation algorithm presented in Chapter 6, Methodology (Speckle detection), the algorithm was tested in a variety of ways that were increasingly representative of the ultimate goal of the study: to discriminate speckle (FDS) patches from non-speckle (non-FDS) ones. The applied testing methods included using artificially-generated samples of known computer simulations, physical phantoms, biological phantoms, and real data.

To evaluate the performance of the proposed method for speckle detection, the Dice similarity index (DS) was used, which is always between 0 and 1. The Dice similarity metric tests for overlap and similarity between results of class labels for image patches and compares them with the ground truth in the case of simulation data.

Where the DS measure is defined as:

$$DS = 2 \frac{A \cap B}{A + B} = 2 \frac{TP}{2TP + FP + FN}$$

where A = image patch labels for ground truth, B = image patch labels obtained by a classifier, TP = true positive, FP = false positive and FN = false negative. Figure 7.2, below, shows the Dice similarity concepts. DS = 0 means there is no similarity, and

$DS = 1$ means two regions or images are identical and there is no difference between their shape and location.

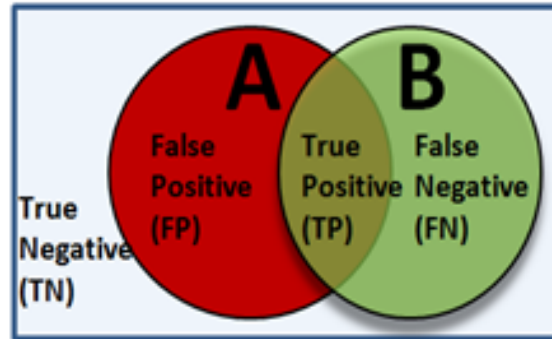


Figure 7.2: Comparing overlap and similarity of two regions using the Dice similarity (DS) index. A = image patch labels for ground truth, B = image patch labels obtained by a classifier. DS has always a value between 0 and 1. If A and B are completely different, that means there is no overlap, the Dice similarity is zero, and true positive is zero. If A and B are identical, that means there is full overlap and the Dice similarity is 1.

7.1.1 Calculation of B-mode image of Cyst Phantom

The cyst phantom consists of a collection of point targets, five high intensity (bright) cysts, and five low intensity (dark) cysts. The point targets have a fixed amplitude of 100, compared to the standard deviation of the Gaussian distributions of 1. The Matlab code (m-files) for generating the cyst phantom can be found in Appendix B. Before running the code (within `simulate_image.m`), a phantom containing cysts must be defined. The user can define the desired object sizes in the code; however the default values (in the `cyst_pht` file) are set to be 6, 5, 4, 3, and 2 mm.

Once the phantom parameters have been defined, the actual phantom can be created using the Field II code. The other parameter settings are as following:

- The number of elements for a linear scan was set to a 192 element transducer.
- 64 active elements with a Hanning apodization in transmit and receive were used.
- The element height was set to 5 mm.
- The width was the wavelength.
- The kerf was 0.05 mm
- The focus was placed at 60 mm.
- Receive focusing was done at 20 mm intervals, 30 mm from the transducer surface.
- The default number of scatterers was set to 100,000, but there were investigations to generate phantoms with a higher and lower number of scatterers, in order to study the effect on both phantom generation and the classification results.

In the generated phantom image, a homogeneous speckle pattern can be seen along with all the features of the phantom. The resulting image is shown in Figure 7.3. The white and black objects in the cyst phantom are both representations of two different object types (e.g. tissue or bone.)

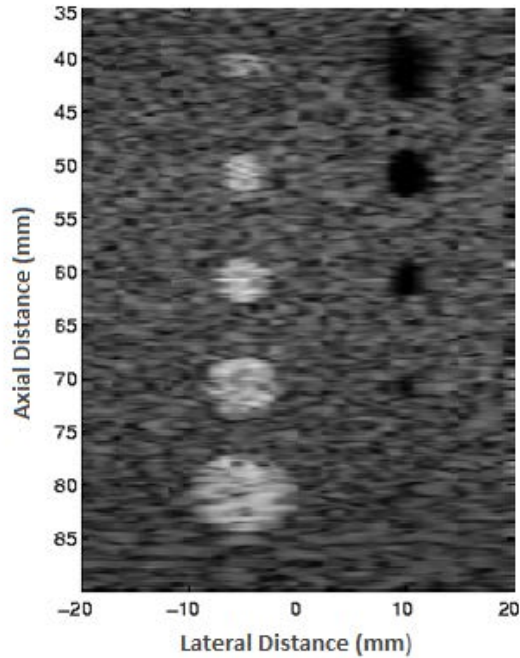


Figure 7.3: The cyst phantom generated using GUI as the simulation software: the phantom contains five point targets, 6, 5, 4, 3, and 2 mm diameter water filled cysts, and 6, 5, 4, 3, and 2 mm diameter high scattering regions.

The cyst phantom is regenerated using the same parameters and settings that Dr. Joergen Jensen, the writer for the Field II acoustic pressure field simulation package, uses, and is shown in Figure 7.4. Thus, the regenerated cyst phantom was used as ground truth, since the user who generated the desired phantom knows the exact location of each object, the amplitudes, etc. and based on visual assessment can judge the performance of the methods.

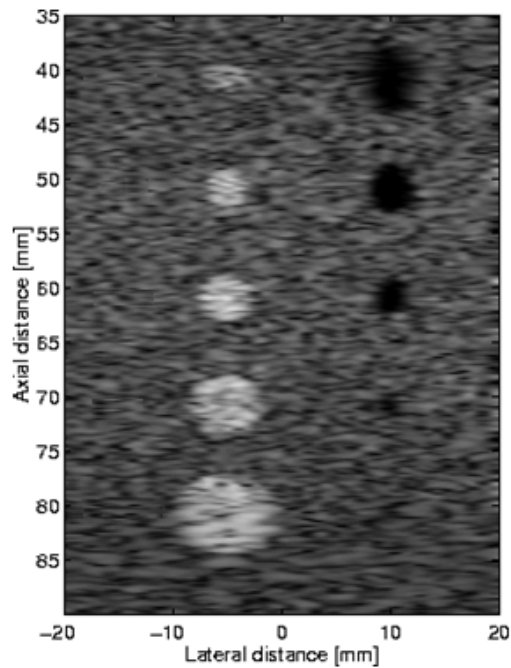


Figure 7.4: Cyst phantom simulated by the Field II program, photo courtesy of Duke University.

Example: Create a computer model of a cyst phantom.

Before running the actual Field II code (within `simulate_image.m`), a phantom containing cysts must be defined. The size of the phantom, size of the cysts, number of cysts, and shape and location of the cysts are some of the parameters that are controlled by the user.

Once the simulated ultrasound measurements have been created, the phantom image can be created using `form_image.m`. This routine finds the envelope of the complex RF data, converts the data to log-magnitude representation, and then outputs the final image with a dynamic range of 60 dB. The phantoms generated in this

section were all the same size (40 mm in the lateral direction and 55 mm in the axial direction.) Now using the GUI introduced in Chapter 5, on Data Collection, one can use the function to form the image from RF data by pushing the form image button.

Figure 7.5 is a screenshot of the GUI used for the phantom generation for this example, which contains five point targets and 6, 5, 4, 3, and 2 mm diameter water-filled cysts, and 6, 5, 4, 3, and 2 mm diameter high scattering regions. All scatterers are situated in a box of $(x,y,z) = (50,10,60)$ mm, and the box starts 30 mm from the transducer surface. To generate this phantom data [script: cyst_pht.m] with organ-specific scripts and mathematical models, the following steps were applied:

- Run the simulation to make RF data [script: simulateImage.m]. Parameters to set prior to invoking the simulation:
 - Transducer center frequency [Hz]
 - Sampling frequency [Hz]
 - Speed of sound [m/s]
 - Wavelength [m]
 - Width of element
 - Height of element [m]
 - Kerf [m]
 - Fixed focal point [m]
 - Number of physical elements
 - Number of elements
 - Number of active elements
- Image reconstruction from RF data [script: formImage.m]. Set image height and width before run.

These steps are graphically visualized in Figure 7.5.

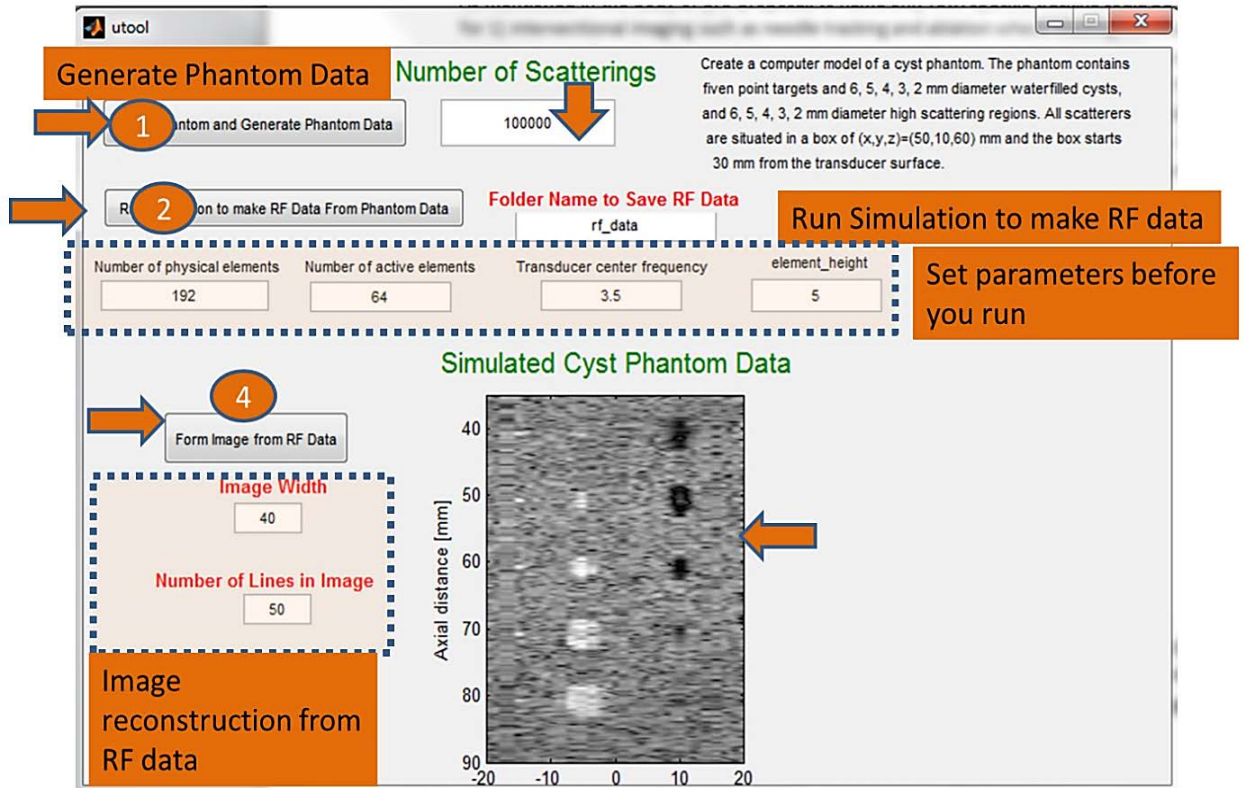


Figure 7.5: Developed UTool Software to generate own phantoms. Snapshot of the ultrasound simulation GUI with steps and examples for the values of the parameters for generation of phantom data using UTool: phantom contains five point targets, 6, 5, 4, 3, and 2 mm diameter water-filled cysts, and 6, 5, 4, 3, and 2 mm diameter high scattering regions.

All generated phantoms were used as input to the speckle detection algorithms embedded in the developed GUI.

7.1.1.1 Cyst Phantom Investigation results: Size, Shape, Number and Location of the Cysts/Objects

In this step, different types of cyst phantoms were generated to serve as subsequent input into algorithms for result generation and analysis. Investigations included the generation of phantoms with changes in the location, size, shape and other properties of the objects, and based on this known data (such as the exact location of each cyst phantom in the simulation,) generation of ground truth.

As one of the interests of this study, the generation of square-shaped cyst phantoms were investigated with the aim of a more controlled investigation into the patch size and effect of the image patch shape (square or rectangular) on the classification results. Alternatively, one can visually or automatically analyze the performance of the algorithms. Figure 7.6 represents a segmented cyst phantom. Figure 7.7 shows a snapshot of the generation of the phantom with square cyst/objects using the ultrasound simulation GUI.

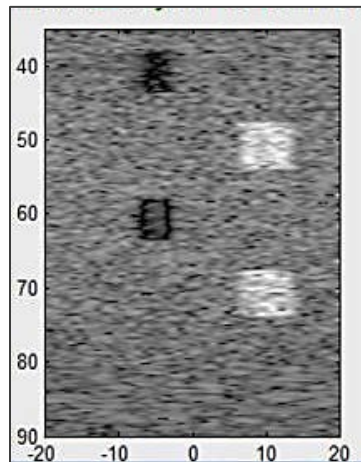


Figure 7.6: Phantom containing 4 square objects.

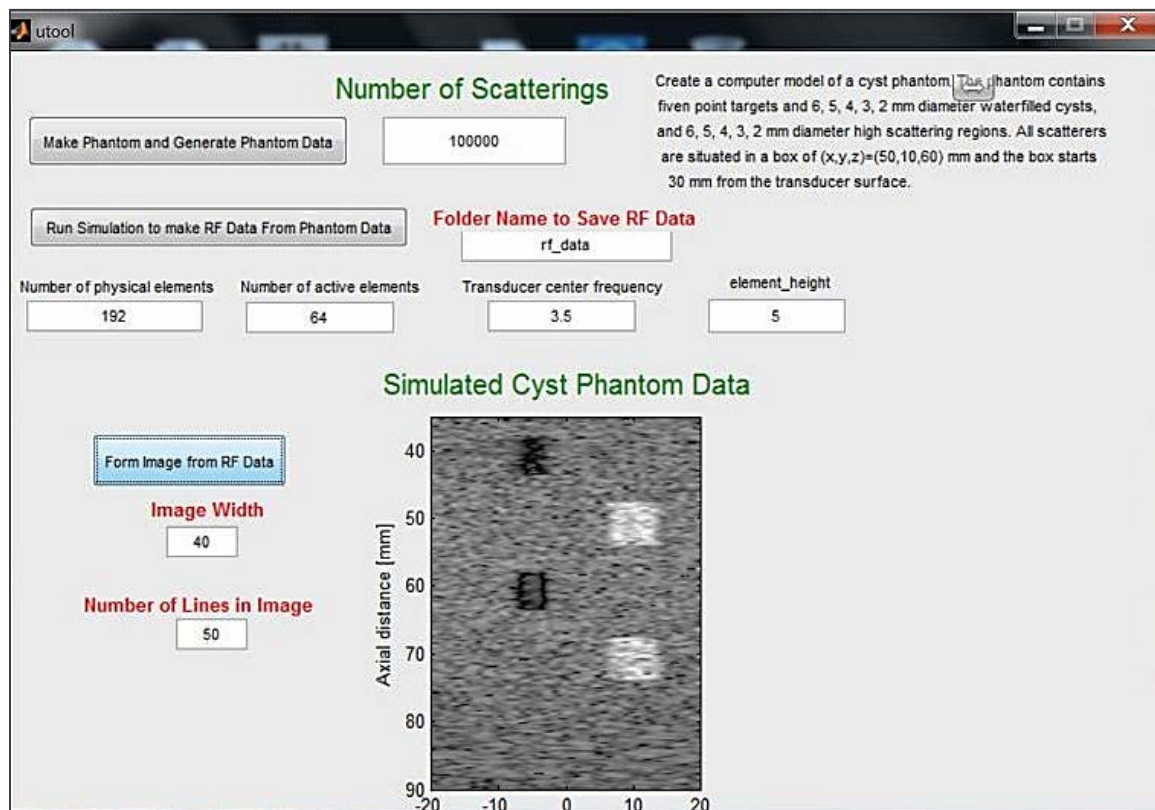


Figure 7.7: Square cyst phantom generation using the UTool GUI.

Attached are samples of the generation of other cyst phantoms.

Figure 7.8 shows a phantom with the 6 mm square cysts. The image was segmented into patch sizes of 100 x 100 pixels.

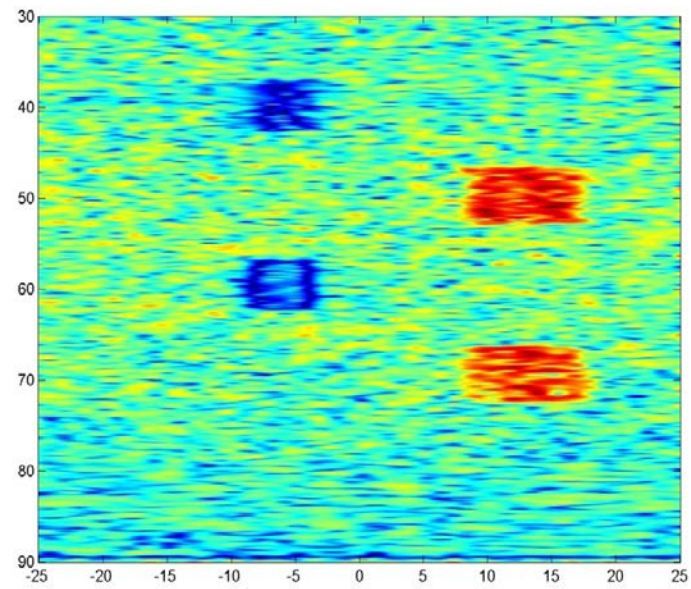


Figure 7.8: Phantom with 6 mm square cysts/objects.

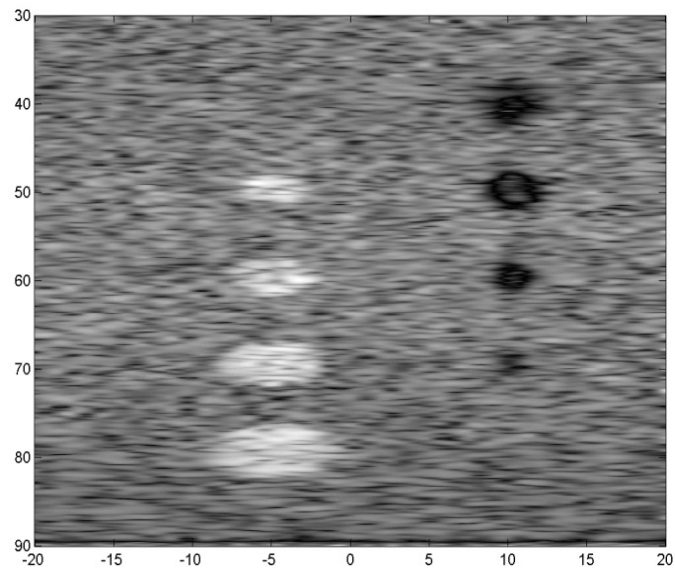


Figure 7.9: Field II cyst phantom without the 2 mm cyst.

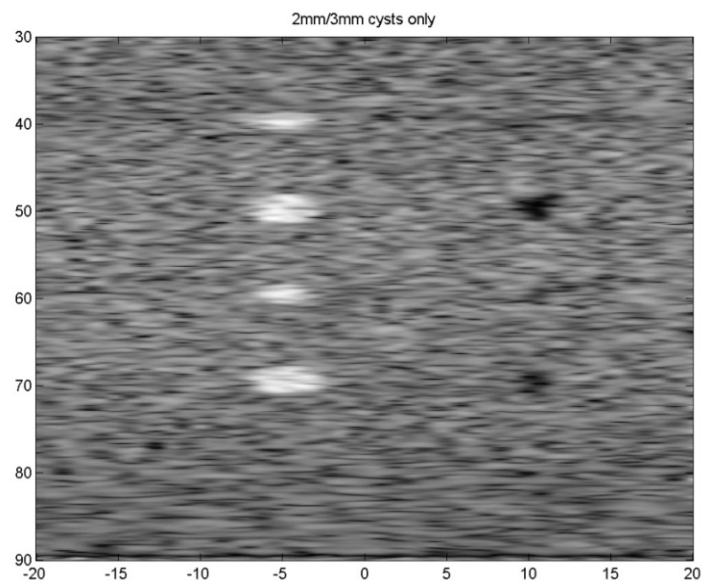


Figure 7.10: Field II cyst phantom generated with two 2 mm and four 3 mm cysts.

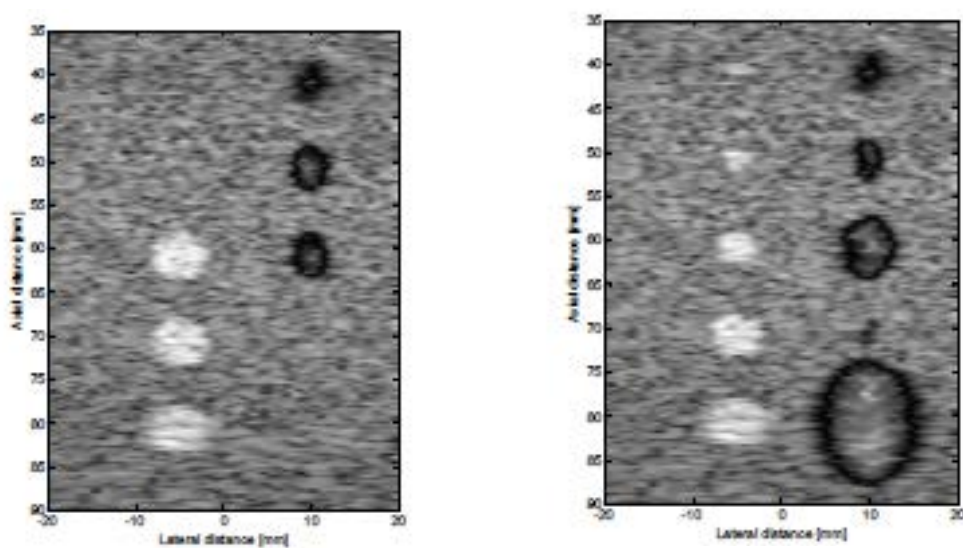


Figure 7.11: Figure A (left) and Figure B (right), two other Field II cyst phantom generated with different set of sizes and different number of cysts for investigations.

7.1.1.2 Feature Selection: Search for Best Statistical Features for Speckle Classification and “PatchStat Toolbox”

This study investigated the application of different features employed in other research groups, and also employed new features to improve the classification results of previous works. Investigated features included Entropy, Homogeneity, Energy, Average Intensity, SNR, Skewness, Kurtosis and others. The following section presents some of the results of the investigations using the cyst phantom for simplicity in visual assessment, but the same investigations were done using the other phantoms in the study. Figure 7.12 shows the cyst phantom using a color scale and HeatMap for better visualization and differentiation between the cyst (non-speckle) and the speckle area. Patch sizes of 50 x 50 were used in the following images to calculate the features.

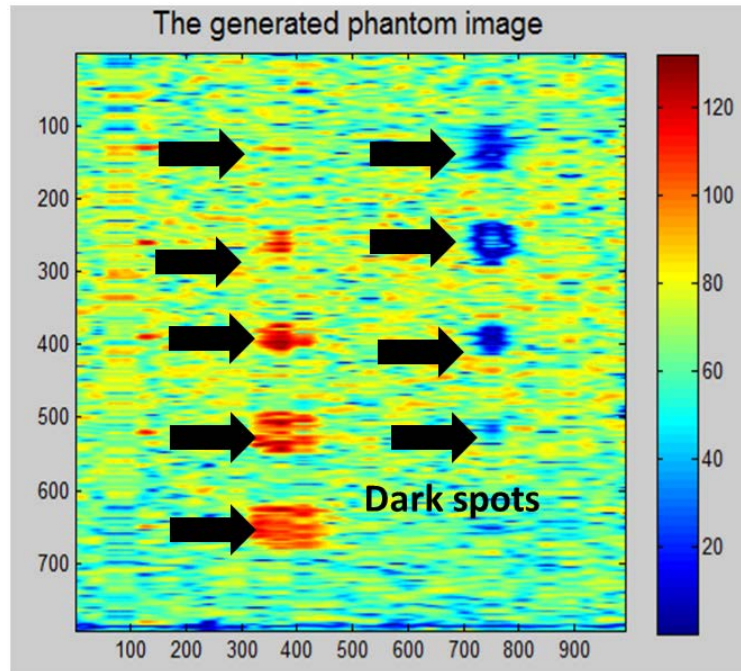


Figure 7.12: Cyst phantom with color scale visualization.

Figure 7.13 shows the probability density function for speckle and non-speckle regions in the reconstructed phantom image. The number of scatterings for speckle regions was 100,000.

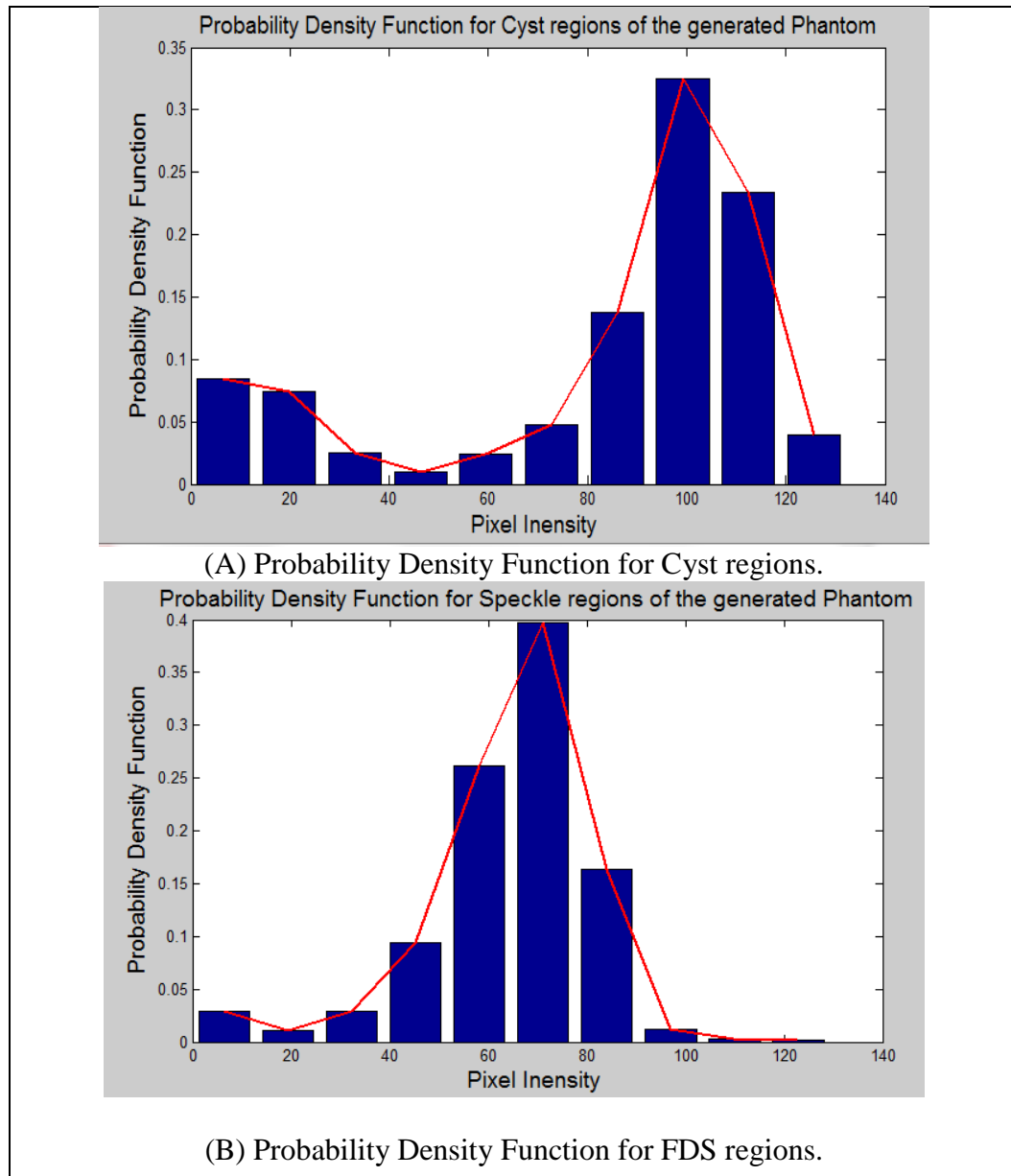


Figure 7.13: Probability density function for cyst regions.

