4. DESIGN OF PREPROCESSING ALGORITHM

The preprocessing stage of is an essential initial step in the proposed patterned fabric defect detection system. This phase focuses on techniques that enhance the patterned fabric image. The main objective of this step is to remove noise introduced during acquisition, while retaining as much as possible the important signal features of the fabric image.

Denoising techniques is considered a challenging research area because the process of denoising is irreversible. Therefore, care should be taken while removing noise, as some techniques mistakenly might remove important regions of an image as noise. For example, a small thread hole should not be considered as noise.

Low image quality is an obstacle for effective extraction of features, texture analysis and defect recognition. Image variances or impulse noise inherently exists in the acquired images and degrades the quality of the fabric images. According to Raman and Himanshu (2010), it is very important to reduce the effect of noise in images, before it can be used for analysis and recognition.

One problem in processing patterned fabric images is the presence of impulse noise (Salt & Pepper) which appears as bright dots or dust particles over the image. They are normally distributed randomly all over the image and hence, its presence affects image interpretation by human and the accuracy of computer-assisted detection techniques.

During image acquisition, inhomogeneities occur due to variance in relative position of the light source, camera position and the textile position. These inhomogeneities make some part of the image appear darker and many have uneven contrast. Moreover, the presence of impulse noise (appearing as bright dots or dust particles) also degrades and distorts the images. Impulse noise can be fixed-valued (salt and pepper) or random-valued noise. Both of these can be mistakenly identified as defect pattern and therefore has to be removed.

Presence of noise degrades spatial and contrast resolution and obscures the underlying structure of an image. Further, it has a negative impact on fabric imaging where the presence of noise shows a reduction of surface detectability of approximately a factor of eight (Bamber and Daft, 1986). This radical reduction in contrast resolution prevents automatic defect recognition and texture analysis algorithm to perform efficiently and gives the image a grainy appearance. Hence, denoising is considered as a critical pre-processing step by many texture analysis and imaging systems, defect detection and inspection systems.

Denoising algorithms for salt and pepper noise detection and removal is an area of research work that has attracted many researchers (Mélange *et al.,* 2011; Mohammad *et al.,* 2011; Hao *et al.*, 2012). An effective noise reduction method for this type of noise involves the usage of any one of the several types of available noise reduction techniques. They include median filter, Vector Median Filter (VMF), anisotropic diffusion filter and wavelet-based filters. All of these methods have been enhanced and optimized for noise removal.

Among the various proposed methods, the median filter is one of the most commonly used non-linear filters. It has already been established that median filters are more efficient in removing salt and pepper noise and are computationally inexpensive algorithms. However, it also has the drawback of smearing detailed regions like edges of the original image.

Several methods have been proposed to solve this problem and they include adaptive filter (Manikandan *et al.*, 2004; Kalavathy and Suresh, 2011), multistate median filter (Chen and Wu, 2001a), weighted median filter (Yang *et al.*, 1995) and switching median filters (Ping *et al.*, 2007). Vector directional filters uses directional image vectors during denoising (Lukac, 2004).

Variations to vector directional filters are the weighted vector direction filter which implement a tracking algorithm to identify the varying signal and noise statistics. Peer Group Filters (PGF) that uses statistical properties of accumulated distances for vector median filtering has also been proposed

(Smolka, 2008). This algorithm switches between vector median and the original central pixel.

Nikolova (2004) and Wang and Zhang (1999) proposed methods which first identified the noisy pixels and then replaced them by using the median filters or its variants. The other pixels are left unchanged. This method had the disadvantage that the noise pixel replacement procedure only considered its neighbouring pixels and did not consider the presence of edges. To avoid smearing in detailed regions, the Switching median filter was modified to include a center weighted median filter which used two thresholds to make the decision of replacement.

The work of Chen and Wu (2001) improved the work of center weighted median filter by including more threshold values. Similarly, Zhang and Karim (2002) used a Laplacian edge detector and the detected edges were preserved during noise removal. Kang and Wang (2009) proposed a rank-order-based switching median filter to solve the problems posed by threshold selection. This work is enhanced in this study to include an adaptive center-weighted median filter, edge preservation step and using a noise detection algorithm to improve visual quality of the 2D patterned fabric image.

This phase of the study presents a noise removal algorithm to solve the problem of fabric image degradation. The proposed algorithm is an unified model termed as Enhanced Directional Switching Median Filter (EDSMF) and considers Salt & Pepper noise. This method is an improved version of the filter proposed by Kang and Wang (2009) (KWF method) who used a modified version of Switching Median Filter (SMF) based on the rank order arrangement to implement impulse noise removal. The KWF method, when used with patterned fabrics, sometimes produces excessive smoothening in highly textured area. Moreover, the visual quality of the denoised image is still low. To solve these problems, the first phase of the proposed patterned fabric fault detection system, the KWF method is enhanced to use an improved directional adaptive criterion that can distinguish between texture and noisy regions and then use a switching filtering only to noisy regions.

The chapter begins with a formal discussion on the fundamentals of noise, a brief discussion on traditional switching median filter followed by the methodology used in the research work and experimental results.

4.1. NOISE IN PATTERNED FABRIC IMAGES

Image noise is defined as the random variation of brightness or color information in images produced by line or CCD or web camera. Image noise is generally regarded as an undesirable by-product during image acquisition. The need for efficient image restoration methods has grown with the massive production of digital images.

An image improvement step is always desirable to extend their range of action. An image is, in general, encoded as grey level, which is composed of pair of values of the form (i, u(i)), where u(i) is the value of the pixel. In the case of grey level images, ‘i’ is a point on a 2D matrix and u(i) is a real value.

