Edge Characteristics in Wavelet-Based Image Coding

Michael B. Wakin
Rice University - ELEC 599
Project Advisor: Richard G. Baraniuk

April 27, 2001

Abstract

Accurate prediction of wavelet coefficients relies on an understanding of the phase effects of edge alignment. This research examines techniques for uncovering edge information based on the available coefficients. These techniques are evaluated in the context of reconstructing an image from quantized wavelet coefficients. A predictor is described which can be trained on the coefficients to capture relationships among the pixels. Another method is presented where the quantized coefficients are interpolated to recreate an underlying continuous function. Based on this research, the interpolation process offers more promise in solving the dequantization problem.

1 Introduction

Edges are important in images, but difficult to handle. The wavelet decomposition, a popular tool for image processing, clearly indicates the presence of edges in the image. In general, one expects to see larger coefficients near an edge. More precise descriptions of the coefficients can be quite challenging, though. For example, a large coefficient at a given resolution may not correspond to a large coefficient at a finer resolution. Also, wavelet coefficients along a given edge may not be equal. The key to unlocking the mystery behind the wavelet nuances is phase.

A discrete wavelet transform can be viewed as a sampled version of a continuous, filtered image. While this continuous, filtered image shows significant energy near an edge, the actual values of the sampled wavelet coefficients depend on the wavelet basis function and the alignment of the sampling with the edge. This alignment, in turn, depends on the edge location and orientation. Techniques to explicitly estimate edge location and orientation in the spatial domain are examined in Appendix A.

In the wavelet domain, it is hoped that information about the edge can be abstracted from the observed coefficients, and that somehow this information may be exploited in processing the image. In particular, this paper focuses on the following problem while searching for information among the wavelet coefficients: Consider an image that has been coded in the wavelet domain (a continuous image that was sampled, then passed through a wavelet filter bank, and whose coefficients have been quantized). Decoding the image can be viewed as a reconstruction process; that is, for each quantized wavelet coefficient, a value must be chosen before implementing the inverse wavelet transform. Instead of choosing the default middle of each quantization bin, it is hoped that edge information may be extracted from the quantized wavelet coefficients which will help the decoder make an informed decision when assigning (predicting) coefficients for reconstruction. We refer to this as the DeQ problem for short.

Section 2 examines a covariance-based prediction scheme and its effectiveness towards the DeQ problem. Section 3 examines a different approach: viewing the wavelet coefficients as different samplings of the same continuous function. This gives hope towards a reconstruction technique as well.

As stated before, Appendix A deals with the spatial domain, considering the detection of an edge in noise and the estimation of its location. Such a technique may be useful for a variety of processing schemes that depend on explicit knowledge of an edge location. This scheme stands separate from the prediction methods, but possible extensions to the prediction ideas will be addressed.

2 Edge-Directed Prediction

Predictive coders use information they have learned about the image to predict the value of a pixel. We examine now a type of predictor which works well near edges. The goal is to take advantage of this effectiveness in solving the DeQ problem.

2.1 The Predictor

[1] describes a covariance-based predictor which aligns itself along an edge orientation. The predictor considers a pixel x_n of the image, and attempts to predict its intensity m_n as a linear combination of intensities of the N nearest neighbors $x_{n-1}, ..., x_{n-N}$ (the neighboring window can be of arbitrary shape, but a typical use involves a causal half-rectangle centered at x_n). Letting $\mathbf{s} = [m_{n-1} \ \ ... \ \ m_{n-N}]^T$, the linear prediction can be written as

$$\hat{m}_n = \mathbf{cs} \tag{1}$$

where the vector $\mathbf{c}$ contains the prediction coefficients. These prediction coefficients are obtained by defining a training window of the M pixels closest to x_n, and assuming that, for each one, the relationship to its nearest N neighbors may be described with $\mathbf{c}$. A least squares technique gives the best fit $\mathbf{c}$ for the data; this value of $\mathbf{c}$ is used to predict the intensity for x_n.

This predictor exploits structure in the data in finding the estimate for $\mathbf{c}$. Along the edge, intensities vary slowly, so the predictor notes that prediction works well in this direction. Across the edge, intensities vary rapidly, leading to unreliable predictions in this direction. In this sense, the predictor aligns itself along the edge. Most of its prediction weight is placed on pixels relative to x_n which line up in the direction of the edge. Thus, the prediction coefficients capture information about the edge in addition to providing reliable predictions for pixel intensities. We hope to make use of the former characteristic when training the predictor along an edge.

2.2 The DeQ Problem

Because the predictor works so well near edges, it is hoped that it may be applied to the DeQ problem. The predictor described in [1] works well in both the spatial and the wavelet domain because edge structures remain dominant in the higher resolution subbands of the wavelet transform.

Consider, then, a wavelet coefficient appearing along an edge in a high resolution subband of the wavelet transform. We treat this subband as its own image and denote the pixel simply as w_n. If the neighbors of w_n are not quantized, then the predictor can be used to predict the true value of w_n. If these neighbors have been quantized, however, the predictor may not be able to identify the structure of the edge. Moreover, even with the correct predictor coefficients, any prediction of w_n would be based on a linear combination of quantized neighboring coefficients. The problem becomes much more difficult. Experimental results indicate that by training the predictor on quantized neighbors, the true value of w_n cannot be reliably predicted as a linear combination of quantized neighbors. Another approach is needed.

We assume that the problems arising from the above approach come from predicting as a linear combination of quantized values. It is suspected that, despite quantization, the predictor may still be able to learn a great deal about the edge structure. We propose a way, then, to utilize this information without directly imposing a faulty linear prediction. This method involves estimating all coefficients jointly, and attempting to impose the predictor structure on these estimates.

The quantized subband coefficients are written as a single vector $\mathbf{w}^Q$; from this, it is desired to obtain an estimate $\hat{\mathbf{w}}$ such that $Q(\hat{\mathbf{w}}) = \mathbf{w}^Q$. In other words, the estimated values must fall in the quantization constraint set $\mathbf{Q}$. The estimated $\hat{\mathbf{w}}$ is used in a reconstruction of the image using the inverse wavelet transform. Performance is measured by comparing the reconstructed PSNR with that obtained by choosing the default middle of each quantization bin.

The algorithm is performed as follows. Each pixel w_n^Q of the block is analyzed. If the variance of its neighbors is high enough, it is assumed to be near an edge. If so, then the predictor is trained on the neighbors of the pixel, yielding a vector $\mathbf{c}_n$ of prediction coefficients. It is desired to impose the structure of $\mathbf{c}_n$ on the estimated coefficient $\hat{w}_n$ and its neighbors $\hat{w}_{n-1}$, ..., $\hat{w}_{n-N}$. The iterations continue for each pixel in the block, and all prediction vectors $\mathbf{c}_n$ are recorded. These are arranged in a matrix $\mathbf{A}$ such that the combined preferred structures arising from the predictors can be written

$$\mathbf{A}\hat{\mathbf{w}} = \mathbf{0} \tag{2}$$

Equation (2) describes a manifold $\mathbf{E} = \{\hat{\mathbf{w}} : \mathbf{A}\hat{\mathbf{w}} = \mathbf{0}\}$ on which the subband prediction coefficients should fall. This manifold does not necessarily intersect the constraint set $\mathbf{Q}$, so the optimal solution is chosen as the closest point in $\mathbf{Q}$ to the manifold.

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbf{Q}} \|\mathbf{A}\mathbf{w}\|^2 \tag{3}$$

Thus the overall divergence from the predictor structure is minimized. [2] provides an iterative scheme for solving this optimization problem.

2.3 Results

We now describe the basic implementation of the system described above. The image is chosen as a 128×128 portion of the standard Building image, and is shown in Figure 1(a). Daubechies79 wavelets with lifting are chosen to implement the wavelet transform. Quantized coefficients are obtained from the standard zerotree coding scheme [3]. By reconstructing the image using quantized wavelet coefficients (assumed to fall at the center of each quantization bin), a PSNR of 24.969 dB is obtained. The reconstructed image is shown in Figure 1(b). The goal of any reconstruction technique is to improve upon this PSNR. Table 1 lists the results for the techniques attempted.

For reconstruction, edge-directed prediction (EDP) is attempted on each of the three subbands (horizontal, vertical, diagonal) at each of the 2 finest resolutions. Each block is treated separately. A variance threshold for each subband is determined in advance to yield reasonable detection of edge locations. The training and prediction windows are chosen as *noncausal* rectangles centered at the pixel, since all of the prediction coefficients are deterministic and do not rely on some sequence of previously predicted pixels. Except near borders of the image, the training window consists of an

Reconstruction Technique	Practical PSNR (dB)	Oracle PSNR (dB)
Bin Centers	24.969	N/A
Basic EDP	24.933	25.072
NQCS	25.002	25.046
Bias Removal	24.987	25.125
NQCS w/ Bias Removal	25.006	25.044
Orthogonal EDP	N/A	24.892

Table 1: Reconstruction Results for Building Image

11×11 block ($M = 120$) and the prediction window consists of the $K = 20$ closest pixels to the pixel of interest. The $\mathbf{c}$ corresponding to this pixel is weighted by the variance of its neighbors; in effect, this places a stronger requirement on the pixel's relation to its neighbors when it falls near a definite edge. The iterative algorithm described by [2] is used to project from the quantization bin centers towards the manifold described by the $\mathbf{A}$ matrix (the collection of $\mathbf{c}$ vectors). The algorithm terminates once the distance converges or meets a certain minimum threshold, or after a certain maximum number of steps.

Once each of the 6 subbands have been predicted, the entire predicted wavelet transform is reassembled, and the predicted image is obtained through the inverse wavelet transform. Using our basic scheme for reconstruction, a PSNR of 24.933 dB is obtained, 0.036 dB lower than the bin-centers method. Recall that we had hoped the predictor would be able to train itself accurately on quantized coefficients, but see now that this may not be the case. To investigate this aspect, the true wavelet coefficients are used to train the predictor in each subband. Thus the $\mathbf{A}$ is that which would arise from an oracle predictor. This $\mathbf{A}$ is used for projection from the bin centers as before. Under this scheme, the reconstructed PSNR reaches only 25.072 dB, barely 0.10 dB higher than the standard metric. This is somewhat discouraging - in practice, the predictor could never be as well trained as the case of the oracle, and 0.10 dB offers very little promise in terms of image enhancement.

