Wavelet Shrinkage Based Image Denoising using Soft Computing

by

Rong Bai

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Noise reduction is an open problem and has received considerable attention in the literature for several decades. Over the last two decades, wavelet based methods have been applied to the problem of noise reduction and have been shown to outperform the traditional Wiener filter, Median filter, and modified Lee filter in terms of root mean squared error (MSE), peak signal noise ratio (PSNR) and other evaluation methods.

In this research, two approaches for the development of high performance algorithms for de-noising are proposed, both based on soft computing tools, such as fuzzy logic, neural networks, and genetic algorithms. First, an improved additive noise reduction method for digital grey scale nature images, which uses an interval type-2 fuzzy logic system to shrink wavelet coefficients, is proposed. This method is an extension of a recently published approach for additive noise reduction using a type-1 fuzzy logic system based wavelet shrinkage. Unlike the type-1 fuzzy logic system based wavelet shrinkage method, the proposed approach employs a thresholding filter to adjust the wavelet coefficients according to the linguistic uncertainty in neighborhood values, inter-scale dependencies and intra-scale correlations of wavelet coefficients at different resolutions by exploiting the interval type-2 fuzzy set theory. Experimental results show that the proposed approach can efficiently and rapidly remove additive noise from digital grey scale images. Objective analysis and visual observations show that the proposed approach outperforms current fuzzy non-wavelet methods and fuzzy wavelet based methods, and is comparable with some recent but more complex wavelet methods, such as Hidden Markov Model based additive noise de-noising method. The main differences between the proposed approach and other wavelet shrinkage based approaches and the main improvements of the proposed approach are also illustrated in this thesis.

Second, another improved method of additive noise reduction is also proposed. The method is based on fusing the results of different filters using a Fuzzy Neural Network (FNN). The proposed method combines the advantages of these filters and has outstanding ability of smoothing out additive noise while preserving details of an image (e.g. edges and lines) effectively. A Genetic Algorithm (GA) is applied to choose the optimal parameters of the FNN. The experimental results show that the proposed method is powerful for removing noise from natural images, and the MSE of this approach is less, and the PSNR of is higher, than that of any individual filters which are used for fusion.

Finally, the two proposed approaches are compared with each other from different point of views, such as objective analysis in terms of mean squared error(MSE), peak signal to noise ratio (PSNR), image quality index (IQI) based on quality assessment of distorted images, and Information Theoretic Criterion (ITC) based on a human vision model, computational cost, universality, and human observation. The results show that the proposed FNN based algorithm optimized by GA has the best performance among all testing approaches. Important considerations for these proposed approaches and future work are discussed.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor, professor M. E. Jernigan, for his advice and support during my stay at University of Waterloo. He provided me an excellent environment conducive to research, gave me carte blanche in determining research directions and helped me explore exciting vistas in image processing.

I am thankful to Dr. David A. Clausi and Dr. Daniel W. Stashuk for serving as my thesis readers and giving me many helpful suggestions on my thesis.

I thank Parthipan Siva (MASc., VIP lab, System Design Engineering), and Prabakar Puvanathasan (MASc. student, PAMI lab, Electrical and Computer Engineering), for their encouragement and support during my research.

I also thank Dr. Zhou Wang (Assistant Professor, Electrical and Computer Engineering), Di Zhang (Ph.D student, VIP lab, System Design Engineering), Dr. Nezamoddin N. Kachouie (VIP lab, System Design Engineering) for offering the testing code.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The ever-increasing number and variety of digital images generated everyday are becoming a major information source in daily life. Examples include natural images, digital commercial television, magnetic resonance images, as well as geographical information systems and astronomy. However, when images are created, transmitted and decoded, they are always distorted by different types of noise. Noise reduction has become a required step for any sophisticated algorithms in computer vision and image processing. A tradeoff between removing noise and blurring the image always exists. How to make a balance between denoising and blurring and obtain clean images has become a big issue in computer vision and image processing. This challenging issue has existed for a long time, yet there is no completely satisfactory solution.

This research only focuses on noise removal techniques for natural images.

1.1 Background

During the past several decades, considerable research has been done on de-noising. Different algorithms are used depending on the noise models. Most natural images are assumed to be corrupted by additive random noise, which usually is modeled as Gaussian noise.

There are many approaches to deal with additive noise in natural images, such as average filters and mean filters. Even though linear filters are useful in a wide variety of applications, there are some situations in which they are not adequate. For example, linear filters do not take into account any structure in images. Therefore, linear filters tend to blur sharp edges, destroy lines and other fine image details, and perform poorly in the presence of noise.

Nonlinear filters, however, can be successfully applied to achieve detail preserving noise reduction since they adopt the local features of an image. Non-linear spatial filters employ a low pass filtering on groups of pixels with the assumption

that the noise occupies the higher region of frequency spectrum. Low-pass filters will not only smooth away noise but also blur edges in images while the high-pass filters can make edges even sharper and improve the spatial resolution but will also amplify the noisy background.

Over the last two decades, wavelets have captured researchers' attention in image denoising due to their properties. Wavelets capability of extracting detailed spatial-frequency information is the main reason for this investigation. This property promises a possibility for better discrimination between the noise and the real data. Successful exploitation of the wavelet transform might minimize the blurring effect or even overcome it completely. Diverse algorithms for noise removal in the wavelet domain were introduced. Most existing denoising algorithms based on wavelets focus on exploiting the advantages of their multiresolution structure to capture more detailed information, or building wavelet coefficient statistical models to represent inter-scale dependencies and intra-scale correlations. It can not be denied that these algorithms can provide better performance than those algorithms using single layer wavelets and treating wavelet coefficients as independent.

However, as the complexities of the wavelet coefficient statistical models increase, the denoising performance is not improved as much as it is expected. Furthermore, the time and computational cost of building and training these statistical models are increasing rapidly. In recent years, researchers introduced artificial intelligence to wavelet based denoising methods since some soft computing tools, such as Neural Network and Fuzzy Logic, have the capabilities of learning and describing uncertainties. Although some new approaches have been proposed [56, 62], the advantages of artificial intelligence have not been fully utilized. This research presents two novel wavelet based approaches that used soft computing tools. Two proposed denoising algorithms exploit attractive features of several existing algorithms, provide adaptation to natural images with different textures, and ensure an appropriate trade-off between noise suppression and detail preservation.

1.2 Thesis Outline

This thesis contains six chapters.

Chapter 2 provides background information, a literature survey on noise reduction, and mechanisms used to evaluate denoising algorithms. Chapter 3 and 4 introduce the proposed algorithms for removing additive noise. Chapter 5 is the concluding chapter. Details of the chapters are briefly described here.

- **Chapter 2: Foundation of Denoising**

  This chapter investigates additive noise models and includes an in-depth literature survey of denoising related research work. Desirable features and complexities of denoising algorithms are discussed. Common mechanisms used to evaluate the performance of denoising algorithms are introduced. A

new image quality assessment method, from the human point of view, is also introduced.

- **Chapter 3: Type II Fuzzy Set Based Wavelet Shrinkage**

  This chapter gives a brief introduction to the soft computing tools used in this research, such as fuzzy logic theory. Concepts of fuzzy logic theory are explained. A popular form of a fuzzy logic theory, a type-2 fuzzy set, is briefly introduced. Finally, this chapter proposes a new denoising algorithm that combines the attractive features of several existing wavelet based algorithms and adapts the threshold using an interval type-2 fuzzy set. The design procedure for an interval type-2 fuzzy set based algorithm is outlined.

- **Chapter 4: Genetic Algorithm and Fuzzy Neural Network Based Algorithm**

  This chapter briefly introduces artificial neural networks and genetic algorithms. The basic elements of neural networks are explained. The back propagation (BP) network and continuous genetic algorithm are briefly described. A new method of removing noise based on fusing the results of different filters using a Fuzzy Neural Network (FNN) is proposed. A Genetic Algorithm (GA) is applied to choose the optimal parameters of the FNN. The GA aims to optimize the parameters of fuzzy models for FNN.

- **Chapter 5: Performance of Proposed Algorithms**

  Comparing with traditional filters and state-of-the-art denoising algorithms, this chapter mainly evaluates the performance of the proposed denoising algorithms from different aspects, such as objective assessment of image quality, computational cost, and universality of the proposed algorithms. Numerical analysis and graphs of results are shown in this chapter.

- **Chapter 6: Conclusion and Future Work**

  This chapter concludes that the proposed algorithms perform better than traditional filters and state-of-the-art denoising algorithms in terms of the chosen evaluation tools, such as mean squared error(MSE), peak signal to noise ratio (PSNR), image quality index (IQI), and Information Theoretic Criterion (ITC). Several major areas of future research are also listed in this chapter.

Chapter 2

Foundation of Denoising

As discussed in Chapter 1, it is difficult to get rid of noise while preserving details of images at the same time. To ensure an appropriate trade-off between noise suppression and detail preservation, proper image and noise models should be built and suitable filters should be designed. In this chapter, common additive noise models, various image denoising methods, the state of the art in wavelet based algorithms and performance evaluation methods are introduced.

2.1 Image and Noise Models

Many of the current denoising techniques are based on assumptions of noise models. In reality, assumptions may not always hold true due to the varied nature and sources of noise. An ideal denoising procedure requires a priori knowledge of the noise; while a practical procedure may not have the required information about the variance of the noise or the noise models. Thus, in practice, most of the algorithms estimate the variance of noise and assume the noise models to compare the performance with different algorithms.

While there have been many assumed additive noise models, only those relevant to this thesis, such as impulse noise models and Gaussian noise models, are discussed below.

2.1.1 Impulse Noise Models

Impulsive noise can be caused by coding or decoding errors, transient noise, errors in analog-to-digital conversion, etc. The properties of spikes, including their statistical characteristics, also vary depending on the situation at hand. The amplitudes of spikes can be different at different image regions. Sometimes, impulsive noise can corrupt separate lines, columns or fragments of images. However, a priori knowledge of spike properties is limited. In order to design image denoising algorithms under

the presence of this type of noise, some general models of the spikes should be assumed.

Impulse noise is a type of additive noise. The properties of impulsive noise are usually determined by the amplitude of spike A and the probability of spike occurrence p_r. In many current impulse noise models for images, corrupted pixels are often replaced with values equal to or near the maximum or minimum intensity values of the allowable dynamic range. For example, salt-and-pepper impulsive noise typically corresponds to fixed values near 0(minimum) or 255 (maximum,for an 8-bit image). In this thesis, a more general noise model in which a noisy pixel is taken as an arbitrary value in the dynamic range according to some underlying probability distribution is introduced. Let $O(i,j)$ and $A(i,j)$ denote the intensity value of the original and the noisy image at position (i,j), respectively. Then, for an impulse noise model with error probability p_r is described as:

$$A(x) = \begin{cases} O(i,j) & 1-p_r \\ \eta(i,j) & p_r \end{cases}$$

where $\eta(i,j)$ is an identically distributed, independent random process with an arbitrary underlying probability density function [7].

Recently, it has been shown that a type of α-stable distribution can approximate impulse noise more accurately than other models [37]. The parameter α controls the degree of impulsiveness which increases as α decreases. The Gaussian ($\alpha = 2$) and the Cauchy ($\alpha = 1$) distributions are the only symmetric α-stable distributions that have closed-form probability density functions.

A symmetric α-stable ($S\alpha S$) random variable is described by its characteristic function:

$$\phi(t) = exp(j\theta t - \gamma \mid t \mid^{\alpha})$$

where j is the imaginary unit, θ is the location parameter (centrality), γ is the dispersion of the distribution and $\alpha \in [0,2]$, which controls the heaviness of the tails, and it is the characteristic exponent. More information about the α-stable noise model can be found in [37].

2.1.2 Gaussian Noise Model

A type of noise which occurs in all recorded images to a certain extent is detector noise. This type of noise is due to the discrete nature of radiation, i.e. the fact that

each imaging system is recording an image by counting photons. Allowing some assumptions (which are valid for many applications), this noise can be modeled with an independent, additive model, where the noise $n(i, j)$ has a μ mean Gaussian distribution described by its standard deviation (σ), or variance. This means that each pixel in the noisy image is the sum of the true pixel value and a random, Gaussian distributed noise value. The 1-D Gaussian distribution has the density function,

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

where σ is the standard deviation of the distribution, and μ is the expectation of the distribution.

2.1.3 Mixed Noise Model

Mixed noise can cause significant difficulties for filtering, image interpretation, and restoration. Different types of noise require the use of different filtering approaches. Therefore, the algorithms of mixed noise removal are often based on hybrid structures using fuzzy and adaptive methods of processing [61, 67, 78].

Let us consider the following typical model with additive noise and spikes,

$$g(i, j) = \begin{cases} f(i, j) + n(i, j) & \text{with probability } 1 - p_r \\ A_{imp}(i, j) & \text{with probability } p_r \end{cases}$$

where $n(i, j)$ denotes the additive noise value for the $ij - th$ pixel, and $A_{imp}(i, j)$ is the amplitude of a spike, which may occur in the $ij - th$ pixel with probability P_r.

2.2 Classification of Denoising Methods

There are two basic approaches to image denoising, spatial domain filtering methods and transform domain filtering methods [50].

2.2.1 Spatial Domain Denoising Methods

A traditional way to remove noise from image data is to employ spatial filters. Two typical methods in the spatial domain are:

- **Linear Filtering**

  Most classical linear image processing techniques are based on the assumption that images are stationary. Even though linear filters are useful in a wide variety of applications, there are some situations in which they are not adequate. For instance, linear filters may involve image processing applications where both edge enhancement and noise reduction are desired. Usually edge enhancement can be considered as a high-pass filtering operation, while noise reduction is most often achieved using low-pass filtering operations. However, linear shift-invariant (LSI) filters do not take into account any structure in images. Therefore, the degree of smoothing is the same over all parts of an image and will cause the loss of some detailed information of the image. However, it is known that people are less sensitive to noise in more detailed regions of an image [29]. If the filter can smooth less in these regions than in the less detailed regions, it will preserve the detailed information of the images while smoothing out noise. Conventionally, linear shift invariant filters do not adapt to image content.

- **Non-linear Filtering**

  Nonlinear filters modify the value of each pixel in an image based on the value returned by a nonlinear function that depends on the neighboring pixels. Nonlinear filters are mostly used for noise removal and edge detection. For example, the traditional nonlinear filter is the median filter. It can efficiently decrease additive noise, especially impulsive noise. There are also many improved median filters, such as the weighted median filter [75], center weighted median filters [12], detail preserving median based filters [1], the multilevel hybrid median filter [65], etc. However, they do not get rid of all additive noise and blur edges to some degree.

2.2.2 Transform Domain Filtering

According to the choice of the "analysis function" [66], the transform domain filtering methods can be classified into the following two categories.

- **Spatial-Frequency Filtering**

  Spatial-Frequency Filtering refers to low pass filters using Fast Fourier Transform (FFT). In frequency smoothing methods [29] the removal of the noise is achieved by designing a frequency domain filter and adapting a cut-off frequency to distinguish the noise components from the useful signal in the frequency domain. These methods are time consuming and depend on the cut-off frequency and the filter function behavior. Furthermore, they may produce frequency artifacts in the processed image.

- **Wavelet domain**

  As mentioned above, noise is usually concentrated in high frequency components of the signal, which correspond to small detail size when performing a wavelet analysis. Therefore, removing some high frequency (small detail components), which may be distorted by noise, is a denoising process in the wavelet domain. Many investigations have been made into additive noise suppression in images using wavelet transforms. Filtering operations in the wavelet domain can be categorized into wavelet thresholding, statistical wavelet coefficient model and undecimated wavelet domain transform based methods.

2.3 State of the Art Methods of Denoising

Since the pioneering work [19] of D. L. Donoho and his collaborators for certain noise models by thresholding, which removes the noise without disturbing important signal features, much of the early work has been done on wavelet noise removal based on thresholding the Discrete Wavelet Transform (DWT) coefficients of an image and then reconstructing it [18, 20, 33]. Researchers have designed different methods to compute the parameters for the thresholding of wavelet coefficients. In addition, Undecimated Wavelet Transform, Multiwavelets, Bayesian and Markov Models based on wavelet transform are also employed for noise deduction. Based on these techniques, wavelet domain denoising methods are classified into the following four categories.

2.3.1 Wavelet Thresholding

The simplicity of wavelet thresholding makes it a popular approach for denoising. In its most basic form, this technique operates in the orthogonal wavelet domain, each coefficient compares to a threshold. If the coefficient is smaller than the threshold it is set to zero, otherwise, it is kept or modified. A systematic theory was developed mainly by Donoho and Johnstone [18, 19, 20, 33]. They have shown that various wavelet thresholding schemes for denoising have near optimal properties in the minimax sense and perform well in image denoising. An extensive review of wavelet thresholding in image processing can be found in [31].

- **Non Adaptive Thresholding**

  Most methods for estimating the threshold assume additive white Gaussian noise (AWGN) and an orthogonal wavelet transform. Among those, best known is the universal threshold of Donoho and Johnstone [19].

$$T_{univer} = \sigma_n \sqrt{2 \log(n)} \tag{2.1}$$

where σ_n is the estimate of the standard deviation of additive white noise and n is the total number of the wavelet coefficients in a given detail image. At different resolution scales, the threshold differs only in the constant factor that is related to the number of the coefficients in a given subband.

The threshold shown in Equation 2.1 is non-adaptive universal threshold, which depends only on the number of data points. It has a tendency of reaching the best performance in terms of MSE when the number of pixels reaches infinity. It is known to yield overly smoothed images because its threshold choice can be large due to its dependence on the number of pixels in the image [19].

- **Adaptive Thresholding**

  To achieve a better denoising performance by thresholding, many strategies have been introduced to make the threshold adaptive.

  Based on the wavelet shrinkage denoising theory proposed by D.L. Donoho, a new thresholding function, SURE (Stein's Unbiased Risk Estimate) shrinkage, is presented in [21]. Unlike traditional hard thresholding, SURE shrinkage has an infinite-order continuous derivatives. By using the new thresholding function, a new adaptive shrinkage method is presented based on SURE. It is very effective in adaptively finding the optimal solution in the least mean square error (LMSE) sense. In addition, SURE is also exploited to build a new estimator for arbitrary multichannel images embedded in Gaussian noise [8]. The SURE based algorithms perform better than VISUShrink (or universal thresholding) also proposed by by D.L. Donoho [19].

  In addition, BayesShrink [16, 58, 59] minimizes the Bayesian risk estimator function assuming a generalized Gaussian prior and thus yielding a data adaptive threshold. Generally speaking, BayesShrink outperforms SUREShrink [19].

It can be concluded that a good threshold should properly be chosen so that most coefficients below the threshold are noise and values above the threshold are signals of interest.

2.3.2 Denoising by Wavelet Coefficient Statistical Model

This category of approaches focuses on some more interesting and appealing properties of the Wavelet Transform such as multiscale dependency between the wavelet coefficients, local correlation between neighborhood coefficients. It aims at building proper models for image data by use of the wavelet transform. A fundamental review of statistical properties of wavelet coefficients can be found in [2, 3, 4, 9]. The following two popular techniques exploit the statistical properties of the wavelet coefficients based on a probabilistic model.

- **Marginal Probabilistic Model**

  A number of researchers have developed homogeneous local probability models for images in the wavelet domain. Specifically, the marginal distributions of wavelet coefficients are highly kurtotic, and usually have a marked peak at zero and heavy tails [10]. The Gaussian mixture model (GMM) [52, 71] and the generalized Gaussian distribution (GGD) [51] are commonly used to model the wavelet coefficients distribution. Although GGD is more accurate, GMM is simpler to use. In [55], a statistical model is built for images decomposed in an over complete wavelet pyramid. Each coefficient of the pyramid is modeled as the product of two independent random variables: an element of a Gaussian random field, and a hidden multiplier with a marginal log-normal prior. The latter modulates the local variance of the coefficients. In [57], a bivariate maximum a posteriori (MAP) estimator which relies on a mixture of bivariate Laplacian models is introduced. This model not only is bivariate but also is a mixture and therefore, it is possible to capture the heavy-tailed nature of the data as well as the interscale dependencies of wavelet coefficients. Cheng et al [13] developed an example-based image denoising algorithm. In this algorithm, image denoising is formulated as a regression problem, which is then solved using support vector regression (SVR). Using noisy images as training sets, SVR models are developed. The models can then be used to denoise different images corrupted by random noise at different levels.

- **Joint Probabilistic Model**

  Usually, Hidden Markov Models (HMM) [28, 40, 63, 73] are efficient in capturing inter-scale dependencies, whereas Markov Random Field(MRF) [15, 36, 53, 54], models are more efficient to capture intrascale correlations.

  However, most of the improvements of wavelet domain HMMs only focus on how to impose an additional dependency structure on the original wavelet domain HMMs to capture the additional dependencies among wavelet coefficients. Besides this, existing methods do not fully consider the effects of noise in high frequency subbands of wavelet transforms. Some simple algorithms of wavelet domain HMMs, such as dividing the subband of wavelet coefficients into blocks, can not be carried out smoothly in a noisy image. Recently, many approaches combined with other statistic models have been proposed. For example, in [40], in order to reduce the effects of image noise and provide powerful tractable probabilistic image models, trained templates are built, which are constructed in the subband of scaling coefficients, then these trained templates are applied to wavelet domain HMM. In [73], Yang et al combined local Gaussian mixture model (LGMM) and HMMs in frame-domain and proposed a Local Contextual hidden Markov model based on frame-domain and a corresponding training algorithm. This model exploits the local statistics of coefficients so that it can effectively capture correlation of wavelet frame coefficients. Moreover, in [28], Ichir et al designed a Bayesian estimation framework using three different models: the independent

Gaussian mixture model, the hidden Markov tree model, and the contextual hidden Markov field model. Each is expressed by the posterior laws. Appropriate Markov chain Monte Carlo algorithms perform unsupervised joint blind separation of the sources and denoising procedures.

Pizurica [53] models local spatial interactions of the wavelet coefficients with a MRF. An iterative updating technique known as iterated conditional modes (ICM) is applied to estimate the binary masks containing the positions of those wavelet coefficients that represent the useful signal in each subband. In [54], a novel anisotropic MRF prior model is proposed. It combines three criteria for distinguishing supposedly useful coefficients from noise: coefficient magnitudes, their evolution across scales and spatial clustering of large coefficients near image edges. In [83], an adaptive multiwavelet thresholding algorithm (AMT) is proposed. This algorithm can automatically determine the wavelet shrinkage thresholding in the multiwavelet domain without a priori knowledge of an image, for instance, the variance of image noise.

In addition, Bui et al applied a hidden Markov tree model to account for the inter-scale, inter-direction, and inter-location dependency [26]. In [23], the interscale dependencies between the coefficients are captured using a hidden Markov tree model. The combined spatial and interscale model gives improvements over hidden Markov models for white noise. However, all approaches mentioned above need to estimate the required parameters which sometimes are hard to obtain. In [35], a hierarchical, nonparametric statistical model is introduced for wavelet representations of natural images. Extending previous work on Gaussian scale mixtures, wavelet coefficients are marginally distributed according to infinite, Dirichlet process mixtures. A hidden Markov tree is then used to couple the mixture assignments at neighboring nodes. Via a Monte Carlo learning algorithm, the resulting hierarchical Dirichlet process hidden Markov tree (HDP-HMT) model automatically adapts to the complexity of different images and wavelet bases.

2.3.3 Denoising by Undecimated Wavelet Transform

The discrete transform is very efficient. Its only drawback is that it is not translation invariant. Translations of the original signal lead to different wavelet coefficients, this may cause visual artifacts such as pseudo-Gibbs phenomenon [50]. To overcome these artifacts and get complete information of the analyzed signal the undecimated discrete wavelet transform (UDWT) is introduced. Since it does not decimate the signal, it produces more precise information for frequency localization. From the computational point of view the undecimated wavelet transform has larger storage space requirements and involves more computations. With the increase of the speed and storage of computers, this issue does not affect computational cost too much.

Mignotte proposes an image denoising algorithm using shrinkage of undecimated wavelet coefficients [49]. It is reported that the algorithm yields improvements in

terms of image quality and lower mean square error, especially when the image is corrupted by strong additive white Gaussian noise. In [85], a novel wavelet filtering method by spatial correlation thresholding based on UDWT is presented. This algorithm multiples two adjacent wavelet scales with some relative translations to line up the edges, and presents a new threshold related to spatial correlation that preserves edges well while greatly reducing noise.

2.3.4 Denoising by Combined Techniques

Generally speaking, wavelet coefficient statistical model based algorithms outperform thresholding based algorithms. However, with the complexity of statistical models, the computational cost rises rapidly. In order to get better results without increasing the computational cost too much, many approaches have been introduced by combining the existing algorithms. For instance, in [55] and [82], a parametric solution for FIR Wiener filtering in the wavelet domain is introduced. This solution takes into account the colored nature of signal and noise in a UDWT, and is adaptively trained via a simple context model. The combined Wiener filter offers impressive denoising performance at modest computational complexity. In [17], the Wiener filter and wavelet shrinkage are combined. This approach builds a new thresholding function based on the wavelet shrinkage to overcome the shortcomings of the discontinuous function in hard thresholding and the permanent bias in soft thresholding. The image noise is filtered out while the edges are well preserved. In [79], a particle filter with wavelet shrinkage are combined to achieve robust performance against inhomogeneous noise mixtures. The particle filter acts to suppress outlier-rich components of the noise while the wavelet domain shrinkage attenuates any remaining, less heavily tailed noise components. The combined algorithm demonstrates excellent rejection of salt-and-pepper noise mixed with additive white Gaussian noise (AWGN). In [69], the Lee filter is extended to the wavelet transform domain. This algorithm is effective for multiscale image denoising.

Furthermore, some algorithms combine soft computing tools to gain better results. A simple but efficient new fuzzy wavelet shrinkage method [62] can be seen as a fuzzy variant of a recently published probabilistic shrinkage method [52] for reducing adaptive Gaussian noise from digital greyscale images. The proposed method can efficiently and rapidly reduce additive Gaussian noise from digital greyscale images, outperforms current fuzzy non-wavelet methods, and is comparable with some recent but more complex wavelet methods [2, 5, 24, 52]. In [56], a novel algorithm based on soft thresholding of wavelet coefficients using an interval type II fuzzy system is developed for reducing speckle noise in Optical Coherence Tomography images. All combined algorithms attain better results than any one of the former algorithms.

In addition, the idea of image fusion is also exploited to improve the denoising results. For example, Ma et al proposed an algorithm based on fusing curvelets and wavelets in 2007 [41]. The Curvelets denoising approach has been widely used

in many fields for its ability to obtain high quality images. The main drawback of curvelets denoising algorithms is that they cause artifacts in denoised images. Ma's algorithm performs pixel fusion of the denoised images by curvelets and wavelets approaches. Firstly the denoised results of curvelets and hidden Markov tree based wavelets are calculated. Then image regions are analyzed with quadtree decomposition. Finally the result is obtained by fusing the weighted pixels of the denoised images by curvelets and wavelets based approaches. These authors then introduced a fuzzy theory to this fusion process, so the local properties of the fuzzy windows are estimated. Image fusion is applied to the curvelets and wavelets denoised images where the weights are decided by local properties.

There are also many image fusion based denoising algorithms in the wavelet domain. For example, an algorithm based on the multiwavelet transform and data fusion is proposed in [86], and another algorithm based on the pixel domain and wavelet domain is presented for fusing and denoising of noisy multifocus images [84].

2.4 Performance Evaluation in Image Denoising

How to fully evaluate the performance of denoising algorithms is also a challenging task. Objective image quality measures play important roles in various image processing applications. Basically, there are two types of objective quality or distortion assessment approaches. The first is mathematically defined measures, such as mean squared error (MSE) and peak signal to noise ratio (PSNR). The second considers human visual system (HVS) characteristics in an attempt to incorporate perceptual quality measures.

In practice, however, the HVS is more tolerant to a certain amount of noise than to a reduced sharpness. Moreover, the visual quality is highly subjective [6], and difficult to express objectively. In addition, the HVS is also highly intolerant to various artifacts, like "blips" and "bumps" in the reconstructed image [18]. The importance of avoiding those artifacts is also meaningful. For instance, in certain applications (like astronomy, or medicine) such artifacts may cause wrong data interpretations. Unfortunately, none of these complicated objective metrics in the literature have shown any clear advantage over simple mathematical measures such as MSE and PSNR under strict test conditions and different image distortion environments.

To evaluate the proposed algorithms in this thesis, four objective assessment tools, MSE, PSNR, Image Quality Index(IQI) and Information Theoretic Criterion (ITC), will be applied to obtain fair and complete performance evaluation.

These four image quality assessment methods are briefly introduced as follows:

- **Mean Squared Error (MSE)**

  The ultimate objective of image denoising is to produce an estimation $\widehat{f}_i$ of the noise-free image f_i , which approximates it best, under given evaluation

criteria. As in any estimation problem, an important objective goal is to minimize the error of the result as compared to the uncorrupted data. In this respect, a common criterion is minimizing the mean squared error (MSE),

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - \widehat{f}_i)^2 \tag{2.2}$$

- **Peak Signal to Noise Ratio (PSNR)**

  In image processing, another common performance measurement is the peak signal to noise ratio (PSNR), which is for grey scale images defined in dB as

$$PSNR = 20log_{10}\frac{255}{\sqrt{MSE}} \tag{2.3}$$

- **Image Quality Index (IQI)**

  Wang proposes an index [70] which is designed by modeling any image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. Although the new index is mathematically defined and does not explicitly employ the human visual system model, experiments on various image distortion types show that it exhibits surprising consistency with subjective quality measurement.

