Comparison and Analysis of Different Image Denoising Algorithms

Jianeng XU, Rui Zhang\*

*Abstract*—Digital images play very important role in daily life applications as well as in the areas of research and technology .But main disadvantage of digital images is that noise is added during their transmission. Noise in the digital images can be removed by several techniques. Selecting appropriate technique for de-noising plays an important role. Several techniques are proposed for image de-noising and each technique has its advantages and disadvantages. Mainly two types of denoising techniques are there .One is spatial domain filtering and other is transform domain filtering. Choice of denoising technique is dependent upon the type of noise which is present in the digital image. In this paper after a brief introduction, overview of different denoising techniques and their analysis is given.

*Index Terms*—Image denoising, mean filter, wiener filter, media filter, wavelet transform

# INTRODUCTION

I

mages are often corrupted with noise throughout acquisition, transmission, and the retrieval from storage media. Numerous dots can be spotted in a Photograph taken with a digital camera under low lighting conditions. Noise corrupts both images and videos. The reason of the denoising algorithm is to remove this noise. Image denoising is required because a noisy image is not pleasant to view. In addition, some fine details in the image may be confused with the noise or vice-versa. Many image processing algorithms such as pattern recognition need a clean image to work effectively. Random and uncorrelated noise samples are not compressible. Such concerns underline the importance of denoising in image and video processing. Images are affected by different types of noise, The work presented herein focuses on a zero mean additive white Gaussian noise (AWGN). Zero mean does not lose generality, as the non-zero mean can be subtracted to get to a zero mean model. In the case of noise being correlated with the signal, it can be de-correlated prior to using this method to mitigate it. The difficulty of denoising can be mathematically presented as follows,

Y=X+N (1)

Where Y is the analyzed noisy image, X is the original image and N is the AWGN noise with variance. The objective is to estimate X for given Y. A most excellent estimate can be written as the conditional mean  .

The difficulty lies in determining the probability density function . The purpose of an image-denoising algorithm is to find a best estimate of X. Although numerous denoising algorithms have been published, there is scope for improvement.

# Additive Noises

In this chapter we discuss noise commonly present in an image. Note that noise is undesired information that contaminates the image. In the image denoising process, information about the type of noise present in the original image plays a significant role. Typical images are corrupted with noise modeled with either a Gaussian, uniform, or salt and pepper distribution.

Noise is present in an image in an additive form .An additive noise follows the rule:

w ( x , y ) = s ( x , y ) + n ( x , y) (2)

where s(x,y) is the original signal, n(x,y) denotes the noise introduced into the signal to produce the corrupted image w(x,y), and (x,y) represents the pixel location. The above

image algebra is done at pixel level.

## Gaussian Noise

Gaussian noise is evenly distributed over the signal. This means that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value. As the name indicates, this type of noise has a Gaussian distribution, which has a bell shaped probability distribution function given by,

C:\Users\CANONXU\AppData\Roaming\Tencent\Users\786460257\QQ\WinTemp\RichOle\{QJ%3YXDCCS@FRMDM[6XERB.jpg (3)

where g represents the gray level, m is the mean or average of the function, and σ is the standard deviation of the noise. Graphically, it is represented as shown in Figure 2.1.

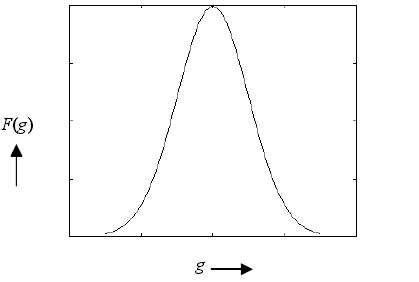


Figure 2.1: Gaussian distribution

When introduced into an image, Gaussian noise with zero mean and variance as 0.05would look as in Image 2.1. Image 2.2 illustrates the Gaussian noise with mean(variance) as 1.5 (10) over a base image with a constant pixel value of 100.

