CHAPTER 2

2.1 INTRODUCTION

Different types of methods have been developed in two major areas like spatial domain and frequency domain for reduction of speckle noise. In the present work the main concentration is on some major areas of de-speckling methods. Mainly the despeckling methods are broadly given as LEE filter [15,16], Frost filter [17], Kaun filter [18], Sigma filter [19], Refined LEE filter, Gamma Map filter [20,21], Wavelet transform [22-28], Curvelet transform, Principal Component based transform, Non Local Mean method [29-31], Block Matching 3-Dimentional method (BM-3D), Compressive Sensing 3-Dimentional method (CS-3D) [32], Probabilistic Patch Based method [33,34], etc. The BM-3D method is generally considered as state of art method in denoising of regular images and despeckling of SAR images.

The efficiency of spatial filters [35-37] mainly depends on the choice of the size and orientation of the local window. All standard speckle filters [38] make use of neighboring pixels statistical characteristic within the local window to calculate the expected value needed to replace the filtered pixel [39]. The size of the filter window will determine the amount of speckle reduced and the visual quality of the denoised image. The filter will start its computation within the filter window from the top left corner of the padded image [40]. If the selected filter is a median filter, the value of the first pixel will be replaced by the median value of its surrounding pixels within the filter window. The same approach is attempted to replace all pixels of the image as shown in figure 2.1.

![Diagram](image)

Fig. 2.1: Standard speckle filter process
2.2 ENHANCED LEE FILTER

Lopes proposed a filter method to divide an image into areas of three major classes [41- 44]. The first category of class is about homogeneous or smoother areas in which the speckle noise is removed by using low pass filter. The second category of class is about heterogeneous areas in which the speckle is to be minimized while preserving edge details and the third category of class is the area holding isolated points, where the filter should preserve the examined point target value [45]. Generally almost all the advanced spatial filters follow this strategy.

The resulting grey-level value \( R \) is:

\[
R = \begin{cases} 
\text{Mean} & \text{for } (\text{SD/\text{Mean}}) \leq \sqrt{1/N\text{look}} \\
\text{Mean} \times \left[ \exp (- \text{damp} ( (\text{SD/\text{Mean}}) - \sqrt{1/N\text{look}} ) / (\sqrt{1 + 2/N\text{look}} )) - (\text{SD/\text{Mean}}) + C_p \times (1 - \exp (- \text{damp} ((\text{SD/\text{Mean}}) - (\sqrt{1/N\text{look}} ))) \right] & \text{for } (\sqrt{1/N\text{look}}) < (\text{SD/\text{Mean}}) < (\sqrt{1 + 2/N\text{look}}) \\
C_p & \text{for } (\text{SD/\text{Mean}}) \geq \sqrt{1/N\text{look}} 
\end{cases}
\]

(2.1)

(2.2)

(2.3)

2.3 WIENER FILTER

Norbert Wiener proposed the concept of Wiener Filtering [46] in the year 1942. There are two methods: (i) Fourier-transform method (frequency-domain) and (ii) mean-squared method (spatial-domain) for implementing Wiener Filter [47].

In fourier transform technique of Wiener Filtering, it needs a prior knowledge of the noise power spectra and the original image. But in spatial domain technique, no such prior knowledge is needed. It is easy to evaluate the mean-squared value. The principle of the Wiener filter is to remove the mean-squared error between the actual output and the desired output.

In the frequency domain, the Wiener Filter [48] is given by

\[
H(u, v) = P_D(u, v) / \left[ P_D(u, v) + P_N(u, v) \right]
\]

where \( P_D(u, v) \) and \( P_N(u, v) \) represent the power spectra of \( d(i, j) \) and \( n(i, j) \), respectively and \( d(i, j) \) and \( n(i, j) \) represent the original image and noise image respectively. This solution is derived in a similar way with that in the space domain. By minimizing

\[
J = E[|D(u, v) - H(u, v)X(u, v)|^2]
\]

(2.4)

