A NEAR-OPTIMAL WAVELET-BASED ESTIMATION TECHNIQUE FOR VIDEO SEQUENCES

BY

MELODY I. BONHAM

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:

Associate Professor Farzad Kamalabadi

ABSTRACT

This thesis presents a method for estimation of a video signal given a data set with Poisson noise. The cameras used in creating video sequences are often charge-coupled devices, which produce data by way of a counting process, leading to noise with a Poisson distribution. Because many applications using video require data with less noise, a method of reducing the noise and estimating the original signal is desired. The method presented in this thesis attempts to accomplish this goal without using a Wiener filter, which can de-noise signals and is optimal in the mean-square error sense, but is hard to implement because second-order statistics may be unknown and because of the inversion of a possibly large matrix. Instead, an approximation of the Wiener filter is accomplished by first performing a one-dimensional discrete Fourier transform in order to decorrelate the video sequence between each two-dimensional frame or across each channel, and then performing a two-dimensional discrete wavelet transform on each of the resulting frames. Thresholding is then implemented, and the inverse transform is applied in order to recover an estimate of the original signal. It is shown that this scheme is effective in improving signal-to-noise ratio in synthetic video sequences and video captured by a camera.

To my Mother and Father

ACKNOWLEDGMENTS

I would like to thank Professor Farzad Kamalabadi for his help and guidance throughout the research process. He has provided the enthusiasm for the topic which has in turn excited my own. I would also like to thank Ian Atkinson for writing the original code which I used, answering my questions, and giving advice for how to set my thesis apart from other work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

fMRI	Functional magnetic resonance imaging
CCD	Charge-coupled device
SNR	Signal-to-noise ratio
KL	Karhunen-Loeve
DWT	Discrete wavelet transform
DFT	Discrete Fourier transform
FFT	Fast Fourier transform

LIST OF SYMBOLS

σ	Standard deviation
σ^2	Variance
$\mathcal{W}(\cdot)$	Wavelet transform operator
τ	Wavelet threshold
τ_B	BayesShrink wavelet threshold

CHAPTER 1

INTRODUCTION

This thesis presents a wavelet-based method of estimating a video sequence given its noisy observation. A video is a good example of a two-dimensional (2D) multichannel signal and can be thought of as a sequence of 2D images or frames.

A signal can contain noise due to the method by which it is created. In the case of a charge-coupled device (CCD), pixel intensities are determined according to a counting process, which gives rise to data with a Poisson distribution. It is possible to increase the signal-to-noise ratio (SNR) of a signal given its noisy observation. If information is known about the original signal and its noise, in particular the second-order statistics of both, many methods exist by which to estimate the signal.

The Wiener filter, which was first introduced in 1949 [1], is perhaps the most well-known signal estimator. It is optimal in the mean-square error sense. The Wiener filter uses second-order statistics of the signal and the noise to decorrelate the data by transforming it into the Karhunen-Loeve domain via a transform known as the Karhunen-Loeve (KL) transform; the signal and noise have different characteristics in this domain, allowing the noise to be recognized and removed with minimal impact on the original signal. However, as the Wiener filter relies on second-order statistics, if they are unknown or cannot be accurately modeled, then another method of signal estimation is required. If the second-order statistics are not known, then the estimation technique is referred to as *blind estimation*; it is *blind* because it is accomplished independently of knowledge of the signal and noise. A method of estimation in the blind case has been found with the use of wavelets.

A wavelet [2] is a signal processing tool which near-optimally decorrelates a signal. In this way, a discrete wavelet transform (DWT) approximates a KL transform, which optimally decorrelates a signal when the signal's second-order statistics are known. Therefore, a DWT also approximates a Wiener filter. Implementation of the discrete wavelet transform is accomplished through the use of a wavelet filter bank. This transform moves the data from its original domain into the wavelet domain. In the wavelet domain, the coefficients which come from the noise alone are small, and coefficients which come from the signal plus the noise are large [3, 4, 5]. The noisy coefficients can be removed by a method known as thresholding. The simplest thresholding to perform is hard thresholding, where coefficients above a certain level are kept and those below are zeroed out. In this way, the noise is reduced when data is returned to its original domain via an inverse discrete wavelet transform.

The wavelet transform method of removing noise from signals has been shown to work for one-dimensional signals as well as two-dimensional signals [6, 7, 8]. An image can be thought of as a two-dimensional signal. Therefore, this wavelet method for removing noise from a two-dimensional signal can be applied to images [9]. Furthermore, images can exist in a sequence in time, known as a video. Video is a case where multiple images are put together to form a three-dimensional signal or a multichannel image. Other multichannel images in addition to video include multispectral images, hyperspectral images, color images, and functional magnetic resonance imaging (fMRI) sequences. The process of de-noising multichannel images has been examined for non-video multichannel images, such as in [10, 11]. In these works, the noise is modeled as additive white Gaussian and the wavelets used are based on an orthogonal filter bank. In reality, the noise may not always be additive white Gaussian noise. A type of noise which occurs frequently instead has a Poisson distribution. Therefore, a method which performs de-noising of video in the presence of Poisson noise is desired. Such a method utilizing wavelets is developed in this thesis.

Orthogonal filter banks have traditionally also been used for wavelet de-noising of images, but, as [2] notes, biorthogonal filter banks have gained in popularity due to their better performance on images because they have more degrees of freedom than orthogonal filter banks and can be symmetric. Biorthogonal filter banks tend to have higher energy compaction, particularly for images with low frequencies [12], where frequency is determined by percentage of total image energy contained in the wavelet subband formed by filtering by two low-pass filters. A biorthogonal filter bank is also used in the de-noising scheme presented in this thesis in order to provide another method of de-noising.

This thesis expands upon a wavelet-based method for removing noise from multichannel images such as functional MRI and hyperspectral data [13, 14]. It extends the method to video by adding a different wavelet filter bank (biorthogonal) which performs better than the one used previously (orthogonal) when applied to single images. In addition, a Poisson noise distribution is used due to the manner in which images are created, instead of the Gaussian noise models used previously.

This thesis is organized in the following manner. In Chapter 2, some background necessary for the development which follows is presented. In particular, this chapter includes a definition and examples of two-dimensional multichannel data; a definition of Poisson noise; a review of the Wiener filter (one-dimensional, two-dimensional, two-dimensional multichannel); and a discussion of the discrete wavelet transform (two-channel filter bank and thresholding techniques). In Chapter 3, the development of a wavelet-based estimation technique is presented. In Chapter 4, filter parameters are chosen and the noise model is discussed. In Chapter 5, results of actual implementation on different video sequences is presented, and the performance of different filter banks is discussed. Chapter 6 contains conclusions and ideas about future work.

CHAPTER 2

BACKGROUND INFORMATION

2.1 Multichannel Images

A single image is made up of multiple orthogonal pixels. A multichannel image is simply a collection of single images. There are many examples of multichannel images, including multispectral images, where each channel is at a different frequency. Another example is a color image. Even though there is only one image visible, it is in fact made up of a combination of other images. Usually a color image can be decomposed into three images representing the color contribution of three colors (red, green, and blue) to the final image. Another is functional MRI (fMRI), where MRI images of neural activity are obtained every few seconds. The channels in this case are different times. Another example, video, is similar to the previous one. Video data is made up of images at different times, so the channels are again across time. While there are many other examples of multichannel images, the focus of this work is on video. Since an image can be treated as a two-dimensional signal, video can be seen as a three-dimensional signal, where two dimensions are spatial (an x-coordinate and a y-coordinate of the location of a particular pixel) and the third dimension is time. The individual images in time contained in video data are often referred to as frames. The time dimension gives the frame number of the pixel of interest, where the first frame is taken to be at time equals zero. With these definitions, video data composed of $N \times N$ images with T frames is a $N \times N \times T$ signal. The location of a particular pixel is given by $[n_1, n_2, t]$, and the pixel value is given by i.

With the signal given by x, this is represented by

$$x\left[n_1, n_2, t\right] = i$$

2.2 Noise with a Poisson Distribution

Video data is typically obtained using cameras. In this thesis, video obtained using a charge-coupled device (CCD) is examined. A CCD records an image by capturing photons. As the photons are absorbed by the CCD array, electrons are excited to a high energy state and captured in a well. The electrons are then counted after a set period of time. The total number counted is a combination of those electrons due to the image intensity, contribution due to thermal noise, and the contribution from read out errors [15]. Because it represents a counting method in order to generate a pixel value, the resulting values will have a Poisson distribution, which expresses the probability that a certain number of events occurs in a fixed period of time, given that the events occur with a known average rate and independently of the time elapsed since the last event. The number of electrons counted in the image acquisition can be modeled with a Poisson distribution [16].

The probability mass function (PMF) of a Poisson distribution is given by

$$Pr\left(X = k\right) = \frac{e^{-\lambda}\lambda^k}{k!}$$

where X is the distribution, k is the number of occurrences of an event, and λ is equal to the expected number of occurrences of an event during the given interval, or the mean of the distribution. An important property to note about Poisson distributions is that the mean is equal to the variance. Thus the standard deviation, which is the square root of the variance, is also equal to the square root of the mean.

A standard metric to measure the fidelity of a signal is the signal-to-noise ratio (SNR). It is a value which gives the ratio of the power of the signal to the power of the noise. SNR for images can be calculated by dividing the mean of the signal by the standard deviation.

With a Poisson distribution, the SNR is simply equal to the square root of the mean. This is the metric by which the performance of the de-noising method presented in this thesis will be evaluated.

2.3 Wiener Filter

In many applications, noisy data is not adequate. A method of reducing the noise in the data is sought. This problem has been studied for one- and two-dimensional signals. Many of the methods of removing noise from one- and two-dimensional signals can be extended to three-dimensional data, such as video sequences. One filter which has been shown to be optimal in the mean-square error (MSE) sense is the Wiener filter. The properties of this filter shall be examined for one-dimensional signals and then extended to two- and three-dimensional signals.

2.4 One-Dimensional Wiener Filter

To begin, it is assumed that the noise present in the observed signal is additive with zero mean. The observed signal, $x\left(n\right)$, is then composed of the desired signal, $s\left(n\right)$, and a noise signal, $w\left(n\right)$, and can be viewed as

$$x\left(n\right) = s\left(n\right) + w\left(n\right) \tag{2.1}$$

A filter, $g\left(n\right)$, is then applied to this signal in order to eliminate the noise component, resulting in a signal, $\hat{s}\left(n\right)$, given by

$$\hat{s}\left(n\right) = g\left(n\right) * x\left(n\right) \tag{2.2}$$

Substituting Equation (2.1) into Equation (2.2) yields

$$\hat{s}\left(n\right) = g\left(n\right) * \left(s\left(n\right) + w\left(n\right)\right) \tag{2.3}$$

The difference between the filtered signal, $\hat{s}\left(n\right)$, and the desired signal, $s\left(n\right)$, is represented by an error signal, $e\left(n\right)$.

$$e\left(n\right) = s\left(n\right) - \hat{s}\left(n\right) \tag{2.4}$$

Plugging Equation (2.3) into Equation (2.4) produces

$$e\left(n\right) = s\left(n\right) - \left(g\left(n\right) * \left(s\left(n\right) + w\left(n\right)\right)\right) = s\left(n\right) - \left(g\left(n\right) * x\left(n\right)\right) \tag{2.5}$$

The filter, g, needs to be chosen so that it minimizes the expected value of the square of the error, e. In order to solve this, is is desirable to first write Equation (2.3) as a convolution sum, such that

$$\hat{s}\left(n\right) = \sum_{k} g\left(k\right)\left[x\left(n-k\right)\right] \tag{2.6}$$

where $g\left(k\right)$ are the coefficients of a filter of order N.

