SPECKLE NOISE REDUCTION IN ULTRASOUND IMAGES USING STATISTICAL AND WAVELET BASED FILTERING TECHNIQUES

ELEC 534 Application of Digital Signal Processing Techniques

PROJECT REPORT
Submitted by

Derrell D'Souza (V00901532)

University of Victoria

Date: April 11th, 2018

ABSTRACT

Speckle noise is a multiplicative noise that degrades the visual quality of ultrasound images and affects the clinical assessment. Hence there is a need for a robust method to despeckle these ultrasound images for routine clinical practice and for good clinical evaluation. Two types of filter are specifically investigated in this project namely – Linear First Order Local Statistical Filter and a Wavelet based Speckle Filter. A new filter named Enhanced Wiener based First Statistical filter is also designed. In this project, the theoretical background, the algorithmic implementation and the MATLAB code for these three filtering techniques will be described. In the end, the filters are evaluated using image quality evaluation metrics.

Keywords: Speckle, Ultrasound, Imaging, Image quality, Wavelet, Wiener

I. INTRODUCTION

When we learn about Applications of Digital Signal Processing techniques, we aim to derive better systems and tools that give us different perspective of the same task. Medical imaging is useful for diagnosis and study of various types of illness. Whenever we perform any signal processing techniques on medical images, it is essential to understand that medical images contain minute details which should be unaltered. A slight change can result in a wrong diagnosis, wrong evaluation and can jeopardize the patients' health. In this project, emphasis is laid on suppressing the speckle noise in ultrasound images. Speckle noise is a degrading intrinsic artefact which is a result of constructive and destructive summation of ultrasound echoes.

Several techniques have been developed in the past for despeckling the ultrasound images. These techniques can be mainly categorized into two basic approaches – the compounding approach and the post processing approach. In compounding approach, methods are used to modify data acquisition procedure to produce several images of the same region and combine them to form a single image. In post processing approach, different filtering techniques are implemented on the B-mode Ultrasound Images after they are generated. In the present scenario, much of the emphasis is laid on post formation filtering techniques. In this project, post formation techniques of wavelet based filtering and local statistics filtering is been implemented.

1.1 Ultrasound Imaging System

The block diagram in fig.1.1 illustrates the path of the RF signal from the transducer to the screen inside the ultrasound imaging system. Here, signal is subject to several transformations which have a severe impact on its statistics. The most important of these is the log compression of the signal, which is employed to reduce the dynamic range of the input signal to match the lower dynamic range of the display device.

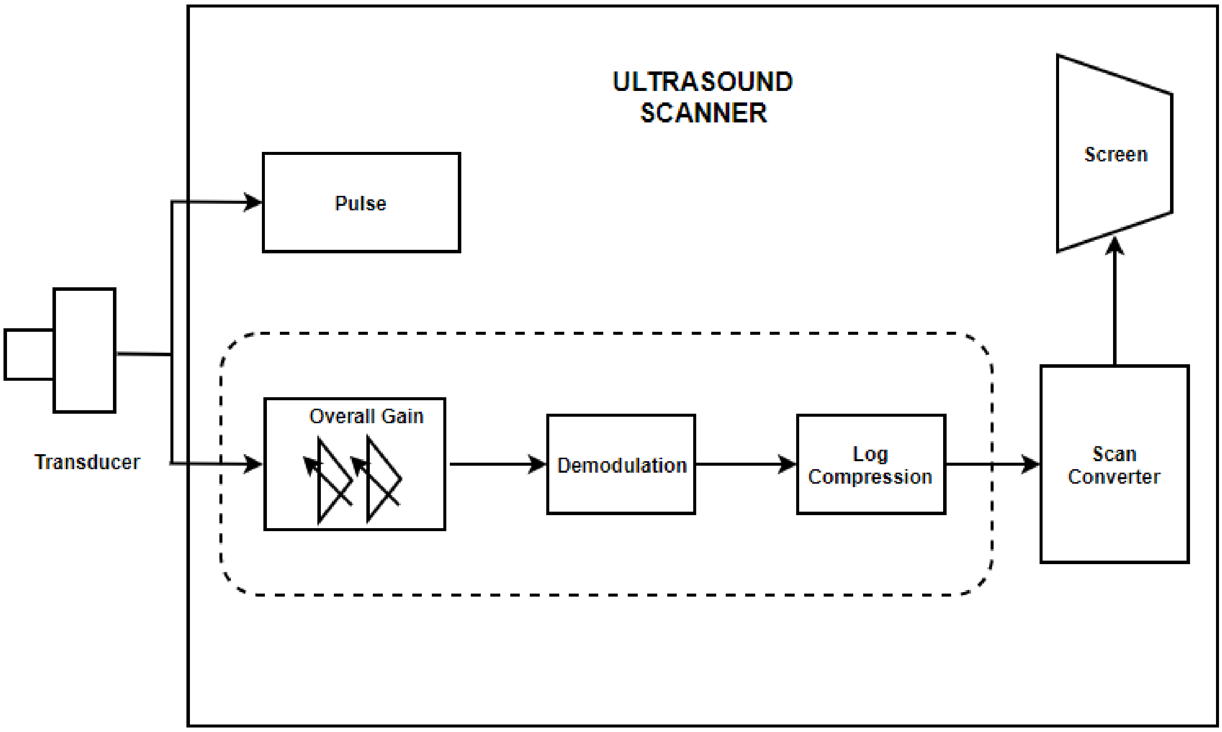

Fig. 1.1 Ultrasound Imaging System and the processing steps involved

The input signal could have a dynamic range of the order of 50–70 dB, whereas a typical display could have a dynamic range of the order of 20–30 Db. Such a relation is normally affected through an amplifier, which has a reducing amplification for a larger input signal.

1.2 Mathematical Modelling of Speckle Noise

The speckle pattern, which is visible as the typical light and dark spots the image is composed of results from destructive interference of ultrasound waves scattered from different sites. For designing and deriving an efficient filter to supress the speckle in ultrasound images, a mathematical model of speckle noise is needed. The speckle noise for ultrasound images is multiplicative. The signal at the output of demodulation system of the ultrasound imaging system in fig. 1.1 may be defined as

$$y_{i,j} = x_{i,j}\, n_{i,j} + a_{i,j} \tag{1.1}$$

where $y_{i,j}$ represents the noisy pixel in the middle of the moving window $x_{i,j}$ represents the noise free pixel, $n_{i,j}$ and $a_{i,j}$ represent the multiplicative and additive noise, respectively, and i, j are the indices of the spatial locations that belong in the 2D space of real numbers, $i, j \in R^2$.