The two main limitations in image accuracy are categorized as noise and blur. Blur is intrinsic to image acquisition systems, as digital images have a finite number of samples and must respect the Shannon-Nyquist sampling conditions (Shannon and Weaver, 1998).

The second main image perturbation is noise. Each one of the pixel values u(i) is the result of a light intensity measurement, usually made by a CCD matrix coupled with a light focusing system. Each captor of the CCD is roughly a square in which the number of incoming photons is being counted for a fixed period corresponding to the obturation time.

When the light source is constant, the number of photons received by each pixel fluctuates around its average in accordance with the central limit theorem. In other terms, fluctuations of order $\sqrt{n}$ for n incoming photons can be expected. In addition, each captor, if not adequately cooled, receives heat spurious photons. The resulting perturbation is usually called "obscurity noise".

The denoising task is considered as the problem of estimating the noise model in a fabric image using which the best method of restoration can be designed. In general, it consists of using a filtering algorithm for this purpose. The filtering algorithm has to be selected carefully with the aim of constructing a denoised image that is as close to the noise free image (Figure 4.1).

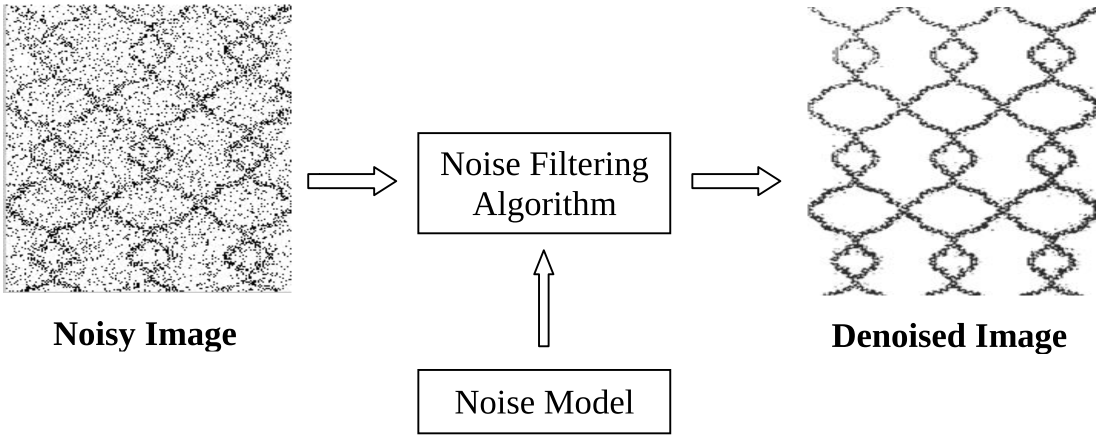


Figure 4.1 : General Noise Removal Process

In an approximation, the noisy image can be modelled as in Equation (4.1)

$$G(x, y) = F(x, y) + \eta(x, y) \tag{4.1}$$

where F(x, y) is the original image pixel ('true' value of pixel), $\eta(x, y)$ is the noise perturbation and G(x, y) is the resulting noisy pixel,with corrdinates x, y. The amount of noise is signal-dependent, that is n(i) is larger when u(i) is larger.

Several types of noise exist and the most common noise found in fabric images is the impulse noise. Examples include Gaussian Noise, Speckle Noise, Impulse Noise, Poisson Noise and Uniform Noise. Out of these, fabric images are often degraded by the presence of impulse noise during acquisition. Details regarding impulse noise are presented in the following paragraphs.

Impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel. Two common types of impulse noise are the Salt & Pepper noise and the random-valued noise. For images corrupted by Salt & Pepper noise (respectively, random-valued noise), the noisy pixels can take only the maximum and the minimum values (respectively, any random value) in the dynamic range.

Salt and pepper noise is the most frequently occurring type in an image. This kind of noise is introduced into images in situations where quick transients, such as faulty switching, take place. It represents itself as randomly occurring white and black pixels. It is also called as “impulsive noise” or “spike noise” (Gonzalez and Woods, 2007).

An image containing Salt & Pepper noise will have dark pixels in bright regions and bright pixels in dark regions. Both are considered to be more serious than all the others, as they cause difficulties in image interpretation. This type of noise can be caused by dead pixels, analog-to-digital converter errors and bit errors in transmission (Shapiro and Stockman, 2001; Boncelet, 2005). Some examples of patterned noiseless fabrics and its noisy counterparts are shown in Figure 4.2.