Using the expectation operator $E\left|\cdot\right|$, the MSE is then given by

$$MSE = E|\hat{s}\left(n\right) - s\left(n\right)|^2 = E\left|\sum_{k} g\left(k\right)\left[x\left(n-k\right)\right] - s\left(n\right)\right|^2 \tag{2.7}$$

MSE is minimized by setting the partial derivatives equal to zero, as follows:

$$\begin{aligned} 0 &= \frac{\partial MSE}{\partial g\left(m\right)} \\ &= E\left\{2\left[\sum_{k} g\left(k\right)\left[x\left(n-k\right)\right] - s\left(n\right)\right] x\left(n-m\right)\right\} \\ &= \sum_{k} g\left(k\right) E\left[x\left(n-k\right) x\left(n-m\right)\right] - E\left[s\left(n\right) x\left(n-m\right)\right] \\ &= \sum_{k} g\left(k\right) R_x\left(m-k\right) - R_{sx}\left(m\right) \end{aligned}$$

where R_x is the autocorrelation of signal x and R_{sx} is the cross-correlation of signals s and x. Solving this equation gives

$$\sum_{k} g\left(k\right) R_x\left(m-k\right) = R_{sx}\left(m\right),\ 0 \leq m < L \tag{2.8}$$

where L is the number of filter coefficients. Equation (2.8) can be written in matrix form as

$$\begin{bmatrix} R_x(0) & R_x(1) & \cdots & R_x(L-1) \\ R_x(1) & R_x(0) & \cdots & R_x(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(L-1) & R_x(L-2) & \cdots & R_x(0) \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(L-1) \end{bmatrix} = \begin{bmatrix} R_{sx}(0) \\ R_{sx}(1) \\ \vdots \\ R_{sx}(L-1) \end{bmatrix} \tag{2.9}$$

To solve Equation (2.9) for the filter coefficients, the matrix of autocorrelations on the left must be inverted, yielding

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(L-1) \end{bmatrix} = \begin{bmatrix} R_x(0) & R_x(1) & \cdots & R_x(L-1) \\ R_x(1) & R_x(0) & \cdots & R_x(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(L-1) & R_x(L-2) & \cdots & R_x(0) \end{bmatrix}^{-1} \begin{bmatrix} R_{sx}(0) \\ R_{sx}(1) \\ \vdots \\ R_{sx}(L-1) \end{bmatrix} \tag{2.10}$$

2.5 Two-Dimensional Wiener Filter

Consider now a two-dimensional observed signal, $\mathbf{x}$, made up of the desired signal, $\mathbf{s}$, and a noise signal, $\mathbf{w}$, where $\mathbf{x} = x(n_1, n_2)$; other bold signals follow the same structure. In this case, all signals are matrices rather than vectors, as was the case for one-dimensional signals. For the two-dimensional case, Equation (2.1) now becomes

$$\mathbf{x} = \mathbf{s} + \mathbf{w} \tag{2.11}$$

Equations (2.2)-(2.5) follow the same pattern, replacing the one-dimensional signals with two-dimensional signals:

$$\hat{\mathbf{s}} = \mathbf{g} * \mathbf{x} \tag{2.12}$$

$$\hat{\mathbf{s}} = \mathbf{g} * (\mathbf{s} + \mathbf{w}) \tag{2.13}$$

$$\mathbf{e} = \mathbf{s} - \hat{\mathbf{s}} \tag{2.14}$$

$$\mathbf{e} = \mathbf{s} - (\mathbf{g} * (\mathbf{s} + \mathbf{w})) = \mathbf{s} - (\mathbf{g} * \mathbf{x}) \tag{2.15}$$

The filter, **g**, is now a two-dimensional filter.

Equations (2.6)-(2.8) follow the same pattern; the only difference is now the arguments are made up of two components and so the sums become double sums, as illustrated here:

$$\hat{s}\left(n_1, n_2\right) = \textstyle\sum_{k_1} \sum_{k_2} g\left(k_1, k_2\right) \left[x\left(n_1 - k_1, n_2 - k_2\right)\right] \tag{2.16}$$

$$\begin{aligned} MSE &= E|\hat{s}\left(n_1, n_2\right) - s\left(n_1, n_2\right)|^2 \\ &= E\left|\textstyle\sum_{k_1} \sum_{k_2} g\left(k_1, k_2\right) \left[x\left(n_1 - k_1, n_2 - k_2\right)\right] - s\left(n_1, n_2\right)\right|^2 \end{aligned} \tag{2.17}$$

$$\textstyle\sum_{k_1} \sum_{k_2} g\left(k_1, k_2\right) R_x\left(m_1 - k_1, m_2 - k_2\right) = R_{sx}\left(m_1\right), \ \left[m_1, m_2\right] \in L \tag{2.18}$$

In this case, the autocorrelation becomes $R_x\left(n_1, n_2\right) = E\left[s\left(m_1 + n_1, m_2 + n_2\right) x\left(m_1, m_2\right)\right]$, and the cross-correlation is $R_{sx}\left(n_1, n_2\right) = E\left[s\left(m_1 + n_1, m_2 + n_2\right) x\left(m_1, m_2\right)\right]$. The matrix form of this equation can also be found.

2.6 Multichannel Wiener Filter

In order to more simply address the multichannel Wiener filter, a new representation of the data is desired. Instead of using the data as a three-dimensional signal, as discussed in a previous section, it is possible to order it as a vector, as has been used in [17]. Taking an $N \times N$ matrix, ordering it lexicographically can be done either by row or by column, as long as each matrix is ordered the same. This means that when ordering a matrix by column, the result is a column vector where each consecutive column is stacked below the previous one, resulting in a $N^2 \times 1$ vector; the process for ordering according to row is the same, except ordering is by subsequent row instead and results in a $1 \times N^2$ vector. First, let the multichannel 2D signal be of size $N \times N \times M$. Each $N \times N$ observed channel, $\mathbf{x}_i$, is made up of a desired signal, $\mathbf{s}_i$, and a noise signal, $\mathbf{w}_i$. Then the i^{th} channel can be represented by

$$\mathbf{x}_i = \mathbf{s}_i + \mathbf{w}_i \tag{2.19}$$

Now, let $\mathbf{y}_i$, $\mathbf{v}_i$, and $\mathbf{p}_i$ be lexicographically ordered $\mathbf{x}_i$, $\mathbf{s}_i$, and $\mathbf{w}_i$, respectively, so that Equation (2.19) now becomes

$$\mathbf{y}_i = \mathbf{v}_i + \mathbf{p}_i \tag{2.20}$$

In order to combine all channels into a single length-N^2M vector, stack each channel vector on top of each other. Let this be defined as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_M^T \end{bmatrix}^T = \mathbf{v} + \mathbf{p} \tag{2.21}$$

A definition of the correlation matrix for the 2D multichannel case is necessary. It is shown in [18] that this is an $N^2M \times N^2M$ matrix $\mathbf{R_{vv}}$ given by

$$\mathbf{R_{vv}} = E\left[\mathbf{v}\mathbf{v}^H\right] = \begin{bmatrix} \mathbf{R}_{\mathbf{v}_1\mathbf{v}_1} & \mathbf{R}_{\mathbf{v}_1\mathbf{v}_2} & \cdots & \mathbf{R}_{\mathbf{v}_1\mathbf{v}_M} \\ \mathbf{R}_{\mathbf{v}_2\mathbf{v}_1} & \mathbf{R}_{\mathbf{v}_2\mathbf{v}_2} & \cdots & \mathbf{R}_{\mathbf{v}_2\mathbf{v}_M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{\mathbf{v}_M\mathbf{v}_1} & \mathbf{R}_{\mathbf{v}_M\mathbf{v}_2} & \cdots & \mathbf{R}_{\mathbf{v}_M\mathbf{v}_M} \end{bmatrix} \tag{2.22}$$

where the $N \times N$ cross-correlation matrix $\mathbf{R}_{\mathbf{v}_i\mathbf{v}_j}$ is given by

$$\mathbf{v}_{\mathbf{v}_i\mathbf{v}_j} = E\left[\mathbf{v}_i\mathbf{v}_j^H\right] \tag{2.23}$$

It is important to note at this point that the noise and signal are considered to be uncorrelated. Therefore, two substitutions can be made; namely, the cross-correlation between the observed signal and the desired signal is equal to the autocorrelation of the desired signal, and the autocorrelation of the observed signal is equal to the sum of the autocorrelations of the desired signal and the noise signal. Also of note, the noise is uncorrelated in both time and space, meaning that the noise correlation matrix $\mathbf{R_{pp}}$ is given by

$$\mathbf{R_{pp}} = E\left[\mathbf{n}\mathbf{n}^H\right] = \sigma_p^2\mathbf{I} \tag{2.24}$$

where σ_p^2 is the variance of the noise and $\mathbf{I}$ is an $N^2M \times N^2M$ identity matrix. Using these substitutions and the definitions of this section, the 2D multichannel version of Equations (2.8) and (2.18) is

$$\begin{aligned}\mathbf{R_{yy}}\mathbf{G}_w &= \mathbf{R_{vy}}\\ \left(\mathbf{R_{vv}} + \mathbf{R_{pp}}\right)\mathbf{G}_w &= \mathbf{R_{vv}}\\ \left(\mathbf{R_{vv}} + \sigma_p^2\mathbf{I}\right)\mathbf{G}_w &= \mathbf{R_{vv}}\end{aligned} \tag{2.25}$$

where $\mathbf{G}_w$ is the optimal estimator or Wiener filter coefficients.

To solve Equation (2.25) for the filter coefficients, invert the sum on the left, to give

$$\mathbf{G}_w = \left(\mathbf{R_{vv}} + \sigma_p^2\mathbf{I}\right)^{-1}\mathbf{R_{vv}} \tag{2.26}$$

Since the focus of this work is the 2D multichannel case, the review of the Wiener filter for this case will go further than the one- and two-dimensional cases. The next step is to write the eigenexpansion of Equation (2.26). Let the rank of $\mathbf{G}_w$ be represented by L_v, the eigenvectors by $\mathbf{u}_{\mathbf{v}_k}$, and the eigenvalues by $\lambda_{\mathbf{v}_k}$. Then, taking $\mathbf{\Lambda_v}$ as

$$\mathbf{\Lambda_v} = \operatorname{diag}\left[\frac{\lambda_{\mathbf{v}_1}}{\lambda_{\mathbf{v}_1} + \sigma_p^2}, \ldots, \frac{\lambda_{\mathbf{v}_{L_v}}}{\lambda_{\mathbf{v}_{L_v}} + \sigma_p^2}\right] \tag{2.27}$$

the eigenexpansion of Equation (2.26) is

$$\mathbf{R_{vv}} = \mathbf{U_v}\mathbf{\Lambda_v}\mathbf{U_v}^H = \sum_{k=1}^{L_v}\mathbf{u}_{\mathbf{v}_k}\frac{\lambda_{\mathbf{v}_k}}{\lambda_{\mathbf{v}_k} + \sigma_p^2}\mathbf{u}_{\mathbf{v}_k}^H \tag{2.28}$$

As is seen in Equation (2.28) above, $\mathbf{R}_{\mathbf{vv}}$ is eigenexpanded as $\mathbf{U}_{\mathbf{v}}\mathbf{\Lambda}_{\mathbf{v}}\mathbf{U}_{\mathbf{v}}^{H}$, which implies that since $\mathbf{U}_{\mathbf{v}}^{H}$ diagonalizes $\mathbf{R}_{\mathbf{vv}}$, it will optimally decorrelate the data in $\mathbf{v}$, which contains the spatial and channel data in a vector. As such, $\mathbf{U}_{\mathbf{v}}^{H}$ is termed the channel-spatial Karhunen-Loeve transform (KL transform, called such because the Wiener filter works on data in the KL domain). An interpretation of Equation (2.28) is that the filter first decorrelates the data in both space and channel (which is time in this case) via $\mathbf{U}_{\mathbf{v}}$, then weights the decorrelated coefficients with $\mathbf{\Lambda}_{\mathbf{v}}$, and finally recorrelates the data with $\mathbf{U}_{\mathbf{v}}^{H}$, which is optimal in the MSE sense, as derived for the 1-D case.

2.7 Discrete Wavelet Transform

Many transforms take data from one domain into another one, where sometimes it is easier to manipulate. The transform of focus in this thesis is the wavelet transform, which puts data into the wavelet domain. The development of the implementation is taken from [2].

2.7.1 Two-Channel Filter Banks

Instead of going into the details of the background of the discrete wavelet transform (DWT), the implementation is examined here. The DWT can be realized through a filter bank, which consists of filters and downsampling and upsampling functions. The simplest filter bank is a two-channel filter bank, as shown in Figure 2.1. The number of channels refers to the number of filters. In this case, $\tilde{h}$ is a highpass filter, and $\tilde{g}$ is a lowpass filter.

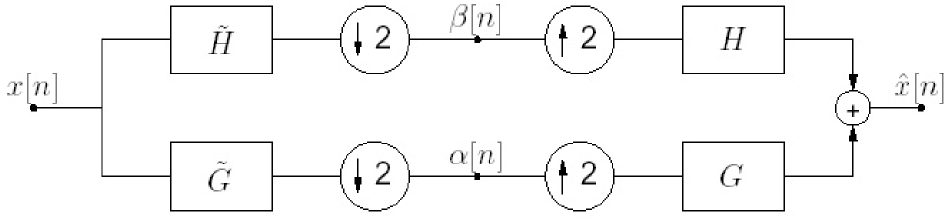


Figure 2.1: Two-channel filter bank.

The downsampling is performed with a factor of two, meaning that every other signal value is discarded, leaving two signals which are half the length of the original signal. This is called the *analysis* side of the filter bank. At this point, the signal is in the wavelet domain where it can be manipulated to reduce noise.