.

The despeckling is performed by estimating the true intensity $x_{i,j}$ as a function of the intensity of the pixel $y_{i,j}$ and some local statistics in the neighbourhood of this pixel. As already mentioned the signal processing inside the scanner will modify the statistics of the signal causing logarithmic compression. As the effect of additive noise is comparatively smaller than the multiplicative noise Eq. 1.1 can be written as

$$y_{i,j} \approx x_{i,j}\, n_{i,j} \tag{1.2}$$

Thus logarithmic transformation will model the signal in Eq. 1.2 into a signal in the additive noise form as

$$\log(y_{i,j}) = \log(x_{i,j}) + \log(n_{i,j}) \tag{1.2}$$

which is equivalent to

$$g_{i,j} = f_{i,j} + nI_{i,j} \tag{1.3}$$

where $\log(y_{i,j})$ which is the observed pixel on the ultrasound display after logarithmic compression is denoted as $g_{i,j}$ and the terms $\log(x_{i,j})$ and $\log\big(n_{i,j}\big)$ which are noise free pixel and noise component after the logarithmic transformation are denoted as $f_{i,j}$ and $nI_{i,j}$ respectively.

1.3 Local Statistical Filter based on First Order Statistics

The working principle of a local statistics filter is the weighted average calculation using sub region statistics to estimate statistical measures over different pixel windows. In this project, first order statistical filter is implemented for speckle suppression. A first order statistical filter utilizes the variances and mean of the neighbourhood as described in Eq.1.4. Consider the equation below,

$$f_{i,j} = \bar{g} + k_{i,j}(g_{i,j} - \bar{g}) \tag{1.4}$$

Here $f_{i,j}$ is the estimated noise free pixel value, $g_{i,j}$ is the noisy pixel in the moving window, $\bar{g}$ is the local mean value of a N_1 x N_2 region surrounding and including pixel $g_{i,j}$, $k_{i,j}$ is the weighting factor with k ϵ [0,1] and *i* and *j* are pixel coordinates. The factor $k_{i,j}$ is a function of local statistics in the moving window. It can be derived in various forms. One particular form which will be used for the project is

$$k_{i,j} = \frac{\sigma^2}{\bar{g}\,\sigma_n^2 + \sigma^2} \tag{1.5}$$

The values σ^2 and σ_n^2 represent the variance in the moving window and variance of the noise in the whole image respectively. In each window the noise variance may be computed as

$$\sigma_n^2 = \sum_{i=1}^{p} \sigma_p^2 / \overline{g_p} \tag{1.6}$$

Where σ_p^2 and $\overline{g_p}$ represent the variance and the mean of the noise in the selected windows respectively, *p* is the index which covers all windows in the whole image. Note that if the value of $k_{i,j}$ is 1 in edge areas than this will result to an unchanged pixel whereas a value of 0 replaces the actual pixel by the local average $\bar{g}$ over a small region of interest.

1.4 Enhanced Wiener Filter Design

In Enhanced Wiener based Statistical Filter, a different weighting factor is used as compared to Eq.1.6.

$$k_{i,j} = \frac{\sigma^2}{\sigma^2 + \alpha\sigma_n^2} \tag{1.7}$$

Here α is a parameter that will decide the strength of regularization. The computational complexity is reduced due to the introduction of multiplication by a constant in the denominator rather than multiplication by the local mean matrix.

1.5 Wavelet based Speckle Filtering

In wavelet filtering, the image is decomposed into wavelet basis and wavelet coefficients are zeroed out to despeckle the image. The advantage here is its ability to analyse the signal both in time domain and in frequency domain. Speckle reduction filtering in wavelet domain is based on Daubechies Symlet wavelet and on the soft-thresholding. The Symlet family of wavelets, although not perfectly symmetrical, was designed to have the least asymmetry and the highest number of vanishing moments for a given compact support.

The threshold for each sub band can be calculated as

$$T = \begin{cases} \left(T_{max} - \alpha(j-1)\right)\sigma_n & if\ T_{max} - \alpha(j-1) \geq T_{min} \\ T_{min}\sigma_n & else \end{cases} \tag{1.8}$$

where α is the decreasing factor between two consecutive levels, T_{max} and T_{min} are maximum and minimum factor for σ_n.

1.6 Evaluation Metrics

The metrics Mean Square Error (MSE), Root Mean Square Error (RMSE), Signal-to-Noise Ratio (SNR) , Peak Signal-to-Noise Ratio (PSNR) and Geometric Average Error (GAE) are used to evaluate the quality of despeckling

1.6.1 Mean-Square Error (MSE)

The Mean-Square Error (MSE) measures the quality change between the original and the processed image. The MSE has been widely used to quantify image quality, and, when it is used alone, it does not correlate strongly enough with perceptual quality. It should be used, therefore, together with other quality metrics and visual perception. It is given

$$MSE = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}[I(m,n) - I'(m,n)]^2 \tag{1.9}$$

Where $I(m,n)$ and $I'(m,n)$ are the reference and the filtered image respectively and $M\ X\ N$ is the size of the image.