As can be seen based on pixel intensity, one cannot separate speckle and non-speckle regions, so statistical model-based clustering is needed to detect speckle

regions, and entropy, energy, homogeneity, average intensity, etc were not found to be suitable classification features for speckle.

To investigate which statistical features are useful for speckle detection, a comparison of different statistical features, including SNR (R), Skewness, Kurtosis, Rayleigh distribution likelihood, Energy, Entropy and Homogeneity was proposed and studied. Below is the definition of each of these features:

- 1- SNR is defined as mean over standard deviation: $\text{mean}(X)/\text{std}(X)$, where:

$X = (\text{pixel intensities for each patch})^{\nu_r}$, and ν_r is moment order for SNR .

- 2- Skewness is a measure of the asymmetry of a distribution. If the left tail of the distribution is more populated than the right tail, the distribution has negative skewness. If the reverse is true, it has positive skewness. If the two are equal, it has zero skewness. This is shown in Figure 7.14.

$$\text{Skewness} = E\left[\left(\frac{X - \mu}{\sigma}\right)^3\right] = \frac{\mu^3}{\sigma^3} = \frac{E[(X - \mu)^3]}{(E[(X - \mu)^2])^{3/2}}$$

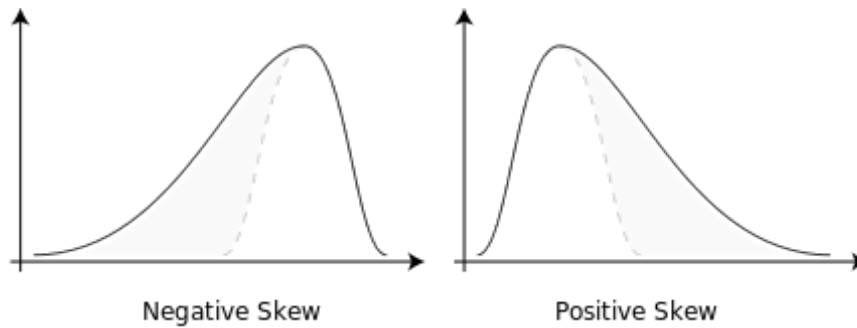


Figure 7.14: Skewness probability density function.

- 3- Kurtosis is the degree of peakedness of a distribution, defined as a normalized form of the fourth central moment of a distribution.

$$\text{Kurtosis} = \frac{E[(X - \mu)^4]}{(E[(X - \mu)^2])^2} = \frac{\mu^4}{\sigma^4}$$

and $X = (\text{pixel intensities for each patch})^{v_s}$, where v_s is moment order for skewness.

- 4- Correlation returns a measure of how correlated a pixel (p) is to its neighbor over the whole image.

$$\sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j}$$

where p is pixel intensity for every patch pixel.

- 5- Energy returns the sum of squared elements (p =pixels) in the image.

$$\sum_{i,j} p(i, j)^2$$

- 6- Homogeneity: Returns a value that measures the closeness of the distribution of elements (p =pixels) in the image.

$$\sum_{i,j} \frac{p(i, j)}{1 + |i - j|}$$

- 7- Average Intensity: Returns a mean value of pixel intensities within an image patch.

$$\frac{\sum_{i,j} p(i, j)}{\text{number of pixels}}$$

By calculating the different statistical features mentioned above and applying different combination of them patch separability can be investigated and optimum results can be reported. Investigation results showed that the optimum patch separability was achieved by choosing R, S, K combination for the statistical features. Figure 7.15 presents the classification results using the five proposed classifiers in this research work: K-means, K-mediod, Fuzzy C-Means, Fuzzy GK, and Fuzzy GG. The results are compared to the generated ground-truth for performance analysis. Results showed that in this case, K-means, Fuzzy C-Means and K-mediod performed the same (Performance=85.42%.) In Fuzzy GK, the number of miss-classifications were 16 out of 96 (Performance=83.34%) and in Fuzzy GG, the number of miss-classifications were 10 out of 96 (Performance=89.6%).

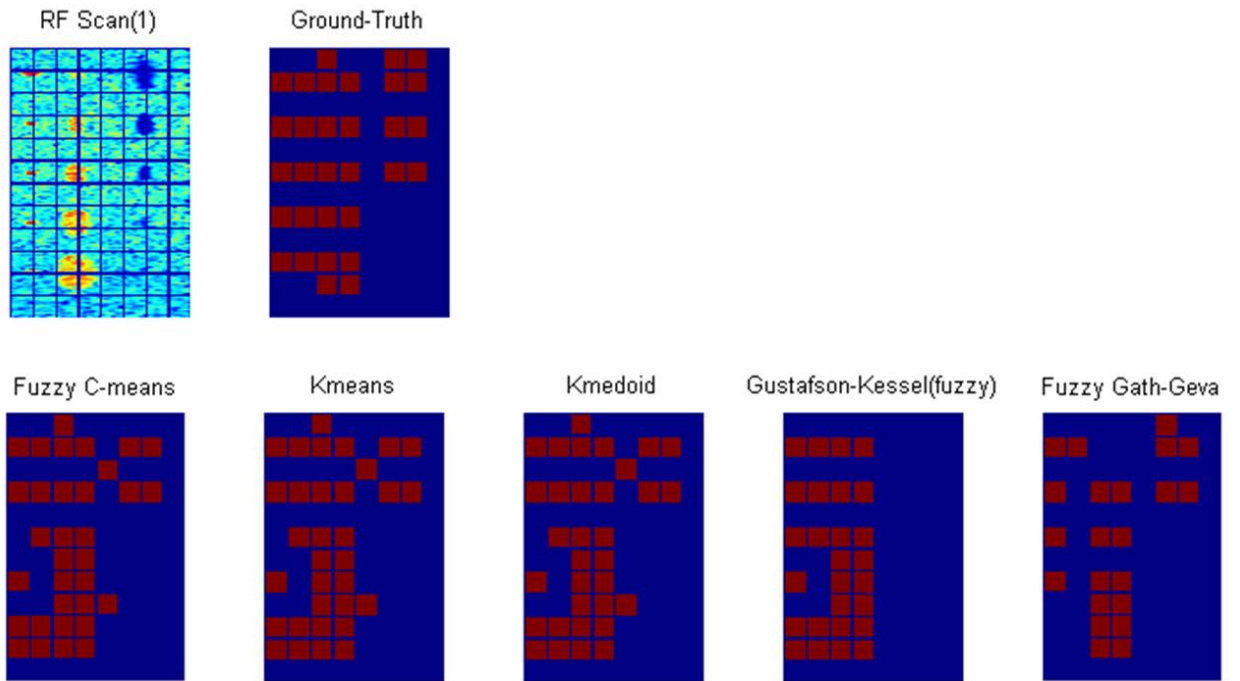


Figure 7.15: Classification results for the proposed five classifiers: Fuzzy C-Means, K-means, K-mediod, Fuzzy GK and Fuzzy GG, using the generated ground-truth.

Entropy, Homogeneity, Energy, and Average Intensity for each patch were also investigated and their effect on the classification is visualized in the figures below.

Entropy is a measure of uncertainty, and thus a statistical measure of randomness that can be used to characterize the texture of the input image. Figure 7.16 represents the classification results where the entropy of the generated cyst phantom is used as a feature in speckle classification studies and as shown this feature was not able to detect all cyst regions. In this image entropy was the only feature used for classification.

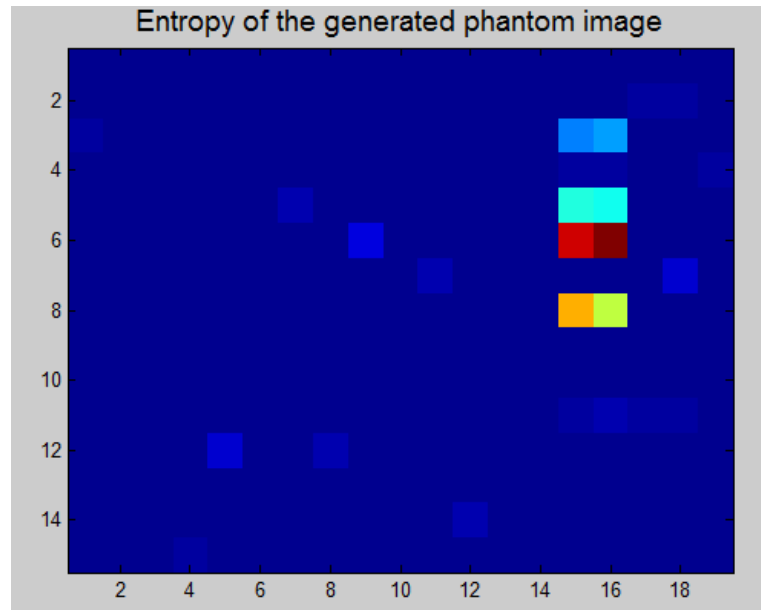


Figure 7.16: Entropy of the cyst phantom (as the only feature used for classification).

Other features such as Homogeneity, Energy, and Average Intensity for each patch were calculated and used in the investigation. The results showed that these features were not detecting all Cyst regions as represented in Figure 7.17 and 7.18, but Average Intensity in Figure 7.19 showed better detection results, however still not comparable to the classification results where SNR, Skewness and Kurtosis features were chosen for classification.

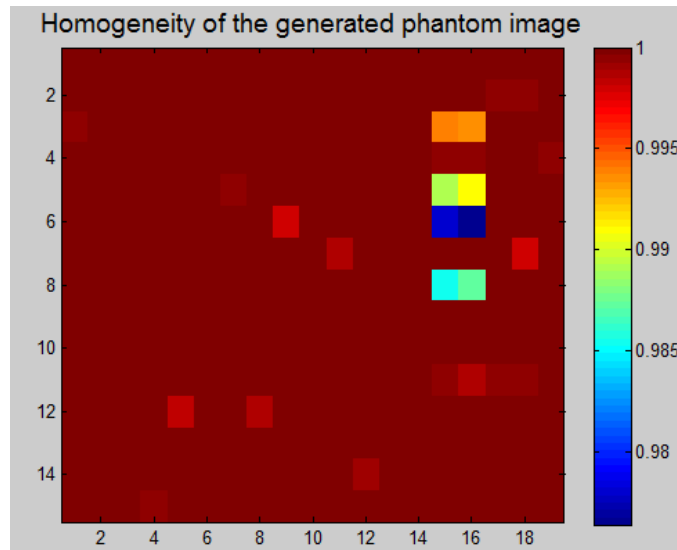


Figure 7.17: Heat map of Homogeneity as a feature.

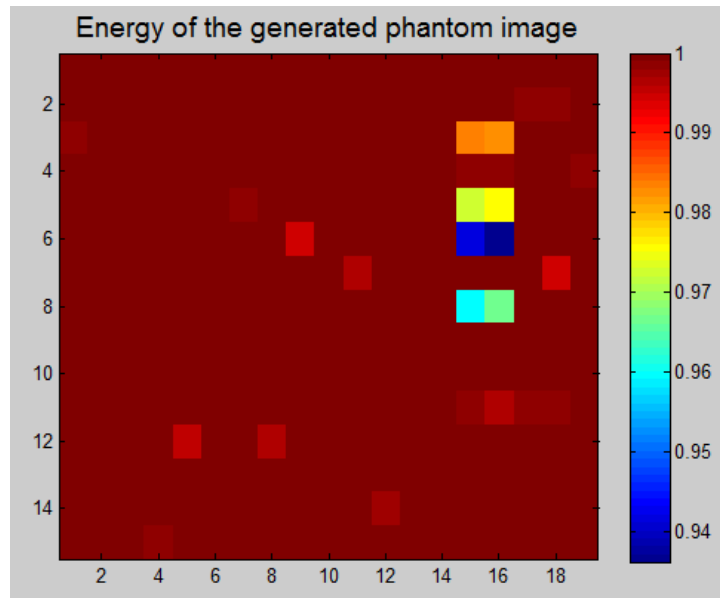


Figure 7.18: Heat map of Energy as a feature.

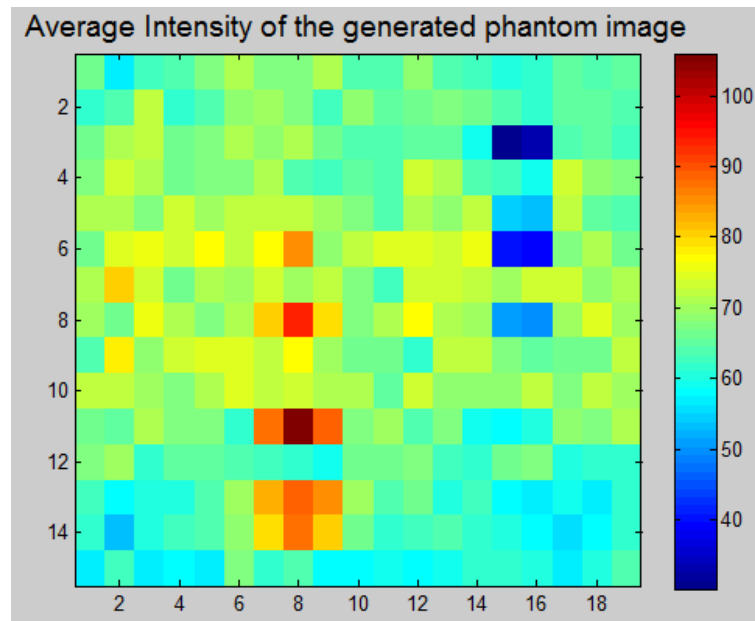


Figure 7.19: Color intensity of patch average intensity feature.

Figure 7.20 presents the heat map for skewness feature with patch sizes of 50 x 50. The heat map analysis results shows how skewness can detect non-speckle objects, although there are some false positives. Thus one needs to add other parameters, such as SNR, Kurtosis, etc. to remove or reduce this false positive.

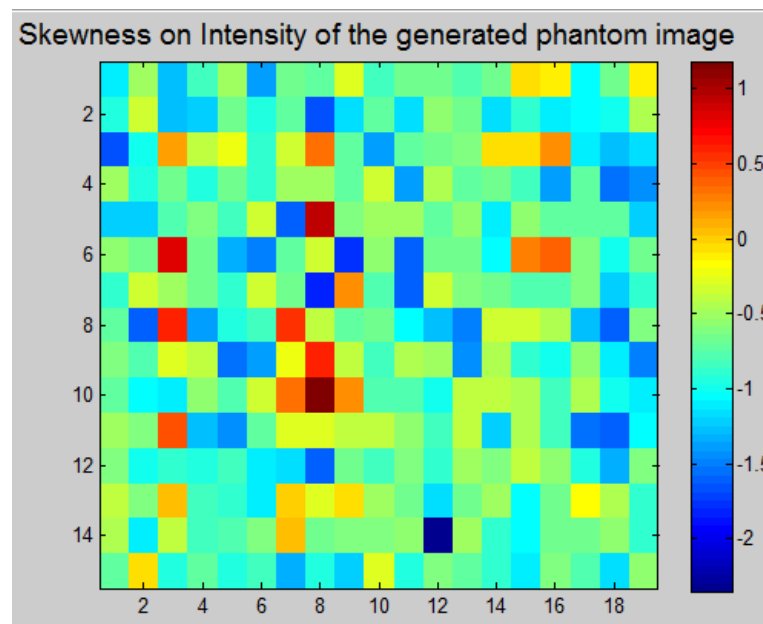


Figure 7.20: Color intensity of Skewness as a feature.

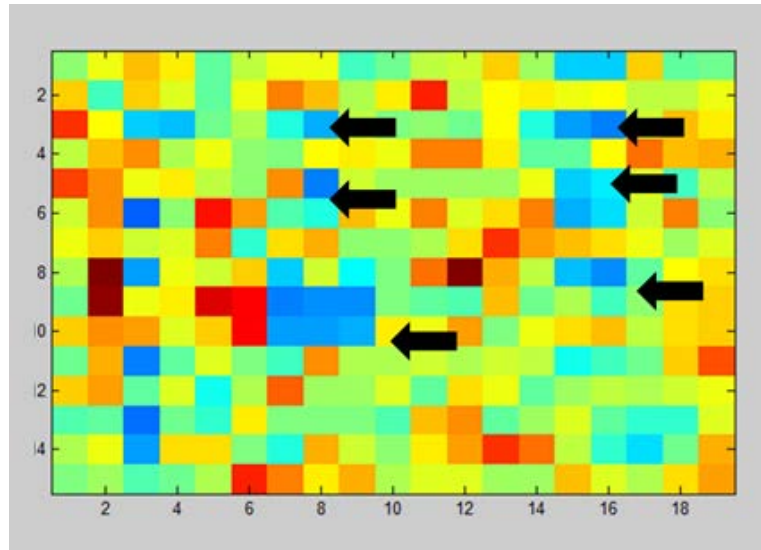


Figure 7.21: Color intensity (Heat map) of SNR (order 10) as a feature. False positives exist.

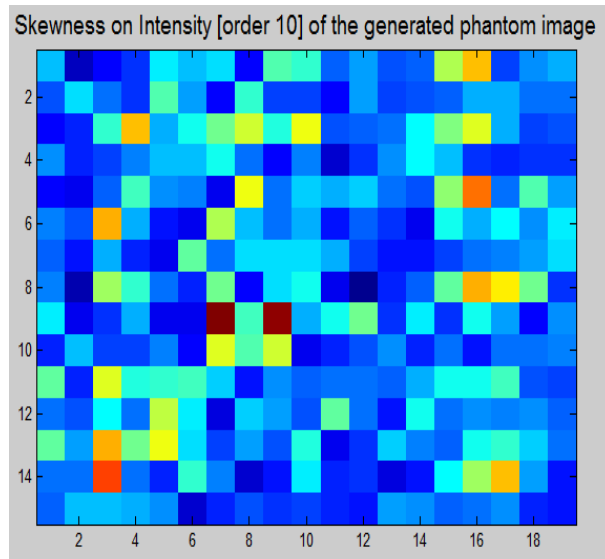


Figure 7.22: Color intensity of Skewness (order 10) as a feature.

We can use thresholding to test separability of the each feature as represented in Figure 7.23.

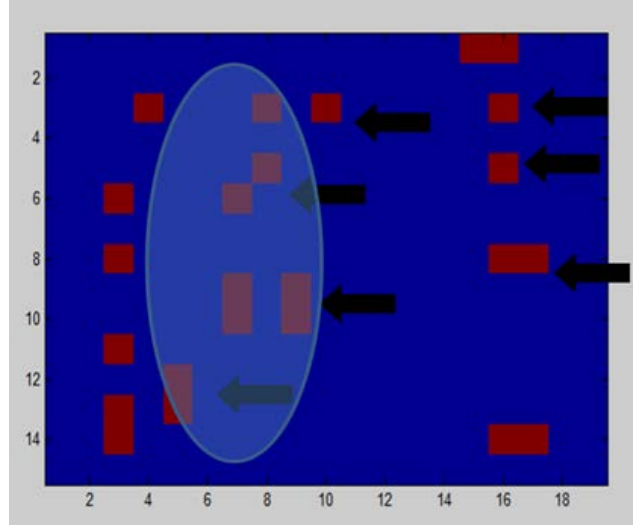


Figure 7.23: Binary image of the threshold heat map of the Skewness (order 10) as a feature. False positives exist.

The results of the investigations also showed that there is no single feature that works the best for classification, and the best results are achieved when fusing different features for classification purposes. The best results were shown when using the three features: SNR, Skewness and Kurtosis.

This work provides a complete simulation analysis to find the optimum powers to classify speckle, considering all three features (SNR, Skewness, and Kurtosis.) For each image patch A, the proposed statistical features in equation (6.1,) in Chapter 6 section 6.2, were calculated as following:

$$R = \frac{mean}{Std} = \frac{\langle A^\nu \rangle}{\sqrt{\langle A^{2\nu} \rangle - \langle A^\nu \rangle^2}}$$

$$Skewness = \frac{(A^\nu - \langle A^\nu \rangle)^3}{(\langle A^{2\nu} \rangle - \langle A^\nu \rangle^2)^{3/2}}$$

$$Kurtosis = \frac{(A^\nu - \langle A^\nu \rangle)^4}{(\langle A^{2\nu} \rangle - \langle A^\nu \rangle^2)^{4/2}}$$

and the optimum value of ν was searched for manually and in an exhaustive search manner. Whereas some studies use the same value of ν on each of the statistical calculations, they are assumed ‘independent’ in this study. One of the goals of this study is to find ‘optimal’ values of the ν variables (denoted ν_R , ν_S and ν_K for the ratio (R), skewness (S), and kurtosis (K) equations respectively) that will result in enhanced separability between the speckle and non-speckle regions of an image. The investigation sought the optimal values of ν_R , ν_S and ν_K that substitute ν in equations respectively. ‘A’ is the pixel intensity vector for each patch. Optimal values in this study are considered the values for which the results of the classification are considered “best classified” outputs.

Previous results [80] suggest that values of ν more than one ($\nu > 1$) perform well for speckle classification. Thousands of sample data are required to reliably calculate R and S [75]. In Dutt [75] and Prager et al. [83], values of ν that reduce this sample size are sought. Since R and S are different order moments of sample data, optimal values for ν in R and S are not necessarily the same. Prager et al. [83] found a fractional moment order (~ 1.8) to be optimal for speckle detection. Other studies have

reported that finding an optimal value for the moment order may not be justifiable and that a simple optimum moment order for statistical features does not exist [84].

In proposed Method 1, one may manually set the values of v_R , v_S and v_K each time results are generated, and use expert opinion and visual confirmation to investigate the goodness of the values chosen to optimize the classification results. Thousands of sample data were used to investigate the optimum results and make conclusions. The values were investigated in the range of values of v that fall in $0.1 < v < 10$, but the optimum values of v were observed to be in range of 1 to 2.5, based on different applications.

Initially, ground truth images were generated using expert opinion (manually labeled as FDS and non-FDS) based on the information available about the characteristics of each phantom in the Field II and reference articles. The user then visually assessed each patch and confirmed the results of the classification using the ground truth.

For further analyzing these three values, besides the visual assessment, a GUI was created and new code and toolbox called ‘PatchStat Toolbox’ was introduced, which has been used to analyze the values of v_R , v_S and v_K in each patch of the image by generating and displaying a HeatMap (2-D color image) for the values of R , S , and K (values of the features that are used in the classification.) This graphical representation can be used to analyze the properties of the (R, S, K) set in each patch. This code uses an image as the input and plots the statistical features R , S , and K

using the color code (color intensity.) The code doesn't have any effect on the classification results, but it is a code developed to visualize the intensity of the values of R , S , and K in all patches in the images. The figures below provide a very simple example for an ultrasound image with one cyst tissue, and represents how this algorithm works for patch statistical analysis. Using this toolbox, one can visually observe and confirm the location of the detected squares in the heat maps of different classification techniques, and compare them with the B-mode image. Figure 7.24 below is a graphical representation of the general program flow.

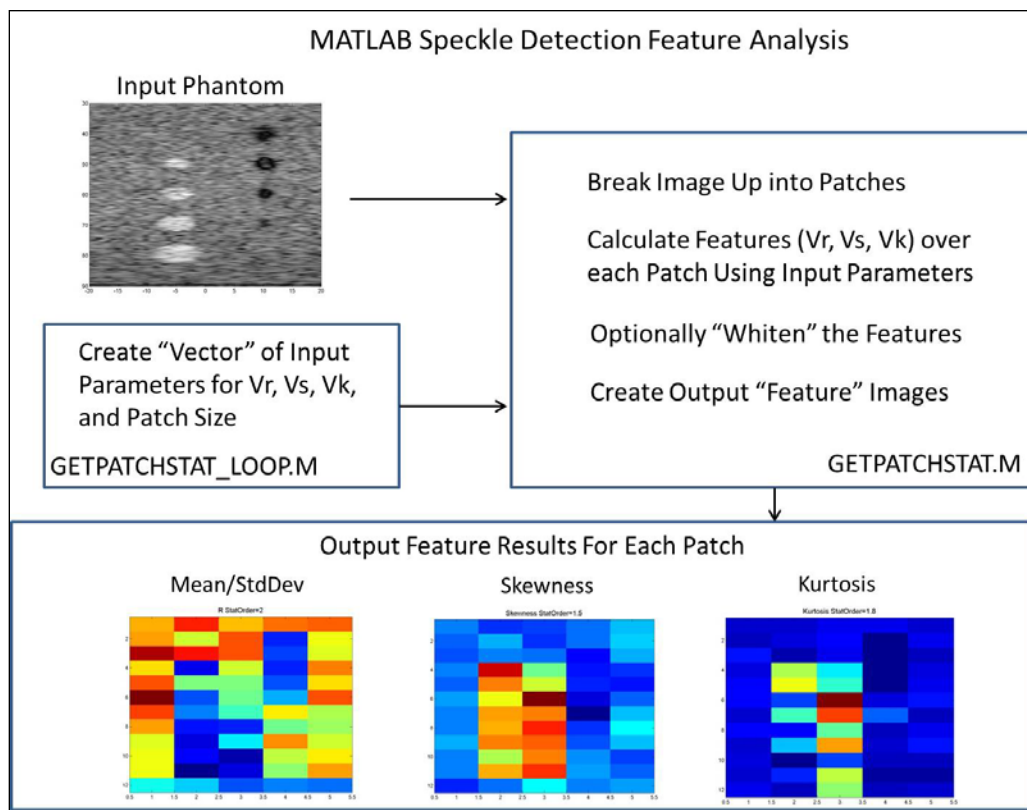


Figure 7.24: Feature analysis for speckle detection block diagram.