After transforming the signal, it is necessary to return it to the original domain, which is time in most cases (or time-space for video data). In order to do this, the process must be reversed. The first thing is to undo the downsampling by upsampling by a factor equal to the original factor, which in the two-channel case here is 2. Then, highpass and lowpass filters h and g, which reverse the original filters $\tilde{h}$ and $\tilde{g}$, are applied. This is called the *synthesis* side of the filter bank.

The filters $\tilde{g}$, $\tilde{h}$, g, and h may be any highpass and lowpass filters. Most of the time, it is desired that if nothing is done to the signal after the analysis is performed, the signal after the synthesis is exactly equal to the original signal. This is called *perfect reconstruction.* In order to achieve perfect reconstruction, the four filters must meet certain criteria. Using the z-transform of the filters, given by $\tilde{G}(z)$, $\tilde{H}(z)$, $G(z)$, and $H(z)$, respectively, the two requirements for perfect reconstruction are

$$\left[\tilde{H}\left(z\right)H\left(z\right)+\tilde{G}\left(z\right)G\left(z\right)\right]=2 \tag{2.29}$$

$$\left[\tilde{H}\left(-z\right)H\left(z\right)+\tilde{G}\left(-z\right)G\left(z\right)\right]=0 \tag{2.30}$$

In order to achieve perfect reconstruction, the four filters must be chosen in order to satisfy Equations (2.29) and (2.30).

It is possible to iterate the analysis process on the lowpass channel. In this case, the output of the lowpass filter, denoted α in Figure 2.1, is now the input to downsamplers and filters identical to $\tilde{H}$ and $\tilde{G}$. The number of times this is done is referred to as the number of *decomposition levels.* In order to undo the process, the same number of levels of the synthesis side is required. Each level is upsampled by the same factor and then filtered with the appropriate highpass or lowpass synthesis filters. Again, if nothing is done to the signal while it is in the wavelet domain, it is possible to achieve perfect reconstruction if the filters follow Equations (2.29) and (2.30).

Orthogonal Two-Channel Filter Banks

Orthogonal filter banks have filters which meet the criteria for a perfect biorthogonal filter bank and also have analysis filters which are time-reversed versions of the synthesis filters. Biorthogonal filter banks will be discussed more in a later section, but in order to give the orthogonal condition, biorthogonal filter banks must be treated briefly here. In particular, the biorthogonal condition that must be satisfied by the filters is

$$G\left(z\right)\tilde{G}\left(z\right)+G\left(-z\right)\tilde{G}\left(-z\right)=2 \tag{2.31}$$

$$H\left(z\right)\tilde{H}\left(z\right)+H\left(-z\right)\tilde{H}\left(-z\right)=2 \tag{2.32}$$

In order to satisfy the time-reverse criterion, the following must be true:

$$\tilde{G}\left(z\right)=G\left(z^{-1}\right) \tag{2.33}$$

$$\tilde{H}\left(z\right)=H\left(z^{-1}\right) \tag{2.34}$$

Therefore, in order to develop the equations for an orthogonal filter bank, plug Equation (2.33) into Equation (2.31) and Equation (2.34) into Equation (2.32), yielding

$$G\left(z\right)G\left(z^{-1}\right)+G\left(-z\right)G\left(-z^{-1}\right)=2 \tag{2.35}$$

$$H\left(z\right)H\left(z^{-1}\right)+H\left(-z\right)H\left(-z^{-1}\right)=2 \tag{2.36}$$

Perhaps the most famous orthogonal filter bank is the Haar filter bank [19], whose synthesis filters are given by

$$G\left(z\right)=1/\sqrt{2}\left(1+z^{-1}\right) \tag{2.37}$$

$$H\left(z\right)=1/\sqrt{2}\left(1-z^{-1}\right) \tag{2.38}$$

The corresponding analysis filters can be determined by using Equations (2.33) and (2.34).

Biorthogonal Two-Channel Filter Banks

One short-coming of orthogonal filter banks is that if all four filters are desired to be finite impulse response (FIR), then the solutions to the orthogonal equations do not have symmetric or antisymmetric impulse responses, which is a necessary condition for linear phase. A case which does have symmetric or antisymmetric impulse responses is a biorthogonal filter bank. As mentioned in the previous section, the biorthogonal conditions that the four filters must satisfy are

$$\begin{gathered} G(z)\tilde{G}(z) + G(-z)\tilde{G}(-z) = 2 \\ H(z)\tilde{H}(z) + H(-z)\tilde{H}(-z) = 2 \end{gathered}$$

It is possible to put the perfect reconstruction conditions into a matrix equation, so that

$$\begin{bmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{bmatrix} \begin{bmatrix} \tilde{G}(z) & \tilde{G}(-z) \\ \tilde{H}(z) & \tilde{H}(-z) \end{bmatrix} = 2\boldsymbol{I} \tag{2.39}$$

where $\boldsymbol{I}$ is a 2-by-2 identity matrix. Now define

$$\mathbf{\Phi}_m(z) = \begin{bmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{bmatrix}$$

and

$$\tilde{\mathbf{\Phi}}_m(z) = \begin{bmatrix} \tilde{G}(z) & \tilde{G}(-z) \\ \tilde{H}(z) & \tilde{H}(-z) \end{bmatrix}$$

so that Equation (2.39) now becomes

$$\mathbf{\Phi}_m(z)\,\tilde{\mathbf{\Phi}}_m(z) = 2\boldsymbol{I} \tag{2.40}$$

Then, taking all filters to be FIR, Equation (2.40) implies that $\det \mathbf{\Phi}_m(z) = \alpha z^l$, where l is an odd integer, and rearranging Equation (2.40) yields $\tilde{\mathbf{\Phi}}_m(z) = 2\left[\mathbf{\Phi}_m(z)\right]^{-1}$. From this latter equation, it can be seen that the highpass filters can be determined from the lowpass filters, according to

$$H(z) = \alpha z^{-(2k+1)} \tilde{G}(-z) \tag{2.41}$$

$$\tilde{H}(z) = \alpha^{-1} z^{2k+1} G(-z) \tag{2.42}$$

where α is any constant other than zero and k is a non-negative integer. In this case, the filters are linear phase, which is a desirable property since the frequency components are all delayed by the same amount; such a property is not possible with FIR orthogonal filter banks.

2.7.2 Two-Dimensional Discrete Wavelet Transform

The two-dimensional discrete wavelet transform (2D DWT) can be applied to images, which, as shown before, can be viewed as two-dimensional signals. It is possible to realize the 2D DWT using the 1D DWT discussed in the preceding section. In this case, the columns and the rows of the image are treated separately. First, the analysis is done by applying an analysis section of a two-channel filter bank to each of the rows of the image, where the highpass filter is given by $H_1(z)$ and the lowpass filter is given by $H_0(z)$, resulting in two side-by-side images which are each half the width of the original and which have been compressed in the horizontal direction; one image is the output of the highpass filter and the other is the output of the lowpass filter. The columns of each of these outputs are then put into another analysis section of the filter bank, resulting in a block of four images, each of which is one-fourth the size of the original image and compressed in the horizontal and vertical directions. This block of four images represents the image after it has been filtered by a lowpass and a lowpass filter (LL), after a lowpass and a highpass filter (LH), after a highpass and a lowpass filter (HL), and after a highpass and a highpass filter (HH), each of which is referred to as a *subband.* This situation can be visualized as in Figure 2.2. It is possible to further decompose the transform by repeating the process on the LL output. The number of times that this is done is again referred to as the number of

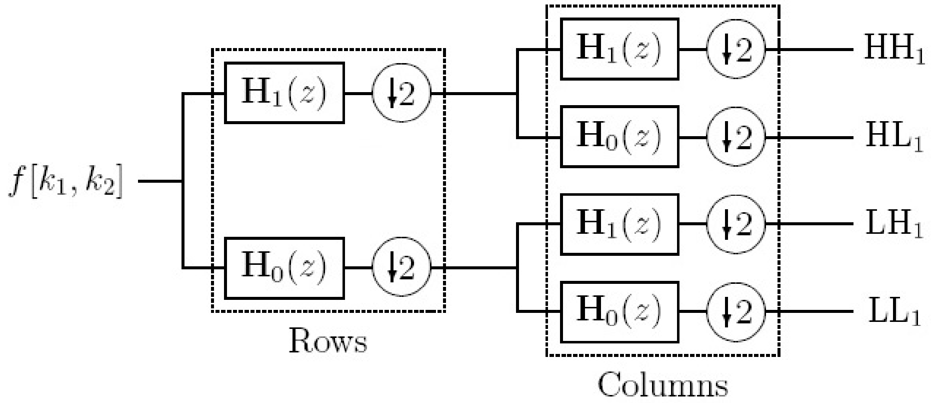


Figure 2.2: Analysis filter bank for 2D wavelet transform.

decomposition levels. Figure 2.3 shows the arrangement of the subbands for three decomposition levels.

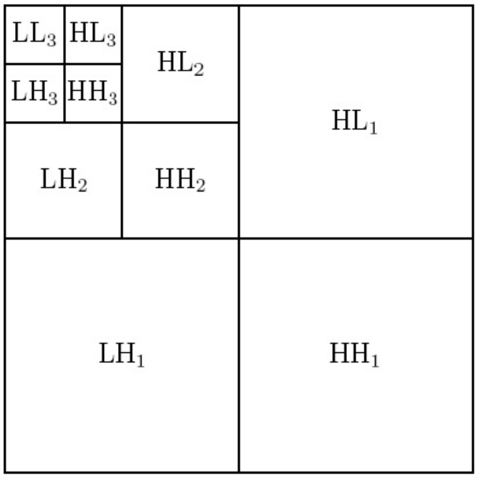


Figure 2.3: Visualization of wavelet domain after three decomposition levels.

In order to undo the transform, the inverse transform is performed, just as in the 1D case. This is accomplished by applying a synthesis filter bank to the subbands in reversed order that the subbands went through the analysis filter bank.

2.8 Wavelet Thresholding

An important feature of the wavelet transform is that, in the wavelet domain, the noise and signal have coefficients in different ranges [3, 4, 5]. The derivation of this for the two-dimensional case follows here. First let the original $N_1 \times N_2$ signal be represented by $s\left(n_1, n_2\right)$, the observed signal by $x\left(n_1, n_2\right)$, and the noisy signal by $w\left(n_1, n_2\right)$, which are related by

$$x\left(n_1, n_2\right) = s\left(n_1, n_2\right) + w\left(n_1, n_2\right) \tag{2.43}$$

Next, assume without loss of generality that N_1 is equal to N_2 with a value of N; in practice, if this is not the case, the signal can be zero-padded. Now, arrange the values of $s\left(n_1, n_2\right)$, $x\left(n_1, n_2\right)$, and $w\left(n_1, n_2\right)$ into N^2 vectors $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{w}$, respectively, so that Equation (2.43) becomes

$$\mathbf{x} = \mathbf{s} + \mathbf{w} \tag{2.44}$$

Let the forward wavelet transform with J decomposition levels be represented by $\mathcal{W}_J\left(\cdot\right)$ and the inverse wavelet transform with J decomposition levels by $\mathcal{W}_J^{-1}\left(\cdot\right)$. The wavelet transform of the observed signal will be denoted by $\mathbf{X}$, and the relationships between $\mathbf{x}$ and $\mathbf{X}$ are

$$\mathbf{X} = \mathcal{W}_J\left(\mathbf{x}\right) \tag{2.45}$$

$$\mathbf{x} = \mathcal{W}_J^{-1}\left(\mathbf{X}\right) \tag{2.46}$$

Using the wavelet transform, Equation (2.44) becomes

$$\mathbf{X} = \mathcal{W}_J\left(s\right) + \mathcal{W}_J\left(w\right) = \mathbf{S} + \mathbf{W} \tag{2.47}$$

Now, assume that the noise is additive white Gaussian with a variance equal to σ^2, which, due to the linearity of the wavelet transform, means that in the wavelet domain, it will still be additive, white Gaussian noise with a variance of σ^2. When a DWT is performed with one decomposition level, most of the energy will be concentrated in the

LL_1 subband and there will be little to no energy in the HH_1 subband. This energy compaction into the LL subband is because the highpass filter acts as a differentiator, and the lowpass filter behaves as an averaging function. Therefore, the energy will be concentrated in a few coefficients. As more decomposition levels are used, the number of coefficients becomes smaller and smaller. The high-energy coefficients will be of large magnitude and correspond to the signal and the noise; the rest of the coefficients will be small and correspond to the noise only.

The goal of de-noising techniques is to remove the noise from a signal while maintaining an undistorted original signal. In the wavelet domain, this can be accomplished by removing the coefficients from the noisy signal and keeping those from the signal and the noise. This operation is called *thresholding.* The goal is to find a level, or threshold, which separates the magnitude of the desired coefficients from that of the noisy coefficients. There are two primary methods of selecting the threshold value: a hard threshold and a soft threshold.