(2.5)
where $D(u,v)$ and $X(u,v)$ represent the discrete Fourier transforms (DFTs) of $d(i,j)$ and $x(i,j)$ respectively. The solution is first obtained as

$$H(u, v) = \frac{E[X(u,v) \cdot D^*(u,v)]}{E[|X(u,v)|^2]}$$

(2.6)

where '*' denotes complex conjugate. When $n(i,j)$ is white noise, the numerator reduces to

$$E[X(u,v) \cdot D^*(u,v)] = E[(D(u,v) + N(u,v)) \cdot D^*(u,v)]$$

(2.7)

$$= E[|D(u,v)|^2]$$

(2.8)

$$= P_D(u,v)$$

(2.9)

and the denominator reduces to

$$E[|X(u,v)|^2] = P_D(u,v) + P_N(u,v)$$

(2.10)

where $P_D(u,v)$ and $P_N(u,v)$ correspond to the power spectra of $d(i,j)$ and $n(i,j)$, respectively.

The output of the Wiener filter is given by

$$Y(u,v) = H(u,v) \cdot X(u,v)$$

(2.11)

$$y(i,j) = IDFT [Y(u,v)]$$

(2.12)

where IDFT means the Inverse DFT.

2.4 THE DISCRETE WAVELET TRANSFORM (DWT)

Mainly the frequency domain filters started with revolution from the invention of wavelets [49,50]. If it is possible to separate the signal from noise, it is very much easy to denoise the images. Based on the same principle the wavelet transform denoising technique works. The basic idea of wavelet technique is that noise mainly exists on the high-frequency components [51-55] and thus can be removed [56-58].

The discrete wavelet transform [59-61] is a discrete version of continuous wavelet transform and its computation may consume significant amount of time and resources, depending on the resolution required. The discrete wavelet transform, which is based on sub-band coding is found to yield a fast computation of wavelet transform [62]. It is easy for implementation and reduces the computation time and resources required [63,64].

Filters are one of the most widely used signal processing functions. Wavelets [65] can be realized by iteration of filters with re-scaling. The resolution of the signal,
which is a measure of the amount of detail information in the signal, is evaluated by
the filtering operations, and the scale is determined by up-sampling and down-
sampling operations.

The DWT is computed by successive lowpass and highpass filtering of the discrete
time-domain signal as shown in figure 2.2. Its significance is in the manner it
connects the continuous time multi-resolution to discrete time filters.

Fig. 2.2: Three-level Wavelet Decomposition Tree

At each decomposition level, the half band filters produce signals spanning only
half the frequency band. This decimation by 2 halves the time resolution as the entire
signal is now represented by only half the number of samples. Thus, while the half
band low pass filtering removes half of the frequencies and thus halves the resolution,
the decimation by 2 doubles the scale.

Fig. 2.3: Three-level Wavelet Reconstruction Tree

Figure 2.3 shows the reconstruction of the original signal from the wavelet
coefficients. Basically, the reconstruction is the reverse process of decomposition.
The approximation and detail coefficients at every level are up sampled by two, passed through the low pass and high pass synthesis filters and then added. This process is continued through the same number of levels as in the decomposition process [66-68].

Do and Vetterli introduced the new two-dimensional contourlet transform [69-74]. This transform is more suitable for constructing a multi-resolution and multi-directional expansions using non-separable pyramid directional filter banks with small redundancy factor. Compared with wavelet, contourlet provides [75] different and flexible number of directions at each scale. It has been successfully employed in image enhancement, denoising and fusion. Non sub-sampled contourlet transform [76-80] decomposition is to compute the multi scale and different direction components of the discrete images. It involves the two stages such as non subsampled pyramid and non subsampled directional filter bank to extract the texture, contours and detailed coefficients.