  The definition of the new quality of index [70]:

  Let $X = x_i|i = 1, 2, ..., N$ and $Y = y_i|i = 1, 2, ..., N$ be the original and tested images respectively, so the quality index is defined as:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)(\bar{x}^2 + \bar{y}^2)} \tag{2.4}$$

  where,

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

It also can be rewritten as,

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \times \frac{2\overline{x}\overline{y}}{\overline{x}^2 + \overline{y}^2} \times \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{2.5}$$

This quality index models any distortion as a combination of three different factors: loss of correlation, luminance distortion, and contrast distortion. This can be observed in Equation 2.5: the first component is the correlation coefficient between X and Y, which measures the degree of linear correlation between X and Y. The range of Q is $[-1, 1]$. The best value is 1 which means that the tested image is exactly equal to the original image.

- **Information Theoretic Criterion (ITC)**

  As discussed above, objective performance evaluation methods treat an image simply as a matrix of numbers. They do not reflect exactly the human perception of images. Since the visual quality of images is also important for evaluating the performance of denoising algorithms, an assessment tool based on the HVS is needed to make a full evaluation of the performance of denoising algorithms.

  Zhang and Jernigan propose an information theoretic criterion (ITC) based on natural scene statistics [80]. By using Gaussian scale mixture model in an information theoretic framework, an algorithm is designed to compute the minimum perceptual information contained in the images from the HVS point of view, and evaluates the image quality in the form of entropy.

  The ITC is defined as,

$$ITC = \frac{\min|_D\{I(X;Y|_z)\}}{H(X|_z)}$$

  where $\min|_D\{I(X;Y|_z)\}$ is the information fidelity criterion that quantifies the statistical information shared between the source and the distorted images. $H(X|_z)$ is defined as the conditional entropy of the perceptual information contained in the image.

  The ITC ranges from 0 to 1. When X, Y has no relationship, $ITC = 0$; when $Y = X$, $ITC = 1$, which means that the test image is exactly equal to the original image.

For the details, please refer to [80].

2.5 Summary

This chapter investigates additive noise models and includes an in-depth literature survey of denoising based on wavelets. Desirable features and complexities of denoising algorithms are discussed.

In addition, it explains common mechanisms used to evaluate the performance of denoising algorithms. A method that incorporate a basic model of the HVS, for performance evaluation of denoising algorithms, is introduced.

According to the current literature, denoising algorithms based on wavelet transform are the best choice for achieving the desired denoising performance. However, the computational complexity must also be considered. Thresholding techniques used with the Discrete Wavelet Transform are the simplest to implement. UDWT and Multiwavelets improve the performance at the expense of computation. HMM based methods seem to be promising but also are complex. Finally, universality of algorithms also needs to be considered. A universal denoising algorithm is a dream of researchers, Although there is no universal method. In this study, the denoised results of the proposed algorithms and existing algorithms are compared under different noise models and variances by means of the evaluation methods introduced above.

It can be expected that future research will focus on building robust statistical models of non-orthogonal wavelet coefficients based on their intra scale and inter scale correlations by employing soft computing tools. Such models combined with soft computing tools can be effectively used for image denoising. The following chapter introduces an algorithm based on the combination of wavelet and a soft computing tool, interval type-2 fuzzy sets.

Chapter 3

Interval Type-2 Fuzzy Set Based Wavelet Shrinkage

This chapter starts with an overview of the basic concepts of wavelets and fuzzy sets. These can be found in many books and papers at different levels of exposition. Two standard books include: C. S. Burrus's book "**Introduction to wavelets and wavelet transforms: a primer**" [10] and Jyh-Shing Jang's book "**Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence**" [30]. Some application books include: Karray's book "**Soft computing and intelligent systems design: theory, tools, and applications**" [34], Ruano's book "**Intelligent control systems using computational intelligence techniques**" [60], Stark's book "**Wavelets and Signal Processing: An Application-Based Introduction**" [66], J. M. Mende's book "**Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions**"[45], and Maarten's book "**Noise reduction by wavelet thresholding**" [31]. Introductory papers include: Zadeh's paper [77], Mendel's papers [46, 47, 48]. More technical ones are Stefan Schulte's paper [62], and Puvanathasan's paper [56]. After the overview, an improved denoising algorithm combining type II fuzzy sets and wavelet shrinkage is presented.

3.1 Introduction to wavelet representation

To understand why wavelets are useful for signal processing, consider some basic concepts:

3.1.1 Wavelet Concept and Its Transforms

Wavelets are functions that are wave-like but localized. These functions oscillate like a wave in a limited portion of time or space. Wavelets can analyze a signal or an image in detail according to scale. One chooses a particular wavelet, stretches it

to meet a given scale and shifts it to find the specific location, while investigating its correlations with the analyzed signal. This analysis is similar to observing the displayed signal from various distances. The signal correlations with wavelets are that large scales reveal coarse features, while small scales discover fine signal structures. It is often said that the wavelet analysis can see both the forest and the trees. In such a scanning through a signal, the scale and the position can change continuously (continuous wavelet transform) or in discrete steps (discrete wavelet transform).

From an engineering point of view, discrete wavelet analysis is a two channel digital filter bank (composed of lowpass and highpass filters), iterated on the lowpass output. The lowpass filtering yields an approximation of a signal (at a given scale), while the highpass (more precisely, bandpass) filtering yields the details that constitute the difference between the two successive approximations. A family of wavelets is then associated with the bandpass, and a family of scaling functions with the lowpass filters. The detailed concept explanations can be found in [10, 44]. In this thesis, only the general aspects of discrete wavelet transform are addressed. Furthermore, the representation of two-dimensional images are discussed.

- **Definition of discrete wavelet transform(DWT)** [10]

  The idea is to express a signal as a superposition of two sorts of elements: scaling functions and wavelets. The scaling functions represent the signal with a lower resolution approximation (lowpass filters) while the wavelets represent the resulting difference components (highpass filters). Both scaling functions and wavelets are weighted and shifted as necessary to exactly represent the signal. A typical representation is the so-called discrete wavelet transform (DWT):

$$f(t) = \sum_k c(k)\varphi_k(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \tag{3.1}$$

  or, more specifically revealing the nature of the shifted components,

$$f(t) = \sum_k c_{j_0}(k)2^{j_0/2}\varphi(2^{j_0/2}t-k) + \sum_k\sum_{j=j_0}^{\infty} d_j(k)2^{j/2}\psi(2^j t-k) \tag{3.2}$$

  Where, $\varphi(t)$ is the scaling function, $\psi(t)$ is the wavelet. The index j represents the scale (coarse to fine resolutions) with j_0 as the coarsest, or lowest resolution level of the representation. Both scaling functions and wavelets are compressed along the time axis by a factor of 2 as the index j is increased by 1. The index k represents the translations. $c_{j_0}(k)$ are the approximation coefficients, and $d_j(k)$ are the difference coefficients. The first sum provides a low resolution, or coarse scale representation, of the signal. The second sum provides the difference components which are across all scales or resolutions and necessary to construct the signal and not contained in the low resolution coarse approximation. The simplest case, the Haar Wavelet Sys-

tem, is examined in Stark's book [66]. Figure 3.1 plots the Haar scaling and wavelet functions and Figure 3.2 shows the two level wavelet decomposition and reconstruction of the cameraman image.

The scaling function and wavelet must be chosen such that the set of compressed, shifted components comprise a basis with which to represent the signals of interest.

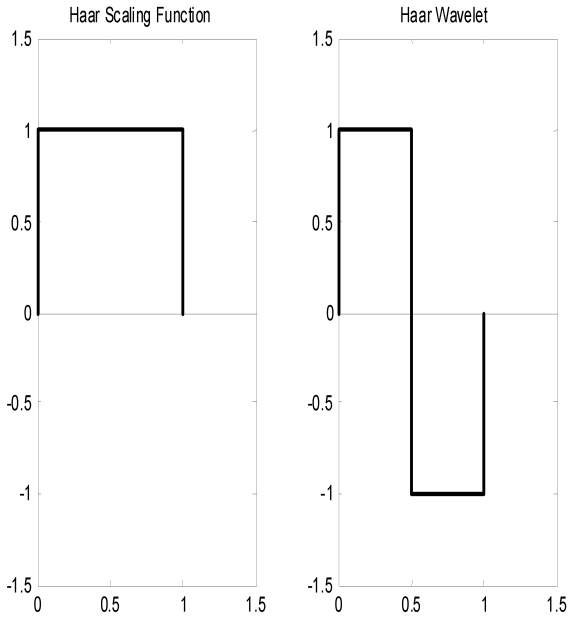


Figure 3.1: The Haar wavelet and scaling functions [66]

Figure 3.2: Haar wavelet decomposition. Left: Original image. Right: Result of the 2-step-DWT

In order to form a basis with which to represent signals, the scaling function

and wavelet must satisfy a pair of recursion equations [44]:

$$\varphi(t) = \sum_n s(n)\sqrt{2}\varphi(2t-n) \tag{3.3}$$

$$\psi(t) = \sum_n w(n)\sqrt{2}\varphi(2t-n) \tag{3.4}$$

Both simply require that the component at a lower resolution be representable by the scaling function compressed and shifted at the next higher resolution. The scaling function coefficients are $s(n)$and the wavelet coefficients are $w(n)$. If the wavelets are to be orthogonal over integer translations, and the wavelets are to span the space of the difference components, the coefficients must be related through

$$w(n) = (-1)^n s(1-n)$$

or, if $s(n)$ is of finite length N (even),

$$w(n) = (-1)^n s(N-1-n)$$

.

From the recursion equation 3.3 and 3.4, the expressions 3.5 and 3.6 for the approximation coefficients and the difference coefficients can be derived at one resolution level in terms of those at the next higher scale [10]:

$$c_j(k) = \sum_m s(m-2k)c_{j+1}(m) \tag{3.5}$$

$$d_j(k) = \sum_m w(m-2k)c_{j+1}(m) \tag{3.6}$$

Thus, wavelet coefficients can be calculated recursively from scale to scale, which yields the complexity of $O(N)$, where N is the length of the data to be transformed. Based on Equation 3.5 and 3.6, the DWT can be implemented by filter banks shown as in Figures 3.3 and 3.4. In addition, Figure 3.5 illustrates the arrangement of approximation and difference or detail coefficients for the 2-D discrete wavelet transform.

- **Definition of Undecimated Wavelet Transform(UDWT)**

  Decimation of the wavelet coefficients is an intrinsic property of the discrete wavelet transform (DWT). The decimation step removes every other of the coefficients of the present level. Thus the computation of the wavelet transform is faster and more compact in terms of storage space. More importantly, the transformed signal can be perfectly reconstructed from the remaining coefficients. Unfortunately, the decimation causes shift variance of the wavelet

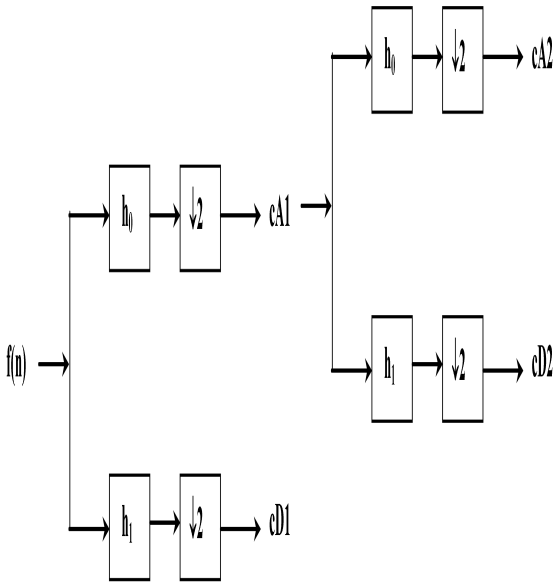


Figure 3.3: Block diagram of the 1-D discrete wavelet transform. cA1 and cD1 are the approximation and detail coefficients at the first, or highest resolution level [32].

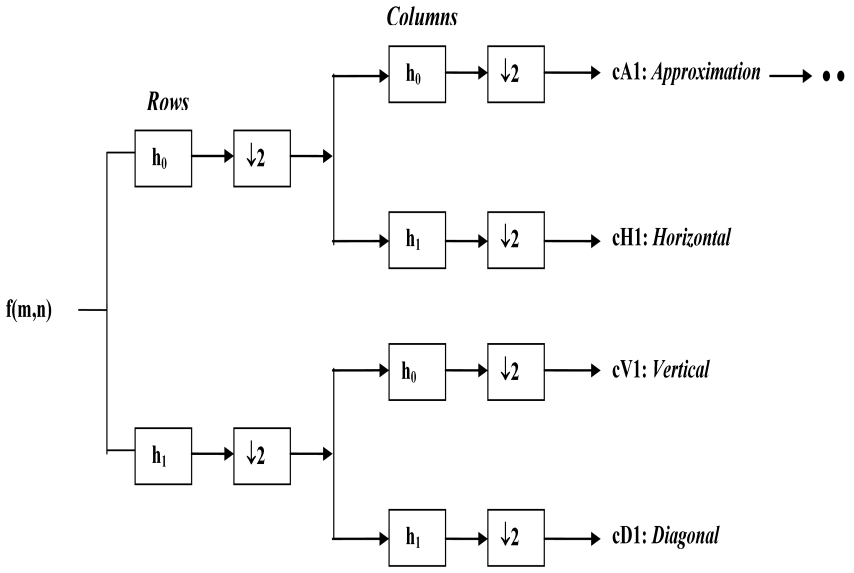


Figure 3.4: Block diagram of the 2-D discrete wavelet transform. cA1, cH1, cV1 and cD1 are the approximation, horizontal, vertical and detail coefficients at the first, or highest resolution level [32]

transform. A formal definition of shift variance was proposed by Simondelli [64]: *shift variance means small shifts in the input signal can cause major variations in the distribution of energy between coefficients at different scales.* Simply speaking, the shift variance means that the DWTs of a signal and its shifted version are not the same (see Figure 3.6).

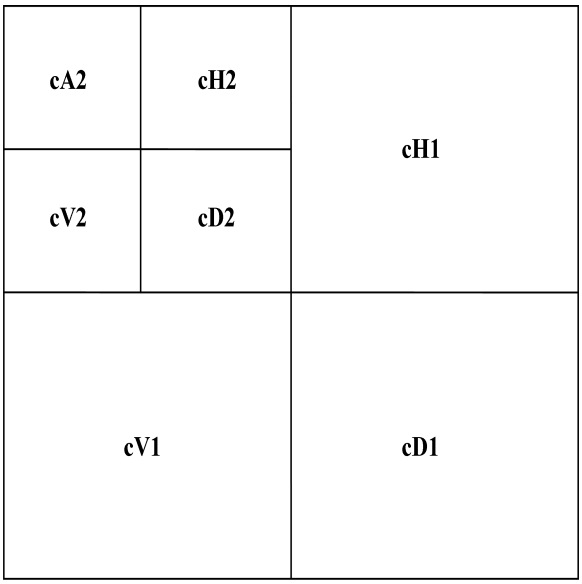


Figure 3.5: Arrangement of approximation (cA2) and difference or detail coefficients for the 2-D discrete wavelet transform. cH1, cV1, and cD1 are the highest resolution horizontal, vertical, and diagonal detail coefficients [32].

Shift invariance has been shown many times, such as [14, 54]. In order to achieve shift invariance, researches have created several other wavelet transform algorithms. One type of these transforms is known as the undecimated wavelet transform (UDWT) of the dyadic filter tree [42].

In an undecimated wavelet transform, a signal is represented with the same number of wavelet coefficients at each scale. These coefficients are samples of the continuous wavelet transform at all integer locations at each dyadic scale function: $\varphi_k(t)$ in Equation 3.1. Such an overcomplete representation results from decomposing a signal into a family of wavelets: $2^{j/2}\psi(2^j t - k)$ in Equation 3.2. Now the wavelets are not linearly independent, they do not constitute a basis but a frame.

An undecimated wavelet transform approaches translation invariance, and is also called the Stationary Wavelet Transform. It is computed with the *à* trous algorithm [43]

Figure 3.7 shows the difference between the decimated wavelet transform and the undecimated wavelet transform(1-D).

Figure 3.8 indicates a 'db2' undecimated wavelet decomposition. Comparing to Figure 3.2, it can be observed that the undecimated wavelet transform yields an increased amount of information about the transformed image compared to the DWT. The number of the wavelet coefficients does not shrink between the transform levels. The additional information is useful for better analysis and understanding of image properties. For instance, in image denoising applications the discrimination between the noise and real data can

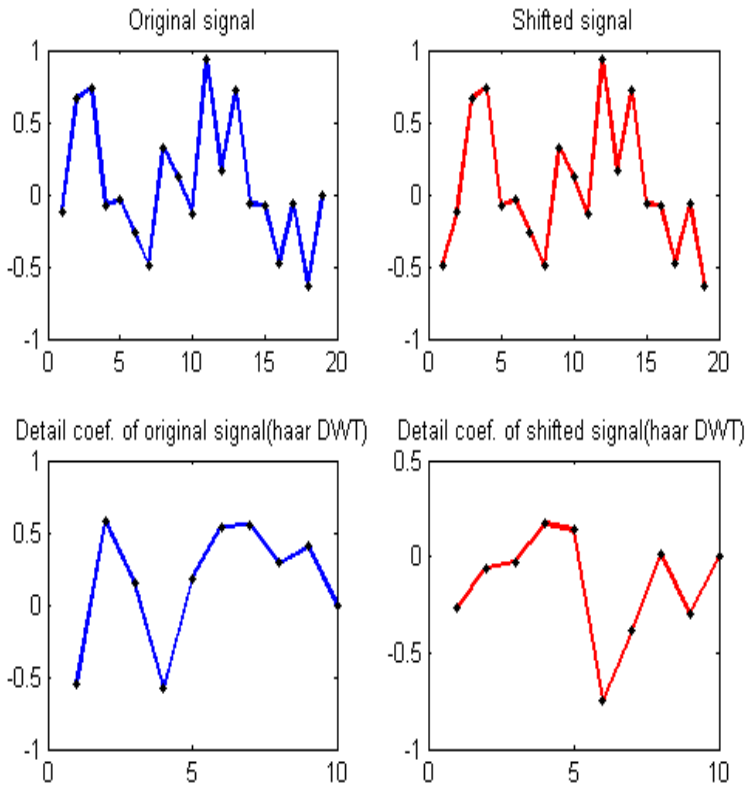
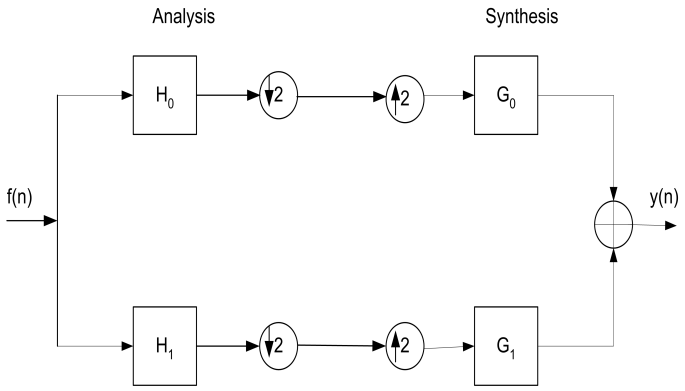


Figure 3.6: The discrete wavelet transform lacks shift-invariance. The upper row indicates a sample signal (left, blue) and the same signal right-shifted by one sample point (right, red). The lower row shows the DWTs of the signals from the upper row. There is a big difference between the DWT coefficients. Obviously, the two signals in the lower row are not shifted versions of each other.

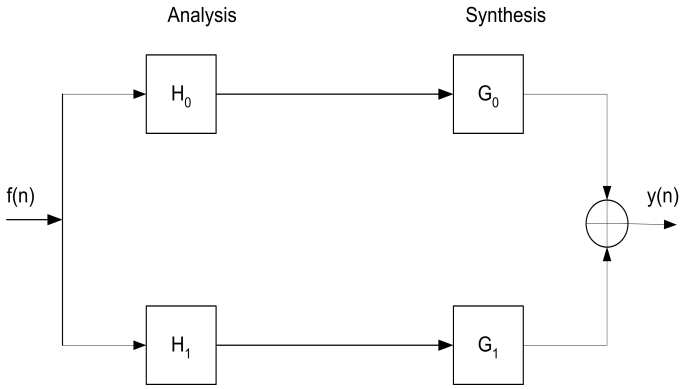
be improved. Much information is especially important when statistical approaches are used for analyzing the wavelet coefficients. The drawbacks of the UDWT are higher computational and memory cost as well as redundancy in the coefficients.

3.1.2 Wavelet Shrinkage Based Image Denoising

Wavelet shrinkage [18] is a denoising technique based on the idea of thresholding every single wavelet coefficient of the signal. Wavelet coefficients having small absolute value are presumed to encode mostly noise and very fine details of the image. Conversely, the important information is encoded by the coefficients having large absolute value. Removing the small absolute value coefficients and then reconstructing the image should produce an image with less noise. The wavelet shrinkage approach can be summarized as follows:



(a) Decimated wavelet transform



(b) Undecimated wavelet transform

Figure 3.7: Decimated and undecimated wavelet transforms [32]. Note in (a), there are down-sampling and up-sampling units, but absent in (b)

1. Apply the wavelet transform to an image.
2. Estimate a threshold value.
3. Remove the coefficients that are smaller than the threshold.
4. Reconstruct the image using the thresholded coefficients.

One challenge in wavelet shrinkage or thresholding approaches is to find an appropriate threshold value. In 1994, Donoho and Johnstone [19] proposed a universal thresholding, and its result is an effective, non-linear, edge preserving smoothing operation. Since then, more sophisticated methods, such as spatially adaptive thresholding methods, have been designed to distinguish coefficients associated with image texture from those associated with noise. An extensive review of wavelet thresholding in image denoising is in [31].

Another open question in wavelet shrinkage algorithms is how to apply the threshold. Two standard thresholding methods are: hard-thresholding, ("keep or kill"), and soft-thresholding ("shrink or kill") [31]. In both cases, the coefficients that are below a certain threshold are set to zero.

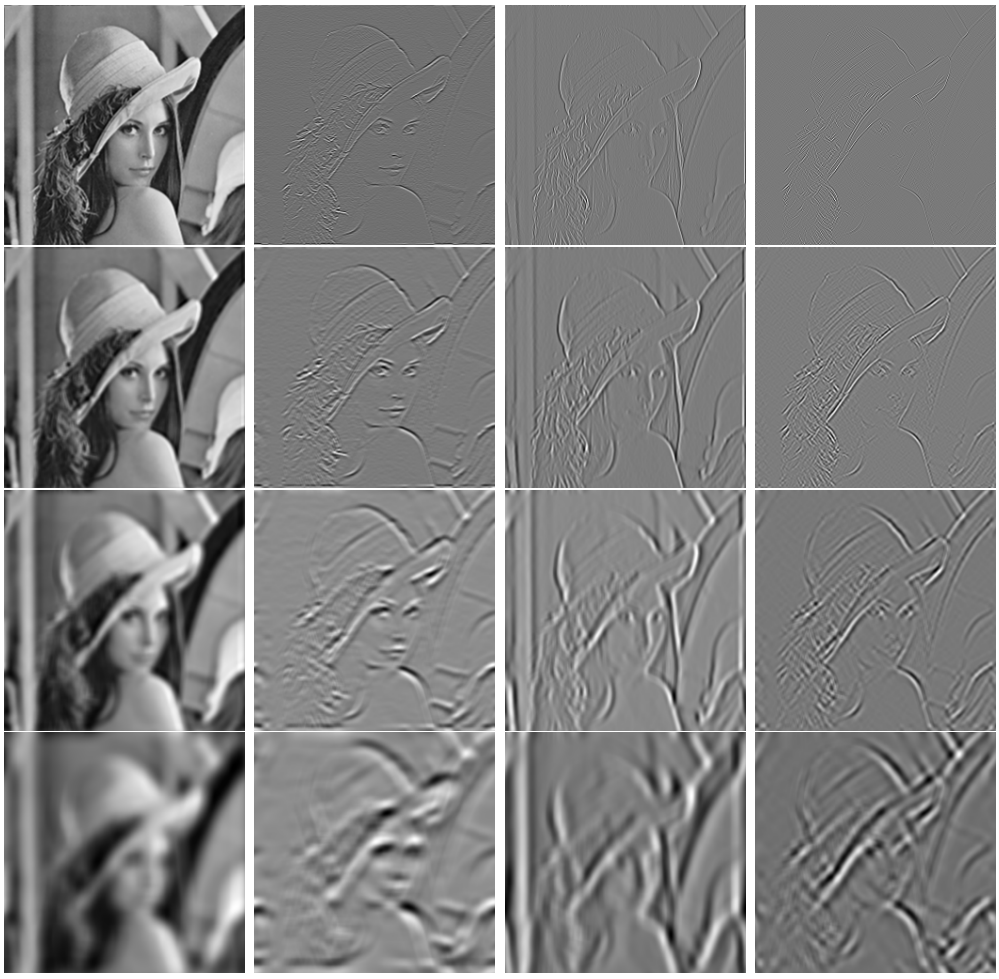


Figure 3.8: 'db2' undecimated wavelet decomposition. An example of the redundant wavelet frame decomposition in four resolution levels. From left to right are images in approximation, horizontal, vertical, and diagonal subbands, respectively.

- **Hard Thresholding**

  Hard thresholding methods zero the coefficients that are smaller than the threshold and leave the remaining ones unchanged.

$$T_{hare}(w) = \begin{cases} 0 & \text{if } |w| \leq T \\ \text{w} & \text{if } |w| > T \end{cases} \tag{3.7}$$

- **Soft Thresholding**

  In soft thresholding, the magnitudes of the coefficients above threshold are reduced by an amount equal to the value of the threshold

$$T_{soft}(w) = \begin{cases} 0 & if\,|w| \leq T \\ \text{sgn}(w)(|w| - T) & if\,|w| > T \end{cases} \tag{3.8}$$

- **How to Apply a Threshold**

  In both hard or soft thresholding methods, each wavelet coefficient is multiplied by a given shrinkage factor γ, which is a function of the magnitude of the coefficient, refer to Figure 3.9).

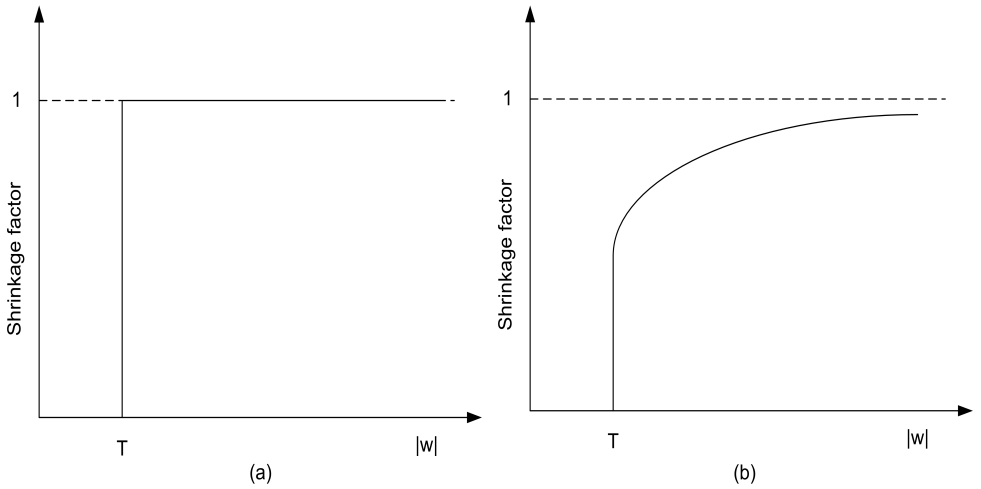


Figure 3.9: Shrinkage factors that multiply the wavelet coefficients in (a) hard-thresholding and (b) soft-thresholding.

As discussed in Chapter 2, wavelet thresholding is a popular approach for noise suppression due to its simplicity. A vast literature has emerged recently on image denoising via wavelet thresholding or shrinkage that was first introduced by Donoho and Johnstone [19]. The most well-known thresholds include universal [19] and Sure [20] proposed by Donoho and Johnstone, Bayes by Chang [11], and fuzzy shrinkage proposed by Schulte [62]. According to the literature, BayesShrink is derived in a Bayesian framework, and outperforms SureShrink most of the time. However, the generalized likelihood ratio of GGD is not always realistic. FuzzyShrinkage, instead of estimating the likelihood ratios for these measurements, imposes on them fuzzy membership functions. A fuzzy shrinkage factor will also express how likely a coefficient is“ a signal of interest, and the process is accomplished by using the appropriate fuzzy norms and co-norms as opposed to the Bayesian formalism and probabilities [62]. In this study, a new shrinkage method is proposed, which is based the soft thresholding and considers the inter and intra-scale persistence. An interval type-2 fuzzy set (IT2 FS) is employed in this method. The detailed algorithm is discussed in Section 3.3.

3.1.3 Noise Variance Estimation

According to the experience of denoising, obtaining the noise variance is a key for any sophisticated denoising algorithms. In some cases, the value of the input noise variance σ is known, or can be measured based on information other than the

distorted data. If images have no information about the noise variance, it must be estimated from the input data. Wavelet based methods commonly use the highest frequency subband of the decomposition for this purpose. In the DWT of an image, the cH_1 subband contains mainly noise. A robust estimation of noise variance $\hat{\sigma}$ is obtained by a median measurement, which is highly insensitive to isolated outliers of potentially high amplitudes. In [19], it is proposed,

$$\hat{\sigma_n} = \frac{median(|cH1|)}{0.6745} \tag{3.9}$$

The estimate in Equation 3.9 is commonly used in image denoising [16, 56, 62] and it is used in this thesis as well. Figure 3.10 shows that the noise estimated method used in this study is satisfactory by comparing the estimated noise variances with the true noise variances added to images, when the added noise is pure Gaussian noise.