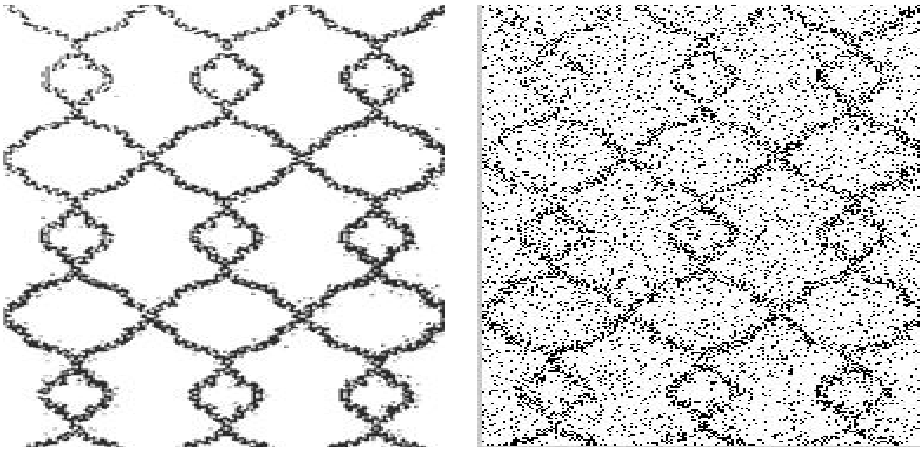
4.2. PROPOSED PREPROCESSING METHOD

Two main objectives of the proposed preprocessing algorithm are

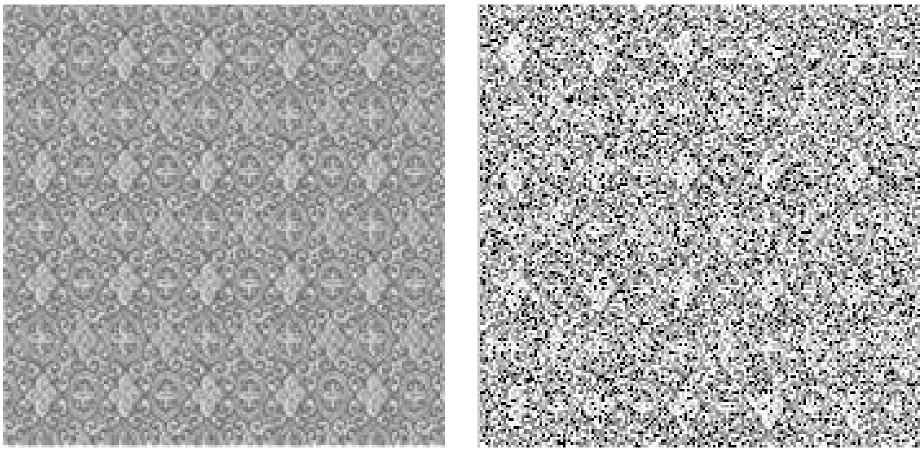
- Eliminate or reduce inhomogeneities introduced due to contrast and lighting variations.

- Eliminate or reduce noise accumulated during acquisition.

Noise-free Image **Noisy Degraded Image**



(a)



(b)

Figure 4.2 : Impulse Noise Images

The sources of the first problem are the relative positions of the light source, the camera and the textile, which results in two problems:

- Some parts of the image appear darker than other parts which mean that the local mean gray level is not be constant all over the image.

- The same effect for the contrast can also be observed in the patterned fabric image. The difference of the gray level in neighboring strips (of the pattern) may not be constant over the image

As described earlier, noise in image has a negative impact on detectability and interpretability of an image. To solve these problems, the proposed algorithm performs preprocessing that enhances the input patterned fabric image by applying techniques that perform noise removal.

The proposed methodology behind the EDSMF denoising algorithm is diagrammatically presented in Figure 4.3. The noise removal part of EDSMF method consists of three main steps.

1. Impulse noise detection using enhanced directional detector from the noisy image X.
2. Create binary image B that identifies noisy pixels and noise free pixels.
3. Perform modified adaptive directional switching median filter to remove noise.

The proposed EDSMF method enhances the traditional Vector Median Filter (VMF) by combining it with adaptive directional detector for impulse noise removal in noisy fabric image. This section presents a brief description on the traditional vector median filter method for denoising, followed by the proposed method.

4.2.1. Traditional Vector Median Filter

All denoising methods depend on a filtering parameter ‘h’. This parameter measures the degree of filtering applied to the image. For most methods, the parameter ‘h’ depends on an estimation of the noise variance σ^2. The result of a denoising method D_h can be defined as a decomposition of any image ‘v’ as given in Equation (4.2).

$$w = D_h v + n(D_h, v) \quad (4.2)$$

where $D_h v$ is more smooth than v and $n(D_h, v)$ is the noise predicted by the method.