1.6.2 Root-Mean-Square Error (RMSE)

The Root-Mean-Square Error (RMSE) is the square root of the squared error averaged over an $M\ X\ N$ window. It is the best approximation of the standard error. It is given by

$$MSE = \sqrt{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}[I(m,n) - I'(m,n)]^2} \tag{1.10}$$

1.6.3 Signal-to-Noise Ratio (SNR)

Although signal sensitivity and image noise properties are important by themselves, their ratio that carries the most significance. The Signal-to-Noise ratio (SNR) is given by

$$SNR = 10log\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[I(m,n)^2 - I'(m,n)^2]}{\sum_{m=1}^{M}\sum_{n=1}^{N}[I(m,n) - I'(m,n)]^2} \tag{1.12}$$

1.6.4 Peak Signal-to-Noise Ratio (PSNR)

The Peak Signal-to-Noise Ratio (PSNR) measures image fidelity, which is how closely the despeckled image resembles the original image. The PSNR is higher for a better transformed image and lower for a poorly transformed image

$$PSNR = 10log\frac{255^2}{MSE} \tag{1.13}$$

1.6.5 Geometric Average Error (GAE)

The value of Geometric Average Error (GAE) is approaching zero if there is a very good transformation (small differences) between the original and despeckled images; otherwise, the value of GAE is high. It is positive only if every pixel value is different between the original and despeckle images.

$$GAE = \left(\prod_{m=1}^{M} \prod_{n=1}^{N} \sqrt{I(m,n) - I'(m,n)} \right)^{1/MN} \tag{1.14}$$

II. IMPLEMENTATION

In this section, the algorithm used for the filter design and the MATLAB code for the same will be explained.

2.1 Filtering Algorithm

2.1.1. First Order Statistical Filter

1. Load the Image for Processing.
2. Specify the region of interest, the window and the number of iterations over which the filtering has to be performed.
3. Compute the noise variance σ_n^2 using Eq.1.7 for the whole image.
4. From the upper left corner of the image compute for each moving window the coefficient $k_{i,j}$ using Eq.1.6.
5. Compute $f_{i,j}$ in Eq.1.5 and replace the noisy middle point in each moving window $g_{i,j}$ with a new computed value $f_{i,j}$.
6. The steps in 4 and 5 has to be repeated for whole image.
7. The steps in 3-6 has to be repeated for n iterations
8. Compute the image quality evaluation metrics and texture features of the original and the despeckled image.
9. Display the original and despeckled images, the image quality and the evaluation metrics and texture features.

2.1.2 Enhanced Wiener based First Order Statistical Filter

1. Load the Image for Processing.
2. Specify the region of interest, the window and the number of iterations over which the filtering has to be performed.
3. Compute the noise variance σ_n^2 using Eq.1.7 for the whole image.
4. From the upper left corner of the image compute for each moving window the coefficient $k_{i,j}$ using Eq.1.8.
5. Compute $f_{i,j}$ in Eq.1.5 and replace the noisy middle point in each moving window $g_{i,j}$ with a new computed value $f_{i,j}$.
6. The steps in 4 and 5 has to be repeated for whole image.
7. The steps in 3-6 has to be repeated for n iterations
8. Compute the image quality evaluation metrics and texture features of the original and the despeckled image.
9. Display the original and despeckled images, the image quality and the evaluation metrics and texture features.

2.1.3 Wavelet based Despeckling Filter

1. Load the Image for Processing.
2. Specify the region of interest, the window and the number of iterations over which the filtering has to be performed.
3. Compute the noise variance σ_n^2 using Eq.1.7. for the whole image.
4. Compute the DWT using Symlet Wavelet for two scales.
5. Compute the threshold using Eq.1.9
6. Apply threshold on the wavelet coefficients for each band.
7. Compute the inverse DWT to reconstruct the whole image.
8. The steps in 3-6 has to be repeated for n iterations
9. Compute the image quality evaluation metrics and texture features of the original and the despeckled image.
10. Display the original and despeckled images, the image quality and the evaluation metrics and texture features.

2.2 MATLAB code:

2.2.1 First Order Statistical Filter

```
%FIRST ORDER STATISTICAL FILTER%
%**************************************************************************
window = [3,3];
g = I; %I is an ultrasound image affected by noise;
iter = 2;
%STEP 1: Load the image for filtering
figure
imshow(I);
title('Original Image')%original image
%STEP 2: Check whether the image that is loaded is gray scale and normalize
if isa(g, 'uint8')
   u8out = 1;
   if (islogical(g))
      logicalOut = 1;
      g = double(g);
   else
      logicalOut = 0;
      g = double(g)/255;
   end
else
   u8out = 0;
end

%initialize a new image f after filtering with zeros
f = g;
%STEP 3: Calculate the noise variance for the whole image
stdnoise=(std2(g).*std2(g))/mean2(g);
noisevar = stdnoise*stdnoise;
%apply n iterations of algorithm to the image
for i = 1:iter
```

```
    fprintf('\rIteration %d',i);
  if iter >=2
      g = f;
  end
%STEP 4: Starting from the left upper corner of the image, compute for each moving window the
coefficient ki,j
%estimate the local mean of f
localMean = filter2(ones(window), g) / prod(window);
%estimate the local variance of f
localVar = filter2(ones(window), g.^2) / prod(window)-localMean.^2;
%compute the result
lmsqr = localMean.*localMean;
t=lmsqr.*noisevar;
k = localVar./ (t + localVar+0.0001); %coefficient of variation
f = localMean + k .*(g-localMean);    % Outpout (Despeckled)image
end
fprintf('\n');
if u8out==1
  if (logicalOut)
    f = uint8(f);
  else
    f = uint8(round(f*255));
  end
end
%STEP 5: Display the output
figure, imshow(f), title('Despeckled Image using Local Statistical filtering')
```

2.2.2 Enhanced Wiener based First Order Statistical Filter

```
%ENHANCED WIENER BASED FIRST ORDER STATISTICAL FILTER%
%**************************************************************************
window = [3,3];
alpha = 0.075; %parameter that decides regularization strength
g = I; %I is an ultrasound image affected by noise;
iter = 5;
%STEP 1: Load the image for filtering
figure
imshow(I);
title('Original Image')%original image
%STEP 2: Check whether the image that is loaded is gray scale and normalize
if isa(g, 'uint8')
    u8out = 1;
    if (islogical(g))
        logicalOut = 1;
        g = double(g);
    else
        logicalOut = 0;
        g = double(g)/255;
    end
```

```
else
   u8out = 0;
end

%initialize a new image f after filtering with zeros
f = g;
%STEP 3: Calculate the noise variance for the whole image
stdnoise=(std2(g).*std2(g))/mean2(g);
noisevar = stdnoise*stdnoise;
%apply n iterations of algorithm to the image
for i = 1:iter
   fprintf('\rIteration %d',i);
  if iter >=2
     g = f;
  end
%STEP 4: Starting from the left upper corner of the image, compute for each moving window the coefficient ki,j
%estimate the local mean of f
localMean = filter2(ones(window), g) / prod(window);
%estimate the local variance of f
localVar = filter2(ones(window), g.^2) / prod(window)-localMean.^2;
%compute the result
k = (localVar)./ (localVar+ (noisevar)*alpha); %coefficient of variation
f = localMean + k .*(g-localMean);    % Outpout (Despeckled)image
end
fprintf('\n');
if u8out==1
  if (logicalOut)
    f = uint8(f);
  else
    f = uint8(round(f*255));
  end
end
%STEP 5: Display the output
figure, imshow(f), title('Despeckled Image using Enhanced Wiener Based Local Statistical filtering')
```