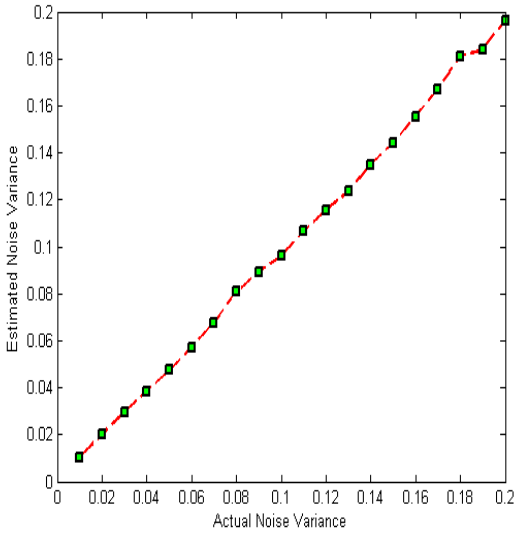


Figure 3.10: Noise Estimation. The result is almost a straight line with a 45-degree angle to both x- and y- axis. It means that the estimated noise variances are approximately equal to the actual noise variances.

3.2 Introduction to Fuzzy Set

Since fuzzy set theory was introduced by Zadeh about four decades ago [77], fuzzy sets have been applied to a variety of applications, ranging from control engineering, qualitative modeling, pattern recognition, signal/image processing, information processing, machine intelligence, decision making, management, finance, medicine, motor industry, robotics, and so on. This section presents the basic concepts of fuzzy sets.

3.2.1 Type-1 Fuzzy Set

In this subsection, brief descriptions about the fuzzy set theory which is applied in this thesis are given.

- **Definition of Type-1 Fuzzy Set**

  Generally, a fuzzy set type-1 (T1 FS) is defined as a collection of ordered pairs and can be expressed by the following notation [34],

  $$A = \{(x, \mu_A(x)) \,|x \in X\}$$

  Where $\mu_A(x)$ is called the membership function (MF) for fuzzy set A. The MF maps each element of X to a membership value between 0 and 1. The value indicates the degree of the elements belonging to the fuzzy set. Larger values denote higher degrees of memberships.

- **Fuzzy Membership Function**

  In practice, a type-1 fuzzy set can be completely characterized by its Membership Function (MF). The more convenient and concise way to define a T1 MF is to express it as a mathematical formula. For example, the symmetric Gaussian function is given by:

  $$F(x; \sigma, c) = \exp(-\frac{(x-c)^2}{2\sigma^2})$$

  The symmetric Gaussian function is illustrated in figure 3.11

  where x is the variable in the intensity domain (usually, it is the gray level between 0 and 255), and σ and c are the parameters which determine the shape of a Gaussian function. The range $[-\sigma, \sigma]$ is called the fuzzy region, where c is usually set as the mid-point of $[-\sigma, \sigma]$, but it is not necessary. If c is defined as the mid-point between $[-\sigma, \sigma]$, then it is called the symmetric Gaussian function. In this thesis, the symmetric Gaussian function is one of the membership functions used to transform an image from the intensity domain into the fuzzy domain. The selection of membership function depends on the application. The S-function, Z-function, and Bell-function are most commonly used, and these membership functions can be found in [34].

3.2.2 Fuzzy Set Type II

Although Zadeh introduced type-2(T2) FSs in 1975, very little literature was published about them until in the middle of nineties. T2 FSs are described by MFs that are characterized by more parameters than these for T1 FSs. Hence, T2 FSs provide more freedom for researchers to design a fuzzy logic system (FLS), and have the potential to outperform T1 FSs, especially in uncertain environments [48].

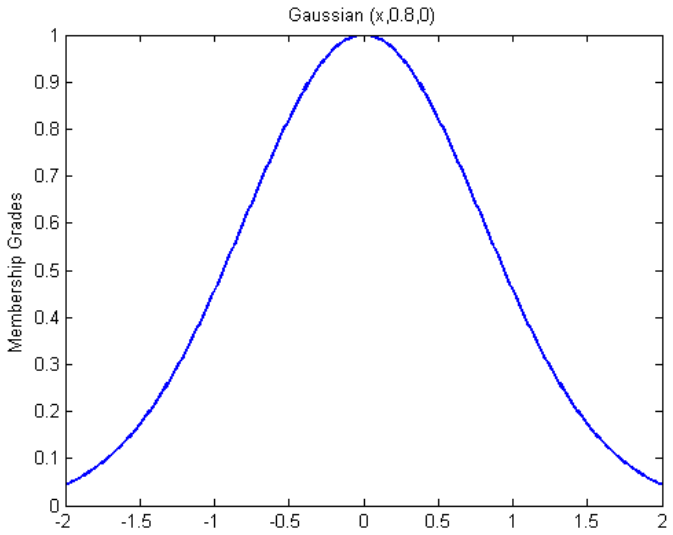


Figure 3.11: The symmetric Gaussian function

- **Definition of A Type-2 Fuzzy Set** [45]

  A T2 FS $\hat{A}$ is defined by a T2 membership function $\mu_{\hat{A}}(x,\mu)$, where $x \in X$ and $\mu \in J_x \subseteq [0,1]$,

$$\hat{A} = \{((x,\mu), \mu_{\hat{A}}(x,\mu)) | \forall x \in X, \forall \mu \in J_x \subseteq [0,1]\}$$

  In which $0 \leq \mu_{\hat{A}}(x,\mu) \leq 1$. $\hat{A}$ can also be expressed in the usual notation of fuzzy sets as

$$\hat{A} = \int_{x \in X} \int_{\mu \in J_x} \frac{\mu_{\hat{A}}(x,\mu)}{(x,\mu)}, \qquad J_x \subseteq [0,1]$$

  Where the double integral denotes the union over all x and μ. In order to define a T2 FS, one can define a T1 FS and assign upper and lower membership degrees to each element to construct the footprint of uncertainty. See the shading area in Figure 3.12. Therefore, a more practical definition for a T2 FS can be given as follows,

$$\hat{A} = \{(x, \mu_U(x), \mu_L(x)) | \forall x \in X, \mu_L(x) \leq \mu(x) \leq \mu_U(x), \mu \in [0,1]\}$$

  For details, please refer to [45, 68].

- **Fuzzy Membership Function**

  One convenient way to visualize a T2 FS membership function is to plot its footprint of uncertainty (FOU) on the 2-D domain of the T2 FS. The heights of a T2 MF (its secondary grades) sit atop its FOU. In Figure 3.12, if a set of Gaussian MFs is filled in (as implied by the shading area), then the FOU is obtained. Assume that the standard deviation of Gaussian primary

MF is known with perfect certainty, but its mean, m, is uncertain and varies anywhere in the interval from $m1$ to $m2$. Suppose the weighting (possibilities) of each FM is uniform, a uniform shading area over the entire FOU can be observed in Figure 3.12. Due to the uniform weighting, this T2 FS is called an Interval T2 FS (IT2 FS) [48].

In this study, T2 FSs are applied for wavelet coefficient thresholding. The reasons are these: (1)In most situation, T2 FSs outperforms T1 FSs, especially in uncertain environments. (2) T2 FSs have more freedom for designing a FLS [47].

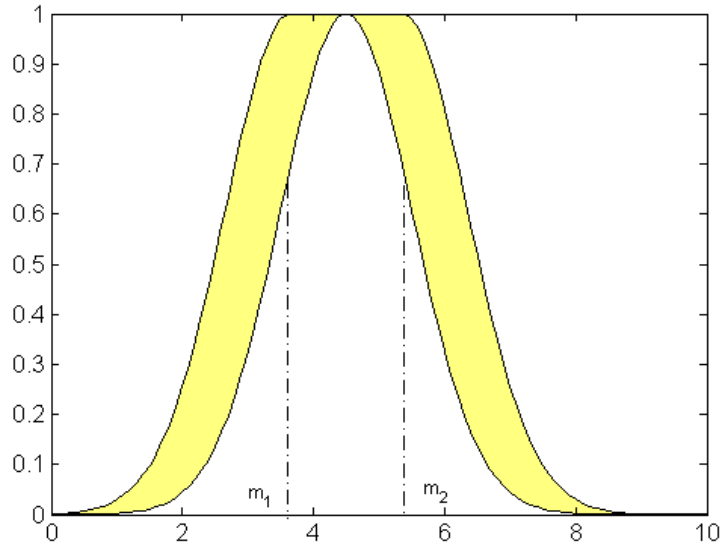


Figure 3.12: FOU of a Gaussian primary MF. Its mean varies in the interval $[m1, m2]$ and its standard deviation is a constant [48].

3.3 Type-2 Fuzzy Set Based Wavelet Shrinkage Algorithm

The IT2 FS theory, introduced above, provides a powerful technique to represent uncertainty and process human knowledge represented as fuzzy if-then rules. Like a T1 FLS, a T2 FLS based image denoising process also has four main stages: (1) image fuzzification, (2) modification of membership values, (3) fuzzy type reduction and (4)image defuzzification [45, 48]. The key of the process lies in the second and third steps. After the image data are transformed from the input plane to the fuzzy membership plane (fuzzification), appropriate fuzzy techniques, such as fuzzy rules, modify the membership values. Then proper fuzzy type reduction methods, such as the center-of-sets type reduction method, degrade the T2 FLS to a T1 FLS. The

main advantages of the proposed algorithm are: (1) the complexity of the method is lower than the HMM based ones [23], (2) removal of some artifacts by using UDWT, (3) addition of new fuzzy rules, correlation of interscale information to improve the noise reduction performance, (4) performance is better than Sure, Bayes, T1 fuzzy shrinkage, and HMM based methods. This point is shown in Chapter 5.

3.3.1 Definition of Membership Function and A Fuzzy Rule

A noisy image can be represented by the following equation in the spatial domain,

$$f(i,j) = y(i,j) + n(i,j) \tag{3.10}$$

where $f(i,j)$, $y(i,j)$, and $n(i,j)$ are the noisy image, original image, and noise respectively. Due to the linearity of the wavelet transform, additive noise in the image domain remains additive in the transform domain as well. Therefore, Equation 3.10 can be modeled as the following,

$$w_{s,d}(i,j) = y_{s,d}(i,j) + n_{s,d}(i,j) \tag{3.11}$$

where $w_{s,d}(i,j)$,$y_{s,d}(i,j)$, and $n_{s,d}(i,j)$ are wavelet coefficients of the noisy image, original image and noise respectively at the scale s and orientation $d \in \{V, H, D\}$.

Since wavelet coefficients are statistically dependent because of two properties of the wavelet transform of natural images: (1) large coefficients will reproduce across the scales (interscale dependencies), (2) if a coefficient is large or small, some of the neighboring coefficients are also likely to be large or small (intrascale dependencies) [62].

In this statement, there are some linguistic ambiguities, such as words "large or small". How "large or small" is "large or small"?

IT2 FSs are a better way to represent the ambiguity. In this study, firstly, three notations, $w_{s,d}(i,j)$, $x_{s,d}(i,j)$ and $c_{s,d}(i,j)$, are assigned to the wavelet coefficient, the neighborhood value, and the correlation map respectively at location (i,j). $w_{s,d}(i,j)$ is calculated by UDWT, $x_{s,d}(i,j)$ is calculated by the following equation introduced in [62]:

$$x_{s,d}(i,j) = \frac{(\sum_{m=-K}^{K}\sum_{n=-K}^{K}|w_{s,d}(i+m,j+n)|) - |w_{s,d}(i,j)|}{(2K+1)^2 - 1}$$

In this study, a hierarchical correlation map value $c_{s,d}(i,j)$, which considers both the correlation between neighboring coefficients (intra-scale) and correlation between two adjacent levels (inter-scale), can be calculated using equation 3.12, modified from [56]. In the generated correlation map, large values indicate position of edges in the original image, and zero valued coefficients correspond to smooth areas which are noise regions [39]. This leads to the following statement used to build a fuzzy

rule: at position (i, j), if the correlation map value is large, then the coefficient at (i, j) represents fine image structures of interest almost certainly and should not be set to zero; while a correlation map value close to zero indicates an area which needs to be smoothed due to noise [56].

$$c_{s,d}(i,j) = \sqrt{\text{inter}_{s,d}(i,j) \times \text{intra}_{s,d}(i,j)} \tag{3.12}$$

$$\text{inter}_{s,d}(i,j) = \sqrt{\frac{\sum_{m=-K}^{K}\sum_{n=-K}^{K} |w_{s,d}(i+m,j+n) \times w_{s+1,d}(i+m,j+n)|}{2K}}$$

$$\text{intra}_{s,d}(i,j) = \sqrt{\prod_{m=-K}^{K}\prod_{n=-K}^{K} |w_{s,d}(i+m,j+n)|}$$

Secondly, define appropriate IT2 FSs to represent these variables. IT2 FS A associates with the large magnitude wavelet coefficient $|w_{s,d}(i,j)|$, IT2 FS B associates with the large neighborhood values $|x_{s,d}(i,j)|$, and IT2 FS C associates with the large correlation map values $|c_{s,d}(i,j)|$. The sigmoid MFs , μ_A, μ_B and μ_C , are chosen to represent the three IT2 FSs respectively. In Figure 3.13, the shaded regions in the figures correspond to the areas where the uncertainty lies and determines if the coefficient is noisy or not. The general equation for a sigmoid MF is given in Equation 3.13 with two parameters, its center c and its width w.

$$\mu(x) = \frac{1}{1 + e^{\frac{-(x-c)}{w}}} \tag{3.13}$$

Equation 3.13 indicates the proposed algorithm depends on two parameters: center c and width w. Obviously, the IT2 FMs change at each scale s and each orientation d. It is known that, in image denoising, it is important that a filtering algorithm should adapt to the noise variances [62]. Therefore, in this study, the parameters c and w are chosen based on the experiments for a better performance in terms of PSNR.

According to Gupta's paper [25], the center c is calculated by:

$$c = \sqrt{\log(L_s)} \times \frac{\hat{\sigma_n}^2}{\hat{\sigma}_l}$$

And the local noise standard deviation $\hat{\sigma}_l$, of the subband is calculate by

$$\hat{\sigma}_l = \sqrt{\max((\frac{1}{n^2}\sum_{i,j=1}^{n} Y_{ij}^2 - \hat{\sigma}_n^2), 0)}$$

where $Y_{ij} \in CH_1$, $n \times n$ is the size of subband.

Based on the experiment, the width w is equal to the standard deviation of estimated noise.

The upper and lower MFs are determined as Equations 3.14, 3.15 and 3.16.

$$\mu_A^{upper} = \frac{1}{1+e^{\frac{-(x-c+\hat{\sigma}_n)}{w}}} \quad \text{and} \quad \mu_A^{lower} = \frac{1}{1+e^{\frac{-(x-c-\hat{\sigma}_n)}{w}}} \tag{3.14}$$

$$\mu_B^{upper} = \frac{1}{1+e^{\frac{-(x-c+\hat{\sigma}_n)}{1.5\times w}}} \quad \text{and} \quad \mu_B^{lower} = \frac{1}{1+e^{\frac{-(x-c-\hat{\sigma}_n)}{1.5\times w}}} \tag{3.15}$$

$$\mu_C^{upper} = \frac{1}{1+e^{\frac{-(x-c+\hat{\sigma}_n)}{0.5\times w}}} \quad \text{and} \quad \mu_C^{lower} = \frac{1}{1+e^{\frac{-(x-c-\hat{\sigma}_n)}{0.5\times w}}} \tag{3.16}$$

where $w = \hat{\sigma}_n$, which is estimated by Equation 3.9.

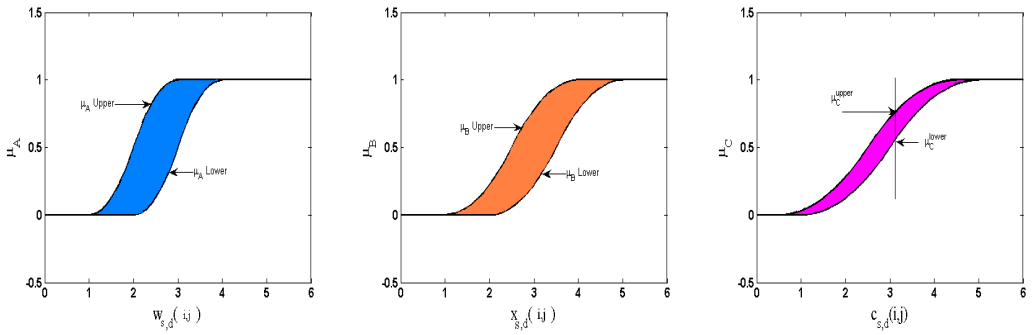


Figure 3.13: Interval Type II Fuzzy Membership functions for the fuzzy variables. Left: large magnitude wavelet coefficient, and Middle: large neighborhood value.Right: Large correlation map value

Thirdly, fuzzy rules are linguistic *IF-THEN* constructions that have the general form "IF A THEN B", where A and B are (collections of) propositions containing linguistic variables. A is called the premise or antecedent and B is the consequence of the rule. In Fuzzy Rule 1 we can distinguish two linguistic variables for the consequent: large wavelet coefficients $|w_{s,d}(i,j)|$, large neighborhood values $|x_{s,d}(i,j)|$ and large correlation map value $|c_{s,d}(i,j)|$. All linguistic terms are modeled as IT2 FSs discussed above.

A designed T2 FLS rule for this proposed algorithm,

IF $|x_{s,d}(i,j)|$ is a large variable

AND $|w_{s,d}(i,j)|$ is a large coefficient

AND $c_{s,d}(i,j)$ is a large correlation map value

OR $|x_{s,d}(i,j)|$ is a large variable

THEN $w_{s,d}(i,j)$ is a signal of interest

3.3.2 Fuzzy Inference Engine and Defuzzification

Inferring the IT2 FLS Rule discussed in subsection 3.3.1 can be considered as an intersection and a union of three fuzzy sets. The intersection $A \cap B \cap C$ of three fuzzy sets A, B and C is generally specified as a t-norm; while the union $A \cup B$ of two fuzzy sets A and B is specified as a co-norm. In fuzzy logic, t-norms (roughly the equivalent of *AND* operations) and co-norms (roughly the equivalent of *OR* operations) are used to represent the intersection of three fuzzy sets and the union of two fuzzy sets, respectively. Other well known t-norms together with their dual co-norms are introduced in [45]. From all possible t-norms and co-norms, the product and the probabilistic sum are chosen to represent *AND* and *OR*, respectively.

t-norm product of three IT2 FSs are calculated by equation 3.17 and 3.18

$$\delta_{s,d}^{upper}(i,j) = \mu_A^{upper}(|w_{s,d}(i,j)|) \cdot \mu_B^{upper}(|x_{s,d}(i,j)|) \cdot \mu_C^{upper}(|c_{s,d}(i,j)|) \tag{3.17}$$

$$\delta_{s,d}^{lower}(i,j) = \mu_A^{lower}(|w_{s,d}(i,j)|) \cdot \mu_B^{lower}(|x_{s,d}(i,j)|) \cdot \mu_C^{lower}(|c_{s,d}(i,j)|) \tag{3.18}$$

The results are used as new IT2 FSs to calculate the fuzzy value by applying the probabilistic sums.

$$\beta_{s,d}^{upper}(i,j) = \delta_{s,d}^{upper}(i,j) + \mu_B^{upper}(|x_{s,d}(i,j)|) - \delta_{s,d}^{upper}(i,j) \cdot \mu_B^{upper}(|x_{s,d}(i,j)|) \tag{3.19}$$

$$\beta_{s,d}^{lower}(i,j) = \delta_{s,d}^{lower}(i,j) + \mu_B^{lower}(|x_{s,d}(i,j)|) - \delta_{s,d}^{lower}(i,j) \cdot \mu_B^{lower}(|x_{s,d}(i,j)|) \tag{3.20}$$

As can be seen in equation 3.17 and 3.18, there are two consequences for the rules each corresponding to the upper and lower limit. These consequences represent the degree by which the wavelet coefficient should be shrunk. The subsection 3.3.3 explains how to shrink the wavelet coefficients of a noisy image.

3.3.3 Output of Algorithm

The shrinkage rule of the proposed method for scale s, direction d and position (i,j) is calculated as follows:

$$\hat{y}_{s,d}(i,j) = \gamma(w_{s,d}(i,j), x_{s,d}(i,j), c_{s,d}(i,j)) \cdot w_{s,d}(i,j) \tag{3.21}$$

with $y_{s,d}(i,j)$ the shrink output coefficient for scale s, direction d and position (i,j) and where $\gamma(w_{s,d}(i,j), x_{s,d}(i,j), c_{s,d}(i,j))$ (simplified as $\gamma_{s,d}(i,j)$, calculated by equation 3.22) is the degree of activation of the fuzzy Rule for the wavelet coefficient $w_{s,d}(i,j)$. This value indicates the membership degree of the interesting signal for the wavelet coefficient $w_{s,d}(i,j)$ in the designed IT2 FLS.

$$\gamma_{s,d}(i,j) = \frac{\beta_{s,d}^{upper}(i,j) + \beta_{s,d}^{lower}(i,j)}{2} \tag{3.22}$$

3.3.4 Algorithm

As discussed above, the complete IT2 FLS based wavelet shrinkage algorithm is summarized as following,

- **Step 1: Fuzzification**

  Choose the proper image features, each corresponding to a linguistic variable. Each feature is then fuzzified by their appropriate Membership Function (MF). In this study, wavelet coefficient $w_{s,d}(i,j)$, neighborhood value $x_{s,d}(i,j)$, and correlation value $c_{s,d}(i,j)$ are chosen to represent image features. The sigmoid IT2 FMs are exploited to fuzzify these three fuzzy sets. For details refer to subsection 3.3.1.

- **Step 2: Fuzzy Rule Building**

  Build knowledge-based fuzzy if-then rules. Utilizing the fuzzy variables to represent the image features, fuzzy rules are obtained. In this study, only one fuzzy rule is built:

  IF ($|x_{s,d}(i,j)|$ is a large variable

  AND $|w_{s,d}(i,j)|$ is a large coefficient

  AND $c_{s,d}(i,j)$ is a large correlation map value

  OR $|x_{s,d}(i,j)|$ is a large variable

  THEN $w_{s,d}(i,j)$ is a signal of interest

- **Step 3: Fuzzy Inference Engine (Rule Firing)**

  For each rule, the fuzzified inputs are combined and a rule strength is obtained. This rule strength is used as the consequence of the rule [45]. In this study, the strength of the rule built in Step 2 is computed by equation 3.17 and 3.18. Once the consequences from various rules have been obtained, they are combined to obtain an output distribution range. In this study, only one fuzzy rule is being used. In general, the probabilistic sum conorm is used to combine various rules.

- **Step 4: Type Reduction and Defuzzification**

  A value representing the scaling factor for the wavelet coefficient at (i, j) is obtained by fuzzy type reducing and defuzzifying the output. There are more than one approach to defuzzify the output distribution [45]. The one employed in this study is the center-of-sets type reduction, the average of the upper and lower value is utilized,

$$\gamma_{s,d}(i,j) = \frac{\beta_{s,d}^{upper}(i,j) + \beta_{s,d}^{lower}(i,j)}{2} \tag{3.23}$$

  Finally, reconstructing the shrunk coefficients $\hat{y}_{s,d}(i,j)$ by using reverse UDWT, the denoised image is obtained.

  Now, a new coefficient can be obtained at position (i, j) by applying Equation 3.23 to Equation 3.21, which has been reduced of noise. The filtering is performed at each scale of the wavelet domain and for each of the three detail coefficients $d \in \{H, V, D\}$.

- **Step 5: Reconstruction**

  Once all the coefficients have been found, applying the inverse 2D-UDWT to an input nature image will result in a denoised final image.

3.4 Summary

This chapter has investigated the possibilities of a soft computing tool, interval type-2 fuzzy logic system (IT2 FLS), to improve the results of a wavelet thresholding procedure. This IT2 FLS is designed for the application to image noise reduction and it combines three objectives:

- capturing the correlations in wavelet coefficients of each subband. Since there is no perfect model for the correlations, an interval type-2 fuzzy set is designed for describing the intra-correlations.
- the correlations in wavelet coefficients of the same scale is also important. Another interval type-2 fuzzy set is applied to represent the inter-correlations.
- a wavelet coefficient itself must also be considered. Therefore the third interval type-2 fuzzy set is assigned to the local wavelet coefficients.

After designing this algorithm carefully, it is expected that this algorithm succeeds in finding more noisy coefficients. In Chapter 5, the performance of noise reduction of this algorithm will be discussed.

Chapter 4

Genetic Algorithm and Fuzzy Neural Network Based Algorithm

This algorithm is an extension of the concept of image fusion. Originally, image fusion deals with combining different sources of information for intelligent systems. The information is signals delivered by different sensors and images from various modalities.

In recent years, the image fusion concept has been applied to image denoising. For example, in [41] and [22], two denoised images of curvelets and wavelets are fused to keep high image quality and remove artifacts. In [81], a fusion technology based on neural networks is presented. To solve trade-offs between the noise attenuation and the detail preservation in image denoising, this technology fuses the results of the multilevel FIR-median hybrid filter (MFMHF) and the median filter. A feature-level image fusion scheme is introduced in [84]. In this scheme, the features are captured by segmenting source images, and then fusion strategies are implemented according to the contributions of the related features on the images.

In summary, no matter what kind of image fusion schemes, the main idea of these fusion schemes is to preserve the detailed information and eliminate the noise of images. In this chapter, a new fusing algorithm based on two filtered images, hidden Markov tree based wavelets (detail preservation) and type-2 fuzzy logic system based undecimated wavelets (noise attenuation), is proposed. The image fusion scheme is based on a fuzzy neural network (FNN), and a modified genetic algorithm is applied to optimize the parameter values of the FNN.

First, this chapter briefly introduces image fusion, fuzzy neural network, and genetic algorithm. It then discusses the proposed FNN algorithm.

4.1 Brief Introduction to Image Fusion

The goal of image fusion is to produce a single image from a set of input images. The fused image should have more complete information which is more useful for

human or machine perception. It has two main advantages: improving reliability (by redundant information) and improving capability (by complementary information) without introducing artifacts or inconsistencies which will distract human observers or the following processing. The second advantage is very useful for image denoising since some wavelet transform based denoising algorithms cause artifacts while removing the noise.

4.2 Brief Introduction to Artificial Neural Networks

An artificial Neural Network (ANN) is a computing tool of artificial intelligence (AI) (others include fuzzy logic, genetic algorithms, and expert systems). "An ANN is a massively parallel distributed processor that stores experimental knowledge, and this knowledge is acquired by a learning process and is stored in the form of parameters of the ANN" [34]. It has the properties of a parallel distributed architecture, learning and generalizing, fault tolerance, nonlinearity, and adaptivity. The learning process in ANNs can be unsupervised or supervised. In an unsupervised learning process, an ANN extracts the features from the input data based on a predetermined performance measurement. In supervised learning, an ANN is fed with the input patterns and the desired output patterns. The parameters of the ANN are adapted such that an input pattern results in the desired pattern at the output of the ANN [30]. The neural network built in the study is based on supervised learning.

4.2.1 Structure of ANNs

Similar to the structure of the brain, ANNs also consist of "neurons" and "connectors". Several architectures of ANNs have been developed, and generally they all have the same components. In a feed-forward ANN, neurons are aligned in rows, called layers. Each neuron in a layer is connected to all neurons of the preceding and succeeding layer [30].

Artificial neural networks can be classified according to the structure that they exhibit. The popular neural networks are: feed forward neural network, radial basis function (RBF) network, Kohonen self-organizing network, and Hopfield network [34].

Figure 4.1 represents the structure of a multi-layered feed forward network. The neurons in this ANN model are grouped in layers which are connected to the direction of the passing signal (from left to right in this case). There are no lateral connections within each layer and also no feed backward connections within the network. The best-known ANN of this type is the perceptron network [34], which is the structure of the neural network applied to this thesis.

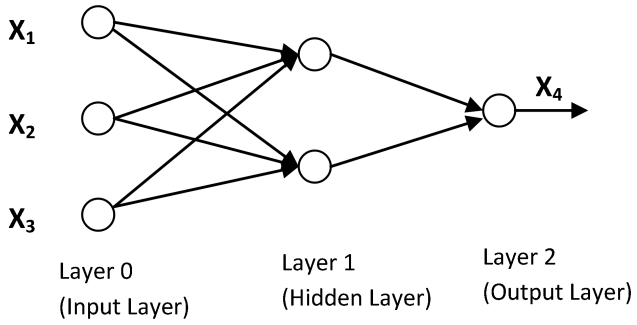


Figure 4.1: Structure of a multi-layered feed forward network [30].

Since ANNs are nonlinear, applying neural networks in modeling nonlinear systems has received extensive attention in the last several decades. Due to the excellent performance of ANNs in modeling nonlinear relationship among a large number of variables, interests in ANNs among researchers has been growing for many years.

In the past decade, most applications of neural networks were based on the back propagation network (BP) algorithm that uses the gradient-descent method to minimize an error function. The BP algorithm is more suitable than an analytical approach for solving the problems which need to predict the output of a complex and nonlinear physical system to its inputs [34]. Since the generalization capability of BP neural networks is low for solving highly nonlinear function mapping problems, the BP algorithm has been considerably improved during the last few years [38, 72].

4.2.2 Back Propagation Learning Algorithm (BPL)

The following is the BP algorithm which is applied to the fuzzy neural network (FNN) in this thesis. The seven steps involved in the back propagation learning algorithm of a given neural network are briefly given below, for details, please refer to [34]:

- **Step 1: Initialization**

  The weights and thresholds are initialized to preset values or small random values.

- **Step 2: Data Selection**

  An appropriate data set is selected for the training of the network. The data set should include input(s) and output(s) to the network which should cover the whole horizontal data space. Assume $(x(k), t(k))$ are the input-output pair of vectors used for training of the neural network.