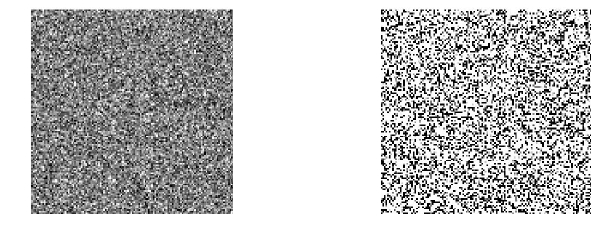


Image 2.1 Gaussian noise Image 2.2: Gaussian noise

(mean=0, variance 0.05) (mean=1.5, variance 10)

## Salt and Pepper Noise

Salt and pepper noise is an impulse type of noise, which is also referred to as intensity spikes. This is caused generally due to errors in data transmission. It has only two possible values, a and b. The probability of each is typically less than 0.1. The corrupted pixels are set alternatively to the minimum or to the maximum value, giving the image a “salt and pepper” like appearance. Unaffected pixels remain unchanged. For an 8-bit image, the typical value for pepper noise is 0 and for salt noise 255. The salt and pepper noise is generally caused by malfunctioning of pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process. The probability density function(PDF) for this type of noise is shown in Figure 2.2. Salt and pepper noise with a variance of 0.05 is shown in Image 2.3

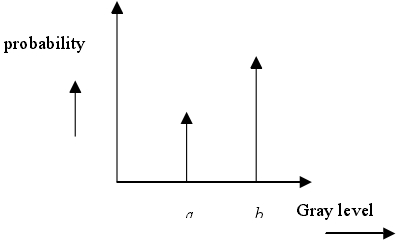


Figure 2.2: PDF for salt and pepper noise

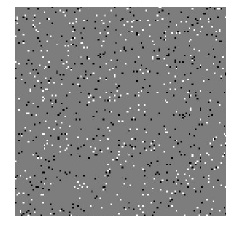


Image 2.3: Salt and pepper noise

In this part, we have discussed various types of noises considered in the thesis along with their distributions. Gaussian noise, salt and pepper noise can be generated from the Matlab Image Processing tool box function library.

## Speckle Noise

Speckle noise [Ga99] is a multiplicative noise. This type of noise occurs in almost all coherent imaging systems such as laser, acoustics and SAR(Synthetic Aperture Radar )imagery. The source of this noise is attributed to random interference between the coherent returns. Fully developed speckle noise has the characteristic of multiplicative noise. Speckle noise follows a gamma distribution and is given as

C:\Users\CANONXU\AppData\Roaming\Tencent\Users\786460257\QQ\WinTemp\RichOle\_I7$QP0UOW%E8V9]7F[{G40.jpg (3)

where variance is  and g is the gray level.

On an image, speckle noise (with variance 0.05) looks as shown in Image 2.4. The gamma distribution is given below in Figure 2.3.

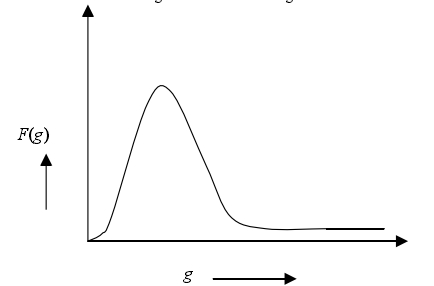


Figure 2.3: Gamma distribution

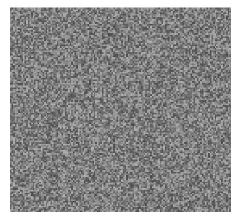


Image 2.4: Speckle noise

The behavior of each of these noises is described in the image 2.5. The first one is the original image-  magnolia flower, which is the City Flower of Shanghai, and the second one is the flower with Gaussian noise. The third one and the last one is the flower with sat & pepper noise, speckle noise respectively.