Example:

- `[R S K] = getPatchStat(Bmode,vR,vS,vK,rpPX,cpPX,do_whitening)`
- Play with patch size to increase separability
- One example patch size which separates cyst patch from speckle patches:

`[r s k]=getPatchStat(im,10,.1,.01,200,200,0);`

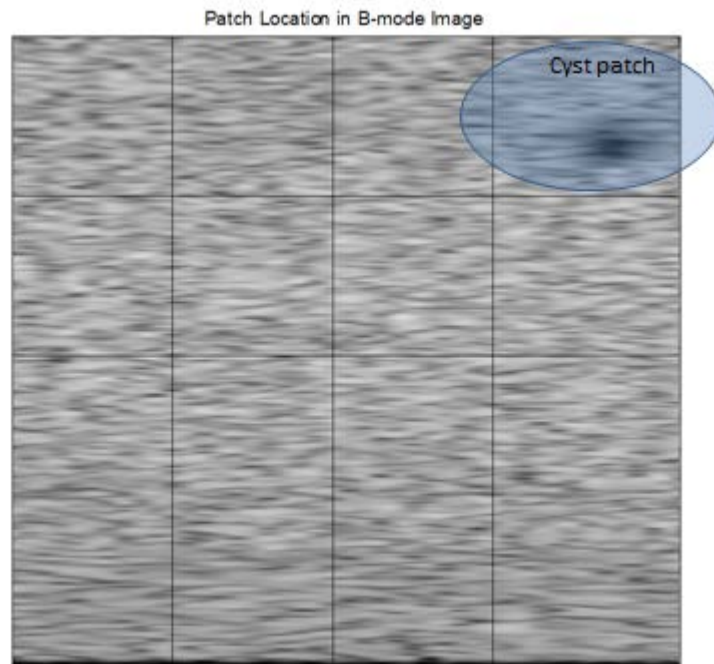


Figure 7.25: B-mode image with a small cyst in a patch at upper right corner of the image.

The statistical feature analysis images created after feeding this image into the algorithm would be as following. The results show that the values of the calculated R,

S and K features are different than the values in other patches and this variation makes the classification possible.

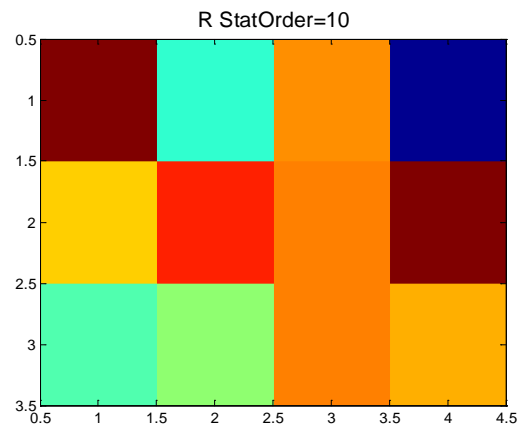


Figure 7.26: Color intensity of R in the patches.

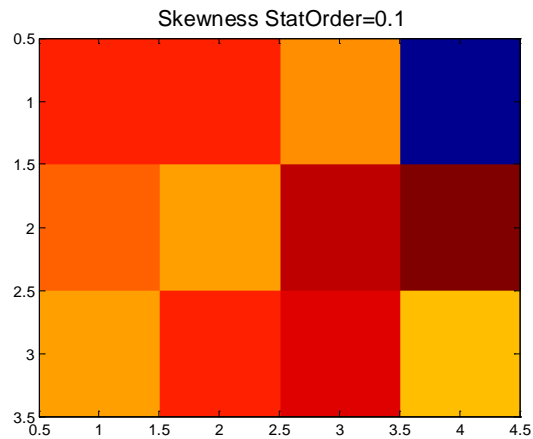


Figure 7.27: Color intensity of S in the patches.

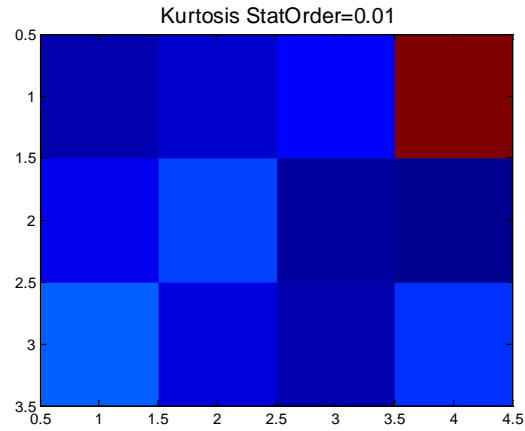


Figure 7.28: Color intensity of K in the patches.

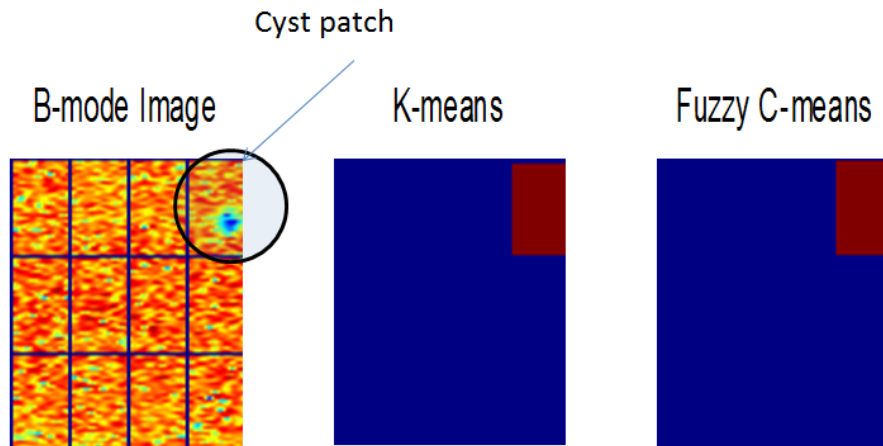


Figure 7.29: 100% accuracy in classification.

7.1.1.3 Results with Method 1

The results represented in this section are samples of thousands of classification results obtained by observing the individual features calculated based on patch-by-patch visual assessment for different values of v_R , v_S and v_K .

This study uses a simulated phantom containing eight cysts (two 3 mm, two 4 mm, two 5 mm, and two 6 mm.) Values of v_R , v_S and v_K , ranging from $0.1 < v < 10$, and values of v_R , v_S and v_K were considered to be independent of each other. Based on other studies, suggested values for possible investigators can be combinations of 0.5, 0.8, 1.0, 1.5, 1.8, 2.0, 3.0, etc. Results for various combinations of values of v_R , v_S and v_K and different patch sizes were generated and studied (e.g., 50 x 50, 80 x 80, 100 x 100 pixels.) The classification outputs for each case were written as JPEG files.

As an initial assessment, all results were visually assessed and confirmed. Using an automated ‘scoring tool’ would have made the evaluation quicker, but would not have been the ‘best’ way to rank the results, as these are going to be used as the ground truth analysis and sometimes a cyst is only partially within a patch, so it is often difficult to determine which patches in an image should really be considered ‘speckle’. Therefore, it was concluded that it would be the best for a person to evaluate the results by visually inspecting and confirming all of the output classification images.

Figure 7.30 represents the classification results of the cyst phantom with a patch size of 100 x 100 pixels, and (2.0, 1.5, 1.8) set for the values of v (values of v_R , v_S and v_K .) using Method 1 and visual inspection and confirmation for all of the output images. It should be noted that there is no relation between the color of the binary image of the patch (red or blue) and the patch being labeled as FDS or non-FDS, as this is assigned randomly by MATLAB in each run. In this specific example, the color blue represents speckle (FDS) and the color red represents non-speckle

(non-FDS) patches. As can be seen, the non-speckle patches are the ones that include the cysts in the image. In this example, Fuzzy C-Means and K-means classifiers performed exactly the same.

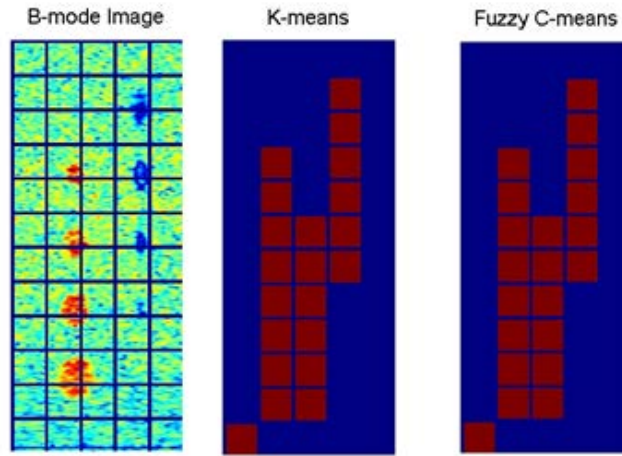


Figure 7.30: Classification results for the $v_R=2.0$, $v_S=1.5$, and $v_K=1.8$ set.

The PatchStat toolbox was used to further analyze and investigate the v values used to calculate R , S and K . The figures below represent the HeatMap for the statistical analysis of the values of the features calculated in each patch of the cyst phantom described for the set of $v_R=2.0$, $v_S=1.5$, and $v_K=1.8$. The intensity of R , S , and K are being measured in each patch.

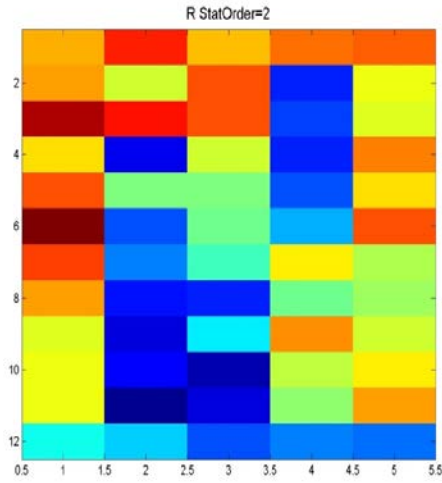


Figure 7.31: Statistical analysis of the R (mean to standard deviation ratio feature) values in the image patches. For $v_R=2.0$, both bright and dark cysts were found and general speckle regions are represented by orange/light red colors.

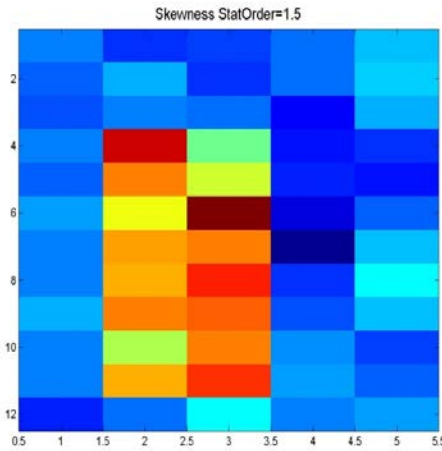


Figure 7.32: Statistical analysis of the S (skewness) feature values for each image patches for $v_S=1.5$. In general, the speckle regions are blueish. The darker cysts are presented with darker blue and the brighter cysts are presented with more orange color intensities. The point here is that the image confirms that in non-speckle regions the values of the intensities of the S are far different than in the speckle regions.

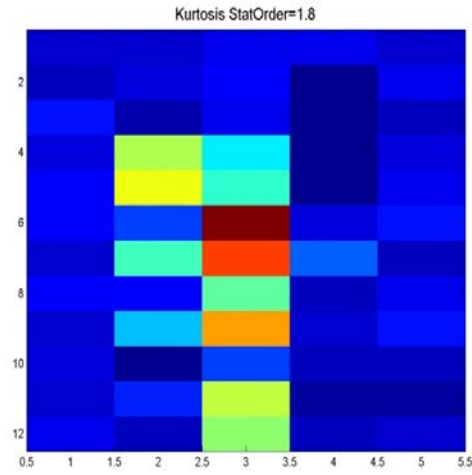


Figure 7.33: Statistical analysis of the K (kurtosis) feature values in the image patches for $v_K=1.8$.

The algorithm results were investigated and confirmed using different phantom sizes and shapes, such as square cyst/object phantom data. The following images are examples of such investigations that were done for different sets of values of v_R , v_S and v_K for the square cyst phantom.

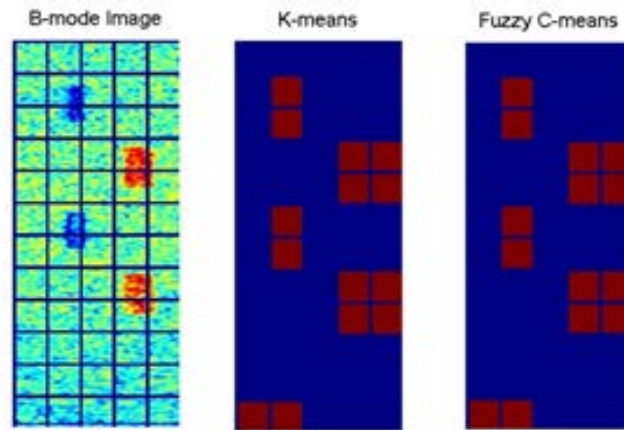


Figure 7.34: Classification results for the $v_R=2.0$, $v_S=1.5$, and $v_K=1.8$ set.

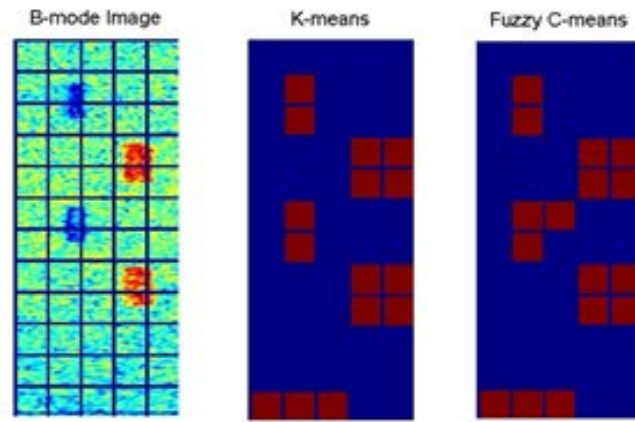


Figure 7.35: Classification results for the $v_R=1.0$, $v_S=1.8$, and $v_K=0.8$ set.

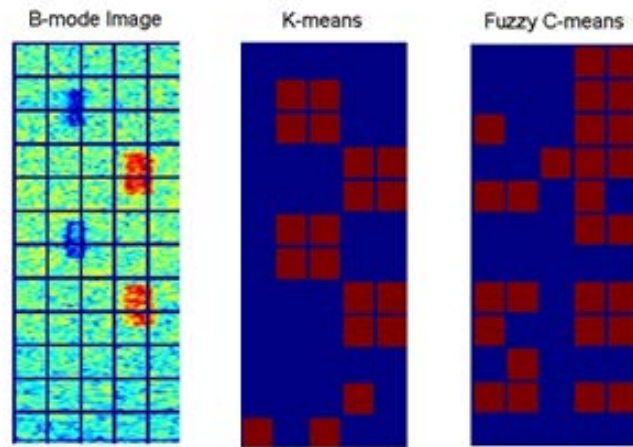


Figure 7.36: Classification results for the $v_R=1.0$, $v_S=0.8$, and $v_K=3.0$ set.

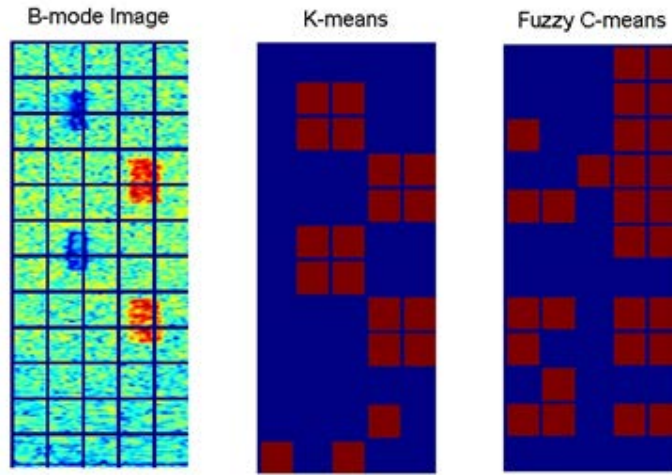


Figure 7.37: Classification results for the $v_R=1.0$, $v_S=0.5$, and $v_K=2.5$ set.

The effect of the chosen number of scatterings on the value of ν in the proposed statistical features was also investigated. The aim of the investigation was to find optimal values using different scattering values from the phantoms generated during the study, such as the phantoms presented in Figure 7.38, which represents two phantoms with different sets of sizes for cysts and a different number of cysts.

Phantoms were generated with different values for the scattering parameter to study the effects of the number of scatterings in phantom generation and classification: e.g., 10000, 50000, 100000, 250000, 1000000, etc.

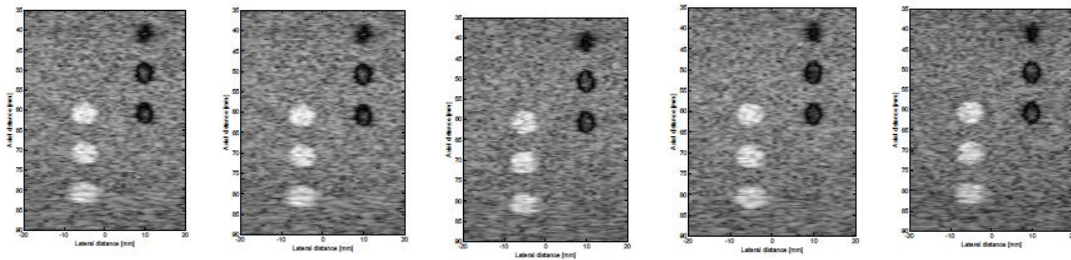


Figure 7.38: Phantom results with 10000, 50000, 100000, 250000, and 1000000 scatterers (left to right).

The values of ν in the range of $0.2 < \nu < 2.5$ with 0.6 step increments were investigated and searched for the best values that generate the best results (using a patch size of 100 x 100 pixels):

```

lower_lim = 0.2; % lower limit to generate values of statistical params
upper_lim = 2.5; % upper limit to generate values of statistical params
step_size = 0.6; % step size to generate values of statistical params
vrr = lower_lim:step_size:upper_lim;
[rv,cv]=size(vrr);
rcount = 1;
for ri=1:1:cv
    e1 = vrr(ri);
    for rii = 1:1:cv
        e2=vrr(rii);
        for riii = 1:1:cv
            e3 = vrr(riii);
            VrVsVk(rcount,:)=e1,e2,e3];
            rcount = rcount + 1;
        end
    end
end
num_sims = rcount - 1;

```

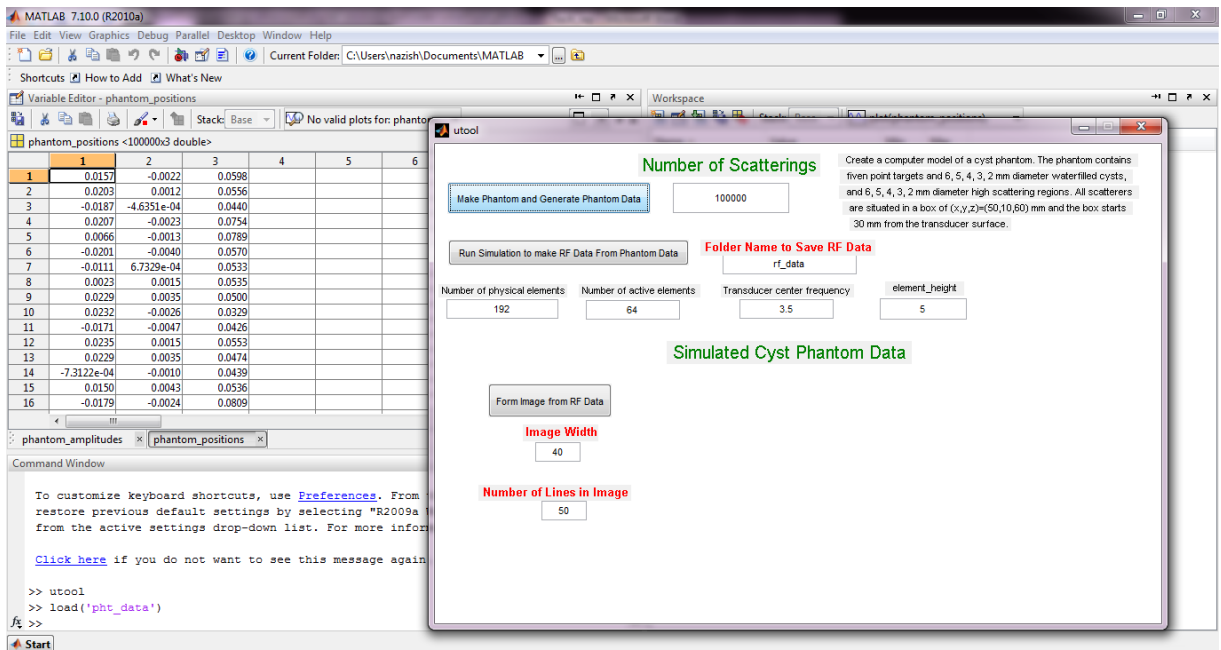
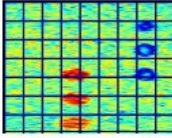
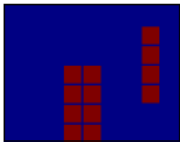
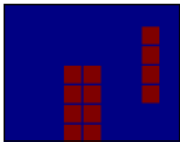
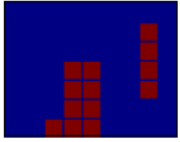
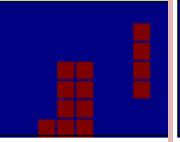
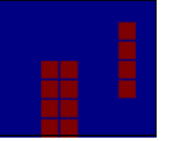
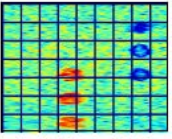
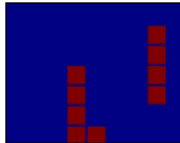
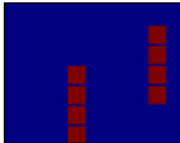
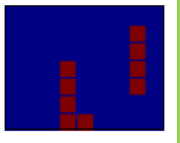
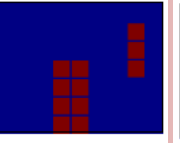
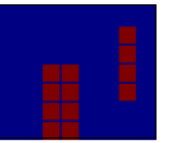
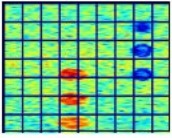
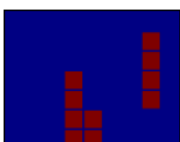

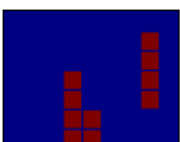
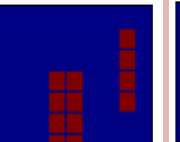
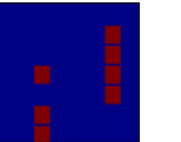
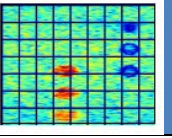
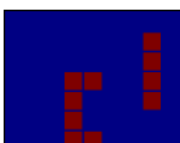
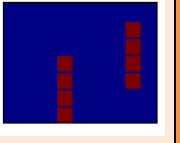
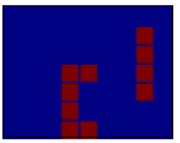
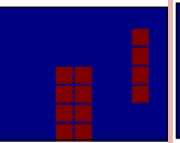
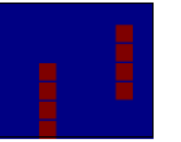


Figure 7.39: Snapshot of cyst phantom generation line by line using the UTool Software.

The increase in the number of scatterings had little effect on the performance of the clustering techniques. The values specified in the table are noted at the first occurrence of resemblance found in the B-mode image and the respective HeatMap. The values can be fine-tuned around the observed values to improve performance and better detection. It seems that overall, the K-means and K-mediod methods showed better performance in this investigation. The following table shows images that correspond to the optimal statistical parameters chosen.

Table 7.1: Classification results with different number of scatterers.

clustering method	k-means	k-mediod	Fuzzy C means	GK	GG
	v_R, v_S, v_K	v_R, v_S, v_K	v_R, v_S, v_K	v_R, v_S, v_K	v_R, v_S, v_K
Phantom with scattering 50,000 	0.8,0.2, 2 	0.8,0.2, 2 	0.8,0.2, 2 	0.8,0.2, 2 	0.8,0.2, 2 
Phantom with scattering 100,000 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 
Phantom with scattering 250,000 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 
Phantom with scattering 1000,000 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 	0.8,0.8,1.4 

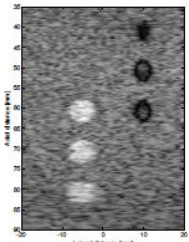
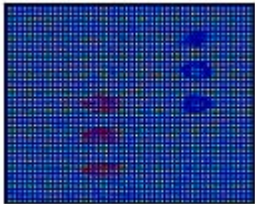
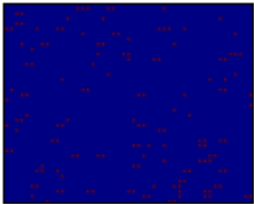
The effect of the selection of patch size on the selection of values for v , and consequently the possible effects on the classification results, were also studied. The number of scatterings was kept constant at 100,000 and the patch sizes were

investigated as: 20 x 20, 50 x 50, 70 x 70, 80 x 80, 100 x 100, 150 x 150, 200 x 200, and 250 x 250.

The following tables show some sample results from the investigation on cyst phantoms. However, different other phantoms were studied that are not reported here.

In this example, it is observed that almost all clustering techniques performed well for patch size of 80 or more. When the patch size increases, almost all clustering techniques were able to mark the areas where possible cysts lie. So, selecting a larger patch size can be suitable at the start of analysis to mark the area where cyst can be, and then fine-tuning of the parameters in the marked area can produce better results.

Table 7.2: Patch size effect on K-means classification.

