

# **Combined Industry, Space and Earth Science Data Compression Workshop**

Prepared for

**National Aeronautics and  
Space Administration**

**Ball Aerospace and  
Technologies Corporation**

by

**Jet Propulsion Laboratory  
California Institute of Technology  
Pasadena, California**

*Proceedings of a workshop held at  
the Snowbird Conference Center  
Snowbird, Utah  
April 4, 1996*

*JPL Publication 96-11*

# **Combined Industry, Space and Earth Science Data Compression Workshop**

Aaron B. Kiely and Robert L. Renner, *Editors*

*Proceedings of a workshop held at  
the Snowbird Conference Center  
Snowbird, Utah  
April 4, 1996*

Prepared for

**National Aeronautics and  
Space Administration**

**Ball Aerospace and  
Technologies Corporation**

by

**Jet Propulsion Laboratory**  
California Institute of Technology  
Pasadena, California

**Page intentionally left blank**

## FOREWORD

The sixth annual Space and Earth Science Data Compression Workshop and the third annual Data Compression Industry Workshop were held as a single combined workshop. The workshop was held April 4, 1996 in Snowbird, Utah in conjunction with the 1996 IEEE Data Compression Conference, which was held at the same location March 31 - April 3, 1996.

The Space and Earth Science Data Compression sessions seek to explore opportunities for data compression to enhance the collection, analysis, and retrieval of space and earth science data. Of particular interest is data compression research that is integrated into, or has the potential to be integrated into, a particular space or earth science data information system. Preference is given to data compression research that takes into account the scientist's data requirements, and the constraints imposed by the data collection, transmission, distribution and archival systems.

Topics of interest for the Data Compression Industry sessions include but are not limited to: multimedia (text, images, video, speech, music, maps, graphics, animation), medical imagery, security applications, wavelets, compressed imagery through low data communication paths. All proposed topics address operating issues such as MIPS/MFLOPS required, throughput, hardware platforms, performance, etc. "This is applied compression."

### WORKSHOP ORGANIZERS

Robert L. Renner  
Ball Aerospace and Technologies Corp.  
TT-5  
P.O. Box 1062  
Boulder, CO 80306-1062  
(303) 939-4599, fax: (303) 939-6024  
e-mail: rrenner@ball.com

Dr. Aaron B. Kiely  
Jet Propulsion Laboratory  
Mail Stop 238-420  
4800 Oak Grove Drive  
Pasadena, CA 91109  
(818) 393-9048, fax: (818) 354-6825  
e-mail: aaron@shannon.jpl.nasa.gov

#### Workshop Committee Members:

Ms. Sherry Chuang, NASA Ames Research Center  
Dr. Sam Dolinar, Jet Propulsion Laboratory  
Prof. V. John Mathews, University of Utah  
Dr. James Tilton, NASA Goddard Space Flight Center

*This publication was prepared by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.*

**Page intentionally left blank**

# CONTENTS

Foreword .....	iii
Contents .....	v

## PAPERS

Wavelet-based Compression for Interactive Progressive Image Transmission .....	1 - 1
<i>Timothy S. Wilkinson</i> The Aerospace Corporation	
The Effects of Transmission Errors on Wavelet Compressed Imagery .....	11 - 2
<i>James H. Kasner, Shankar Venkataraman, and Paul Farrelle</i> Optivision, Inc.	
Low Bitrate Compression With Downsampling and JPEG .....	21 - 3
<i>Kristo Miettinen</i> Eastman Kodak Company	
The Evaluation of Low Bit Rate and Multi-Component Image Compression Algorithms for Inclusion in the National Imagery Transmission Format Standard (NITFS) .....	31 - 4
<i>Jill McCabe, Kristo Miettinen, and Joseph Reitz</i> Eastman Kodak Company	
Live Video Communication over Computer Networks Using MPEG .....	41 - 5
<i>Ciro Aloisio Noronha, Jr. and Feiling Jia</i> Optivision, Inc.	
Global and Local Distortion Inference During Embedded Zerotree Wavelet Decompression .....	51 - 6
<i>A. Kris Huber and Scott E. Budge</i> Utah State University	
Experimental Studies on a Compact Storage Scheme for Wavelet-Based Multiresolution Subregion Retrieval .....	61 - 7
<i>A. Poulakidas, A. Srinivasan, O. Egecioglu, O. Ibarra, and T. Yang</i> University of California, Santa Barbara	
JPEG Lossless Image Compression and High Compressibility .....	71 - 8
<i>Xiaolin Wu<sup>†</sup>, Nasir Memon<sup>‡</sup>, Glen Langdon<sup>§</sup>, Don Speck<sup>§</sup>, Theodore Goodman<sup>§</sup>, and Jim Spring<sup>§</sup></i> <sup>†</sup> University of Western Ontario, Canada; <sup>‡</sup> Northern Illinois University; <sup>§</sup> University of California, Santa Cruz	

Compression of Aerial Ortho Images Based on Image Denoising .....	81	-9
<i>A. Langi and W. Kinsner</i>		
University of Manitoba and TRILabs, Canada		
Wavelet Coding for Remote Sensed Data .....	91	-10
<i>Zia-ur Rahman</i>		
Science and Technology Corporation		
Phase-preserving Compression of Seismic Data using the Self-adjusting Wavelet Transform .....	101	-11
<i>H. Guo and C. S. Burrus</i>		
Rice University		
The Effects of Wavelet-based Data Compression on Flat Field Calibration for Remote Sensing Applications .....	110	-12
<i>Susan P. Hagerty<sup>†</sup> and Jerome M. Shapiro<sup>‡</sup></i>		
<sup>†</sup> Ball Aerospace and Technologies Corp., <sup>‡</sup> Aware, Inc.		
Perceptually Lossless Wavelet Compression .....	120	-13
<i>Andrew B. Watson<sup>†</sup>, Gloria Y. Yang<sup>‡</sup>, Joshua A. Solomon<sup>†</sup>, and John Villasenor<sup>‡</sup></i>		
<sup>†</sup> NASA Ames Research Center, <sup>‡</sup> University of California, Los Angeles		

Wavelet-based Compression for Interactive Progressive Image Transmission  
Timothy S. Wilkinson

The Aerospace Corporation  
2350 E. El Segundo Blvd., El Segundo, CA 90245-4691  
e-mail: wilk@avila.aero.org

9-82  
013172  
268895  
101.

## 1. ABSTRACT

The wavelet transform is nicely suited to image compression applications. The nature of the transform permits quantization and coding that provide reconstructions with quality superior to that of many other techniques at equivalent bit rates. Superiority of the approach appears to be especially evident at very low compressed rates. The simultaneous localization of both spatial and frequency content inherent in the standard Mallat decomposition allows successful implementation of coding strategies over a wide range of complexity.

In this paper, a simple technique is described which exploits the structure of the wavelet decomposition to permit the progressive transmission of images on an interactive basis. This technique is ideally suited to image transmission in a limited communications bandwidth environment. It may also be applied to image archive and browsing systems. A summary of the system is as follows. A scalar quantization and entropy coding scheme is used to produce a database of available imagery. Users submit requests for imagery to the database via a graphical user interface. Upon initial request, a low resolution version of the image, which corresponds to the lowest resolution coefficients in the wavelet decomposition, is transmitted and reconstructed. The user can then isolate specific regions of interest within the image and request additional levels of detail. If all levels of detail are sent, the transmitted image is visually indistinguishable from the original. Processing requirements are outlined and the efficiency of the progressive transmission strategy is examined as a function of communications bandwidth and processor speed.

## 2. INTRODUCTION

The wavelet transform, pioneered by the works of Mallat [1] and Daubechies [2,3], has become a useful tool for many problems in signal processing. For grayscale images it provides a convenient framework for exploiting the spatial localization of image detail at many scales. Such exploitation might take the form of image compression, in which the various decomposition levels are represented with different fidelity. It might also involve multi-resolution analysis, in which different spatial regions are examined at different resolutions.

An application that spans both arenas is *interactive progressive image transmission*. For purposes of this paper, progressive image transmission is defined as the transmission of an image such that certain regions may be represented with higher spatial resolution than others. Furthermore, transmissions are ordered from lowest to highest resolution within a region so that the final product is built or refined in a layered fashion. A desirable property

is that the maximum resolution representation be, at minimum, visually indistinguishable from the original image.

In this paper it will be shown that simple wavelet transform-based image compression provides an excellent framework for progressive transmission. Furthermore, the spatial localization of the transform can be exploited to make the progressive transmission interactive in a client/server environment. While useful for archive/browsing architectures, the application seems most suited to image transmission over severely limited communications channels. Necessary computational resources are discussed in relation to available communications bandwidth.

### 3. WAVELET IMAGE COMPRESSION AND PROGRESSIVE TRANSMISSION

Many references address wavelet-based image compression. A brief overview is presented here. For more in-depth treatments, the reader is referred to [5].