2.8.1 Hard Threshold vs. Soft Threshold

A hard threshold is the simplest way to perform thresholding. In choosing a value for a hard threshold, the goal is to determine a number below which most of the noisy coefficients lie and above which are most of the signal coefficients. This threshold is applied to every subband in the wavelet domain, except usually the LL_J subband, since this contains the coefficients with the most energy, causing no significant change due to thresholding. Each coefficient value is compared to the threshold value; if it is less than the threshold, then it is set to zero, and if it is greater than the threshold, it is left unchanged. When the inverse wavelet transform is applied, the only non-zero coefficients are those which contain information about the original signal, which implies that the majority of the noise has been removed.

A soft threshold is one which zeros out those coefficients below it and also shrinks the coefficients which are kept toward the threshold value. The choice of this value impacts the estimation of the original data and therefore must be chosen carefully, just as in the hard threshold selection. If the soft threshold operator is represented by $\eta_\tau^s(k)$, where τ is the threshold value and k is the wavelet coefficient that is being examined, then soft thresholding can be represented as

$$\eta_\tau^s(k) = \begin{cases} k\frac{|k|-\tau}{|k|} & |k| > \tau \\ 0 & \text{else} \end{cases} \tag{2.48}$$

2.8.2 Threshold Selection

There are two ways to apply the threshold to the coefficients: *globally* and *adaptively.* A global application means that one threshold is selected for all of the coefficients, except perhaps for those in the LL subband. The coefficients contained in the LL subband are generally large and therefore represent the signal plus noise rather than the noise alone. Applying the global threshold to this subband will have little, if any, effect, since most coefficients will be above the value.

A global threshold is attractive because it is simple to determine and to implement. However, it is not ideal since the subbands have different properties, causing the threshold to work well for some subbands and poorly for others [3]. A threshold which changes depending on the characteristics of the subband being examined is desired; such a threshold is referred to as an adaptive threshold, since it adapts itself to the particular area which is being thresholded. As a result of this adaptation, an adaptive threshold performs better than a global threshold.

Several methods for determining an adaptive threshold [6, 7, 20, 3] have been developed. One such method is known as the *BayesShrink threshold* [7], which was developed to work well on images, given certain characteristics of images. An important characteristic is that the wavelet coefficients of natural images in each of the HH, HL, and LH subbands, known as the *detail subbands*, have a generalized Gaussian distribution [7, 21, 22]. Therefore, the average MSE of each subband can be approximated by the Bayesian squared error risk. The BayesShrink threshold selection seeks to minimize this Bayesian risk. In addition to being suited to natural images, the BayesShrink threshold selection is also attractive because it is relatively simple to calculate and therefore does not add much computation time.

The BayesShrink threshold, τ_B, is shown in [7] to be calculated according to

$$\tau_B = \frac{\sigma_N^2}{\sigma_S} \tag{2.49}$$

where σ_N^2 is the variance of the noise of the image, which is either known or can be estimated, and σ_S is the standard deviation of the desired signal itself, which is unknown. Knowing that the noise is white Gaussian and that the observed signal is equal to the sum of the desired signal and the noise signal, assuming that X and N are uncorrelated, the standard deviation of the observed signal is equal to the sum of the standard deviation of the desired signal and the standard deviation of the noise signal; the standard deviation of the observed signal is given by

$$\sigma_X^2 = \sigma_S^2 + \sigma_N^2 \tag{2.50}$$

In this case, in order to solve Equation (2.49), only σ_X^2 must be known in order to solve

$$\sigma_S = \sqrt{\sigma_X^2 - \sigma_N^2} \tag{2.51}$$

As stated, in order to be adaptive, a different threshold must be calculated for each detail subband, which means that the variance must be calculated for the signal of each detail subband. An estimation technique for calculating the variance of the observed signal in each subband is therefore required. According to [18], this can be estimated by calculating

$$\hat{\sigma}_{X_{xx_j}}^2 = \frac{2^{2j}}{N^2} \sum_{i=1}^{\frac{N^2}{2^{2j}}} \left(\mathbf{X}_{xx_j}\left(i\right)\right)^2 \tag{2.52}$$

where $\mathbf{X}_{xx_j}\left(i\right)$ is the wavelet coefficient in the xx subband at decomposition level j.

Variance must be positive, in which case Equation (2.51) becomes

$$\hat{\sigma}_{S_{xx_j}} = \max\left(\sqrt{\hat{\sigma}^2_{X_{xx_j}} - \sigma^2_N}, 0\right) \tag{2.53}$$

Making the appropriate substitutions, the adaptive BayesShrink wavelet threshold can be found by calculating

$$\tau_{B,xx_j} = \begin{cases} \frac{\sigma^2_N}{\sqrt{\hat{\sigma}^2_{S_{xx_j}}}} & \hat{\sigma}^2_{S_{xx_j}} \neq 0 \\ 0 & \text{else} \end{cases} \tag{2.54}$$

CHAPTER 3

ESTIMATION ALGORITHM DEVELOPMENT

So far, a multichannel Wiener filter has been proposed as the best estimator for multichannel images because it optimally decorrelates the signal. However, there are several drawbacks in the implementation of the Wiener filter, such as required knowledge of second-order statistics and an inversion of a large matrix, which make it undesirable for use and which drive the search for a different estimation technique. This chapter puts forth such a technique based on developments in [10, 11, 13, 14] with a 2D DWT and IDWT.

3.1 Channel-Spatial KL Transform

An important first step in realizing a substitute for the multichannel Wiener filter is to show that the KL transform described in Chapter 2 can be split into a channel KL transform and a spatial KL transform. First, an assumption is made about the coherence between any two channels. Let c_{ij} be a scalar which represents the channel coherence between any two channels i and j and therefore will lie between 0 and 1. Then, taking the $N \times N$ correlation matrix $\mathbf{R}_{\mathbf{v}}$, the correlation between any two channels can be written as

$$\mathbf{R}_{\mathbf{v}_i\mathbf{v}_j} = c_{ij}\mathbf{R}_{\mathbf{v}} \tag{3.1}$$

Assuming that there are M channels, Equation (3.1) can be written in matrix form according to Equation (2.22) as

$$\mathbf{R_{vv}} = \begin{bmatrix} c_{11}\mathbf{R_v} & c_{12}\mathbf{R_v} & \cdots & c_{1M}\mathbf{R_v} \\ c_{21}\mathbf{R_v} & c_{22}\mathbf{R_v} & \cdots & c_{2M}\mathbf{R_v} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1}\mathbf{R_v} & c_{M2}\mathbf{R_v} & \cdots & c_{MM}\mathbf{R_v} \end{bmatrix} = \mathbf{C} \bigotimes \mathbf{R_v} \tag{3.2}$$

where $\mathbf{C}$ is an $M \times M$ matrix with entry (i, j) equal to c_{ij}; the symbol $\bigotimes$ represents the Kronecker product, which is defined as follows: if $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $p \times q$ matrix, then the Kronecker product

$$\mathbf{A} \bigotimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

The separability of the autocorrelation matrix into a matrix $\mathbf{C}$ and a matrix $\mathbf{R_v}$ is valid, as shown in [17]. The components of Equation (3.2) can be eigenexpanded as

$$\mathbf{C} = \mathbf{U_C \Lambda_C U_C^H} \tag{3.3}$$

$$\mathbf{R_v} = \mathbf{U_{R_v} \Lambda_{R_v} U_{R_v}^H} \tag{3.4}$$

In this case, $\mathbf{U_C^H}$ decorrelates the signal across the channels, and $\mathbf{U_{R_v}^H}$ decorrelates the signal in space. $\mathbf{U_C^H}$ is therefore called the *channel KL transform*, and $\mathbf{U_{R_v}^H}$ is called the *spatial KL transform*. If a property of Kronecker products is applied as shown in [23], the channel-spatial KL transform $\mathbf{U_v^H}$ can be written as

$$\mathbf{U_v^H} = \mathbf{U_C^H} \bigotimes \mathbf{U_{R_v}^H} \tag{3.5}$$

3.2 Substitution for Channel KL Transform

In many cases, because the channel coherence matrix $\mathbf{C}$ is not readily known and cannot be easily estimated, the channel KL transform $\mathbf{U}_{\mathbf{C}}^{H}$ cannot be simply calculated. An approximation of the channel KL transform is therefore sought.

In looking for a substitute for the channel KL transform, the goal is to find a transform which decorrelates the signal across the channels. Distance between channels of a video sequence, for example, is defined by the time between the frames, or the frequency. When the distance between channels is small, meaning that the correlation between channels is high, then $\mathbf{C}$ is a Toeplitz symmetric matrix. In order to diagonalize a Toeplitz symmetric matrix, which optimally decorrelates the data, a DFT can be applied. Therefore, the channel KL transform can be replaced by a normalized $M \times M$ DFT matrix $\mathbf{F}_M/\sqrt{M}$, which is attractive since the exact channel coherence matrix need not be known and because fast methods of implementation, such as the fast Fourier transform (FFT), have been developed [24].

3.3 Substitution for Spatial KL Transform

Now a replacement for the spatial KL transform is sought. As was brought forth in Chapter 2, the 2D DWT does a good job of distinguishing between pure noise and the signal plus noise in the wavelet domain. As such, it is a relatively good approximation of the spatial KL transform, which also decorrelates the signal with respect to space. The 2D DWT can therefore be used in place of the KL transform. In addition, instead of weighting the coefficients, as is the case when the spatial KL transform is used, an adaptive BayesShrink thresholding of the coefficients can be done instead. This will allow the coefficients containing signal information to be retained and will also remove the coefficients corresponding to noise.

The 2D DWT is a desirable substitution to make since, unlike the KL transform, it does not require knowledge of the second-order statistics of the noise or the signal. In addition, it is comparatively simpler to implement since it does not require the inversion of a large matrix.

3.4 Noise Which Is Not Additive White Gaussian

An important issue not yet addressed is the noise which is present. In the presentation of the Wiener filter and the DWT, it was assumed the noise was additive white Gaussian noise. In reality, as stated earlier, in many cases, and in particular those utilizing a CCD camera, the noise will instead have a Poisson distribution. It is necessary to determine whether the substitution of a DFT and a 2D DWT in place of a 3D KL transform will perform well in the presence of Poisson noise, and to make any changes which will enhance their performance.

If the noise is non-Gaussian, the decorrelating transform may be influenced by the noise, making the decorrelation less than optimal. In this case, it is necessary to take a step to *whiten* the noise, that is, making the noise appear as white noise instead. White noise has a flat power spectral density, which means that the power in all frequency bands is equal; also, its autocorrelation function is a delta function over the appropriate spatial dimension.

Whitening is accomplished by examining a small neighborhood of pixels, whose indices are contained in $\mathcal{L}_s(i)$, around the pixel of interest, $s[i]$. Then, the first and second moments for the original data can be estimated by calculating

$$\bar{s}\,[i] = \textstyle\sum_{k\in\mathcal{L}_s(i)} w\,(i-k)\,x\,[k] \tag{3.6}$$

$$\bar{s}^2\,[i] = \textstyle\sum_{k\in\mathcal{L}_s(i)} w\,(i-k)\,(x\,[k])^2 \tag{3.7}$$

where, for all i, $w\,(i-k)$ is a weighting function which has the property

$$\sum_{k\in\mathcal{L}_x(i)} w\,(i-k) = 1 \tag{3.8}$$

Then, the local noise variance for Poisson noise, p, can be calculated as

$$\hat{\sigma}_p\left[i\right]^2 = \max\left(\bar{s}\left[i\right], 0\right) \tag{3.9}$$

and for Gaussian noise, p as

$$\hat{\sigma}_p\left[i\right]^2 = \max\left(\bar{s}^2\left[i\right] - \left(\bar{s}\left[i\right]\right)^2, 0\right) \tag{3.10}$$

Next, normalizing the observed data values by the estimated standard deviation of the noise converts the noise into approximately white Gaussian noise with unit variance. For the Poisson case, this is

$$\frac{y\left[i\right]}{\hat{\sigma}_p\left[i\right]} \approx \mathcal{N}\left(\sqrt{s\left[i\right]}, 1\right) \tag{3.11}$$

and for the Gaussian case

$$\frac{y\left[i\right]}{\hat{\sigma}_p\left[i\right]} \approx \mathcal{N}\left(\frac{s\left[i\right]}{\hat{\sigma}_p}, 1\right) \tag{3.12}$$

both are approximate because the local noise variance is calculated based on estimates of the first and second moments. This converted data can then be used in the estimation technique since its noise is approximately white Gaussian noise; the result must be multiplied by a factor of $\hat{\sigma}_p$ in order to cancel the normalization.