Phantom type/Clustering method	K-means			B-mode image	Detected Image
	v_R, v_S, v_K				
Patch size 20x20	2	0.8	2		

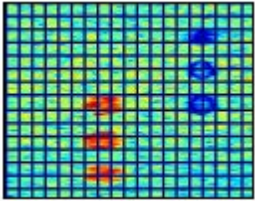
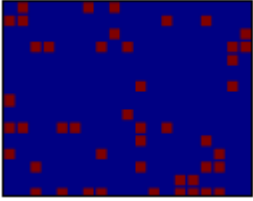
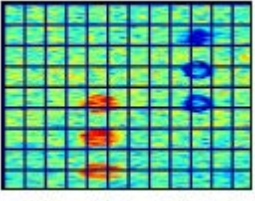
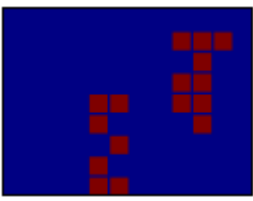
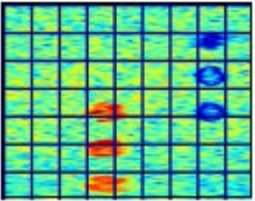
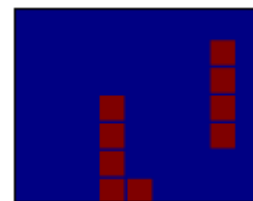
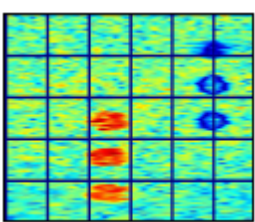
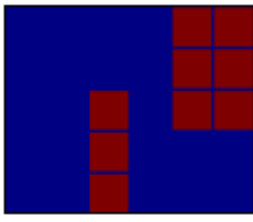
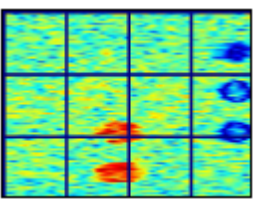
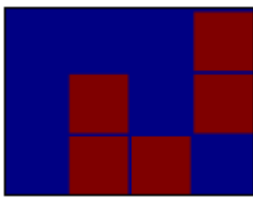
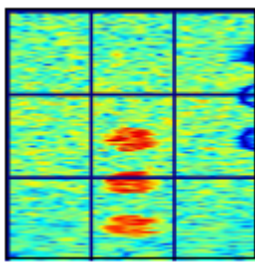
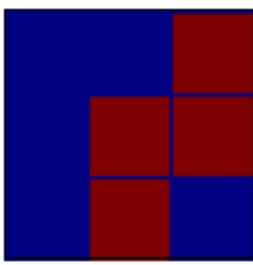
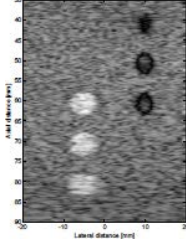
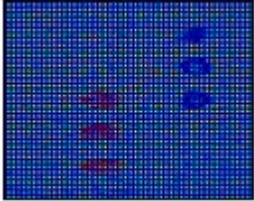
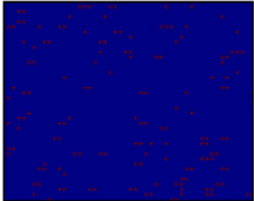
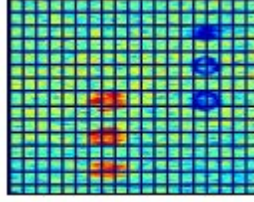
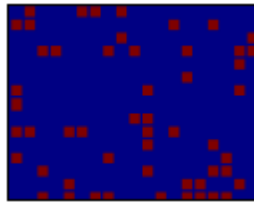
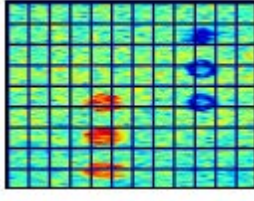
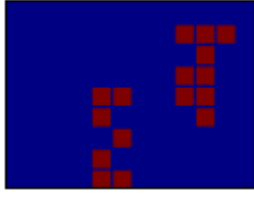
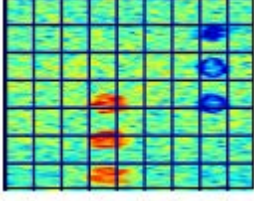

Patch size 50x50	1.4	0.8	2		
Patch size 80x80	1.4	0.8	1.4		
Patch size 100x100	1.4	1.4	2		
Patch size 150x150	0.8	0.2	0.8		
Patch size 200x200	2	2	0.8		
Patch size 250x250	0.8	0.8	1.4		

Table 7.3: Patch size effect on K-mediod classification.

Phantom type/Clustering method	K-mediod			B-mode image	Selected image
	v_R, v_S, v_K				
Patch size 20x20	NA ¹	NA	NA		
Patch size 50x50	NA	NA	NA		
Patch size 80x80	1.4	0.8	1.4		
Patch size 100x100	2	0.8	1.4		

¹ NA means that there were no sets of values of v that were found to show valid and logic classification for this technique with the corresponding patch size.

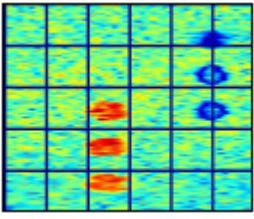
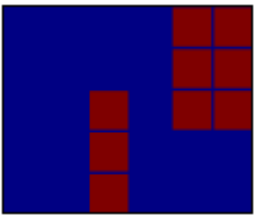
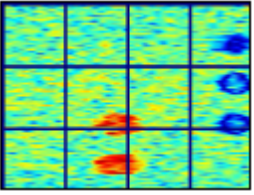
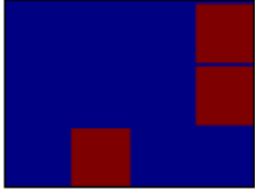
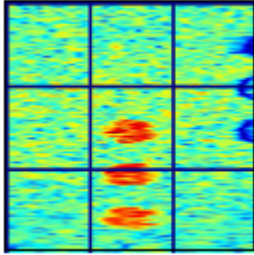
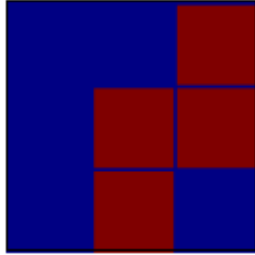
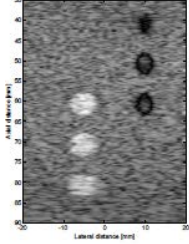
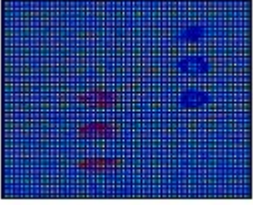
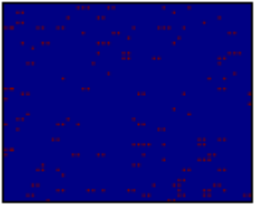
Patch size 150x150	0.2	0.2	0.2		
Patch size 200x200	2	2	1.4		
Patch size 250x250	0.8	1.4	1.4		

Table 7.4: Patch size effect on Fuzzy C-Means classification.

Phantom type/Clustering method	Fuzzy C-Means			B-mode image	Selected image
	v_R, v_S, v_K				
Patch size 20x20	NA	NA	NA		

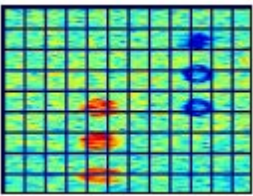
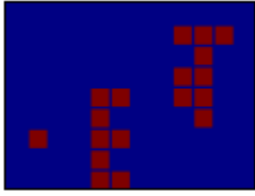
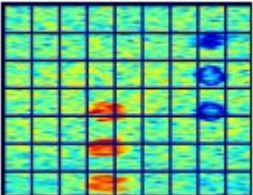
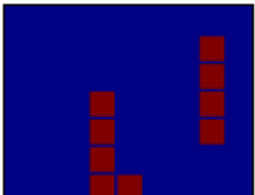
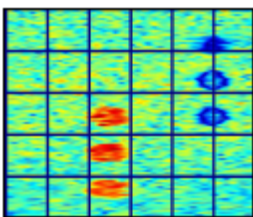
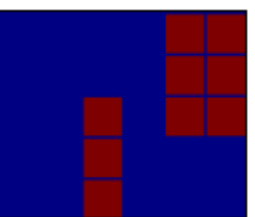
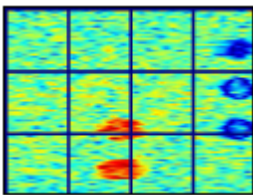
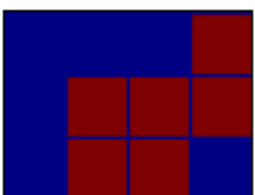
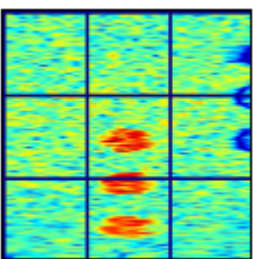
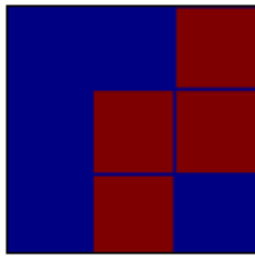
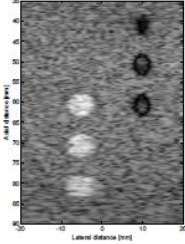
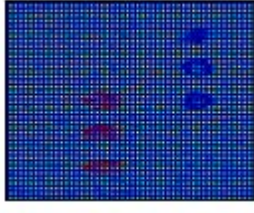
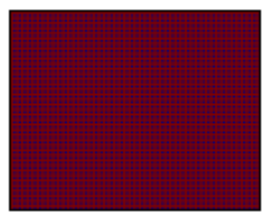
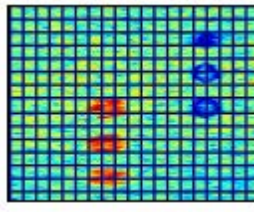
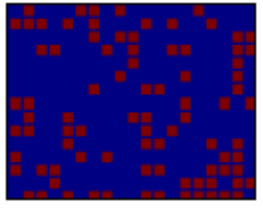
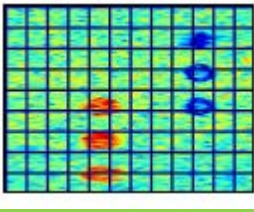
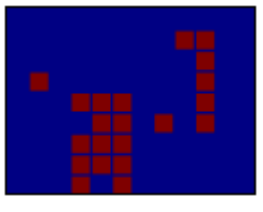
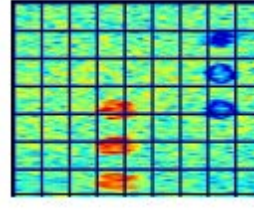
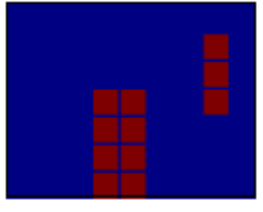
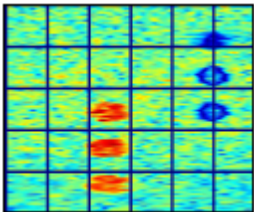
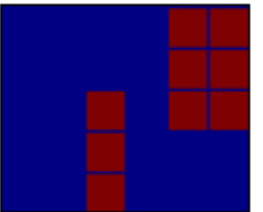
Patch size 50x50	2	2	1.4		
Patch size 80x80	2	1.4	1.4		
Patch size 150x150	0.8	0.2	0.8		
Patch size 200x200	2	2	0.8		
Patch size 250x250	0.8	0.8	1.4		

Table 7.5: Patch size effect on GK classification.

Phantom type/clustering method	GK			B-mode image	Selected image
	ν_R, ν_S, ν_K				
Patch size 20x20	NA	NA	NA		
Patch size 50x50	NA	NA	NA		
Patch size 80x80	0.2	2	0.8		
Patch size 100x100	0.8	2	1.4		
Patch size 150x150	0.2	0.2	0.2		

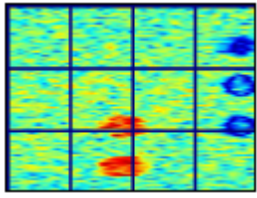
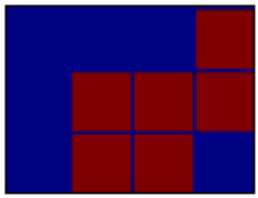
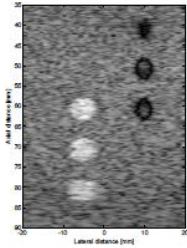
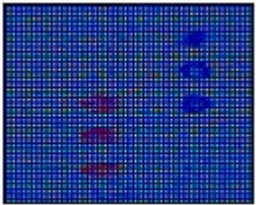
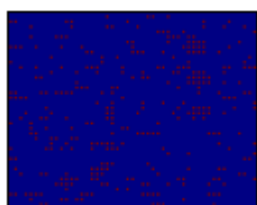
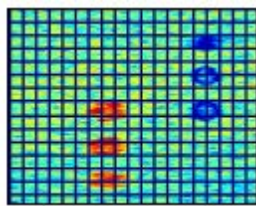
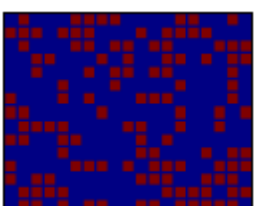
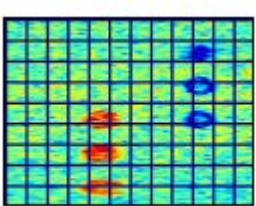
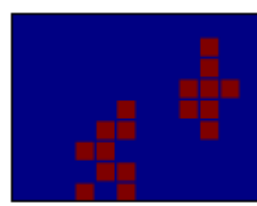
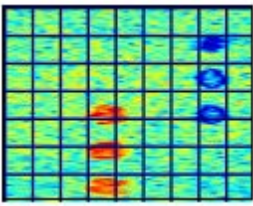
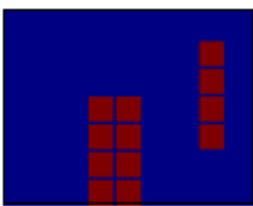
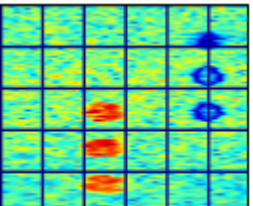
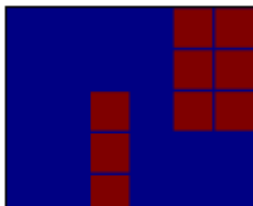
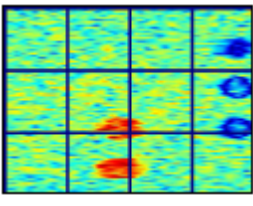
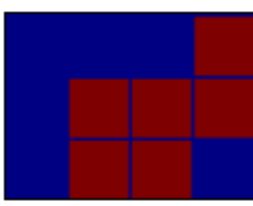
Patch size 200x200	2	0.2	0.8		
--------------------	---	-----	-----	--	---

Table 7.6: Patch size effect on GG classification. Phantom type/clustering method	GG			B-mode image	Selected image
	v_R, v_S, v_K				
Patch size 20x20	0.8	1.4	2		
Patch size 50x50	0.8	2	0.8		
Patch size 80x80	0.2	1.4	0.8		

Patch size 100x100	0.8	1.4	0.8		
Patch size 150x150	0.2	0.2	0.2		
Patch size 200x200	2	0.2	0.2		

The patch size effect on the selection of the statistical features was also investigated, considering the previously introduced cyst phantom and study of skewness and SNR of patches within the image.

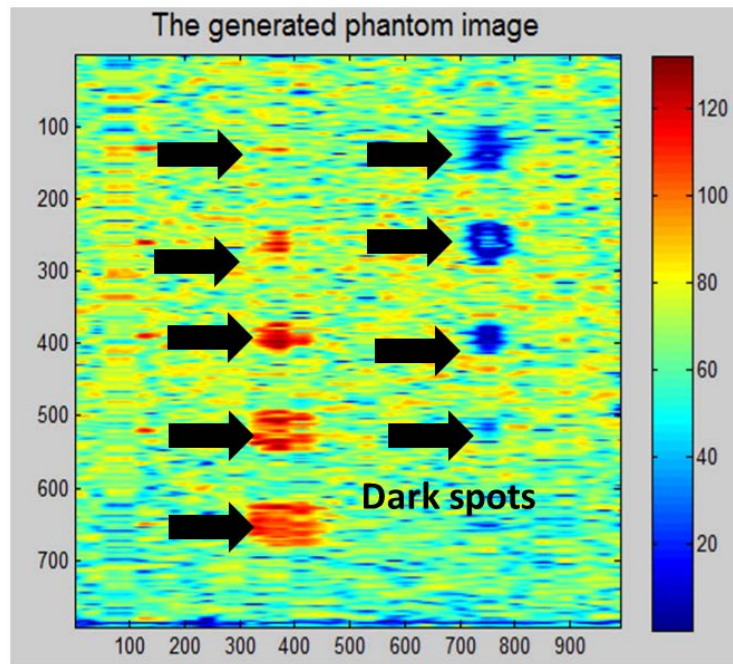
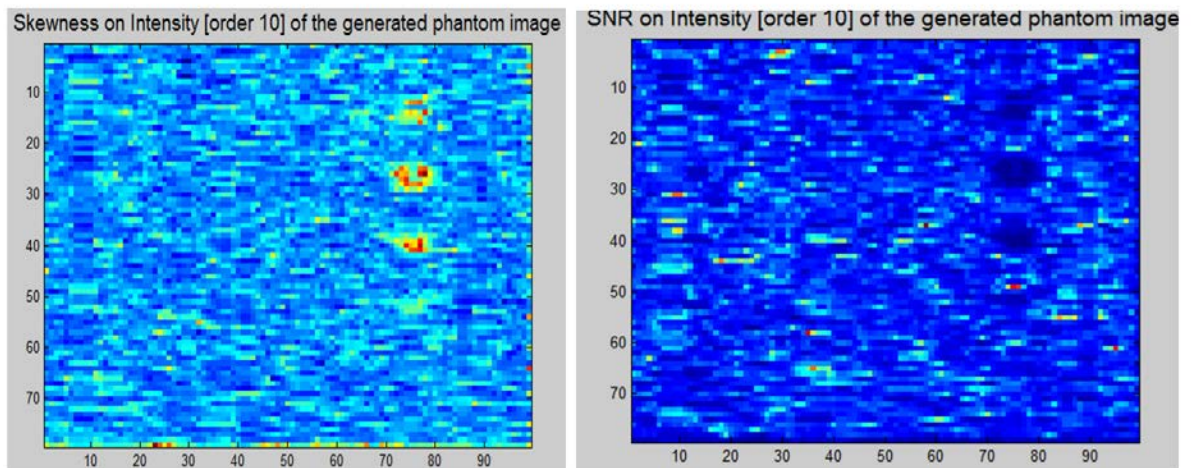
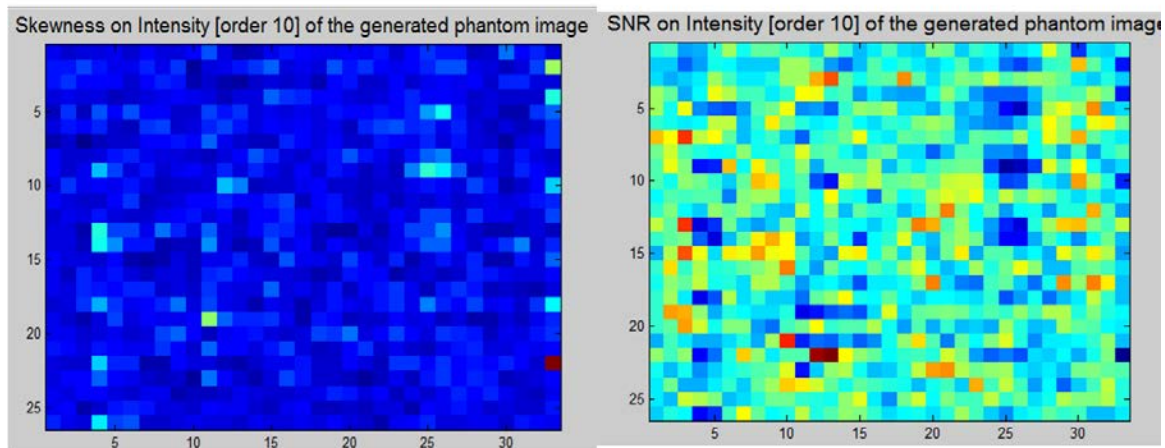


Figure 7.40: Cyst phantom with dark and bright cysts in color scale.



Patch size=10x10

Figure 7.41: Skewness and SNR for cyst phantom with patch size 10x10. Cysts are not obvious in skewness and SNR.



Patch size=30x30

Figure 7.42: Skewness and SNR for cyst phantom with patch size 30x30. Cysts are partially obvious in skewness and SNR.

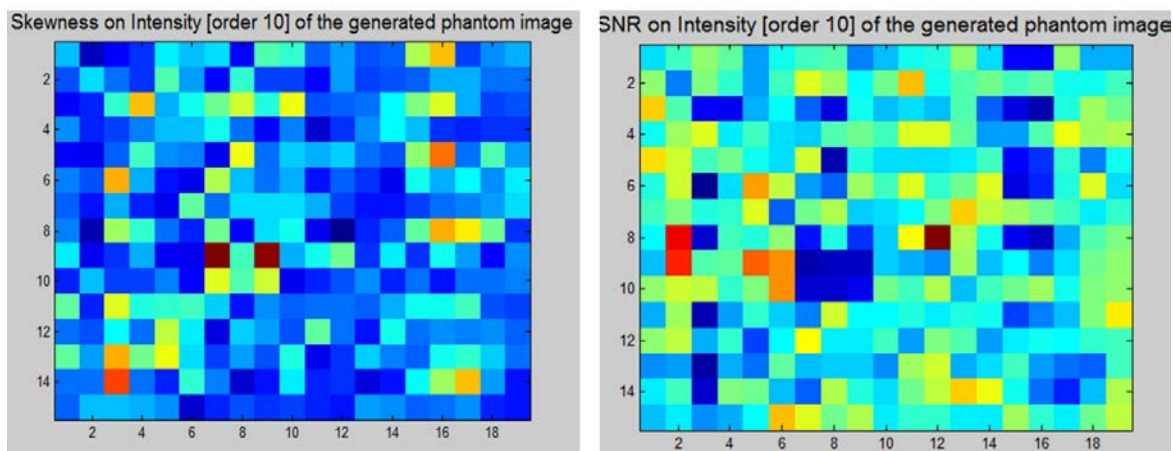


Figure 7.43: Skewness and SNR for cyst phantom with patch size 50x50. Cysts are obvious in skewness and SNR.

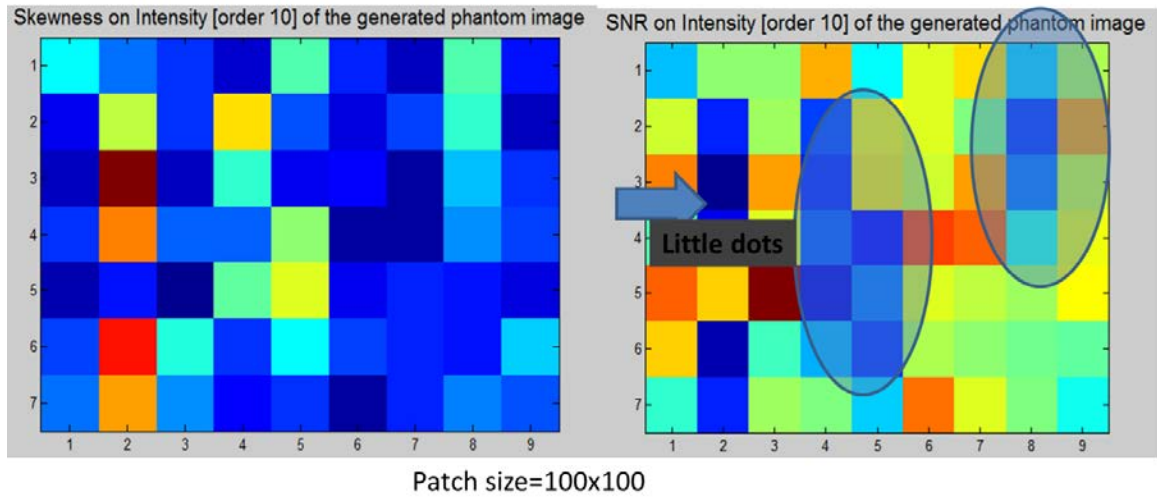


Figure 7.44: Skewness and SNR for cyst phantom with patch size 100x100. Cysts are obvious in skewness and SNR.

7.1.1.4 Performance Analysis Using Cyst Phantom for Method 1

Method 1 was evaluated using B-mode images of cyst phantoms with 100,000 scatterers and 50 RF lines [29, 94]. First, the images were segmented into image patches of 100 x 100 pixels to compute the desired statistical features. This same patch size was used for both simulations to allow for a comparative analysis on the speckle detection performance of the two methods. While K-Distribution is good for modeling image patches with a lower amount of scattering, however, the image patch size needs to be reasonable and cannot to be overly large or small [29].

Then the proposed statistical features - R, Skewness and Kurtosis - for the image patch were calculated. After calculating the statistical features for each image patch, we can classify them as FDS or non-FDS using unsupervised clustering techniques. For this purpose, this study applied five pattern classification techniques:

K-means, K-medoid, Fuzzy C-Means, Gustafson-Kessel fuzzy classifier and Gath-Geva fuzzy classifier.

Figure 7.45 demonstrates the results of the discussed classification approaches for speckle detection in a cyst phantom. Speckle detection results for five different unsupervised classifiers are presented. The total number of 100 x 100 patches for the phantom image was 24. Patches classified as fully developed speckles (FDS) are shown as black. All methods except GK-fuzzy classifier performed the same. Orders for statistical features respectively were 1,1 and 0.5 [29, 30].

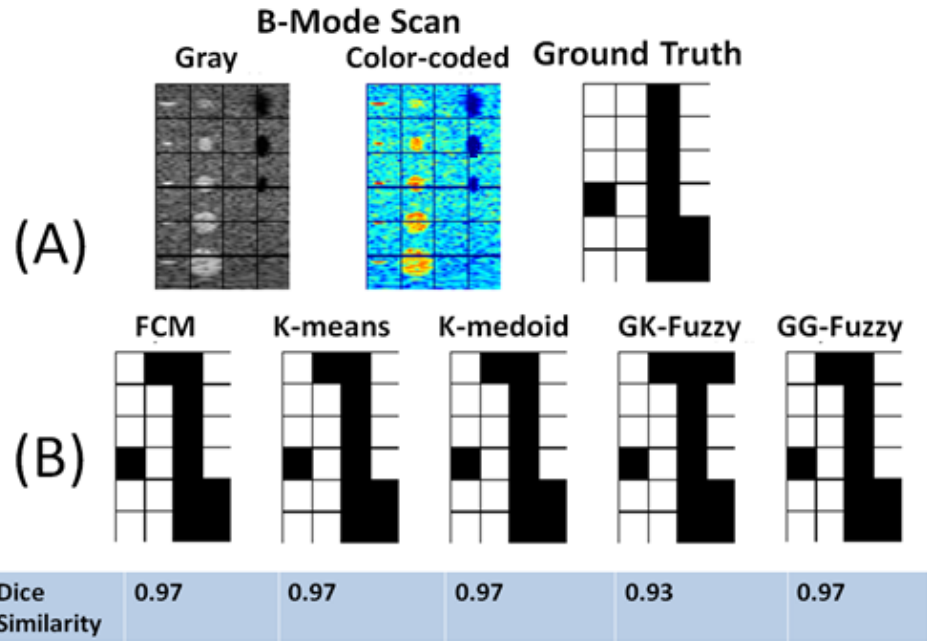


Figure 7.45: (A) Simulated ultrasound image of a Cyst phantom and the Ground Truth created by expert manual labeling (White (1) color is representing Cyst patches and black (0) is representing fully developed speckle (FDS) patches). (B) Speckle detection results for five different unsupervised classifiers. Total number of 100x100 patches for the phantom image was 24. Patches classified as fully developed speckles (FDS) are shown as black. All methods except GK-fuzzy classifier performed the same. Orders for statistical features respectively were 1,1 and 0.5 [29].