An interesting result is presented in [4] which pertains to this problem. It is suggested for problems involving projection onto convex sets (POCS), that instead of projecting within the quantization constraint set $\mathbf{Q}$, a subset $\mathbf{Q}_N$ called the narrow quantization constraint set (NQCS) may yield better performance. Put simply, this is because projections tend to fall on the boundary of $\mathbf{Q}$, which is likely not the case for the original image. Whereas each coefficient w_n was constrained in $\mathbf{Q}$ to some interval $[w_n^Q - \sigma_n, w_n^Q + \sigma_n]$, the NQCS technique restricts coefficients in $\mathbf{Q}_N$ to the interval $[w_n^Q - \mu\sigma_n, w_n^Q + \mu\sigma_n]$ where $0 < \mu \leq 1$. For our analysis, μ is chosen to be 0.2. Using this in the projection technique, we achieve a PSNR of 25.002 dB - a small practical improvement over the standard bin-centers reconstruction. Using an oracle predictor yields a PSNR of 25.046 dB.

The performance of our technique still leaves much to be desired. An investigation of the predicted wavelet coefficients reveals that the projection algorithm tends to predict many wavelet coefficients on the small end of their constraint interval; this algorithm has the property that it yields the optimal solution of smallest norm. Thus, the projection process reduces the energy of the wavelet transform. This is not always a bad thing, but it is also not a bias that should be held by the predictor. We would like for the expected energy of the predicted coefficients to remain unchanged. To correct for this, we make a change of variables. Instead of letting $\mathbf{c}$ describe the relationships

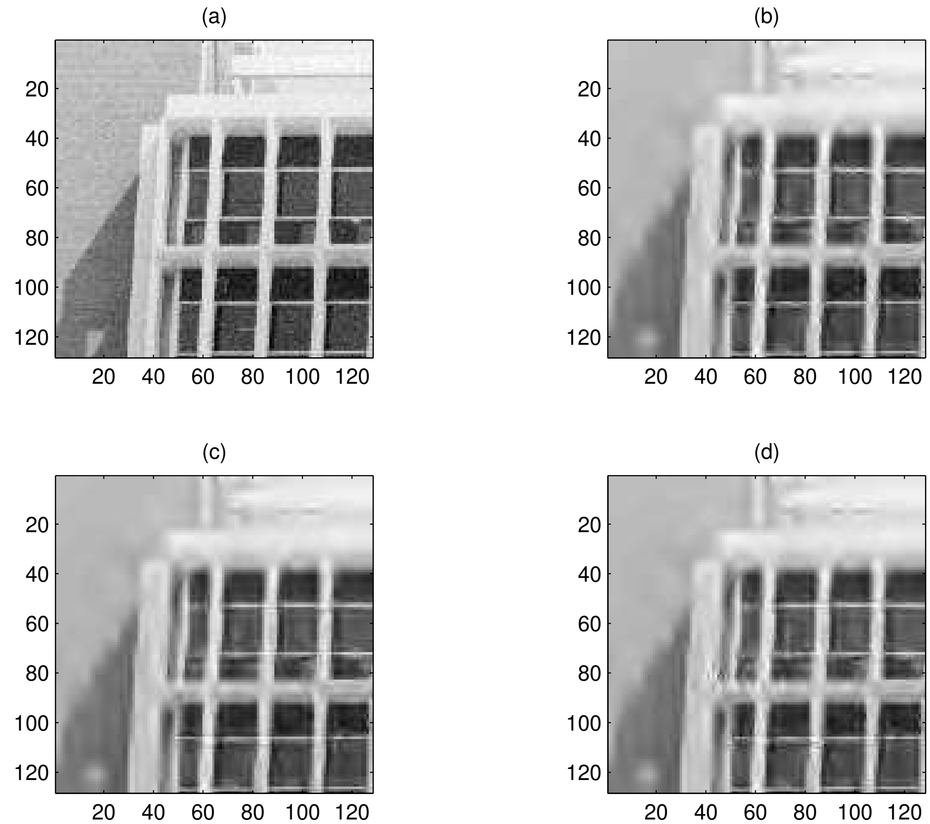


Figure 1: Reconstruction Results: (a) Original Image; (b) Reconstruction via Bin Centers (PSNR = 24.969 dB); (c) Oracle Reconstruction Using Bias Removal (PSNR = 25.125 dB); (d) Reconstruction Using Orthogonal EDP (PSNR = 24.892 dB).

among pixel values, each pixel is written as $w_n = w_n^Q + \epsilon_n$. The prediction equation given by $\mathbf{c}$ can be rewritten in terms of the variables ϵ_n. Because these variables also fall within convex constraint sets, they can be predicted using the projection scheme. Due to the bias of the projection, each ϵ_n favors values near zero, so the corresponding pixel predictions are biased towards w_n. Using this bias removal scheme (with no NQCS), a PSNR of 24.987 dB is achieved. The oracle predictor achieves 25.125 dB in this case. The resulting image from the oracle predictor (the best of any of our reconstruction techniques) is shown in Figure 1(c). The differences from the bin-centers method are difficult to distinguish.

For completeness, the NQCS method and the bias removal scheme are combined, yielding a PSNR of 25.006 dB. In this case, the oracle gives a PSNR of 25.044 dB.

The very small improvements obtained through these methods indicate that the predictor may be helping a bit to capture information about edge structure, but that a new method must be found for exploiting that information.

2.4 Refined Technique

We think now about the predictor's strengths and seek a new way to apply it to our problem. As stated in [1], most of the prediction weight falls in the direction of the edge. However, if pixels vary slowly in this edge direction, it may be impossible to recover information lost by quantization. It it still hoped, though, that the predictor may learn information about pixels in a direction perpendicular to the edge, and enforce the proper relationships among them.

We ask, then, whether the structure of these pixels sitting perpendicular to the edge is something that can be captured by the predictor. It would seem that this question is equivalent to asking whether these pixels could be modelled as a low-order AR process. If each perpendicular arrangement of pixels followed the same linear relationship, the predictor would likely be able to identify it. To seek an answer towards this question, a new scheme is developed, using a modified prediction window. For edges near vertical, the prediction window is simply a horizontal row (5 pixels on each side) - thus the predictor is prevented from learning about the structure along the edge. Using the original coefficients to train this orthogonal predictor, we reach a PSNR of only 24.892 dB in the reconstructed image (see Figure 1(d)).

Thus, our preliminary analysis indicates that the predictor cannot accurately capture behavior across the edge. Further analysis in this area could investigate the AR model for wavelet coefficients across an edge and better judge the potential use of an orthogonal predictor.

2.5 Database Idea

We turn now to an examination of the predictor coefficients obtained near an edge. This analysis leads to one final approach for using the predictor in the DeQ problem.

An image is created containing a single edge (see Figure 2). The wavelet transform is obtained, and the predictor is trained at various locations along the edge in each of the 3 finest subbands. Figure 3 shows the wavelet coefficients in these subbands, and Figure 4 shows the prediction coefficients obtained at three different locations along the edge. In these plots, gray represents the value 0; positive numbers are lighter and negative number are darker. It is observed that the roughly the same prediction coefficients appear regardless of the predictor's position on the edge. Indeed, these training windows were not carefully aligned along the edge - any predictor 'close' to the edge would yield the same coefficients. Typical values for these coefficients are also shown in Figure 4.

It can be shown that the prediction coefficients obtained in a given subband do not depend on the pixel intensity values of the original image. These intensities affect only the scaling of the wavelet coefficients; the prediction coefficients are invariant to such scaling. The consequence of this is a powerful result: the set of prediction coefficients in a given subband depends *only* on the angle of the edge in the image.

A new approach for the DeQ problem is conceived: a database can be constructed containing the ideal prediction coefficients for edges of each angle. The image is then reconstructed using the available quantized data (the bin-centers method). In the spatial domain, edges are identified using the local variance threshold, and their angles are estimated (Appendix A gives methods for estimating edge orientation). By referencing the database, these angle estimates give the set of prediction coefficients to use in the projection scheme in each subband.

By combining the available information, this method has the advantage of being able to obtain a good estimate for edge orientation. However, experiments do not support the anticipated improvements from this approach. On the building image, it was necessary to test with a higher zerotree bitrate to prevent too many subband coefficients from falling in the deadzone. The quantized image

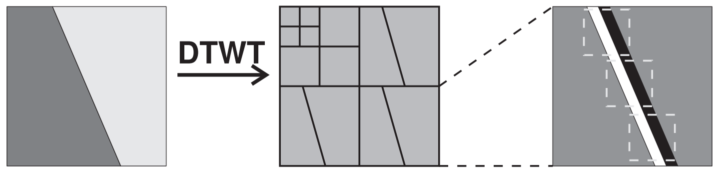


Figure 2: Predicting at Various Locations Along Edge

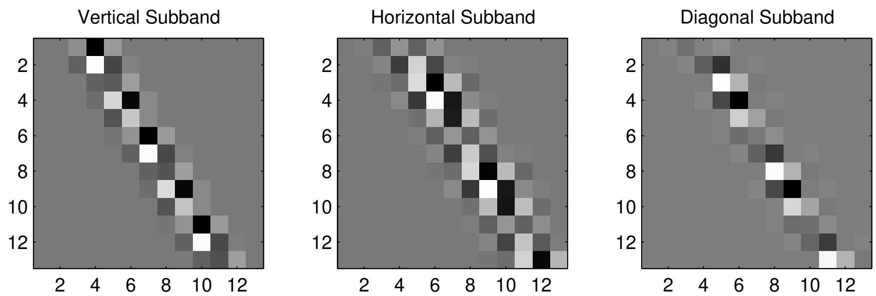
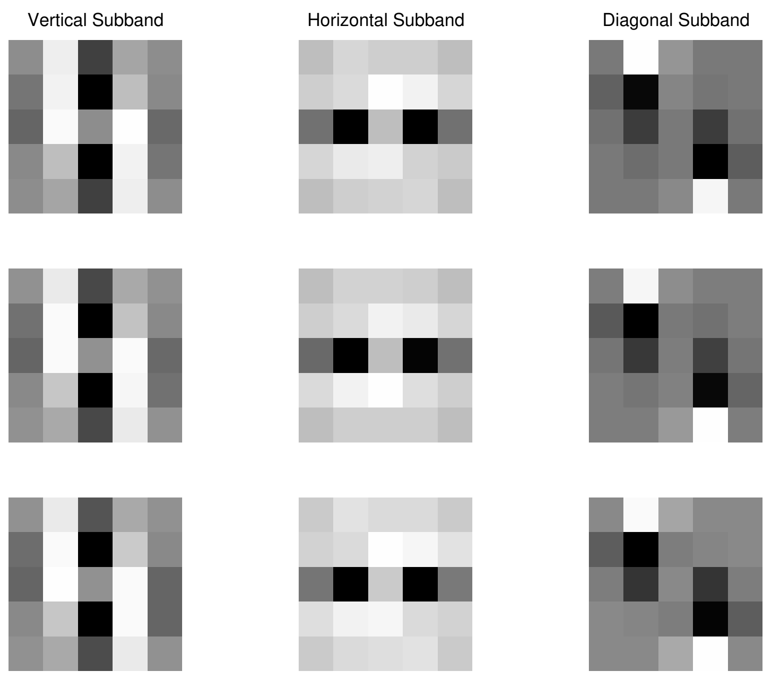