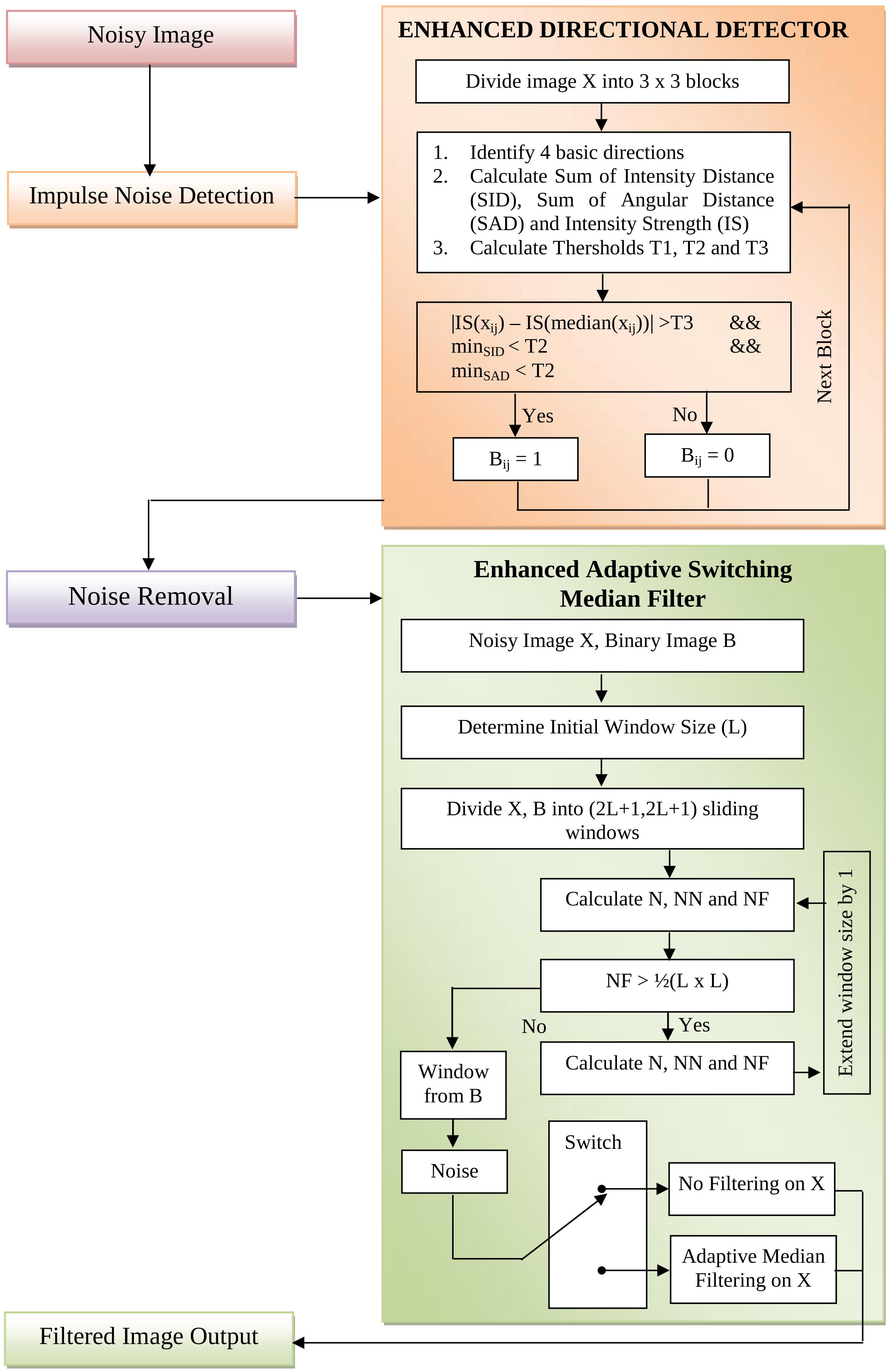


Figure 4.3 : Enhanced Directional Switching Median Filter (EDSMF)

In the present environment, it is needed to make sure that the denoising method smoothens v and also to make sure that the contents lost due to noise are recovered. For this purpose, the study uses a variant of median filter, called vector median filter and combines it with directional noise detection and adaptive switching vector median filter.

Median filter, a non-linear filtering technique, uses a window that moves over a signal and at each point, the median value of the data within the window is taken as the output. The impulse response of the median filter is zero and thus makes its use attractive to suppress impulsive noise. Median filters are robust and are well-suited for data smoothing when the noise characteristics are not known and also has the capability to preserve edges.

Using these vector signal properties, the VMF are built. In the vector median approach, the samples of the vector-valued input signal are processed as vectors as opposed to component-wise scalar processing. The vector median operation inherently utilizes the correlation between the signal components giving the filters some desirable properties.

A vector median filter is defined as the vector that corresponds to the minimum sum of distances to all other vector pixels. The selection of the pixel with minimum sum of distance may be readily visualized as finding the pixel nearest to the ‘center’ of the pixels within the neighborhood viewed as a cluster in the gray level space.

The VMF algorithm has three main steps. The first step, after dividing an image into fixed-equal sized windows, computes the Euclidean distance from every pixel to every other pixel in its neighborhood in the current window chosen. The algorithm, in the next step, arranges the vector pixels of this window in ascending order on the basis of the sum of distances. The ordering used for sum of distances is associated with the vector pixels also. The vector pixel with the smallest sum of distances is the vector median pixel. The vector median filter is represented using Equation (4.3).

$$X_{VMF} = \text{vectormedian(window)} \tag{4.3}$$

If δ_i is the sum of the distances of the i^{th} vector pixel with all the other vectors in the kernel and is calculated using Equation (4.4).

$$\delta_i = \sum_{n=1}^{N} d(x_i, x_n) \tag{4.4}$$

where $d(X_i, X_n)$ represents a distance measure between the i^{th} and the n^{th} neighboring vector pixels with $(1 \leq i < N)$ and X_i and X_N are vectors with N=9. The ordering may be illustrated using Equation (4.5) and this implies the same ordering to the corresponding vector pixels (Equation 4.6). In these equations, the subscripts represent the ranks.

$$\delta_{(1)} \leq \delta_{(2)} \leq \ldots \leq \delta_{(9)} \tag{4.5}$$

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(9)} \tag{4.6}$$

Since the vector pixel with the smallest sum of distances is the vector median pixel, it will correspond to rank 1 of the ordered pixels (Equation 4.7).

$$X_{VMF} = X(1) \tag{4.7}$$

The steps are consolidated in Figure 4.4. The VMF is highly effective in removing impulsive noise but also has the following disadvantages.

- It fails to distinguish thin lines and boundaries from impulsive noise and usually filters them out because it interprets these fine details as some noise.
- High computation cost – due to repeated distance calculation of similar values in the filtering window. These can be removed to reduce calculation cost.
- More than one pixel derives the minimum distance thereby more than one qualified pixel to replace the center pixel.