7.1.2. Calculation of B-mode image of synthetic Kidney

A B-mode image of a synthetic kidney was recreated identically according to previously described methods [100]. Anatomical phantoms attempt to model images as they appear in real human patients. To accomplish this, a bitmap image of scattering strength is generated from a given ROI. This generated map determines the multiplied factor onto the scattering amplitude generated from the Gaussian distribution [100]. The map is also used to model the difference in the tissue densities

as well as the differences in the speed of sound perturbations in the given biological tissue. Simulated boundaries were generated by creating lines in the scatterer map along which the strong scatterers were placed [100], and are marked by the completely white lines appearing in the scatterer maps. While the current mode is 2D, it can be expanded to 3D as well. The elevation direction is implemented by making a 15 mm thickness for the scatter positions randomly distributed in the interval [29, 30, 100]. Optical pictures from the Visible Human Project of National Library of Medicine (NLM) database of the NIH (<http://www.nlm.nih.gov/research/visible/>) were used to generate the scatterer map. This Visible Human Project database includes full body human cadaver scanning generated from high resolution CT and MR imaging of sliced cadavers into 1 mm sectionals for generating the images [29, 30].

A left kidney phantom in a longitudinal sonographic plane was then created and 1,000,000 scatterers were randomly distributed within the phantom. This was done with a scatter amplitude following a Gaussian distribution with a standard deviation calculated from the scatter map. The phantom was scanned with a 7 MHz 128 element phased array transducer with $\lambda/2$ spacing and Hanning apodization. A single transmit focus 60 mm from the transducer was used, and focusing during reception was at 5 to 150 mm in 1 mm increments. The image consists of 128 lines with 0.7 degrees between lines. Figure 7.60 below displays the artificial kidney scan. (<http://field->

ii.dk/examples/kidney_example/kidney_example.html). The m-files (matlab code) can be found at Appendix C [29, 30].

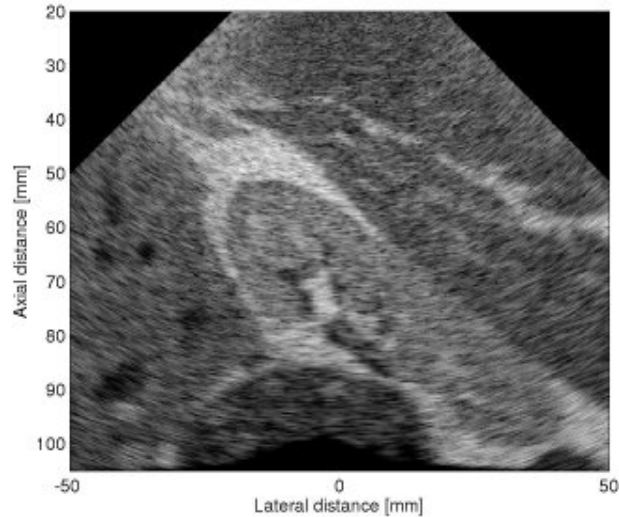


Figure 7.46: Artificial kidney scan [105].

7.1.3. Calculation of B-mode image of synthetic Fetus

Calculation of a B-mode image of a synthetic fetus was recreated identically according to previously described methods [101]. The methods described in this section are therefore directly recreated from those described in [101]. This work then focused on fetal phantoms that, like kidney phantoms, attempt to simulate images would be observed in a real human fetus including the same distributions and parameters (http://field-ii.dk/examples/fetus_example/fetus_example.html).

A phantom for a three month old fetus was made [94]. 200,000 scatterers were randomly distributed within the phantom, and a Gaussian distributed scatter

amplitude with a standard deviation was determined by the scatter map [101]. The phantom was scanned with a 5 MHz 64 element phased array transducer with $\lambda/2$ spacing and Hanning apodization. A single transmit focus 70 mm from the transducer was used, and focusing during reception was at 40 to 140 mm in 10 mm increments. The image consists of 128 lines with 0.7 degrees between lines [29, 30, 101].

The fetus reconstructed phantom image is shown below as directly replicated from prior work [101]. There is clear anatomical delineation detail at the level of the scatterer map. The same boundary features as the kidney image are also seen. The images have many of the features from real scan images, but still lack details. This can be ascribed to the low level of details in the bitmap images, and also that only a 2D model is used. The images do show great potential, however, for making powerful, fully synthetic phantoms that can be used for image quality evaluation.

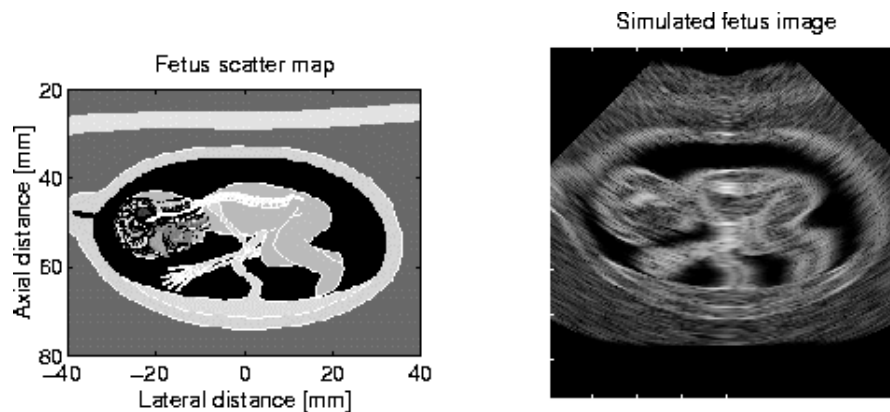


Figure 7.47: Artificial fetus in 12th week [101].

The proposed speckle classification scheme was evaluated and the performance of each unsupervised classification technique was calculated using the fetus phantom as described above [29, 30]. The figures below respectively show the performance of the classification methods for speckle detection for the simulated fetus phantom. Figures 7.49 and 7.50 show the performance of the classification methods for speckle detection.

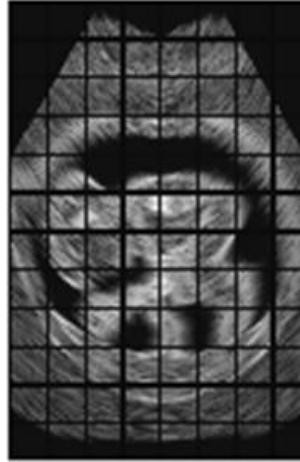


Figure 7.48: Simulated fetus phantom and patches. Images were segmented into 12x8 image patches, where each image patch had a size of 100x100 pixels [29, 30].

The proposed statistical features (R , S and K) were evaluated and the performance of the five different machine learning techniques were compared. Furthermore, the optimum values for v_R , v_S and v_K were investigated, which are represented in this section using the developed tools.

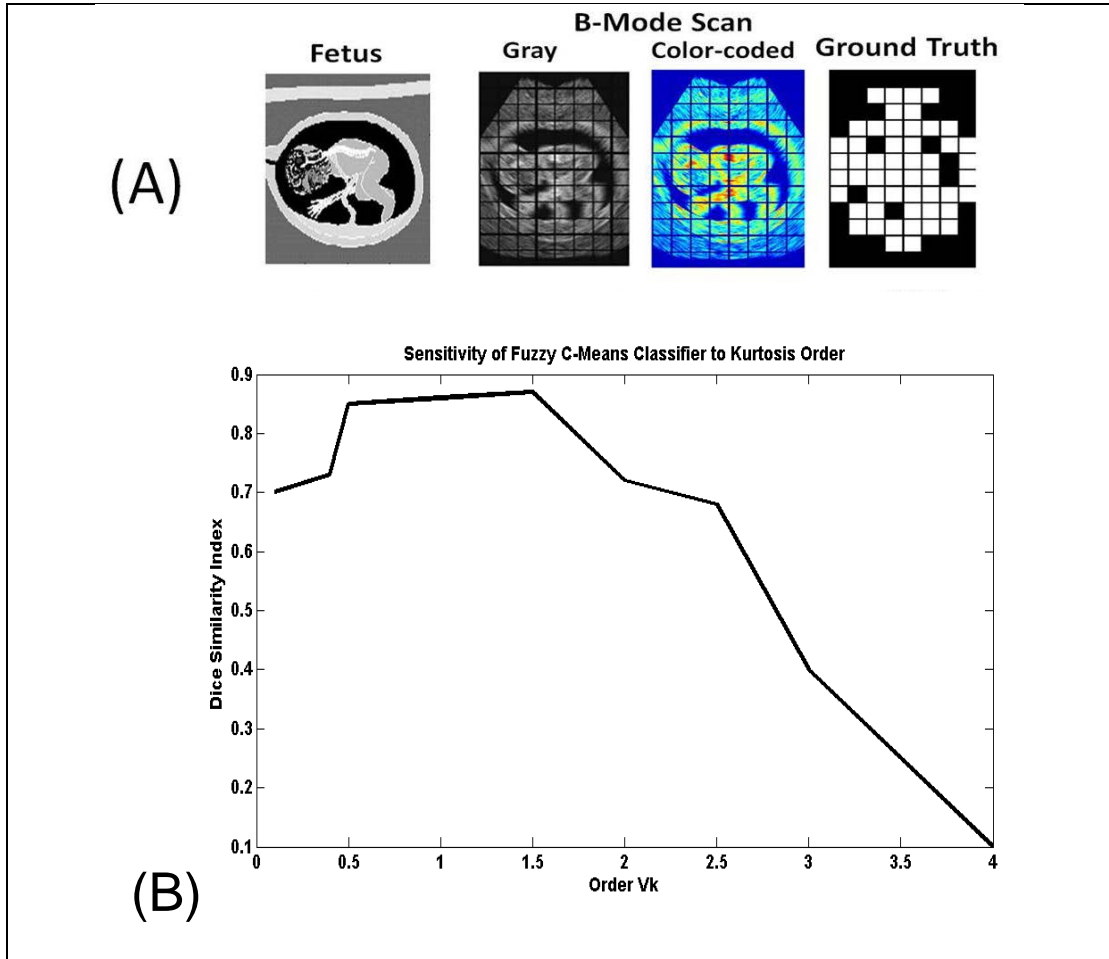


Figure 7.49: Sensitivity analysis for Method 1: (A) Input image patches. (B) Sensitivity of Fuzzy C-Means classifier to orders of statistical features. v_K varies from 0 to 10, while other orders, v_R , v_S are set to 1, (1, 1, at v_K). It can be seen that best results (max Dice similarity) is achieved at $v_K=1.5$ and best v_K values are in range $0.5 < v_K < 1.5$.

To evaluate performance of the proposed method for speckle detection, the Dice similarity index (DS) was used. The Dice similarity index is always between 0 and 1. DS=0 means there is no similarity and DS=1 means two regions or images are identical and there is no difference between their shape and location.

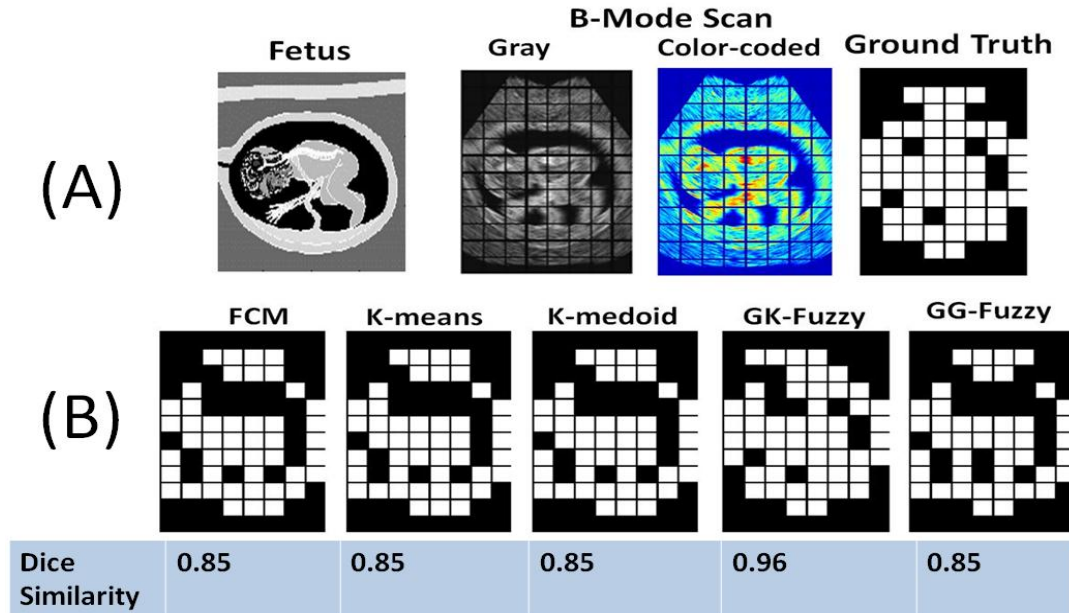


Figure 7.50: Simulated ultrasound image of a fetus in 12th week (A) and speckle detection results for 5 classifiers (B.) Patches classified as FDS are shown as black. In this case, FCM, K-means and K-mediod performed the same. GK- and GG-fuzzy classifiers were able to decrease false positives and improve accuracy of the speckle detection. Total number of patches (100x100 pixels) for the phantom image was 96. Orders for features were 10,1 and 0.01 [29, 30].

7.1.4. Calculation of B-mode image of synthetic Heart

Like the kidney phantoms, the heart phantoms attempt to generate images as they will be seen in real human subjects and with the same distribution, parameters and settings applied for their model.

Several heart phantoms were generated for investigational purposes. Heart phantom generation is exactly like kidney, but the code attached to Appendix C.

Indeed the main difference is that instead of a kidney bitmap file one needs to use a heart bitmap file.

Then the following steps need to be followed:

- 1. Run [human_kidney_phantom.m](#), but read attached bmp instead.
- 2. Run [sim_kidney.m](#), but save RF data in a new folder, modify line `cmd=['save rf_data/rf_ln',num2str(i),'.mat rf_data tstart']` to `cmd=['save rf_data_pht/rf_ln',num2str(i),'.mat rf_data tstart']`. By doing this generated rf data will be stored in folder `rf_data_pht`.
- 3. Invoke Utool and set the RF data path in Utool to the `rf_data_pht` location on the computer.
- 4. Push 'pushbutton form image' to reconstruct image from RF data.
- 5. Use the "global im" in command line to access the phantom image.

Thus, to evaluate the proposed speckle classification scheme and calculate the performance of each unsupervised classification technique, this study used the simulated B-mode ultrasound images with 100,000 scatterers and 128 RF lines. After resizing the reconstructed images to the size of 1200 x 800 pixels, they were segmented into 12x8 image patches, where each image patch had size of 100 x 100 pixels. Then statistical features for each image patch were calculated using the optimization routine explained in the method section. The following statistical features for the image patches were calculated: R, S, K and Maximum Likelihood (ML) for the Rayleigh distribution. After calculating statistical features for each

image patch, you will have three-dimensional features for each image patch which can be classified as FDS or non-FDS using unsupervised clustering techniques. To classify patches as FDS and non-FDS, the extracted features were fed into unsupervised clustering techniques with the same five pattern classification techniques explained in the method section: K-means, K-medoid, Fuzzy C-Means, Gustafson-Kessel fuzzy classifier and Gath-Geva fuzzy classifier.

In order to evaluate the performance of the proposed speckle detection technique, two different phantoms were used: short-axis echocardiographic images and left-ventricle (LV) cardiac images [102].

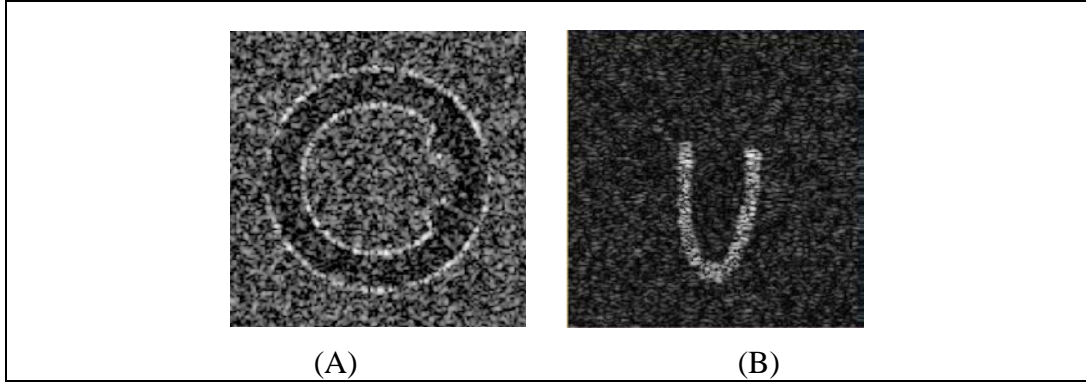


Figure 7.51: Simulation Examples: (A) short-axis echocardiographic image and (B) left ventricle echocardiographic image.

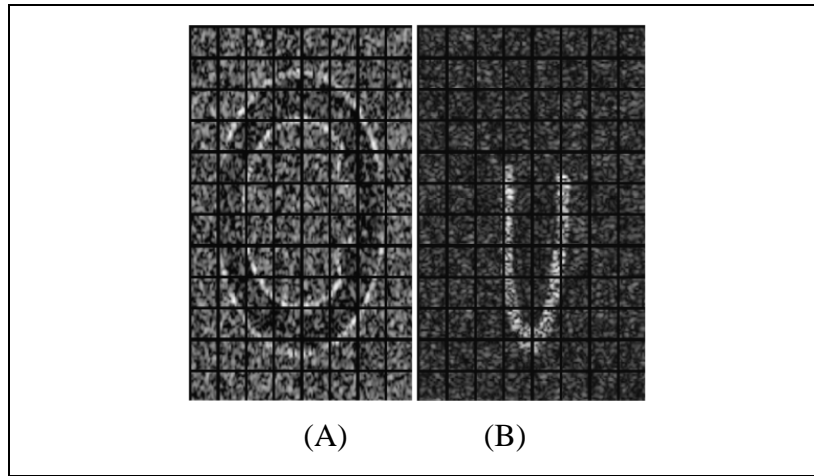


Figure 7.52: Simulation examples after image segmentation: (A) short-axis (end diastolic). (B) left ventricle phantom and patches. Images were segmented into 12x8 image patches, where each image patch had size of 100x100 pixels.

The figures below respectively show the performance of the classification methods for speckle detection, for a simulated heart. To evaluate the performance of the proposed method for speckle detection, this study used the Dice similarity index (DS), which is always between 0 and 1. DS=0 means there is no similarity and DS=1 means two regions or images are identical and have no difference between their shape and location.

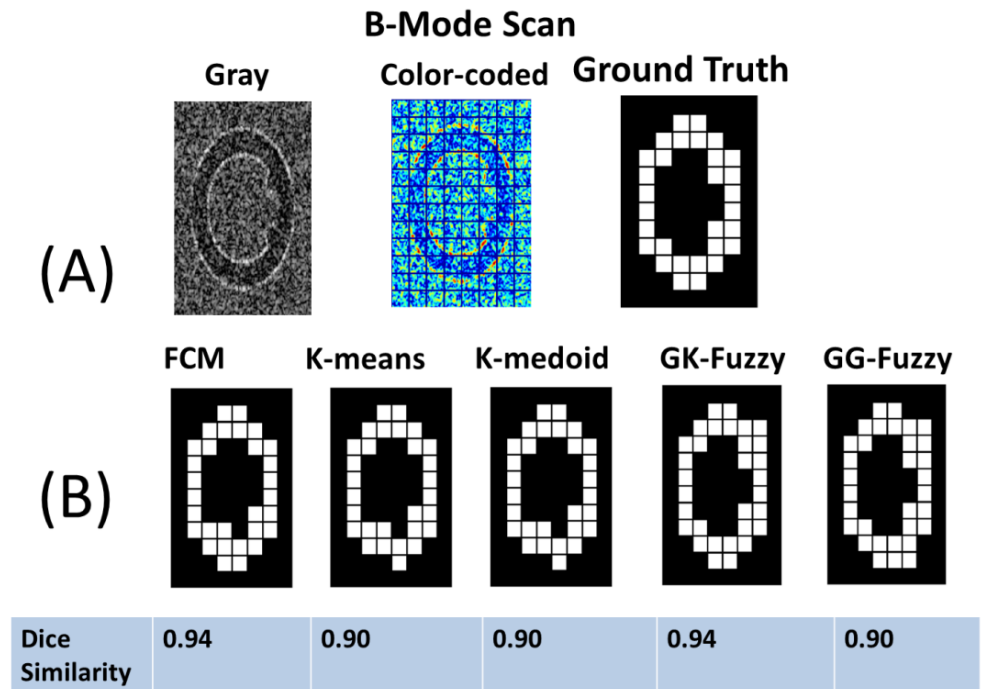


Figure 7.53: (A) Simulated ultrasound image of the heart in short axis (end diastolic). (B) Speckle detection results for five different unsupervised classifiers. Patches classified as fully developed speckles (FDS) are shown as black. Total number of patches (100x100 pixels) for the phantom image was 96. Orders for the statistical features respectively were 1,1 and 1.

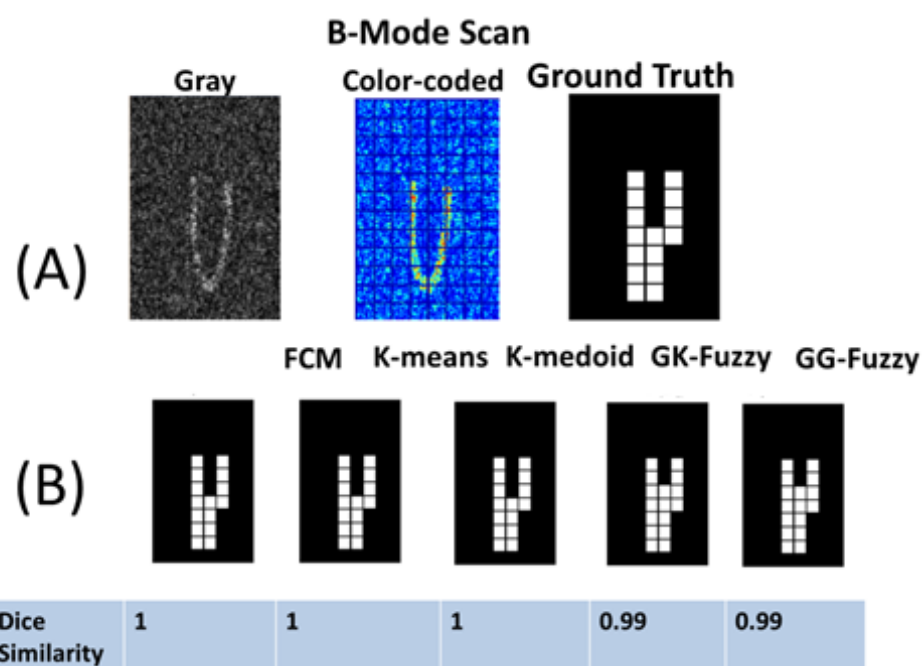


Figure 7.54: (A) Simulated ultrasound image of left ventricle (B) and speckle detection results for five different unsupervised classifiers. Patches classified as fully developed speckles (FDS) are shown as black. Total number of patches (100x100 pixels) for the phantom image was 96. Orders for statistical features respectively was 1,1 and 1.

The figures below show the results of classification for the simulated left ventricle ultrasound images in three different tissue contrasts and speckle detection results for the Gustafson-Kessel Fuzzy classifier. Patches classified as fully developed speckles (FDS) are shown as black. Average performance of the classification methods for the speckle detection of images is shown in Table 7.7.

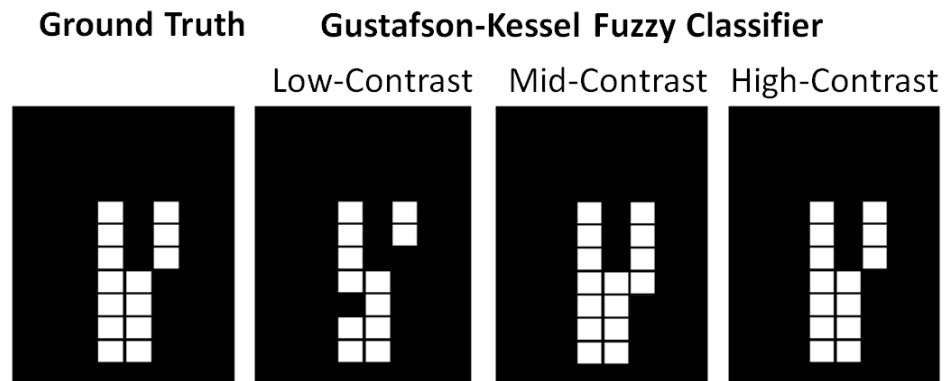
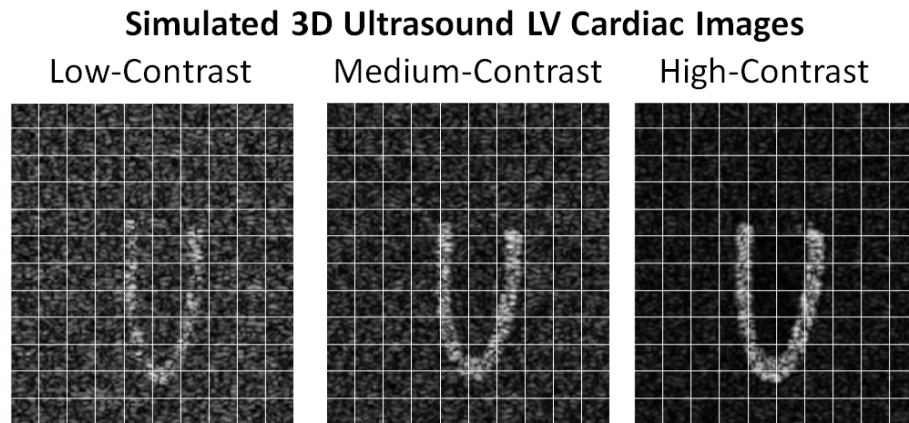
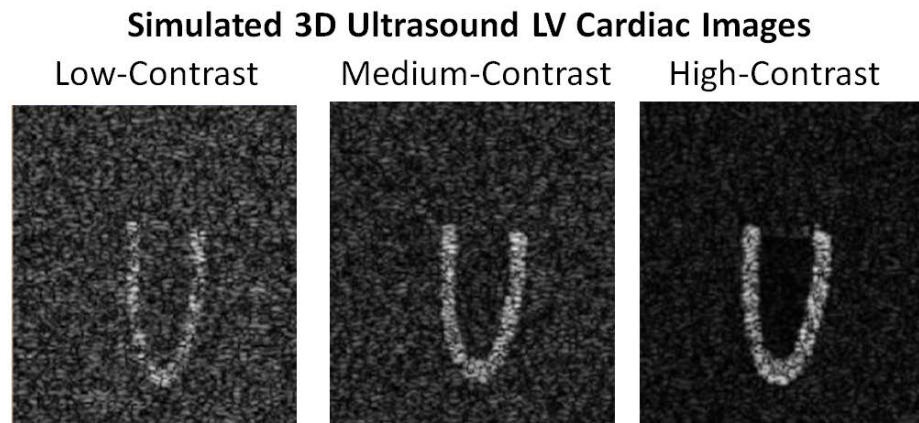


Figure 7.55: Simulated left ventricle ultrasound images in three different tissue contrasts and Speckle detection results for the Gustafson-Kessel fuzzy classifier. Patches classified as fully developed speckles (FDS) are shown as black.

Table 7.7: Average performance of the classification methods for speckle detection of simulated left ventricle ultrasound images in three different tissue contrasts and speckle detection results for the Gustafson-Kessel fuzzy classifier.