Figure 3: Actual Wavelet Coefficients



	.43	-.30	.11	
-.19	.49	-.65	.24	-.03
-.20	.48	X	.46	-.19
-.03	.25	-.68	.45	-.18
	.11	-.31	.41	

	.08	.08	.06	
.06	.11	.20	.18	.09
-.33	-.75	X	-.73	-.30
.10	.21	.24	.13	.06
	.06	.06	.07	

	.58	.07	-.02	
-.15	-.60	.00	-.07	-.01
-.06	-.32	X	-.29	-.05
-.01	-.05	-.02	-.56	-.13
	-.01	.10	.60	

Figure 4: Predictor Coefficients Trained on Various Locations

thus started with a PSNR of 31.812 dB. Implementing the database scheme with NQCS and bias removal improved the image only to 31.818 dB.

Despite the fact that the predictor captures information about the edge, using its coefficients in a projection scheme has failed to produce the desired results. We now look at the DeQ problem from a different angle.

3 Interpolation Analysis

We consider now a different approach. It is believed that, by considering the signal processing involved in the wavelet transform, information about edges (and phase) may be retrieved even from quantized wavelet coefficients.

We begin by interpreting the wavelet transform in a context which will explain the theory behind our interpolation approach. We then explore the simplifications and assumptions necessary for the method to succeed.

3.1 The Wavelet Transform

For simplicity of explanation, the following analysis will focus on the diagonal subband of the discrete wavelet transform at the highest resolution. We refer to this subband as W_{DIAG}.

In theory, there exist two methods for obtaining the coefficients W_{DIAG}. Figure 5 diagrams each process, and Figure 6 illustrates each in the Fourier domain. Beginning with a continuous image IM_C, traditional computational methods involve sampling this image to obtain a discrete representation IM_D. This discrete image is passed through a filter bank to yield the discrete wavelet transform IM_W. The subband W_{DIAG} is simply one quadrant of IM_W.

By considering the continuous wavelet transform, we see another method for obtaining W_{DIAG}. First, the continuous image IM_C is filtered (in each dimension) by the wavelet basis function ψ, which acts as a bandpass filter. The resulting continuous image IM_F is then sampled at the filter's cutoff frequency. These samples are decimated by a factor of 2 to yield the discrete coefficients W_{DIAG}.

3.2 Interpolation Idea

We consider a hypothetical image with a single sharp edge passing through. It is clear that, after applying the continuous wavelet basis filter, rectangular samples of IM_F taken at various locations along the edge would be identical. Because IM_C is constant along the edge, so would IM_F be constant along the edge. Thus, each horizontal cross-section of IM_F would yield the same 1-D function (with different offsets). It follows that the rows of W_{DIAG} would consist of various samplings of the exact same 1-D function, having different offsets. This is where phase enters the picture.

Thus there is hope for solving the DeQ problem. Because of the local nature of the wavelet basis filter, we expect that on small areas of the image consisting only of an edge, the wavelet coefficients meet the above assumption: that they arise from sampling the same function with different offsets. Because the image IM_F is bandlimited after being filtered, the 1-D function can be reconstructed from the true wavelet coefficients using interpolation. Thus, the offsets among the rows may also be determined. [5] states that, even after quantization, this underlying 1-D function can be reconstructed to a certain accuracy. After this function has been reconstructed, by determining the offset of each row, the true wavelet coefficients can easily be estimated.

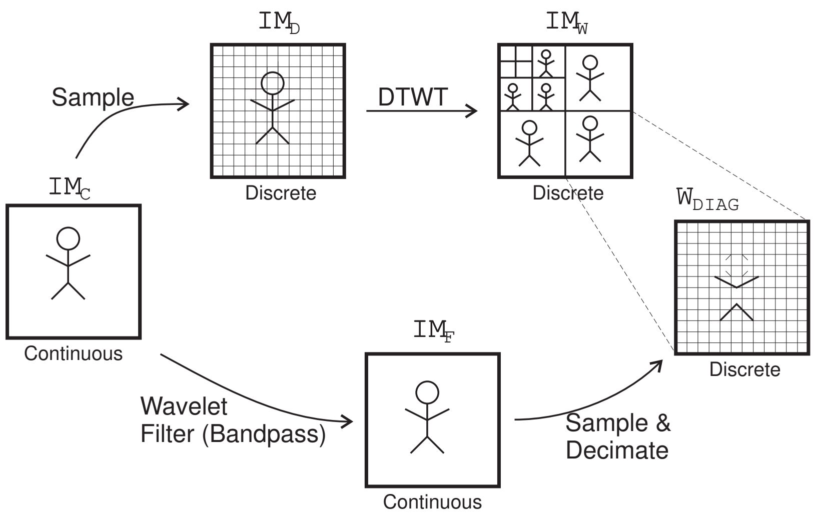


Figure 5: Two methods for obtaining a wavelet transform subband.

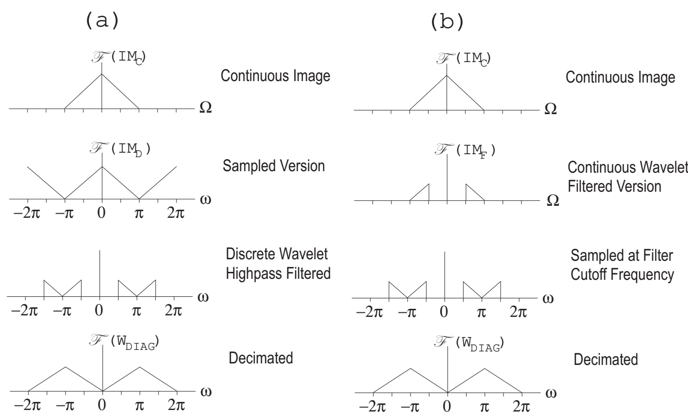


Figure 6: (a) Fourier effects of traditional discrete wavelet transform; (b) Alternative method.

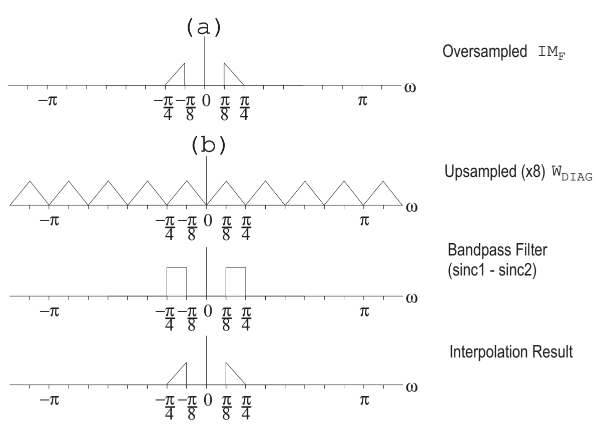


Figure 7: (a) The interpolation goal is to construct an oversampled version of IM_F; (b) the interpolation process starting from the subband wavelet coefficients.

We now attempt to reconstruct the continuous IM_F from a set of wavelet coefficients W_{DIAG}. These wavelet coefficients arise from the wavelet transform on an artificial image IM_D containing a single edge. The discrete image is obtained through the typical "averaged sampling" of a hypothetical continuous edge - where pixels along the edge are weighted according to the edge location. Symmetrical linear-phase Daubechies79 wavelets are used to obtain the wavelet transform. Limited by the discrete capabilities of computers, we choose to construct a highly oversampled version of IM_F. This is accomplished by first upsampling W_{DIAG} by a factor of 8, then filtering with the appropriate bandpass filter. (Higher oversampling factors have been attempted but give visually similar results). Figure 7 illustrates the interpolation idea in the frequency domain. The bandpass filter is created as a difference of two sinc functions.

The outcome of this preliminary test is shown in Figure 8. The frequency domain effects of this process are shown in 9. The results are not as expected; the reconstructed block shows ringing and periodicity in the edge direction. Thus, some assumptions in the above analysis must be flawed. We now investigate these issues and their impact on the DeQ problem.

3.3 Interpolation Issues

The preceding analysis justifying the interpolation idea makes a few assumptions which must be addressed to obtain the desired results.

3.3.1 Prefiltering

The first simplification in the above process can be seen in Figure 5. While the discrete wavelet transform often acts on the sampled discrete image IM_D directly, this does not yield the true wavelet coefficients. Instead, the wavelet filter bank should be applied to the initial scaling coefficients. While the scaling coefficients are often very similar to the actual data at that resolution, the difference may be enough to cause problems for our interpolation analysis.

[6] gives a method for approximating the true scaling coefficients based on the discrete image. The algorithm involves prefiltering the discrete image with samples of the scaling function taken on the integers. This method has been attempted in our analysis: the scaling coefficients for the ideal edge have been approximated before taking the wavelet transform. Interpolation on the resulting wavelet coefficients reveals results almost identical to Figure 8. Thus, the prefiltering issue does not seem to be causing the unexpected interpolation results. The remaining analysis on interpolation will be performed without prefiltering, since it would be impossible to implement in the DeQ problem.