In the basic wavelet compression architecture, the pyramid decomposition of Mallat, as shown in figure 1, is performed as the first step in the process. Low-pass and high-pass filters are applied (separably) in each direction with the results down-sampled by a factor of 2 in each direction. At the image boundaries, the finite extent of the convolution is typically handled by reflection. This results in a transform with a number of elements equal to the number of pixels in the original image. The decomposition is typically repeated upon the all low-pass branch as permitted by the original image size and filter length.

The second step in the compression process is typically a resolution-dependent quantization. In general, the wavelet transform isolates the most crucial image information at the lower resolution levels of the decomposition pyramid. Thus, quantizers for those levels typically have smaller widths and therefore introduce relatively little distortion. Transform coefficients at higher resolutions tend to be less important in reconstructing the image, and thus the corresponding quantizers are typically wider. Quantizers can be scalar or vector, they can be uniform or variable-width, and they can be fixed across a decomposition level or spatially adaptive. Unless the quantization strategy is image independent, sufficient information regarding the quantizer must be stored to allow reconstruction during image expansion.

In the final step of the compression process, one typically performs entropy reduction coding on the quantized coefficients. Run length coding is often employed to reduce redundancy in areas of little detail. Huffman or arithmetic coding is then applied to perform final entropy reduction. In either case, sufficient information regarding the code must be stored to permit decoder reconstruction, i.e., the code itself, a sufficient description of the probability model, etc.

The reconstruction process is a straightforward inversion of the compression steps. The entropy code is undone to produce the quantized coefficients. The de-quantized coefficients

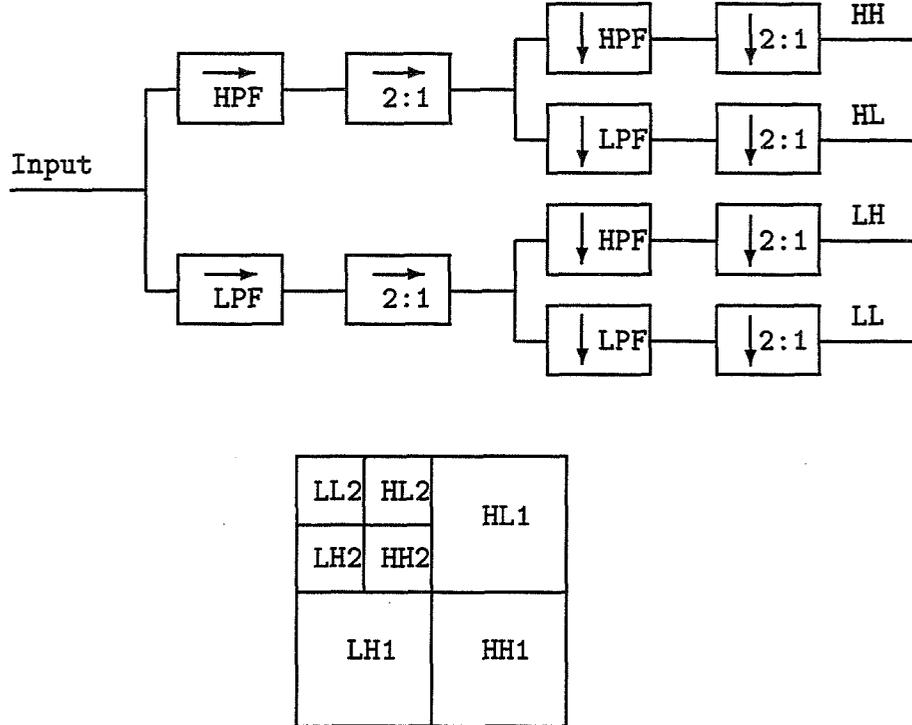


Figure 1: Wavelet decomposition flow diagram and pyramid structure.

are produced from knowledge of the quantization process. Finally, the inverse wavelet transform is applied as in figure 2 to produce the expanded image.

This hierarchical image representation naturally provides a mechanism for constructing a progressive transmission system as defined in section 2. In particular, a layered approach to increasing image resolution is easily achieved. To make this more specific, consider the two-level decomposition illustrated in figure 1. Initially, a low resolution version of the image is available in the form of sub-image LL2. This sub-image roughly represents the original image degraded in resolution by a factor of 4 in each direction. Even though sub-image LL1 is not directly present in the pyramid, it can be obtained by appealing to the recursive nature of the wavelet decomposition and inverting the transform used to obtain sub-images LL2, LH2, HL2, and HH2. If sub-image LL2 is already on hand, only the the additional detail sub-images must be transmitted. A similar recursion allows LL0, the original image, to be obtained from the reconstructed LL1 and all of the high-pass details at that level.

Proof that multiple resolutions are available within different reconstructed regions requires only the observation that the decomposition involves filters of a finite length. More specifically, consider a particular region of size  $w$  by  $h$  pixels in the original image. Using filters of length  $L$ , that area is mapped to a region of size  $\frac{(L+w-1)}{2}$  by  $\frac{(L+h-1)}{2}$  samples in each of the four resulting sub-images. No values outside the resulting regions are required to reconstruct the original higher resolution area. Again appealing to the recursive nature of the decomposition, the same argument can be extended to account for regions of an image

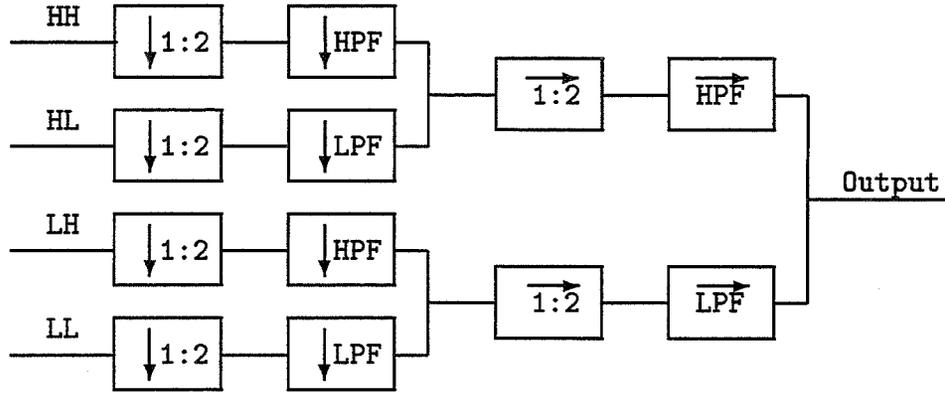


Figure 2: Inverse wavelet transform flow diagram.

as they propagate down the transform pyramid. Thus multiple image regions can be reconstructed at differing resolutions. Another way to visualize this process is to consider that one has the entire wavelet decomposition available. Selectively setting coefficients equal to zero results in a loss of some resolution level in a given region. The effect of this operation is the inverse of progressive transmission, where initially all coefficients are unknown and the receiver steadily and selectively gains more knowledge of their values.

Since the compression process can be separated from the transmission process, the final part of the system definition can be satisfied. In particular, it is possible to determine compression system parameters that provide a visually lossless result when the maximum level of detail in a given region is requested. Some of the issues involved in selecting compression options to facilitate progressive image transmission are discussed in the next section.

#### 4. TAILORING COMPRESSION TO PROGRESSIVE TRANSMISSION

Superior compression performance is generally obtained with more sophisticated components in the description above. For example, arithmetic coders are known to outperform Huffman coders in most circumstances, though they are comparatively somewhat more complicated to implement. Progressive transmission presents a number of operational and architectural requirements that favor simple or uncommon strategies for compression. This section addresses several aspects of the specific choices in the compression architecture that were made to facilitate progressive transmission.

##### 4.1. Filtering and Decomposition

As shown in section 6, the time required to update a region in this progressive transmission architecture is directly proportional to the length of the reconstruction filter(s). Thus, this system uses short convolution kernels, typically the Daubechies 4 of 6-tap filters. While superior compression performance can be obtained with other kernels, such as the bi-orthogonal 7-9 kernel, these advantages are typically realized at compressed rates much lower than those required for visually lossless reconstruction.

An unconventional choice is preferred regarding handling of image edges. The system typically operates on either 1024x1024 or 512x512 pixel image chunks. Padded values of zero are added to the edges to accommodate the finite convolution extent. This allows extremely large images to be blocked into chunks and compressed independently with minimal processing impact on the reconstruction system. If a desired region spans two chunks, the linearity of the wavelet transform allows addition of the required coefficients after proper alignment. This in turn allows a single convolution sequence to reconstruct the adjacent regions from the two chunks. Note that if reflection or circular wrapping logic is applied during the decomposition, independent convolutions must be applied to extract each of the adjacent regions separately. Padding the image with zeros does imply additional storage cost. For a chunk of  $w$  by  $w$  pixels and a filter length of  $L$  taps, roughly  $2Lw$  additional transform samples must be stored than in the edge reflection case. Again, when targeting visually lossless compressed rates, this choice does not impose a major storage penalty.