- **Step 3: Forward Propagation**

  For any iteration k, forward the kth input from the input layer and propagate it all the way through the network to the output layer. Calculate $tot_i^{(l)}$ and $o_i^{(l)} = f(tot_i^{(l)})$ for all the neurons in all layers.

  $tot_i^{(l)}$: the sum of all signals reaching node (i) at hidden layer (l) coming from previous layer $(l-1)$.

  $o_i^{(l)}$: the output of the jth neuron at layer $(l-1)$ (the one located just before layer l).

- **Step 4: Error Calculation**

  Update the total error value by using $E = E+E(k)$ and calculate the gradient δ_i^L for the neurons of the output layer L by using the following equation,

$$\delta_i^{(L)} = (t_i - o_i^{(L)})(f'(tot_i^{(L)})) \qquad \text{for } l \prec L \tag{4.1}$$

- **Step 5: Back propagation**

  Update the weights according to,

$$\Delta w_{ij}^{(l)} = \eta \delta_i^{(l-1)} o_j^{(l-1)} \qquad \text{for } l \prec L$$

  using equation 4.1, and proceeding backward using,

$$\delta_i^{(l)} = f'(tot_i^{(l)}) \sum_{p=1}^{n_l} \delta_p^{(l+1)} w_{pi}^{(l+1)} \qquad \text{for } l \prec L$$

- **Step 6: Epoch Training**

  Repeat the whole process starting from step 2 until all training data are passed through the process once. One epoch training is completed once all exemplars (training data) have been used.

- **Step 7: Error validation**

  Cumulative error E in the output layer is checked with the maximum tolerable error E_{max}. The network is said to be trained if $E < E_{max}$; otherwise the training process should continue for another epoch, and the process is repeated till the error condition: $E < E_{max}$ is satisfied.

4.3 Fuzzy Neural Networks

As discussed in Chapter 3, section 3.2, a fuzzy model provides a mechanism to represent and manipulate uncertainty within an image. Section 4.2 shows that an

ANN models a nonlinear system whose performance can be measured by efficiency and accuracy [34]. Therefore, both a FLS and an ANN are a powerful computing tool in AI.

However, both of these soft computing tools have some limitations. For instance, in a fuzzy logic system, a set of simple "if-then" rules usually represent behaviors of a given system. However, these rules are unable to tackle knowledge stored in the form of numerical data. Furthermore, the extraction of "if-then" rules is very tedious or even impossible to attain for the data sets with large numbers of patterns. The problem becomes more challenging when the knowledge about a system is stored in both forms: linguistic (fuzzy sets) and numerical (data sets). On the other hand, neural networks are unable to explicitly represent knowledge. For instance, it is difficult for a neural network to explicitly quantize the meaning of weights among the nodes once it has been trained. In addition, it is hard to incorporate additional knowledge into the system without retraining the neural network. It is even more difficult to extract the representation of knowledge from the linguistic data patterns [34].

To overcome the limitations of both fuzzy and neural systems, researchers have proposed incorporating fuzzy logic reasoning within a learning architecture of neural networks [74]. A lot of research has tackled the issues of constructing neural networks using fuzzy neurons. The integration of fuzzy logic systems with neural networks reduces the limitations of fuzzy systems in terms of lack of learning while strengthening the neural network features in terms of explicit knowledge representation.

4.3.1 Architecture of Proposed Fuzzy Neural Network

In this study, a Fuzzy Neural Network (FNN) is built for fusing the filtered images. The FNN architecture is a standard back-propagation (BP) neural network with fuzzy sets applied to the input layer. A fuzzy layer is added between the input layer and the hidden layer of the BP network to represent linguistic uncertainty of being fused images. The BP network is used to train the FNN to get optimal fusing results. The structure of the proposed FNN is illustrated in Figure 4.2.

The proposed FNN has four layers inspired by [81], the input layer, the fuzzy layer, the hidden layer, and the output layer. In the input layer there are two neurons i_1 and i_2. i_1 and i_2 are matrices which represent the intensity value of each pixel within the input images.

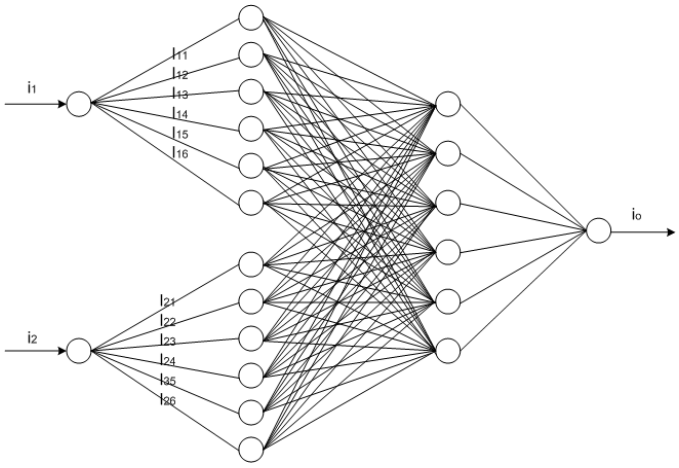


Figure 4.2: The structure of the proposed FNN

4.3.2 Fuzzy Membership Function of Each Neuron

Each neuron of the input layer is mapped to six neurons with the membership function defined as follows,

$$I_{jk} = \frac{1}{1 + \exp(-\frac{i_j - m_k}{s_k})} \qquad \text{when } k = 1, 2 \;\; j = 1, 2 \tag{4.2}$$

$$I_{jk} = \exp\left(-\left(\frac{i_j - m_k}{s_k}\right)^2\right) \qquad \text{when } k = 3, 4 \;\; j = 1, 2 \tag{4.3}$$

$$I_{jk} = 1 - \frac{1}{1 + \exp(-\frac{i_j - m_k}{s_k})} \qquad \text{when } k = 5, 6 \;\; j = 1, 2 \tag{4.4}$$

where i_j is the jth input neuron, I_{jk} is the result of the kth fuzzy neuron fusing the number jth input neuron. m_k and the s_k are the parameters of the FNN which will be optimized by the Genetic Algorithm (GA).

4.4 Brief Introduction to Genetic Algorithm

The least squares based optimization of a back propagation network can be trapped in a local minimum of a nonlinear objective function because it is derivative based. Genetic algorithms (GAs) are derivative-free, stochastic optimization methods, and therefore are less likely to get trapped [34].

Actually, GAs are general purpose search algorithms. A genetic algorithm likes mimics the evolution of populations. First, different possible solutions to a problem are generated. They are tested for their performance, that is, how good a solution is in terms of a cost function. A fraction of the good solutions is selected, and the

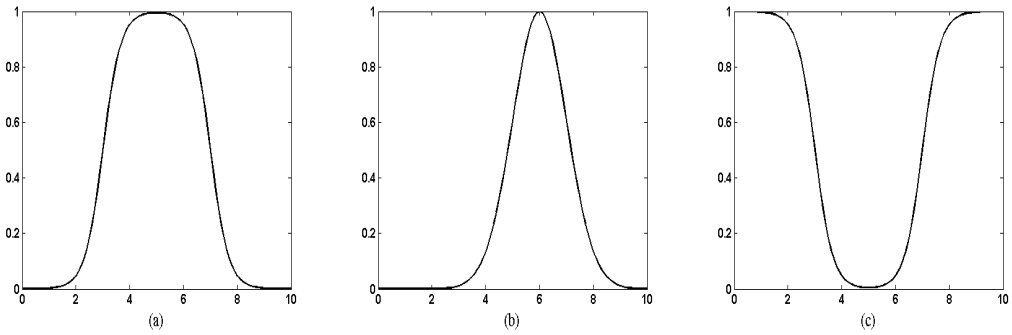


Figure 4.3: Fuzzy Membership functions of fuzzy neurons. (a) Fuzzy membership function 4.2. (b) Fuzzy membership function 4.3. (c) Fuzzy membership function 4.4

.

others are eliminated (survival of the fittest). Then the selected solutions undergo the processes of reproduction, crossover, and mutation to create a new generation of possible solutions, which is expected to perform better than the previous generation. Finally, production and evaluation of new generations is repeated until convergence. Such an algorithm searches for a solution from a broad spectrum of possible solutions, rather than where the results would normally be expected [30]. The robust part of GAs is their ability to exploit the information accumulated about an initially unknown search space in order to bias subsequent searches into useful subspaces. The key feature of GAs is offering a valid approach to problems requiring effective search techniques, especially in large, complex, and unknown search spaces. The penalty is the high computational cost.

In this study, a modified GA is used to optimize the parameters of the fuzzy neural network created in this thesis.

4.4.1 Procedures for Implementing the Genetic Algorithm

The brief introduction of the GA implementing procedure is listed as follows. It is modified from [34] for optimization of the parameters of the FNN employed in the proposed algorithm. The cost function, which is used to test the solutions generated in each iteration, is replaced by the FNN introduced in section 4.3.

- **Step 1:** initialize the variables which the GA needs, such as population size and the mutation rate, and the initial population (a set of solutions represented by chromosomes.).
- **Step 2:** perform the following steps until the redefined condition is achieved:
  - evaluate the FNN for each chromosome
  - select a new generation

- calculate the total fitness of the population
- calculate the probability of selection for each chromosome
- calculate a cumulative probability for each chromosome
- generate a random float number $r \in [0, 1]$
- make sure that r is located in between $[0, 1]$
- apply genetic operators (crossover, mutation)
- choose the new offspring as the current population

- **Step 3:** Go back to step 2 if the optimization requirement is not attained

  The process of implementing this GA is summarized in the flowchart diagram (see Figure 4.4). For the detailed information about the search process, please refer to [27].

4.4.2 Flowchart of GA

The flowchart of the modified GA is shown in Figure 4.4. In the flowchart, training FNN replaces the cost function which is used in common GAs.

4.5 Summary

This chapter has investigated the possibilities of exploiting soft computing tools, FNN and GA, to improve the performance of noise reduction in images. There are three objectives:

First, a neural network is designed for fusing two filtered images. With the purpose of achieving a better result, a fuzzy layer is added between the input layer and the hidden layer of the neural network.

Second, a modified genetic algorithm is applied the to find optimal parameters for this fuzzy neural network.

Third, a better result by fusing filtered images to make a trade-off between edge preservation and noise suppression is expected.

The performance of this proposed algorithm, GA-FNN, will be demonstrated and discussed in Chapter 5.

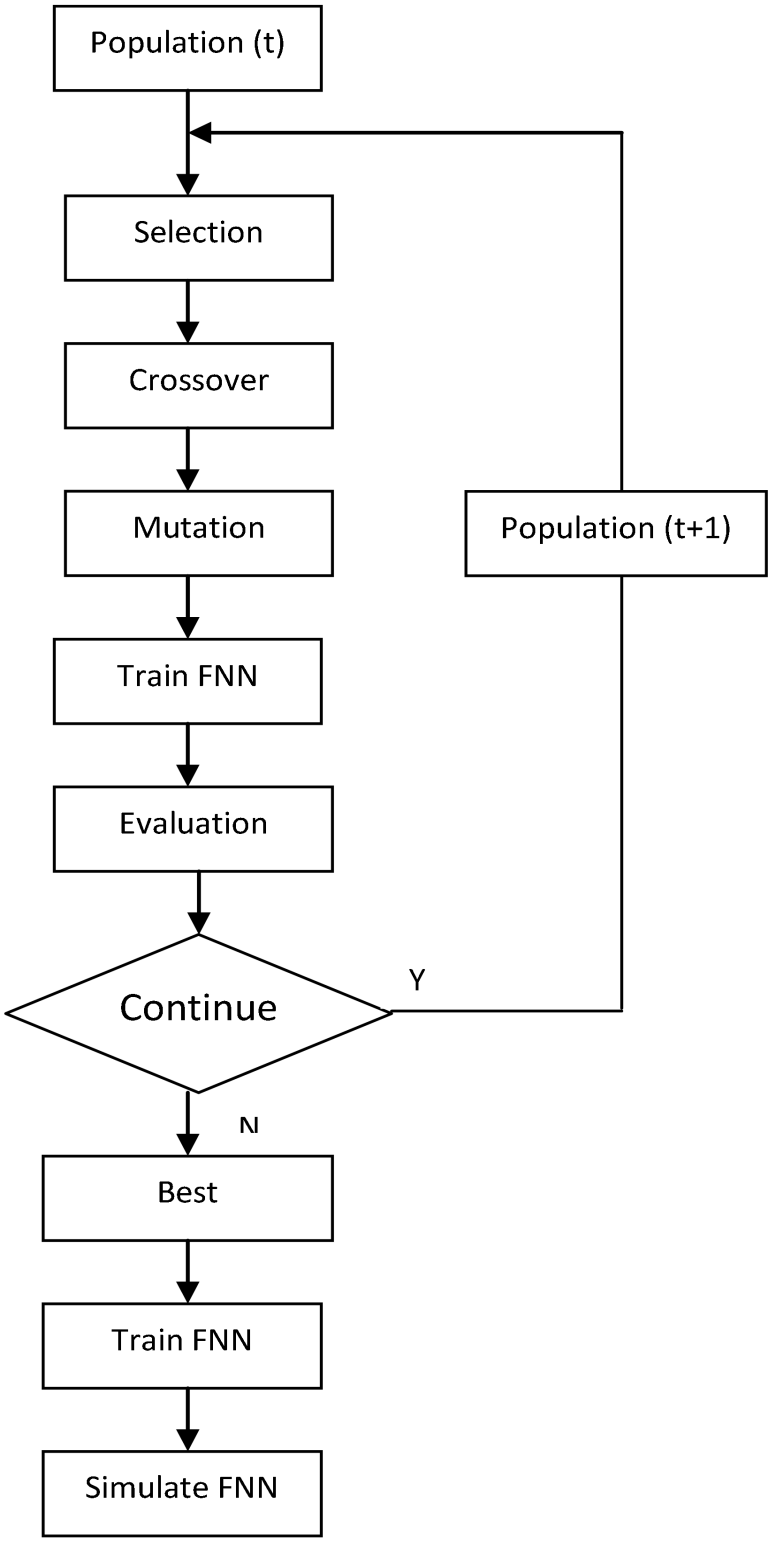


Figure 4.4: The data flowchart of the genetic algorithm modified from [27].

Chapter 5

Performance of Proposed Algorithms

Complete evaluation of the performance of an algorithm is challenging. Generally, it includes the quality of denoised images, computational cost, and universality of the denoising algorithms. As discussed in Chapter 2, there are many methods to assess the quality of denoised images. However, there is no single method which assesses the quality of denoised images fully and precisely. Some are objective methods which exploit mathematics to assess the quality of images; some are based on the Human Vision System (HVS), which attempt to assess the image quality subjectively with objective numbers. However, since the HVS is highly subjective, it is hard to represent with objective numbers. So far there has been no HVS model which describes the real HVS precisely.

In this study, both mathematical and HVS based methods, such as MSE, PSNR, image quality index (IQI) and information theoretic criterion (ITC) are exploited. Although those methods can not provide a full and complete assessment of the denoised image quality, they are sufficient for comparing the denoising results with other denoising approaches. In this thesis, traditional denoising methods, such as median filter, and the state of the art methods, such as VisuShrinkage (VisuS) with soft thresholding [18, 19, 20], BayesShrinkage (BayesS) [11, 59] and Type-1 FuzzyShrinkage (T1 FShrink) [62], Hidden Markov Tree (HMT), are used to compare with the proposed algorithms, Type-2 Fuzzy Shrink (T2 FShrink) and GA based FNN (GA-FNN). In addition, the proposed algorithms are also evaluated from different points of view, such as the computational cost and universality, in order to get a full and complete evaluation of the denoising performance.

5.1 Test Image Selection

As mentioned in Chapter 1, in this study, only natural images are involved. The purpose of choosing images for testing is to investigate the performance of different

denoising approaches—-how they remove the noise while preserving the detailed information, such as edges and lines, of these images. In order to achieve a thorough evaluation of the proposed algorithms, a set of standard test images are chosen from a popular image database, the USC-SIPI Image Database (University of Southern California). Those images are digital image files used across different institutions to test image denoising algorithms. Therefore, different algorithms can be compared using the same standard test images. The test images used in this study are chosen from Volume 3: Miscellaneous. For example, image lena, Baboon, and etc., are selected for testing the algorithms. These images are listed below in Figure 5.1.

5.2 Image Quality Assessment

To show the results of different approaches and compare the quality of filtered images, different kinds of noise were added to the test images. The first experiment was done only on Gaussian noise, and the second experiment was on a mixed noise model with Gaussian and impulsive noise. All simulations were implemented with the computer language of MATLAB on a Pentium IV personal computer (1.6 GHz, 512MB).

5.2.1 Image Quality Assessment for Pure Gaussian Noise

In this section, the results of the test images distorted by pure Gaussian noise are demonstrated and discussed.

- **Test Process**

  In this experiment, Gaussian Noise with variance values ranging from 0.01 to 0.1 are added to the test images shown in Figure 5.1. To obtain a thorough evaluation of the proposed algorithms, comparisons are made among (1) traditional filter, (2) well-known wavelet-based algorithms, and (3) recently developed fuzzy shrinkage method. More precisely, the algorithms selected for comparison are,

  1. **Traditional Filter** Median filter
  2. **Wavelet-Based Algorithms** VisuShrinkage(VisuShrink) [18, 19, 20]. BayesShrinkage (BayesShrink) [11, 59]. Hidden Markov Tree(HMT) algorithm [23].
  3. **Fuzzy Set Based Algorithm** FuzzyShrinkage (T1 FShrink) [62]

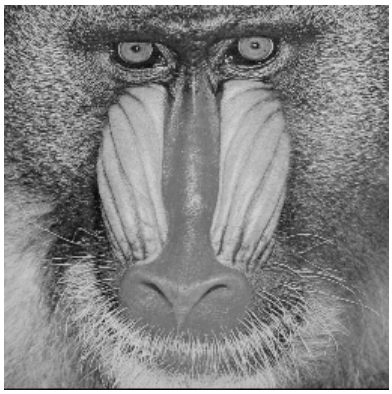
(a) Test Image “lena”

(b) Test Image “man”

(c) Test Image “house”

(d) Test Image “lake”

(e) Test Image “pepper”



(f) Test Image “baboon”

Figure 5.1: The Test Images. The first row represents natural images of human beings, one has relative clear background (left), another has a quite complicated background (right). The second row indicates a house with many straight lines (left) and a natural scene (right). In the third row, one image has many curve edges (left) and another has a lot of fine hair (right) which may be considered as noise by denoising algorithms. (Courtesy of Signal and Image Processing Institute, University of Southern California.)

- **Conditions of Implementation** In this study, specific test conditions are set to compare the results.

  1. The stationary discrete wavelet transform with Daubechies wavelets (db4), four resolution scales and neighborhood of size 9×9 ($K = 4$), are employed to test all algorithms for result comparisons.
  2. The feedforward backpropagation network is a built-in "newff" network in MatLab. The min and max values for all input elements are 0 and 1; Transfer function of ith layer is "tansig"; Backpropagation network training function is "trainlm"; Backpropagation weight/bias learning function is "learngdm"; Performance function is "mse".
  3. The parameters for GA to optimize the parameters of the FNN: population size is 20; mutation rate is 0.1; fraction of population kept: 0.5; maximum number of iterations: 10; cost function is in terms of PSNR.
  4. The best parameters are obtained under condition 3. Filtered "baboon" images by HMT and T1 FShrink algorithms are chosen to achieve the best parameters.

  In order to achieve a clear and full comparison of the performance of all algorithms, the experiments are carried out for those six test images, see Figure 5.1, each of them has the size 256×256.

- **Observation and Discussion**

  1. **Results of Image "lena"**

     In Figure 5.2, it can be observed that median, VisuShrink and BayesShrink filters preserve the edges in "lena" but do not remove noise completely. T1 FShrink and HMT do better job than the former three algorithms. However, there are some artifacts in the filtered image of T1 FShrink algorithm. This proves the statement (Refer to Chapter 3) that the properties of the single scale discrete wavelet transform and the shift variant of DWT can cause the artifacts. T2 FShrink, solves this problem using multilevel stationary DWT. But T2 FShrink smooths the edges in images a little more than HMT and T1 FShrink do. GA-FNN outperforms all other algorithms. These observations can be viewed in the filtered edge intensity profile of pure Gaussian noise model, Figure 5.14.

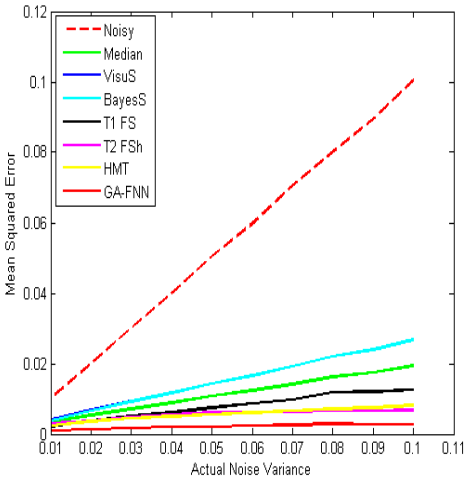
     In addition, from the performance of all algorithms assessed by different tools (see Figure 5.3), combined with the performance at specific noise variance values: 0.01, 0.05, 0.1(see Table 5.1, 5.2, 5.3), it can be concluded that GA-FNN and T2 FShrink algorithms perform the best among those algorithms in terms of MSE, PSNR, IQI, and ITC at most of the noise variance values, especially when the noise variance value is larger.

(a) Original Image (b) Noisy Image (c) Median Filter

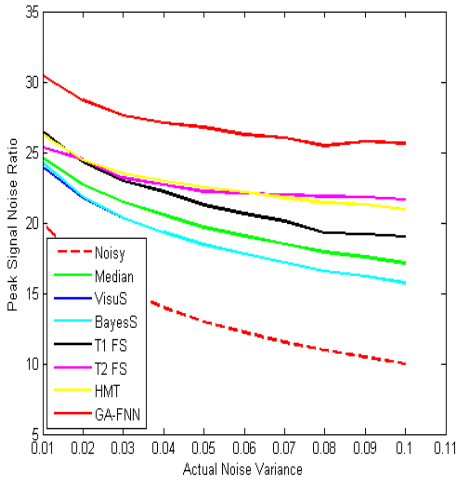
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

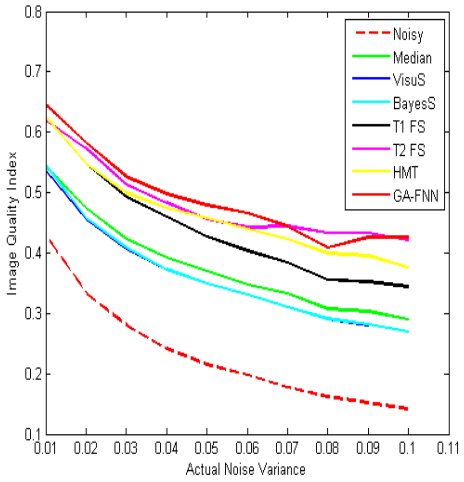
Figure 5.2: Results of "lena" with Gaussian noise variance 0.05. From the point of human vision, median, VisuShrink and BayesShrink filters do not remove noise completely. T1 FShrink causes some artifacts. T2 FShrink and GA-FNN outperforms other algorithms, but T2 FShrink smooths the edges more.



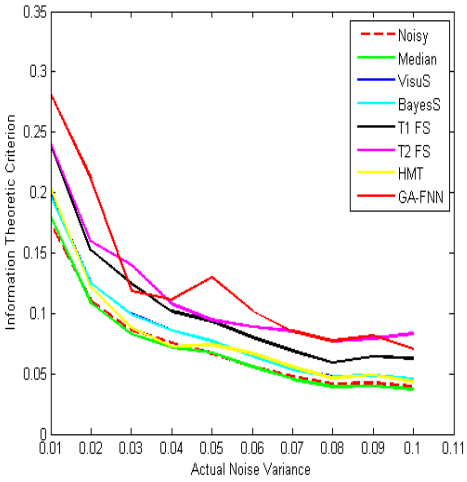
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.3: Performance of removing pure Gaussian noise (image "lena"). It can be observed that GA-FNN and T2 FShrink algorithms perform the best among those algorithms in terms of MSE, PSNR, IQI, and ITC, especially when the noise variance value is larger.

Table 5.1: Results of Image "lena" with Gaussian noise variance 0.01

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01014	19.9713	0.429	0.1741
Median	0.003462	24.6125	0.543	0.1802
VisuS	0.004033	23.9536	0.536	0.1982
BayesS	0.003764	24.2513	0.545	0.1963
HMT	0.002371	26.2645	0.6253	0.2034
T1 FShrink	0.002254	26.4923	0.6242	0.2391
T2 FShrink	0.002916	25.9536	0.6219	0.2413
GA-FNN	0.000905	30.4342	0.6463	0.2823

Table 5.2: Results of Image "lena" with Gaussian noise variance 0.05

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05032	12.9823	0.2152	0.06693
Median	0.01073	19.6914	0.3719	0.06732
VisuS	0.01429	18.4291	0.3489	0.07722
BayesS	0.01431	18.4323	0.3492	0.07714
HMT	0.00557	22.5384	0.4574	0.07414
T1 FShrink	0.007523	21.2426	0.4269	0.0935
T2 FShrink	0.005933	22.2725	0.4592	0.09913
GA-FNN	0.002094	26.7926	0.4791	0.1299

Table 5.3: Results of Image "lena" with Gaussian noise variance 0.1

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.10051	9.97923	0.1415	0.03953
Median	0.01935	17.1329	0.2899	0.03703
VisuS	0.02675	15.7264	0.2691	0.04583
BayesS	0.02675	15.7263	0.2691	0.04583
HMT	0.00808	20.9242	0.3756	0.04331
T1 FShrink	0.01255	19.0143	0.3441	0.06254
T2 FShrink	0.006743	21.7123	0.4224	0.08239
GA-FNN	0.002728	25.6414	0.4256	0.07012

In above Table 5.1, 5.2, and 5.3, the proposed algorithms and the best two results in terms of MSE, PSNR, IQI and ITC are underlined.

2. **Results of Image "man"**

   The performance of all test algorithms at the noise variance 0.05 is illustrated in Figure 5.4. Subjectively, these results show that median filter, VisuShrink and BayesShrink preserve the edges in "man" but do not remove noise completely. T1 FShrink and HMT do better job but still causes some artifacts in the filtered image of T1 FShrink. T2 FShrink solves this problem using multilevel stationary DWT. But T2 FShrink smooths the edges in the image a little more than HMT does, GA-FNN algorithm outperforms all the algorithms. All observations can be proved in the filtered edge intensity profile of pure Gaussian noise model, Figure 5.14.

   Furthermore, from the thorough assessment by diverse mathematical analysis, see Figure 5.5, and the performance at specific noise variances:0.01, 0.05, 0.1 illustrated in Table 5.4, 5.5, and 5.6, the following observations can be attained: GA-FNN performs the best among those algorithms in terms of MSE, PSNR, and IQI at different noise level; T2 FShrink outperforms other algorithms in terms of ITC at any noise levels, and both of them perform better than other algorithms at higher noise levels in terms of mathematical assessment tools.

   In summary, the proposed algorithms perform the best in natural images with human beings, especially in images distorted by higher level Gaussian noise.

3. **Results of Image "house"**

   The performance of all test algorithms at the noise variance 0.05 is illustrated in Figure 5.6. From these results, it can be observed that median filter, VisuShrink and BayesShrink preserve the edges in "house" but do not remove noise completely. T1 FShrink and HMT do better job than the former three algorithms, especially HMT obtains a better result at lower noise level (See Table 5.7). However, T1 FShrink still causes the artifacts in the image. T2 FShrink, remove more noise but smooths the edges in "house" image a little more than HMT does, GA-FNN outperforms the all the algorithms. All observations can be perceived in Figure 5.14.

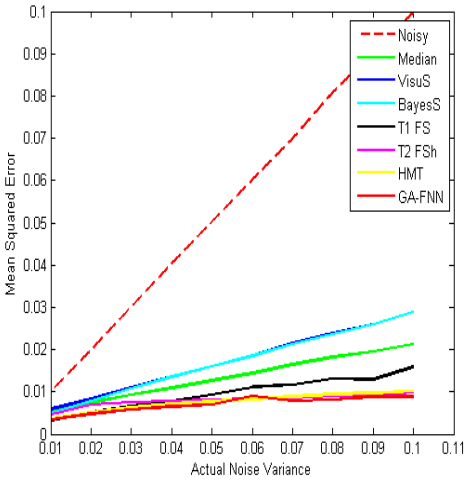
   From the fully mathematical analysis results in Figure 5.7, and the performance at specific noise variances:0.01, 0.05, 0.1 shown in Table 5.7, 5.8, and 5.9, the following observations can be attained, GA-FNN performs the best among those algorithms in terms of MSE, PSNR, IQI, and ITC at most of noise levels; T2 FShrink outperforms other algorithms in terms of MSE, PSNR, IQI, and ITC at higher noise levels, but slightly worse than HMT at lower noise levels in terms of MSE, PSNR, and IQI, see Figure 5.7.