3.3.2 Aliasing

Bandpass wavelet filters do not have perfect cutoffs in the frequency domain. Thus a continuous image which is not bandlimited will not be truly bandlimited after continuous filtering (see Figure 6(b)). After sampling, aliasing will occur which affects the wavelet coefficients and threatens the reconstruction procedure. The effect can be seen in the Fourier transform of these coefficients, shown in the first column of Figure 9.

To address the issue of high frequency aliasing, a sample image is created by passing the ideal edge through a lowpass filter. The wavelet coefficients are computed for this sample image and interpolated. The results are shown in Figures 10 and 11.

Interpolation results are greatly improved by smoothing the original image. Thus, aliasing that arises from high frequency components of the image causes great problems in interpolation. This

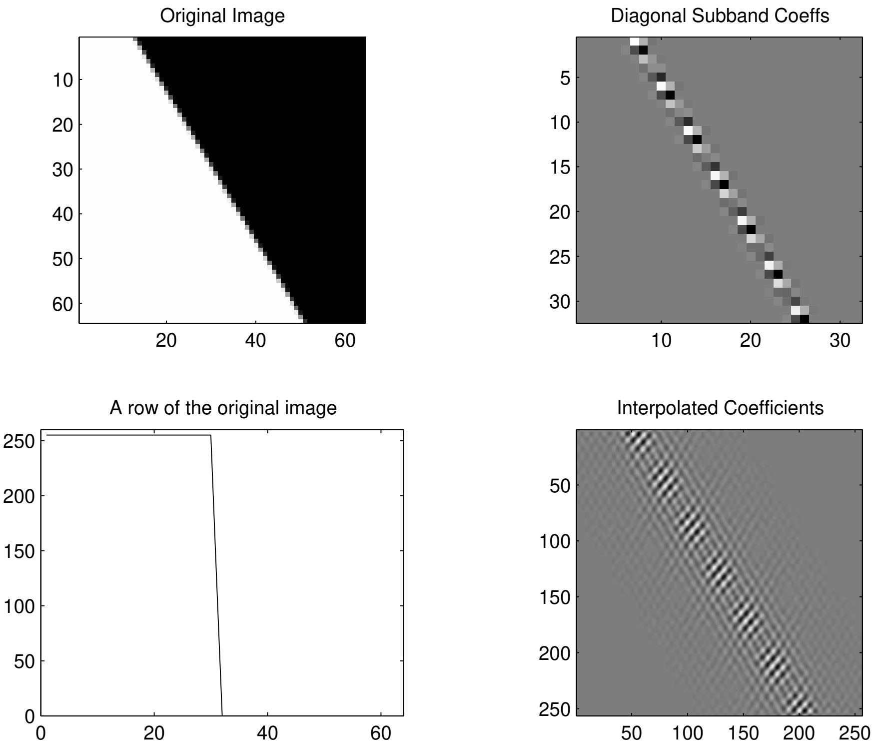


Figure 8: Interpolating the Wavelet Coefficients.

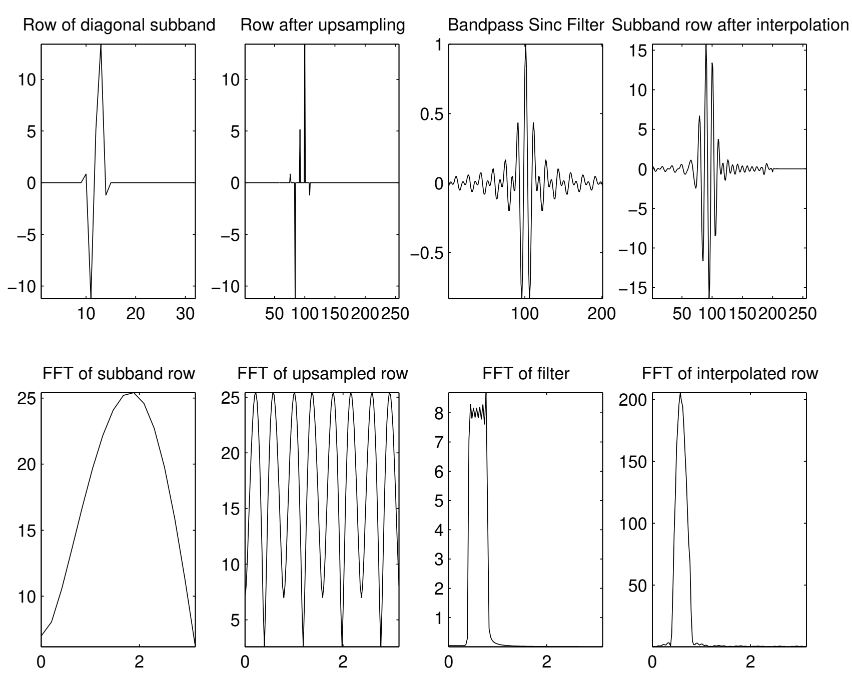


Figure 9: Interpolation Process in the Frequency Domain.

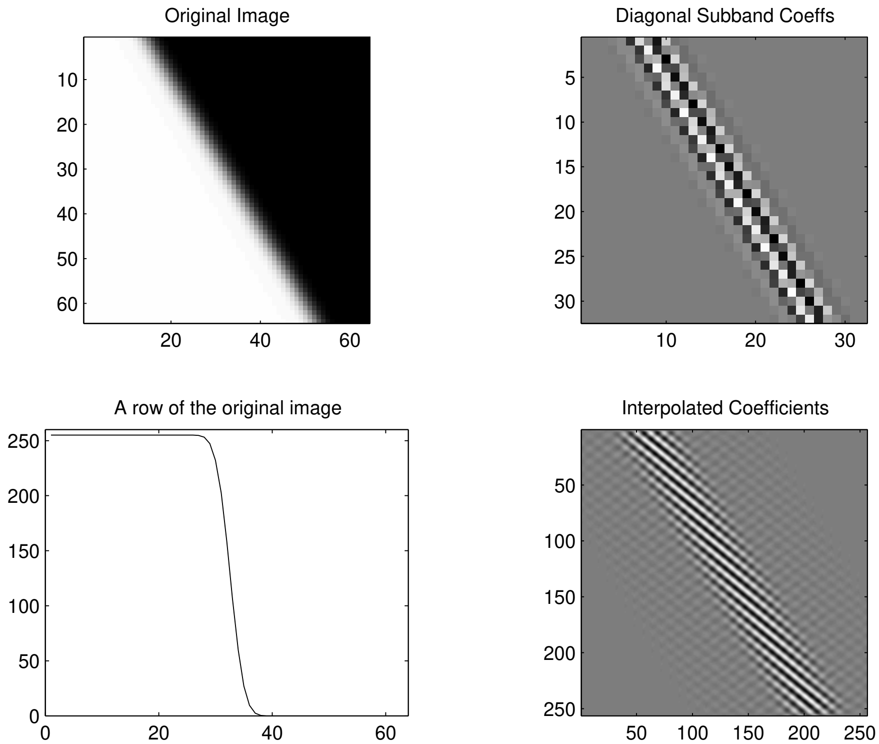


Figure 10: Interpolating the Wavelet Coefficients of a Smooth Edge.

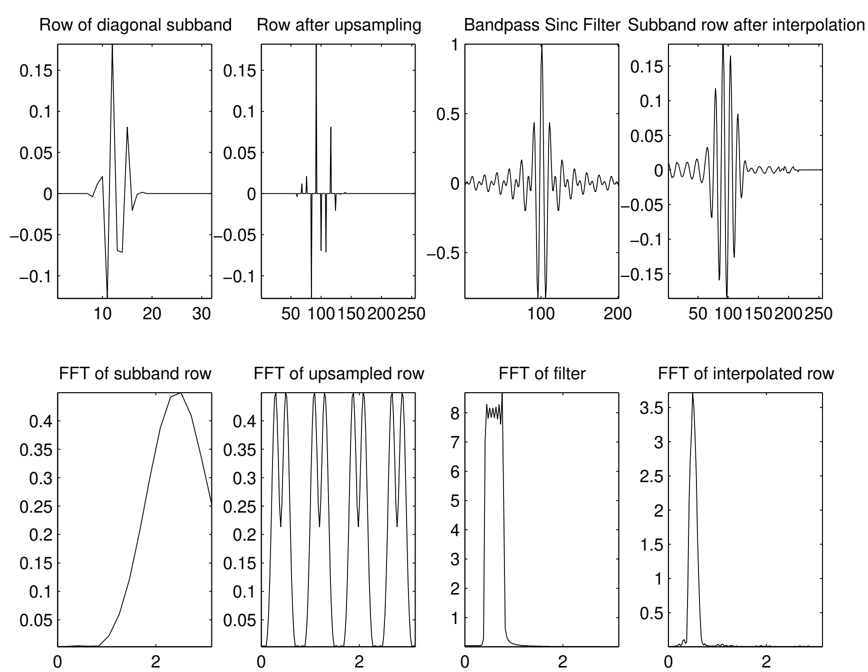


Figure 11: Interpolation Process in the Frequency Domain for a Smooth Edge.

may present difficulties in solving the DeQ problem, where smoothing the original image is both undesirable and impossible.

Despite the known effects of this high frequency aliasing, removing high frequency components of the continuous image does not result in a perfect reconstruction. This is likely due to the fact that aliasing results also from ringing on the low end of the wavelet filter's frequency response. An examination of natural images may be necessary to identify the severity of this issue.

3.4 Interpolation Results

Now that the significant issues have been addressed, we present more experiments involving interpolation. These will reveal the potential usefulness in dequantization.

3.4.1 Changing the Angle

Using the same smoothing filter from section 3.3.2, a variety of images are constructed, each containing an edge at a different angle. The diagonal subband coefficients are computed and interpolated. The resulting images are shown in Figure 12. As can be seen, interpolation works best for angles closest to the 45-degree line (the inherent direction of the subband filter). It is supposed, then, that in all three subbands, interpolation will work best for edges closest in angle to the filter's direction.

The interpolation process is easily extended to the horizontal and vertical subbands. In each filter's highpass direction, the bandpass sinc is used for interpolation. In each filter's lowpass direction, a standard lowpass sinc is used for interpolation. In the next section, we use all three subbands to test interpolation on natural images.

3.4.2 Natural Images

Now that aliasing has been identified as an issue, the best way to judge its importance may be to evaluate the interpolation procedure on natural images. This should give an indication into the usefulness of this procedure in solving the DeQ problem.