2.2.3 Wavelet based Despeckling Filter

```
%WAVELET BASED SPECKLE FILTER%
%***************************************************************************
iter = 50;
g = I;
if isa(g, 'uint8')
  u8out = 1;
  if (islogical(g))
    logicalOut = 1;
    g = double(g);
  else
    logicalOut = 0;
```

```
    g = double(g)/255;
  end
else
  u8out = 0;
end

for i = 1:iter
  fprintf('\rIteration %d',i);
  if i >=2
      g=f;
  end

    [thr, sorh, kepapp]=ddencmp('den', 'wv', g); %wavelet toolbox function used to select default values for denoising
    f=wdencmp('gbl', g, 'sym4', 2, thr, sorh, kepapp);  %wavelet toolbox function to perform denoising

end        % end for i itterations

figure,imshow(I), title('Original Image');
figure, imshow(f), title('Despeckled Image using Enhanced Wavelet Based Speckle Filtering')
```

2.2.4 Evaluation Metrics

The following Matlab Function is designed to evaluate the metrics MSE, RMSE, SNR , PSNR and GAE according to equations 1.10, 1.11, 1.12, 1.13 and 1.14.

```
function metrics = evalmetrics(Orig_Image,Esti_Image)

% This function evalmetrics calculates various performance metrics as

%
% INPUTS: Orig_Image = Original Image
%        Esti_Image = Estimation of the original image obtained from a
%        noisy image after filtering it.
% OUTPUT:
%        metrics = is a structure with following fields
%                Mean-Square Error (MSE)
%                Root-Mean-Square Error (RMSE)
%                Signal-to-Noise Ratio (SNR)
%                Peak Signal-to-Noise Ratio (PSNR)
%                Geometric Average Error (GAE)


%---Checking Input Arguments
if nargin<1||isempty(Esti_Image), error('Input Argument: Estiamted Image Missing');end

%---Implementation starts here
if (size(Orig_Image)~= size(Esti_Image)) %---Check images size
    error('Input images should be of same size');
else
    %---Mean-Square Error(MSE) Calculation
```

```
    Orig_Image = im2double(Orig_Image);%---Convert image to double class
    Esti_Image = im2double(Esti_Image);%---Convert image to double class
    [M N] = size(Orig_Image);%---Size of Original Image
    err = Orig_Image - Esti_Image;%---Difference between two images
    metrics.MSE = (sum(sum(err .* err)))/(M * N);
%---Root-Mean-Square Error(RMSE) Calculation
metrics.RMSE = sqrt(metrics.MSE);

%---Signal-to-Noise Ratio(SNR) Calculation
    metrics.SNR = 10*log10((1/M*N)*sum(sum(Orig_Image.*Orig_Image))/(metrics.MSE));

    %---Peak Signal-to-Noise Ratio(PSNR) Calculation
    if(metrics.MSE > 0)
        metrics.PSNR = 10*log10(255*255/metrics.MSE);
    else
        metrics.PSNR = 99;
    end
    metrics.GAE = GAE(Orig_Image,Esti_Image);
end
```

2.2.5 Geometric Average Error Function

The function for GAE has also been developed

```
%Geometric Average Error
function T = GAE(P,Q);
len=(size(P,1));
wid=(size(P,2));
P=double(P)/255;
Q=double(Q)/255;
M=ones(len);
T=1;


for i = 1:len
    for j=1:wid
        Pr=P(i,j);
        Qr=Q(i,j);
        mag=((Pr-Qr)^2)^0.5;
        T=T*mag;
    end
end
T=T^(1/(len*wid));
```

III. RESULTS AND EVALUATIONS

3.1 Ultrasound Images for Evaluation

Fig.3.1 and Fig.3.2 show the original Ultrasound Image and the Ultrasound Image Corrupted by Speckle Noise.