(a) Original Image (b) Noisy Image (c) Median Filter

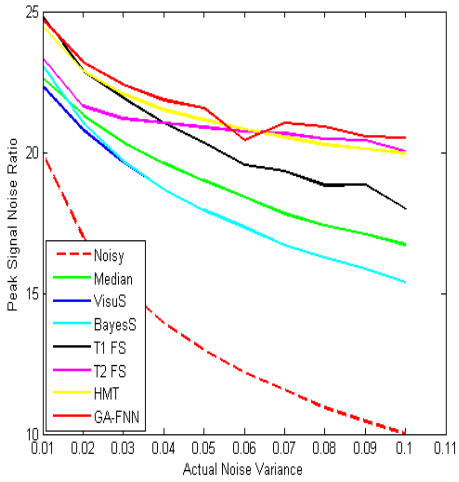
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

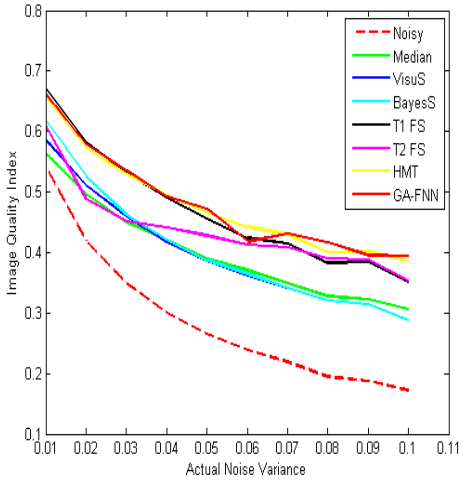
Figure 5.4: Results of "man" with Gaussian noise variance 0.05. From the point of human vision, median, VisuShrink and BayesShrink filters do not remove noise completely. T1 FShrink causes some artifacts. T2 FShrink and GA-FNN outperforms other algorithms, but T2 FShrink smooths the edges more.



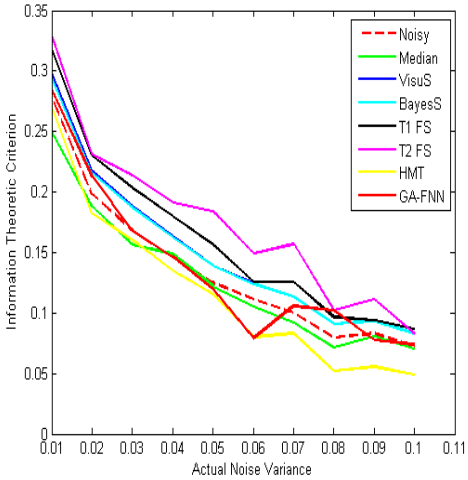
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.5: Performance of removing pure Gaussian noise (image "man"). GA-FNN performs the best among those algorithms in terms of MSE, PSNR, and IQI at different noise levels; T2 FShrink outperforms other algorithms in terms of ITC at different noise levels, and both of them perform better than other algorithms at higher noise levels in terms of mathematical assessment tools.

Table 5.4: Results of Image "man" with Gaussian noise variance 0.01

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.010057	19.9753	0.5423	0.2780
Median	0.005459	22.6287	0.5639	0.2492
VisuS	0.005836	22.3386	0.5861	0.2981
BayesS	0.004942	23.0614	0.6187	0.2936
HMT	0.003566	24.4776	0.6567	0.2706
T1 FShrink	0.003294	24.8230	0.6708	0.3177
T2 FShrink	0.004643	23.3325	0.6101	0.3296
GA-FNN	0.003380	24.7110	0.6621	0.2841

Table 5.5: Results of Image "man" with Gaussian noise variance 0.05

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05019	12.9938	0.2657	0.1250
Median	0.01261	18.9933	0.3905	0.1221
VisuS	0.01606	17.9423	0.3858	0.1395
BayesS	0.01601	17.9565	0.3872	0.1393
HMT	0.007678	21.1476	0.4656	0.1154
T1 FShrink	0.009229	20.3487	0.4555	0.1564
T2 FShrink	0.008123	20.9026	0.4214	0.1785
GA-FNN	0.006964	21.5715	0.4718	0.1198

Table 5.6: Results of Image "man" with Gaussian noise variance 0.1

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09979	10.0091	0.1722	0.07251
Median	0.02126	16.7235	0.3064	0.07014
VisuS	0.02887	15.3955	0.2882	0.08314
BayesS	0.02888	15.3947	0.2884	0.08304
HMT	0.01009	19.9597	0.3859	0.04885
T1 FShrink	0.01584	18.0029	0.3512	0.06920
T2 FShrink	0.009898	20.0446	0.3448	0.08659
GA-FNN	0.008878	20.5170	0.3944	0.07403

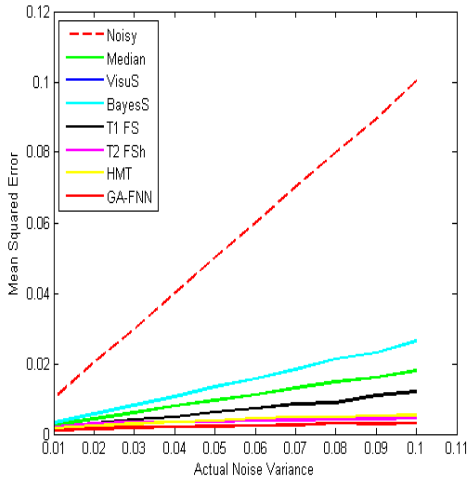
In Table 5.4, 5.5, and 5.6 the proposed algorithms and the best two results are underlined.

(a) Original Image (b) Noisy Image (c) Median Filter

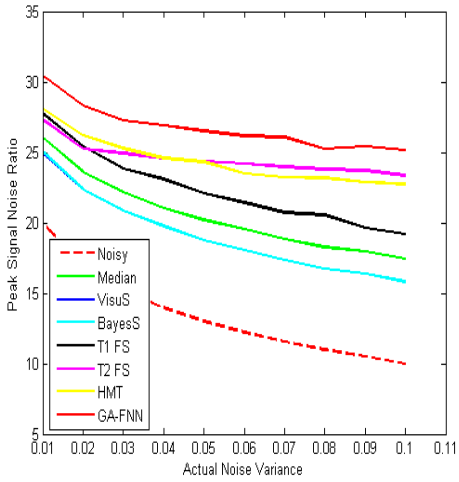
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

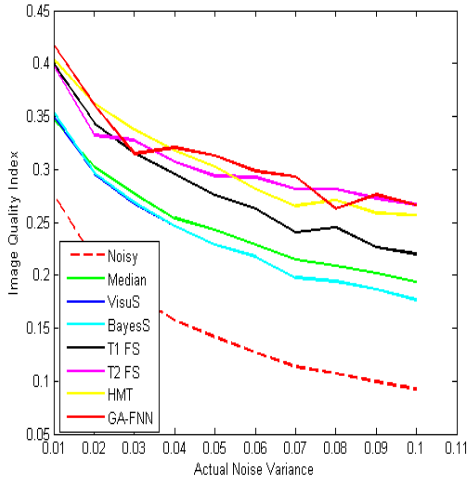
Figure 5.6: Results of "house" with Gaussian noise variance 0.05. From the point of human vision, median, VisuShrink and BayesShrink filters do not remove noise completely. T1 FShrink causes some artifacts. T2 FShrink and GA-FNN outperforms other algorithms, but T2 FShrink smooths the edges more.



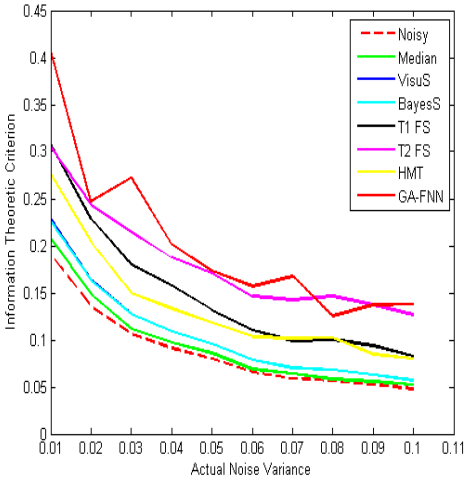
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.7: Performance of removing pure Gaussian noise (image "house"). GA-FNN performs the best among those algorithms in terms of MSE, PSNR, IQI, and ITC at most noise levels; T2 FShrink outperforms other algorithms in terms of MSE, PSNR, IQI, and ITC at higher noise levels, but slightly worse than HMT at lower noise levels in terms of MSE, PSNR, and IQI.

Table 5.7: Results of Image "house" with Gaussian noise variance 0.01

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.010108	19.9536	0.2762	0.1911
Median	0.002474	26.0659	0.3473	0.2077
VisuS	0.003186	24.9676	0.3497	0.2289
BayesS	0.003145	25.0243	0.3537	0.2257
HMT	0.001556	28.0804	0.4040	0.2757
T1 FShrink	0.001666	27.7837	0.3991	0.3077
T2 FShrink	0.001844	27.3433	0.3985	0.3173
GA-FNN	0.000900	30.4574	0.4173	0.4075

Table 5.8: Results of Image "house" with Gaussian noise variance 0.05

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05004	13.0065	0.1417	0.07974
Median	0.00955	20.2002	0.2430	0.08610
VisuS	0.01338	18.7339	0.2289	0.09594
BayesS	0.01338	18.7341	0.2290	0.09586
HMT	0.003701	24.3170	0.2926	0.1183
T1 FShrink	0.006211	22.0687	0.2754	0.1322
T2 FShrink	0.003534	2.4518	0.3021	0.1826
GA-FNN	0.002237	2.6504	0.3126	0.1734

Table 5.9: Results of Image "house" with Gaussian noise variance 0.1

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.1003	9.98783	0.09261	0.04784
Median	0.0180	17.4358	0.1936	0.05252
VisuS	0.0263	15.7970	0.1769	0.05706
BayesS	0.0263	15.7932	0.1769	0.05695
HMT	0.00532	22.7395	0.2568	0.07964
T1 FShrink	0.01206	19.1867	0.2197	0.08285
T2 FShrink	0.004522	23.4464	0.2640	0.1294
GA-FNN	0.003061	25.1415	0.2662	0.1378

In above Table 5.7, 5.8, and 5.9, the proposed algorithms and the best two results in terms of MSE, PSNR, IQI and ITC are underlined.

4. **Results of Image "lake"**

   From the filtered results shown in Figure 5.8, it can be observed that median filter, VisuShrink and BayesShrink preserve the edges in "lake" but do not remove noise completely, T1 FShrink and HMT do better job than the former three algorithms. However, there are also some artifacts in the filtered image of T1 FShrink algorithm. GA-FNN algorithm performs the best, but T2 FShrink seems to smooth the edges in the image a little more than HMT does. In the filtered edge intensity profile of pure Gaussian noise model, Figure 5.14, all observations can be clearly noticed.

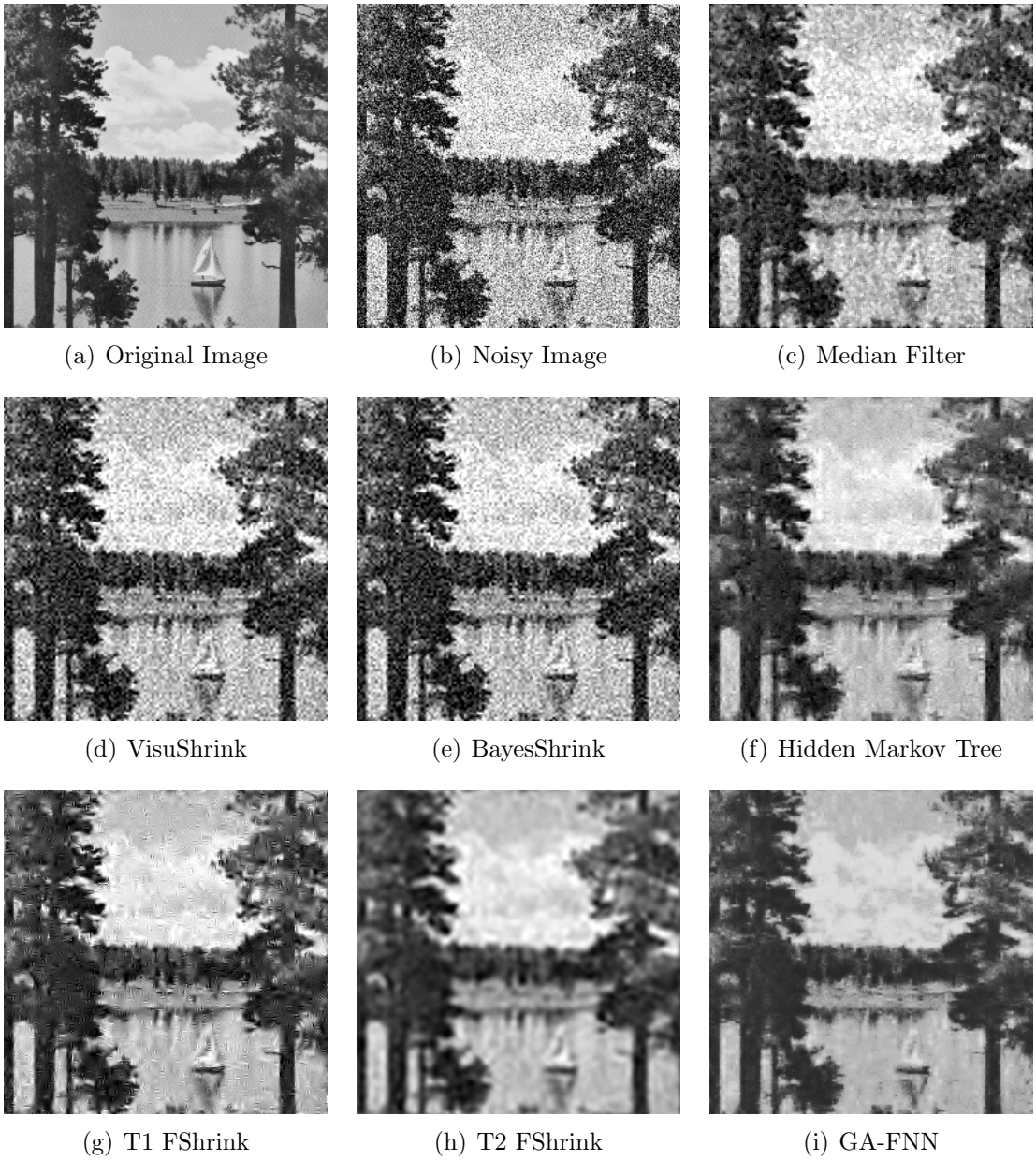
   In addition, from the completely image quality assessment in Figure 5.9 and specific mathematical analysis results at the noise variances:0.01, 0.05, 0.1 in Table 5.10, 5.11, and 5.12, the following observations can be attained: GA-FNN performs the best in terms of MSE, PSNR, and IQI at different noise levels; T2 FShrink outperforms other algorithms in terms of ITC at any noise levels; T2 FShrink performs better at higher noise levels than other algorithms except GA-FNN.

   In summary, T2 FShrink and GA-FNN algorithms perform better than any other algorithms in the natural images with straight lines and complex natural scene in terms of MSE, PSNR, IQI, and ITC.

5. **Results of Image "pepper"**

   From filtered results in Figure 5.10, it can be observed that GA-FNN and T2 FShrink algorithms perform the better than other algorithms, and there are also some artifacts in the filtered image of T1 FShrink algorithm. In the intuitive edge intensity profile of filtered images, see Figure 5.14, it can be perceived that median filter, VisuShrink and BayesShrink preserve the edges in "pepper" but do not remove noise completely. T1 FShrink and HMT do better job than the former three algorithms. T2 FShrink smooths the edges in the image a little more than T1 FShrink and HMT do. GA-FNN outperforms the best.

   Furthermore, from the completely image quality assessment in Figure 5.11 and the mathematical analysis results at selected noise variances: 0.01, 0.05, 0.1 in Table 5.13, 5.14, and 5.15, the following observations can be attained: GA-FNN performs the best among those algorithms in terms of PSNR at any noise levels; T2 FShrink outperforms other algorithms in terms of ITC at any noise levels; and T2 FShrink performs better at higher noise levels than other algorithms except GA-FNN.

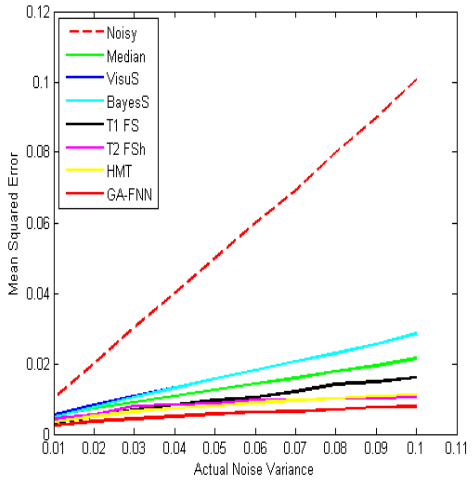


(a) Original Image (b) Noisy Image (c) Median Filter

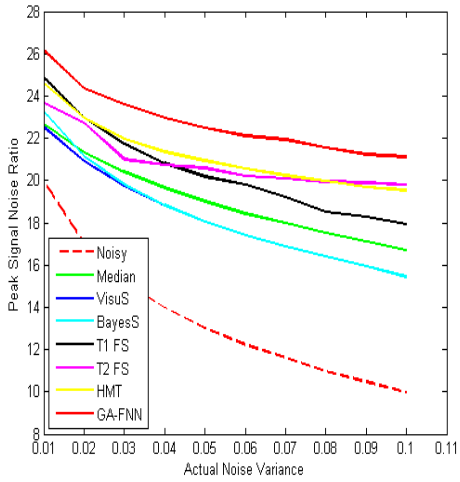
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

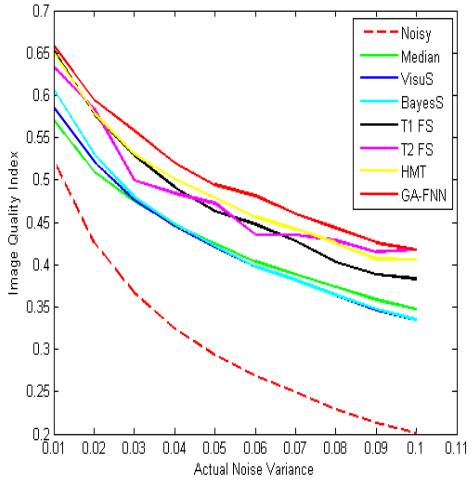
Figure 5.8: Results of "lake" with Gaussian noise variance 0.05. From the point of human vision, median, VisuShrink and BayesShrink filters do not remove noise completely. T1 FShrink causes some artifacts. T2 FShrink and GA-FNN outperforms other algorithms, but T2 FShrink smooths the edges more.



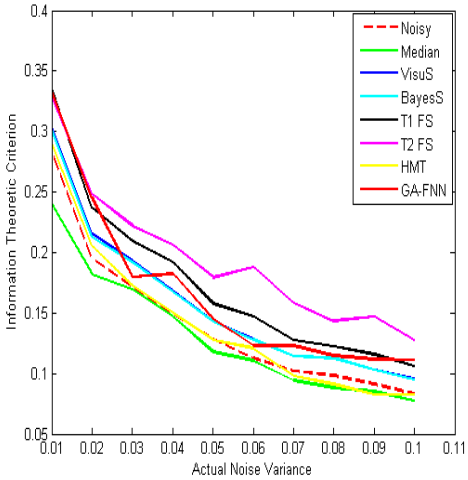
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.9: Performance of removing pure Gaussian noise (image "lake"). It can be observed that GA-FNN performs the best in terms of MSE, PSNR, and IQI at different noise level; T2 FShrink outperforms other algorithms in terms of ITC at any noise level; T2 FShrink performs better at higher noise level than other algorithms except GA-FNN.

Table 5.10: Results of Image "lake" with Gaussian noise variance 0.01

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.010061	19.9734	0.5234	0.2834
Median	0.005413	22.6653	0.5704	0.2401
VisuS	0.005577	22.5363	0.5855	0.3042
BayesS	0.004713	23.2666	0.6069	0.3014
HMT	0.003471	24.5958	0.6496	0.2896
T1 FShrink	0.003253	24.8770	0.6527	0.3352
T2 FShrink	0.004285	23.6801	0.6337	0.3287
GA-FNN	0.002417	2.6167	0.6581	0.3322

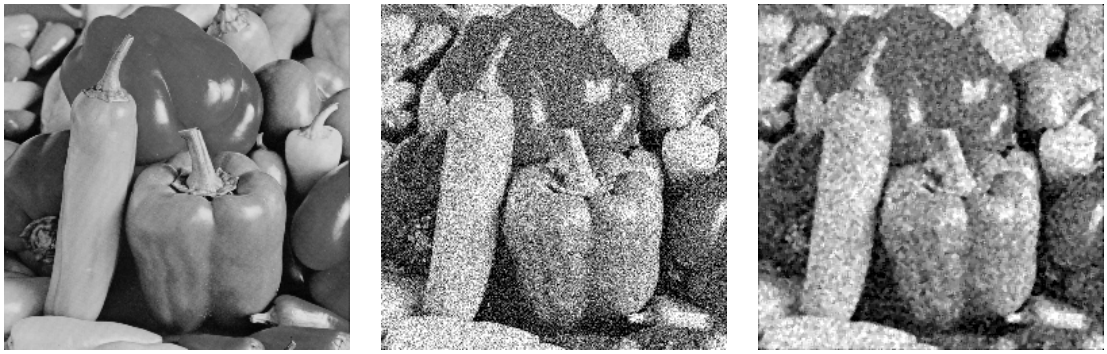
Table 5.11: Results of Image "lake" with Gaussian noise variance 0.05

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.04989	13.0192	0.2930	0.1287
Median	0.01254	19.0173	0.4255	0.1178
VisuS	0.01574	18.0230	0.4201	0.1435
BayesS	0.01568	18.0456	0.4215	0.1433
HMT	0.00805	20.9434	0.4789	0.1276
T1 FShrink	0.00962	20.1668	0.4632	0.1578
T2 FShrink	0.008684	20.6129	0.4684	0.1793
GA-FNN	0.005646	22.4829	0.4941	0.1448

Table 5.12: Results of Image "lake" with Gaussian noise variance 0.1

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.1007	9.96899	0.2014	0.0832
Median	0.0214	16.6934	0.3471	0.0776
VisuS	0.0285	15.4469	0.3344	0.0953
BayesS	0.0286	15.4425	0.3349	0.0950
HMT	0.0111	19.5368	0.4056	0.0825
T1 FShrink	0.0161	17.9293	0.3833	0.1058
T2 FShrink	0.0104	19.8400	0.4105	0.1288
GA-FNN	0.0077	21.1235	0.4172	0.1113

In Table 5.10, 5.11, and 5.12, the proposed algorithms and two best results are underlined in terms of MSE, PSNR, IQI, and ITC.

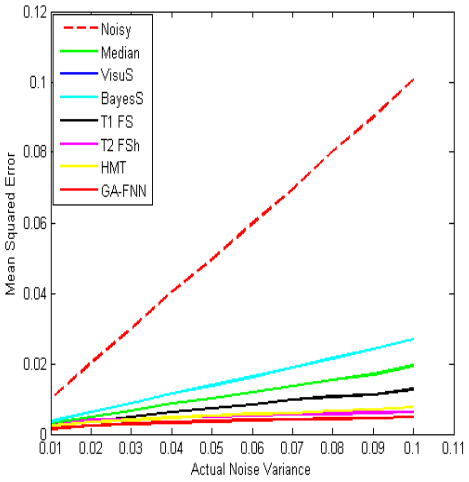


(a) Original Image (b) Noisy Image (c) Median Filter

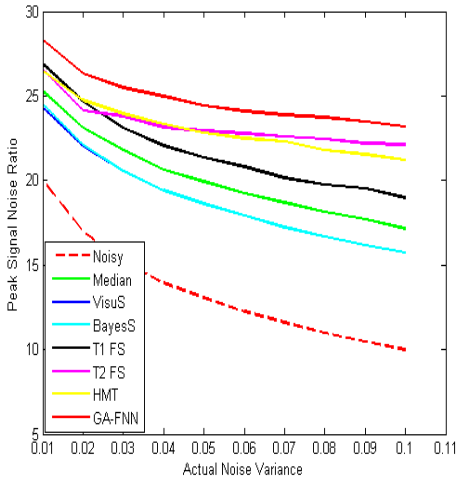
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

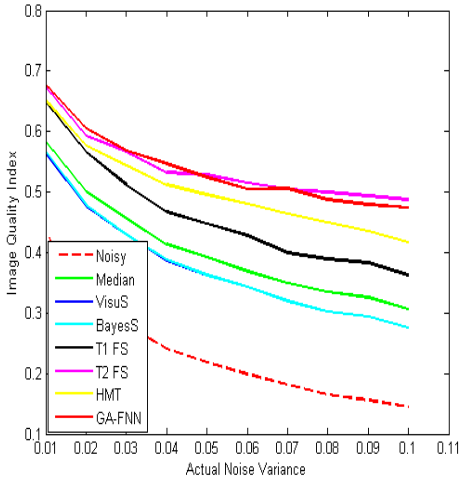
Figure 5.10: Results of "pepper" with Gaussian noise variance 0.05. From the point of human vision, median, VisuShrink and BayesShrink filters do not remove noise completely. T1 FShrink causes some artifacts. T2 FShrink and GA-FNN outperforms other algorithms, but T2 FShrink smooths the edges more.



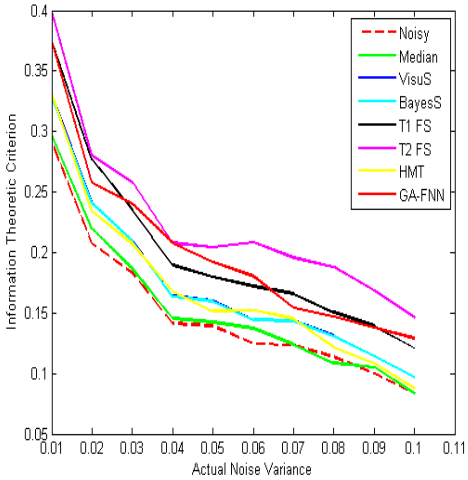
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.11: Performance of removing pure Gaussian noise (image "pepper"). It can be observed that GA-FNN performs the best among those algorithms in terms of PSNR at any noise level; T2 FShrink outperforms other algorithms in terms of ITC at any noise level; and T2 FShrink performs better at higher noise level than other algorithms except GA-FNN.

Table 5.13: Results of Image "pepper" with Gaussian noise variance 0.01

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.009899	20.0443	0.4323	0.2913
Median	0.002972	25.2701	0.5831	0.2966
VisuS	0.003730	24.2831	0.5628	0.3316
BayesS	0.003537	24.5138	0.5676	0.3290
HMT	0.002230	26.5173	0.6516	0.3315
T1 FShrink	0.00205	26.8906	0.6485	0.3745
T2 FShrink	0.002202	26.5715	0.6729	0.3990
GA-FNN	0.001477	28.3074	0.6771	0.3747

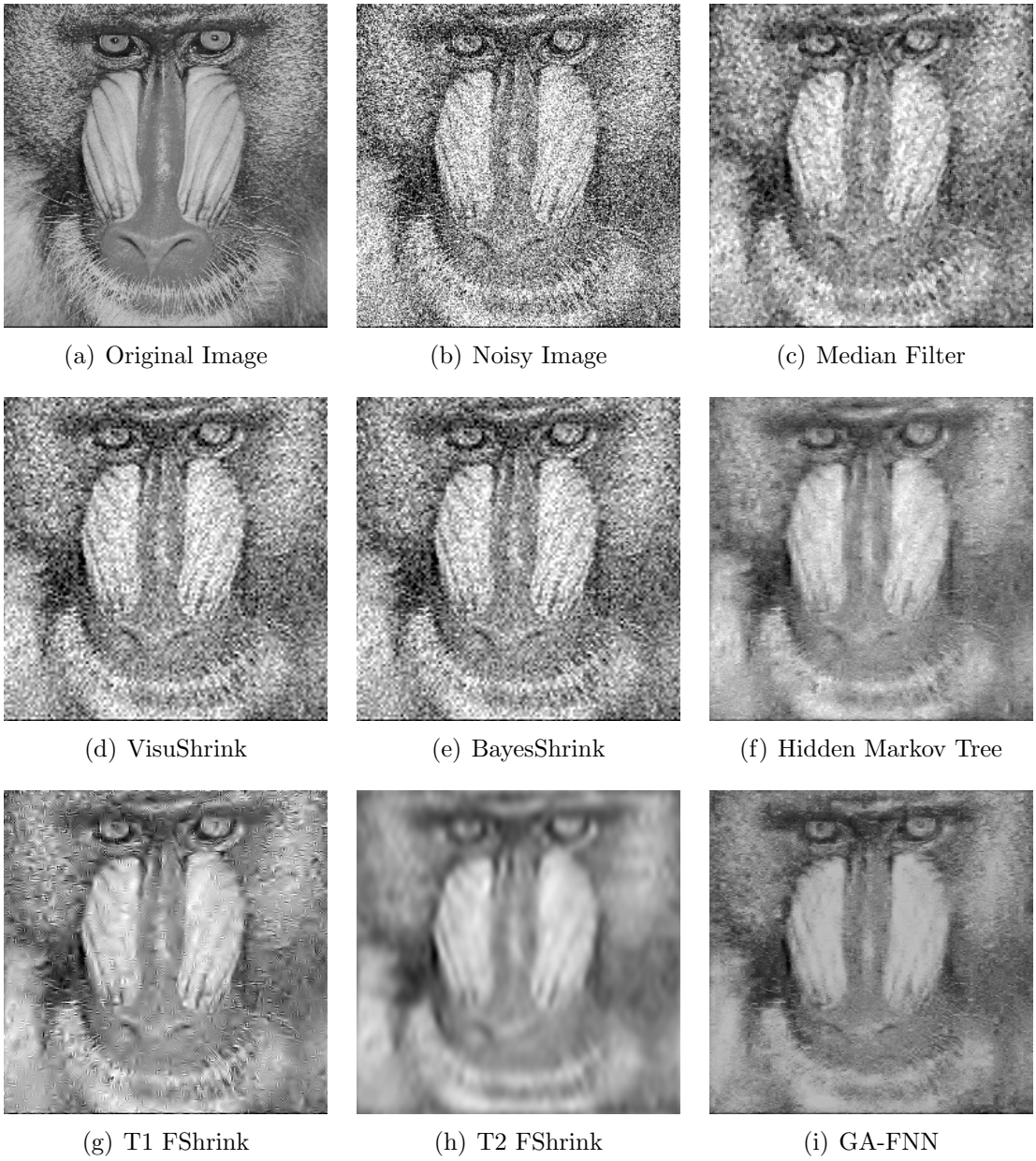
Table 5.14: Results of Image "pepper" with Gaussian noise variance 0.05

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.04960	13.0451	0.2178	0.1393
Median	0.01016	19.9296	0.3917	0.1423
VisuS	0.01375	18.6157	0.3625	0.1600
BayesS	0.01376	18.6140	0.3628	0.1597
HMT	0.00522	22.8248	0.4952	0.1516
T1 FShrink	0.00729	21.3731	0.4475	0.1799
T2 FShrink	0.00501	23.0030	0.5339	0.1987
GA-FNN	0.00362	24.4098	0.5238	0.1920

Table 5.15: Results of Image "pepper" with Gaussian noise variance 0.1

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.10059	9.97428	0.1448	0.08402
Median	0.01933	17.1375	0.3057	0.08380
VisuS	0.02686	15.7095	0.2755	0.09753
BayesS	0.02686	15.7088	0.2756	0.09752
HMT	0.00756	21.2157	0.4165	0.08781
T1 FShrink	0.01264	18.9834	0.3624	0.12111
T2 FShrink	0.006144	22.1156	0.4869	0.14654
GA-FNN	0.004810	23.1786	0.4743	0.12913

In Table 5.13, 5.14, and 5.15, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.