A 128×128 portion of the Cameraman image is chosen as an example. This block contains a portion of the arm and exhibits relatively sharp edges at a few different angles. The wavelet coefficients and interpolation results are shown in Figure 13. As expected, interpolation seems to work best for edges in the direction of the subband filter.

Next, A 100×100 portion of the Boats image is chosen. This block contains a variety of edges at various levels of sharpness. The results are shown in Figure 14.

The inevitable aliasing does not seem to prevent the interpolated image from exhibiting constant behavior along the edge. To examine this perceived constant behavior, a portion of the interpolated Boats image is examined. The pair of diagonal wires which starts at the top left portion of the image is chosen because it appears to interpolate well. Various rows of the interpolated image are chosen, and after proper alignment, the rows are plotted in Figure 15. These row sections confirm the fact that the interpolated function remains roughly constant along the diagonal.

Based on these examples from natural images, the interpolation procedure seems to hold some promise towards solving the DeQ problem.

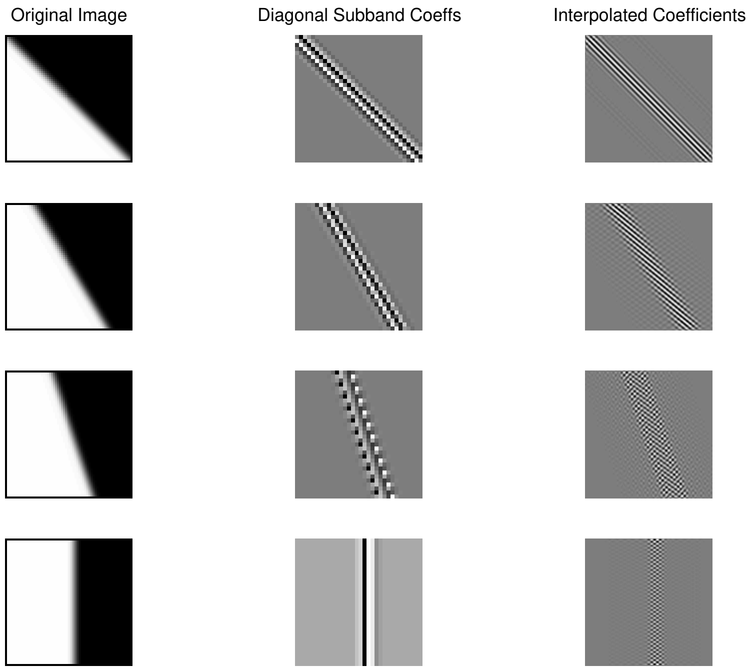


Figure 12: Interpolation Results for a Variety of Edge Orientations.

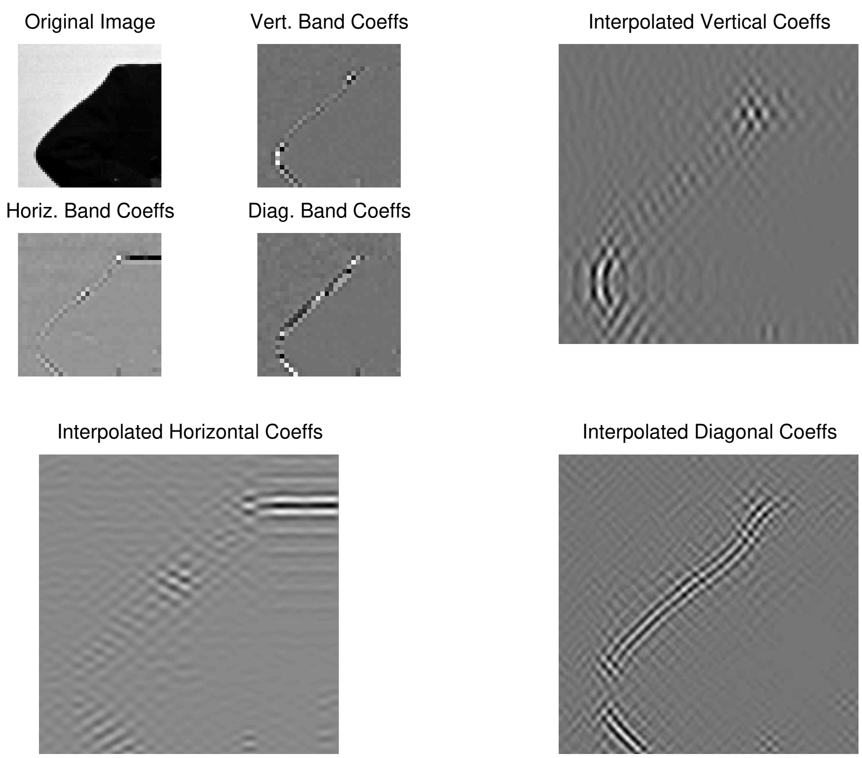


Figure 13: Interpolation on Cameraman's Arm.

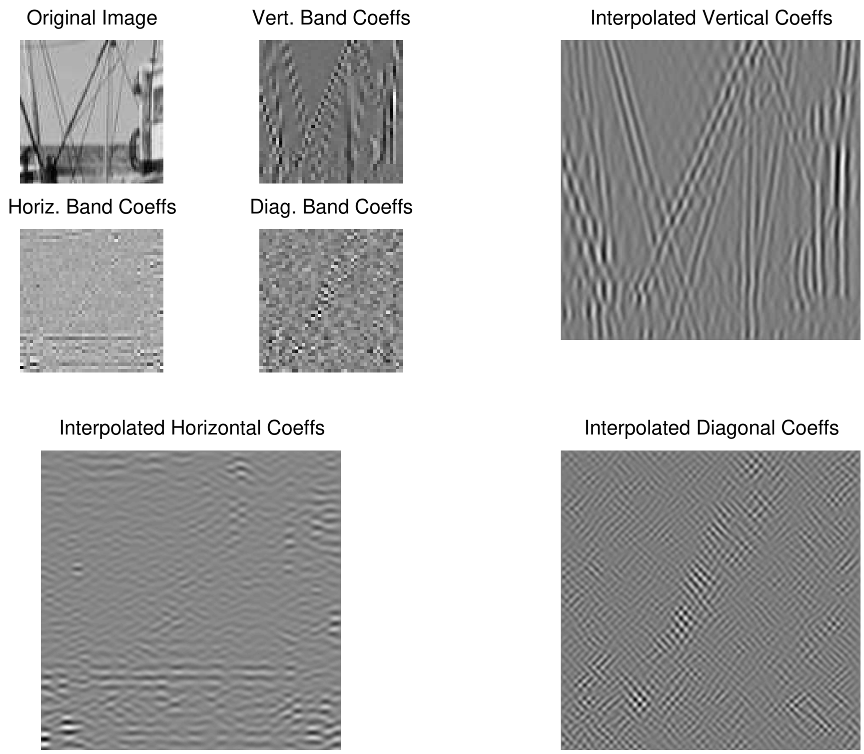


Figure 14: Interpolation on a Portion of the Boats Image.

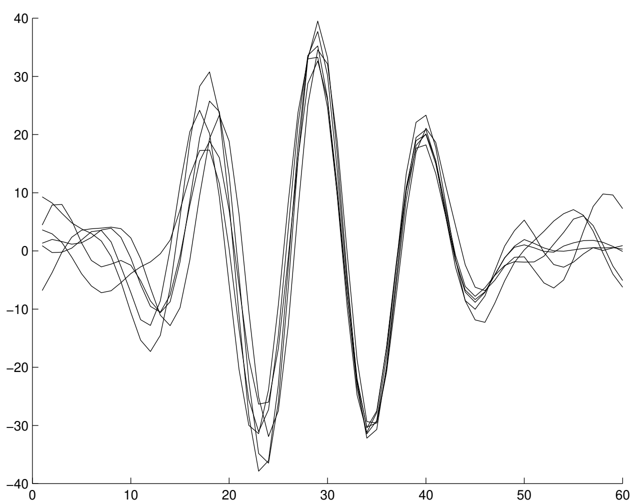


Figure 15: Cross-sections of an Interpolated Edge.

4 Conclusions

Solving the DeQ problem presents a unique challenge. Namely, the reconstruction technique from coded wavelet coefficients must improve upon the image quality intended from the decoder. Thus, information must be exploited that is ignored by the original coder. This research attempts to capture phase information from wavelet coefficients. Our analysis has examined two main ideas for capturing or identifying such information: using a predictor, and interpolating the coefficients.

The predictor, while quite effective in wavelet image coding, does not seem to capture the appropriate information to solve the DeQ problem. Although the predictor can accurately predict one coefficient in terms of its neighbors, enforcing this relationship when reconstructing from quantized coefficients does not seem to restore the proper alignment to the coefficients.

The interpolation technique attempts to discern the continuous function underlying the wavelet coefficients in each subband. On nonquantized coefficients in natural images, this function can be reasonably reconstructed. Further analysis for solving the DeQ problem should focus on the requirements of [5] for reconstructing a function from quantized samples. It is expected that the [5] technique will work only when the coding bitrate is sufficiently high, as the underlying function is required to cross a certain number of quantization levels. A major challenge in applying this to practical problems will be identifying regions in the reconstructed image where the function cleanly follows an edge. Appendix A deals with a similar issue in the spatial domain.

A Appendix: Edge Detection and Estimation

In image processing, a common method for analysis is to break the image into smaller blocks of $N \times N$ pixels. Each block is assumed to be small enough to allow at most one edge to pass through. This analysis will concentrate on such blocks of data; specifically, the situation where an edge enters the block through its top and leaves through the bottom (the edge does not cut a corner). We call such an edge a "top-bottom edge". In such a situation, we model the edge as follows (see Figure 16). The parameters α and β completely characterize the edge location and take real values between 0 and N. The parameters A and B define the unit pixel intensity on the left and right sides of the edge, respectively. A typical assumption in discrete images limits the range of these real parameters between 0 and 255.