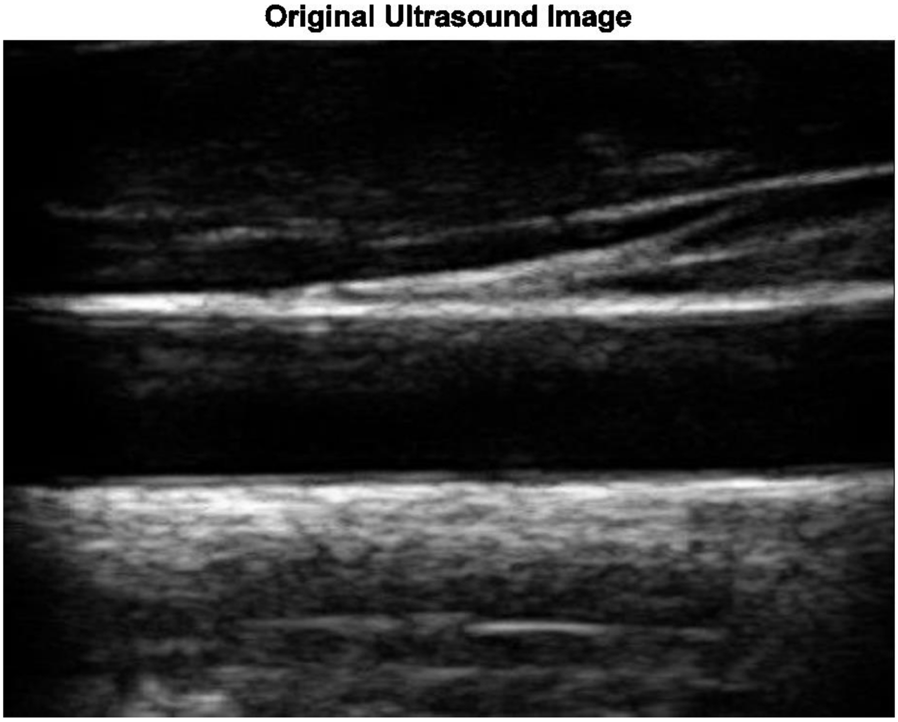

Fig 3.1

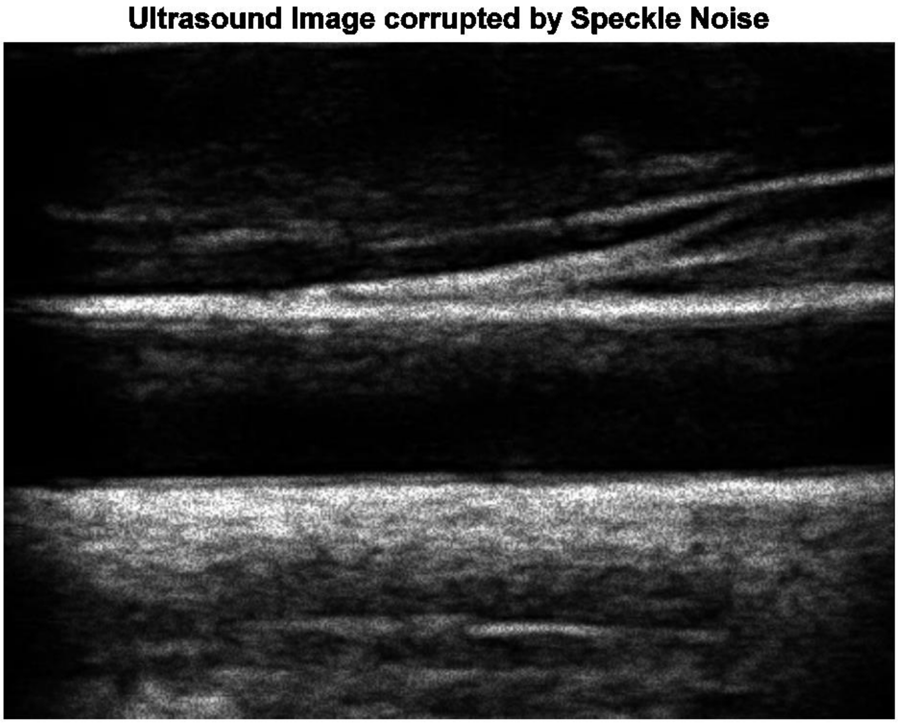

Fig 3.2

3.2 First Order Local Statistical Filter

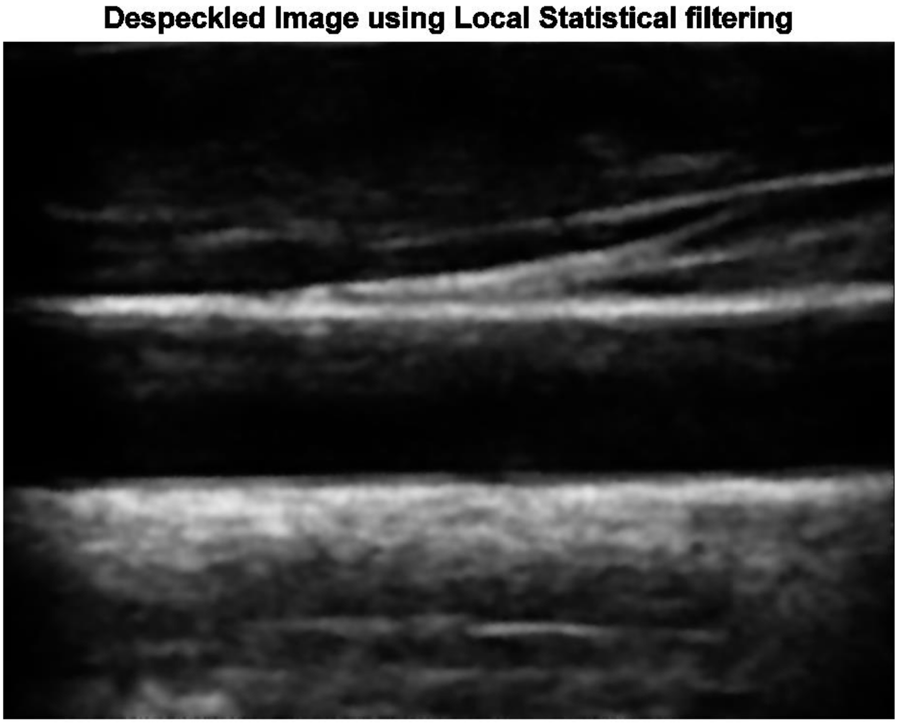

Fig 3.3 Window : [3, 3] , Iterations: 5

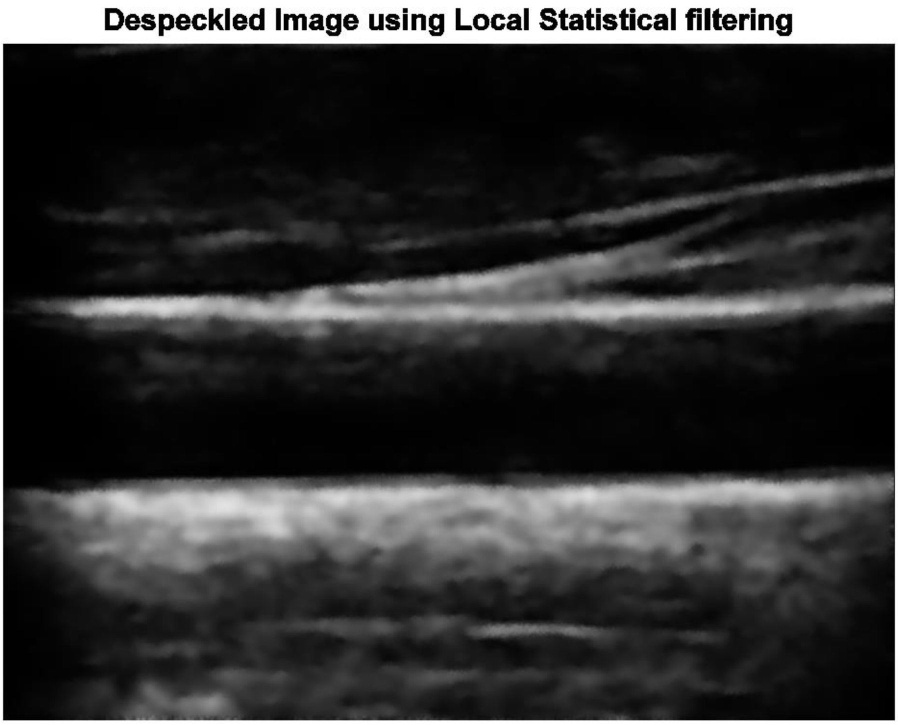

Fig 3.4 Window: [5, 5], Iterations: 3

3.3 Enhanced Wiener based First Order Statistical Filter

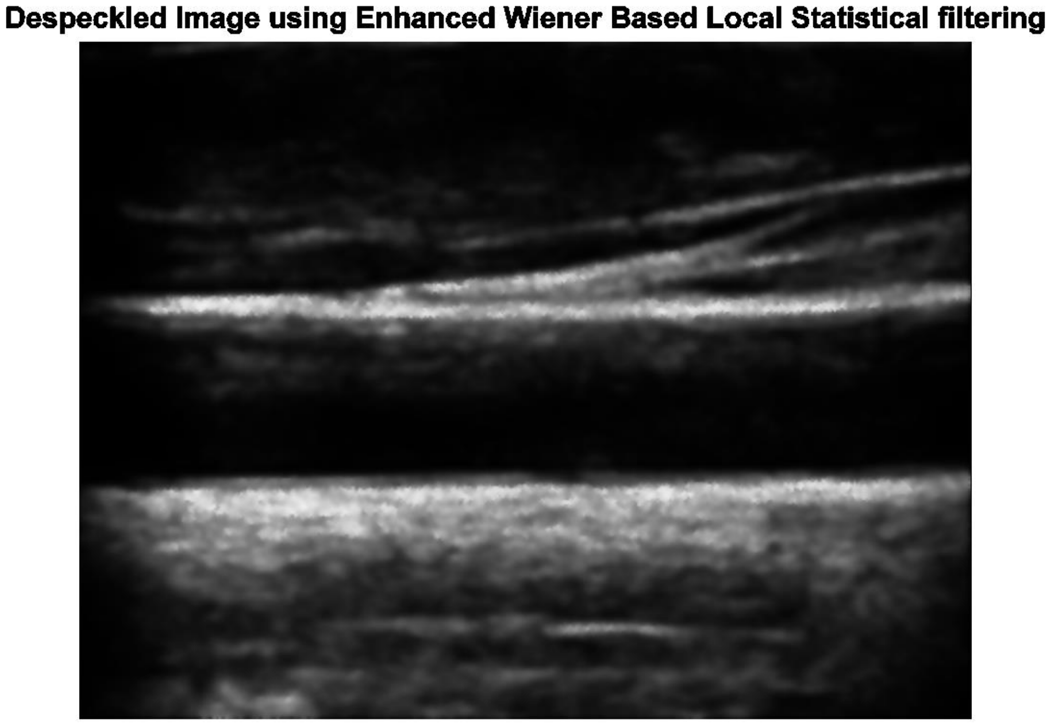

Fig. 3.5 Window : [3, 3] , Iterations: 5, alpha = 0.075

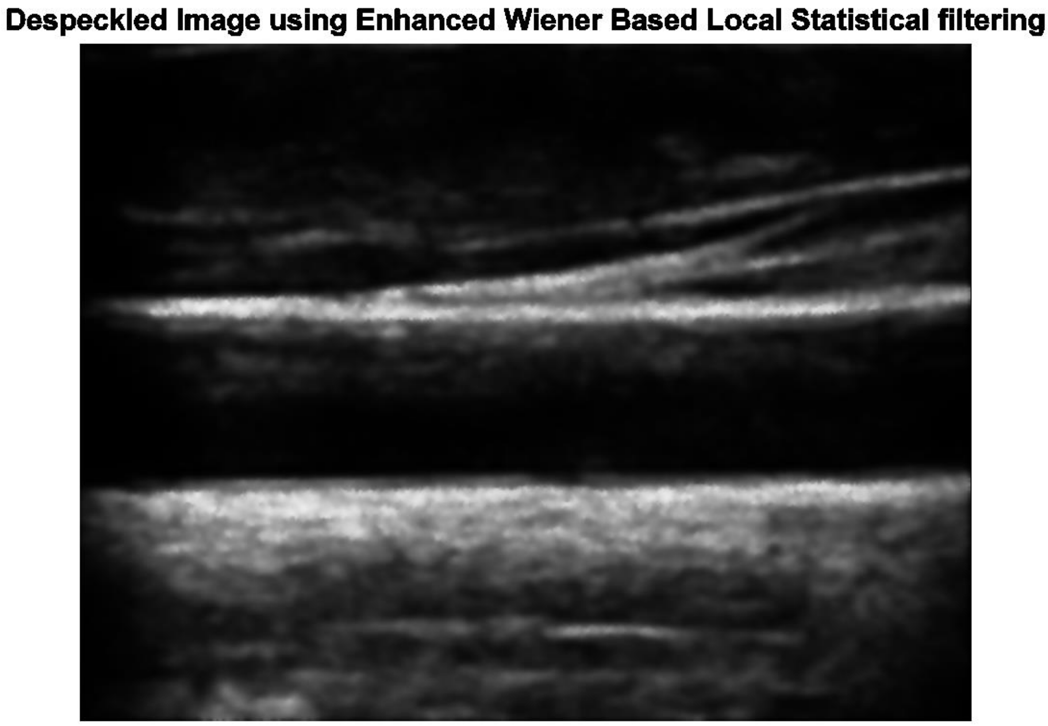

Fig. 3.6 Window : [3, 3] , Iterations: 5, alpha = 0.090

Despeckled Image using Enhanced Wiener Based Local Statistical filtering

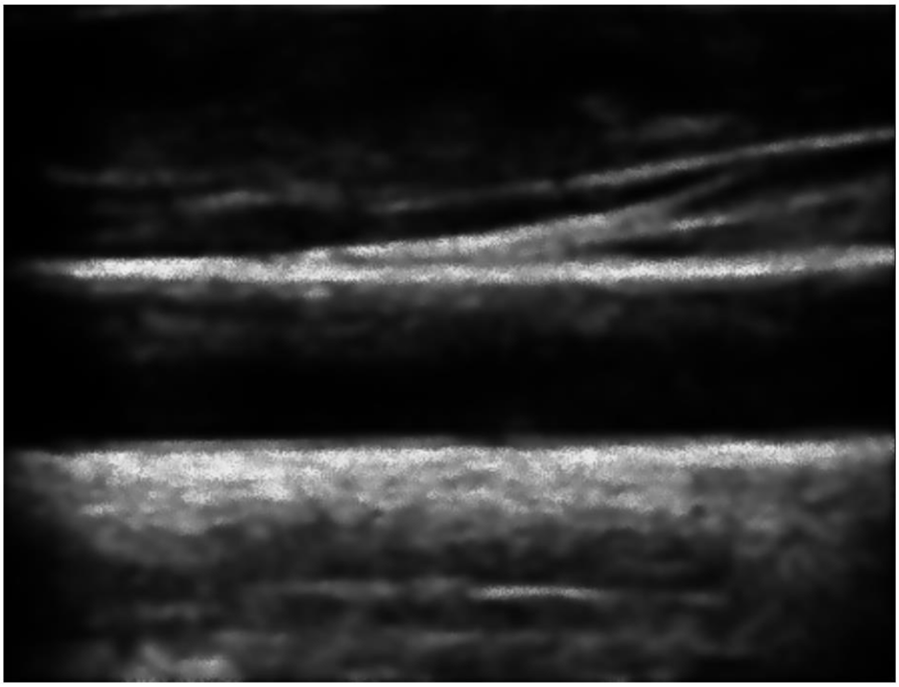

Fig. 3.7 Window : [5, 5] , Iterations: 3, alpha = 0.075

Despeckled Image using Enhanced Wiener Based Local Statistical filtering

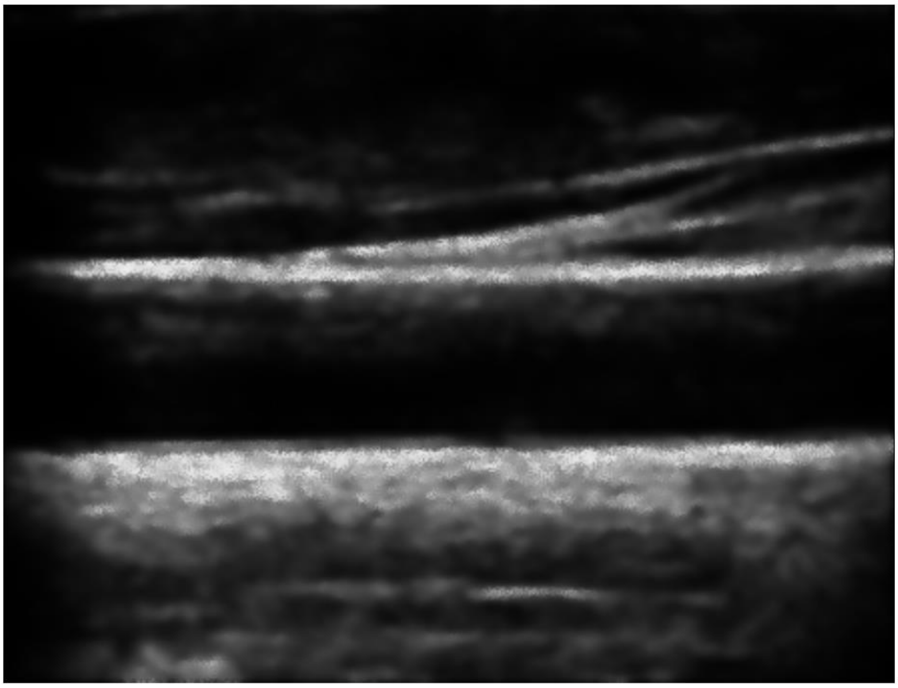

Fig. 3.8 Window : [5, 5] , Iterations: 3, alpha = 0.075

3.4 Wavelet based Speckle Filter

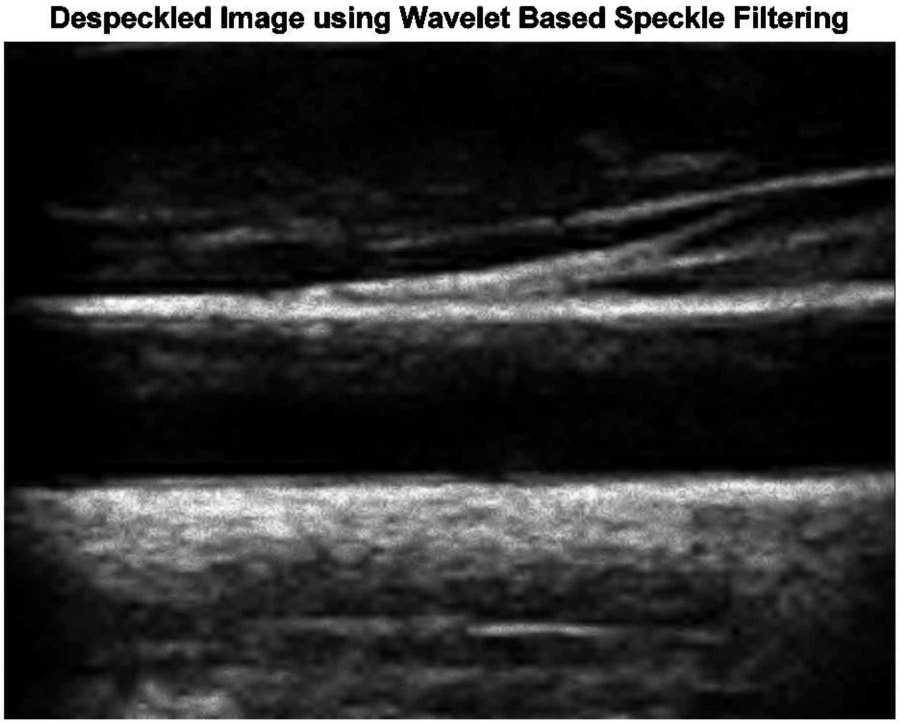

Fig 3.9 Iterations: 3

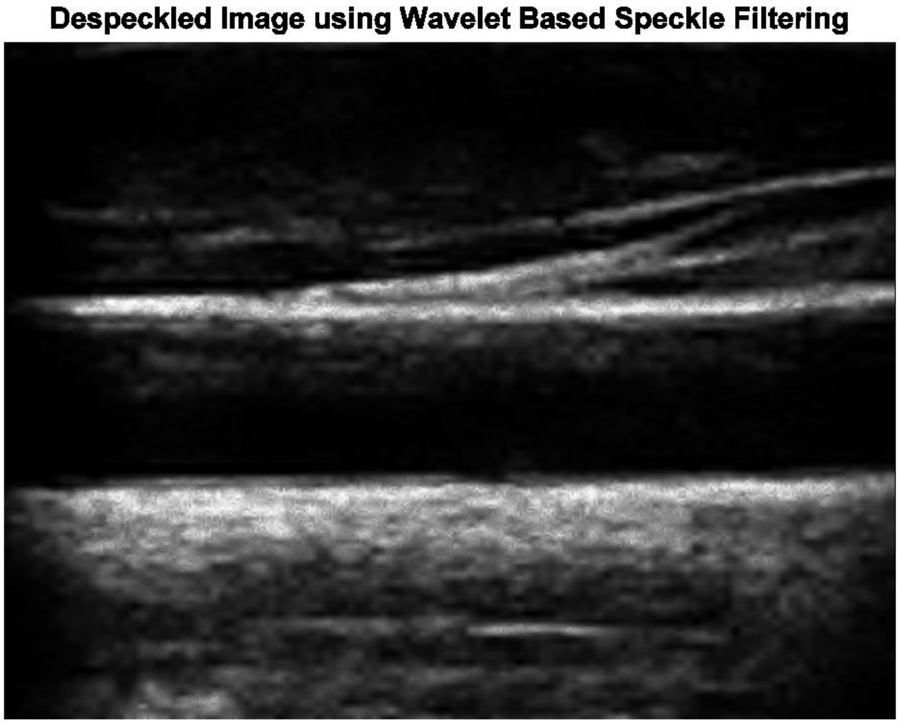

Fig. 3.10 Iterations: 5

3.5 Evaluation

Metric	First Order Local Statistical Filter [window, iter]		Enhanced Wiener based First Order Statistical Filter [window, iter, alpha]				Wavelet based Speckle filter [iter]	