Methods	True Classifications	Miss- Classifications	Accuracy
Fuzzy C-Means (FCM)	90.5	5.5	94.27%
K-means	88.0	8.0	91.67%
K-mediod	88.0	8.0	85.42%
Gath-Geva fuzzy	92.0	4.0	95.84%
Gustafson-Kessel fuzzy (FCM+ Mahalanobis distance)	95.5	2.5	97.40%

Total number of 100x100 pixel patches for each image was 96.

Figure 7.56 shows the performance of the classification methods for speckle detection for the Gustafson-Kessel classifier, compared side-by-side with both simulation images.

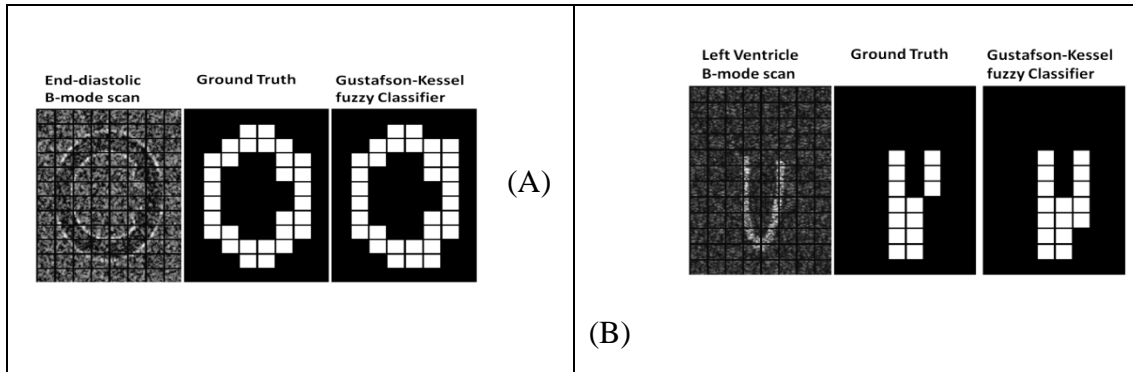


Figure 7.56: Simulation examples and speckle detection results for the Gustafson-Kessel fuzzy classifier: A) short-axis and B) left ventricle echocardiographic images. Patches classified as fully developed speckles (FDS) are shown as black.

The average performance of the classification methods for speckle detection of the images is shown in Table 7.8.

Table 7.8: Average performance of the classification methods for speckle detection of images in Figure 7.76. Total number of 100x100 pixel patches for each image was 96.

Methods	Accuracy
Fuzzy C-Means (FCM)	94.27%
K-means	91.67%
K-mediod	85.42%
Gath-Geva fuzzy	95.84%
Gustafson-Kessel fuzzy	97.40%

7.2 Real data results

The figures below respectively show the performance of the classification for both Method 1 and Method 2 in case of real data. Figure 7.57 presents the real ultrasound image of a heart phantom using 3D Ultrasound [103], and Figure 7.58 is a real ultrasound image including the radiologist observation. A 3.5-5 MHz probe was used to scan the kidney.

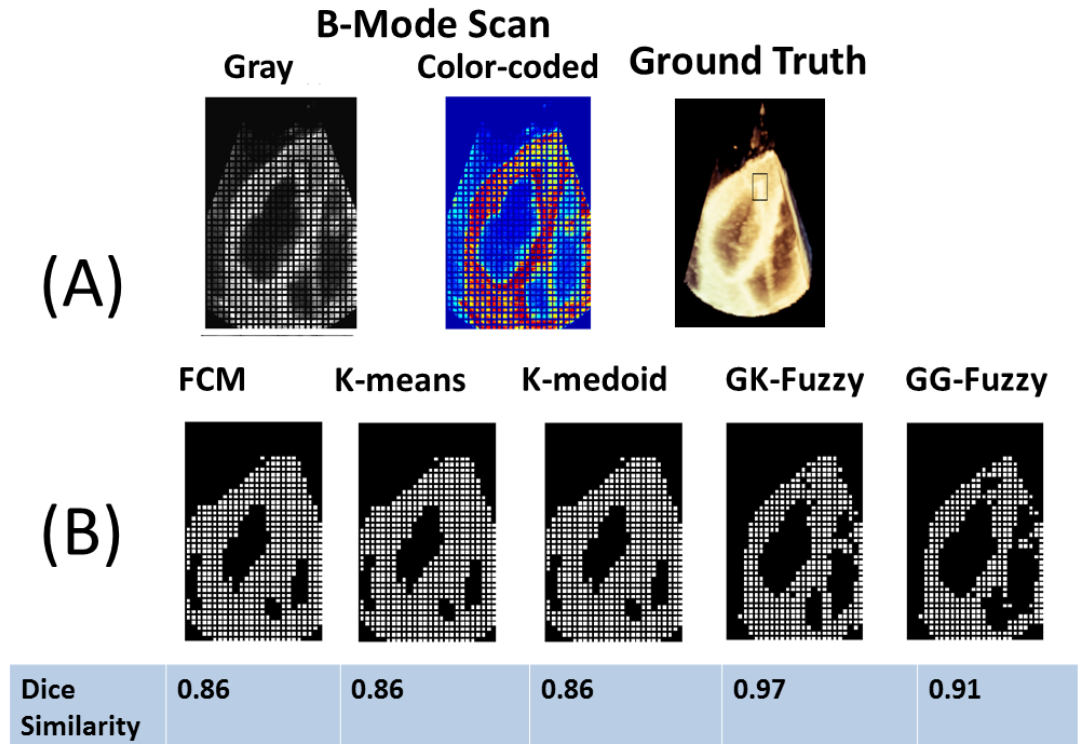


Figure 7.57: **(Results for Method 1):** (A) Ultrasound image of the right ventricle (B) and speckle detection results for five different unsupervised classifiers. Patches classified as fully developed speckles (FDS) are shown as black (Patches 30x30 pixels). Orders for statistical features respectively was 1,1 and 1.

To evaluate the performance of the proposed method for speckle detection, this study used the Dice similarity index (DS), which is always between 0 and 1. DS=0 means there is no similarity and DS=1 means two regions or images are identical and there is no difference between their shape and location.



Figure 7.58: Ultrasound image of normal right kidney (Sagital) and segmented image, including the radiologist observations (<http://www.sonoguide.com/renal.html>).

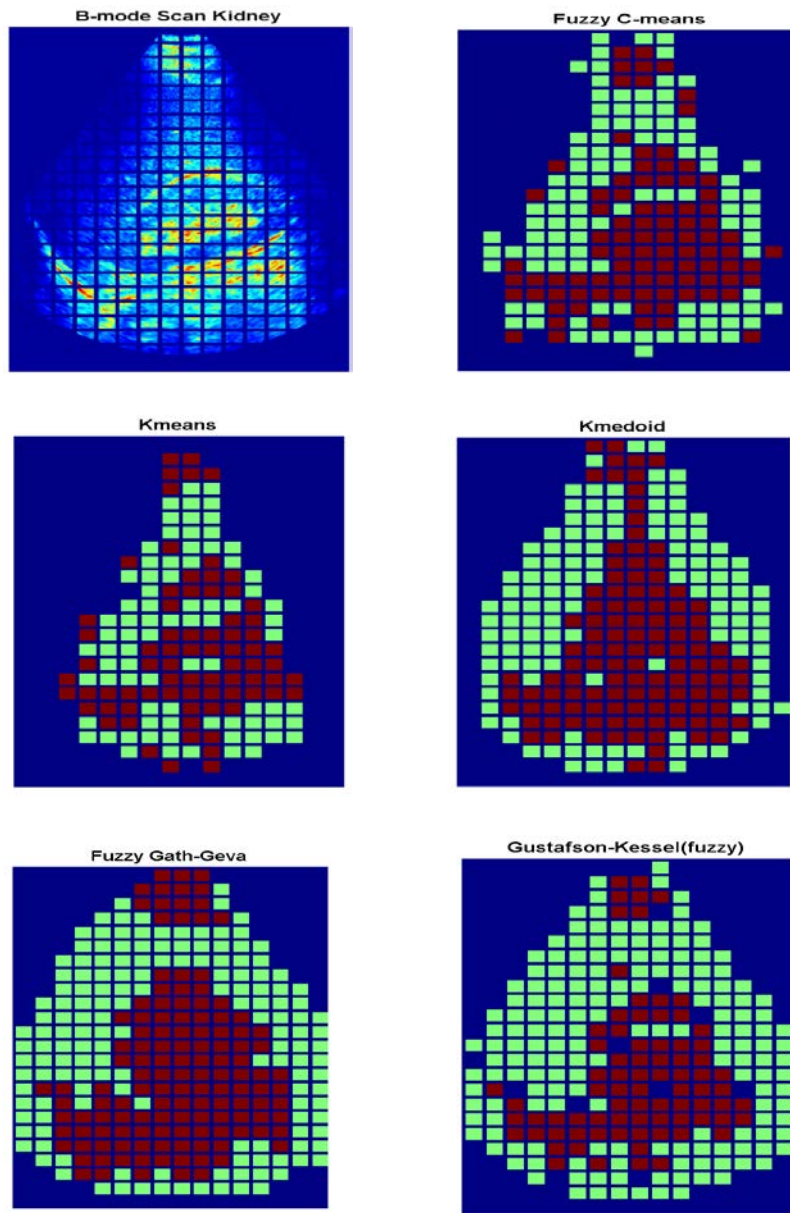


Figure 7.59: Results for recursive speckle tracking, fully automatic ultrasound image of the kidney (Sagittal).

7.3 Comparative Analysis of Method 1 and Method 2

The figures and analysis below show the performance comparison of the classification for both Method 1 and Method 2. Figure 7.60 presents the real ultrasound image of a heart phantom using 3D Ultrasound [103], and patch sizes of 100x100 were tested. The algorithm was modified to investigate the R , S , K values as one of the factors of interest. Input image pixel intensity range is [-16.87 255.92].

As mentioned in prior sections various values of v (powers of A) can be used to calculate the R , S and K [5, 81-83] for better discrimination of these features. In Method 1 the user investigates and chooses the optimum value set of v_R , v_S and v_K , depending on the criteria. For Method 1 in the analysis with the heart image in this section respectively value set of (1, 0.5, 0.25) was used for the values of v_R , v_S and v_K .

To evaluate the performance of the proposed method for speckle detection, this study used the Dice similarity index (DS), which is always between 0 and 1. DS=0 means there is no similarity and DS=1 means two regions or images are identical and there is no difference between their shape and location.

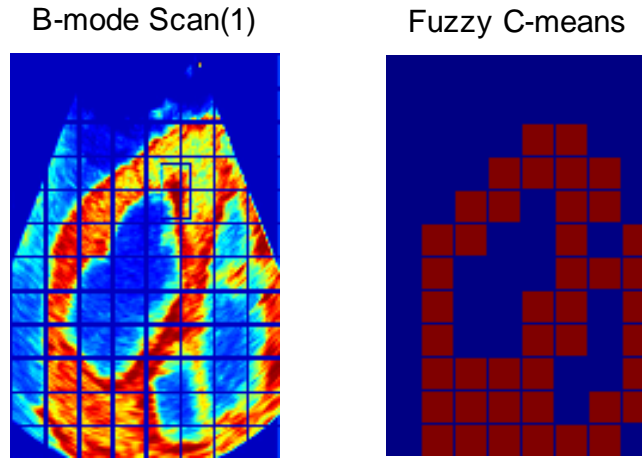


Figure 7.60: Results for heart ultrasound image with speckle detection results presented for Method 1 using Fuzzy C-means method and $v_R=1$, $v_S=0.5$, $v_K=0.25$. Blue is FDS and red patches are non-FDS.

The proposed FASD method (Method 2) was performed using the same heart image as Method 1 as the input. The input image was segmented into 100x100 size patches. The statistical features for each patch were extracted. The same statistical features as in Method 1 were used (R , S , and K). R , S , and K values are dependent of pixel intensity as well as moment orders. One may investigate the ranges for R , S , and K based on the application and the specific image, because the image intensity will vary. Due to this dependency to pixel intensity values for R , S , and K calculation, in fact the moment order values and combination are the only important factor here. After extracting features for each image patch A , the classification scheme shown in Figure 6.4 was used. This figure illustrates the proposed exhaustive search by varying moment orders for R , S and K features from 0 to L to determine the occurrence of two

classes, FDS and non-FDS. In this test the exhaustive search was done from 0 to 3 with non-uniform splitting of this range to:

0.2500 0.5000 1.0000 1.7500 2.5000 3.0000

then number of iterations (N in Figure 6.5) will be $6^3=216$.

As explained in Method 2, if a class occurs in more than 50 percent of the total iterations, the corresponding patch is classified to that class. Then the proposed ensemble classification schema shown in Figure 6.5 was used to combine classification results and get a final single classification result.

With 216 iterations there will be $N=216$ classification results and by use of schema at Figure 6.5, we can do majority voting to combine classification results and get final result. This is shown in Figure 7.61.

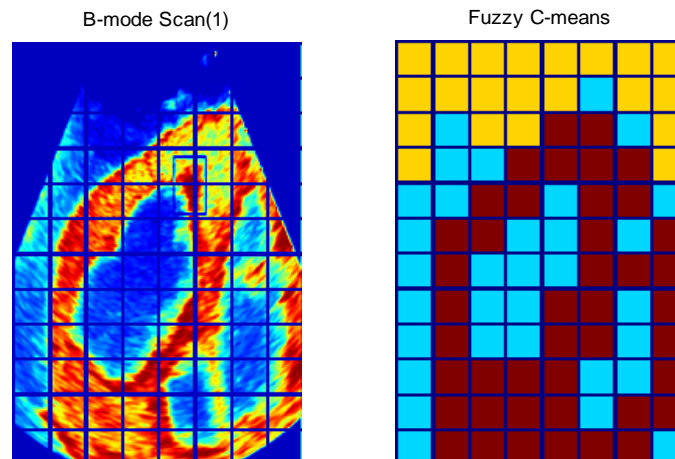


Figure 7.61: Results for heart ultrasound image with speckle detection results presented for Method 2 using Fuzzy C-means method. Light blue= FDS; Red=Non-FDS; Orange=Background.

As can be seen from both Figures 7.60 and 7.61, patch classification to FDS vs. non-FDS (performance) was the same for both methods.

The reason both methods performed and yielded in the same result is that the algorithm in Method 2 does the same job that a human does in Method 1. Method 1 is based on try and error and in Method 2 it is the algorithm that does this try and error process to find the best values for ν . The presented results in this section may be used to validate the proposed automatic Method 2.

The input image pixel intensity is in the range of [-16.87723 to 255.9292] by use of the value set of (1, 0.5, 0.25) for the ν_R , ν_S and ν_K . Using this value set for ν_R , ν_S and ν_K , the prospective R , S , and K value range will be as following:

Method 1

SNR raneg for FDS = [0 3.8608]

SNR raneg for non-FDS = [1.2641 9.6323]

Skewness raneg for FDS = [-1.46886 18.0182]

Skewness raneg for non-FDS = [-2.818 0.40166]

Kurtosis raneg for FDS = [0 64.4119]

Kurtosis raneg for non-FDS = [1.47728 18.0432]

The heatmap representation is presented in Figure 7.62.

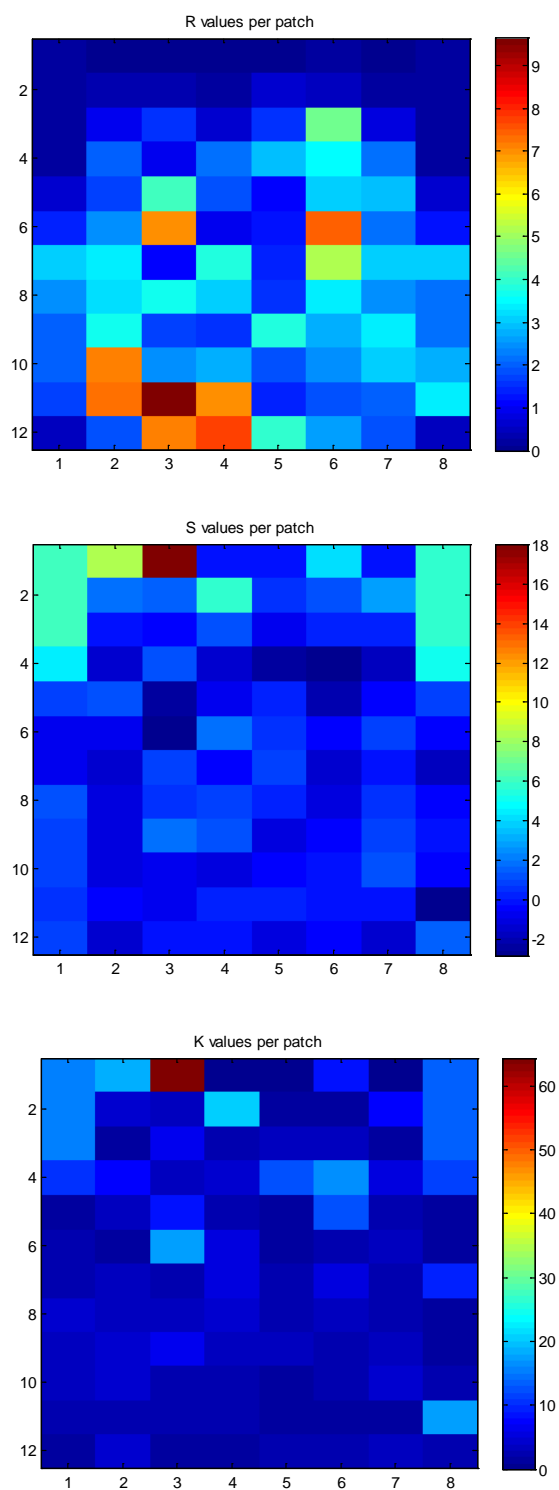


Figure 7.62: Heatmap image of the R, S, K values in Method 1 for the heart image.

In case of Method 2 algorithm in this specific case and with order values set to:

Order values used = 0.2500 0.5000 1.0000 1.7500 2.5000 3.0000

Length (order value) $^3=216$ combinations

Thus number of iterations (N in Figure 6.5) will be $6^3=216$.

Out of 216 combinations, Method 2 algorithm found following 62 combinations for moment orders that can be used for good results. As for each of these 62 different combinations we will get different R , S , and K values. Figure 7.61 and Figure 7.63 are representation of a typical R , S , and K value which can be compared with method 1. Thus one may get the same results between two methods if the optimum value for method 1 is chosen.

Here is the list of order combinations won (Voted Order Combinations):

Table 7.9: List of Voted Order Combinations for this specific test.

Voted Order Combinations		
v_R	v_S	v_K
0.25	0.25	2.5
0.25	0.5	1.75
0.25	0.5	2.5
0.25	0.5	3
0.25	1	2.5
0.25	1.75	0.25
0.25	1.75	0.5
0.25	1.75	1
0.25	1.75	2.5

0.25	2.5	3
0.5	0.25	1
0.5	0.25	3
0.5	0.5	3
0.5	1	0.25
0.5	1	3
0.5	1.75	2.5
0.5	1.75	3
0.5	2.5	1.75
0.5	3	1
0.5	3	2.5
1	0.25	0.5
1	0.25	1.75
1	0.5	0.25
1	1	0.5
1	1	1
1	1.75	0.25
1	1.75	1
1	1.75	3
1	2.5	1.75
1	2.5	2.5
1	3	1.75
1.75	0.25	0.5
1.75	0.25	1.75
1.75	0.5	0.25
1.75	1	0.5
1.75	1	1
1.75	1.75	0.5
1.75	1.75	2.5
1.75	1.75	3
1.75	2.5	0.5
1.75	2.5	1.75
1.75	2.5	3
1.75	3	1.75
2.5	0.25	0.25
2.5	0.25	0.5
2.5	0.25	1.75
2.5	0.5	1
2.5	0.5	2.5
2.5	0.5	3

2.5	1.75	0.25
2.5	1.75	1.75
2.5	3	0.25
2.5	3	1
3	0.5	0.5
3	0.5	1
3	0.5	3
3	1	2.5
3	1.75	1
3	1.75	1.75
3	2.5	0.5
3	2.5	1.75
3	3	0.5

The output result for method 2 is presented below. These results are shown for a randomly selected moment orders from winner list (62 combinations listed in the table above) and value set of (1.7500, 3.0000, 2.5000) for the v_R , v_S and v_K . The heatmap representation is presented in Figure 7.63. Input Image Intensity range is [-16.87723 255.9292].

Orders: $v_R = 1.75$, $v_S = 3$ and $v_K = 2.5$

SNR raneg for FDS = [0.39899 5.6481]

SNR raneg for non-FDS = [0.11147 1.5321]

Skewness raneg for FDS = [0.023207 5.2137]

Skewness raneg for non-FDS = [5.18348 19.3831]

Kurtosis raneg for FDS = [1.35932 27.0965]

Kurtosis raneg for non-FDS = [25.88054 354.0084]

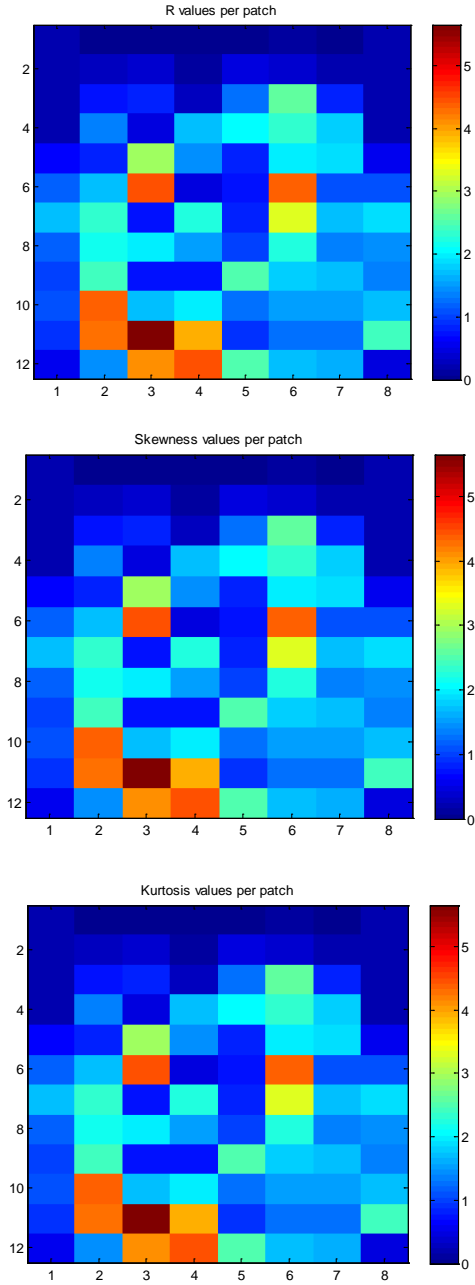


Figure 7.63: Heatmap image of the R, S, K values in Method 2.

Chapter 8

Conclusion and Outlook

8.1 Summary and Conclusion

There is currently no robust algorithm for automatic detection of speckle regions in ultrasonic images. This thesis proposed and tested such an algorithm. A novel FASD (Fully Automatic Speckle Detection) algorithm was developed based on the statistical models (Rayleigh distribution and K-distribution) which were studied and used to predict the behavior of the RF signal for different tissue types (mechanism study). Kurtosis was introduced as a new feature for speckle detection. This thesis proposes that a minimum of 3 features is sufficient to best describe speckle and improve previous results. More features will increase the computation

cost. Also optimum patch separability (best statistical features) for speckle detection purposes was achieved by choosing SNR, Skewness and Kurtosis combination for the statistical features. This thesis proposes that a minimum of 3 features is sufficient to best describe speckle and improve previous results.

Statistical features (R , S , and K) were calculated and reported. Five different unsupervised clustering techniques (K-means, K-mediod, Fuzzy C-Means, Gustafson-Kessel fuzzy and Gath-Geva fuzzy) were used for classifying ROIs in the image to speckle (FDS) and no-speckle (non-FDS) regions. As discrimination of these features are all dependent of the values of v_R , v_S and v_K . Values of v_K was investigated and reported for the first time in value sets of v_R , v_S and v_K as optimum speckle classification depends on how well one can discriminate the desired statistical features.

This research also proposes a systematic way to analyze and find the optimum value set of v_R , v_S and v_K , depending on the criteria and application which is discussed in the results section. (Appendix A: **PatchStat Toolbox**). Extremely large numbers of computer simulations, with thousands of sample data, have been done to investigate the optimum values for v_R , v_S and v_K .

Two robust classification algorithms are developed during the study:

- Method 1(Semi-Automatic Classification): A robust unsupervised speckle detection for simulation with available ground truth.

- Method 2 (Fully-Automatic Classification): An optimization search and ensemble classification scheme to fully automatic detection of speckle regions for real data with no available ground truth.

After creating ground truth data, each patch of the simulated phantom images was labeled as either FDS or non-FDS, using expert opinion. The same simulated images were then used to validate the results of the proposed methods. In the case of real data, using expert opinion would not be a possibility, thus the proposed fully automatic method, which uses voting for speckle (FDS) vs no-speckle (non-FDS) patch classification, can be used without user intervention. Also, the sensitivity of the proposed speckle detection scheme to the image patch sizes and their shapes were tested and reported.

The proposed FASD algorithm overcomes some of the limitations of speckle classification and detection as well as some of the limitations of previous methods, which are mainly based on supervised techniques and require an expert to manually label a portion of the training data. The proposed methodology is a radically different technique that is based on unsupervised clustering techniques and a combination of different statistical models for the fully automatic speckle detection. The proposed methodology can be easily implemented in clinical practice or integrated with any ultrasonic devices.

Based on the results represented, one can conclude that it is possible to automatically detect fully developed speckles by using the Rayleigh and K-distribution statistical

models and ensemble classifiers (a hybrid classification scheme based on both hard and soft boundary type unsupervised classifiers). Also overall fuzzy clustering techniques are more advantageous for speckle classification and detection purposes compared to the hard clustering techniques because objects that may exist on the boundaries between several classes are not required to fully belong to one of the clusters imaging [58]. This assertion was tested utilizing both hard and fuzzy partitioning techniques for competitive speckle detection imaging [90, 91] in simulation and images. The classification methods can be ranked as follows: 1) Gustafson-Kessel fuzzy; 2) Gath-Geva fuzzy; 3) Fuzzy C-Means; 4 and 5) K-means and K-mediod. To improve the specificity and sensitivity of the machine-learning speckle detection scheme, this study stayed focused on feature whitening [104] to increase the distance between features before applying unsupervised classification techniques [29, 30].

The performance of both the static (Method 1) and recursive (Method 2) methods were tested on statistical enough real and simulation data. A typical result of the performance evaluation and analysis for Method 1 is presented and summarized in Table 8.1 and Figure 8.2 below which was used for overall conclusion for the methods.

Table 8.1: Average speckle detection performance for Method 1.