Image 2.5: (a)original image (b)Gaussian noise

(c) Salt &pepper noise (d) Speckle noise

# Basic Theory Of Image Denoising

There are many different kinds of image denoising algorithms. These can be broadly classified into two classes:

### • Spatial domain filtering

### • Transform domain filtering

As evident from the names, spatial domain filtering refers to filtering in the spatial domain, while the transform domain filtering refers to filtering in the transform domain. Image denoising algorithms those can use wavelet transforms fall into transform domain filtering. Spatial domain filtering can be divided further on the basis of the type of filter used:

### 1) • Linear filters

### • Non-Linear filters

An example of a linear filter is the Wiener filter in the spatial domain. As an example of a non-linear filter is the median filter. Median filtering is pretty useful in getting rid of Salt and Pepper type noise. Some spatial filters be liable to cause blurring in the denoised image. Transform domain filters be liable to cause Gibbs oscillations in the denoised image.

## A. Linear Filtering- Mean Filter

A mean filter acts on an image by smoothing it; that is, it reduces the intensity variation between adjacent pixels. The mean filter is nothing but a simple sliding window spatial filter that replaces the center value in the window with the average of all the neighboring pixel values including itself. By doing this, it replaces pixels, that are unrepresentative of their surroundings. It is implemented with a convolution mask, which provides a result that is a weighted sum of the values of a pixel and its neighbors. It is also called a linear filter. The mask or kernel is a square. Often a 3× 3 square kernel is used. If the coefficients of the mask sum up to one, then the average brightness of the image is not changed. If the coefficients sum to zero, the average brightness is lost, and it returns a dark image. The mean or average filter works on the shift-multiply-sum principle.

As we use a 3× 3 kernel shown in Figure 3.1. Note that the coefficients of this mask sum to one, so the image brightness is retained, and the coefficients are all positive, so it will tend to blur the image.

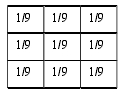


Figure 3.1: A constant weight 3× 3 filter mask

Example 3.1: For the following 3× 3 neighborhood, mean filtering is applied by convoluting it with the filter mask shown in Figure 3.2.

C:\Users\CANONXU\AppData\Roaming\Tencent\Users\786460257\QQ\WinTemp\RichOle\U0BBM3I7OM2BXLIPDX4AGIK.jpg

This provides a calculated value of 78. Note that the center value 200, in the pixel matrix, is replaced with this calculated value 78. This clearly demonstrates the mean filtering process.

Computing the straightforward convolution of an image with this kernel carries out the mean filtering process. It is effective when the noise in the image is of impulsive type. The averaging filter works like a low pass filter, and it does not allow the high frequency components present in the noise to pass through. It is to be noted that larger kernels of size 5× 5 or 7× 7 produces more denoising but make the image more blurred. A trade off is to be made between the kernel size and the amount of denoising.

The filter discussed above is also known as a constant coefficient filter because the weight matrix does not change during the whole process. Mean filters are popular for their simplicity and ease of implementation. We have implemented the averaging filter using Matlab. The image “monkey.jpg” is read into the program as shown in Image 3.1(a). This image is of size 256× 256. Gaussian noise is added to this image as shown in Image 3.1(b). The pixel values of this corrupted image are copied into a 2-dimentional array of size 256× 256.



Image 3.1: (a) original image (b) Gaussian noise

Then, we use the 3× 3 weight matrix，9× 9 weight matrix and 15× 15 weight matrix for simulation. A weight matrix is initialized. Selecting the window over the 256× 256 pixel matrix, the weighted sum of the selected window is computed. The result replaces the center pixel in the window. For the next iteration, the window moves by one column to the right. The window movement is considered in the horizontal direction first and then in the vertical direction until all the pixels are covered. The result is shown in the Image 3.2. The lower the matrix is, the better the performance.



Image 3.2: (a) Mask 3\*3 (b) Mask 9\*9 (c) Mask 15\*15

## Linear Filtering-Wiener Filter

The Wiener filter can be used in image processing to remove noise from a picture. In [mathematics](http://en.wikipedia.org/wiki/Mathematics), Wiener deconvolution  is an application of the [Wiener filter](http://en.wikipedia.org/wiki/Wiener_filter) to the [noise](http://en.wikipedia.org/wiki/Noise) problems inherent in [deconvolution](http://en.wikipedia.org/wiki/Deconvolution" \o "Deconvolution). It works in the [frequency domain](http://en.wikipedia.org/wiki/Frequency_domain), attempting to minimize the impact of deconvolved noise at frequencies which have a poor [signal-to-noise ratio](http://en.wikipedia.org/wiki/Signal-to-noise_ratio).