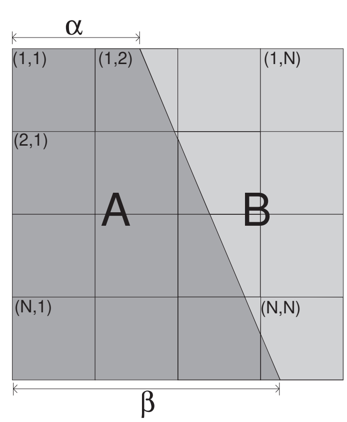


Figure 16: Parameterization of a Top-Bottom Edge

We consider, then, a detection problem: given an $N \times N$ block of an image, does a top-bottom edge pass through the block? The block is assumed to be observed in the presence of additive white Gaussian noise. Along with this problem comes one of estimation: assuming an edge does pass through the block, what are its parameters?

The detection problem can be summarized as follows:

$$\begin{aligned}&\mathcal{M}_0 : \text{Block contains noise only}\\&\mathcal{M}_1 : \text{Block contains top-bottom edge in the presence of noise}\end{aligned}$$

Due to the unknown parameters in the models, the generalized likelihood ratio test (GLRT) is the method chosen to solve the detection problem. As will be shown, though, the maximum-likelihood estimate of parameters under $\mathcal{M}_1$ is quite a challenge. It will therefore be necessary to approximate the maximum of the likelihood function. We will investigate such techniques and use simulation to judge their effectiveness.

A.1 Defining the Problem

Let $\mathbf{R}$ be the observed $N \times N$ block of the discrete image, where pixel (i, j) is denoted by $R_{i,j}$. Noise is modeled by $\mathbf{W}$, where $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_{N^2 \times N^2})$. Define $\mathbf{X}$ as the mean field: the expected intensity level of the image based on the edge parameters. The matrix $\mathbf{X}$ can be viewed as the signal in the presence of noise. Under both models, $R_{i,j} = X_{i,j} + W_{i,j}$ where $W_{i,j} \sim \mathcal{N}(0, \sigma_w^2)$.

Note that $\mathbf{X}$ is characterized completely by the (unknown) edge paramaters α, β, A, and B. Under $\mathcal{M}_0$, $X_{i,j} = A \; \forall \; i, j$. Under $\mathcal{M}_1$, $\mathbf{X}$ depends on the parameters of the edge. For pixels not touching the edge, $X_{i,j}$ is either A or B, but for pixels crossing the edge, the intensity depends on the position of the edge in the pixel. For the detection problem, the GLRT can be written as:

$$\Lambda(\mathbf{R}) = \frac{\max\limits_{\alpha,\beta,A,B,\sigma_w^2} \left(\prod\limits_{i,j=1}^{N} p_{W_{i,j}|\mathcal{M}_1,\mathbf{X},\sigma_w^2}(R_{i,j} - X_{i,j}|\mathcal{M}_1, \mathbf{X}, \sigma_w^2) \right)}{\max\limits_{A,\sigma_w^2} \left(\prod\limits_{i,j=1}^{N} p_{W_{i,j}|\mathcal{M}_0,\mathbf{X},\sigma_w^2}(R_{i,j} - X_{i,j}|\mathcal{M}_0, \mathbf{X}, \sigma_w^2) \right)} \tag{4}$$

The denominator (corresponding to $\mathcal{M}_0$) is a familiar estimation problem. We think of $R_{i,j}$ as a sequence of i.i.d. Gaussian random variables with unknown mean and variance. [7] gives the maximum likelihood estimates for the unknown parameters:

$$\hat{A}^{\mathrm{ML}_0} = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j} \qquad\qquad \hat{\sigma}_w^{2\,\mathrm{ML}_0} = \frac{1}{N^2} \sum_{i,j=1}^{N} (R_{i,j} - \hat{A}^{\mathrm{ML}_0})^2 \tag{5}$$

The numerator (corresponding to $\mathcal{M}_1$) requires more effort.

A.2 Estimation of Edge Parameters

We now assume the presence of a top-bottom edge and describe a series of techniques for finding maximum likelihood estimates of the unknown edge paramaters α, β, A, and B. Recall that these parameters correspond to a unique mean field $\mathbf{X}$. Thus, estimating these four parameters is equivalent to choosing a particular mean field $\mathbf{X}$ from the space $\mathbf{E}$ of such valid fields. The space $\mathbf{E}$ of images containing noise free top-bottom edges can be viewed as a 4-Dimensional manifold (parameterized by α, β, A, and B) in the N^2-Dimensional space of images. We will further investigate this interpretation in a moment.

The noise parameter σ_w^2 must also be estimated in addition to the edge parameters. It can be shown that

$$\hat{\sigma}_w^{2\mathrm{ML}_1} = \frac{1}{N^2} \sum_{i,j=1}^{N} (R_{i,j} - \hat{X}_{i,j}^{\mathrm{ML}_1})^2 = \frac{1}{N^2} \left\| \mathbf{R} - \hat{\mathbf{X}}^{\mathrm{ML}_1} \right\|^2 \tag{6}$$

Plugging this into the log-likelihood, we see:

$$\max_{\mathbf{X},\sigma_w^2} \log(p_{\mathcal{M}_1}) = -\frac{N^2}{2} \log\left(\frac{2\pi}{N^2} \left\| \mathbf{R} - \hat{\mathbf{X}}^{\mathrm{ML}_1} \right\|^2\right) - \frac{N^2}{2} \tag{7}$$

Thus, the maximum likelihood estimate of $\mathbf{X}$ corresponds to the closest point[1] on the manifold $\mathbf{E}$ to the observed data $\mathbf{R}$. The estimate of σ_w^2 simply depends on the distance from the data to the estimated mean field $\hat{\mathbf{X}}^{\mathrm{ML}_1}$. Despite the now simple statement of the problem, its solution is quite difficult. The manifold is not a convex set, eliminating most standard projection methods. Also, the equation for $\mathbf{X}$ is generally nonlinear in the parameters A, B, α, and β. Lacking a precise solution to the problem, we present a series of techniques for approximating the maximum likelihood estimates of the edge parameters.

A.2.1 Exhaustive Search

One potential technique for estimation involves trying all possible edge locations and mean values, then selecting that arrangement which comes closest to the observed data. However, because all edge parameters take continuous values, a truly exhaustive search of the manifold is impossible. Moreover, attempts to discretize the search limit the precision of the estimates. If α and β are restricted to be integers between 0 and N, while A and B are restricted to be integers between 0 and 255, then the search involves $(256^2) * N^2$ possible mean fields. Each potential mean field must then be compared to the observed data, requiring $\mathcal{O}(N^2)$ operations at each step. At $\mathcal{O}(N^4)$, the overall computational complexity of such a search is considered to be unacceptably high. In addition, in situations with low noise, one might hope to estimate the parameters with greater precision.

A more intelligent algorithm would not attempt to search the whole manifold. Instead, it would directly use the observed data to form an estimate of the parameters. From there, a search could be performed locally on the manifold. Such methods for more direct parameter estimates are now presented.

A.2.2 Row Sums

The Row Sums technique reduces the manifold $\mathbf{E}$ to a more manageable one. Each row of observed data is summed to produce an aggregate statistic $S_i = \sum_{j=1}^{N} R_{i,j}$. In the noise free case, the vector $\mathbf{S}$ can be expressed in terms of the edge parameters. It can be shown that

$$S_i = \alpha \left(\frac{N - i + \frac{1}{2}}{N}\right)(A - B) + \beta \left(\frac{i - \frac{1}{2}}{N}\right)(A - B) + BN \tag{8}$$

Suppose for the moment that A and B are known, and that only α and β need be estimated. Then the above expression may be rewritten as $S_i = \alpha M_{i,1} + \beta M_{i,2} + C_i$ where $\mathbf{M}$ is defined by A and B, and $C_i = BN \ \forall\ i$. Letting $\mathbf{y} = [\alpha\ \beta]^T$, we have in matrix form, $\mathbf{S} = \mathbf{M}\mathbf{y} + \mathbf{C}$. Thus, the

[1] Here we treat of the space of all images as $\mathbb{R}^{N^2}$ and use the ℓ^2-norm.

expression for the expected row sums is affine in terms of α and β. In the noisy case, the observed data $\mathbf{S}$ may not exactly obey the affine relationship above. However, since summing the noise across a row gives another zero-mean Gaussian random variable, we simply seek the values of α and β which give the best fit to the data. That is, we seek the $\hat{\mathbf{y}}$ which minimizes $\|\mathbf{S}-\mathbf{C}-\mathbf{M}\mathbf{y}\|^2$. The pseudoinverse provides the solution for the linear regression problem: $\hat{\mathbf{y}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T(\mathbf{S}-\mathbf{C})$.

Note that this technique depends on knowledge of A and B. However, using the sample mean to estimate A (or B) requires knowledge of the edge location (which depends on α and β). Due to this intermingling of the parameters, an iterative algorithm is proposed as follows:

Step 1 Estimate A and B in a simple manner. Because we assume the presence of a top-bottom edge in the block, a reasonable estimate of A might be the sample mean of the data in the left column only. Similarly, B is estimated as the sample mean of the data in the right column only.

Step 2 Use the estimates of A and B to implement the Row Sums technique described above, giving estimates of α and β.

Step 3 Estimate A and B. This time, it is possible to begin with a better guess of the edge location. The estimate for A (similarly B) can be calculated as the sample mean of all pixels lying to the left (right) of the estimated edge location. Note that, for accurate estimates of α and β, this method gives the true maximum likelihood estimate of A and B.

Step 4 Given the estimates of A, B, α, and β, calculate the estimated $\hat{\mathbf{X}}$ and find $\|\mathbf{R}-\hat{\mathbf{X}}\|$.

Step 5 Repeat Steps 2-4, obtaining new estimates for the parameters and $\mathbf{X}$. Continue repeating this algorithm as long as the distance $\|\mathbf{R}-\hat{\mathbf{X}}\|$ decreases. The final estimate of the algorithm is the $\hat{\mathbf{X}}$ closest to $\mathbf{R}$.

Note that, as long as the algorithm proceeds, each step of the algorithm offers a better estimate $\hat{\mathbf{X}}$. Thus, if the algorithm does not terminate, the error $\|\mathbf{R}-\hat{\mathbf{X}}\|$ will eventually converge to some value (and it may be safely terminated with small error). In practice, no cases have been found where the algorithm does not terminate after about 5 steps. There is no reason to claim that this algorithm gives the true maximum likelihood estimate of $\mathbf{X}$. However, the complexity of this approach can be shown to be $\mathcal{O}(N^2)$, much smaller than the Exhaustive Search.