Step 1	**:**	Calculate Vector Median as sum of distances from every pixel to every other pixel in its neighbourhood in the filtering window.
Step 2	**:**	Select the pixel with minimum distance as vector median of that window.
Step 3	**:**	Replace noise pixel with vector median

Figure 4.4 : Steps in Vector Median Filter

The enhanced version solves both these problems by using a procedure to differentiate edges / boundaries from other regions and applying VMF only to other regions. The problem is solved by minimizing the repeated calculations involved. This is performed by using a procedure that calculates the minimum distance in a fast manner and applying VMF only to those pixels that are affected by impulse noise. The problems of more than one pixel having the same minimum distance and distortion are solved by using a simple rule-based distance calculation. The enhanced version of VMF has the advantage of edge preservation and reduction of computation complexity.

4.2.2. Impulse Noise Detection

In the Salt & Pepper noise model, only two possible values are, a and b, and the probability of obtaining each of them is less than 0.1 (otherwise, the noise would vastly dominate the image). For an 8-bit/pixel image, the typical intensity value for pepper noise is close to 0 and salt noise is close to 255 (Figure 4.5).

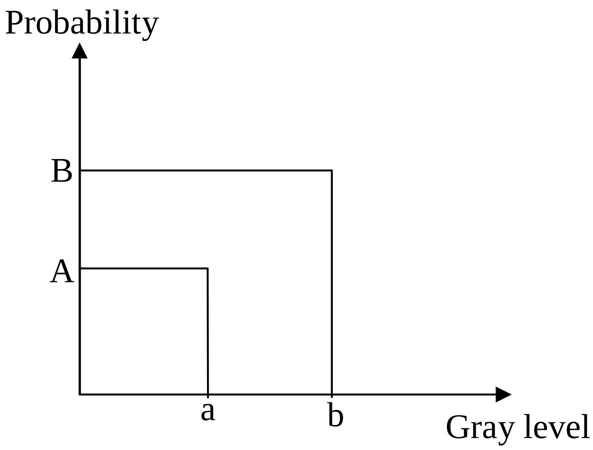


Figure 4.5 : Probability density function for Salt & Pepper Noise Model

Let X be the input noisy image. The first step of the algorithm is to identify noisy and noise free pixels in the image. Consider a sliding window of a noisy image X of size 3 x 3 with current pixel coordinates (i, j). In the proposed filter, four directions are considered, namely, TLBR (Top Left-Bottom Right), TCBC (Top Centre-Bottom Centre), TRBL (Top Right-Bottom Left), CRCL (Center Right - Centre Left) (Figure 4.6).

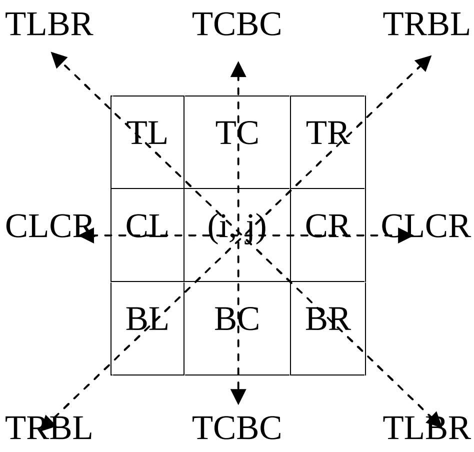


Figure 4.6 : Directions Considered

In the next step, two measures namely, Sum of Intensity Distance (SID) and Sum of Angular Distance (SAD) are calculated for each direction. The SIDs for the four directions are calculated using Equations (4.8) to (4.11).

$$SID_{ij}(TLBR) = |L(x_{ij}) - L(TL_{ij}) + |L(x_{ij}) - L(BR_{ij})| \quad (4.8)$$

$$SID_{ij}(TRBL) = |L(x_{ij}) – L(TR_{ij})+ |L(x_{ij}) – L(BL_{ij})| \quad (4.9)$$

$$SID_{ij}(TCBC) = |L(x_{ij}) – L(TC_{ij})+ |L(x_{ij}) – L(BC_{ij})| \quad (4.10)$$

$$SID_{ij}(CLCR) = |L(x_{ij}) – L(CL_{ij}) + |L(x_{ij}) – L(CR_{ij})| \quad (4.11)$$

where L(.) is the brightness value of pixel x. The SAD of two pixels x_1 and x_2 can be calculated using Equation (4.12).

$$\arccos\left(\frac{x_1 x_2}{\sqrt{x_1^2}\sqrt{x_2^2}}\right) \quad (4.12)$$

Using the above equation, the SAD of the four directions are calculated using Equation (4.13) to (4.16).

$$SAD_{ij}(TLBR) = AD(x_{ij}, TL_{ij}) + AD(x_{ij}, BR_{ij}) \quad (4.13)$$

$$SAD_{ij}(TRBL) = A(x_{ij}, TR_{ij}) + AD(x_{ij}, BL_{ij}) \quad (4.14)$$

$$SAD_{ij}(TCBC) = AD(x_{ij}, TC_{ij}) + AD(x_{ij}, BC_{ij}) \quad (4.15)$$

$$SAD_{ij}(CLCR) = AD(x_{ij}, CL_{ij}) + AD(x_{ij}, CR_{ij}) \quad (4.16)$$

Identify the minimum SID and SAD from the calculated values Let this be denoted min_{SID} and min_{SAD}. If min_{SID} is less than a threshold T1 and min_{SAD} is less than another threshold T2, then the pixel x_{ij} is treated as a noisy pixel, else it is treated as a normal uncorrupted pixel.

4.2.3. Threshold Estimation

The two thresholds T1 and T2 are calculated using the Genetic Algorithm (GA) proposed by Goldberg (1989). GAs provide a learning method inspired by evolutionary biology. GAs are the most popular class of evolutionary algorithms that use mechanisms such as reproduction, mutation, crossover (also called recombination), natural selection, and survival of the fittest to simulate biological evolution (Holland, 1992).