#### 4.2. Quantization

In this application, a uniform scalar quantizer is preferred to others for a variety of reasons. First, the uniform scalar quantizer can be communicated with a single parameter - the bin width. De-quantization is performed by a simple multiplication, or if table look-up is desired, a non-iterative process can be used to construct the table. A Lloyd-Max quantizer, in contrast, would require either iterative construction during de-quantization, or storage of a large library of pre-computed tables: either choice would place additional burden on the expander.

A vector quantizer might provide some performance gains but would suffer two drawbacks: the codebook for each decomposition level (or region) would have to be transmitted as overhead sometime during system operation, and additional expander logic would be required to extract only the desired coefficients from the vector quantized blocks. A state-dependent quantizer, such as a trellis-coded quantizer, would imply that state information be transmitted as required, e.g., at the end of each row of a region, to resynchronize the decoder state. For small rectangular regions, this resynchronization information would comprise a relatively large fraction of the transmitted packets.

#### 4.3. Entropy Coding

All decomposition levels (other than the all low-pass sub-image) are assumed to have coefficients that obey a Laplacian distribution. The Laplacian parameter is easily estimated by the absolute mean of the coefficients. Furthermore, given the quantizer width, a Huffman code that matches the expected coefficient distribution can easily be constructed at both the compressor/server and the expander/client. Communicating the entropy code therefore requires only a single parameter. At the relatively high compressed bit rates required for visually lossless compression, rate control using this quantizer is simple due to consistently accurate estimation of the coded bit rate.

Stable compressor rate control is also a factor that influences the choice against run-length

coding. A more critical factor, however, is that 'interesting' image regions, those most likely to be updated in a progressive fashion, typically contain few coefficients equal to zero. Furthermore, even in the presence of some zero coefficients, the run-length coder would likely benefit the server only, since updated regions are typically much smaller in size than the original image, and run-length coding benefits would therefore be proportionally reduced.

## 5. APPLICATIONS

Two major applications are envisioned for interactive progressive image transmission. The scheme fits nicely with image archive and browse architectures. It also directly supports image transmission over low bandwidth channels. Each of these applications is addressed briefly in the sections that follow.

### 5.1. Archive/Browse

Under this scenario, the image is compressed to a relatively high target rate, perhaps 2.5 to 3.0 bits per pixel, so that visually lossless, archive quality is achieved. (A residual coding scheme could be employed to allow fully lossless reconstruction.) The compressed images are available via server, and the transmission rate is assumed to be very high - local area network, ethernet, or higher rates.

The user can browse 'thumbnails' of the available imagery simply by viewing the all low-pass version of each available set. When an image of interest is found, the desired image can be requested at any resolution  $\frac{G}{2^{i-1}}$ ,  $i = 1, \dots, N$ , where  $G$  is the original resolution of the image and  $N$  is the number of levels in the wavelet decomposition. The required resolution levels are then provided to the browser where the wavelet transform is inverted. Alternatively, the need for decoding software at the browser can be eliminated by allowing the server to perform the inverse transform and then transmit the result.

In either case, network transmission is minimized by providing the image only at the resolution required by the browser. The emphasis in this application, however, is on the compressed image and its format. The wavelet compression process on its own significantly reduces the storage costs associated with the archive. At the same time, it naturally provides a sequence of lower resolution overviews to aid users in selecting a piece of data appropriate to their needs.

### 5.2. Limited Communications Bandwidth

Suppose that the user at the end of a low bandwidth communications link (perhaps as low as 2.4kbits per second in some problem environments) desires useful imagery in as little time as possible. Transmission of an entire high resolution scene at full dynamic range would be prohibitive due to time constraints. The following strategy might dramatically improve the situation. To begin, send a low resolution version of the image; this corresponds to transmitting the all low-pass portion of the wavelet transform. If the wrong image is

transmitted, little time has been lost because the low-pass portion of the wavelet pyramid is small relative to the original image size. If the image appears correct, the client can request another level of detail for any image subsection. In that case, the appropriate subset of transform coefficients is extracted from the compressed file by the server and transmitted to the client. The client then inverts the wavelet transform only for the requested region and updates that portion of the image to the next higher resolution. The process can be continued until the desired level of fidelity is achieved in all areas of interest.

In this scenario, the emphasis of the application is on the compression savings and the spatial localization properties of the wavelet transform. Exploiting these properties can result in a reduction in the time required to transmit and receive a useful image product. The exact nature of the savings is determined by the specifics of the transform, the transmission link, and the client's processing capabilities as illustrated in the following section.

## 6. COMMUNICATIONS/PROCESSING TRADE

An image of a certain width and height is transmitted across a link of fixed rate. The initial transmission is a low resolution version of a large image. A user selects a region of interest in the low resolution version, and requests it at twice its current resolution. The request can be honored in one of two ways. Raw pixel data for the entire requested region at the higher resolution can be transmitted. Alternatively, wavelet progressive transmission can be used so that the higher resolution piece is obtained by filtering operations applied to the current piece and a transmission of additional image detail. The preferred method is to be determined based upon minimal time consumption.

In the development that follows, the following definitions will be used:

- $L$  = length of decomposition filter, pixels
- $B$  = precision of original data, bits/pixel
- $R$  = transmission link rate, bits/second
- $F$  = processor rate in floating point operations (flops)/second

It will be assumed that the computational cost of inverting the wavelet transform dominates the processing time for the client. Note that this assumption neglects any cost of memory management that the client must perform. Memory management involving the image display is likely to be roughly equal for each of the approaches. Memory management costs associated with input/output of received transmission are also likely to be roughly equivalent. For large images, however, memory management in computing the inverse wavelet transform may be non-negligible. Analysis of these effects would require specific knowledge of the implementation platform and are beyond the scope of this paper.

Now assume that a region of size  $\frac{w}{2}$  by  $\frac{h}{2}$  is present and it is desired to increase the resolution by a factor of two. The resulting image will be  $w$  by  $h$  pixels in size. Each of the resolution enhancement strategies is now analyzed in terms of total time required.

### 6.1. Increased resolution from raw pixel data in requested region

In this case, raw data for each of the desired  $wh$  pixels must be transmitted. This operation will require a time

$$t_1 = \frac{whB}{R} \quad (1)$$

seconds. The time required to display the received image is neglected and assumed equal for both strategies.

### 6.2. Increased resolution from decomposed data and inverse filtering

It is first necessary to derive an expression for the total number of operations required to invert a single level of the wavelet transform as illustrated in figure 2. In this development, it is assumed that  $w, h \gg L$ , so that additional operations due to the finite convolution extent are negligible compared to those required for the central region. In the column filtering stage of reconstruction, all four detail sub-images of width  $\frac{w}{2}$  are upsampled to an output width of  $w$ . Computation of each output point requires only  $\frac{L}{2}$  multiplications and  $\frac{L}{2}$  additions, since half of the upsampled points are known to be zeros. Since the output width of this stage is  $\frac{w}{2}$ , a total of  $4(\frac{L}{2} + \frac{L}{2})\frac{w}{2}h = 2Lwh$  operations are required to perform the inverse column filtering. Pairs of the resulting data streams are added together at this point, requiring a total of  $wh$  additions.

Now the row filtering operations begin. By similar analysis, the number of operations need to perform the high-pass and low-pass row filtering is  $2Lwh$  flops. Finally, the two filter outputs of width  $w$  and height  $h$  are added, requiring  $wh$  flops, to form the higher resolution output. The total number of operations to reconstruct the initial image from its first decomposition level is then:

$$2Lwh + wh + 2Lwh + wh = 2(2L + 1)wh. \quad (2)$$

Since the wavelet decomposition is a floating point operation, the difference in transmitted coefficient precision must be addressed. In the progressive transmission architecture as described, each decomposition level is quantized so that a given target bit rate is achieved. Thus exact analysis of the number of bits required to send the required coefficients is difficult. For convenience, it is assumed here that the resulting decomposition coefficients are truncated to the precision of the original image. The analysis is therefore pessimistic regarding the time required to transmit the required wavelet data. (This discrepancy is addressed in the following section.)

One of the four sub-images of size  $\frac{w}{2}$  by  $\frac{h}{2}$  that will be filtered is already present. Three sub-images must therefore be transmitted, each with a precision of  $B$  bits. The time for transmission is

$$t_{2,trans} = \frac{3whB}{4R} \quad (3)$$

seconds. After reception, the inverse filtering for one level must be performed. From the development above, this will require  $2(2L + 1)wh$  flops and require

$$t_{2,flt} = \frac{2(2L + 1)wh}{F} \quad (4)$$

seconds. The total time for this method is then

$$t_2 = t_{2,trans} + t_{2,flt} = \frac{3whB}{4R} + \frac{2(2L + 1)wh}{F} \quad (5)$$

seconds.