	FCM	Kmeans	Kmediod	GK (Fuzzy)	GG (Fuzzy)
Cyst(Simulation)	0.97	0.97	0.97	0.93	0.97
Fetus (Simulation)	0.85	0.85	0.85	0.96	0.85
Heat Ventricle (Simulation)					
Heart ventricle Real)	0.94	0.91	0.85	0.97	0.95
Mean	0.905	0.8975	0.8825	0.9575	0.92
Standard Deviation	0.051235	0.047631	0.050683	0.0163936	0.04582576

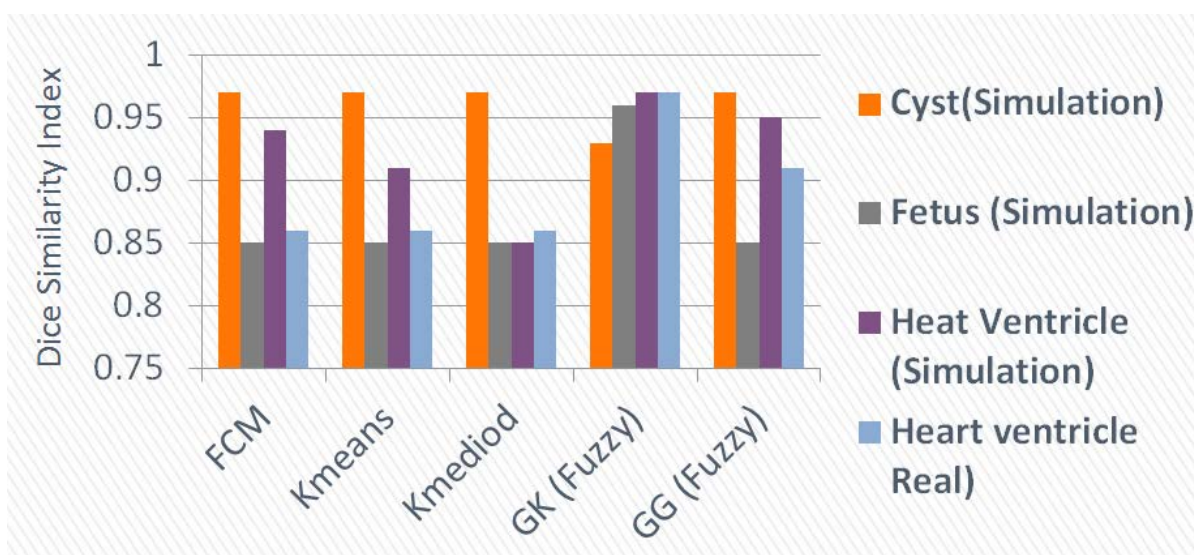


Figure 8.1: Average speckle detection performance.

8.2 Future Directions and Applications

8.2.1 Applications

Speckle affects and limits the application of automated computer-aided analysis techniques and algorithms such as edge detection, volume rendering, and 3D display, mainly due to the interference between ultrasound waves reflected from microscopic scattering through the tissue.

The results of this speckle detection work can be applied in two key applications of speckle tracking and reduction in medical imaging. Possible benefits of such applications are a better quality in sonographic images and potential improvements in the accuracy of interventional radiology procedures, such as thermal ablation of tumors (e.g. liver cancer, lung cancer), by allowing for better accuracy in needle localization relative to the therapeutic target in areas of the image with a large magnitude of speckle artifact. This in turn would further the effectiveness of ultrasound-guided procedures as opposed to CT-guided procedures that involve radiation exposure. The proposed algorithm can also be used in investigations to quantify the amount of coherent scattering, compensating for the inaccuracies caused by elevational distances between ultrasonic frames in real tissues.

Improved speckle detection can help different applications, including segmentation, sensorless 3D sonographic imaging, speckle quantitation, suppression or cancellation in biological tissues. Despeckling represents a tradeoff between noise suppression and the loss of information. This is one of the main concerns for experts

of image processing and computer aided diagnosis (CAD) that limits the existing techniques for speckle suppression. These techniques must be chosen in such a way that they do not mask or eliminate the small components and structures in the image, while reducing or removing speckle, and preserving as much information as possible. Filtering is not always the best solution to reduce speckles in the ultrasound images. Rather than filtering, speckle detection can be used to improve the performance of adaptive speckle suppression techniques as an alternative solution to preserve the anatomical information the best in the ultrasound images.

8.2.2 Future Directions

The introduced FASD method in this dissertation provides an optimization search and ensemble classification scheme for a fully automatic detection of speckle regions for real data with no available ground truth. Multiple phantom data and real data were used to generate results and test the performance of the algorithm. Future research can employ cloud computing on statistically big enough real data to further analyze the performance of Method 2. Also as the proposed model has a recursive nature, cost function in this method can be optimized along with parallel computing techniques at a much reduced computational load.

Additional future research directions include developing adaptive filtering and suppression techniques to better preserve the important features of ultrasound images. Current speckle filtering techniques represents a tradeoff between noise suppression and the loss of information. These techniques must be chosen in such a way that they do not mask or eliminate the small components and structures in the image, while

reducing or removing speckle, and preserving as much information as possible. Rather than filtering, speckle detection can be used to introduce adaptive speckle suppression techniques as an alternative solution to preserve more anatomical information in ultrasound images.

Current research studied only rectangular patches. One possible future direction could investigate other patch shapes such as cone shape or triangle shapes, etc. It is anticipated that using cone-shape image patches might improve the speckle detection performance. Further investigation on this technique could be done for routine clinical uses such as those involving needle localization in ultrasound-guided procedures including thermal ablation of tumors, vascular access, biopsies, and fetal ultrasound.

Appendices

Appendix A



speckle_detection.m



getPatchStat.m

Test Phantom Stat

```
function [R S K] =  
getPatchStat(Bmode,vR,vS,vK,rpPX,cpPX,do_whitening)  
%Bmode = imresize(Bmode,[1200 500]);  
ep = 0.00001;  
R = [];  
S = [];  
K = [];  
M = [];  
  
[m,n] = size(Bmode);  
im = Bmode;  
for i=1:(m/rpPX);  
    for j=1:(n/cpPX);  
        A = Bmode((i-1)*rpPX+1:i*rpPX,(j-  
1)*cpPX+1:j*cpPX);
```

```

        if vR~=0
            AvR          = A.^vR;
            meanAvR       = nanmean(nanmean(AvR));
            A2vR          = AvR.^2;
            meanA2vR      = nanmean(nanmean(A2vR));
            sigmaAR       = sqrt(meanA2vR-meanAvR^2);
            R(i,j)        = meanAvR/(sigmaAR+ep);
        end
        if vS~=0
            AvS          = A.^vS;
            meanAvS       = nanmean(nanmean(AvS));
            A2vS          = AvS.^2;
            meanA2vS      = nanmean(nanmean(A2vS));
            sigmaAS       = sqrt(meanA2vS-meanAvS^2);
            S(i,j)        = skewness(AvS(:)); %nanmean(nanmean((AvS-
meanAvS).^3))/(sigmaAS^3+ep);
        end
        if vK~=0
            AvK          = A.^vK;
            meanAvK       = nanmean(nanmean(AvK));
            A2vK          = AvK.^2;
            meanA2vK      = nanmean(nanmean(A2vK));
            sigmaAK       = sqrt(meanA2vK-meanAvK^2);
            K(i,j)        = kurtosis(AvK(:)); %nanmean(nanmean((AvK-
meanAvK).^4))/(sigmaAK^4+ep);
        end
        %       x1          = reshape(A,1,size(A,1)*size(A,2));
        %       [~,pci]      = mle(x1,'distribution','rayleigh');
        %       if vM~=0
        %           M(i,j)    = pci(1,1);
        %       end

        im((i-1)*rpPX+1,:)=0;
        im(:,(j-1)*cpPX+1)=0;
    end
end

if do_whitening
    R          = R/max(max(R));
    S          = S/max(max(S));
    K          = K/max(max(K));
    %       M          = M/max(max(M));
end

% figure,
% subplot(131), imagesc(R)
% subplot(132), imagesc(S)
% subplot(133), imagesc(K)
close all
figure(1), imshow(im,[]), title('Patch Location in B-mode
Image','FontSize',14)
figure(2), imagesc(real(R)), title(['R Stat', 'Order=',
num2str(vR)], 'FontSize',14)
figure(3), imagesc(real(S)), title(['Skewness Stat', 'Order=',
num2str(vS)], 'FontSize',14)

```

```
figure(4), imagesc(real(K)), title(['Kurtosis Stat', 'Order=',
num2str(vK)], 'FontSize',14)
```

```
K11          = [];
R11          = [];
S11          = [];
M11          = [];
sR           = size(R);
sS           = size(S);
sK           = size(K);
% sM         = size(M);
if ~isempty(R),R11 = reshape(R,sR(1)*sR(2),1);end
if ~isempty(S),S11 = reshape(S,sS(1)*sS(2),1);end
if ~isempty(K),K11 = reshape(K,sK(1)*sK(2),1);end
% if ~isempty(M),M11 = reshape(M,sM(1)*sM(2),1);end

w            =([R11 S11 K11  ]); %M11]);

figure(5), scatter3(w(:,1),w(:,2),w(:,3),200,'filled'),
xlabel('R Stat Feature','FontSize',14)
ylabel('Skewness Stat Feature','FontSize',14)
zlabel('Kurtosis Stat Feature','FontSize',14)
```

Appendix B

UTool Source Code

The routine *field.m* initializes the field system, and should be modified to point to the directory holding the Field II code and m-files. The routine *mk_pht.m* is then called to make the file for the scatterers in the phantom. The script *sim_img.m* is then called. Here the field simulation is performed and the data is stored in RF-files; one for each RF-line done. The files are stored in the sub-directory *rf_data*. The data is then subsequently processed by *make_image.m* to yield the image. The data for the scatterers are read from the file *pht_data.mat*.

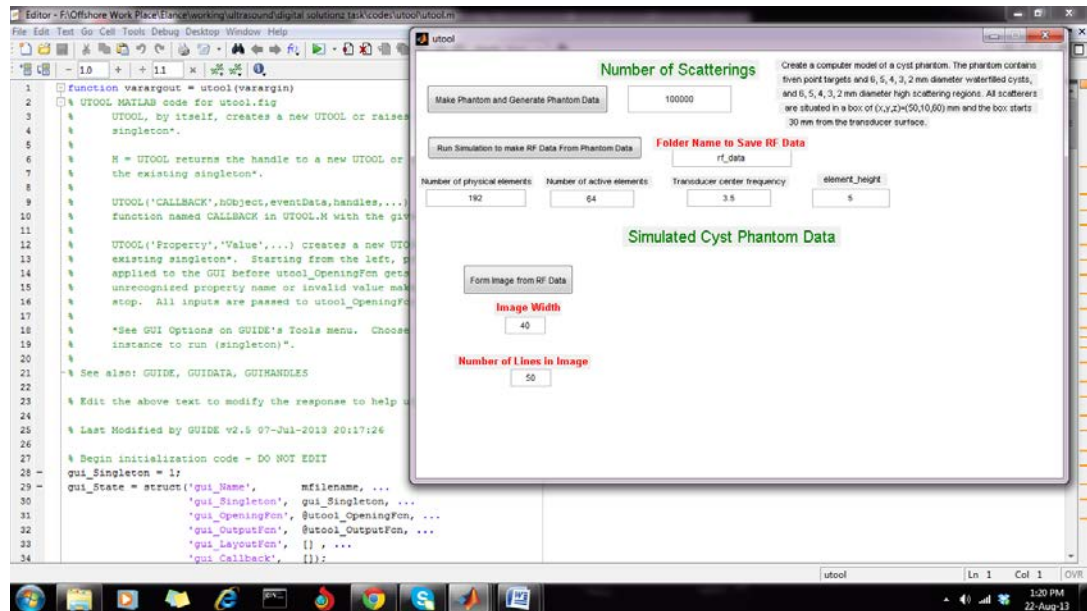


Figure B.1: Running 'Utool' in Matlab.

In order to generate the phantoms, the respective properties of the phantoms must be physically coded in the file `cyst_pht.m`. Here, the code for generating 6, 5, 4, 3, and 2 mm phantoms is already present, which can be used as desired.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      UTOOL SOURCE CODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function varargout = utool(varargin)

% UTOOL MATLAB code for utool.fig
%     UTOOL, by itself, creates a new UTOOL or raises the existing
%     singleton*.
%
%     H = UTOOL returns the handle to a new UTOOL or the handle to
%     the existing singleton*.
%
%     UTOOL('CALLBACK',hObject,eventData,handles,...) calls the
local
```

```

%      function named CALLBACK in UTOOL.M with the given input
arguments.
%
%      UTOOL('Property','Value',...) creates a new UTOOL or raises
the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before utool_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to utool_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help utool

% Last Modified by GUIDE v2.5 14-Jan-2013 23:00:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @utool_OpeningFcn, ...
                  'gui_OutputFcn',    @utool_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin[77])
    gui_State.gui_Callback = str2func(varargin[77]);
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before utool is made visible.
function utool_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to utool (see VARARGIN)

% Choose default command line output for utool
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes utool wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = utool_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout[77] = handles.output;

function Ns_Callback(hObject, eventdata, handles)
% hObject handle to Ns (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ns as text
% str2double(get(hObject,'String')) returns contents of Ns as
a double

% --- Executes during object creation, after setting all properties.
function Ns_CreateFcn(hObject, eventdata, handles)
% hObject handle to Ns (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Fimg.
function Fimg_Callback(hObject, eventdata, handles)
% hObject handle to Fimg (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global im
im=formImage(handles);

```



```

function img_width_Callback(hObject, eventdata, handles)
% hObject      handle to img_width (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of img_width as text
%        str2double(get(hObject,'String')) returns contents of
img_width as a double

% --- Executes during object creation, after setting all properties.
function img_width_CreateFcn(hObject, eventdata, handles)
% hObject      handle to img_width (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Nlines_Callback(hObject, eventdata, handles)
% hObject      handle to Nlines (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nlines as text
%        str2double(get(hObject,'String')) returns contents of
Nlines as a double

% --- Executes during object creation, after setting all properties.
function Nlines_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Nlines (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in sim.
function sim_Callback(hObject, eventdata, handles)
% hObject      handle to sim (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
saveLocation=get(handles.saveLocation,'String');
folder_dir=[cd '\',saveLocation];
try
    mkdir(folder_dir)
end
simulateImage (handles,saveLocation)

function saveLocation_Callback(hObject, eventdata, handles)
% hObject      handle to saveLocation (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of saveLocation as
text
%         str2double(get(hObject,'String')) returns contents of
saveLocation as a double

% --- Executes during object creation, after setting all properties.
function saveLocation_CreateFcn(hObject, eventdata, handles)
% hObject      handle to saveLocation (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Nelements_Callback(hObject, eventdata, handles)
% hObject      handle to Nelements (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nelements as text
%         str2double(get(hObject,'String')) returns contents of
Nelements as a double

% --- Executes during object creation, after setting all properties.
function Nelements_CreateFcn(hObject, eventdata, handles)

```

```
% hObject      handle to Nelements (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Nact_Callback(hObject, eventdata, handles)
% hObject      handle to Nact (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of Nact as text
%        str2double(get(hObject,'String')) returns contents of Nact
as a double
```

```
% --- Executes during object creation, after setting all properties.
function Nact_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Nact (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Tcf_Callback(hObject, eventdata, handles)
% hObject      handle to Tcf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of Tcf as text
%        str2double(get(hObject,'String')) returns contents of Tcf
as a double
```

```
% --- Executes during object creation, after setting all properties.
function Tcf_CreateFcn(hObject, eventdata, handles)
```

```

% hObject      handle to Tcf (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function element_height_Callback(hObject, eventdata, handles)
% hObject      handle to element_height (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of element_height as
text
%         str2double(get(hObject,'String')) returns contents of
element_height as a double

% --- Executes during object creation, after setting all properties.
function element_height_CreateFcn(hObject, eventdata, handles)
% hObject      handle to element_height (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in makePhantom.
function makePhantom_Callback(hObject, eventdata, handles)
% hObject      handle to makePhantom (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
Ns=get(handles.Ns,'String');
[phantom_positions, phantom_amplitudes] = cyst_pht(Ns);
size(phantom_positions)
%save(['pht_data_' data], 'phantom_positions', 'phantom_amplitudes')
save pht_data.mat phantom_positions phantom_amplitudes

```

```

function [positions, amp] = cyst_phantom (N)
% Calling: [positions, amp] = cyst_phantom (N);
%
% Parameters:  N - Number of scatterers in the phantom
%
% Output:      positions - Positions of the scatterers.
%              amp       - amplitude of the scatterers.
%
% Version 2.2, April 2, 1998 by Joergen Arendt Jensen

function [positions, amp] = cyst_phantom (N)
if ischar(N)
N=str2double(N);
end
x_size = 50/1000;    % Width of phantom [mm]
y_size = 10/1000;    % Transverse width of phantom [mm]
z_size = 60/1000;    % Height of phantom [mm]
z_start = 30/1000;   % Start of phantom surface [mm];

% Create the general scatterers

x = (rand (N,1)-0.5)*x_size;
y = (rand (N,1)-0.5)*y_size;
z = rand (N,1)*z_size + z_start;

% Generate the amplitudes with a Gaussian distribution

amp=randn(N,1);

% Make the cyst and set the amplitudes to zero inside

% 6 mm cyst
r=6/2/1000;          % Radius of cyst [mm]
xc=10/1000;          % Place of cyst [mm]
zc=10/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);
amp = amp .* (1-inside);

% 5 mm cyst
r=5/2/1000;          % Radius of cyst [mm]
zc=20/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);
amp = amp .* (1-inside);

% 4 mm cyst
r=4/2/1000;          % Radius of cyst [mm]
zc=30/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);
amp = amp .* (1-inside);

```

```

% 3 mm cyst
r=3/2/1000;          % Radius of cyst [mm]
zc=40/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);
amp = amp .* (1-inside);

% 2 mm cyst
r=2/2/1000;          % Radius of cyst [mm]
zc=50/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2);
amp = amp .* (1-inside);

% Make the high scattering region and set the amplitudes to 10
times inside

% 6 mm region
r=5/2/1000;          % Radius of cyst [mm]
xc=-5/1000;          % Place of cyst [mm]
zc=50/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;
amp = amp .* (1-inside) + 10*amp .* inside;

% 5 mm region
r=4/2/1000;          % Radius of cyst [mm]
zc=40/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;
amp = amp .* (1-inside) + 10*amp .* inside;

% 4 mm region
r=3/2/1000;          % Radius of cyst [mm]
zc=30/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;
amp = amp .* (1-inside) + 10*amp .* inside;

% 3 mm region
r=2/2/1000;          % Radius of cyst [mm]
zc=20/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;
amp = amp .* (1-inside) + 10*amp .* inside;

% 2 mm region
r=1/2/1000;          % Radius of cyst [mm]
zc=10/1000+z_start;

inside = ( ((x-xc).^2 + (z-zc).^2) < r^2) ;

```

```

amp = amp .* (1-inside) + 10*amp .* inside;

% Place the point scatterers in the phantom

for i=N-5:N
    x(i) = -15/1000;
    y(i) = 0;
    z(i) = z_start + (10+5*10)/1000 + (i-N)*10/1000;
    amp(i) = 20;
end

% Return the variables

positions=[x y z];

function simulateImage(handles, folderName)
% Example of use of the new Field II program running under
% Matlab.
%
% This example shows how a linear array B-mode system scans an
image
%
% This script assumes that the field_init procedure has been called
% Here the field simulation is performed and the data is stored
% in rf-files; one for each rf-line done. The data must then
% subsequently be processed to yield the image. The data for the
% scatterers are read from the file pht_data.mat, so that the
procedure
% can be started again or run for a number of workstations.
%
% Example by Joergen Arendt Jensen and Peter Munk,
% Version 1.2, August 14, 1998, JAJ.

% Ver. 1.1: 1/4-98: Procedure xdc_focus_center inserted to use the
new
%                               focusing scheme for the Field II program
% Ver. 2.0: 13/8 2007: Parallel version that checks whether the
simulation
%                               of a line has been made before, which makes
it possible
%                               to run the code in parallel on multiple
workstations.

% Generate the transducer apertures for send and receive

%f0=3.5e6;                % Transducer center frequency [Hz]
f0=str2double(get(handles.Tcf, 'String'))*1e6;
fs=100e6;                % Sampling frequency [Hz]
c=1540;                  % Speed of sound [m/s]
lambda=c/f0;             % Wavelength [m]
width=lambda;            % Width of element

```

```

%element_height=5/1000; % Height of element [m]
element_height=str2double(get(handles.element_height,'String'))/1000
;
kerf=0.05/1000; % Kerf [m]
focus=[0 0 70]/1000; % Fixed focal point [m]
%N_elements=192; % Number of physical elements
N_elements=str2double(get(handles.Nelements,'String'));
%N_active=64; % Number of active elements
N_active=str2double(get(handles.Nact,'String'));

% Set the sampling frequency

set_sampling(fs);

% Generate aperture for emission

xmit_aperture = xdc_linear_array (N_elements, width, element_height,
kerf, 1, 10,focus);

% Set the impulse response and excitation of the xmit aperture

impulse_response=sin(2*pi*f0*(0:1/fs:2/f0));
impulse_response=impulse_response.*hanning(max(size(impulse_response
)))';
xdc_impulse (xmit_aperture, impulse_response);

excitation=sin(2*pi*f0*(0:1/fs:2/f0));
xdc_excitation (xmit_aperture, excitation);

% Generate aperture for reception

receive_aperture = xdc_linear_array (N_elements, width,
element_height, kerf, 1, 10,focus);

% Set the impulse response for the receive aperture

xdc_impulse (receive_aperture, impulse_response);

% Load the computer phantom

if ~exist('pht_data.mat')
    disp('Scatterer positions should be made by the script mk_pht')
    disp('before this script can be run')
    return
else
    load pht_data
end

% Set the different focal zones for reception

focal_zones=[30:20:200]'/1000;
Nf=max(size(focal_zones));

```



```

focus_times=(focal_zones-10/1000)/1540;
z_focus=60/1000;           % Transmit focus

% Set the apodization
apo=hanning(N_active)';

% Do linear array imaging

no_lines=50;                % Number of lines in image
image_width=40/1000;        % Size of image sector
d_x=image_width/no_lines;   % Increment for image

% Do imaging line by line

for i=[1:no_lines]

    % Test if the file for the line exist.
    % Skip the simulation, if the line exists and
    % go to the next line. Else make the simulation

    %file_name=['rf_data/rf_ln',num2str(i),'.mat'];
    file_name=[folderName, '/rf_ln',num2str(i),'.mat'];

    if ~exist(file_name)

        % Save a file to reserve the calculation

        %cmd=['save rf_data/rf_ln',num2str(i),'.mat i'];
        cmd=['save ', folderName, '/rf_ln',num2str(i),'.mat i'];
        eval(cmd);

        disp(['Now making line ',num2str(i)])

        % The the imaging direction

        x= -image_width/2 +(i-1)*d_x;

        % Set the focus for this direction with the proper reference
        point

        xdc_center_focus (xmit_aperture, [x 0 0]);
        xdc_focus (xmit_aperture, 0, [x 0 z_focus]);
        xdc_center_focus (receive_aperture, [x 0 0]);
        xdc_focus (receive_aperture, focus_times, [x*ones(Nf,1),
        zeros(Nf,1), focal_zones]);

        % Calculate the apodization

        N_pre = round(x/(width+kerf) + N_elements/2 - N_active/2);
        N_post = N_elements - N_pre - N_active;
        apo_vector=[zeros(1,N_pre) apo zeros(1,N_post)];
        xdc_apodization (xmit_aperture, 0, apo_vector);

```

```

xdc_apodization (receive_aperture, 0, apo_vector);

% Calculate the received response

[rf_data, tstart]=calc_scatt(xmit_aperture, receive_aperture,
phantom_positions, phantom_amplitudes);

% Store the result

%cmd=['save rf_data/rf_ln',num2str(i),'.mat rf_data tstart'];
cmd=['save ', folderName, '/rf_ln',num2str(i),'.mat rf_data
tstart'];

disp(cmd)
eval(cmd);
else
disp(['Line ',num2str(i),' is being made by another machine.'])
end
end
end

```

function im=formImage(handles)

```

% Compress the data to show 60 dB of
% dynamic range for the cyst phantom image
%
% version 1.3 by Joergen Arendt Jensen, April 1, 1998.
% version 1.4 by Joergen Arendt Jensen, August 13, 2007.
% Clibrated 60 dB display made

f0=3.5e6; % Transducer center frequency [Hz]
fs=100e6; % Sampling frequency [Hz]
c=1540; % Speed of sound [m/s]
% Number of lines in image
no_lines=str2double(get(handles.Nlines,'String')); %50;

% Size of image sector
image_width=str2double(get(handles.img_width,'String'))/1000;
%40/1000;

d_x=image_width/no_lines; % Increment for image

% Read the data and adjust it in time

min_sample=0;
for i=1:no_lines

% Load the result

cmd=['load rf_data/rf_ln',num2str(i),'.mat'];
disp(cmd)
eval(cmd)

```

```

% Find the envelope

rf_env=abs(hilbert([zeros(round(tstart*fs-min_sample),1);
rf_data]));
env(1:max(size(rf_env)),i)=rf_env;
end

% Do logarithmic compression

D=10; % Sampling frequency decimation factor

disp('Finding the envelope')
log_env=env(1:D:max(size(env)),:)/max(max(env));
log_env=20*log10(log_env);
log_env=127/60*(log_env+60);

% Make an interpolated image

disp('Doing interpolation')
ID=20;
[n,m]=size(log_env);
new_env=zeros(n,m*ID);
for i=1:n
    new_env(i,:)=abs(interp(log_env(i,:),ID));
end
[n,m]=size(new_env);

fn=fs/D;
%figure
%handles.axes1
axes(handles.axes1)
%clf

xr=((1:(ID*no_lines-1))*d_x/ID-no_lines*d_x/2)*1000;
yr=((1:n)/fn+min_sample/fs)*1540/2*1000;

range=[400: 1190];
yr1=yr(range);
im=new_env(range,1:end-10);
M1=ceil((xr(end)-xr(1))/10); xr0=floor(xr(1):M1:xr(end));
M2=ceil((yr1(end)-yr1(1))/10); yr0=floor(yr1(1):M2:yr1(end));

image(((1:(ID*no_lines-1))*d_x/ID-
no_lines*d_x/2)*1000,((1:n)/fn+min_sample/fs)*1540/2*1000,new_env)

xlabel('Lateral distance [mm]')
ylabel('Axial distance [mm]')
colormap(gray(127))
axis('image')
axis([-20 20 35 90])

```

Appendix C

The routine *field.m* initializes the field system, and should be modified to point to the directory holding the Field II code and m-files. The routine *make_scatteres.m* is then called to make the file for the scatterers in the phantom. The file *kidney_cut.bmp* holds the scatterer map of the kidney. The script *sim_kidney.m* is then called. Here the field simulation is performed and the data is stored in RF-files; one for each RF-line done. The data for the scatterers is read from the file *pht_data.mat*, so that the procedure can be started again or run for a number of workstations. The data must then subsequently be processed to do the polar to rectangular mapping to yield the image. This is done by the routine *make_polar*, which creates an image with a dynamic range of 50dB[105].

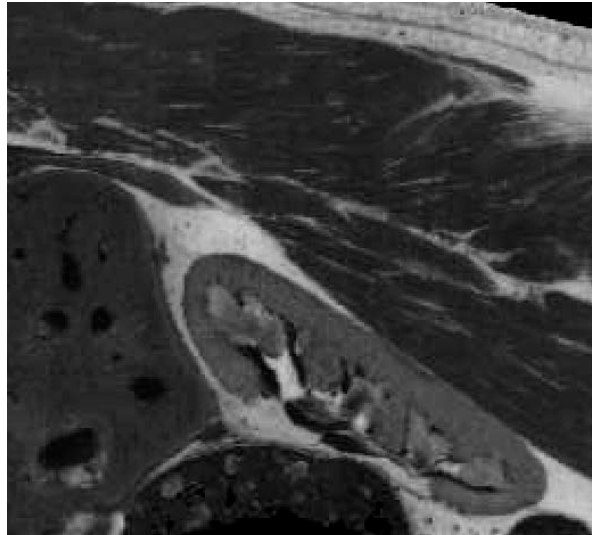


Figure C.1: A sample Kidney bitmap [105].