Genetic algorithms have been successfully applied to a wide variety of scientific and engineering optimization or search problems. They can search spaces of hypotheses containing complex interacting parts, where the impact of each part on an overall hypothesis is difficult to model (Mitchell, 2002). The relative insensitivity of GAs to noise and the requirement of no domain knowledge make them a powerful tool to optimize the process of classification, especially when the domain knowledge is costly to exploit or unavailable (Vafaie and Jong, 1992).

Genetic algorithms begin the search for solutions in a population of initial hypotheses that traditionally are generated at random. Each hypothesis, called an individual or a chromosome, represents a potential solution of the problem. Individuals are encoded as bit strings whose interpretation depends on applications. Typically, individuals are represented in binary as strings of 0's and 1's. The initial population then evolves in generations. In each generation, every individual of the current population is evaluated according to the fitness function F, which is a predefined numerical measure for the problem at hand. A new population is generated by stochastically selecting the current fittest individuals. Some of the selected individuals are modified to produce new offspring individuals by mutating and recombinating parts of them.

Some of these selected individuals are passed to the next generation intact. The new population is then used in the next iteration of the algorithm. Random search strategies powered by the genetic operators (mutation and crossover) are designed to move the population away from local optima where many algorithms (e.g., greedy hill climbing) face hindrance. In the GA based image fusion and selection method, there are several operations that need to be determined. They are chromosome encoding and fitness function.

In chromosome encoding, a binary encoding scheme is used where a binary bit string represents an individual. Each individual represents a feature subset. The individuals are encoded by L-bit binary vectors. The bit with value 1 in a vector represents the corresponding feature being selected, while the bit with value 0 means the opposite. The length of each chromosome is determined by the number of features N.

Thus, in the encoding scheme used, the chromosome is a bit string whose length is determined by the number of parameters in the image. Each parameter is associated with one bit in the string. If the i^{th} bit is 1, then the i^{th} parameter is selected, otherwise, that component is ignored. Each chromosome thus represents a different parameter subset. In order to solve the problem of threshold value selection, the chromosomes are represented as an element vector with two values {T1, T2}. The genetic algorithm is designed to optimize two objectives:

(i) maximize classification accuracy of the feature subset

(ii) minimize the number of features selected.

For this purpose, the Mean square error (MSE) between the original image and the restored image is used as a fitness function. MSE is calculated using Equation (4.17).

$$\mathrm{MSE}(Z,Y) = \frac{1}{3}\sum_{i=1}^{N}\sum_{k=1}^{3}\left(Z_i^k - Y_i^k\right)^2 \qquad (4.17)$$

where Z_i^k and Y_i^k are the k^{th} component value of i^{th} pixel in the original image and the restored image respectively. Here, N is the total number of pixels in an image. During the process of genetic algorithm operation the value of fitness function gradually decreases while the number of generations grows. The genetic algorithm based threshold estimation procedure is given in Figure 4.7. The process when applied on the test fabric images of Figure 4.2 is presented in Figure 4.8.