### 6.3. Timing comparison

The methods require equal time when

$$\begin{aligned} \frac{whB}{R} &= \frac{3whB}{4R} + \frac{2(2L + 1)wh}{F} \\ \frac{B}{R} &= \frac{3B}{4R} + \frac{2(2L + 1)}{F} \\ \frac{B}{4R} &= \frac{2(2L + 1)}{F}. \end{aligned} \quad (6)$$

If

$$\frac{B}{4R} \leq \frac{2(2L + 1)}{F}, \quad (7)$$

then the transmission of raw pixel data is time optimal: otherwise, the filtering method is preferred. This result is independent of the number of pixels desired and depends only on the link rate, the data precision, the length of the reconstruction filter, and the processor speed.

This relation shows that under conditions of severely limited bandwidth or abundant processing power, a progressive transmission system based on image subbands is preferable to transmission of raw data if overall speed is the only factor that must be optimized. To make the illustration more concrete, assume typical values for image dynamic range  $B$  of 8 bits and filter length  $L$  of 6 taps. Then the ratio of progressive transmission time required to raw transmission time required is

$$\frac{t_2}{t_1} = \frac{3}{4} \left( 1 + \frac{13R}{3F} \right). \quad (8)$$

The factor of  $\frac{3}{4}$  at the front of this expression can be traced to the fraction of samples transmitted in the progressive transmission case relative to the raw case. Had the analysis taken into account the compression savings achieved by the approach, this factor would instead be  $\frac{3}{4}\alpha$ ,  $\alpha < 1$ , so that

$$\frac{t_2}{t_1} = \frac{3}{4}\alpha \left( 1 + \frac{13R}{3F} \right), \quad (9)$$

with a typical value of  $\alpha \approx \frac{1}{3}$  corresponding to 3:1 compression.

The above expression provides two insights. First, in order for progressive transmission to make sense, the time ratio must be smaller than 1. This is achieved when  $\frac{F}{R} > 13$ . This is generally true for even modest desktop computers when the communications link rate is near modem rates. (For example, a 1 megaflop machine combined with a 38.4kbps link provides  $\frac{F}{R} \approx 25$ .) Second, the ultimate time savings given infinite processing speed is limited by the pyramid structure and the compression ratio to a factor of  $\frac{4}{3\alpha}$ . Under these conditions, assuming approximately 3:1 compression, the maximum savings in time is a factor of 4. Such savings can be critical in tactical applications, or can make imagery-based products more appealing in consumer applications by substantially decreasing response latency.

## 7. ACKNOWLEDGMENTS

This material is based upon work supported in part by the Aerospace Sponsored Research Program.

## 8. REFERENCES

- [1] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pat. Anal. Mach. Intel.*, 11(7):674–693, July 1989.
- [2] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, 1988.
- [3] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Info. Thy.*, 36(5):961–1005, September 1990.
- [4] W. R. Zettler, J. Huffman, and D. C. P. Linden. Application of compactly supported wavelets to image compression. In *Image processing algorithms and techniques*, volume 1244, pages 150–160. SPIE, 1990.
- [5] R. A. DeVore, B. Jawerth, and B. J. Lucier. Image compression through wavelet transform coding. *IEEE Trans. Info. Thy.*, 38:719–746, March 1992.

52-82  
613173  
068896  
108

## The Effects of Transmission Errors on Wavelet Compressed Imagery

James H. Kasner, Shankar Venkataraman, and Paul Farrelle  
Optivision Inc.  
1480 Drew Avenue, Suite 130  
Davis, CA 95616  
e-mail: {kasner,shankar,farrelle}@optivision.com

### Abstract

*The discrete wavelet transform has proven to be an effective technique for the compression of digital imagery at very low rates. This has led to increasing interest in commercial, international, medical, and governmental arenas in the wide-spread use and standardization of this technology. As the popularity of wavelet based image compression systems grows, issues regarding the transmission of wavelet-compressed imagery must be addressed. The analysis, correction, and containment of bit errors in the transmission of wavelet-compressed imagery represents a challenging area of research.*

*The focus of this paper is to investigate transmission error effects in an adaptive wavelet image coder that utilizes universal trellis-coded quantization (UTCQ) [1, 2] and adaptive arithmetic coding [3 - 5] to generate coded bitstreams, and propose a re-initialization mechanism for error recovery. Bit error rates (BERs) in the  $10^{-5}$  to  $10^{-3}$  range are considered and the visual manifestations of bit errors in the reconstructed image are shown. The usage of a trellis quantizer complicates error containment since corrupted quantization indices may lead to loss of tracking of the trellis state and propagation of decoding error. Comparisons to NITFS-JPEG, where restart intervals are mandated to minimize error propagation, are made.*

### 1. Introduction

Wavelet image compression systems suffer unique difficulties in a noisy environment. In addition to the customary problems associated with entropy-coded data streams, the hierarchical nature of wavelet systems can increase the effect of transmission errors in the reconstructed image. Given a wavelet filter of length  $N$ , and a dyadic pyramid  $k$  levels deep, the spatial extent of reconstructed image pixels affected by an error in a single wavelet coefficient can be shown to be  $M \times M$  pixels, where  $M = 2^k + (2^k - 1)(N - 2)$ . For an 8-tap filter pair and 5 level pyramid, an error in one of the lowest resolution wavelet coefficients will affect an area of  $218 \times 218$  pixels. Coefficients nested deep in the resolution hierarchy will have a more profound impact on the image reconstruction should an error occur there. On the other hand, the non-hierarchical JPEG modes do not suffer from this problem since the error effects are constrained to the  $8 \times 8$  DCT block in which the coefficient in error is located.

Error resiliency to transmission errors can be obtained by careful design of the compressed representation; it does not necessarily exclude any particular image representation schemes such as transform coding, vector quantization, or wavelets. Unfortunately, compression algorithms are usually designed assuming an idealized noise-free channel. A key aspect of any practical compression implementation for transmission is the interaction of channel errors with the compressed data stream. In general, transmission error effects can be considerably minimized or even, in certain cases, completely eliminated by providing error protection using error detection codes and forward error correcting codes (FEC) on the compressed bitstream. The selection and implementation of an appropriate error detection and error correction mechanism, known as channel coding, depends on the characteristics of the transmission channel. Traditional designs approach source coding and the channel coding separately. This approach is reinforced by network protocol stacks such as the ISO 7-layer stack. This layered approach allows a modular solution that can be readily reconfigured for different scenarios. However, overall throughput can be maximized when the source and channel coding are jointly designed. This is especially true for low bandwidth noisy channels rather than high bandwidth ethernet, FDDI, and ATM networks.

The National Imagery Transmission Format Standard (NITFS) [6] addresses the problem of image transmission in noisy environments. The current set of algorithms are extensions to JPEG that accommodate error detection and recovery using re-initialization points, called restart intervals, in the data stream. Depending on the application,

image regions in which errors are detected can be marked as erroneous or, in duplex transmissions, a protocol can be implemented to re-transmit such data segments. NITFS has adopted the Tactical Communications Protocol 2 (TACO2) as part of its suite. TACO2 provides a packet-based communications protocol designed to operate in arenas not addressed by other protocols [7]. Imagery sent over noisy, low-bandwidth, simplex communication channels is an example. An image compression system which provides some measure of robustness to noise is necessary, but these considerations must include the communication protocol in use.

## 2. Adaptive Wavelet-UTCQ Image Coder

Wavelet based compression has many attractive features due to its multiresolution nature. Figure 1 shows the analysis process for a one dimensional signal. A single *level* decomposition yields two outputs: the output of the high pass G filter contains high-resolution details from the signal and the output from filter H is a low pass filtered version of the input. Subsampling at the output of each filter prevents the number of output samples from growing relative to the input. The filters H and G are carefully designed to ensure perfect reconstruction, even after subsampling, in the absence of quantization errors. After the first level of decomposition, the low pass signal is then further analyzed using the same filters, as shown, until the original has been split into a desired number of subbands which are the outputs of a series of filters with logarithmically reducing pass bands.

In the case of two dimensional images, separable wavelets can be used where the one dimensional analysis described above is applied independently in the horizontal and vertical directions. After the first level of the decomposition we have four subimages which we term: LL, LH, HL, and HH where H indicates high pass (output of filter G) and L indicates low pass (output of filter H). The first letter indicates the type of filtering applied in the horizontal direction with the second relating to the vertical processing. Again the low pass image (LL) is further decomposed as shown in Figure 2 leading to a multiresolution decomposition. The values in the subarrays, known as wavelet coefficients, are then quantized, losslessly encoded and transmitted. In the adaptive wavelet based image coder under consideration here, UTCQ [1, 2] is employed to quantize the wavelet coefficients, and adaptive arithmetic coding [3 - 5] to losslessly encode the quantized coefficients. Since this algorithm is known to perform well at low bitrates, it is being considered for inclusion as a second generation bandwidth compression algorithm in the NITFS.