### 2.5 PRINCIPAL COMPONENT ANALYSIS-LOCAL PIXEL GROUPING METHOD (PCA-LPG)

#### 2.5.1 Modeling of spatially adaptive PCA denoising

In Principal Component Analysis domain [81,82], the scene signal is mostly captured by several leading principal components, while the last few components with low variances are mainly due to noise. On the other hand, locally learned principal component analysis bases, a series of mutually orthogonal directions with sequentially largest variances, have shown better capability of representing structural features, e.g., image edges and texture. PCA is a classical de-correlation technique in statistical signal processing and it is pervasively used in pattern recognition and dimensionality reduction. In the PCA based Local Pixel Grouping, it has been explained as a model for a pixel and its nearest neighbors as a vector variable. The training samples of this variable are selected by grouping the pixels with similar local spatial structures to the underlying one in the local window. With such an LPG procedure, the local statistics of the variables can be accurately computed so that the image edge structures can be well preserved after shrinkage in the PCA domain for noise removal.
It is assumed that the noise \( 'u' \) corrupted in the original image \( 'I' \) is white additive with zero mean and standard deviation \('\sigma'\), i.e. \( I_u = I + u \), where \( I_u \) is the observed noisy image. The image \( I \) and noise \( u \) are assumed to be uncorrelated. The goal of denoising is to obtain an estimation, denoted by \( \hat{I} \), of \( I \) from the observation \( I_u \). The denoised image \( \hat{I} \) is expected to be as close as possible to \( I \). An image pixel is described by two quantities, the spatial location and its intensity, while the image local structure is represented as a set of neighboring pixels at different intensity levels. Since most of these mantic information of an image is conveyed by its edge structures, edge preservation is highly desired in image denoising. Referring to figure 2.4, for an underlying pixel to be denoised, it is taken a \( K \times K \) window centered on it and is denoted by \( x = [x_1 \ldots x_m]^T \), \( m = K^2 \), the vector containing all the components within the window. Since the observed image is noise corrupted, it is given by

\[
x_u = x + u
\]

The noisy vector of \( x \), where \( x_u = [x_u^1 \ldots x_u^m] \), \( u = [u_1 \ldots u_m] \) and \( x_u^k = x_k + u_k \), where \( k = 1, \ldots , m \).

In order to remove the noise from \( x_u \) by using PCA, it is required a set of training samples of \( x_u \) so that the co-variance matrix of \( x_u \) and hence the PCA transformation matrix can be calculated. For this purpose, \( L \times L \) (\( L \gg K \)) training block is centered on \( x_u \) to find the training samples, as shown in figure 2.4.

The simplest way is to take the pixels in each possible \( K \times K \) block within the \( L \times L \) training block as the samples of noisy variable \( x_u \). In this way, there are totally \((L-K+1)^2\) training samples for each component \( x_u^k \) of \( x_u \). However, there can be very different blocks from the given central \( K \times K \) block in the \( L \times L \) training window so that taking all the \( K \times K \) blocks as the training samples of \( x_u \) will lead to in accurate estimation of the co-variance matrix of \( x_u \), which subsequently leads to in accurate estimation of the PCA transformation matrix and finally results in much noise residual. Therefore, selecting and grouping the training samples that are similar to the central \( K \times K \) block is necessary before applying the PCA transform for denoising.
2.5.2 Local pixel grouping

Grouping the training samples similar to the central $K \times K$ block in the $L \times L$ training window is indeed a classification problem and thus different grouping methods, such as block matching, correlation-based matching, K-means clustering, etc., can be employed based on different criteria. Among them, the block matching method may be the simplest yet very efficient one.

There are totally $(L-K+1)^2$ possible training blocks of $x_u$ in the $L \times L$ training window. It is denoted by $X^u_0$ the column sample vector containing the pixels in the central $K \times K$ block and denote by $X^u_i, i=1,2,\ldots, (L-K+1)^2-1$.

The sample vectors corresponding to the other blocks. Let $X_0$ and $X_i$ be the associated with noise less sample vectors of $X^u_0$ and $X^u_i$, respectively. It can be easily calculated that

$$e_i = \frac{1}{m \sum_{k=1}^{m} x_0(k)-x_i(k^2)}$$  \hspace{1cm} (2.14)

Suppose it is taken $n$ sample vectors of $x_u$, including the central vector. For the convenience of expression, these sample vectors can take as $X^u_0, X^u_1, \ldots, X^u_{n-1}$. The noise less counter parts of these vectors are denoted as $X_0, X_1, \ldots, X_{n-1}$, accordingly. The training data set for $x_u$ is then formed by

$$X_u = [X^u_0, X^u_1, \ldots, X^u_{n-1}]$$  \hspace{1cm} (2.15)

The noise less counter part of $X_u$ is denoted as $X = [X_0, X_1, \ldots, X_{n-1}]$
2.6 FAST DISCRETE CURVELET TRANSFORMS (FDCT)