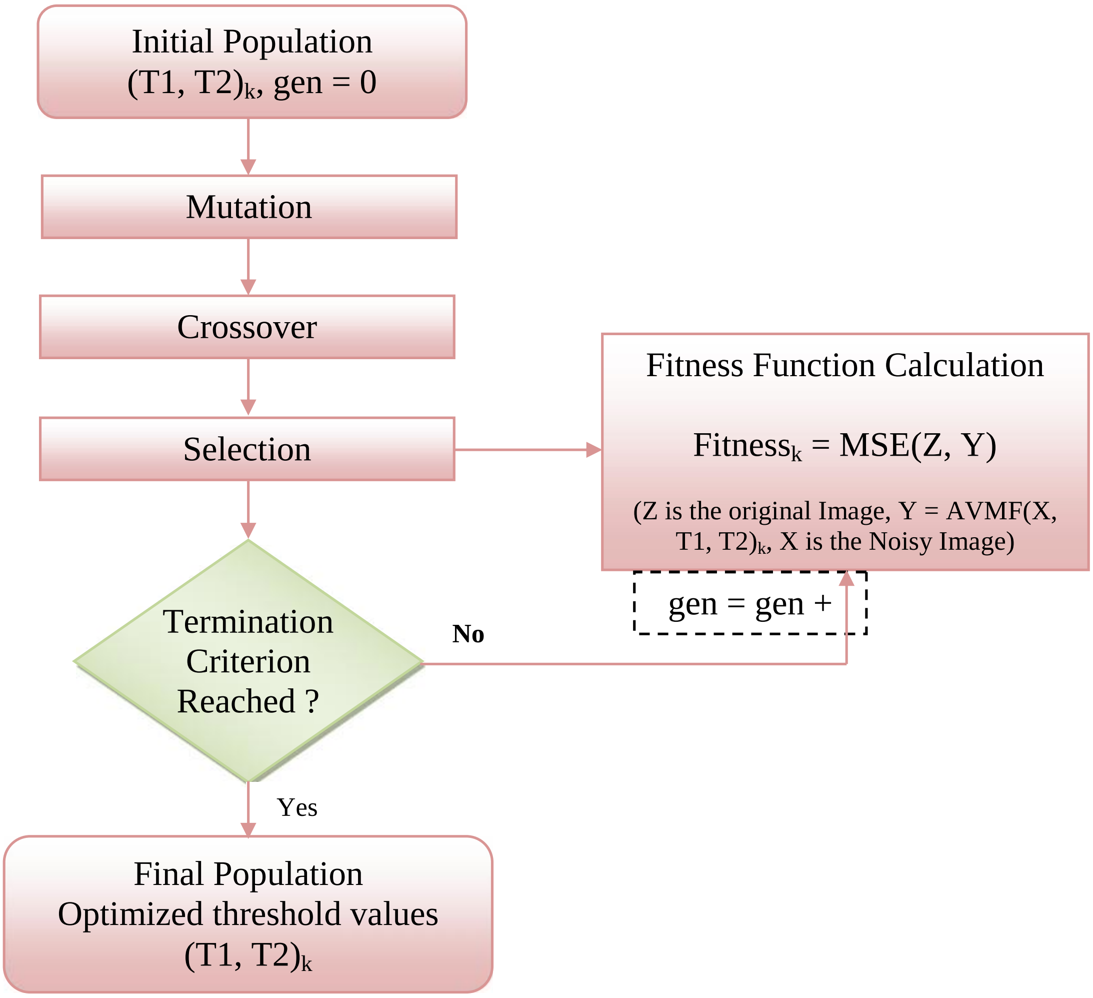
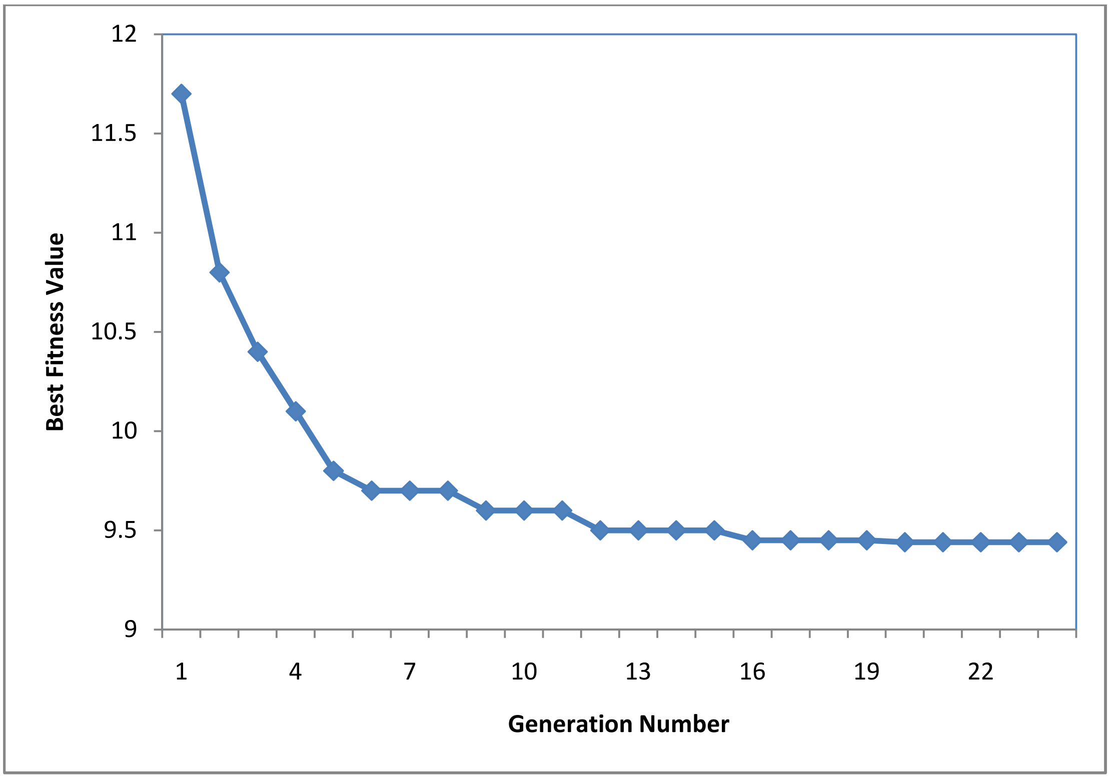
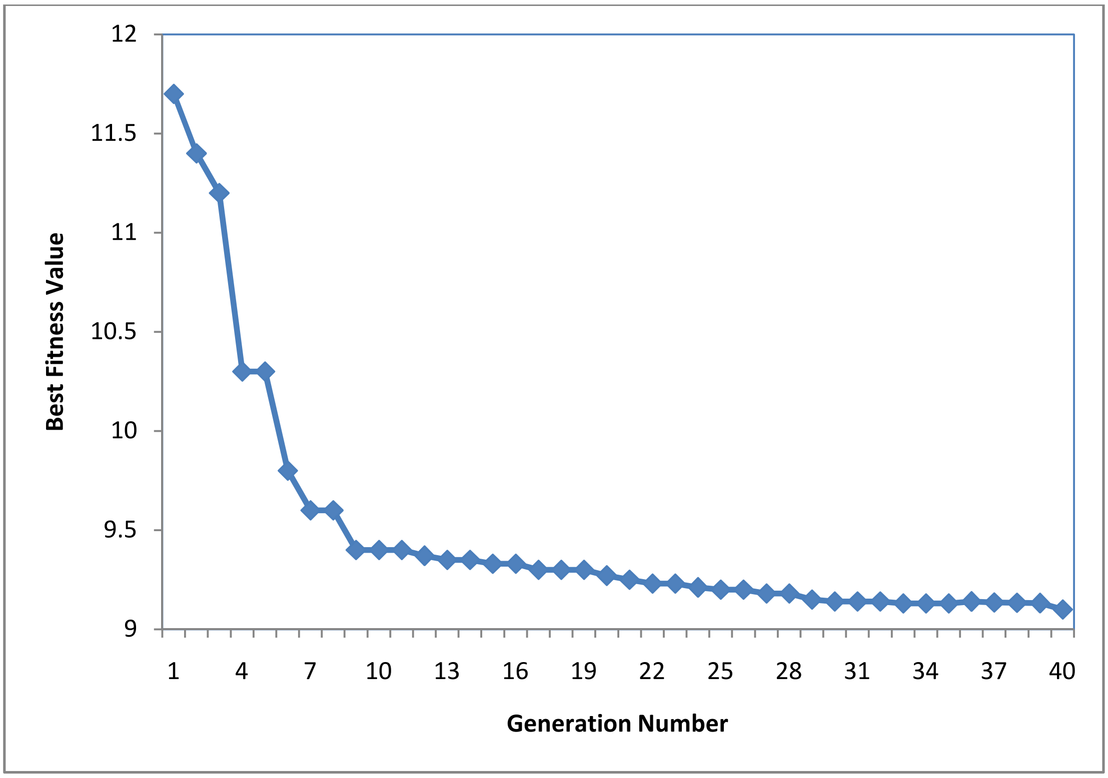