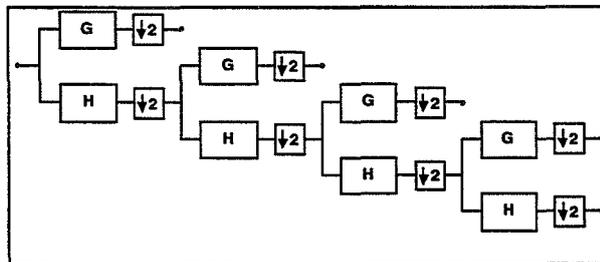


FIGURE 1. Block diagram showing the wavelet decomposition in one dimension.

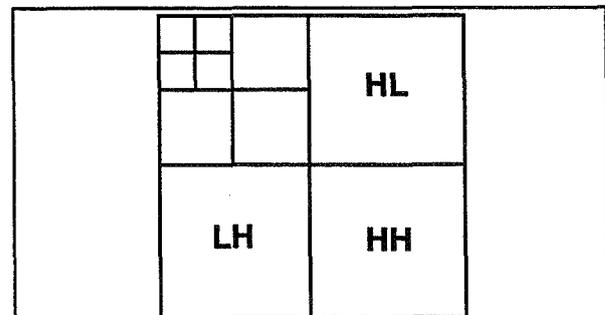


FIGURE 2. Two dimensional wavelet decomposition over 3 levels.

Trellis coded quantization (TCQ) [8] is a form of quantization where a sequence of data is quantized not sample by sample but as a whole. In this way, TCQ is a form of vector quantization. Scalar codebooks are partitioned into subsets and assigned to the branches of a trellis. The trellis limits the choice of codebook subset in a manner that allows use of a larger (greater rate) codebook than that of a scalar quantizer while maintaining the same rate. Entropy-constrained trellis coded quantization (ECTCQ) [9, 10] was developed to improve the high-rate performance of TCQ and allow easier access to non-integer coding rates. It has been shown to be an effective technique for quantizing memoryless sources with low to moderate complexity. ECTCQ achieves mean square error (MSE) performance near (within about 0.5 dB) the rate-distortion bound at all non-negative encoding rates. ECTCQ, however, requires storage of codebooks and a computationally expensive training procedure.

UTCQ was developed to address some of the implementation concerns regarding ECTCQ. UTCQ uses uniform thresholds and codewords for quantization. No stored codebooks are required. “On-the-fly” codeword training is performed for a small subset (four codewords, two on either side of the zero codeword) of the reconstruction levels during the quantization process. These trained codewords and the remaining uniform codewords are used during dequantization. It is a remarkable fact that by merely training four codewords, performance virtually identical to training all codewords is achieved. The overhead incurred by sending the trained codewords is minimal (3 bytes per quantized sequence). The quantization/dequantization process is completely characterized by the trained codewords, uniform step-size and trellis. Performance of UTCQ is within 0.1 dB of ECTCQ for most encoding rates.

Usage of stored codebooks limits ECTCQ to those rates for which codebooks were designed. UTCQ can access continuous rates for a given distribution. A simple modeling procedure must be completed for each distribution. From the viewpoint of entropy coding, UTCQ is preferable to ECTCQ. ECTCQ requires two variable-rate codes per quantized data sequence. For symmetric source distributions, UTCQ requires a single variable-rate code, reducing the number of codewords or calculations an entropy coder must have or perform.

The performance of the wavelet-UTCQ image coder can further be improved by employing classification of wavelet coefficients [4]. From observation of wavelet coefficient amplitudes it is clear that even in a single subband the wavelet coefficients are not homogeneous. Thus it may improve coder performance if the coefficients were grouped into smaller, more similar sets. To improve the performance of the wavelet-based coding, an adaptive scan technique that classifies subblocks of wavelet coefficients into four classes based on their variance is used. This allows the rate allocation to better match coefficient statistics and allocate bits where needed. In the simulations presented here, this feature is not used for the sake of simplicity.

### 3. “Spreading Error” Effect

The hierarchical nature of the wavelet representation coupled with the filtering processes used in wavelet analysis and synthesis, leads to what we call a “spreading error” effect. A given wavelet coefficient in a subband, contributes to a specific region in a reconstructed image. The spatial extent of this region is a function of the wavelet coefficient’s location in the resolution hierarchy and wavelet filter lengths. Figure 3 illustrates this concept. From a single coefficient error in a low resolution subband, as we synthesize, moving up in resolution level, the number of affected coefficients increases in number and spatial extent.

We may express the number of affected coefficients in the final reconstructed signal from a single coefficient error as follows:

$$M(0) = 1$$

$$M(k) = 2M(k-1) + N - 2 \quad (k \geq 1)$$

Where  $M(k)$  represents the number of affected coefficients  $k$  levels up from the original bad coefficient, and  $N$  is the length of the wavelet filter. A closed-form expression may be obtained as shown below.

$$M = 2^k + (2^k - 1)(N - 2) \quad (1)$$

The spatial extent of the spreading error can become very large after synthesizing up only a few resolution levels. Table 1 shows this growth assuming 8-tap filters (similar to what would be seen with the (9,7) wavelets of Daubeschies [17]).



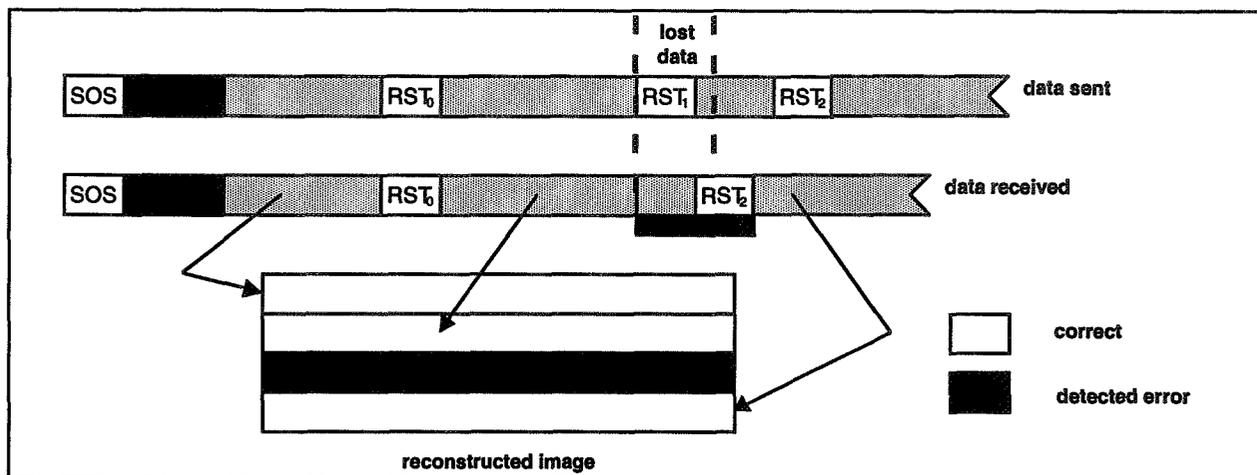


FIGURE 4. Error recovery NITFS-JPEG.

#### 4.2. Adaptive Wavelet-UTCQ Coder Without Re-initialization

Figure 5 shows the performance of the wavelet-UTCQ coder when no provisions have been made for re-initialization. The quantization indices from all subbands have been entropy-coded into a continuous bitstream. A random BER of  $6 \times 10^{-5}$  was applied to a 512 x 512 image encoded at 0.47 bpp. The header of the compressed file was not damaged, but approximately 1/3 of the way through the image, errors in the bitstream have caused the arithmetic coder to decode incorrect data. This shows the need for multiple access points into the entropy coded stream. Other simulations have shown that in general a Huffman encoded stream without restart markers will perform better than the arithmetic encoded stream. The opportunities for re-initialization by chance are more prevalent with Huffman coding.

#### 4.3. Re-Initialization in Wavelet Encoded Imagery

Since wavelet encoding is *not* a block based approach, inserting restart intervals into wavelet encoded imagery is a more complex problem than in JPEG. Note that in a block based approach it can be guaranteed that the effect of errors will not cross the restart interval boundaries. The same is not true in wavelet encoding. As a result, the error effects can be visible throughout the image and may even render the entire image useless. This is illustrated in Figures 6 and 7 where the low-frequency subband (LFS) and high-resolution, high-frequency subbands have been removed from a four level dyadic pyramid coding of Lenna at 1.0 bpp. In Figure 6 the LFS represents 1/256 of all coefficients and  $\approx 40\%$  of the compressed file. In Figure 7, 3/4 of all coefficients have been removed and a missing coefficient in the LFS has significant energy and contributes roughly to 1/4 of the reconstructed image. A missing coefficient in Figure 7 has little energy and affects a much smaller local area. Clearly errors in the LFS and lower resolution subbands must be avoided. This suggests that a hierarchical FEC procedure should be applied in the communications protocol.

Successful prevention of transmission error spread in the reconstructed wavelet encoded imagery can be accomplished by:

- locating the data segments containing wavelet coefficient in error (*error location*).
- providing a mechanism for the lossless arithmetic decoder, and the UTCQ dequantizer to correctly decode and dequantize the wavelet coefficients following those segments in error (*partial decodability*).
- completely negating the contribution of the wavelet coefficients in error by zeroing out the data segments containing them (*negation*).