A.2.3 Change Detection

An established theory exists for detection of parameter changes in 1-D signals. For example, [8] gives methods for detecting the change of a mean parameter of a sequence of Gaussian variables. We attempt now to apply this theory to the problem at hand. An obvious 1-D signal in the block is a given row i. Each row can be viewed as a sequence of Gaussian variables with unknown mean A to the left of the edge and unknown mean B to the right of the edge. The point of change, k_i, is also unknown. In the typical 1-D point-of-change problem, the transition is assumed to be abrupt; that is, the mean changes immediately from A to B (see Figure 17). In this situation, [8] gives the maximum likelihood estimates of the unknown parameters:

$$(\hat{k}_i, \hat{A}_i, \hat{B}_i) = \arg\min_{1\le k\le N} \inf_{A,B} \left(\sum_{j=1}^{k-1} (R_{i,j}-A)^2 + \sum_{j=k}^{N} (R_{i,j}-B)^2 \right) \tag{9}$$

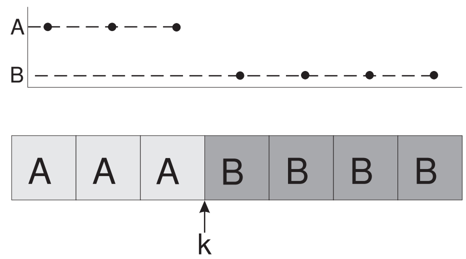


Figure 17: Typical 1-D Point-of-Change Problem

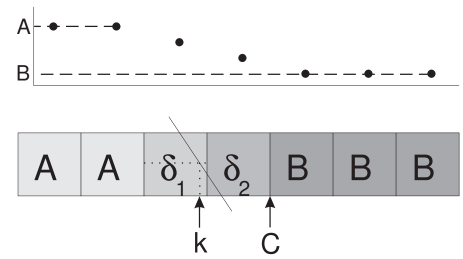


Figure 18: Modified 1-D Point-of-Change Problem

The estimates may easily be found by trying all values of k, finding the sample mean on both sides of the break point for each k, and choosing the $\hat{k}$ which maximizes the likelihood. Applying this technique to a row of the image, however, would fail to model subtle changes near the edge. We concentrate instead on a modified 1-D point-of-change problem, which accounts for pixels crossed by the edge, whose means are neither A nor B. In general, the change in means is not abrupt, but (for a top-bottom edge) may involve as many as two pixels in the transition. The general scenario is shown in Figure 18.

The modified point-of-change problem involves two new unknown paramaters: δ_1 and δ_2, corresponding to the means of the two pixels in the transition area. (C is an integer corresponding to the first pixel with mean B, so if the edge crosses only one pixel, we have the situation where $\delta_1 = A$. In Figure 18, $C = 4$.) Adapting the solution from the traditional point-of-change problem, the ML estimates for the unknown paramaters are given by:

$$(\hat{C}_i, \hat{A}_i, \hat{B}_i, \hat{\delta}_{1i}, \hat{\delta}_{2i}) = \arg \min_{2 \leq C \leq N} \inf_{A,B,\delta_1,\delta_2} \begin{pmatrix} \sum_{j=1}^{C-2} (R_{i,j} - A)^2 + (R_{i,C-1} - \delta_1)^2 \\ + \\ (R_{i,C} - \delta_2)^2 + \sum_{j=C+1}^{N} (R_{i,j} - B)^2 \end{pmatrix} \tag{10}$$

A potential solution would then be to iterate over the index C, calculating sample means for A and B at each step. Assume WLOG that $\hat{A} \geq \hat{B}$. For a given C, $\hat{\delta}_1$ and $\hat{\delta}_2$ must also be calculated such that they satisfy $\hat{A} \geq \hat{\delta}_1 \geq \hat{\delta}_2 \geq \hat{B}$. If the observed data in those two pixels meet this criteria, then the data itself may be used as the ML estimate for $\hat{\delta}_1$ and $\hat{\delta}_2$. Otherwise, high values must be clipped to $\hat{A}$, and low values must be clipped to $\hat{B}$. If the observed $\delta_2 > \delta_1$, then these values are averaged, and the estimate for both is taken to be the average. Once parameters A, B, δ_1, and δ_2 have been estimated for each C on a given row, then $\hat{C}$ is chosen as the index which maximizes the likelihood. Finally, using estimates $\hat{\delta}_1$ and $\hat{\delta}_2$ corresponding to this index $\hat{C}$, the estimated point of change $\hat{k}$ is the horizontal offset where the edge crosses the vertical middle of the row (see Figure 18). The estimate is given by $\hat{k} = \hat{C} - \left(\frac{\hat{\delta}_1 + \hat{\delta}_2 - 2\hat{A}}{\hat{B} - \hat{A}}\right)$.

Now that the 1-D point-of-change problem can account for the type of edges encountered in our problem, we propose a method for estimating the overall edge location in the block. A potential algorithm for edge estimation would be to estimate the edge crossing of each row, then use a linear regression on these points to estimate the edge location in the image. The mean parameters may then be estimated using the sample mean on each side of the estimated edge. This is known to provide the ML estimate of the means for a given edge location.

Note that the 1-D point-of-change problem can be solved both when the mean parameters are known and unknown. Thus, this method does not depend on an initial estimate of A and B. The following algorithm with complexity roughly $\mathcal{O}(N^3)$ is proposed:

Step 1 Iterate over all rows i. On each row, solve the "unknown means, unknown point-of-change" problem. The estimated points of change over the rows are combined to give a least-squares best fit for the edge in the image (giving estimates for α and β), and the sample means on each side of the edge give estimates for A and B.

Step 2 Given the estimates of A, B, α, and β, calculate the estimated $\hat{\mathbf{X}}$ and find $\|\mathbf{R} - \hat{\mathbf{X}}\|$.

Step 3 Use the current estimates of A and B to implement the "known means, unknown point-of-change" problem on each row. Find the best fit edge to these points of change, giving estimates for α and β, and use these to estimate A and B in the traditional fashion.

Step 4 Repeat Steps 2-3, obtaining new estimates for the parameters and $\mathbf{X}$. Continue repeating this algorithm as long as the distance $\|\mathbf{R} - \hat{\mathbf{X}}\|$ decreases.

A.2.4 Exhaustive Search (Revisited)

Thus far, the discussion of the point-of-change problem has focused on the 1-D case. The problem, though, has a nice extension to 2 dimensions. Indeed, our whole investigation can be stated as a 2-D point-of-change problem, where the "point-of-change" is the edge itself. The solution of the 1-D problem [8] gives insight into an approach for a true solution of the 2-D ML problem.

In one dimension, all possible values for the "point of change" are considered, and ML estimates for mean parameters are obtained using sample means on each side. Extending this idea to two dimensions, all possible edge locations would need to be considered, but for each potential location, we now know that the sample mean on each side would yield maximum likelihood estimates of A and B. Thus, solving the 2-D point-of-change problem reduces to an exhaustive search, such as that explored in Section A.2.1, but the complexity is reduced. Now, the search may be conducted only over α and β; for each location, the estimates of A and B may be computed, rather than searching over all possible values. Again, due to the continuous nature of the parameters α and β, a truly exhaustive search of the manifold is impossible. The discrete search is implemented, searching the integers for the estimates of α and β. The results are presented in Section A.3.

A.3 Results

Unfortunately, no closed form expression exists for a sufficient statistic arising from the above techniques. Even with perfect estimators, the test can be expressed by:

$$\frac{\left\|\mathbf{R} - \hat{\mathbf{X}}^{\mathrm{ML}_0}\right\|^2}{\left\|\mathbf{R} - \hat{\mathbf{X}}^{\mathrm{ML}_1}\right\|^2} \underset{\mathcal{M}_0}{\overset{\mathcal{M}_1}{\gtrless}} \gamma \tag{11}$$

The denominator measures the distance from the observed data to the manifold $\mathbf{E}$. The numerator measures the distance to a subset $\mathbf{E}_0$ of this manifold consisting of blocks where $A = B$. Note that this quotient is always at least one, since $\mathbf{E}_0 \subset \mathbf{E}$.

The unknown noise variance seems to play a part in both the numerator and denominator. This leads to speculation that the detector may have constant false-alarm rate (CFAR). Scaling the observed data by a constant would yield the same statistic because (with fixed edge location)

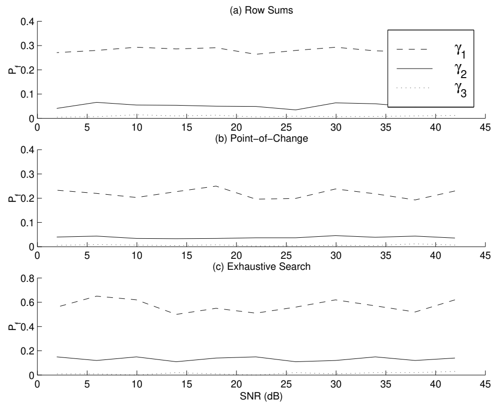


Figure 19: P_f vs. SNR for Various Techniques, Thresholds

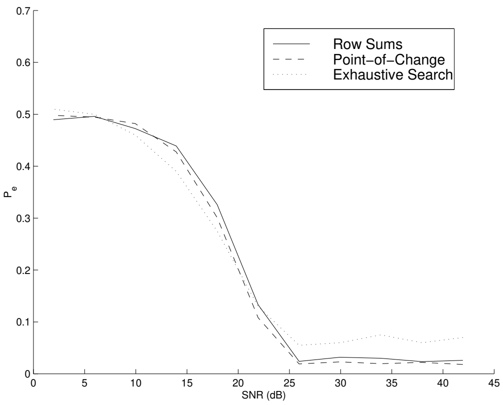


Figure 20: P_e vs. SNR for Threshold γ_2

the manifolds are linear in the intensity parameters A and B. Thus, it is believed that, with good estimators, the false-alarm rate will be invariant to the signal-noise ratio.

Unfortunately, due to the complicated shape of the manifold, it seems that analytic expressions for error probabilities cannot be obtained. We turn instead to simulation results to obtain these probabilities.