3.5 Presentation of Algorithm

Using the preceding sections, an algorithm for estimating an original 2D multichannel data set from its noisy observations is now outlined concretely:

1. Estimate noise variance for each pixel by using a neighborhood around it.

2. Normalize each pixel value by the standard deviation found before.

3. Decorrelate data across the channels using DFT.

4. For each channel, decorrelate data in space using 2D DWT.

5. For each channel, remove noisy coefficients using BayesShrink adaptive wavelet thresholding.

6. For each channel, recorrelate data in space using 2D IDWT.

7. Recorrelate data across channel using IDFT.

CHAPTER 4

BEST FILTER AND NOISE MODEL

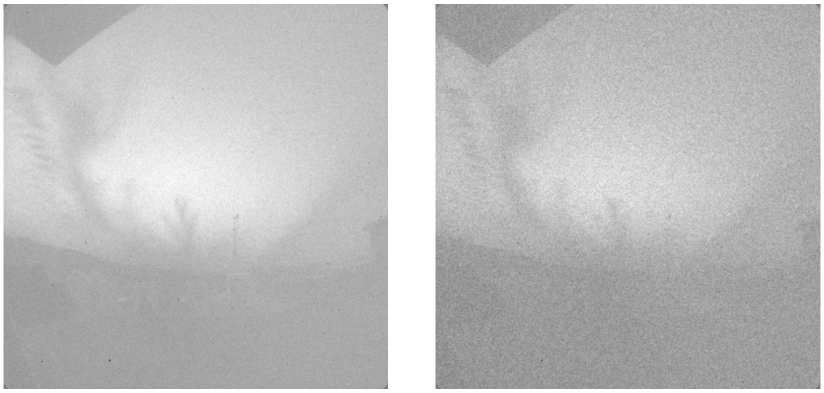
4.1 Biorthogonal Filter Bank

In this thesis, in contrast to what is done in previous work regarding de-noising multichannel images, biorthogonal filter banks are used instead of orthogonal filter banks. Because of this choice, the performance of de-noising using biorthogonal filter banks on two-dimensional data needs to be assessed to assure that it will perform well when used in the algorithm given in the preceding chapter.

Biorthogonal filter banks tend to perform better when two-dimensional signals such as images are of interest. This improved performance is because its filters are all linear phase, meaning that the frequency components are all delayed by the same amount. Also, previous work has shown that for images with low frequency, biorthogonal wavelet filters perform better than orthogonal filters due to better energy compaction [12]. The performance of an orthogonal filter bank, which was used in the previous application [13] of the algorithm presented, compared to the performance of the biorthogonal filter, which is presented in this thesis, needs to be assessed. To do this, a single frame from a video sequence created by a video camera is selected. Then, additive white Gaussian noise is added, and the SNR of the noisy image, calculated according to

$$\mathrm{SNR}\left(\mathbf{v}, \hat{\mathbf{v}}\right) = 10\,\log_{10}\left(\frac{\mathbf{v}^T\mathbf{v}}{\left(\mathbf{v}-\hat{\mathbf{v}}\right)^T\left(\mathbf{v}-\hat{\mathbf{v}}\right)}\right) \tag{4.1}$$

where $\mathbf{v}$ is the original image and $\hat{\mathbf{v}}$ is the noisy image, is 25.62 dB. Both the original and the noisy images can be seen in Figure 4.1.



(a) Original data.

(b) Data with additive white Gaussian noise.

Figure 4.1: Original and noisy data.

Then, the DWT is applied to the single frame of noisy data. Two different filter banks, an orthogonal and a biorthogonal, are used. The orthogonal has been used in the past, so the performance of the biorthogonal filter bank is of greater interest here. In determining the best quadrature mirror filter (QMF) and dual quadrature mirror filter (DQMF), several are examined. Four options available for the QMF and the DQMF are triangle, Deslauriers, average-interpolating, and Villasenor filters. After creating filters of all types, it was found that the Villasenor filter yields the most SNR improvement. Five different sets of filter coefficients for the QMF and DQMF, as described in [25], were used in this thesis. For the first, the filters are given by:

$$\mathrm{QMF} = \begin{bmatrix} .037828 \\ -.023849 \\ -.110624 \\ .377403 \\ .852699 \\ .377403 \\ -.110624 \\ -.023849 \\ .037828 \end{bmatrix}^T \quad \mathrm{DQMF} = \begin{bmatrix} -.064539 \\ -.040689 \\ .418092 \\ .788486 \\ .418092 \\ -.040689 \\ -.064539 \end{bmatrix}^T$$

The second contains the following filters:

$$\mathrm{QMF} = \begin{bmatrix} -.008473 \\ .003759 \\ .047282 \\ -.033475 \\ -.068867 \\ .383269 \\ .767245 \\ .383269 \\ -.068867 \\ -.033475 \\ .047282 \\ .003759 \\ -.008473 \end{bmatrix}^T \quad \mathrm{DQMF} = \begin{bmatrix} .014182 \\ .006292 \\ -.108737 \\ -.069163 \\ .448109 \\ .832848 \\ .448109 \\ -.069163 \\ -.108737 \\ .006292 \\ .014182 \end{bmatrix}^T$$

The third is comprised of

$$\mathrm{QMF} = \begin{bmatrix} 0 \\ -.129078 \\ .047699 \\ .788486 \\ .788486 \\ .047699 \\ -.129078 \end{bmatrix}^T \quad \mathrm{DQMF} = \begin{bmatrix} 0 \\ .018914 \\ .006989 \\ -.067237 \\ .133389 \\ .615051 \\ .615051 \\ .133389 \\ -.067237 \\ .006989 \\ .018914 \end{bmatrix}^T$$

The fourth contains

$$\mathrm{QMF} = \tfrac{1}{4\sqrt{2}} \left[\begin{array}{ccccc} -1 & 2 & 6 & 2 & -1 \end{array} \right]$$
$$\mathrm{DQMF} = \tfrac{1}{2\sqrt{2}} \left[\begin{array}{ccc} 1 & 2 & 1 \end{array} \right]$$

The fifth pair of filters set forth in the Villasenor model is

$$\mathrm{QMF} = \tfrac{1}{\sqrt{2}} \left[\begin{array}{ccc} 0 & 1 & 1 \end{array} \right]$$
$$\mathrm{DQMF} = \tfrac{1}{8\sqrt{2}} \left[\begin{array}{ccccccc} 0 & -1 & 1 & 8 & 8 & 1 & -1 \end{array} \right]$$

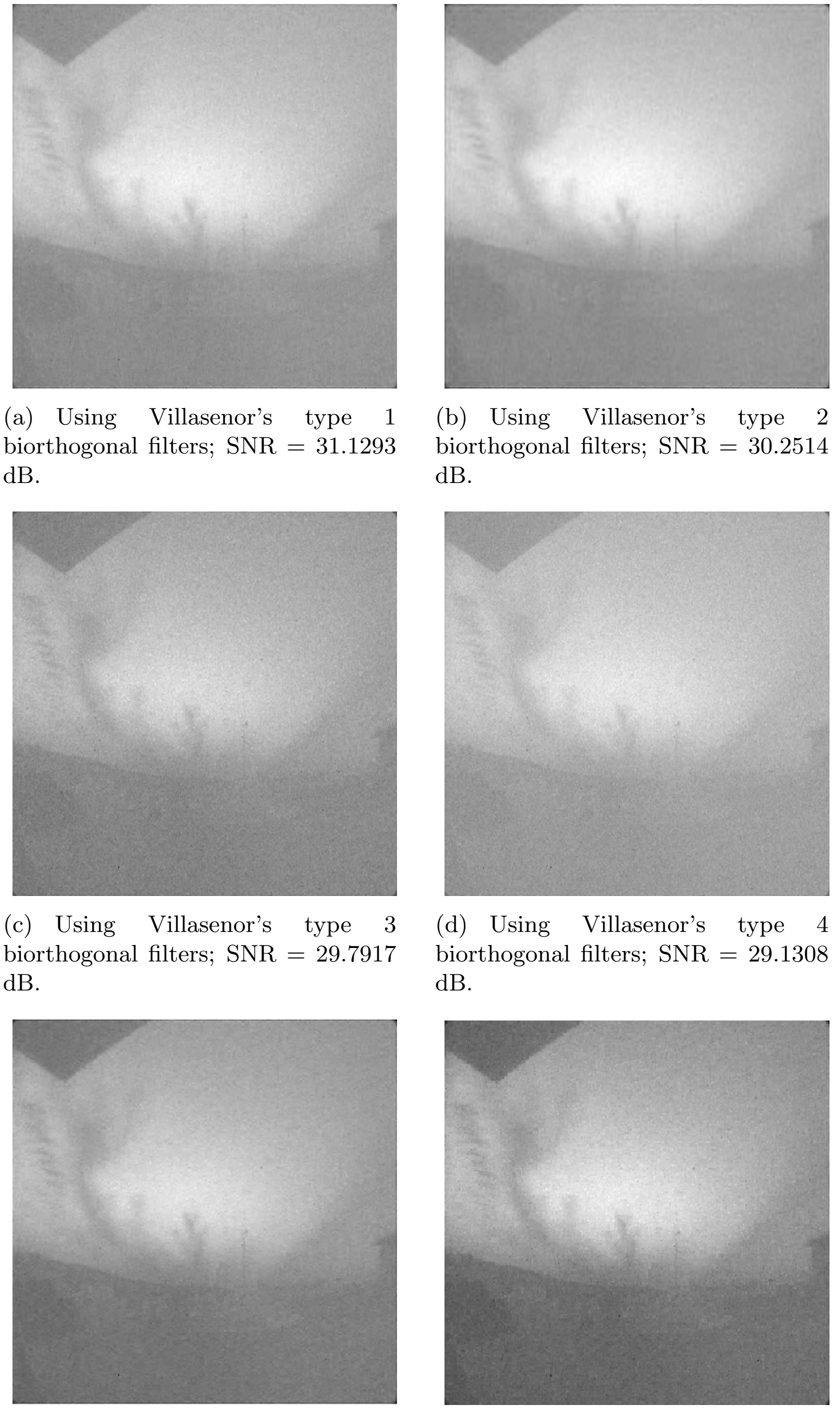
A DWT is performed with each of these filter banks and also with an orthogonal filter bank, as has been done in the past. Then, adaptive thresholding using the BayesShrink method is applied. Finally, the IDWT is done, yielding de-noised versions of the noisy data. The visual results can be seen in Figure 4.2.

The SNR values for the de-noised images are given in Table 4.1. Each filter bank provides an improvement over the noisy image, which, as stated, has an SNR value of 25.62 dB. The type 1 Villasenor filter selection gives a higher SNR than the orthogonal filter bank, and the rest of the biorthogonal filters give lower SNR values. Based solely on the SNR, the biorthogonal can be said to perform just as well as the orthogonal filter bank used heretofore.

Table 4.1: SNR values (in dB) for the six de-noised single frame images created with the different filters.

Filter Bank	SNR (dB)
Type 1 Villasenor	31.1293
Type 2 Villasenor	30.2514
Type 3 Villasenor	29.7917
Type 4 Villasenor	29.1308
Type 5 Villasenor	30.6823
Orthogonal	30.9741

Examining the images in Figure 4.2 from a subjective visual standpoint, all of the filters, biorthogonal and orthogonal, do a good job of bringing out the structure in the sky region



(a) Using Villasenor's type 1 biorthogonal filters; SNR = 31.1293 dB.

(b) Using Villasenor's type 2 biorthogonal filters; SNR = 30.2514 dB.

(c) Using Villasenor's type 3 biorthogonal filters; SNR = 29.7917 dB.

(d) Using Villasenor's type 4 biorthogonal filters; SNR = 29.1308 dB.

(e) Using Villasenor's type 5 biorthogonal filters; SNR = 30.6823 dB.

(f) Using orthogonal filters; SNR = 30.9741 dB.

Figure 4.2: De-noised data using five different biorthogonal filter banks and an orthogonal filter bank.

of the image. In the noisy image, these are somewhat smeared and the edge-detail is lost, particularly in the left side of the image, where in the original, there are many horizontal regions. These are brought back out in the de-noised versions. Types 3 and 4 visually perform the worst, with more detail still lost and noise still very visible. Types 1 and 2 appear to do the best of all the biorthogonal filters, as type 5 is still somewhat grainy due to noise. The image de-noised using an orthogonal filter bank is darker than those created using biorthogonal filter banks and still is quite noisy. Therefore, it can be said that, at least visually, biorthogonal filter banks with Villasenor's type 1 and 2 QMF and DQMF filters outperform orthogonal filter banks when applied to images degraded by additive white Gaussian noise.