In an attempt to add some measure of error resilience to the wavelet-UTCQ coder, the entropy-coded bitstream was broken at subband boundaries. Each sequence of subband quantization indices is arithmetically encoded individually and pointers placed in the file header to the start of each entropy-coded segment. Each subband of coefficients is treated as a data packet, if a bit error occurs within a data segment the subband is nulled out. This will allow the decoder to attempt recovery of the compressed image by using those subbands that do not have bit errors present. Since the trellis quantizer starts coding at the beginning at each subband, we will not need to worry about it propagating errors. Simulations were run with random bit errors at BERs in the range of  $10^{-5}$  to  $10^{-3}$ . During simulations the header was protected and no burst or add/drop bits were allowed.

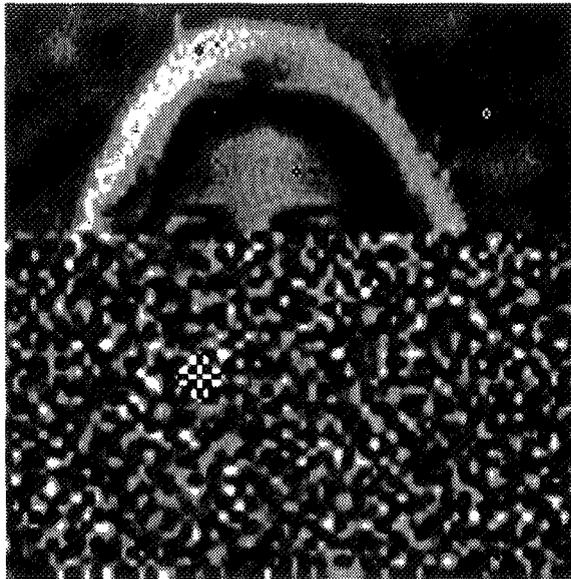


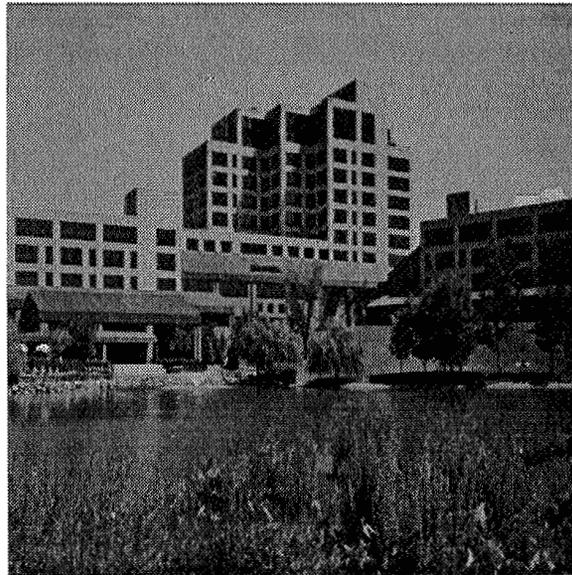
FIGURE 5. Wavelet-UTCQ, BER  $6 \times 10^{-5}$ , no re-initialization points.



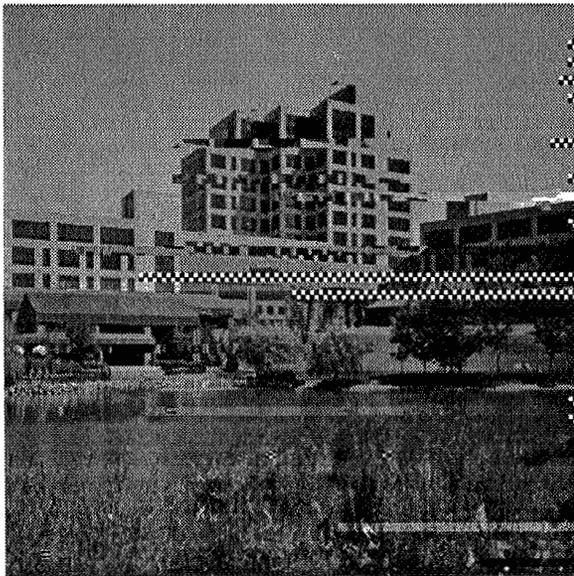
FIGURE 6. Wavelet-UTCQ, 1.0 bpp, low-frequency subbands removed.



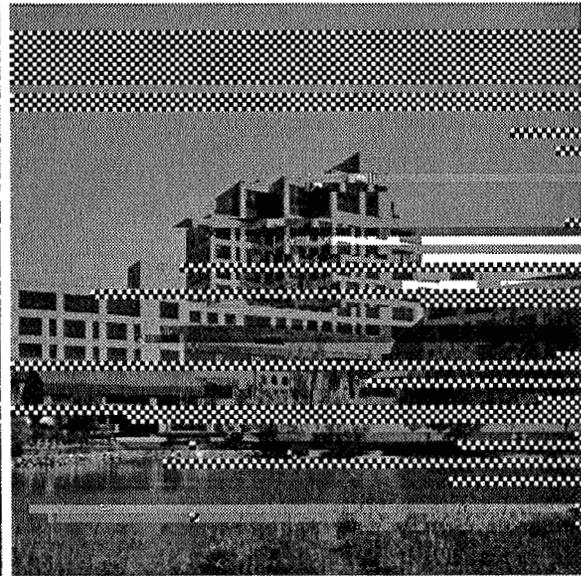
FIGURE 7. Wavelet-UTCQ, 1.0 bpp, high-resolution high frequency subbands removed.



(a)



(b)



(c)

Figure 8: NITFS JPEG compression with a restart interval of 64 macroblocks, CR= 6.12 (1.307BPP). Reconstructed from (a) uncorrupted compressed bitstream. (b) compressed bitstream corrupted by average burst error rate of  $9.77 \times 10^{-4}$  with 2 bits average burst error length. (c) compressed bitstream corrupted by average burst error rate of  $1.95 \times 10^{-3}$  with 2 bits average burst error length.

## 5. Noise Performance

**NITFS-JPEG** Simulations were carried out by injecting burst errors and single bit errors at several rates into the compressed bitstreams corresponding to various images compressed at several bitrates. In general, the noise performance of NITFS-JPEG (with the restart interval mechanism turned on) compressed imagery was satisfactory even at high channel error rates. Figure 8 presents a reconstructed image from an uncorrupted bitstream and bitstreams corrupted by burst errors at two different burst error rates:  $9.77 \times 10^{-4}$  ( $\approx 10^{-3}$ ) and  $1.95 \times 10^{-3}$  ( $\approx 2 \times 10^{-3}$ ) with an average burst error length equal to 2 bits. It contains an image compressed with NITFS-JPEG. This simulation clearly illustrates the usefulness of restart intervals in NITFS-JPEG compressed imagery, particularly for applications where the reconstructed imagery is used by humans. It yields easily recognizable reconstructed imagery without unduly increasing the bitrate, thus offering the best compromise between quality and bitrate for transmitting compressed imagery over noisy channels.

**Wavelet-UTCQ** Figures 9 and 10 illustrate two realizations of bit errors introduced into a 52-band, 0.25bpp encoding. A BER of  $10^{-4}$  was used for both images. These two figures represent the best and worst decoded images. Include with these images are coefficient maps indicating the subbands that incurred bit errors and were removed. Clearly the bit errors in the LFS of Figure 10 account for its poor quality. If bit errors are confined to high-frequency subbands as in Figure 9, the reconstruction is quite good. In general, bit errors of the order  $10^{-3}$  produced no useful output and bit errors in the  $10^{-5}$  range were usually not noticeable.



FIGURE 9(a). Wavelet-UTCQ, 0.25 bpp, BER  $10^{-4}$ .

1	2	5	6	17	18						
3	4	7	8			29	30				
9	A	D	E	19							
B	C	F	G								
21	22	25	26			31					
23	24	27	28								
33	34	37	38					43		44	
35	36	39	40								
45								49		50	
47								51		52	

FIGURE 9(b). Wavelet-UTCQ, 0.25 bpp, BER  $10^{-4}$ , removed subbands are shaded.



FIGURE 10(a). Wavelet-UTCQ, 0.25 bpp, BER  $10^{-4}$ .

2	5	6		18			
4	7				29	30	
9	A	D	E	19			
B	C	F	G				41
	22	25	26		31		42
	24	27					
33		34		37	38		
						43	44
35				39	40		
45				46		49	50
47				48		51	52

FIGURE 10(b). Wavelet-UTCQ, 0.25 bpp, BER  $10^{-4}$ , removed subbands are shaded.