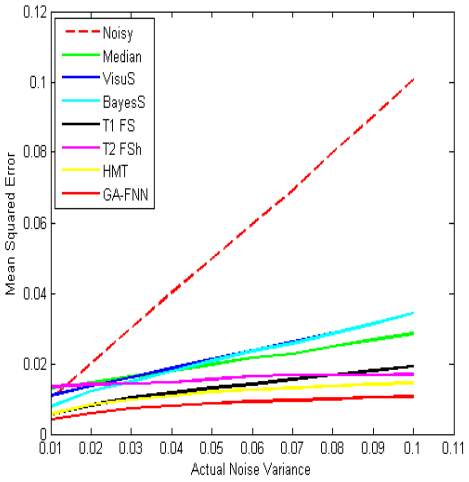


(a) Original Image (b) Noisy Image (c) Median Filter

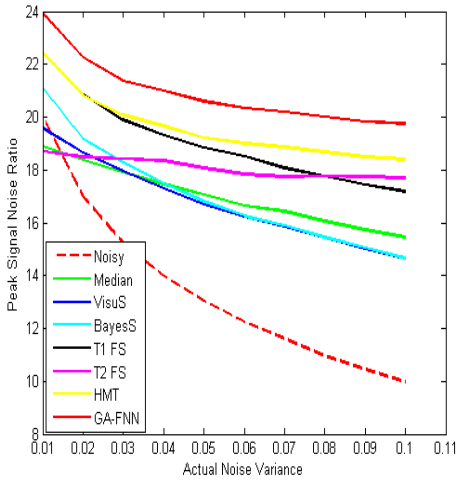
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

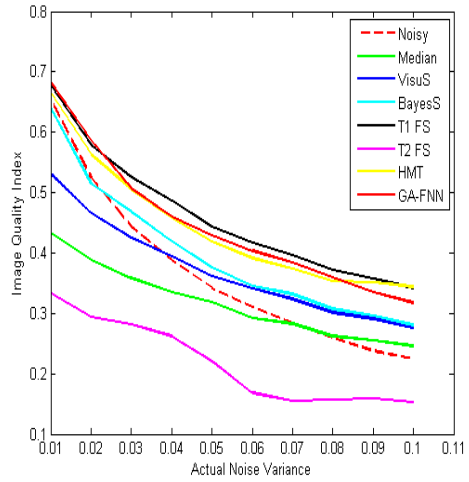
Figure 5.12: Results of "baboon" with Gaussian noise variance 0.05. From the point of human vision, median, VisuShrink and BayesShrink filters do not remove noise completely. T1 FShrink causes some artifacts. T2 FShrink and GA-FNN outperforms other algorithms, but T2 FShrink smooths the edges more.



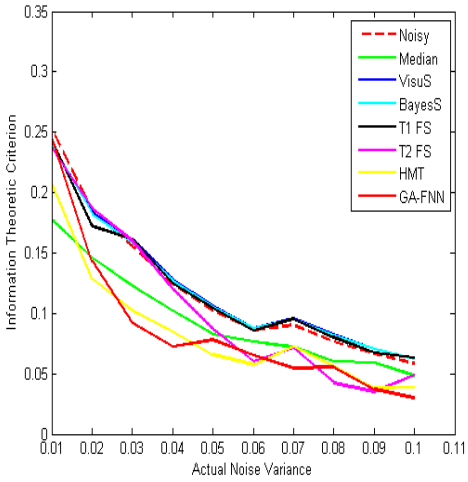
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.13: Performance of removing pure Gaussian noise (image "baboon"). It can be observed that GA-FNN performs the best among those algorithms in terms of MSE and PSNR at at any noise level; T2 FShrink does not perform well in terms of MSE, PSNR, IQI and ITC.

Table 5.16: Results of Image "baboon" with Gaussian noise variance 0.01

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.009967	20.0142	0.6541	0.2527
Median	0.012862	18.9069	0.4341	0.1775
VisuS	0.011006	19.5835	0.5310	0.2396
BayesS	0.007766	21.0983	0.6407	0.2404
HMT	0.005715	22.4297	0.6650	0.2073
T1 FShrink	0.005683	22.4540	0.6774	0.2434
T2 FShrink	0.01346	21.7085	0.3328	0.2310
GA-FNN	0.004033	23.9434	0.6829	0.2453

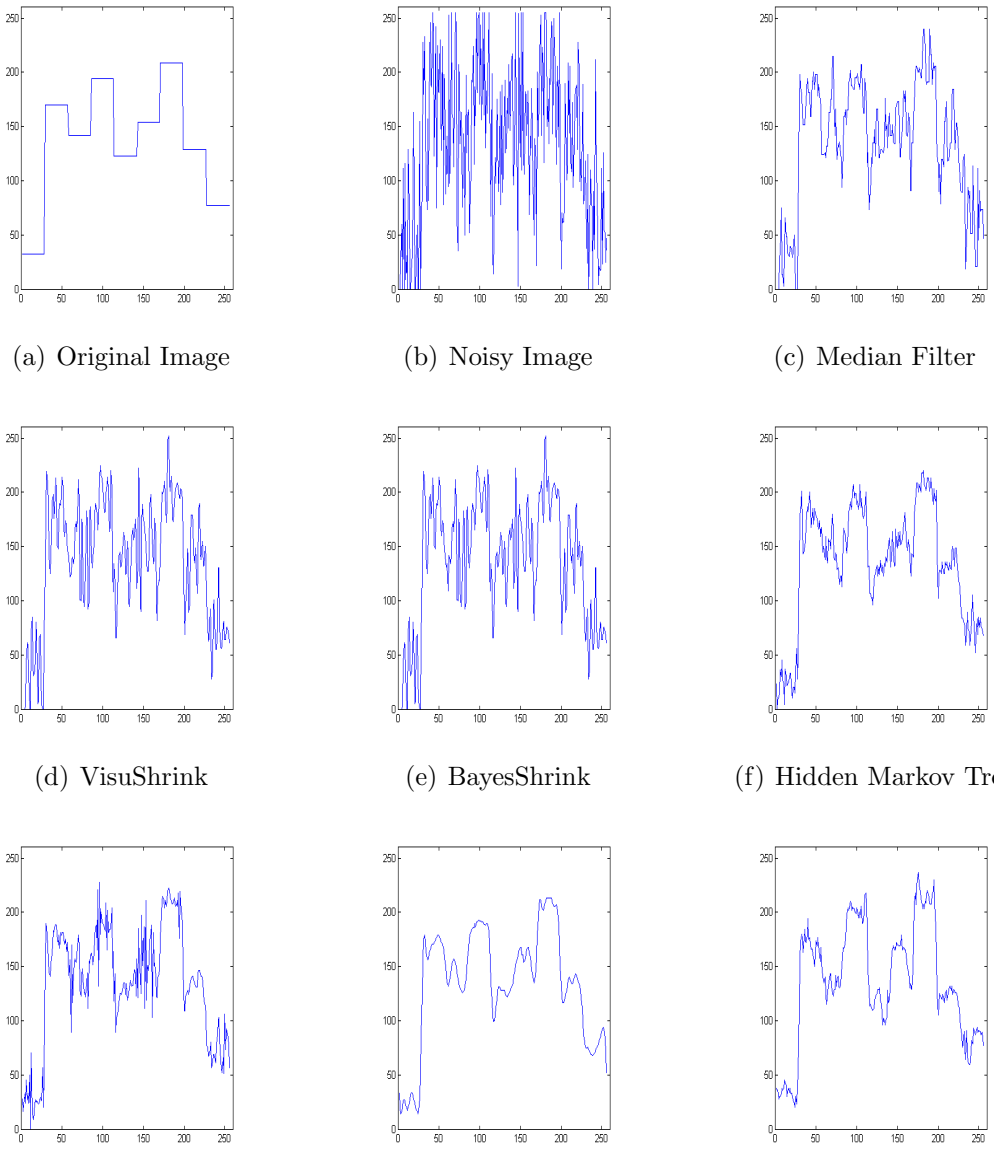
Table 5.17: Results of Image "baboon" with Gaussian noise variance 0.05

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.04973	13.0334	0.3424	0.1021
Median	0.01968	17.0594	0.3180	0.0825
VisuS	0.02139	16.6987	0.3610	0.1061
BayesS	0.02086	16.8066	0.3771	0.1056
HMT	0.01197	19.2175	0.4191	0.06587
T1 FShrink	0.01555	18.0827	0.2245	0.0861
T2 FShrink	0.01303	18.8511	0.4435	0.1046
GA-FNN	0.008735	20.5875	0.4295	0.07811

Table 5.18: Results of Image "baboon" with Gaussian noise variance 0.1

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.10069	9.97015	0.2244	0.05843
Median	0.02845	15.4588	0.2463	0.04836
VisuS	0.03439	14.6358	0.2761	0.06241
BayesS	0.03429	14.6479	0.2802	0.06229
HMT	0.01446	18.3978	0.3440	0.03881
T1 FShrink	0.01916	17.1751	0.3419	0.06302
T2 FShrink	0.01696	17.7058	0.1534	0.04290
GA-FNN	0.01056	19.7631	0.3178	0.02989

In Table 5.16, 5.17, and 5.18, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.



(a) Original Image (b) Noisy Image (c) Median Filter

(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

Figure 5.14: Filtered edge intensity profile of pure Gaussian noise model (noise variance 0.05). Comparing the proposed T2 FShrink (h) with HMT (f), the proposed T2 FShrink algorithm smooths the edges more. But the proposed GA-FNN (i) solves this problem since it combines the characters of HMT (f) and T2 FShrink (h).

6. **Results of Image "baboon"**

   From the results in Figure 5.12, it can be observed that median filter, VisuShrink and BayesShrink preserve the edges in "baboon" but do not remove noise completely. T1 FShrink and HMT perform better than the former three algorithms. However, T1 FShrink algorithm still causes some artifacts in the filtered image. T2 FShrink solves this problem using multilevel stationary DWT. But T2 FShrink blurs the edges in the image. GA-FNN outperforms the best among those algorithms. For intuitive observations, see Figure 5.14, the edge intensity profile of filtered images shows that T2 FShrink algorithm removes much noise but smooth the edges in images; GA-FNN algorithm performs better than other algorithms.

   Additionally, the full image quality assessment in Figure 5.13 and the mathematical analysis results at selected noise variances: 0.01, 0.05, 0.1 in Table 5.16, 5.17, and 5.18 indicate that GA-FNN performs the best among those algorithms in terms of MSE and PSNR at at any noise level; T2 FShrink does not perform well in terms of MSE, PSNR, IQI and ITC.

   In summary, GA-FNN, T2 FShrink and other algorithms perform worse when denoising the images with fine hair, such as "baboon" image in terms of MSE, PSNR, IQI and ITC, especially, T2 FShrink algorithms is sensitive to this kind of images. The reason is that animal hair is rather noise-like. It is hard for denoising algorithms to determine whether a pixel on the image is noisy or hairy. However, GA-FNN and T2 FShrink algorithms perform better than any other algorithms in the natural images with curve edges, such as "pepper" image.

5.2.2 Quality Assessment of Images with Mixed Noise

In this section, the performance of all algorithms is evaluated when denoising images with mixed noise models, such as Gaussian Noise and impulsive noise.

- **Test Process**

  In this experiment, Gaussian noise and impulsive noise, both with variance values ranging from 0.01 to 0.1, are added to the test images used in pure Gaussian noise model discussed in Section 5.2.1. In order to obtain a thorough evaluation of the performance, comparisons are made among the same algorithms used in Section 5.2.1.

- **Conditions of Implementation**

  All implementation conditions are the same as those in testing pure Gaussian noise model. For details, please refer to Section 5.2.1.

- **Observation and Discussion**

  1. **Results of Image "lena"**

     Subjectively, the results in Figure 5.15 show that median filter, VisuShrink and BayesShrink preserve the edges in "lena" but do not remove noise completely. T1 FShrink and HMT perform better than the former three algorithms. However, T1 FShrink algorithm causes some artifacts in the filtered image. T2 FShrink solves this problem using multilevel stationary DWT. But T2 FShrink blurs the edges in the image. GA-FNN performs the best among those algorithms. For intuitive observations, see Figure 5.27, the edge intensity profile of filtered images shows that T2 FShrink algorithm removes much noise but smooths the edges in images; GA-FNN algorithm performs better than other algorithms.

     Additionally, the full image quality assessment in Figure 5.16 and the mathematical analysis of the mixed noise model at the selected noise variances: 0.01, 0.05, 0.1 in Table 5.19, 5.20, and 5.21 indicate that T2 FShrink and GA-FNN algorithms outperform other algorithms at any noise levels in terms of MSE, PSNR, IQI and ITC.

  2. **Results of Image "man"**

     Subjectively, the results in Figure 5.17 and Figure 5.27 show that T2 FShrink and GA-FNN algorithms outperform other algorithms. T2 FShrink blurs the edges in the image. T1 FShrink algorithm still causes artifacts. In the edge intensity profile of filtered images (see Figure 5.27), the observations can be summarized: median filter, VisuShrink and BayesShrink, T1 FShrink and HMT do not remove the noise completely; T2 FShrink removes much noise but smooths the edges of the image; GA-FNN performs the best.

     Furthermore, the full image quality assessment in Figure 5.18 and the mathematical analysis of the mixed noise model at the selected noise variances: 0.01, 0.05, 0.1 in Table 5.22, 5.23, and 5.24 demonstrate that T2 FShrink and GA-FNN outperform other algorithms in terms of MSE and PSNR at diverse noise variances of the mixed noise. GA-FNN performs best in terms of IQI. However, in terms of ITC, there is no algorithm always performs the best at any noise levels.

     In summary, T2 FShrink and GA-FNN perform better than other algorithms when denoising the mixed noise model, especially when images have a stationary background. For example, image "lena".

(a) Original Image (b) Noisy Image (c) Median Filter

(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

Figure 5.15: Results of "lena" with Gaussian and Impulsive Noise (Both noise variance are 0.05). T2 FShrink and GA-FNN outperforms other algorithms. But T2 FShrink blurs the edges in the image.

Table 5.19: Results of Image "lena" with Gaussian and Impulsive Noise (Both variances are 0.01.)

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01236	19.0790	0.4019	0.1653
Median	0.00354	24.5142	0.5430	0.1773
VisuS	0.00469	23.2872	0.5095	0.1870
BayesS	0.00481	23.1798	0.5115	0.1790
HMT	0.00350	24.5559	0.5549	0.1791
T1 FShrink	0.00401	23.9732	0.5595	0.2095
T2 FShrink	0.00316	25.0025	0.5972	0.2377
GA-FNN	0.00109	29.6154	0.6128	0.2377

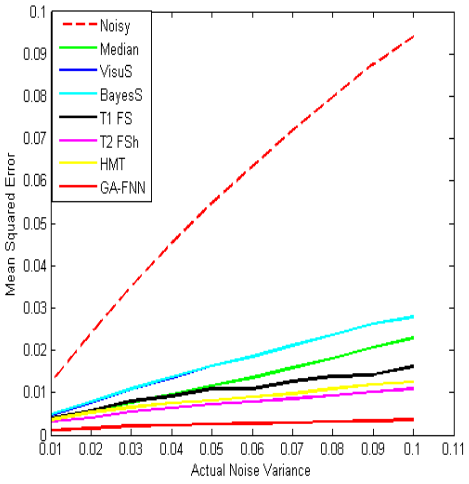
Table 5.20: Results of Image "lena" with Gaussian and Impulsive Noise (Both variances are 0.05.)

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05475	12.6162	0.1935	0.05813
Median	0.01157	19.3685	0.3612	0.06468
VisuS	0.01625	17.8918	0.3181	0.06719
BayesS	0.01635	17.8639	0.3185	0.06652
HMT	0.00816	20.8806	0.3863	0.05528
T1 FShrink	0.01085	19.6446	0.3627	0.07699
T2 FShrink	0.00721	21.4227	0.4226	0.09367
GA-FNN	0.00252	25.9800	0.4468	0.1065

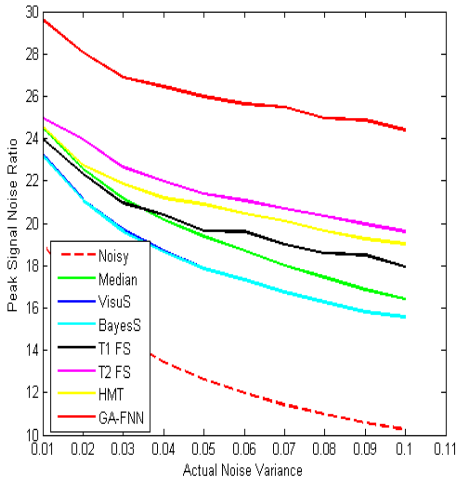
Table 5.21: Results of Image "lena" with Gaussian and Impulsive Noise (Both variance are 0.1.)

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09421	10.2591	0.1226	0.02190
Median	0.02281	16.4178	0.2654	0.02629
VisuS	0.02780	15.5599	0.2328	0.02509
BayesS	0.02780	15.5594	0.2329	0.02509
HMT	0.01252	19.0248	0.2992	0.01771
T1 FShrink	0.01609	17.9355	0.2756	0.02583
T2 FShrink	0.01070	19.7047	0.3544	0.04416
GA-FNN	0.00363	24.4000	0.3558	0.04198

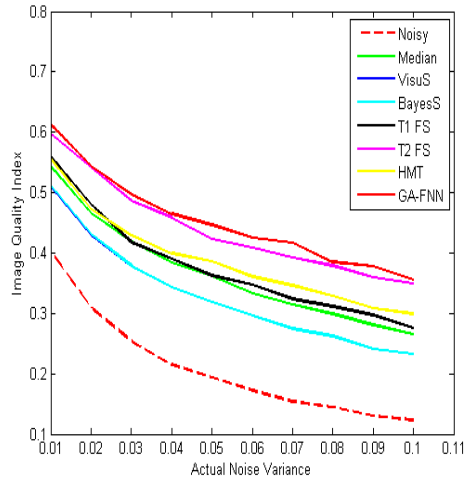
In Table 5.19, 5.20, and 5.21, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.



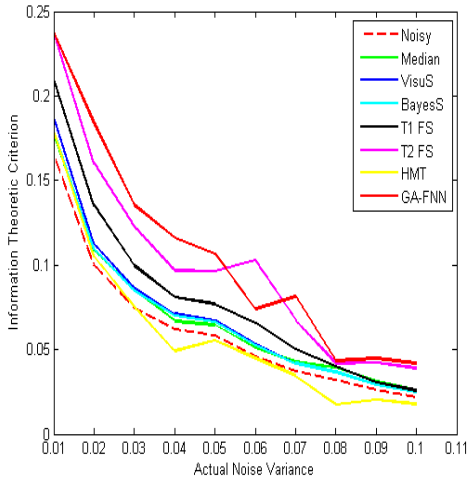
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

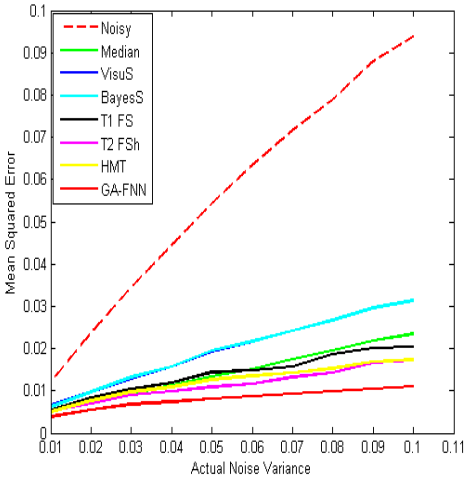
Figure 5.16: Performance of removing mixed noise model (image "lena"). T2 FShrink and GA-FNN algorithms outperform other algorithms at any noise level in terms of MSE, PSNR, IQI and ITC.

(a) Original Image (b) Noisy Image (c) Median Filter

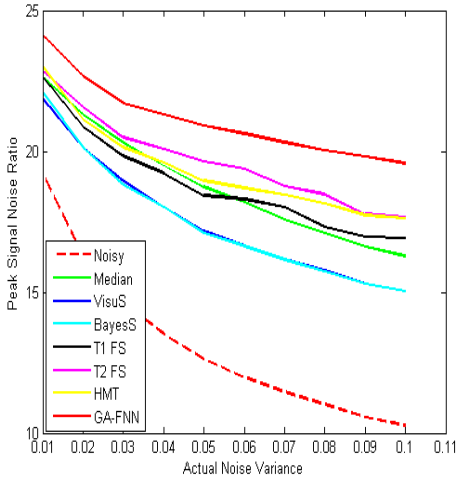
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

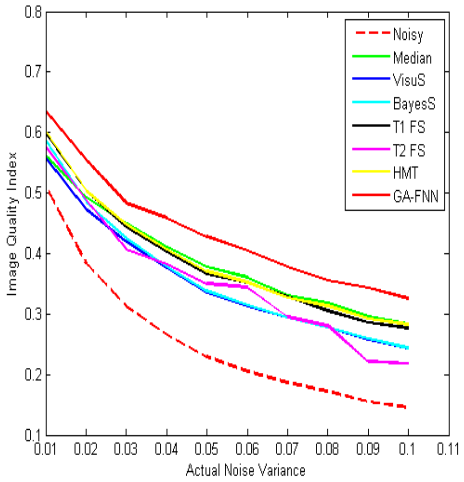
Figure 5.17: Results of "man" with Gaussian and Impulsive Noise (Both noise variance are 0.05). T2 FShrink and GA-FNN algorithms outperform other algorithms. T2 FShrink blurs the edges in the image.



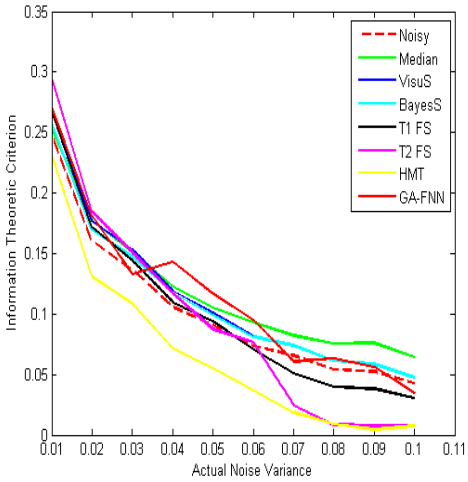
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.18: Performance of removing mixed noise model (image "man"). T2 FShrink and GA-FNN outperform other algorithms in terms of MSE and PSNR at diverse noise variances of the mixed noise. GA-FNN performs best in terms of IQI. However, in terms of ITC, there is no algorithm always performs the best at any noise levels.

Table 5.22: Results of Image "man" with Gaussian and Impulsive Noise (Both variance are 0.01)

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01201	1.9205	0.5116	0.2485
Median	0.00547	2.2623	0.5616	0.2534
VisuS	0.00652	2.1858	0.5576	0.2686
BayesS	0.00618	2.2091	0.5868	0.2565
HMT	0.00499	23.0176	0.6014	0.2314
T1 FShrink	0.00545	22.6367	0.59881	0.2672
T2 FShrink	0.00510	22.9224	0.5830	0.3009
GA-FNN	0.00386	24.1315	0.6358	0.2710

Table 5.23: Results of Image "man" with Gaussian and Impulsive Noise (Both variance are 0.05.)

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05432	12.6506	0.2295	0.0910
Median	0.01337	18.7394	0.3773	0.1052
VisuS	0.01922	17.1621	0.3348	0.1009
BayesS	0.01947	17.1059	0.3380	0.0988
HMT	0.01270	18.9625	0.3607	0.0550
T1 FShrink	0.01436	18.4271	0.3660	0.0937
T2 FShrink	0.01073	19.6942	0.3706	0.1064
GA-FNN	0.00853	20.6922	0.4160	0.1113

Table 5.24: Results of Image "man" with Gaussian and Impulsive Noise (Both variance are 0.1.)

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09380	10.2778	0.1459	0.04277
Median	0.02348	16.2920	0.2838	0.06396
VisuS	0.03135	15.0376	0.2436	0.04740
BayesS	0.03135	15.0376	0.2437	0.04741
HMT	0.01726	17.6304	0.2251	0.00731
T1 FShrink	0.02034	16.9165	0.2768	0.03049
T2 FShrink	0.01700	1.7697	0.2828	0.01410
GA-FNN	0.01083	19.6524	0.3287	0.03445

In Table 5.22, 5.23, and 5.24, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.

3. **Results of Image "house"**

   In Figure 5.19, it can be observed that T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more. T1 FShrink algorithm still causes artifacts. In the edge intensity profile of filtered images (see Figure 5.27), the observations can be summarized: median filter, VisuShrink and BayesShrink, T1 FShrink and HMT do not remove the noise completely; T2 FShrink removes much noise but smooths the edges of the image; GA-FNN performs the best.

   In addition, the thorough image quality assessment in Figure 5.20 and the mathematical analysis of the mixed noise model at the selected noise variances: 0.01, 0.05, 0.1 in Table 5.25, 5.26, and 5.27 show that T2 FShrink and GA-FNN algorithms outperform other algorithms in terms of MSE, PSNR, IQI, and ITC at most noise levels.

4. **Results of Image "lake"**

   Subjectively, the results in Figure 5.21 show that T1 FShrink algorithm still causes artifacts. T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more. The observations from the edge intensity profile of filtered images, see Figure 5.27 can be perceived: median filter, VisuShrink and BayesShrink, T1 FShrink and HMT do not remove the noise completely; T2 FShrink removes much noise but smooths the edges of the image; GA-FNN performs the best.

   Additionally, the complete image quality assessment in Figure 5.22 and the mathematical analysis of the mixed noise model at the selected noise variances: 0.01, 0.05, 0.1 in Table 5.28, 5.29, and 5.30 indicate that T2 FShrink and GA-FNN algorithms outperform other algorithms in terms of MSE and PSNR and perform better than other algorithms when the mixed noise level is lower in terms of IQI and ITC.

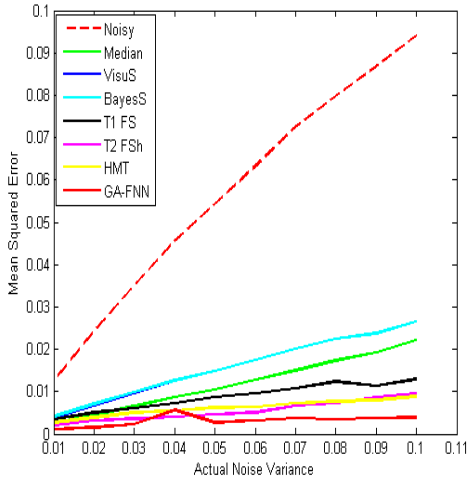
   In summary, through the analysis from different points of view, in terms of four assessment tools, T2 FShrink and GA-FNN algorithms perform better than other algorithms when they remove the mixed noise from the natural scene images with simple background, such as "house" image.

(a) Original Image (b) Noisy Image (c) Median Filter

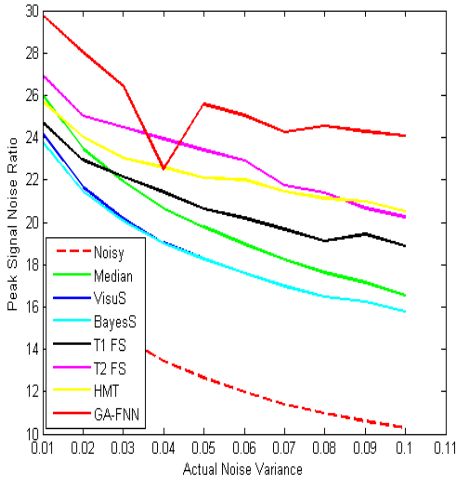
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

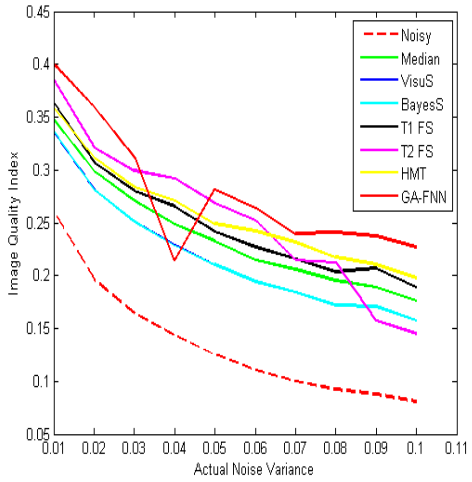
Figure 5.19: Results of "house" with Gaussian and Impulsive Noise (Both noise variance are 0.05). T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more.