Because the 2D DWT is performed on each frame after the video sequence is decorrelated by the channel KL transform, applying a 2D DWT using a biorthogonal filter bank as shown here should work as well as or better than when an orthogonal filter bank is used, as was done previously. This assumes, of course, that if the noise is Poisson instead of additive white Gaussian, as was the case in the preceding discussion, then the whitening process is effective.

4.2 Noise Model

The other difference between what is accomplished in this work and what has been done previously is that video data is being examined, whereas previously the methodology was applied to hyperspectral data or fMRI data, usually with additive white Gaussian noise. In video, the data is likely to be degraded by Poisson noise instead of Gaussian noise. Section 3.4 presented a method for whitening the noise in a sequence containing Poisson noise. The applicability of this method to actual data must be examined.

This whitening method was put forth in [10], where it was applied to hyperspectral data with Poisson noise. The quality of the method was examined in terms of the cumulative distribution functions (CDFs) of a Poisson distribution and a Gaussian distribution. A

Poisson distribution has a CDF given by

$$\frac{\Gamma\left(\lfloor t+1 \rfloor, s[i]\right)}{\lfloor t \rfloor !} \tag{4.2}$$

where Γ is the Gamma function and ! is the factorial function. As stated earlier, it is possible to map a Poisson distribution to a Gaussian distribution with unit variance. This mapping is possible because the resulting observation for values of interest, $s\left[i\right]$, which are large, is an approximation of a Gaussian distribution with mean and variance equal to $s\left[i\right]$ so that $y\left[i\right] \sim \mathcal{N}\left(s\left[i\right], s\left[i\right]\right)$. A Gaussian distribution has a CDF given by

$$\int_{-\infty}^{t} \frac{1}{\sqrt{2\pi s\left[i\right]}} e^{-\frac{(t-s[i])^2}{2s[i]}} \tag{4.3}$$

The validity of the distribution given above can be found by taking the maximum of the absolute difference between the CDF given in Equation(4.2) and Equation (4.3), so that

$$\text{Maximum CDF Error} = \max_{t>0} \left| \frac{\Gamma\left(\lfloor t+1 \rfloor, s[i]\right)}{\lfloor t \rfloor !} - \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi s\left[i\right]}} e^{-\frac{(t-s[i])^2}{2s[i]}} \right| \tag{4.4}$$

For values of $s\left[i\right]$ above 20, this maximum error is below 0.05, and as $s\left[i\right]$ gets larger, the maximum error approaches zero. Therefore it can be said that the approximation of $y\left[i\right] \sim \mathcal{N}\left(s\left[i\right], s\left[i\right]\right)$ is accurate and appropriate.

As previously described, a CCD counts electrons in order to determine a pixel value. This means that a longer exposure time will generate more photons incident on the CCD array. The data used in this thesis were 630-nm emissions captured by a high-grade scientific narrow-field CCD which uses an exposure rate between 1 and 2 minutes. In [26], the number of photons emitted per cubic centimeter per second is calculated for 630-nm emissions. The result is that for an altitude between 200 and 300 km, between 2 and 8 photons per cm^3 per second are emitted. Assuming a 60-second exposure time, there will be between 120 and 480 photons per cm^3 emitted by airglow. The CCD camera used to collect the data in this thesis is similar to that used in [27], where the dark count after the camera was cooled to -35 to -40°C was $\sim$0.5 counts per second per pixel. Therefore, in a 60-second exposure, approximately 30 counts per pixel will be detected in the absence of light. The number of electrons counted for every pixel when detecting airglow is more than the dark count and is more than 20, yielding a low CDF error, meaning that the whitening process is appropriate for this data. If, on the other hand, an image was of low-contrast and dark or captured using a very fast shutter time, the values may all be small and therefore yield a larger error.

CHAPTER 5

EXPERIMENTAL RESULTS

In order to test the algorithm developed in Chapter 3, it is applied to two different video sequences, one of which is synthetic and the other from an actual CCD camera. First, the data is combined with Poisson noise. Then the estimation technique put forth in the algorithm above is applied, using a Daubechies wavelet of length-6, an adaptive threshold, a DFT to approximate the channel KL transform, and either an orthogonal or a biorthogonal filter bank. Based on the results of the biorthogonal filters applied to single images given in Chapter 4, only Villasenor's type 1, 2, and 5 are used, as these increased the SNR more than the others. After recorrelation, the performance is measured by calculating the SNR of the data in each frame before and after applying the estimation scheme. SNR is calculated according to Equation 4.1. In addition, the MSE per frame is also calculated for the noisy data and the estimated data, and visual observations are also made.

5.1 Notes on Data Used in Algorithm

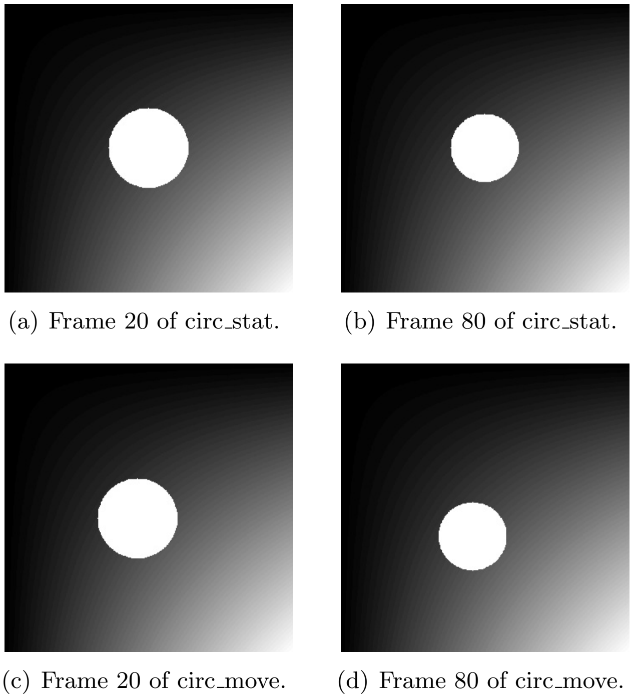
Two different video sequences were used in the algorithm: a synthetically created video as well as an actual video obtained by a camera. As mentioned previously, CCDs employ a counting method to generate pixel values, which leads to noise with a Poisson distribution. Because of this additional noise, the original data captured by a camera may look degraded prior to the Poisson distribution being applied, and the final estimation may actually be better visually (a subjective metric) but have a lower SNR. In order to remove the Poisson distribution due to the CCD, an artificial video was developed. To better understand the reasoning behind how the synthetic video sequence was generated, a discussion of the actual video is required.

The actual data examined are views of the airglow in the ionosphere above the earth's surface from a ground-based camera. Airglow is a very weak emission of light by the earth's ionized atmosphere. It causes the night sky never to be completely dark, and is most visible about ten degrees off the horizon. This phenomenon is visible to the naked eye at high altitudes. The study and characterization of the airglow is important because it makes it possible to detect structures that may have adverse effects on radio communications which traverse the ionosphere. The camera from which the data used in this thesis was obtained was mounted on the ground near the equator and aimed slightly above the horizon, as in [28, 27] and others. The foreground, consisting of objects on the ground (buildings, cars, and the ground) remains constant throughout the video, while the sky above and in the background is constantly shifting. The airglow appears as elongated structures and moves throughout the video. It is also important to note that the video is a grayscale, quantized to range in intensity from 0 to 255.

To counteract the noisy nature of the original video, a video is synthesized which intends to mimic the motion of the airglow in the actual video. The results of the algorithm on the synthetic data and the actual data are presented in the next two sections.

5.2 Synthetic Data Results

In order to mimic the behavior of the airglow in the actual data, a gradient background with intensity varying from 0 to 255 is created. A circle is placed on top, with a radius that grows and contracts throughout 100 frames; this sequence is represented as *circ_stat*. Another sequence is created with the same gradient background and circle with varying radius, but this time the center of the circle moves around the frames; this sequence is represented as *circ_move*. Frames 20 and 80 from each sequence are shown in Figure 5.1. To begin, a Poisson distribution with a standard deviation of 20 is applied to the sequence. The corresponding frames with noise are shown in Figure 5.2. It may be hard to see, but the noise has smoothed the background, which was somewhat striated originally, and has


(a) Frame 20 of circ_stat.

(b) Frame 80 of circ_stat.

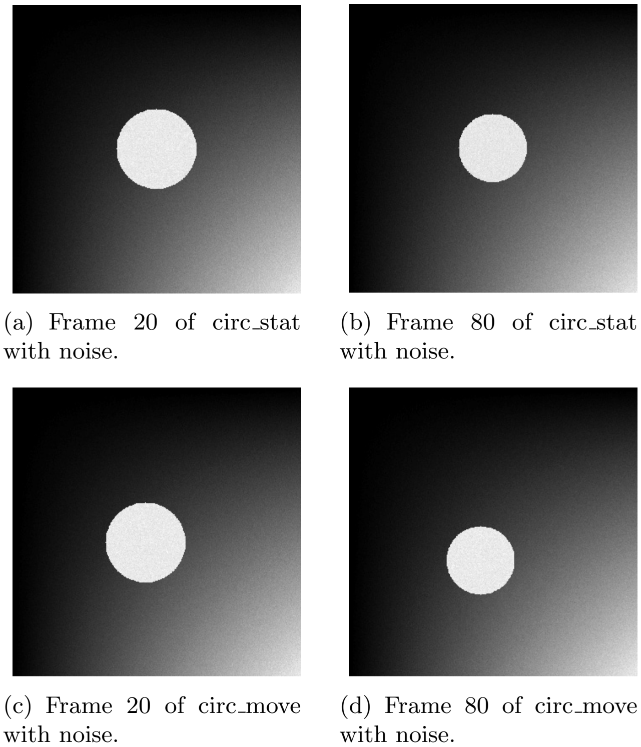
(c) Frame 20 of circ_move.

(d) Frame 80 of circ_move.

Figure 5.1: Frames from original synthetic data sequences.

also made the uniformly white circle non-uniform. In order to perform the estimation, two different wavelet filter banks are used. The first is an orthogonal filter bank, as was used in the estimation technique in [11]. The second is a biorthogonal filter bank with filter coefficients given in Villasenor's best five biorthogonal filters [25]; in particular, the second set of filter coefficients is used since it yields the best SNR improvement when compared to the other four. Statistics of the noisy data is also calculated locally, as described in the previous section on whitening the noise, since the noise is Poisson instead of additive Gaussian. The results of the orthogonal filter are shown in Figure 5.3, and the results of the biorthogonal filter are shown in Figure 5.4.

From the visual results it can be seen that the orthogonal filter does a good job of removing the noise from the data. There is a reduction of noise in the background and also in the circles themselves, although the circles are not returned to their original uniform intensity. It also appears that the circle is better estimated in the center, instead of the edges, where there is still a larger amount of noise. There is not a very noticeable difference between the stationary and moving data; both appear to perform well with an

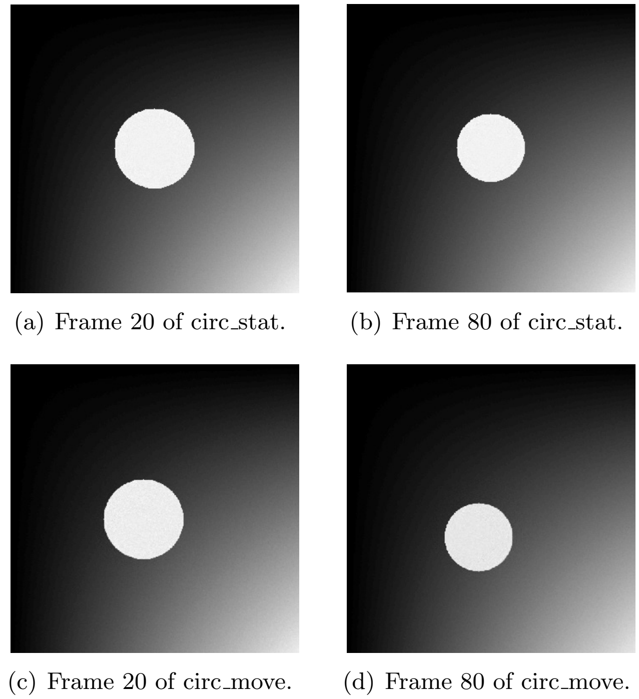

(a) Frame 20 of circ_stat with noise.

(b) Frame 80 of circ_stat with noise.

(c) Frame 20 of circ_move with noise.

(d) Frame 80 of circ_move with noise.

Figure 5.2: Frames from synthetic data sequences with Poisson noise with a standard deviation of 20.