## 6. Summary and Conclusions

The proposed technique to insert re-initialization points into the wavelet-UTCQ coder proved to be simple and provided some measure of error recovery. Certainly the performance is enhanced over that shown in Figure 5. The overhead incurred by this technique is approximately 2 - 3 bytes per coefficient sequence. Compared with the NITFS-JPEG error recovery using restart intervals, using wavelet coefficient subbands for resynchronization is somewhat lacking. The number of coefficients in a given subband is too large and the entropy-coded data too long for BERs above  $10^{-4}$ . Using subbands for re-initialization makes the error recovery a function of rate as well. As the encoding rate is increased, the length of entropy-coded data in a subband increases thus increasing the probability of a bit error occurring in a subband for a fixed BER.

We have seen that which coefficients incur bit errors is important. The spreading error effect for low-resolution coefficients, coupled with their high energies, makes their protection paramount. As seen in Figure 7, the high-frequency subbands could be sent without any FEC at all. Non-hierarchical JPEG modes do not show this property. JPEG localizes its errors in an image and they are usually visible regardless of where they occur. Figures 7 and 10 have shown that if we can constrain the bit errors in the wavelet-TCQ bitstream to the higher-resolution coefficients, we can achieve image reconstruction visually superior to JPEG.

Future work will concentrate on producing a "packetized" wavelet-UTCQ system. Implementation of an entropy coder (Huffman or arithmetic) capable of producing fixed length data packets will allow for much improved error recovery. Each data packet will now correspond to a variable number of input coefficients. For the decoder to interpret the data stream in the event of a packet loss, the number of wavelet coefficients in each packet, location of the first coefficient in the resolution hierarchy, and trellis state associated with the first coefficient must be included for each packet. Packets occurring at the end of a subband may contain considerable empty space and tradeoffs between packet length, FEC overhead, and unused packet bytes must be investigated. This system would work well with the TACO2 protocol. If the packet sizes are synchronized between the communication and compression software, a lost communication packet will correspond to a lost data packet.

## 7. References

- [1] J. H. Kasner and M. W. Marcellin, "Universal Trellis Coded Quantization", in *Int. Symposium on Information Theory*, Whistler, BC, September, 1995
- [2] J. H. Kasner, "Adaptive Wavelet Compression of Imagery", Ph.D Dissertation, The University of Arizona, 1995.
- [3] I. H. Whitten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *ACM Trans. on Mathematical Software*, pp. 520--540, June 1987.
- [4] D. M. Abrahamson, "An adaptive dependency source model for data compression," *Comm. of the ACM*, pp. 77-83, Jan. 1989.
- [5] R. L. Joshi and T. R. Fischer and J. H. Kasner and M. W. Marcellin, "Arithmetic and Trellis Coded Quantization", *Proc. IEEE Int. Symp. on Info. Theory*, Trondheim, Norway, June, 1994.
- [6] MIL-STD-2500A, "The National Imagery Transmission Format Standard (NITFS)", June, 1993.
- [7] MIL-STD-2045-44500, "Tactical Communications Protocol 2 (TACO2) for the National Imagery Transmission Format Standard, June, 1993".
- [8] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss Markov sources" , *IEEE Trans. on Commun.*, COM-38, pp. 82-93, January, 1990.
- [9] T. R. Fischer and M. Wang, "Entropy- constrained trellis- coded quantization", *IEEE Trans. on Info. Theory*, IT-38, pp. 415-426, March, 1992
- [10] M. W. Marcellin, "On entropy- constrained trellis coded quantization", *IEEE Trans. on Commun.*, pp. 14-16, January, 1994
- [11] ISO IS 10918-1/CCITT Recommendation T.81, "Digital Compression and Coding of Continuous-Tone Still Images, Part I: Requirements and Guidelines", 1993.
- [12] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," Tech. Rep. 11217- 900529- 08 TM, AT & T Bell Laboratories, Murray Hill, NJ, May 1990.

53-82  
 013174  
 10P.  
 068897

# Low Bitrate Compression With Downsampling and JPEG

Kristo Miettinen, miettine@felix.kodak.com  
 Eastman Kodak Company, 1447 St. Paul Street, Rochester, NY 14653-7208

**Abstract.** Image compression with the JPEG DCT algorithm incorporated into the National Imagery Transmission Format Standard (NITFS) does not naturally extend to low bitrates due to the fairly small blocksize of JPEG (8 by 8) and the inefficiencies of interblock coding. A natural extension of JPEG to lower bitrates consists of resampling an image to a smaller size and compressing the smaller image with JPEG DCT at a higher bitrate.

The JPEG standard [1] defines hierarchical JPEG compression with resampling by a factor of two in each spatial dimension, repeated as many times as necessary. The NITFS compression standard [2] does not include the hierarchical extensions to JPEG DCT compression, but does include data structures in its ancillary information fields for transmitting image reduction and zoom factors. These can be used to implement low bitrate compression with the JPEG DCT compression engine in the manner of hierarchical JPEG, but with arbitrary resampling ratios.

## 1.0 Introduction

Filters for resampling prior to compression and interpolation after reconstruction can be designed to satisfy various criteria. These include canceling all aliased copies of DC in the periodic spectrum of the discrete-position (spatially quantized) representation of an image prior to resampling or interpolation, having passband ripples in the resampling and interpolation filters with opposite phase so that the combined effect of the resampling and interpolation is maximally flat passband response, and minimizing total aliased energy throughout the compression and reconstruction chain.

This presentation, paper, and demonstration present analysis of the joint filter optimization problem applying windowed sinc filters of equal length and jointly optimized windows, applying the time-domain duals of classical Nyquist filters, and also applying truncated sinc filters of jointly optimized unequal length convolved with spatially shifted sample-and-hold filters. Results and comparisons to hierarchical JPEG filters are presented both in terms of aliased and baseband power spectra in the processing chain, and in terms of compressed and reconstructed imagery.

## 2.0 Image Models

Modeling imagery as a 2-D Gauss-Markov stochastic process allows complete description of the stochastic process through definition of the functional form of its power spectral density. For most general purposes the models of the underlying continuous image processes should be circularly symmetric in the spatial and spectral domains, allowing the substitution of Hankel transforms in  $r$  and  $\rho$  for Fourier transforms in  $\{x, y\}$  and  $\{\xi, \psi\}$ . However for practical purposes the downsampling and upsampling filters will be separable, so that the circularly symmetric image models must yield to separable models for designing row and column filters.

For the purposes of this study the same basic modeling concept will be applied for both separable and circularly symmetric modeling: an image is taken to consist of regions of uniform intensity randomly tessellated by curved edges, with no correlation in intensity between pixels belonging to regions separated by an edge. With this model, the autocorrelation function of the image becomes a pure function of separation, proportional to the probability of no edge falling on a spatial segment of any given length. For randomly occurring edges this probability decays exponentially, so that the autocorrelation models and their associated power spectral densities become:

**Symmetric Model in  $r, \rho$**

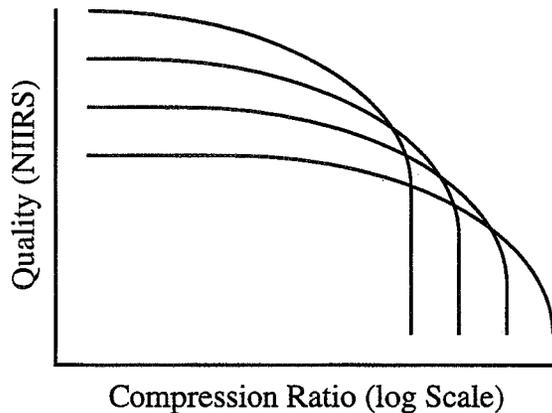
$$R(r) \sim e^{-\alpha|r|} \Leftrightarrow F^2(\rho) \sim \frac{P}{(\alpha + \rho^2)^{3/2}}$$

**Separable Model in  $x, \xi$**

$$R(x) \sim e^{-\alpha|x|} \Leftrightarrow F^2(\xi) \sim \frac{P}{\alpha + \xi^2}$$

### 3.0 Motivation: Efficient Bitrate Hypothesis

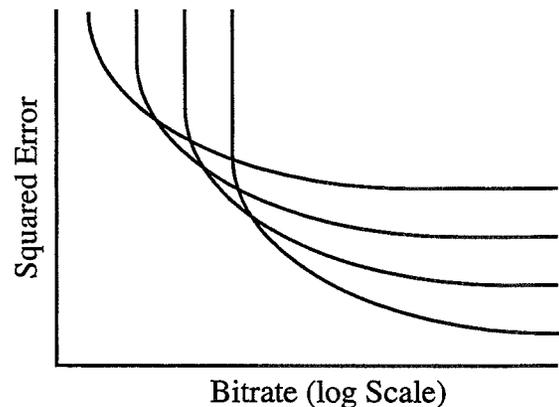
The motivation for this effort can be stated as the *efficient bitrate hypothesis*: that for JPEG (or any other fixed resolution compression algorithm) there is an efficient bitrate at which the algorithm comes closest to reaching the limiting rate-distortion function for the chosen distortion measure, and that the optimal way to operate JPEG at lower bitrates is to reduce the input image size so that the algorithm can operate on the reduced image at its efficient rate.