	[3,3], 5	**[5,5], 3**	**[3,3] ,5,0.075**	**[3,3],5,0.09**	**[5,5],3,0.075**	**[5,5],3,0.09**	**3**	**5**
MSE	8.1245e-04	0.0010	8.3449e-04	8.5297e-04	0.0012	0.0012	0.0010	9.2344e-04
RMSE	0.0285	0.0318	0.0289	0.0292	0.0349	0.0349	0.0321	0.0304
SNR	71.7487	70.7846	71.6325	71.5374	69.9980	70.1100	70.7215	71.1926
PSNR	79.0328	78.0687	78.9166	78.8215	77.2821	77.3941	78.0056	78.4767
GAE	0	0	0	0	0	0	0	0

Table 3.1 Summary of Findings from the Speckle Suppression of the Ultrasound Image.

Comments:

From the Table 3.1, it can be inferred that the First Order Statistical Filter performs best for speckle noise reduction. The newly designed Enhanced Wiener based First Order Statistical Filter closely matches the performance of the First Order Statistical Filter. As the number of iterations is increased in wavelet filtering it performs better. However the performance doesn’t improve by much for wavelet filtering for higher iterations. The GAE is 0.00 for all cases, and this can be attributed to the fact that the information between the original and the processed images remains unchanged. Visually not much difference can be observed in the images however the first order local statistical filter performs better than wavelet filtering. The wavelet filtering does not eliminate a lot of noise and there is an effect of white dot observed which distorts the image very much.