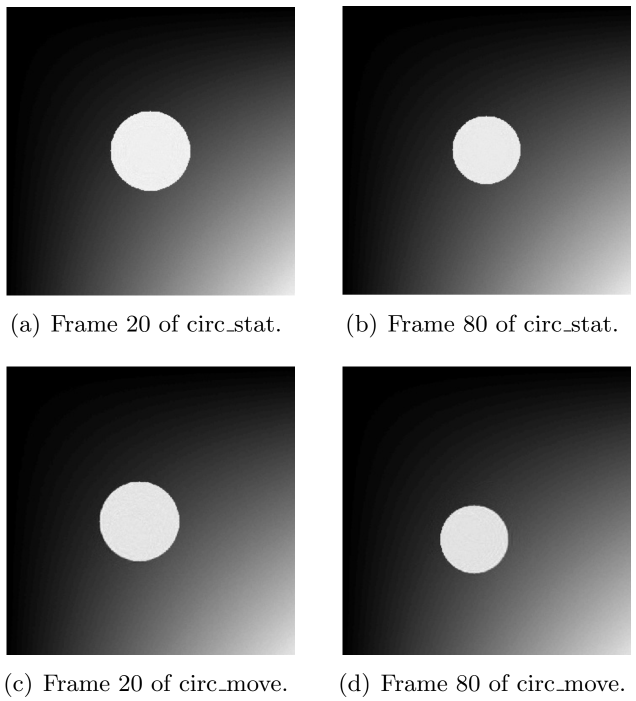

(a) Frame 20 of circ_stat.

(b) Frame 80 of circ_stat.

(c) Frame 20 of circ_move.

(d) Frame 80 of circ_move.

Figure 5.3: Frames from synthetic data sequences after estimation with an orthogonal filter.



(a) Frame 20 of circ_stat. (b) Frame 80 of circ_stat.

(c) Frame 20 of circ_move. (d) Frame 80 of circ_move.

Figure 5.4: Frames from synthetic data sequences after estimation with biorthogonal filter.

orthogonal filter.

The biorthogonal filter seems to do a better job of returning the striation to the background than the orthogonal filter does. However, it seems to suffer with regards to the circles. Again, there is more improvement in the center of the circles than around the edges. The edges still contain noise, and instead of smoothly decreasing toward the center of the circles, the noise decreases in concentric rings. Again, there is not much difference between the stationary and the moving data.

The SNR per frame for the noisy, orthogonal, and biorthogonal data is calculated for the stationary and the moving synthetic data and displayed in the graphs given in Figure 5.5 and Figure 5.6.

One thing to note is that the SNR for the noisy data of both sets increases, decreases, and then increases again. It changes direction at the same frames for both sequences. These points are where the radius of the circle changes from increasing to decreasing, and vice versa. In the estimated case with the orthogonal filter bank, these breaks are seen as spikes of much lower SNR, and the effect is more exaggerated in the case of the

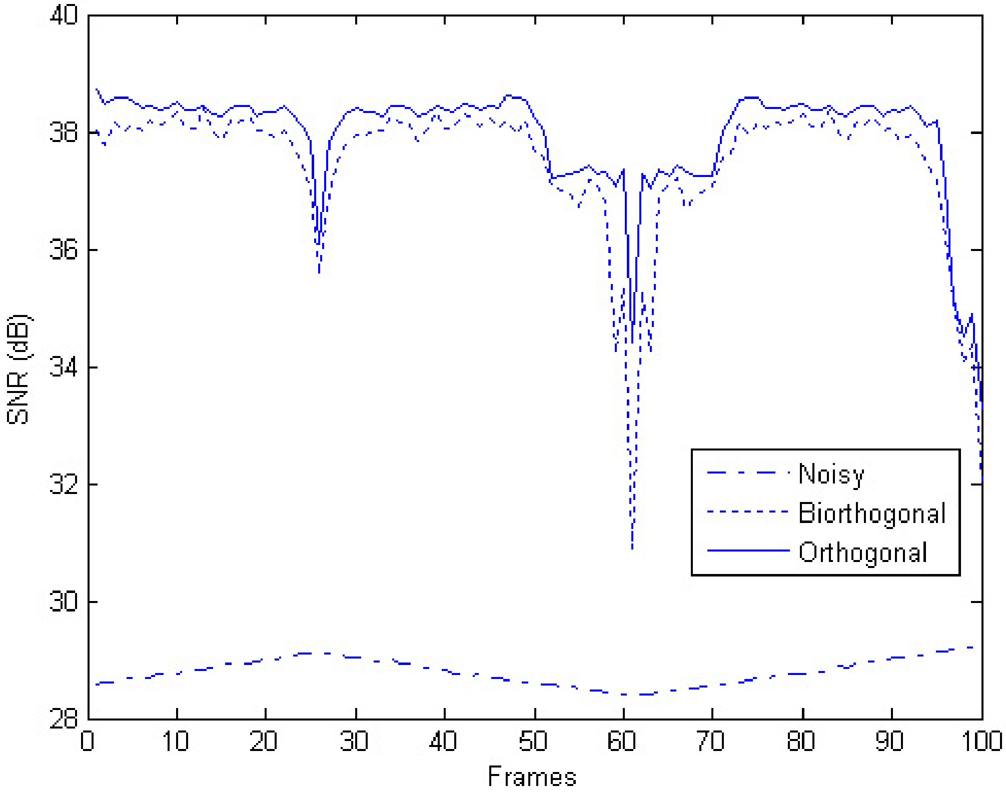


Figure 5.5: SNR per frame for circ_stat.

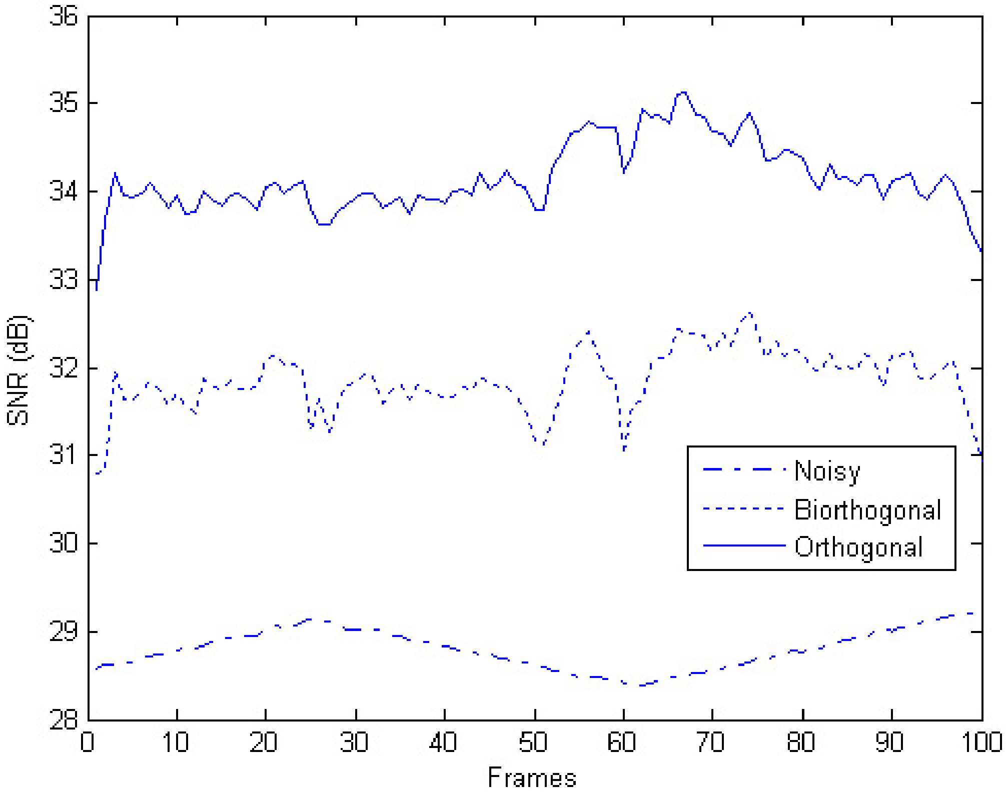


Figure 5.6: SNR per frame for circ_move.

biorthogonal filter. The estimation technique seems to have problems when there is a great difference between successive frames. Regardless, the SNR has been increased for both the stationary and moving data from around 29 dB to 38 dB. The orthogonal filter performs better in the stationary case.

In the case of the moving data, the SNR of the noisy sequence is a little higher than in the stationary case, but still follows the same pattern. For the estimation with an orthogonal filter, the SNR improves to around 34 dB, and for the estimation with a biorthogonal filter, it improves to 32 dB. There are no spikes of lower SNR when the radius changes direction; instead, the SNR remains fairly constant throughout the frames. It is safe to assume that if there had been a change in direction which accompanied the change in radius, the SNR would have decreased as a result of the higher frequency associated with such a change.

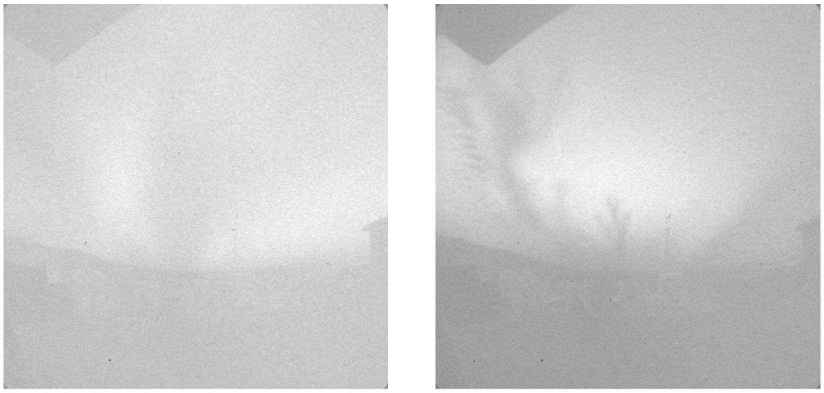
5.3 Camera Airglow Data Results

Another video sequence, this one taken from an actual CCD camera, is examined. The camera remains focused in the same direction throughout the entire sequence. The result is that the features in the foreground, which include buildings, vehicles, and an antenna, remain stationary and constant during the video. The atmosphere is the focus of the images because it changes during the time that the camera is collecting data.

The first step in this sequence is to scale the data intensity contained in the frames to be from 0 to 255 in order to compare the results with those of the synthetic data. The video contains 151 frames. Of those, frames 50 and 100 are chosen as examples of the whole sequence. One thing to note is that the intensity of the entire sequence changes throughout. It gets darker and lighter, as can be seen in the two frames that are chosen to represent the rest of the data. The original scaled data is shown in Figure 5.7.

Next, just like the synthetic data, the sequence is made noisy by Poisson noise with a standard deviation of 20, resulting in the frames shown in Figure 5.8.

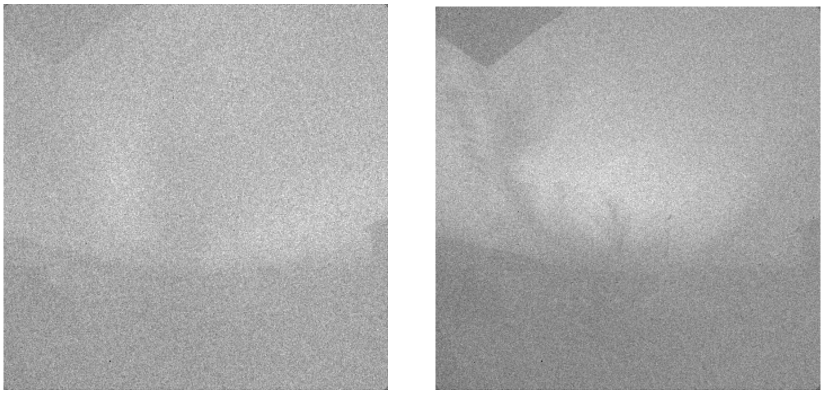
The final step is to use the estimation technique prescribed in this work to remove the



(a) Frame 50.

(b) Frame 100.

Figure 5.7: Frames from original actual camera data.

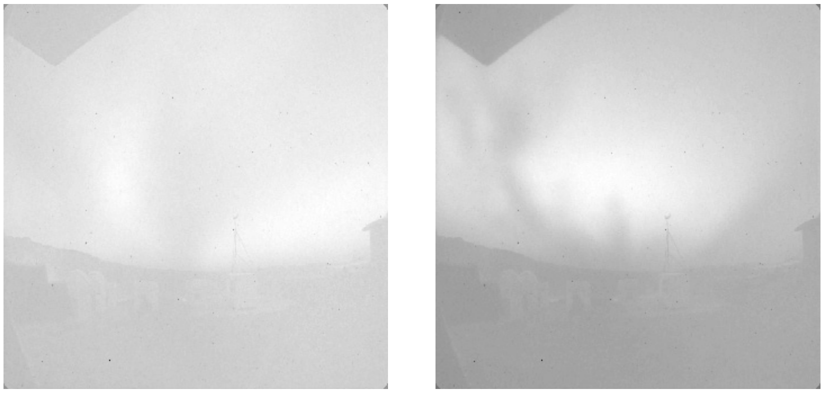


(a) Frame 50.