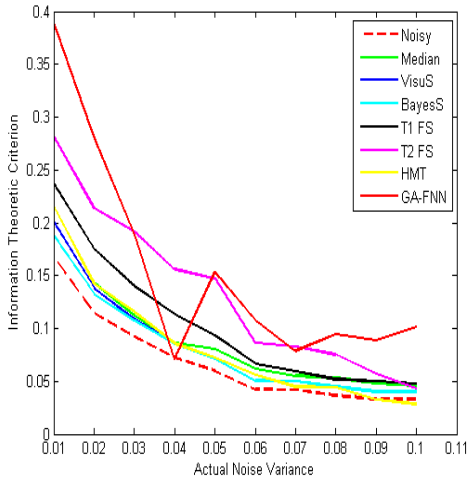
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.20: Performance of removing mixed noise model (image "house"). T2 FShrink and GA-FNN algorithms outperform other algorithms in terms of MSE, PSNR, IQI, and ITC at most noise levels.

Table 5.25: Results of Image "house" with Gaussian and Impulsive Noise (Both variance are 0.01.)

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01254	19.0184	0.2610	0.1672
Median	0.00251	25.9954	0.3478	0.2157
VisuS	0.00382	24.1797	0.3354	0.2008
BayesS	0.00419	23.7811	0.3370	0.1879
HMT	0.00269	25.7081	0.3586	0.2157
T1 FShrink	0.00338	24.7152	0.3636	0.2370
T2 FShrink	0.00202	26.9369	0.3854	0.2819
GA-FNN	0.00106	29.7557	0.4000	0.3895

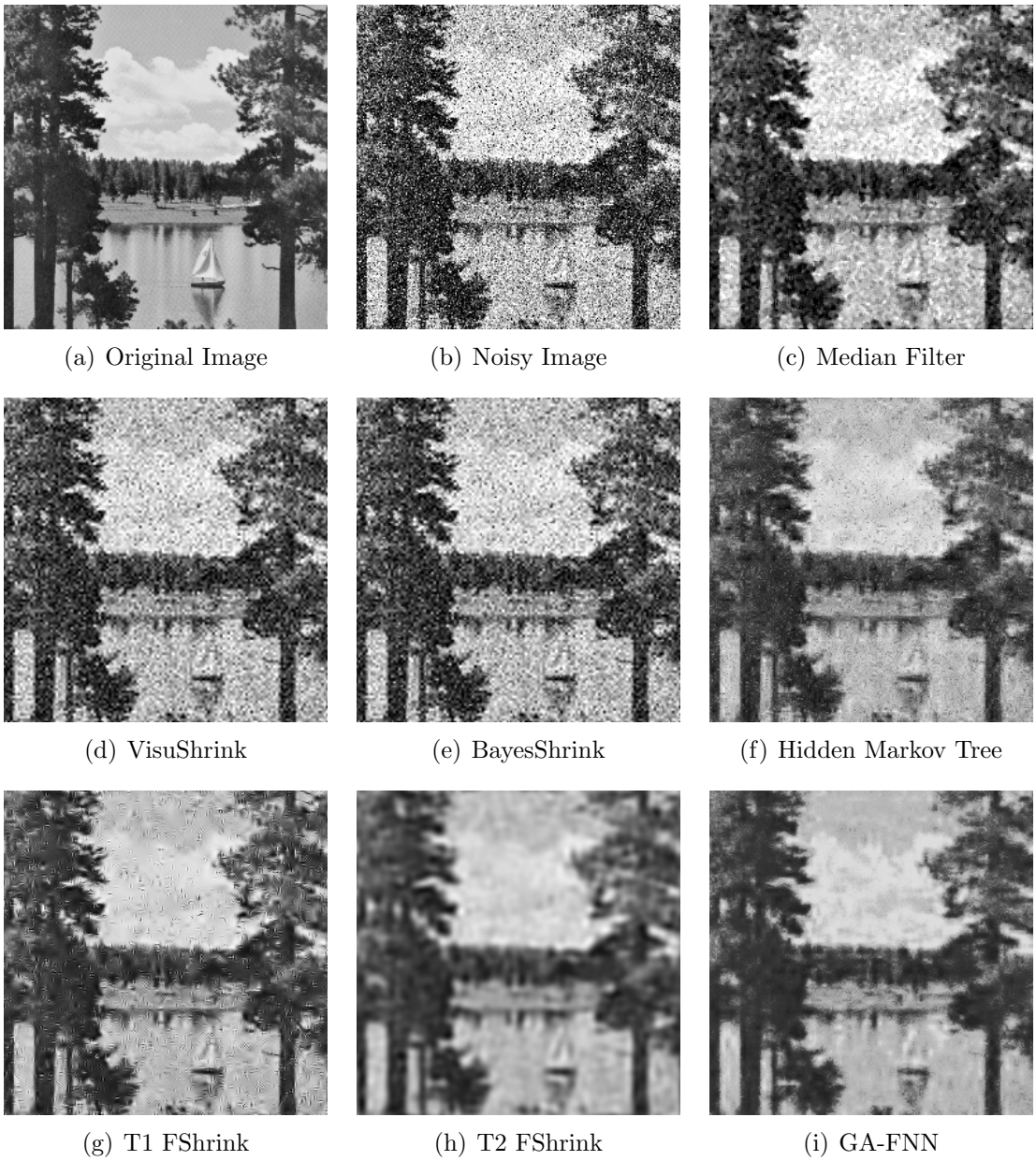
Table 5.26: Results of Image "house" with Gaussian and Impulsive Noise (Both variance are 0.05.)

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05441	12.6433	0.1254	0.06002
Median	0.01050	19.7891	0.2329	0.08042
VisuS	0.01481	18.2937	0.2104	0.07191
BayesS	0.01488	18.2735	0.2104	0.07171
HMT	0.00619	22.0831	0.2490	0.07228
T1 FShrink	0.00867	20.6217	0.2417	0.09379
T2 FShrink	0.00451	23.4607	0.2733	0.1485
GA-FNN	0.00277	25.5760	0.2813	0.1534

Table 5.27: Results of Image "house" with Gaussian and Impulsive Noise (Both variance are 0.1)

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09409	10.2647	0.0808	0.03276
Median	0.02215	16.5451	0.1763	0.04547
VisuS	0.02653	15.7610	0.1576	0.03981
BayesS	0.02654	15.7604	0.1576	0.03978
HMT	0.00889	20.2470	0.1979	0.02854
T1 FShrink	0.01300	18.8602	0.1895	0.04789
T2 FShrink	0.00945	20.5127	0.1469	0.04867
GA-FNN	0.00391	24.0778	0.2270	0.10171

In Table 5.25, 5.26, and 5.27, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.

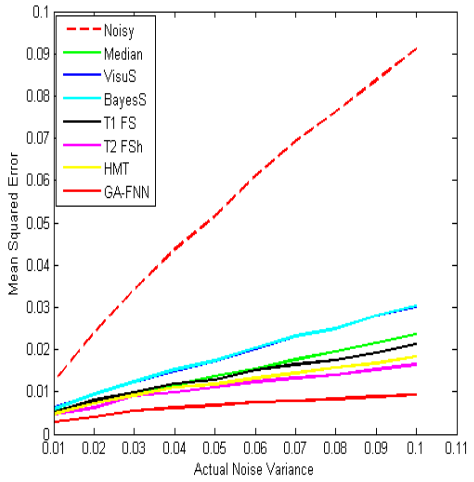


(a) Original Image (b) Noisy Image (c) Median Filter

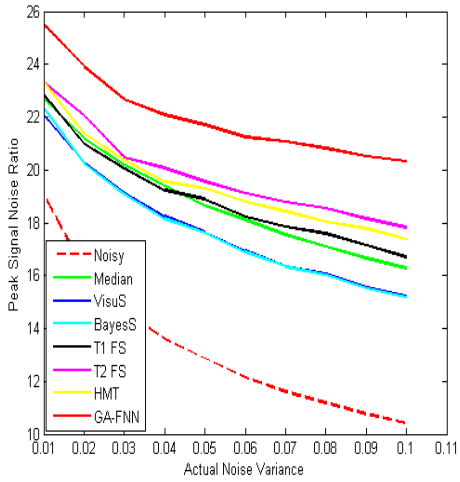
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

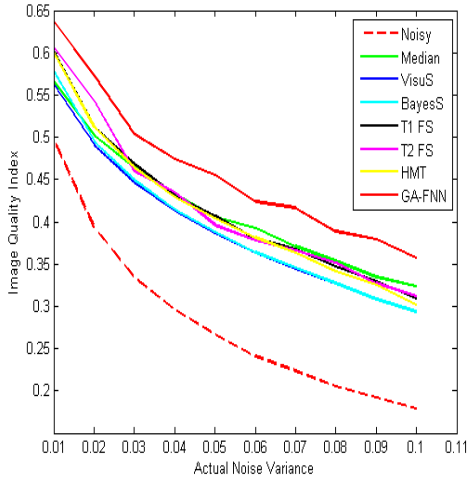
Figure 5.21: Results of "lake" with Gaussian and Impulsive Noise (Both noise variance are 0.05). T2 FShrink and GA-FNN algorithms outperform other algorithms. T2 FShrink blurs the edges in the image.



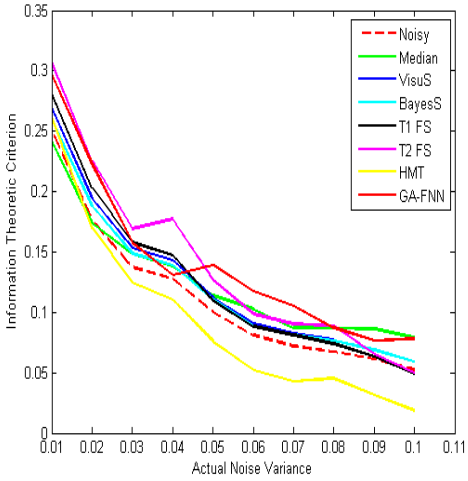
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.22: Performance of removing mixed noise model (image "lake"). T2 FShrink and GA-FNN algorithms outperform other algorithms in terms of MSE and PSNR and perform better than other algorithms when the mixed noise level is lower in terms of IQI and ITC.

Table 5.28: Results of Image "lake" with Gaussian and Impulsive Noise (Both variance are 0.01.)

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01236	19.0785	0.4972	0.2512
Median	0.00540	22.6743	0.5661	0.2412
VisuS	0.00621	22.0681	0.5627	0.2692
BayesS	0.00583	22.3445	0.5786	0.2602
HMT	0.00466	23.3143	0.6001	0.2616
T1 FShrink	0.00518	22.8548	0.6020	0.2808
T2 FShrink	0.00459	23.3790	0.6095	0.3072
GA-FNN	0.00282	25.5039	0.6372	0.2967

Table 5.29: Results of Image "lake" with Gaussian and Impulsive Noise (Both variance are 0.05.)

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05156	12.8771	0.2667	0.0998
Median	0.01366	18.6464	0.4057	0.1137
VisuS	0.01719	17.6476	0.3863	0.1120
BayesS	0.01736	17.6045	0.3880	0.1102
HMT	0.01175	19.2986	0.4048	0.0761
T1 FShrink	0.01287	18.9056	0.4031	0.1095
T2 FShrink	0.01098	19.5941	0.4078	0.1317
GA-FNN	0.00676	21.6988	0.4553	0.1387

Table 5.30: Results of Image "lake" with Gaussian and Impulsive Noise (Both variance are 0.1.)

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09113	10.4034	0.1791	0.05237
Median	0.02360	16.2712	0.3235	0.07906
VisuS	0.03024	15.1942	0.2937	0.05918
BayesS	0.03032	15.1829	0.2939	0.05899
HMT	0.01832	17.3716	0.3020	0.01856
T1 FShrink	0.02132	16.7129	0.3092	0.04870
T2 FShrink	0.01613	17.9225	0.3238	0.06195
GA-FNN	0.00926	20.3330	0.3572	0.07822

In Table 5.28, 5.29, and 5.30, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.

5. **Results of Image "pepper"**

   In Figure 5.23, it can be observed that T1 FShrink algorithm still causes artifacts. T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more. The observations from the edge intensity profile of filtered images, see Figure 5.27, can be perceived: median filter, VisuShrink and BayesShrink, T1 FShrink and HMT do not remove the noise completely; T2 FShrink removes much noise but smooths the edges of the image; GA-FNN performs the best.

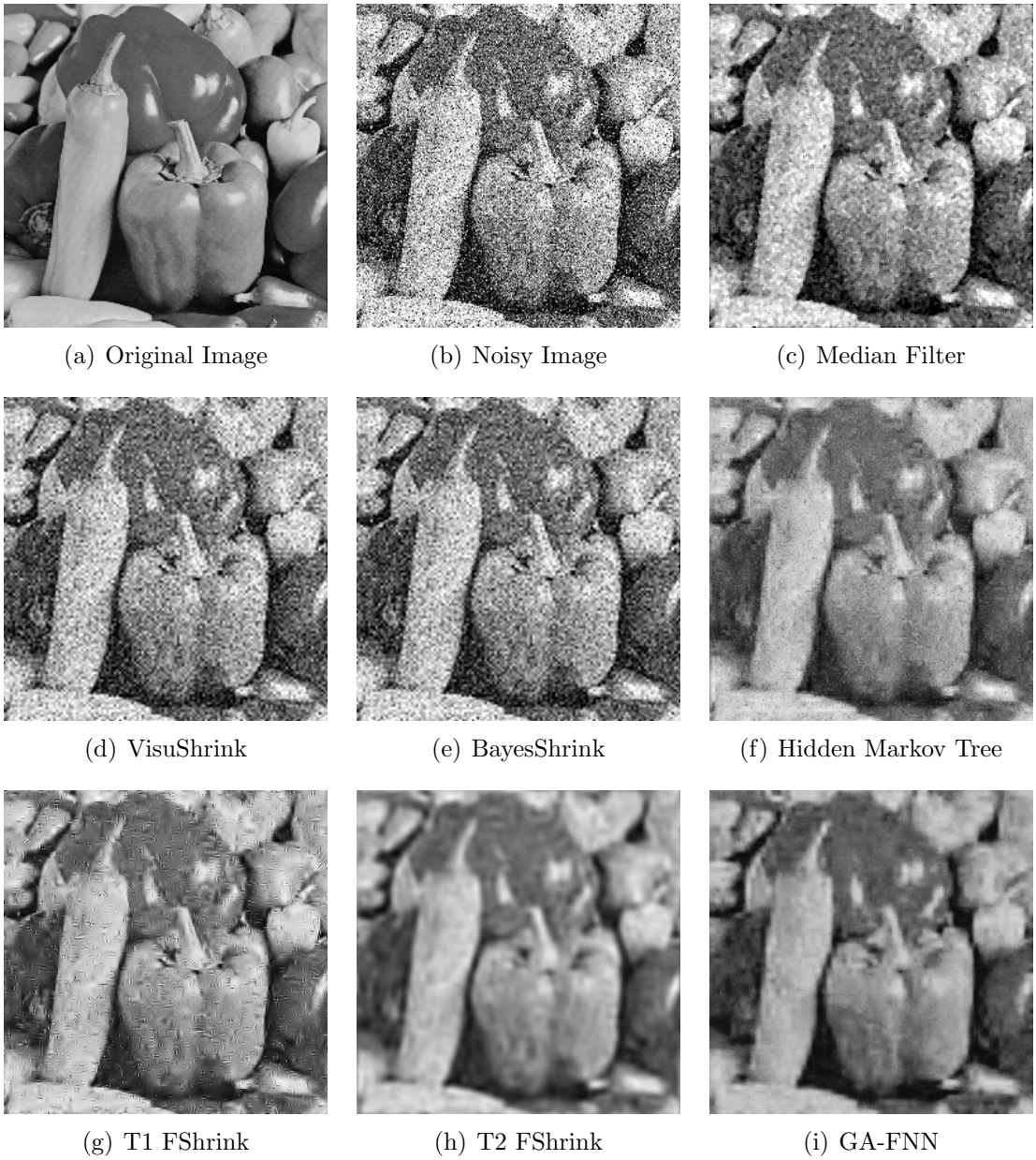
   Furthermore, the full image quality assessment in Figure 5.24 and the mathematical analysis of the mixed noise model at the selected noise variances: 0.01, 0.05, 0.1 in Table 5.31, 5.32, and 5.33 show that T2 FShrink and GA-FNN algorithms outperform other algorithms at any noise levels in terms of MSE, PSNR, and IQI, and also perform better at low noise levels in terms of ITC.

6. **Results of Image "baboon"**

   In Figure 5.25, it can be noticed that T1 FShrink algorithm still causes artifacts. T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more. From the edge intensity profile of filtered images, see Figure 5.27, the observations can be perceived: median filter, VisuShrink and BayesShrink, T1 FShrink and HMT do not remove the noise completely; T2 FShrink removes much noise but smooths the edges of the image; GA-FNN performs the best.

   In addition, the thorough image quality assessment in Figure 5.26 and the mathematical analysis of the mixed noise model at the selected noise variances: 0.01, 0.05, 0.1 in Table 5.34, 5.35, and 5.36 indicate that GA-FNN outperforms other algorithms in terms of MSE and PSNR. T2 FShrink performs better than other algorithms at low noise levels in terms of ITC.

   In summary, no algorithm performs better than others in terms of four assessment tools when denoising the images with fine hair, such as "baboon" image. T2 FShrink algorithm is sensitive to this kind of images. The reason is that animal hair is rather noise-like. It is hard for denoising algorithms to determine whether a pixel on the image is noisy or hairy. However, GA-FNN and T2 FShrink algorithms perform better than any other algorithms in the natural images with curve edges, such as "pepper" image.

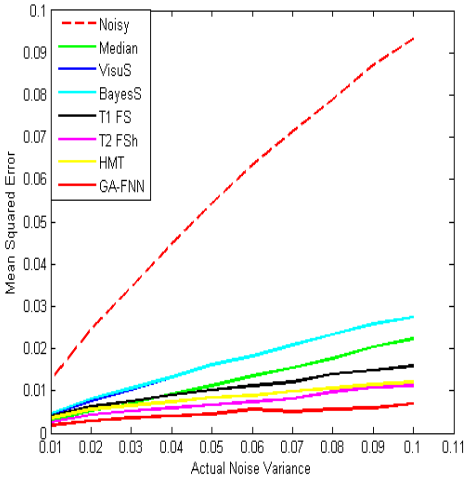


(a) Original Image (b) Noisy Image (c) Median Filter

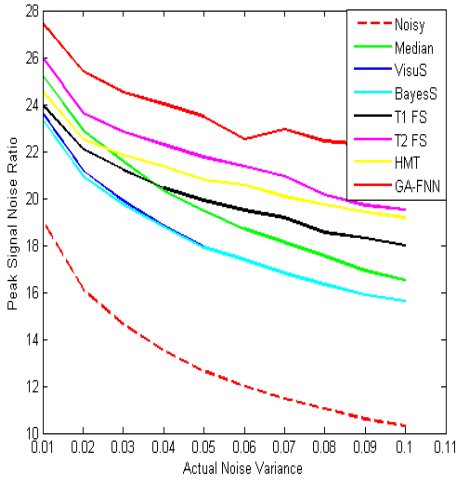
(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

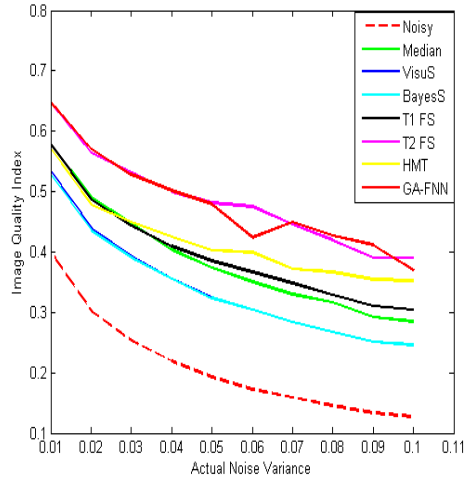
Figure 5.23: Results of "pepper" with Gaussian and Impulsive Noise (Both noise variance are 0.05). T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more.



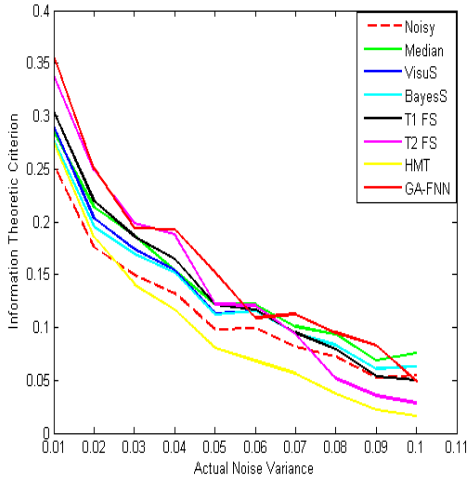
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.24: Performance of removing mixed noise model(image "pepper"). T2 FShrink and GA-FNN algorithms outperform other algorithms at any noise levels in terms of MSE, PSNR, and IQI, and also perform better at low noise levels in terms of ITC.

Table 5.31: Results of Image "pepper" with Gaussian and Impulsive Noise (Both variance are 0.01.)

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01235	19.0840	0.4060	0.2607
Median	0.00299	25.2429	0.5885	0.2895
VisuS	0.00433	23.6325	0.5393	0.2979
BayesS	0.00465	23.3228	0.5338	0.2846
HMT	0.00346	24.6077	0.5778	0.2800
T1 FShrink	0.00395	24.0386	0.5846	0.3130
T2 FShrink	0.00247	2.6070	0.6497	0.3523
GA-FNN	0.00179	2.7448	0.6477	0.3568

Table 5.32: Results of Image "pepper" with Gaussian and Impulsive Noise (Both variance are 0.05.)

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05439	12.6450	0.1921	0.1120
Median	0.01127	19.4814	0.3797	0.1406
VisuS	0.01577	18.0226	0.3297	0.1315
BayesS	0.01585	17.9997	0.3295	0.1306
HMT	0.00835	20.7836	0.4059	0.0878
T1 FShrink	0.01036	19.8441	0.3834	0.1329
T2 FShrink	0.00658	21.8148	0.4852	0.1628
GA-FNN	0.00447	23.4941	0.4789	0.1522

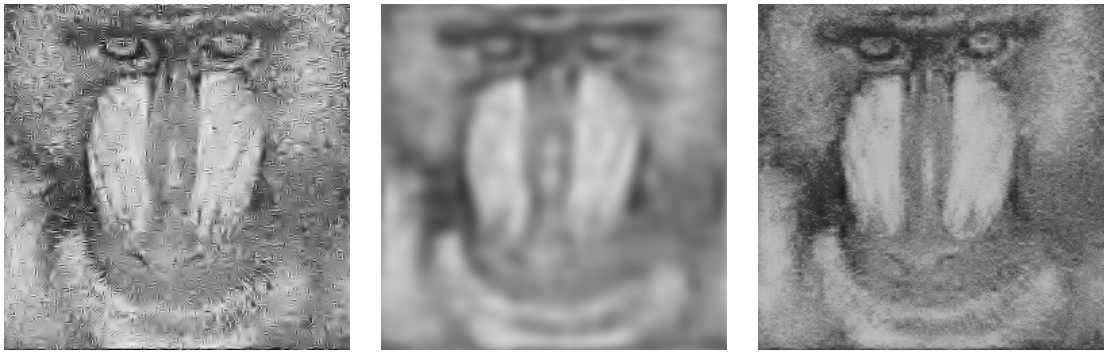
Table 5.33: Results of Image "pepper" with Gaussian and Impulsive Noise (Both variance are 0.1.)

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09380	10.2777	0.1234	0.05393
Median	0.02271	16.4374	0.2815	0.07856
VisuS	0.02757	15.5951	0.2423	0.06308
BayesS	0.02758	15.5933	0.2423	0.06305
HMT	0.01264	18.9821	0.3397	0.02029
T1 FShrink	0.01655	17.8122	0.2936	0.05375
T2 FShrink	0.01258	19.0005	0.3554	0.02705
GA-FNN	0.00688	21.6212	0.3707	0.04852

In Table 5.31, 5.32, and 5.33, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.

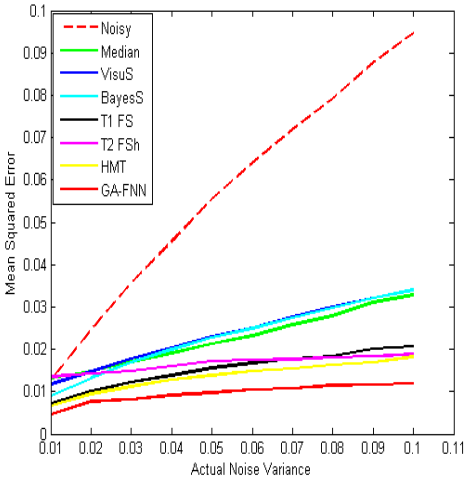
(a) Original Image (b) Noisy Image (c) Median Filter

(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

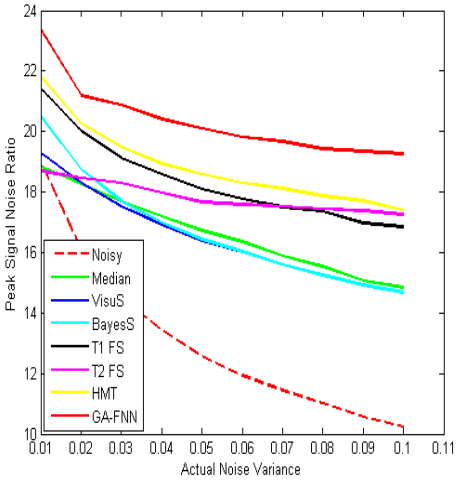


(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

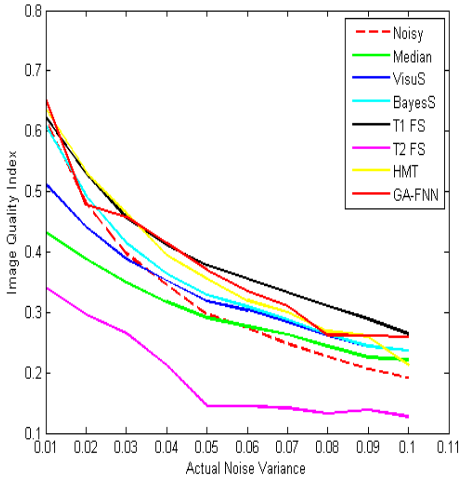
Figure 5.25: Results of "baboon" with Gaussian and Impulsive Noise (Both noise variance are 0.05). T2 FShrink and GA-FNN algorithms outperform other algorithms in removing the mixed noise, but T2 FShrink blurs the edges in the image more.



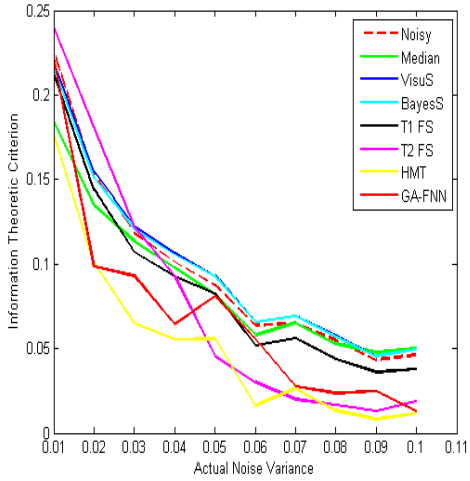
(a) Result Assessed by MSE



(b) Result Assessed by PSNR



(c) Result Assessed by IQI



(d) Result Assessed by ITC

Figure 5.26: Performance of removing mixed noise model (image "baboon"). GA-FNN outperforms other algorithms in terms of MSE and PSNR. T2 FShrink performs better than other algorithms at low noise levels in terms of ITC.

Table 5.34: Results of Image "baboon" with Gaussian and Impulsive Noise (Both variance are 0.01.)

$var = 0.01$	MSE	$PSNR$	IQI	ITC
Noisy	0.01255	19.0138	0.6187	0.2259
Median	0.01304	18.8472	0.4328	0.1844
VisuS	0.01173	19.3088	0.5118	0.2191
BayesS	0.00892	20.4982	0.6087	0.2165
HMT	0.00659	21.8072	0.6366	0.1770
T1 FShrink	0.01350	18.6967	0.3420	0.2398
T2 FShrink	0.00719	21.4298	0.6234	0.2125
GA-FNN	0.00458	23.3826	0.6520	0.2218

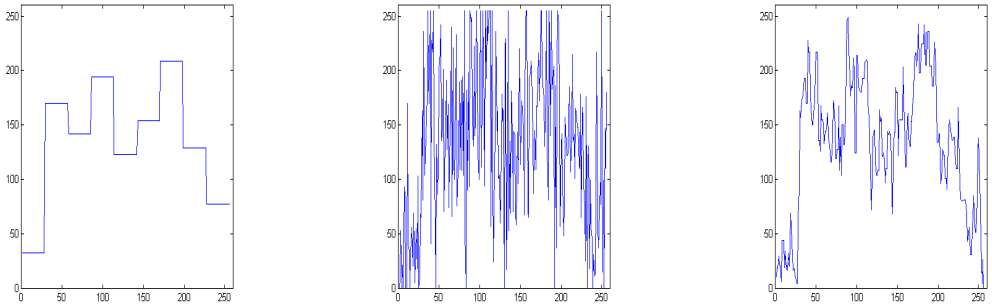
Table 5.35: Results of Image "baboon" with Gaussian and Impulsive Noise (Both variance are 0.05.)