IV. DISCUSSION

The wavelet toolbox in MATLAB is very useful for denoising the Ultrasound Image. The computations and code length have been drastically reduced due to the available of the inbuilt functions for denoising and for choosing the default values for wavelet filtering. The newly proposed Enhanced Wiener Filter reduces the computation cost of the First Order Statistical Filter owing to the usage of a constant alpha instead of using the calculated local mean matrix for multiplication in the denominator. The proposed filter is able to match the performance of the First Order Statistical Filter

V. CONCLUSION

This project has focused on dealing with a specific characteristic noise pattern present in ultrasound images known as speckle noise. Different filters are compared and were evaluated based on evaluation metrics The difference in performance between the filters designed in this project for speckle suppression in ultrasound image is small, but the choice of the correct window size and number of iterations as well as the parameter alpha(in Enhanced Wiener Filtering) is important.

VI. REFERENCES

[1] Adam D., Beilin-Nissan S., Friedman Z., Behar V. (2006). The combined effect of spatial compounding and nonlinear filtering on the speckle reduction in ultrasound images. Ultrasonics 44 (2) 166-181.

[2] V. Dutt, “Statistical analysis of ultrasound echo envelope,” Ph.D. dissertation, Mayo Graduate School, Rochester, MN, 1995.

[3] J.U. Quistgaard, “Signal acquisition and processing in medical diagnostic ultrasound,” *IEEE Signal Process. Mag.*, vol. 14, no. 1, pp. 67–74, January 1997

[4] J.S. Lee, “Speckle analysis and smoothing of synthetic aperture radar images,” *Comp. Graphics Image Process.*, vol. 17, pp. 24–32, 1981, doi:10.1016/S0146-664X(81)80005-6.

[5] D.L. Donoho, “Denoising by soft thresholding,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, 1995

[6] R. Gonzalez and R. Woods, *Digital image processing*, 2nd ed., Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2002.

[7] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: From error measurement to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, April 2004, doi:10.1109/TIP.2003.819861.

[8] S. Winkler, “Vision models and quality metrics for image processing applications,” PhD Dissertation, University of Lausanne, Lausanne, Switzerland, December 21, 2000.