```
% Creates a phantom for a liver from a MR scan of the liver.
% The size of the image is 100 x 100 (width and depth)
% and the thickness is 15 mm.
% The phantom starts 2 mm from the transducer surface.
%
% Ver. 1.1, March 29, 2000, Jørgen Arendt Jensen

function [positions, amp] = human_kidney_phantom (N)

% Load the bitmap image

[liv_kid, MAP]=bmpread('kidney_cut.bmp');

% Define image coordinates

liv_kid=liv_kid';
[Nl, Ml]=size(liv_kid);

x_size = 100/1000 ;      % Size in x-direction [m]
dx=x_size/Nl;           % Sampling interval in x direction [m]
z_size = 100/1000 ;      % Size in z-direction [m]
dz=z_size/Ml;           % Sampling interval in z direction [m]
y_size = 15/1000;        % Size in y-direction [m]
theta = 35/180*pi;       % Rotation of the scatterers [rad]
theta = 0;
z_start = 2/1000;

% Calculate position data

x0 = rand(N,1);
x = (x0-0.5)* x_size;
```



```

f0=7e6; % Transducer center frequency [Hz]
fs=100e6; % Sampling frequency [Hz]
c=1540; % Speed of sound [m/s]
lambda=c/f0; % Wavelength [m]
width=lambda/2; % Width of element
element_height=5/1000; % Height of element [m]
kerf=lambda/10; % Kerf [m]
focus=[0 0 90]/1000; % Fixed focal point [m]
N_elements=128; % Number of physical elements

% Set the sampling frequency

set_sampling(fs);
set_field('show_times', 5)

% Generate aperture for emission

xmit_aperture = xdc_linear_array (N_elements, width, element_height,
kerf, 1, 5, focus);

% Set the impulse response and excitation of the xmit aperture

impulse_response=sin(2*pi*f0*(0:1/fs:2/f0));
impulse_response=impulse_response.*hanning(max(size(impulse_response
)))';
xdc_impulse (xmit_aperture, impulse_response);

excitation=sin(2*pi*f0*(0:1/fs:2/f0));
xdc_excitation (xmit_aperture, excitation);

% Generate aperture for reception

receive_aperture = xdc_linear_array (N_elements, width,
element_height, kerf, 1, 5, focus);

% Set the impulse response for the receive aperture

xdc_impulse (receive_aperture, impulse_response);

% Load the computer phantom

load pht_data

% Set the different focal zones for reception

focal_zones=[5:1:150]'/1000;
Nf=max(size(focal_zones));
focus_times=(focal_zones-10/1000)/1540;
z_focus=60/1000; % Transmit focus

% Set the apodization

```

```

apo=hanning(N_elements)';
xdc_apodization (xmit_aperture, 0, apo);
xdc_apodization (receive_aperture, 0, apo);

% Do phased array imaging

no_lines=128; % Number of lines in image
image_width=90/180*pi; % Size of image sector [rad]
dtheta=image_width/no_lines; % Increment for image

% Do imaging line by line

for i=1:128

    if ~exist(['rf_data/rf_ln',num2str(i),'.mat'])

        cmd=['save rf_data/rf_ln',num2str(i),'.mat i']
        eval(cmd)

        % Set the focus for this direction

        theta= (i-1-no_lines/2)*dtheta;
        xdc_focus (xmit_aperture, 0, [z_focus*sin(theta) 0
z_focus*cos(theta)]);
        xdc_focus (receive_aperture, focus_times,
[focal_zones*sin(theta) zeros(max(size(focal_zones)),1)
focal_zones*cos(theta)]);

        % Calculate the received response

        [rf_data, tstart]=calc_scatt(xmit_aperture, receive_aperture,
phantom_positions, phantom_amplitudes);

        % Store the result

        cmd=['save rf_data/rf_ln',num2str(i),'.mat rf_data tstart']
        eval(cmd)
    end

end

% Free space for apertures

xdc_free (xmit_aperture)
xdc_free (receive_aperture)

```


List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
CAD	Computer Aided Diagnosis
dB	Decibels
DS	Dice Similarity
FASD	Fully Automatic Speckle Detection
FDS	Fully Developed Speckles
GG	Gath-Geva
GK	Gustafson-Kessel
GUI	Graphical User Interface
SD	Standard Deviation
US	ultrasound
ML	Maximum Likelihood
PSF	Point Spread Function
PDF	Probability Distribution Function
R	Ratio
S	Skewness
K	Kurtosis
RF	Radio Frequency
ROI	Region Of Interest

References

1. Burckhardt, C.B., *Speckle in ultrasound B-mode scans*. IEEE Trans. Sonics Ultrasonics, 1978. **SU-25**(1): p. 1–6.
2. R.F. Wagner, S.W.S., J.M. Sandrik, and H. Lopez, *Statistics of speckle in ultrasound B-scans*. IEEE Trans. Sonics Ultrasonics, 1983. **30**: p. 156–163.
3. Goodman, J.W., *Some fundamental properties of speckle*. J. Opt. Soc. Am., 1976. **66**(11): p. 1145–1149.
4. Yu, Y. and S.T. Acton, *Speckle reducing anisotropic diffusion*. IEEE Trans Image Process, 2002. **11**(11): p. 1260-70.
5. R.W. Prager, A.H.G., G.M. Treece, and L. Berman, *Speckle detection in ultrasound images using first order statistics*. GUED/F-INFENG/TR 415, University of Cambridge, Dept. of Engineering, July 2002: p. 1–17.
6. Loizou, C.P., et al., *Comparative evaluation of despeckle filtering in ultrasound imaging of the carotid artery*. IEEE Trans Ultrason Ferroelectr Freq Control, 2005. **52**(10): p. 1653-69.
7. Lee, J.S., *Speckle analysis and smoothing of synthetic aperture radar images*. Comp. Graphics Image Process., 1981. **17**: p. 4–32.
8. L. Busse, T.R.C., and J.R. Fienup, *A model based approach to improve the performance of the geometric filtering speckle reduction algorithm*. IEEE Ultrasonic Symp., 1995. **2**: p. 1353–1356.
9. D.T. Kuan, A.A.S., T.C. Strand, and P. Chavel, *Adaptive restoration of images with speckle*. IEEE Trans. Acoust., 1987. **ASSP-35**: p. 373–383.

10. Lasaygues, P., et al., *Progress towards in vitro quantitative imaging of human femur using compound quantitative ultrasonic tomography*. Phys Med Biol, 2005. **50**(11): p. 2633-49.
11. V.S. Frost, J.A.S., K.S. Shanmungan, and J.C. Holtzman, *model for radar images and its application for adaptive digital filtering of multiplicative noise*. IEEE Trans. Pattern Anal. Mach. Intell., 1982. **4**(2): p. 157-165.
12. Lee, J.S., *Digital image enhancement and noise filtering by using local statistics*. IEEE Trans. Pattern Anal. Mach. Intell., 1980. **PAMI-2**(2): p. 165-168.
13. Lee, J.S., *Refined filtering of image noise using local statistics*. Comput. Graphics Image Process, 1981. **15**: p. 380-389.
14. Eltoft, S.S.a.T., *Homomorphic wavelet based-statistical despeckling of SAR images*. IEEE Trans. Geosci. Remote Sens., 2004. **42**(4): p. 711-721.
15. Saniie, J., T. Wang, and N.M. Bilgutay, *Analysis of homomorphic processing for ultrasonic grain signal characterization*. IEEE Trans Ultrason Ferroelectr Freq Control, 1989. **36**(3): p. 365-75.
16. S. Jin, Y.W., and J. Hiller, *An adaptive non-linear diffusion algorithm for filtering medical images*. IEEE Trans. Inf. Technol. Biomed., December 2000. **4**(4): p. 298-305.
17. Weickert, J., B.H. Romeny, and M.A. Viergever, *Efficient and reliable schemes for nonlinear diffusion filtering*. IEEE Trans Image Process, 1998. **7**(3): p. 398-410.
18. Rougon, N. and F. Preteux, *Controlled anisotropic diffusion*, in *Conference on Nonlinear Image Processing VI, IS&T/SPIE Symposium on Electronic Imaging, Science and Technology*. 1995: San Jose, CA. p. 1-12.
19. Black, M.J., et al., *Robust anisotropic diffusion*. IEEE Trans Image Process, 1998. **7**(3): p. 421-32.
20. Rerona, P. and J. Malik, *Scale-space and edge detection using anisotropic diffusion*. IEEE Trans. Pattern Anal. Mach. Intell., 1990. **12**(7): p. 629-639.
21. Abd-Elmoniem, K.Z., A.B. Youssef, and Y.M. Kadah, *Real-time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion*. IEEE Trans Biomed Eng, 2002. **49**(9): p. 997-1014.
22. Cherkassky, S.Z.a.V., *Image denoising using wavelet thresholding and model selection*. Proceedings of the IEEE International Conference on Image Processing, Vancouver, BC, Canada, November 2000: p. 1-4.
23. C. P. Loizou, C.S.P., M. Pantziaris, T. Tyllis, A. Nicolaides, *Quality evaluation of ultrasound imaging in the carotid artery based on normalization and speckle reduction filtering*. Med Biol Eng Comput, 2006. **44**(5): p. 414-426.
24. Bamber, J.C. and J.V. Phelps, *Real-time implementation of coherent speckle suppression in B-scan images*. Ultrasonics, 1991. **29**(3): p. 218-24.
25. Loizou, C.P., C.S. Patticheis, and C. Pattichis, *Despeckle Filtering Algorithms*. 2008: Morgan & Claypool Publishers.

26. Achim, A., A. Bezerianos, and P. Tsakalides, *Novel Bayesian multiscale method for speckle removal in medical ultrasound images*. IEEE Trans Med Imaging, 2001. **20**(8): p. 772-83.
27. Hao, X., S. Gao, and X. Gao, *A novel multiscale nonlinear thresholding method for ultrasonic speckle suppressing*. IEEE Trans Med Imaging, 1999. **18**(9): p. 787-94.
28. Dutt, V., *Statistical analysis of ultrasound echo envelope*. Ph.D. dissertation, Mayo Graduate School, Rochester, MN, 1995.
29. Arezou Akbarian Azar, H.R.a.E.M.B., *Speckle Detection in Echocardiographic Images*. InTech, 2012.
30. Azar, A.A., H. Rivaz, and E. Boctor, *Speckle detection in ultrasonic images using unsupervised clustering techniques*. Conf Proc IEEE Eng Med Biol Soc, 2011. **2011**: p. 8098-101.
31. Busse, L., T.R. Crimmins, and J.R. Fienup, *A model based approach to improve the performance of the geometric filtering speckle reduction algorithm*. IEEE Ultrasonic Symp., 1995. **2**: p. 1353-1356.
32. *Laser speckle and related phenomena*. Edited by j. C. Dainty. Appl Opt, 1984. **23**(16): p. 2661.
33. Rivaz, H., *Motion Estimation In Ultrasound Imaging*. 2010: Proquest, UMI Dissertation Publishing. p. 236.
34. Feldman, M.K., S. Katyal, and M.S. Blackwood, *US artifacts*. Radiographics, 2009. **29**(4): p. 1179-89.
35. Shankar, P.M., *Speckle reduction in ultrasound B-scans using weighted averaging in spatial compounding*. IEEE Trans Ultrason Ferroelectr Freq Control, 1986. **33**(6): p. 754-8.
36. Prager, R.W., et al., *Decompression and speckle detection for ultrasound images using the homodyned K -distribution*. Pattern Recogn. Lett., 2003. **24**(4-5): p. 705-713.
37. Bankman, I.N., et al., *Segmentation algorithms for detecting microcalcifications in mammograms*. IEEE Trans Inf Technol Biomed, 1997. **1**(2): p. 141-9.
38. Bezdek, J.C., L.O. Hall, and L.P. Clarke, *Review of MR image segmentation techniques using pattern recognition*. Med Phys, 1993. **20**(4): p. 1033-48.
39. Pham, D.L., C. Xu, and J.L. Prince, *Current methods in medical image segmentation*. Annu Rev Biomed Eng, 2000. **2**: p. 315-37.
40. Bevan, P.D. and M.D. Sherar, *B-scan ultrasound imaging of thermal coagulation in bovine liver: log envelope slope attenuation mapping*. Ultrasound Med Biol, 2001. **27**(3): p. 379-87.
41. Dussik, K.T., *On the possibility of using ultrasound waves as a diagnostic aid*. Neurol. Psychiat. **174**:153-168. 1942.
42. Elatrozy, T., et al., *The effect of B-mode ultrasonic image standardisation on the echodensity of symptomatic and asymptomatic carotid bifurcation plaques*. Int Angiol, 1998. **17**(3): p. 179-86.
43. Hill, C.R., J.C. Bamber, and G.R. Haar, *Physical Principles of Medical Ultrasonics*. 2004: Wiley.

44. Lamont, D., et al., *Risk of cardiovascular disease measured by carotid intima-media thickness at age 49-51: lifecourse study*. BMJ, 2000. **320**(7230): p. 273-8.
45. Loizou, C.P., et al., *Quality evaluation of ultrasound imaging in the carotid artery based on normalization and speckle reduction filtering*. Med Biol Eng Comput, 2006. **44**(5): p. 414-26.
46. McDicken, W.N., *Diagnostic Ultrasonics: Principles and Use of Instruments*. . John Wiley & Sons, Inc., 1976.
47. Wilhjelm, J.E., et al., *Visual and quantitative evaluation of selected image combination schemes in ultrasound spatial compound scanning*. IEEE Trans Med Imaging, 2004. **23**(2): p. 181-90.
48. Wells, P.N. and H.D. Liang, *Medical ultrasound: imaging of soft tissue strain and elasticity*. J R Soc Interface, 2011. **8**(64): p. 1521-49.
49. Hruska, D.P., J. Sanchez, and M.L. Oelze, *Improved diagnostics through quantitative imaging*. Conf Proc IEEE Eng Med Biol Soc, 2009: p. 1956-9.
50. Hangiandreou, N.J., *AAPM/RSNA physics tutorial for residents. Topics in US: B-mode US: basic concepts and new technology*. Radiographics, 2003. **23**(4): p. 1019-33.
51. Abu-Zidan, F.M., A.F. Hefny, and P. Corr, *Clinical ultrasound physics*. J Emerg Trauma Shock, 2011. **4**(4): p. 501-3.
52. Whittingham, T.A., *Medical diagnostic applications and sources*. Prog Biophys Mol Biol, 2007. **93**(1-3): p. 84-110.
53. W. R. Hedrick, D.L.H., *Image and Signal Processing in Diagnostic Ultrasound Imaging*. Journal of Diagnostic Medical Sonography, 1989. **5**(5): p. 231-239.
54. Rose JS, B.A., *Fundamentals of Ultrasound*. Lippincott Williams & Wilkins, 2006: p. 27-41. (Textbook chapter).
55. Hykes, D.L., *Ultrasound Physics and Instrumentation*. Books on Demand.
56. Lichtenstein, D.A., *Basic Notions of Critical Ultrasound*, in *Whole Body Ultrasonography in the Critically Ill*. 2010, Springer: New York. p. 3-10.
57. Shung, K.K., *Diagnostic Ultrasound: Imaging and Blood Flow Measurements*. 2005: Taylor & Francis.
58. Rose JS., S.B., *Ultrasound physics and knobology*. Mosby-Year book Inc, 1997: p. 10-38.
59. van Soest, G., et al., *Pitfalls in plaque characterization by OCT: image artifacts in native coronary arteries*. JACC Cardiovasc Imaging, 2011. **4**(7): p. 810-3.
60. J.M. Thijssen, B.J.O., P.C. Hartman, G.J. Rosenbusch, *Correlations between acoustic and texture parameters from RF and B-mode liver echograms*. Ultrasound med Biol., 1993. **19**: p. 13-20.
61. Goodman, J.W., *Statistical Optics*. Wiley-Interscience, New York, 1985.
62. Meghoufel, A., et al., *Ultrasound B-scan image simulation, segmentation, and analysis of the equine tendon*. Med Phys, 2010. **37**(3): p. 1038-46.

63. Meghoulfel, A., et al., *Tissue characterization of equine tendons with clinical B-scan images using a shock filter thinning algorithm*. IEEE Trans Med Imaging, 2011. **30**(3): p. 597-605.
64. Smolíková, R., *Neural and statistical modeling of ultrasound backscatter*. Ph.D. dissertation, University of Louisville, Louisville, KY, 2002.
65. Cobbold, R.S.C., *Foundations of Biomedical Ultrasound*. New York, NY: Oxford University Press, 2007.
66. Eltoft, T., *Modeling the amplitude statistics of ultrasonic images*. IEEE Trans Med Imaging, 2006. **25**(2): p. 229-40.
67. Seabra, J. and J. Sanches. *Modeling log-compressed ultrasound images for radio frequency signal recovery*. in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. 2008.
68. Ordsmith, R.J. and M. Abramowitz, *Handbook Mathematical Functions with Formulas Graphs and Mathematical Tables*. Computer Journal, 1967. **10**(1): p. 45.
69. Sanches, J.M. and J.S. Marques, *Compensation of log-compressed images for 3-D ultrasound*. Ultrasound in medicine & biology, 2003. **29**(2): p. 239-253.
70. Bernard, O., et al., *Segmentation of Myocardial Regions in Echocardiography Using the Statistics of the Radio-Frequency Signal*, in *Functional Imaging and Modeling of the Heart*, F. Sachse and G. Seemann, Editors. 2007, Springer Berlin / Heidelberg. p. 433-442.
71. Dutt, V. and J.F. Greenleaf, *Ultrasound echo envelope analysis using a homodyned K distribution signal model*. Ultrason Imaging, 1994. **16**(4): p. 265-87.
72. Shankar, P.M., *A model for ultrasonic scattering from tissues based on the K distribution*. Physics in Medicine and Biology, 1995. **40**(10): p. 1633.
73. Bernard, O., J. D'Hooge, and D. Friboulet, *Statistics of the radio-frequency signal based on K distribution with application to echocardiography*. Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on, 2006. **53**(9): p. 1689-1694.
74. Molthen, R.C., P.M. Shankar, and J.M. Reid, *Characterization of ultrasonic B-scans using non-rayleigh statistics*. Ultrasound in medicine & biology, 1995. **21**(2): p. 161-170.
75. Dutt V, G.J., *Speckle analysis using signal to noise ratios based on fractional order moments*. Ultrasonic Imaging, 1995. **17**: p. 251-268.
76. Prager, R.W., et al., *Analysis of speckle in ultrasound images using fractional order statistics and the homodyned k-distribution*. Ultrasonics, 2002. **40**(1-8): p. 133-7.
77. Wear, K.A., et al., *Statistical properties of estimates of signal-to-noise ratio and number of scatterers per resolution cell*. Journal of the Acoustical Society of America, 1997. **102**: p. 635-641.
78. Popp, K.A.W.a.R.L., *Methods for estimation of statistical properties of envelopes of ultrasonic echoes from myocardium*. IEEE Transactions on Medical Imaging, 1987. **MI-6**: p. 281-291.

79. Ossant, F., et al., *Effective density estimators based on the K distribution: interest of low and fractional order moments*. Ultrason Imaging, 1998. **20**(4): p. 243-59.
80. Rivaz, H., E.M. Boctor, and G. Fichtinger. *P3E-9 Ultrasound Speckle Detection Using Low Order Moments*. in *Ultrasonics Symposium*, 2006. IEEE. 2006.
81. Lang, A., et al., *Multi-modal registration of speckle-tracked freehand 3D ultrasound to CT in the lumbar spine*. Med Image Anal, 2012. **16**(3): p. 675-86.
82. Krupa, A., G. Fichtinger, and G.D. Hager, *Real-time tissue tracking with B-mode ultrasound using speckle and visual servoing*. Med Image Comput Comput Assist Interv, 2007. **10**(Pt 2): p. 1-8.
83. R. W. Prager, A.H.G., G. M. Treece, L. H. Berman, *Analysis of speckle in ultrasound images using fractional order statistics and the homodyned k-distribution*. 2002. **40**: p. 133-137.
84. Martin-Fernandez, M. and C. Alberola-Lopez, *On low order moments of the homodyned-k distribution*. Ultrasonics, 2005. **43**: p. 283-290.
85. M. P. Wachowiak, R.S., J. M. Zurada, and A. S. Elmaghraby, *Estimation of K distribution parameters using neural networks*. IEEE Transactions on Biomedical Engineering, 2002. **49**: p. 617-620.
86. A. Mezache and F. Soltani, *A new approach for estimating the parameters of the K-distribution using fuzzy-neural networks*. IEEE Transactions on Signal Processing, 2008. **56**: p. 5724-5728.
87. Gullo, F., G. Ponti, and A. Tagarelli, *Clustering Uncertain Data Via K-Medoids*, in *Proceedings of the 2nd international conference on Scalable Uncertainty Management*. 2008, Springer-Verlag: Naples, Italy. p. 229-242.
88. Abonyi, J., R. Babuska, and F. Szeifert, *Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2002. **32**(5): p. 612-621.
89. Yih, J.-M. and S.-F. Huang, *Unsupervised clustering algorithm based on normalized Mahalanobis distances*, in *Proceedings of the 9th WSEAS international conference on Applied computer and applied computational science*. 2010, World Scientific and Engineering Academy and Society (WSEAS): Hangzhou, China. p. 180-184.
90. Liu, H.-C., et al., *Fuzzy C-Mean Clustering Algorithms Based on Picard Iteration and Particle Swarm Optimization*, in *2008 International Workshop on Geoscience and Remote Sensing*. 2008: Shanghai. p. 838-842.
91. Tatti, N., *Distances between Data Sets Based on Summary Statistics*. J. Mach. Learn. Res., 2007. **8**: p. 131-154.
92. Bezdek, J.C., *Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions*. IEEE Transactions on Computers, 1975. **24**: p. 835-838.
93. Gath, I., *Unsupervised Optimal Fuzzy Clustering*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. **11**: p. 773-780.

94. Jensen, J.A. *Simulation of advanced ultrasound systems using Field II*. in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*. 2004.
95. Stepanishen, P.R., *Pulsed transmit/receive response of ultrasonic piezoelectric transducers*. J.Acoust.Soc.Am. 69, pp. 1815-1827, 1981. **69**(J.Acoust.Soc.Am): p. pp. 1815-1827.
96. Jensen, J.A., *A Model for the Propagation and Scattering of Ultrasound in Tissue*. J.Acoust.Soc.Am, 1991. **89**: p. pp. 182-191.
97. Jensen, J.A. and N.B. Svendsen, *Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers*. IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 1992. **39**: p. pp. 262-267.
98. Jensen, A., *Linear description of ultrasound imaging systems, Notes for the International Summer School on Advanced Ultrasound Imaging*. Technical University of Denmark, 1999.
99. *Calculation of B-mode image of cyst phantom*. 2013 [cited 2013 9/1/2013]; Available from: (http://field-ii.dk/examples/cyst_phantom/cyst_phantom.html).
100. *Calculation of B-mode image of synthetic kidney*. 2013 [cited 2013 9/1/2013]; Available from: http://field-ii.dk/examples/kidney_example/kidney_example.html.
101. *Calculation of B-mode image of synthetic fetus*. 2013 [cited 2013 9/1/2013]; Available from: http://field-ii.dk/examples/fetus_example/fetus_example.html.
102. Bamber, J.C. and R.J. Dickinson, *Ultrasonic B-scanning: a computer simulation*. Phys Med Biol, 1980. **25**(3): p. 463-79.
103. Thienphrapa, P., et al., *Tracking and characterization of fragments in a beating heart using 3D ultrasound for interventional guidance*, in *Medical Image Computing and Computer Assisted Intervention Conference (MICCAI)*. 2011: Toronto, Canada.
104. Sammon, J.W., Jr., *A Nonlinear Mapping for Data Structure Analysis*. Computers, IEEE Transactions on, 1969. **C-18**(5): p. 401-409.
105. Jensen, J.A. and N.B. Svendsen. *Calculation of B-mode image of synthetic kidney*. Field: Program for Simulating Ultrasound Systems; Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers 1992, 1996 [cited 2013 September 1]; Available from: http://field-ii.dk/examples/kidney_example/kidney_example.html.

Vita

Arezou Akbarian Azar is a PhD candidate in School of Biomedical Engineering, Science & Health Systems, at the Drexel University. She received the Bachelor of Science degree in Computer Science from Tabriz University in 2005, the Masters of Science in Signal Processing, Tampere University of Technology (TUT). Her PhD research was conducted as a trainee in Division of Medical Imaging Physics (DMIP) lab in Johns Hopkins University Hospital, Division of Medical Imaging Physics. Her research interests are medical imaging, signal/image processing, machine learning, pattern recognition, bio-statistics, and image guided surgical technologies. She was ranked third position nationwide in scientific competitions, 1995. She received the Oak Ridge Institute for Science and Education (ORISE) fellowship from Food and Drug Administration (FDA), Center for Devices and

Radiological Health (CDRH), summer 2008. She was a board member of the university IEEE (Institute of Electrical and Electronics Engineers) society.

Her research resulted in publications of a book chapter “Speckle Detection in Echocardiographic Images”, InTech, 2012 as well as conference papers “Speckle Detection in Ultrasonic Images Using Unsupervised Clustering Techniques”, IEEE EMBC 2011, “Novel QT Stability Monitoring Algorithm in Clinical ECG Recordings”, 30th annual scientific sessions, Heart Rhythm 2009, Boston, MA. “Monitoring Blood Oxygenation Changes due to Acute Pain Stimuli using functional Near-Infrared Spectroscopy (fNIRS)”, 31st Annual Conference of the IEEE EMBS, Minneapolis/St. Paul, 2009. “Applying Fuzzy-C-Means Classifier to Process Functional Near-Infra-Red Data for Assessment of Pain Stimulus Intensity”, the 2009 Biomedical Engineering Society Annual Fall Meeting, BMES 2009. “Human Computer Interaction- Applying Fuzzy C-Means, Recurrent NeuralNetwork and Wavelet Transforms for Voluntary Eye Blink Detection”, the 15th European Signal Processing Conference, EUSIPCO 2007. “E-learning” Conference on Information Communication and Telecommunication (ICT 2004), 2004.