$var = 0.05$	MSE	$PSNR$	IQI	ITC
Noisy	0.05542	12.5637	0.2973	0.08739
Median	0.02126	16.7238	0.2911	0.08189
VisuS	0.02304	16.3752	0.3180	0.09320
BayesS	0.02273	16.4339	0.3288	0.09259
HMT	0.01388	18.5768	0.3554	0.05580
T1 FShrink	0.01714	17.6598	0.1447	0.04562
T2 FShrink	0.01551	18.0940	0.3779	0.08256
GA-FNN	0.00979	20.0924	0.3706	0.08093

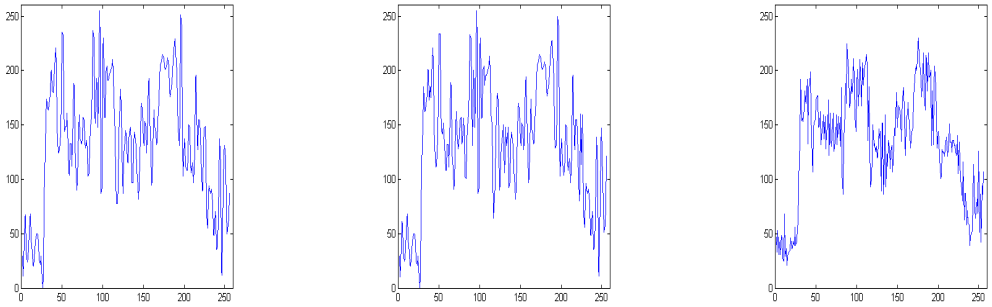
Table 5.36: Results of Image "baboon" with Gaussian and Impulsive Noise (Both variance are 0.1.)

$var = 0.1$	MSE	$PSNR$	IQI	ITC
Noisy	0.09468	10.2372	0.1916	0.04619
Median	0.03283	14.8372	0.2217	0.05036
VisuS	0.03403	14.6816	0.2364	0.04950
BayesS	0.03402	14.6821	0.2365	0.04951
HMT	0.01818	17.4023	0.2116	0.01168
T1 FShrink	0.02070	16.8399	0.2646	0.03780
T2 FShrink	0.01884	17.2482	0.1274	0.01866
GA-FNN	0.01184	19.2652	0.2595	0.01318

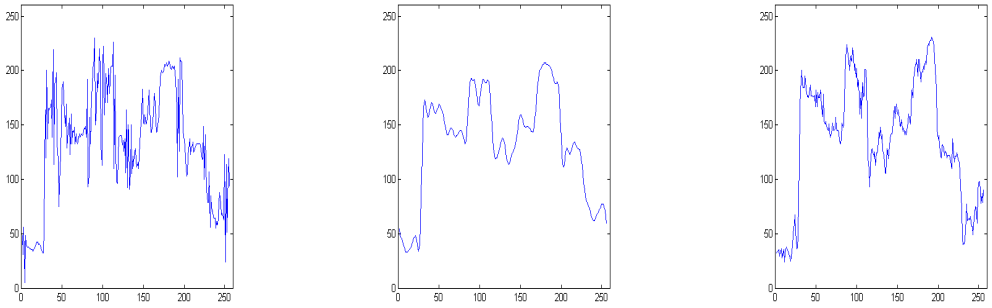
In Table 5.34, 5.35, and 5.36, the proposed algorithms and two best results, in terms of MSE, PSNR, IQI, and ITC, are underlined.



(a) Original Image (b) Noisy Image (c) Median Filter



(d) VisuShrink (e) BayesShrink (f) Hidden Markov Tree

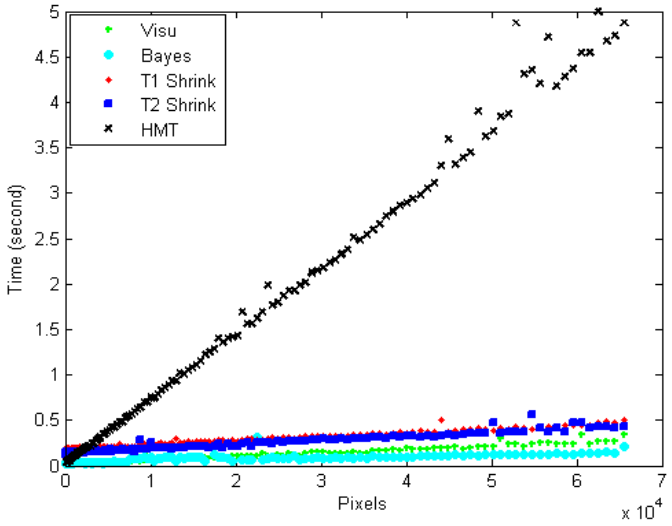


(g) T1 FShrink (h) T2 FShrink (i) GA-FNN

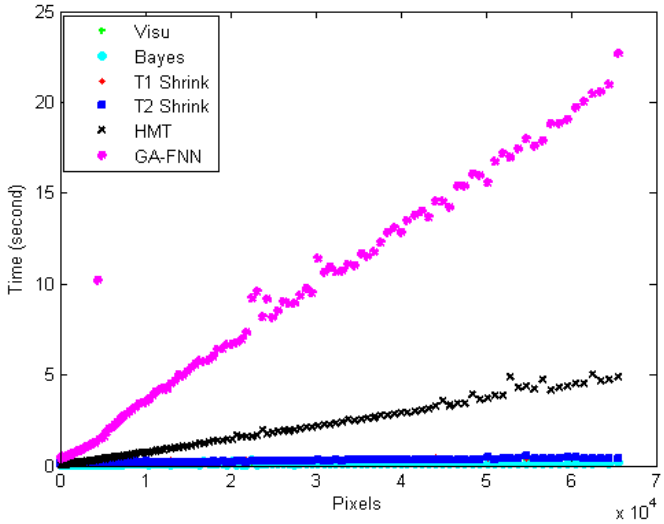
Figure 5.27: Filtered edge intensity profile of mixed additive noise model, variance both are 0.05. Comparing the proposed T2 FShrink (h) with HMT (f), the proposed algorithm T2 FShrink smooths the edges a little more. But the proposed GA-FNN (i) solves this problem since it combines the advantages of HMT (f) and T2 FShrink (h), which are preserving edges and removing noise respectively.

5.2.3 Computational Cost

To fully and completely evaluate the performance of the denoising algorithms, it is not adequate if only the quality of filtered images is assessed. Computing expense is also an important criteria, especially in a real time system. The following figures show the comparison of the computational cost between the proposed algorithms and other algorithms. Since some algorithms are partially coded in C language, such as T1 FShrink, the comparison may not be complete. However it still can provide a rough impression of the computing cost of all algorithms.



(a) Computational cost without GA-FNN



(b) Computational cost with GA-FNN

Figure 5.28: Comparison of computational cost. It can be observed that the proposed GA-FNN has the highest computational expense.

5.2.4 Universality

From the conclusion obtained in Section 5.2.1 and 5.2.2, the main advantages and drawbacks of T2 FShrink and GA-FNN are list as follows,

- GA-FNN algorithm performs best in terms of MSE, PSNR, IQI and ITC in most cases but it has the highest computing expense.
- T2 FShrink algorithm is much faster than GA-FNN and also faster than hidden Markov Tree based algorithm, see Figure 5.28(a) and 5.28(b), however, it smooths some details in the image, especially when denoising the images with fine contents, such as animal hair.

In summary, to design a denoising algorithm which always performs the best no matter dealing with what images is a dream of researchers. However, from the results of the performance of the proposed algorithms and the comparisons with other algorithms, it can be concluded that no algorithm is perfect in practice so far. Therefore, it is practical in the real world, if attaining a better result is critical, GA-FNN algorithm would be a good choice; if the denoising speed is of greater concern, then T2 FShrink algorithm will be a better choice.

5.3 Summary

In Section 5.2.1 and 5.2.2, the performance of denoising the pure Gaussian noise and the mixed noise of all algorithms are demonstrated and comparisons are made among them. In Section 5.2.3, the computational cost of the proposed algorithms is investigated. In Section 5.2.4, the universality of the proposed algorithms is evaluated. Based on the results and discussion, conclusions are made in this section.

5.3.1 Performance of Removing Pure Gaussian Noise

Based on the comparison and discussion in Section 5.2.1, the following conclusions can be made,

1. When dealing with human, natural scene images with simple background, the proposed GA-FNN and T2 FShrink algorithms outperform HMT, T1 FShrink, BayesShirnk, VisuShrink, and median filters in terms of MSE, PSNR, IQI and ITC, especially at higher noise levels.
2. The proposed GA-FNN always performs the best in term of all assessment tools while has the highest computing cost 5.28(b). T2 FShrink algorithm performs better at higher noise level with the fast computing speed. However it smooths the edges more. These conclusions can be verified by the intuitive graph of edge intensity profile in Figure 5.14.

3. No algorithm performs better than the others in terms of all assessment tools when denoising an image with fine contents, such as fine hair, see Figure 5.13. T2 FShrink algorithm is more sensitive to this type of images.

5.3.2 Performance for Removing Mixed Additive Noise

Based on observations in Section 5.2.2, the performance of proposed algorithms in dealing with mixed additive noise can be summarized here.

1. When dealing with human, natural scene images with simple background, the proposed GA-FNN and T2 FShrink algorithms outperform HMT, T1 FuzzyShrink, BayesShrink and VisuShrink, in terms of MSE, PSNR, IQI, and ITC. However the proposed T2 FShrink algorithm performs slightly worse than hidden markov tree based algorithm when denoising an image with fine content, such as fine hair.

2. The proposed GA-FNN always performs the best in term of all assessment tools while has the highest computing cost 5.28(b). T2 FShrink algorithm performs better at lower noise level with the fast computing speed. However it smooths the edges more. These conclusions can be verified by the intuitive graph of edge intensity profile in Figure 5.27.

3. No algorithm performs better than the others in terms of all assessment tools when denoising the image with fine contents, such as fine hair, see Figure 5.26. T2 FShrink algorithm is more sensitive to this type of images.

Chapter 6

Conclusion and Future Work

In this thesis, wavelet shrinkage based image denoising using soft computing is addressed. First, a current survey of the state-of-the-art denoising algorithms and motivation of this thesis are described. Second, interpretations and illustrative explanations of wavelets are presented. Third, two new wavelet shrinkage based image denoising algorithms are proposed. Fourth, the comparisons among all algorithms are demonstrated and the performances are evaluated in terms of four assessment tools, MSE, PSNR, IQI, and ITC. In this chapter, the conclusions are summarized.

6.1 Conclusion and Discussion

As presented and summarized in Chapter 5, the proposed T2 FShrink and GA-FNN perform better than other algorithms in terms of MSE, PSNR, IQI, and ITC in most cases. This section summarizes the discussion and outlines the main conclusions.

1. **Natural Images with Human Beings**

   From the results of "lena" image in Figure 5.3 and 5.16, it can be observed that T2 FShrink and GA-FNN perform better when dealing with mixed noise model in terms of MSE, PSNR, IQI, and ITC. From the observations of "man" image in Figure 5.5 and 5.18, it still can be noticed that T2 FShrink and GA-FNN algorithms perform better when dealing with mixed noise model in terms of MSE, PSNR, IQI, and ITC. Therefore, it can be concluded that T2 FShrink and GA-FNN algorithms outperform other algorithms when denoising Natural Images with human beings, no matter what noise models.

2. **Natural Images with Straight Lines and Scenes**

   From the observations of "house" image in Figure 5.7 and 5.20, it can be seen that T2 FShrink and GA-FNN perform better when dealing with mixed noise model in terms of MSE, PSNR, IQI, and ITC. In "lake" image, see Figure 5.9 and 5.22, it can be perceived that T2 FShrink and GA-FNN perform better

when dealing with mixed noise model in terms of MSE, PSNR, IQI, and ITC. Therefore, it can be concluded that T2 FShrink and GA-FNN algorithms outperform other algorithms when denoising Natural Images with Straight Lines and Scenes, no matter what noise models.

3. **Natural Images with Curve Edges and Fine Structure**

   From the observations of "pepper" image in Figure 5.11 and 5.24, it can be seen that T2 FShrink and GA-FNN perform better when dealing with mixed noise model in terms of MSE, PSNR, IQI, and ITC, especially at lower noise levels. From the results of "baboon" image in Figure 5.13 and 5.26, it is hard to observe whether T2 FShrink and GA-FNN perform better or not when dealing with different noise models. The reason is that, when image content includes fine details such as animal hair, denoising algorithms can not determine if a region is noisy or fine detail properly. Therefore, it can be concluded that the proposed algorithms perform better in natural images with curve edges no matter what noise models. Like other algorithms, T2 FShrink and GA-FNN do not perform very well in dealing with images with fine structure, such as animal hair.

4. **Efficiency of Proposed Algorithms**

   Based on the discussion in Section 5.2.3, the proposed GA-FNN algorithm outperforms other algorithms but its computing cost is the highest. Another propose algorithm T2 FShrink works much faster than GA-FNN does. This can be viewed in Figure 5.28(a) and 5.28(b).

5. **Universality of Proposed Algorithms**

   Based on the discussion in Section 5.2.4, the proposed GA-FNN algorithm outperforms other algorithms in different images, types of noise, and noise variances. Therefore, it has the most universality among these algorithms.

Generally speaking, the proposed algorithms outperform other algorithms in terms of MSE, PSNR, IQI and ITC in most cases. Suggestions to improve the proposed algorithms to overcome their drawbacks will be discussed in Section 6.2.

6.2 Main Areas of Future Work

As discussed in Chapter 4 and 5, both T2 FShrink and GA-FNN algorithms are related to automatic thresholding process, and the threshold calculation is highly related to the noise estimation. Some future work on T2 FShrink and GA-FNN algorithms will improve their performance.

- **Noise Estimation**

As discussed in Chapter 4, the threshold is a key to wavelet shrinkage based denoising algorithms, and its calculation is based on the noise estimation. If the noise estimation is not precise, attaining a better denoising will be very difficult. In this thesis, the noise model is known, so the noise estimation method exploited in this study works well. However, if the noise model is unknown, the method of the noise estimation needs to be modified. In addition, in order to obtain a precise noise estimation at different resolution scales, the noise at subbands may also need to be concerned [76].

- **Parameter Optimization**

  Although both proposed algorithms perform better than other algorithms, their performance still can be improved by optimizing some parameters. For instance, for the T2 FShrink algorithm, a genetic algorithm can also be applied to select the optimal widths and centers of the type-2 fuzzy membership functions for achieving better denoising results.

- **Computational Cost Reduction**

  As concluded in Chapter 5, GA-FNN algorithm performs the best among all algorithms. However, this algorithm has a higher computing cost. Since the built-in fuzzy neural network is applied to GA-FNN algorithm, it shows many figures when training the network. One convenient way to reduce the computing time is to code the training process in C language. Another method to decrease the time cost is to employ Self-Generating Neural Networks (SGNNs). These neural networks are self-organizing neural networks, whose network structures and parameters do not need to be set by users, and its learning process needs no iteration. It would be worth investigating whether SGNNs are able to perform as well with reduced computational complexity.

References

[1] N. Alajlan, M. Kamel, and M. E. Jernigan. Detail preserving impulsive noise removal. *Signal Processing: Image Communication,*, 19(10):993–1003, 11 2004. 7

[2] Z. Azimifar, P. Fieguth, and M. E. Jernigan. Modeling the correlation structure of images in the wavelet domain. *Electrical and Computer Engineering, 2001. Canadian Conference on*, 2:1123–1127, 2001. 9, 12

[3] Z. Azimifar, P. Fieguth, and M. E. Jernigan. Wavelet shrinkage with correlated wavelet coefficients. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 3:162–165, 2001. 9

[4] Z. Azimifar, P. Fieguth, and M. E. Jernigan. Correlated wavelet shrinkage: models of local random fields across multiple resolutions. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 3:157–160, Sept. 2005. 9

[5] E. J. Balster, Y. F. Zheng, and R. L. Ewing. Feature-based wavelet shrinkage algorithm for image denoising. *Image Processing, IEEE Transactions on*, 14(12):2024–2039, Dec. 2005. 12

[6] P. G. J. Barten. *Contrast sensitivity of the human eye and its effects on image quality*. SPIE Optical Engineering Press. Bellingham, WA, c1999. 13

[7] A. Ben Hamza and H. Krim. Image denoising: a nonlinear robust statistical approach. *Signal Processing, IEEE Transactions on*, 49(12):3045–3054, Dec 2001. 5

[8] A. Benazza-Benyahia and J. C. Pesquet. An extended sure approach for multicomponent image denoising. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 2:945–948, 17-21 May 2004. 9

[9] R. W. Buccigrossi and E. P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *Image Processing, IEEE Transactions on*, 8(12):1688–1701, Dec 1999. 9

[10] C. S. Burrus, R. A. Gopinath, and H. T. Guo. *Introduction to wavelets and wavelet transforms : a primer*. Prentice Hall, Upper Saddle River, N. J., c1998. 10, 17, 18, 20

[11] S. G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532(1)–1533, 2000. 26, 46, 47

[12] T. Chen and H. R. Wu. Adaptive impulse detection using center-weighted median filters. *Signal Processing Letters, IEEE*, 8(1):1–3, Jan 2001. 7

[13] H. Cheng, J. W. Tian, J. Liu, and Q. Z. Yu. Wavelet domain image denoising via support vector regression. *Electronics Letters*, 40(23):1479–1481, Nov. 2004. 10

[14] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. Technical report, Department of Statistics, 1995. 22

[15] Y. Q Cui and K. Wang. Markov random field modeling in the wavelet domain for image denoising. *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, 9:5382–5387, Aug. 2005. 10

[16] M. Dai, C. Peng, A. K. Chan, and D. Loguinov. Bayesian wavelet shrinkage with edge detection for SAR image despeckling. *Geoscience and Remote Sensing, IEEE Transactions on*, 42(8):1642–1648, Aug. 2004. 9, 27

[17] W. Z. Dai and Y. L. Ye. Image denoising based on combination of wiener filter and wavelet shrinkage. *Integration Technology, 2007. ICIT '07. IEEE International Conference on*, pages 421–425, March 2007. 12

[18] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995. 8, 13, 23, 46, 47

[19] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. 8, 9, 24, 26, 27, 46, 47

[20] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995. 8, 26, 46, 47

[21] S. Fei and R. Zhao. Adaptive wavelet shrinkage for image denoising based on sure rule. *Signal Processing, The 8th International Conference on*, 1:16–20, 2006. 9

[22] N. Z. Feng, L. Y. Ma, and Y. Shen. Fuzzy partition based curvelets and wavelets denoise algorithm. *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, pages 23–26, Dec. 2007. 37

[23] B. Goossens, A. Pizurica, and W. Philips. Removal of correlated noise by modeling spatial correlations and interscale dependencies in the complex wavelet domain. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 1:317–320, 2007. 11, 31, 47

[24] K. K. Gupta and R. Gupta. Feature adaptive wavelet shrinkage for image denoising. *Signal Processing, Communications and Networking, 2007. ICSCN '07. International Conference on*, pages 81–85, Feb. 2007. 12

[25] S. Gupta, L. Kaur, R. C. Chauhan, and S. C. Saxena. A wavelet based statistical approach for speckle reduction in medical ultrasound images. In *(2003). IEEE TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, number 2, pages 534–537, India, Oct. 2003. 32

[26] H. B. Ha and T. N. Truong. Image denoising using the non-uniform directional filter bank and hidden markov tree. *Digital Signal Processing Workshop, 12th - Signal Processing Education Workshop, 4th*, pages 214–217, Sept. 2006. 11

[27] R. L. Haupt. *Practical genetic algorithms*. Wiley-Interscience, Hoboken, N.J., 2nd ed edition, c2004. 44, 45

[28] M. M. Ichir and A. Mohammad-Djafari. Hidden markov models for wavelet-based blind source separation. *Image Processing, IEEE Transactions on*, 15(7):1887–1899, July 2006. 10

[29] A. K. Jain. *Fundamentals of digital image processing*. Prentice Hall, Englewood Cliffs, NJ, c1989. 7

[30] J. S. Jang, R. C. T. Sun, and E. J. Mizutani. *Neuro-fuzzy and soft computing : a computational approach to learning and machine intelligence*. Prentice Hall, Upper Saddle River, N.J., c1997. 17, 38, 39, 43

[31] M. Jansen. *Noise reduction by wavelet thresholding*, volume 161 of *Lecture notes in statistics (Springer-Verlag)*. Springer, New York, c2001. 8, 17, 24

[32] F. Jin. *Wavelet-based image and video processing*. University of Waterloo, Waterloo, Ont., 2004. 21, 22, 24

[33] I. M. Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the Royal Statistical Society B*, 59(2):319–351, 1997. 8

[34] F. O. Karray and Clarence W. D. S. *Soft computing and intelligent systems design: theory, tools, and applications*. Pearson/Addison Wesley, Harlow, England ; New York, 2004. 17, 28, 38, 39, 41, 42, 43

[35] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan. Image denoising with nonparametric hidden markov trees. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 3:121–124, 2007. 11

[36] H. I. Koo and N. I. Cho. Prior model for the mrf modeling of multi-channel images. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 1:713–716, April 2007. 10

[37] E. E. Kuruoglu, W. J. Fitzgerald, and P. J. W. Rayner. Near optimal detection of signals in impulsive noise modeled with a symmetric a-stable distribution. *Communications Letters, IEEE*, 2(10):282–284, Oct 1998. 5

[38] C. M. Lee, S. S. Yang, and C. L. Ho. Modified back-propagation algorithm applied to decision-feedback equalization. *Vision, Image and Signal Processing, IEE Proceedings*, 153(6):805– 809, 2006. 39

[39] Y. Li and C. Moloney. Selective wavelet coefficient soft thresholding scheme for speckle noise deduction in sar images. *IEEE Workshop on Nonlinear Signal and Image Processing*, 1997. 31

[40] Z. W. Liao, C. M. Lam, and Y. Y. Tang. Image processing using template model and wavelet domain hidden markov model. *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 7:4302–4307, Aug. 2004. 10

[41] L. Y. Ma, J. C. Ma, and Y. Shen. Pixel fusion based curvelets and wavelets denoise algorithm. *Engineering Letters*, 14(2):130–134, 06 2007. 12, 37

[42] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(7):710–732, Jul 1992. 22

[43] S. G. Mallat. *Wavelet tour of signal processing*. Academic Press, San Diego, CA, 1999. 22

[44] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, Jul 1989. 18, 20

[45] J. M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, New Jersey, 2001. 17, 29, 30, 34, 35, 36

[46] J. M. Mendel. Fuzzy sets for words: a new beginning. *International Conference on Fuzzy Systems*, pages 37–42, 2003. 17

[47] J. M. Mendel. Type-2 fuzzy sets: Some questions and answers. *Newsletter of the IEEE Neural Networks Society*, 1:10–13, 2003. 17, 30

[48] J. M. Mendel. Type-2 fuzzy sets and systems: an overview. *Computational Intelligence Magazine, IEEE*, 2(1):20–29, Feb. 2007. 17, 28, 30

[49] M. Mignotte. A post-processing deconvolution step for wavelet-based image denoising methods. *Signal Processing Letters, IEEE*, 14(9):621–624, Sept. 2007. 11

[50] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and Jr. F. C. Harris. Survey of image denoising techniques. In *in Proceedings of Global Signal Processing Expo and Conference (GSPx 04),*, Santa Clara, CA, USA, September 2004. 6, 11

[51] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized gaussian and complexity priors. *Information Theory, IEEE Transactions on*, 45(3):909–919, Apr. 1999. 10

[52] A. Pizurica and W. Philips. Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising. *Image Processing, IEEE Transactions on*, 15(3):654– 665, March 2006. 10, 12

[53] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy. A wavelet-based image denoising technique using spatial priors. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 3:296–299, 2000. 10, 11

[54] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy. A joint inter- and intrascale statistical model for bayesian wavelet based image denoising. *Image Processing, IEEE Transactions on*, 11(5):545–557, May 2002. 10, 11, 22

[55] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Adaptive wiener denoising using a gaussian scale mixture model in the wavelet domain. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 2:37–40, Oct 2001. 10, 12

[56] P. Puvanathasan and K. Bizheva. Speckle noise reduction algorithm for optical coherence tomography based on interval type II fuzzy set. In *Optical Express*, volume 15, pages 15747–15758, 2007. 2, 12, 17, 27, 31, 32

[57] H. Rabbani, M. Vafadust, I. Selesnick, and S. Gazor. Image denoising based on a mixture of bivariate laplacian models in complex wavelet domain. *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 425–428, Oct. 2006. 10

[58] S. Ray, A. Chan, and B. Mallick. Bayesian wavelet shrinkage in transformation based normal models. *Image Processing. 2002. Proceedings. 2002 International Conference on*, 1:876–879, 2002. 9

[59] S. Ray and B. K. Mallick. A bayesian transformation model for wavelet shrinkage. *Image Processing, IEEE Transactions on*, 12(12):1512–1521, Dec. 2003. 9, 46, 47

[60] A. E. Ruano. *Intelligent control systems using computational intelligence techniques.* Institution of Electrical Engineers, London, c2005. 17

[61] F. Russo. A FIRE filter for detail-preserving smoothing of images corrupted by mixed noise. *Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on*, 2:1051–1055, Jul 1997. 6

[62] S. Schulte, B. Huysmans, A. Pizurica, E. E. Kerre, and W. Philips. A new fuzzy-based wavelet shrinkage image denoising technique. In *ACIVS*, volume 4179 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 2006. 2, 12, 17, 26, 27, 31, 32, 46, 47

[63] L. Sendur and I. W. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *Signal Processing, IEEE Transactions on*, 50(11):2744–2756, Nov 2002. 10

[64] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *Information Theory, IEEE Transactions on*, 38(2):587–607, Mar 1992. 21

[65] H. S. Song, G. Q. Wang, and X. M. Zhao. A new adaptive multistage median filter. *pdcat*, 0:826–828, 2005. 7

[66] H. G. Stark. *Wavelets and Signal Processing: An Application-Based Introduction*. Springer, Berlin, Heidelberg, 2005. 7, 17, 19

[67] A. Taguchi. Removal of mixed noise by using fuzzy rules. *Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings KES '98. 1998 Second International Conference on*, 1:176–179, Apr 1998. 6

[68] H. R. Tizhoosh. Image thresholding using type II fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 38(12):2363–72, 2005. 29

[69] X. Wang. Lee filter for multiscale image denoising. *Signal Processing, The 8th International Conference on*, 1:16–20, 2006. 12

[70] Z. Wang and A. C. Bovik. A universal image quality index. *Signal Processing Letters, IEEE*, 9(3):81–84, Mar 2002. 14

[71] Z. M. Wang. Image denoising based on probability wavelet shrinkage with gaussian model. *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, 2:544–548, Nov. 2007. 10

[72] S. S. Yang, C. L Ho, and C. M. Lee. HBP: improvement in BP algorithm for an adaptive MLP decision feedback equalizer. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 53(3):240–244, 2006. 39

[73] X. Y. Yang and X. D. Zhang. Frame-based image denoising using local contextual hidden markov model. *Natural Computation, 2007. ICNC 2007. Third International Conference on*, 1:164–170, Aug. 2007. 10

[74] I. C. Yeh. Modeling chaotic two-dimensional mapping with fuzzy-neuron networks. *Fuzzy Sets and Systems*, 105(3):421–427, 1999. 41

[75] L. Yin, R. K. Yang, M. Gabbouj, and Y. Neuvo. Weighted median filters: a tutorial. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 43(3):157–192, Mar 1996. 7

[76] X. H. Yuan and B. P. Buckles. Subband noise estimation for adaptive wavelet shrinkage. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 4:885–888, Aug. 2004. 99

[77] L. A. Zadeh. Fuzzy sets. *Inform. Control*, 8:338–353, 1965. 17, 27

[78] M.E. Zervakis and A. N. Venetsanopoulos. Linear and nonlinear image restoration under the presence of mixed noise. *Circuits and Systems, IEEE Transactions on*, 38(3):258–272, Mar 1991. 6

[79] Y. Zhai, M. Yeary, V. DeBrunner, J. P. Havlicek, and O. Alkhouli. Image restoration using a hybrid combination of particle filtering and wavelet denoising. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2:II–790–3, 11-14 Sept. 2005. 12

[80] D. Zhang and M. E. Jernigan. An information theoretic criterion for image quality assessment based on natural scene statistics. *Image Processing, 2006 IEEE International Conference on*, pages 2953–2956, Oct. 2006. 15

[81] D. B. Zhang, Y. N. Wang, H. X. Huang, and L. Z. Yi. Image fusion filtering based on fuzzy neuron network. *Journal of Data Acquisition and Processing*, 21(1):95–98, Mar. 2006. 37, 41

[82] H. Zhang, A. Nosratinia, and R. O. Jr. Wells. Image denoising via wavelet-domain spatially adaptive FIR Wiener filtering. *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00.*, 4:2179–2182, 2000. 12

[83] X. W. Zhang, L. Zhu, and X. B. Zheng. Study to the image denoising algorithm based on multiwavelet transforms. *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, 4:1803–1807, Nov. 2007. 11

[84] Y. J. Zhang and L. L. Ge. Region-based image fusion approach using iterative algorithm. *icis*, 0:202–207, 2008. 13, 37

[85] C. X. Zheng and Y. M. Zhang. Low-field pulsed NMR signal denoising based on wavelet transform. *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, pages 1–4, June 2007. 12

[86] W. Zheng and J. X. Chen. A method of image noise reduction based on multiwavelet transform and multiscale data fusion. *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 200–203, 28 2007-Dec. 1 2007. 13