The Wiener deconvolution method has widespread use in [image](http://en.wikipedia.org/wiki/Image) deconvolution applications, as the frequency spectrum of most visual images is fairly well behaved and may be estimated easily.

The operation of the Wiener filter becomes apparent when the filter equation above is rewritten:


\begin{align}
 G(f) & = \frac{1}{H(f)} \left[ \frac{ |H(f)|^2 }{ |H(f)|^2 + \frac{N(f)}{S(f)} } \right] \\
      & = \frac{1}{H(f)} \left[ \frac{ |H(f)|^2 }{ |H(f)|^2 + \frac{1}{\mathrm{SNR}(f)}} \right]
\end{align}
 (7)

Here, \ 1/H(f) is the inverse of the original system, and \ \mathrm{SNR}(f) = S(f)/N(f) is the [signal-to-noise ratio](http://en.wikipedia.org/wiki/Signal-to-noise_ratio). When there is zero noise (i.e. infinite signal-to-noise), the term inside the square brackets equals 1, which means that the Wiener filter is simply the inverse of the system, as we might expect. However, as the noise at certain frequencies increases, the signal-to-noise ratio drops, so the term inside the square brackets also drops. This means that the Wiener filter attenuates frequencies dependent on their signal-to-noise ratio.

The Wiener filter equation above requires us to know the spectral content of a typical image, and also that of the noise. Often, we do not have access to these exact quantities, but we may be in a situation where good estimates can be made. For instance, in the case of photographic images, the signal (the original image) typically has strong low frequencies and weak high frequencies, and in many cases the noise content will be relatively flat with frequency.

We have implemented the wiener filter using Matlab. The image “monkey.jpg” is read into the program and the Gaussian noise is added to this image firstly. Then, we use the 3× 3 weight matrix，9× 9 weight matrix and 15× 15 weight matrix for simulation. The result is shown in the Image 3.3. It looks that the 9× 9weight matrix can achieve a better performance.



Image 3.3: (a) Mask 3\*3 (b) Mask 9\*9 (c) Mask 15\*15

## Nonlinear Filtering- Median Filter

A median filter belongs to the class of nonlinear filters unlike the mean filter. The median filter also follows the moving window principle similar to the mean filter. A 3× 3, 5× 5, or 7× 7 kernel of pixels is scanned over pixel matrix of the entire image. The median of the pixel values in the window is computed, and the center pixel of the window is replaced with the computed median. Median filtering is done by, first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. Note that the median value must be written to a separate array or buffer so that the results are not corrupted as the process is performed. Figure 3.2 illustrates the methodology. Neighborhood values: 115, 119, 120, 123, 124, 125, 126, 127, 150. Median value: 124.

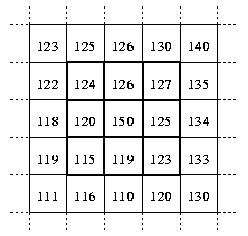


Figure 3.2: Concept of median filtering.

The central pixel value of 150 in the 3×3 window shown in Figure 3.3 is rather unrepresentative of the surrounding pixels and is replaced with the median value of 124.

The median is more robust compared to the mean. Thus, a single very unrepresentative pixel in a neighborhood will not affect the median value significantly. Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter. These advantages aid median filters in denoising uniform noise as well from an image.

To verify the performance of median filter for image denoising, we have implemented the media filter with the image “monkey.jpg”, which is read into the program and the Gaussian noise is added to this image firstly. Then, we use the 3× 3 weight matrix，9× 9 weight matrix and 15× 15 weight matrix for simulation. The result is shown in the Image 3.4. It looks that the 3× 3 weight matrix can achieve a better performance. As matrix level becomes higher , the image turns blur.