The second implementation which is referred to as the FDCT [83,84] via wrapping, and whose architecture is as follows:

1. Apply the 2D FFT and obtain Fourier samples
   \[ \hat{f}[n_1, n_2], \quad n/2 \leq n_1, n_2 < n/2. \]
2. For each scale \( j \) and angle \( l \), form the product
   \[ \hat{U}_{j,l}[n_1, n_2] \cdot \hat{f}[n_1, n_2] \]
3. Wrap this product around the origin and obtain
   \[ \tilde{f}[n_1, n_2] = W(\hat{U}_{j,l}, \hat{f}) \]
   where the range for \( n_1 \) and \( n_2 \) is now \( 0 \leq n_1 < L_{1,j} \) and \( 0 \leq n < L_{2,j} \)
   (for \( \theta \) in the range \((-\pi/4, \pi/4)\)).
4. Apply the inverse 2D FFT to each \( f_{j,l} \) hence collecting the discrete coefficients
   \( c^D(j, l, k) \).

This algorithm has computational complexity \( (n^2 \log n) \) and in practice, its computational cost does not exceed that of 6 to 10 two dimensional FFTs.

2.7 BLOCK MATCHING 3D ALGORITHM

A combination of transform domain and spatial domain image denoising algorithms is presented in BM-3D algorithm [85]. It combines non-local filtering approach with other effective denoising tools, like wavelet shrinkage and wiener filtering. The SAR oriented version of SAR BM-3D, follows the rationale and overall structure of BM-3D but modifies its basic steps to adapt to the multiplicative nature of speckle. The SAR BM-3D algorithm [86,87] despeckles SAR images by combining the concepts of non-local filtering and wavelet-domain shrinkage, which has a better capacity to preserve relevant details while smoothing homogeneous areas. However, the smoothing of homogeneous areas and the preserving of edges are still not well balanced in these methods. The BM-3D method is generally considered as state of art method in denoising of images and despeckling of SAR images.

This general procedure is implemented in two different forms to compose a two-step algorithm. This algorithm is illustrated in figure 2.5 and proceeds as follows:
2.7.1 Step 1: Basic estimate

a) Block-wise estimates: For each block in the noisy image, the following are to be done.

i) Grouping: Blocks that are similar to the currently processed one are to be found and to be stacked together in a 3D array (group).

ii) Collaborative hard-thresholding: A 3D transform to the formed group is to be applied and the noise to be attenuated by hard-thresholding of the transform coefficients, invert the 3D transform to produce estimates of all grouped blocks, and return the estimates of the blocks to their original positions.

b) Aggregation: The basic estimate of the true-image is to be computed by weighted averaging all of the obtained block-wise estimates that are overlapping.

![Flow chart of BM-3D method](image)

Fig. 2.5: Flow chart of BM-3D method

2.7.2 Step 2: Final estimate

Using the basic estimate, perform improved grouping and collaborative wiener filtering.

a) Block-wise estimates: For each block, the following are to be done

i) Grouping: BM is to be used within the basic estimate to find the locations of the blocks similar to the currently processed one. Using these locations, two groups (3D arrays) shall be formed, one from the noisy image and one from the basic estimate.

ii) Collaborative wiener filtering: A 3D transform shall be applied on both groups. Wiener filtering is to be performed on the noisy image using the energy spectrum of the basic estimate as the true (pilot) energy spectrum and estimates of all grouped
blocks are to be produced by applying the inverse 3D transform on the filtered coefficients and return the estimates of the blocks to their original positions.

b) Aggregation: A final estimate of the true image shall be computed by aggregating all of the obtained local estimates using a weighted average.

2.8 COMPRESSIVE SENSING 3D METHOD

The compressive sensing (CS) theory proved that any sparse signal or image can be reconstructed from samples fewer than number of elements in a signal or image. These subsets are taken as measurement vectors in CS framework to obtain multiple SAR images by solving convex optimization problem. The pixel-wise averaging of multiple compressive reconstructed images would lead to better results compared to conventional despeckling techniques. In this work, selective 3 dimensional (3D) filtering of multiple reconstructed images shall be employed to further improve despeckling results [88-91].