Figure 4.7 : Optimized Threshold Value Estimation Procedure

The analysis of the dependencies shows that during the process of genetic algorithm operation and hence during the growth of the number of generations the value of fitness function gradually decreases, reaching approximately 20% relative to its initial value. And it is possible to achieve this result after only 15-20 generations. Performed experiments show that optimized parameters of the filter, averaged on several runs of genetic algorithm, differ insignificantly from each other for different test images. Thus, in the proposed denoising algorithm, the optimized parameters acquired on test image Figure 4.2b are used.



Convergence of GA Optimization Procedure for Figure 4.2a



Convergence of GA Optimization Procedure for Figure 4.2b

Figure 4.8 : Convergence Behaviour of GA Optimization Procedure

4.2.4. Formation of Binary Image

The genetic-based threshold calculation algorithm has the drawback that when the impulse noise is uniformly distributed, both T1 and T2 are very high. It is a well-known fact that high threshold values cannot detect noisy pixels correctly and often misclassify them as noise free pixels. Hence, a small threshold value is desired. But with a low threshold value the number of noise free pixels preserved reduces. For this reason, there exists a trade-off in the selection of threshold values and optimal threshold should detect maximum noisy pixel while preserving all noise free pixels. For this purpose, in the proposed algorithm, a third threshold value, T3, is used. T3 is calculated by sorting the pixels in the windows in ascending order of intensity excluding the central pixel x_{ij}. The index position indicates the Intensity Strength (IS) of the pixel using which the second step of noise detection is performed (Equation 4.18).

$$B_{ij} = \begin{cases} 1 & |IS(x_{ij}) - IS(median(x_{ij}))| > T3 \\ & \&\&\ min_{SID} < T1\ \&\&\ min_{SAD} < T2 \\ 0 & otherwise \end{cases} \qquad (4.18)$$

Here, T3 is calculated using Equation (4.19).

$$T3 = 0.5(4L^2 + 4L + 3) \cdot p(1 - p) \qquad (4.19)$$

where $(4L^2 + 4L + w)$ is the number of the sorted sample data including the current pixel $x_{i,j}$ which multiples 3 times and p is the noise density. This formula works since theoretically, the noise density in the entire image is identical to the noise density in the sliding window. For example, if p=40% and w=1, then T3 is set as 3 in the 3×3 sliding window. Thus, the result of noise detection is a binary image with a value zero indicating noise free pixels and a value 1 indicating a noisy pixel and is denoted as image B.

4.2.5. Enhanced Directional Switching Median Filter

The next stage of the algorithm considers the binary image from the previous step and performs median filtering only for the noisy pixels. This process is called as the switching median filter. In this study, an directional switching median filter is used.

The VMF algorithm proposed in the previous section (Section 4.2.1) has the drawback of excessive smoothening during noise removal process. In order to solve this problem and to improve the visual quality of the image the switching median filter was modified to use an adaptive switching concept that switches to noise removal only when noisy regions are detected. The steps involved are given below.

Step 1 : Determine initial window size, L, using Equation (4.20) where ND is the noise density.

$$L = \begin{cases} ND \le 40 & 1 \\ 41 \le ND \le 60 & 2 \\ ND \ge 61 & 3 \end{cases} \tag{4.20}$$

Step 2 : This step starts by dividing both the noisy image X and its corresponding binary image B into (2L+1 x 2L+1) sliding windows

Step 3 : Calculate the total number of pixels (NP) number of noisy pixels (NNP) and number of noise free pixels (NFP) in the current filtering window of X using B.

Step 4 : Repeat Step 4a until NFP > ½(L x L)

Step 4a : Extend window size by 1 on all the four sides.

Step 5 : Replace noisy pixel with the median of noiseless pixels.

4.3. CONCLUSION

The need for efficient image restoration methods has grown with the massive production of images produced by the state-of-the-art cameras. As the acquired images often have unwanted pixels introduced, the need for efficient denoising algorithms that can help researchers during analysis has also increased. In spite of various solutions being proposed, an efficient technique that meets all the demands of fabric imaging systems is still a very active research area. In the first phase of the proposed noise removal algorithm, a modified version of switching vector median filter that uses a preprocessing step that identified noise and noise free pixels and a noise removal step that uses an adaptive switching vector median filter was introduced. This step is used by the proposed non-motif and motif based algorithms. The design of the non-motif based algorithms is presented in the next chapter (Chapter 5, Design of Non-Motif-Based Algorithms).