Image 3.4: (a) Mask 3\*3 (b) Mask 9\*9 (c) Mask 15\*15

## Wavelet Transforms

Wavelets are mathematical functions that analyze data according to scale or resolution. They aid in studying a signal in different windows or at different resolutions. For instance, if the signal is viewed in a large window, gross features can be

noticed, but if viewed in a small window, only small features can be noticed.

Wavelets provide some advantages over Fourier transforms. For example, they do a good job in approximating signals with sharp spikes or signals having discontinuities. Wavelets can also model speech, music, video and non-stationary stochastic signals. Wavelets can be used in applications such as image compression, turbulence, human vision, radar, earthquake prediction, etc..

The term “wavelets” is used to refer to a set of orthonormal basis functions generated by dilation and translation of scaling function φ and a mother wavelet ψ. The finite scale multiresolution representation of a discrete function can be

called as a discrete wavelet transform. DWT is a fast linear operation on a data vector, whose length is an integer power of 2. This transform is invertible and orthogonal, where the inverse transform expressed as a matrix is the transpose of the transform matrix. The wavelet basis or function, unlike sines and cosines as in Fourier transform, is quite localized in space. But similar to sines and cosines, individual wavelet functions are localized in frequency.

The orthonormal basis or wavelet basis is defined as

 (4)

The scaling function is given as

 (5)

where ψ is called the wavelet function and j and k are integers that scale and dilate the wavelet function. The factor ‘j’ in Equations (4) and (5) is known as the scale index, which indicates the wavelet’s width. The location index k provides the position. The wavelet function is dilated by powers of two and is translated by the integer k. In terms o f the wavelet coefficients, the wavelet equation is

 (6)

where g0, g1, g2 are high pass wavelet coefficients. Writing the scaling equation in terms of the scaling coefficients as given below, we get

 (7)

The function φ(x) is the scaling function and the coefficients h0

, h1, h2,… are low pass scaling coefficients. The wavelet and scaling coefficients are related by the quadrature mirror relationship, which is

 (8)

The term N is the number of vanishing moments. A graphical representation of DWT is shown in Figure 4.1. Note that, Y0 is the initial signal.

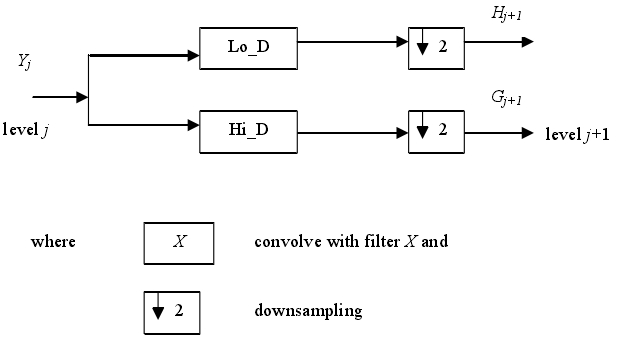


Figure 3.3: A 1-Dimensional DWT - Decomposition step

As mentioned earlier, the wavelet equation produces different wavelet families like Daubechies, Haar, coiflets, etc. Wavelets are classified into a family by the number of vanishing moments N. Within each family of wavelets there are wavelet subclasses distinguished by the number of coefficients and by the level of iterations.

In the last , we implement the media filter with the image “monkey.jpg”, which is read into the program and the Gaussian noise is added to this image firstly. Then, we use the wavelet tranform for first reconstruction, second reconstruction and third reconstruction. The result is shown in the Image 3.3.As we can see from the image, the first construction gets the best performancr and the third construction gets the worst performance. Therefore, there is no doubt that we had better use the first construction when implement the wavelet for image denoising.



Image 2.5: (a) First (b) Second (c) Third

# Comparison

The selection of the denoising technique is application dependent. So, it is necessary to learn and compare denoising techniques to select the technique that is apt for the application in which we are interested.

This part deals with the comparison of the denoising techniques, namely, linear including mean filter, wiener filter and non-linear filtering, wavelet based denoising.