(b) Frame 100.

Figure 5.8: Frames from actual camera data with Poisson noise with a standard deviation of 20.

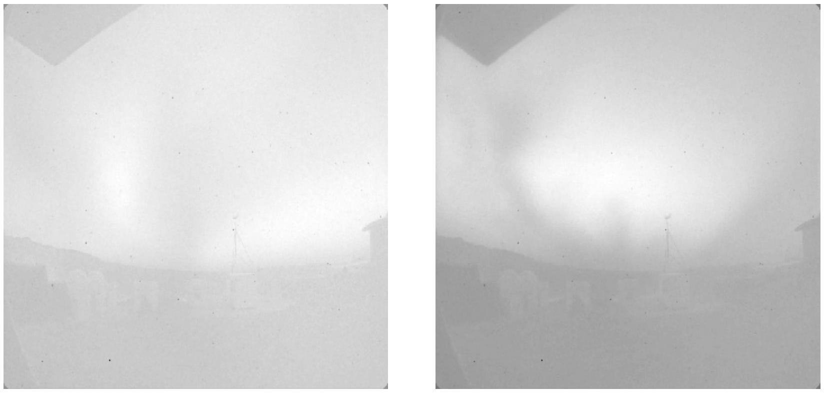
noise and approximate the original. Again this is done with an orthogonal and a biorthogonal filter bank, as shown in Figure 5.9 and Figure 5.10, respectively.



(a) Frame 50.

(b) Frame 100.

Figure 5.9: Frames from air sequence after estimation using an orthogonal wavelet filter bank.



(a) Frame 50.

(b) Frame 100.

Figure 5.10: Frames from air sequence after estimation using a biorthogonal wavelet filter bank.

The original data in this case is not very clear. It seems to already contain noise. Adding noise with the same standard deviation as in the synthetic data produces a much worse video sequence. Hardly any details can be made out in either the original or the noisy frames. After the estimation technique is applied, the images are much clearer and crisper. The foreground is now made up of different buildings, whereas previously it looked to be of almost uniform intensity. The sky is also clearer. Some of the details of the atmospheric structures, however, have been lost, which can particularly be seen in frame 100. Above the horizon toward the left of the image, there are multiple almost horizontal stripes stacked on top of each other in the original image. In the estimated image, both with an orthogonal and a biorthogonal filter bank, these are smoothed over and the detail is lost. The orthogonal filter does a slightly better job of preserving the original data than the biorthogonal filter. Because the foreground is not moving, both estimation techniques do a good job of bringing out the features; in the images above the horizon, however, things are not stationary and therefore the technique performs more poorly. As the focus of the study is on the airglow, the technique is not ideal since these features are lost.

Now the estimated SNR per frame of the noisy, orthogonal, and biorthogonal multichannel images is calculated for the airglow data and displayed in the graph in Figure 5.11.

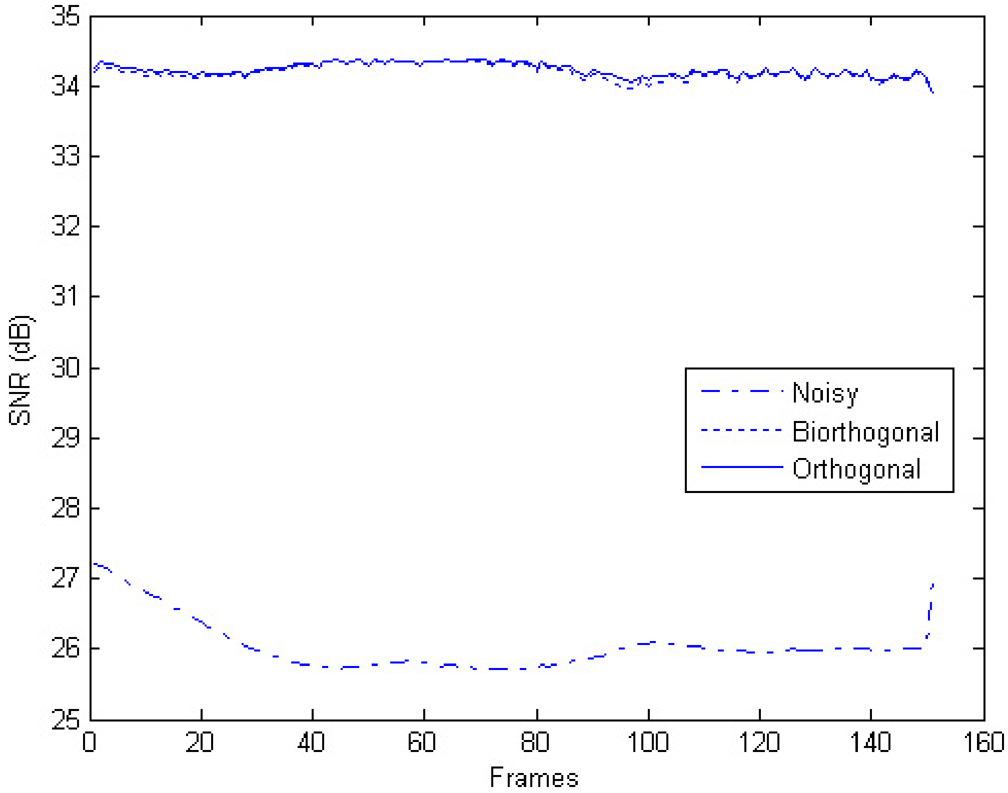


Figure 5.11: SNR per frame for air data.

The estimated SNR of the noisy data remains around 26 dB throughout the frames. The estimation data using both the orthogonal and biorthogonal filter banks have an SNR of a little more than 34 dB. Both filters perform essentially the same, with the orthogonal filter yielding slightly higher SNR throughout. This 8 dB improvement may be slightly skewed, since the SNR is computed with respect to the original data which contains noise since it is taken from an actual CCD source; the estimated data may in fact therefore be better visually than the original.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis provides a method of reducing the noise in video sequences. CCD cameras, which are commonly used to capture a video sequence, create data with a Poisson distribution because of the counting method of collection. Removal of this noise without any direct knowledge of the original data is desired. A multichannel Wiener filter is one estimator to use. However, implementation of a multichannel Wiener filter requires knowledge of the second-order statistics of the signal and noise, and also requires the inversion of a large matrix which gets larger with a longer video sequence. Therefore, an alternative which approximates the performance of the Wiener filter is sought.

The Wiener filter, which employs a KL transform, can be separated into a channel KL transform and a spatial KL transform, which optimally decorrelate the signal across the channels and in space, respectively. The channel KL transform can be approximated by a DFT, and the spatial KL transform can be approximated by a 2D DWT. Instead of weighting the coefficients as is done in the KL transform, wavelet thresholding can be performed. Coefficients in the wavelet domain represent either the signal plus the noise (large coefficients) or the noise only (small coefficients). Thresholding keeps those coefficients corresponding to the signal portion and removes those coefficients corresponding to the noise only. A successful thresholding technique is an adaptive threshold which is chosen according to the BayesShrink method. In order to get back the original data, the inverse 2D DWT is applied to the thresholded data, and then the inverse DFT is applied to recorrelate the data. What is left is data which is less noisy and more accurately reflects the original, undistorted video data.

This method is applied to two different sequences, using both an orthogonal filter bank and a biorthogonal filter bank, and the results show a great improvement in the SNR of the estimated signal over the SNR of the noisy signal. In addition, the visual quality of the estimated data is much cleaner and crisper than that of the noisy data. Overall, this method is deemed successful at removing noise from noisy video sequences.

6.2 Future Work

The main contribution of this work is the extension of previous techniques to video data. Therefore, a possible future direction would be to try these methods on different multichannel images.

Another possible extension of this work would be to change the way in which a pixel is viewed throughout the channels. In this thesis, a pixel location is treated as constant throughout the entire video sequence; for example, the top right pixel in the first frame is the top right pixel in every frame of the video. In reality, the image that the video is portraying may be moving. For example, take a video of a balloon rising. The top of the balloon has a certain intensity value and this value moves upward in the image over time. In general, a pixel with a certain intensity can be said to change location throughout the frames. Methods for creating motion vectors which attempt to track the movement of, in the balloon example, the top of the balloon, or, more generally, a pixel or block of pixels throughout a video sequence have been developed, such as that used in MPEG encoding. These vectors could be used instead of the frames themselves. Decorrelation across the frames could be performed on the motion vectors instead, which could allow combination of video de-noising and encoding as in [29].

A further possible area for future work could be in the wavelet filter bank that is used. It is possible that there exists a set of filters which does a better job of decorrelating the data. Research into this possibility could yield results beyond those given in this thesis.

REFERENCES

[1] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* New York, NY: Wiley, 1949.

[2] M. Vetterli, J. Kovacevic, and V. K. Goyal, "The world of Fourier and wavelets: Theory, algorithms and applications," class notes for ECE 544, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 2007.

[3] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[4] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[5] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[6] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1522–1531, September 2000.

[7] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, September 2000.

[8] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Processing Letters*, vol. 6, pp. 300–303, December 1999.

[9] D. Wei, U. Rajashekar, and A. C. Bovik, "Wavelet denoising for image enhancement," in *Handbook of Image and Video Processing.* New York, NY: Academic Press, 2005, pp. 157–166.

[10] I. Atkinson, "Techniques for approximating optimal linear estimators of multi-dimensional data," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2007.

[11] I. Atkinson, "Wavelet-based near-optimal estimation of multichannel two-dimensional signals," M.S. thesis, University of Illinois at Urbana-Champaign, May 2003.

[12] S. Rout, “Orthogonal vs. biorthogonal wavelets for image compression,” M.S. thesis, Virginia Polytechnic Institute and State University, August 2003.

[13] I. A. F. Kamalabadi, S. Mohan, and D. L. Jones, “Asymptotically optimal blind estimation of multichannel images,” *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 992–1007, April 2006.

[14] I. C. Atkinson, F. Kamalabadi, D. L. Jones, and K. R. Thulborn, “Blind estimation for localized low contrast-to-noise ratio bold signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 6, pp. 879–890, December 2008.

[15] C. Boncelet, “Image noise models,” in *Handbook of Image and Video Processing.* New York, NY: Academic Press, 2005, pp. 397–410.

[16] R. A. Boie and I. J. Cox, “An analysis of camera noise,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 671–674, June 1992.

[17] B. Hunt and O. Kubler, “Karhunen-Loeve multispectral image restoration, part I: Theory,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 592–600, June 1984.

[18] H. Stark and J. Woods, *Probability and Random Processes with Applications to Signal Processing*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2001.

[19] C. K. Chui, *An Introduction to Wavelets.* San Diego, CA: Academic Press, 1992.

[20] Y. Chen and C. Han, “Adaptive wavelet threshold for image denoising,” *Electronics Letters*, vol. 41, no. 10, pp. 586–587, May 2005.

[21] M. N. Do and M. Vetterli, “Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance,” *IEEE Transactions on Image Processing*, vol. 11, pp. 146–158, February 2002.

[22] S. G. Mallat, “A theory for multiresolutional signal decomposition: The wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[23] J. W. Brewer, “Kronecker products and matrix calculus in system theory,” *IEEE Transactions on Circuits and Systems*, vol. 25, pp. 772–781, September 1978.

[24] J. W. Cooley and J. W. Tukey, “An algorithm for machine calculation of complex Fourier series,” *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.

[25] J. D. Villasenor, B. Belzer, and J. Liao, “Wavelet filter evaluation for image compression,” *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1053–1060, August 1995.

[26] T. Ogawa, N. Balan, Y. Otsuka, K. Shiokawa, C. Ihara, T. Shimomai, and A. Saito, “Observations and modeling of 630 nm airglow and total electron content associated with traveling ionospheric disturbances over Shigaraki, Japan,” *Earth, Planets and Space*, vol. 54, no. 1, pp. 45–56, 2002.

[27] L. B. Brown, A. J. Gerrard, J. W. Meriwether, and J. J. Makela, “All-sky imaging observations of mesospheric fronts in OI 557.7 nm and broadband OH airglow emissions: Analysis of frontal structure, atmospheric background conditions, and potential sourcing mechanisms,” *Journal of Geophysical Research*, vol. 109, 2004.

[28] J. J. Makela, “A review of imaging low-latitude ionospheric irregularity processes,” *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 68, pp. 1141–1158, 2006.

[29] L. Jovanov, A. Pizurica, S. Schulte, P. Schelkens, A. Munteanu, E. Kerre, and W. Philips, “Combined wavelet-domain and motion-compensated video denoising based on video codec motion estimation methods,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 3, 2009.