This despeckling framework is comprised of three major steps; selection of subsets of pixels from SAR images, reconstruction of image from each subset of pixels using CS theory [92-94] and statistical combining of multiple reconstructed images by employing selective 3D filtering as shown in figure 2.6.

![Flow chart of CS-3D algorithm](image)

Fig. 2.6: Flow chart of CS-3D algorithm

In order to formulate multiple partially overlapped subsets of pixels from m×n SAR image, 'x', first of all SAR image is divided into P mutually exclusive (ME) groups of pixels. The pixels in each group follow regular pattern, and pixels of same
group are separated by fixed distance in horizontal and vertical directions. In figure 2.7, the pixels belonging to each group are represented by a unique symbol. These groups do not overlap and each group contains distinct pixels. The distance between pixels of same group in horizontal and vertical directions are $h$ and $v$ respectively. In figure 2.7, $h$ is 4, $v$ is 2 and $P = h \cdot v = 8$. Each group contains $S = N/P$ (where $N = m \times n$) pixels. If image is divided into more number of groups, the distances between pixels of same groups increase and vice versa.

Fig. 2.7: Selection of multiple mutually exclusive subsets of pixels.

Let $p$-th group of pixels be $u_p$ with $p = 1, 2, \ldots, P$. If pixels in $u_p$ are selected in regular pattern as shown in figure 2.7, it can be given by following expression by taking leverage from Matlab syntax

$$u_p = x(\eta : v : m, \tau : h : n)$$

(2.16)

where

$$\eta = 1, 2, \ldots, v \text{ and } \tau = 1, 2, \ldots, h$$

and

$$p = \tau + (\eta - 1)h$$

(2.17)

with $p \in \{1, 2, \ldots, P\}$ and maximum value of $p = P (= v \cdot h)$ when $\eta = v$ and $\tau = h$. Similarly, vectors containing indices of pixels in original SAR image corresponding to $u_p$ can be given as

$$\lambda_p = \{(i-1)n + j \mid i = \eta:v:m \text{ and } j = \tau:h:n\}$$

(2.18)
where \( p, \eta, \tau \) are same as for (2.16). An example of selection of \( u_p \) for
\[
h = 4 \text{ and } v = 2 \text{ is given as } \\
u_1 = \{ x(i, j)|i = 1 : 2 : m; j = 1 : 4 : n \} \\
u_2 = \{ x(i, j)|i = 1 : 2 : m; j = 2 : 4 : n \} \\
u_3 = \{ x(i, j)|i = 1 : 2 : m; j = 3 : 4 : n \} \\
u_4 = \{ x(i, j)|i = 1 : 2 : m; j = 4 : 4 : n \} \\
u_5 = \{ x(i, j)|i = 2 : 2 : m; j = 1 : 4 : n \} \\
u_6 = \{ x(i, j)|i = 2 : 2 : m; j = 2 : 4 : n \} \\
u_7 = \{ x(i, j)|i = 2 : 2 : m; j = 3 : 4 : n \} \\
u_8 = \{ x(i, j)|i = 2 : 2 : m; j = 4 : 4 : n \}
\]
(2.19)

If \( Q \) number of mutually exclusive groups of pixels are selected from \( P \) number of ME groups, total number of all possible subsets can be found by following expression
\[
T = P! / (Q! (P-Q)!) 
\]
(2.20)
where symbol '!' represents factorial.

The SAR image can be reconstructed from each subset \( y_t, t \in \{1, T\} \) using CS theory [95-97] by considering \( y_t \) as compressive samples. The compressive sampling equation (2.16) becomes
\[
y_t = S_t x_t 
\]
(2.21)

\( S_t \) is \( t \)-th sampling matrix corresponding to \( y_t \). It is \( M \times N \) matrix produced by taking exactly one '1' in each of \( M \) rows corresponding to the location of pixel of \( y_t \) with in SAR image.