As we can see from the Image4.1, we implement the filters noted before with the image which is added with the gaussian noise. Image 4.1(a) shows the original image and the (b) shows the image with gaussian noise. After that, we use mean filter, wiener filter, median filter and wavelet transform to denoise the image, which we can see from the Image 4.1(c)-(d). It is easy to find that the wiener filter and the median filter achieve the better performance for gaussian noise.



Image4.1 (a) Original (b) Gaussian noise

(c) Mean filter (d) Wiener filter

(e) Median filter (f) Wavelet transform

Then, we implement the filters noted before with the image which is added with the salt & pepper noise. Image 4.2(a) shows the original image and the (b) shows the image with gaussian noise. After that, we use mean filter, wiener filter, median filter and wavelet transform to denoise the image, which we can see from the Image 4.2(c)-(d). It is easy to find that the wiener filter achieves the better performance for salt & pepper noise.



Image4.2 (a) Original (b) Salt & pepper noise

(c) Mean filter (d) Wiener filter

(e) Median filter (f) Wavelet transform

In the last, we implement the filters noted before with the image which is added with the speckle noise. Image 4.3(a) shows the original image and the (b) shows the image with Gaussian noise. After that, we use mean filter, wiener filter, median filter and wavelet transform to denoise the image, which we can see from the Image 4.3(c)-(d). It is easy to find that the wiener filter and the median filter achieve the better performance for Speckle noise and the wavelet transform is not suitable for image with speckle noise.



Image4.3 (a) Original (b) Speckle noise

(c) Mean filter (d) Wiener filter

(e) Median filter (f) Wavelet transform

# Conclusion

In this paper, we have reviewed the various methodologies for image denoising. The performance of several well known algorithms for denoising images were investigated and demonstrates the different types of noises that may cause a natural image in real life, such as amplification noise, shot noise, quantization noise. This work was devoted to the review on the performances of image denoising algorithms based on various methods including the Wavelet transform. The Wavelet transform and its characteristics were investigated through literature review. Effects of different Wavelet bases on the denoising performance were studied. Wavelets may be good for denoising of images because of their energy compactness, sparseness and correlation properties still; simple thresholding methods are inadequate in their denoising performance. There are various image denoising filters such as moving average, Wiener filtering, median filtering and the non-local mean algorithm can be adopted for getting the optimum Image denoising performance.

References

1. *Anestis Antoniadis, Jeremie Bigot, “Wavelet Estimators in Nonparametric Regression: A Comparative Simulation Study,” Journal of Statistical Software, Vol 6, I 06, 2001*
2. *Castleman Kenneth R, Digital Image Processing, Prentice Hall, New Jersey, 1979.*
3. *S. Grace Chang, Bin Yu and Martin Vetterli, “Adaptive Wavelet Thresholding for Image Denoising and Compression,” IEEE Trans. Image Processing, Vol 9, No. 9, Sept 2000, pg 1532-1546.*
4. *David L. Donoho and Iain M. Johnstone, “Adapting to Unknown Smoothness via Wavelet Shrinkage,” Journal of American Statistical Association, 90(432):1200-1224, December 1995.*
5. Langis Gagnon, “Wavelet Filtering of Speckle Noise-Some Numerical Results,” Proceedings of the Conference Vision Interface 1999, Trois-

Riveres.

1. Amara Graps, “An Introduction to Wavelets,” IEEE Computational Science and Engineering, summer 1995, Vol 2, No. 2.
2. David Harte, Multifractals Theory and applications, Chapman and Hall/CRC, New York, 2001.
3. Reginald L. Lagendijk, Jan Biemond, Iterative Identification and Restoration of Images, Kulwer Academic, Boston, 1991.
4. Wayne Niblack, An Introduction to Digital Image Processing, Prentice Hall, New Jersey, 1986.
5. S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, San Diego, CA, 1999.
6. B. Vidakovic, Statistical modeling by wavelets, John Wiley and Sons, Inc. New York, 1999.
7. J.N. Lin, X. Nie, and R. Unbehauen, “Two-Dimensional LMS Adaptive

Filter Incorporating a Local-Mean Estimator for Image Processing,” IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol 40, No.7 July 1993, pg. 417-428.