A.3.1 Detection Performance

For model 1, we create an 8×8 block containing a single edge. The parameter values are as follows: $A = 120$, $B = 130$, $\alpha = 1.6$, $\beta = 6.4$. Model 0 is constructed with the same signal energy by giving all pixels an intensity of $A = 125.1$.

Three values of γ are chosen for evaluation: $\gamma_1 \approx 1.45$, $\gamma_2 \approx 2.40$, and $\gamma_3 \approx 3.96$. The false-alarm probabilities for each of our algorithms are shown in Figure 19. As expected, the false-alarm rate is roughly constant with the SNR, and higher thresholds yield fewer false-alarms. In addition, simulations confirm that the detection probability increases with the threshold. Under the assumption that $\pi_0 = \pi_1 = 0.5$, the overall error probability P_e can be computed. We choose γ_2 as a reasonable threshold, and compare the performances of the three algorithms in Figure 20. For input SNR below about 25dB, all three techniques seem to offer roughly the same error probability. At higher SNR, the Exhaustive Search makes more errors than the Row Sums and Point-of-Change techniques. As will be shown in the next section, this is likely due to the limited precision of the search.

A.3.2 Estimation Performance

In the noise-free case, both the Row Sums technique and the Point-of-Change technique estimate the precise position of the edge. The Exhaustive Search techniques comes close; our approach, however, limits the precision of $\hat{\alpha}$ and $\hat{\beta}$ to the integers.

We thus investigate the performance of these techniques in the presence of noise. A single clean edge is constructed in an 8×8 block, and various levels of noise are added. The edge positions and mean values for the noisy blocks are estimated; these are combined to form a block based on the estimated mean field $\hat{\mathbf{X}}^{\mathrm{ML}_1}$. The SNR is computed between this estimated block and the original

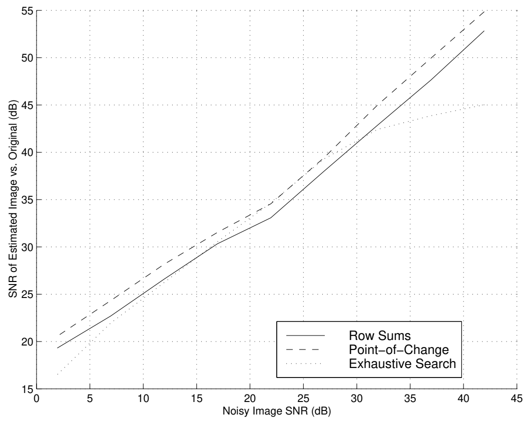


Figure 21: Performance of Estimation Techniques

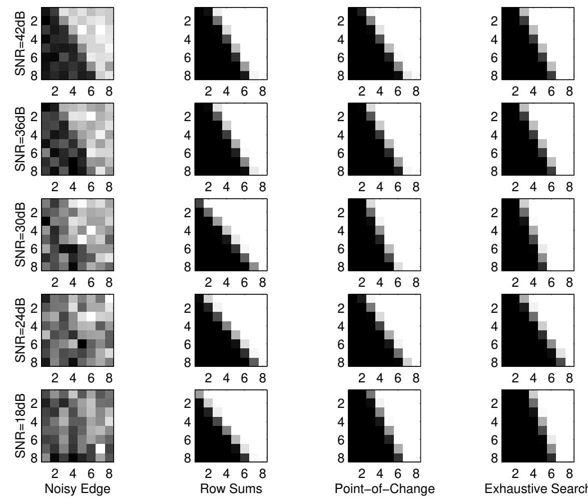


Figure 22: Estimation of Ideal Edge in Noise

noise-free block. A number of trials are run with the same noise power, and the average estimation SNR is computed (the average is computed from the sum of the SNR's, not the sum of the MSE's). Figure 21 plots the input SNR vs. the estimation SNR. The process can be viewed as a type of noise removal where the image is known to contain a single noise-free edge. Under this interpretation, the estimation procedures improve the image by an average of 10-12 dB. Examples of the estimated edge locations are shown in Figure 22.

The Point-Of-Change algorithm consistently outperforms the Row Sums technique by 2-3dB. Thus, its estimates are believed to be closer to the original edge. The Exhaustive Search performs quite well at lower input SNR but loses out at higher SNR due to its limited precision. However, over the entire range of tested SNR, the Point-Of-Change algorithm seems to give superior performance to the other techniques.

A.3.3 Examples from Natural Images

Eight blocks from the familiar Lena image are examined. Figure 23 shows the original locations of the blocks. Some blocks (#1 - #4) are taken from regions with apparent edges. Block #5 is taken from a smooth region on the shoulder. Blocks #6 and #7 are taken from regions with slight gradients, and block #8 seems to have a diagonal edge passing through with roughly equal mean on both sides.

Figure 24 shows the detection/estimation results. Each detector chooses Model 1 for every block except #5. For block #5, the Exhaustive Search detects an edge while the other two techniques do not. Thus, the detectors seem to be very sensitive to the presence of edges or gradients in the block. The detection of edges in blocks #6 and #7 could arguably be called false-alarms, but the estimated edge locations seem to match well with the context in the Lena image. While the Exhaustive Search seems especially sensitive to false-alarms, it does the best job of estimation in block #8.

Figure 23: Analysis Blocks from Lena Image (256×256)

References

[1] X. Li, *Spatially Adaptive Statistical Modeling and its Applications to Image Processing*, Ph.D. thesis, Princeton University, March 2000.

[2] A. Sabharwal and L. C. Potter, "Convexly constrained linear inverse problems: Iterative least-squares and regularization," *IEEE Trans. Signal Processing*, To Appear.

[3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.

[4] S. H. Park and D. S. Kim, "Theory of projection onto the narrow quantization constraint set and its application," *IEEE Trans. Im. Proc.*, vol. 8, no. 10, pp. 1361–1373, October 1999.

[5] Z. Cvetković and M. Vetterli, "On simple oversampled A/D conversion in $L^2(\mathbb{R})$," *IEEE Transactions on Information Theory*, vol. 47, no. 1, pp. 146–154, January 2001.

[6] P. Abry and P. Flandrin, "On the initialization of the discrete wavelet transform algorithm," *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 32–34, February 1994.

[7] D. H. Johnson, *ELEC 530: Detection Theory*, Rice University, 2001, Course Notes.

[8] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, PTR Prentice Hall, Englewood Cliffs, NJ, 1993.

Special thanks to Hyeokho Choi for many discussions about wavelets, prediction, and interpolation, and to Justin Romberg for help in developing the Row Sums technique for edge estimation.

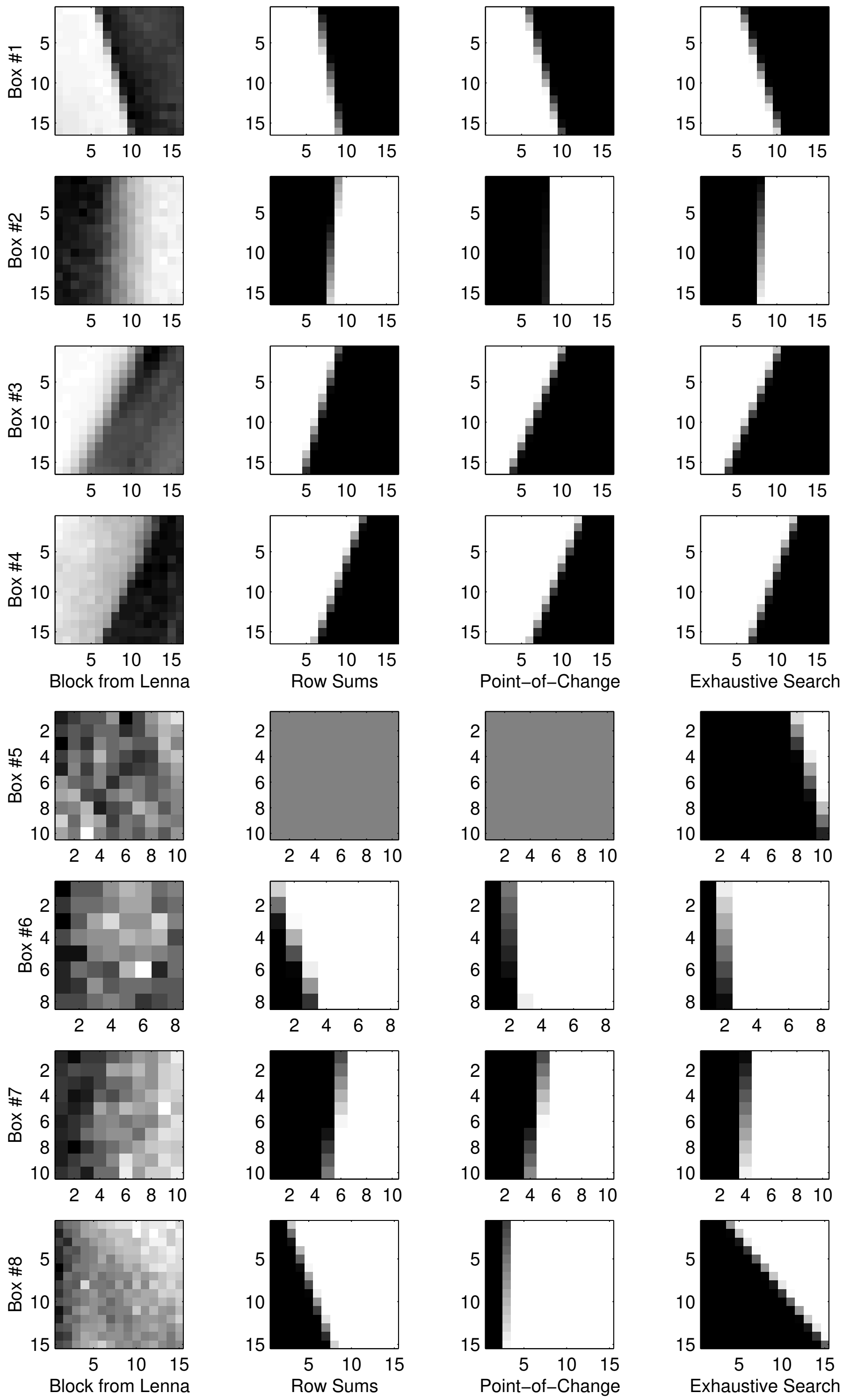


Figure 24: Detection/Estimation Results for Lena Analysis