In mathematical form, despeckled image, \( \hat{X} \) can be written as
\[
\hat{X} = \text{MEDIAN} \left( \{ x_t(k_1, k_2) \}; \ t \in \{1, T\}\right) \\
\quad k_1 \in \{i-R, i+R\} \\
\quad k_2 \in \{j-R, j+R\} \text{ if } G(i,j) < \text{TH}
\]
and
\[
\hat{X} = (1/T) \cdot \sum_{t=1}^{T} x_t(i,j) \quad \text{otherwise} 
\]
(2.22)
where MEDIAN represents the median operation, \( T \) is the number of reconstructed images, and \( 2R + 1 \) is the size of selected window for each image.
2.9. QUALITY MEASUREMENT

The following parameters will give us about a method how far that method can despeckle the noise and preserve the textures. Based upon the image and method used, the results vary. Equivalent number of looks plays major role to judge whether the despeckle method is good or not. Generally mean square error and peak signal to noise ratio values play a major decision to say whether the image denoising method is up to the mark or not.

2.9.1. Mean Square Error (MSE): The Mean squared error (MSE) measures the average absolute difference between two images. If the MSE value is small, it is nearer to the original image

\[ \text{MSE} = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} (x - \bar{x})^2 \]  

(2.23)

Where \( x \) and \( \bar{x} \) are the original and despeckled SAR images respectively. If this value is low, the PSNR value will be high and vice versa. Its value must be minimum as much as possible.

2.9.2. Peak Signal to Noise Ratio (PSNR): PSNR is the ratio between the maximum signal power and the corrupting noise power. It is the factor that judges whether a method is providing good denoising scheme or not. Its value must be maximum as much as possible. The higher the value means the higher the image quality will be. It plays a decisive role in all image processing areas.

\[ \text{PSNR} = 10 \log_{10} \left( \frac{2^n - 1}{\text{MSE}} \right) \]  

(2.24)

2.9.3. Equivalent Number of Looks (EQNL or ENL): EQNL value plays crucial role in coherent systems like SAR processing. The EQNL values speak about the efficiency in smoothing speckle noise of image over homogeneous areas. It is given by,

\[ \text{EQNL} = \left( \frac{\text{mean}}{\text{standard deviation}} \right)^2 \]  

(2.25)

Measuring of EQNL values in uniform area of SAR image is very important. A larger value of EQNL usually corresponds to a better quantitative performance. The
EQNL value depends on the size of the image area that is to be measured. If the image area is high, the EQNL value will be high and vice versa.

2.9.4. **Coefficient of Correlation (CC):** Correlation coefficient gives how far the two images correlated to each other, that means how far the despeckled image is near by the original image. If this value is near to 1, it means there exists good correlation between the original image and despeckled image.

\[
C_{x, \hat{x}} = \frac{E[(x - \mu_x)(\hat{x} - \mu_{\hat{x}})]}{\sigma_x \sigma_{\hat{x}}}
\]  
(2.26)

Where \( \mu_x \) and \( \mu_{\hat{x}} \) are mean values of original and despeckled SAR images respectively and \( \sigma_x \) and \( \sigma_{\hat{x}} \) are standard deviations of original and despeckled images respectively.

2.9.5. **Speckle Suppression Index (SSI):** The Speckle Suppression Index is given as the ratio of coefficient of variance of speckle removes SAR image to the coefficient of variance of original SAR image and is given below

\[
SSI = \left( \frac{\text{var}(\hat{x})}{\text{mean}(\hat{x})} \right)^{1/2} \left[ \frac{\text{mean}(x)}{(\text{var}(x))^{1/2}} \right]
\]  
(2.27)

It should be less than 1. Lower the value means higher the speckle reduction. Ideal value is zero. Let \( x \) and \( \hat{x} \) be original and the speckle eliminated SAR images respectively.

2.9.6. **Edge Save Index or Edge preserve Index (ESI or EPI):** Edge Save Index corresponds to the edge preservation capability of SAR image of the despeckling technique.

\[
ESI = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n-1} |\hat{x}(i,j+1) - \hat{x}(i,j)|}{\sum_{i=1}^{m} \sum_{j=1}^{n-1} |x(i,j+1) - x(i,j)|}
\]  
(2.28)

Where \( \hat{x} \) is despeckled SAR image, \( x \) is original SAR image, \( m \) is the number of rows in SAR image and \( n \) is the number of columns in SAR. If this value is more, the preservation of the edges of the image is more. It should be high. A better visible image is always having reasonable good ESI value.