The efficient bitrate hypothesis is illustrated with sheaves of possible quality/compression ratio and rate/distortion curves for JPEG with varying degrees of subsampling. Note that for logarithmic quality scales such as the National Imagery Interpretability Rating Scale (NIIRS) and logarithmic mapping of compression ratios the shape of the curves is consistently similar but that scaling the input image shifts the curves down and to the right. Shifting the curves down accounts for the irreversible losses introduced in the subsampling, and shifting the curves right accounts for the fact that identical JPEG operating bitrates reflect lower effective bitrates given higher subsampling ratios.

Note also the existence of asymptotes both in rate and in distortion for each curve: the overhead bitrate of JPEG imposes a minimum bitrate, while the irreversible losses of subsampling will limit the algorithm's ability to code to arbitrarily high fidelity.

The envelopes of the sheaves consist of similar points on separate curves, suggesting that for optimal rate/distortion or quality/compression ratio performance the compound algorithm should be operating at the same relative point (the “knee” or “elbow”) on the JPEG operating curve, with the sampling ratio adjusted to choose an absolute operating point on the envelope of the subsampled JPEG operating curves.



#### 3.1 Searching for the Efficient Bitrate: The Diving-Bell Theorem

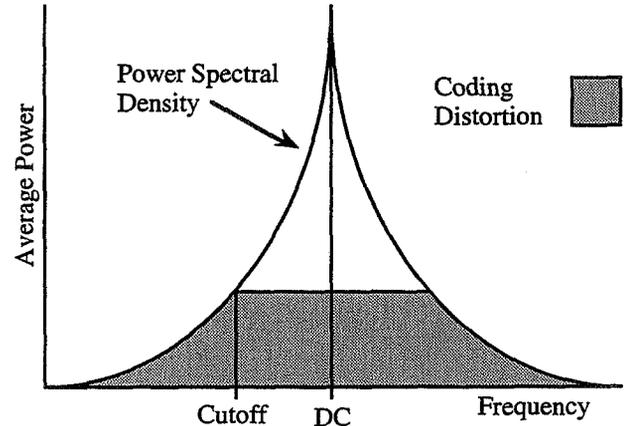
The source coding theorem for a Gaussian random process source (unnamed by Gallager [3], which I call the Diving Bell Theorem) states that for efficient coding of a Gaussian source to a squared error criterion, the coding distortion will be constant across all frequencies, except that frequencies for which the source power is below the distortion threshold shall not be coded.

The diving-bell interpretation of this theorem is that if you plot the power spectral density of a source, the efficient distortion levels to which the source is ideally coded correspond to the various possible levels of the air/water interface under a diving bell shaped as the power spectral density, filled with air from below. This is the source-coding dual of the more famous noisy channel coding Water Filling Theorem of Shannon [4].

The efficient bitrate of an ideal coder can be derived given a shape for the diving bell (i.e. a model for the functional form of the power spectral density of imagery), the source coding theorem, and Shannon's logarithmic relation for limiting bitrate as a function of signal power to distortion ratio for the Gaussian source [4]. The derivation proceeds by dividing the limiting entropy of the circularly symmetric Gaussian power spectral density coded to the distortion

threshold  $D$  by the size of a critically sampled square image encompassing that portion of the power spectrum being coded (for which the power density is above the distortion threshold).

In an arbitrarily densely sampled unitary Fourier transform relationship a real image of given square size transforms into a complex matrix of the same size, with conjugate symmetry about the midpoint. Thus the size of a critically sampled image is the same as the size of a bounding square about the coded frequencies, while the coded rate is bounded by the entropy integral (allowing distortion  $D$ ) taken in a half-plane (only half of the complex frequencies are independent). In the entropy integral the entropies of the independent real and imaginary signal components must be added together.



**Diving Bell Interpretation of the Source Coding Theorem**

The same result can be established for the periodic even extension of an image, for which case the frequency domain is the cosine transform domain.

The cosine transform interpretation eliminates both complex values and conjugate symmetry in the transform domain, but assumes periodic even extension. Using the cosine transform interpretation is consistent with use of the DCT for decorrelation. The analytical result is identical in either case.

### 3.2 Efficient Bitrate of an Ideal Coder for Inverse Cubic Power Spectral Density

The efficient bitrate of an ideal compressor can be derived for an image model by taking the ratio of the entropy integral under the diving bell for the coded frequencies to the area of the bounding square of the coded frequencies.

Taking the inverse cubic power spectral density model, the radius of the diving bell can be derived by equating the power density and the distortion:

$$\frac{P}{\rho^3} < D \Rightarrow 0 \leq \rho < \sqrt[3]{\frac{P}{D}}$$

The critical bitrate expression can then be developed as a function of the signal power coefficient to distortion ratio  $P/D$  (which ratio is the same both for the complex power and distortion, and for their real or imaginary components). The coefficient  $P$  is simply a scaling factor to account for the unspecified units of power and unmade assumptions about total image power level; as  $P$  cancels out of the final analytical solution these assumptions need not be made.

$$Ideal \frac{bits}{pixel} = \frac{\int_{\rho=0}^{\sqrt[3]{PD}} \int_{\phi=0}^{\pi} 2 \cdot \rho \cdot \log_4 \frac{P}{D \cdot \rho^3} \cdot d\phi \cdot d\rho}{4 \left(\frac{P}{D}\right)^{2/3}} = \frac{3\pi}{8 \cdot \ln 4} \cong 0.8498 \text{ b/p}$$

Note that by taking the power spectral density model in the limit as  $\alpha$  goes to zero this ideal critical bitrate becomes independent of the overall source power level  $P$  or the threshold distortion  $D$ : Varying the source power or the allowed distortion varies both the coded rate and the image size together, so that the overall bitrate is constant.

For values of  $\alpha$  other than zero the peak of the power spectral density about zero is rounded and finite rather than divergent. In this case the limiting bitrate above ( $\sim 0.85$  b/p) is an upper bound on the general ideal bitrate given the symmetrical inverse cubic power spectral density model, which in turn would be an approximation of efficient bitrates for practical compressors.

For noisy or sharpened imagery a symmetric inverse square power spectral density model may be more appropriate. For this model the similarly derived efficient bitrate is  $\pi/4 \ln 4$ , or 0.5665 b/p. Finally, the preliminary results given in the demonstration are for JPEG operating at 0.65 b/p.

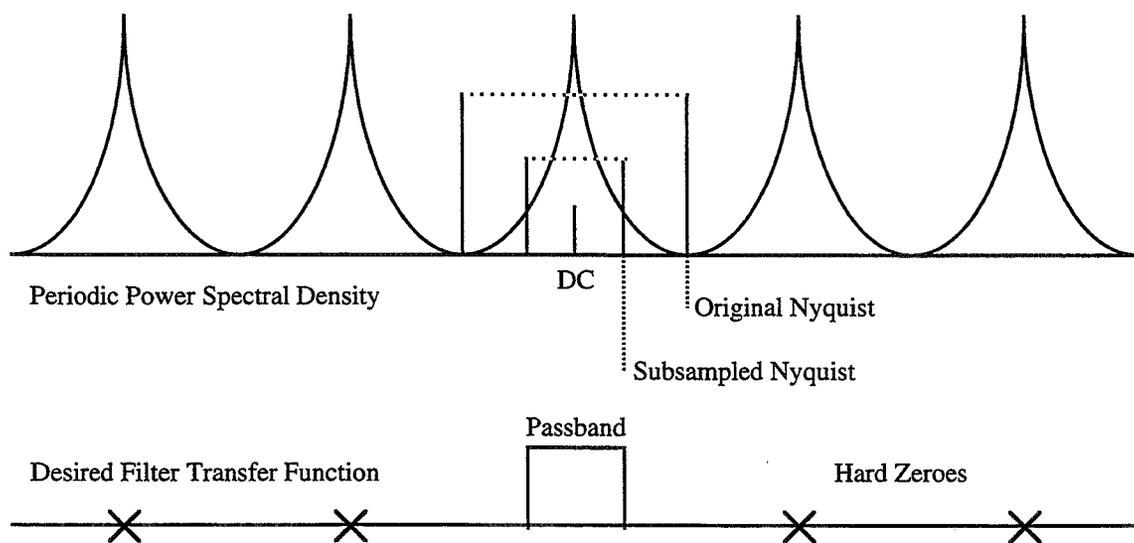
## 4.0 Desirable Qualities For Filters

Ideal linear low-pass filters have a  $\text{sinc}(2\pi f_c x)$  impulse response, with  $f_c$  defining the cutoff frequency in cycles per sampling interval. Practical linear low-pass filters are often designed by applying a suitable window function to the  $\text{sinc}(2\pi f_c x)$  impulse response to create a finite-length impulse response whose transfer function is approximately ideal. Nearly any actual finite-length lowpass filter can be designed in this manner, since the ratio of any desired filter's impulse response to the  $\text{sinc}(2\pi f_c x)$  function can be interpreted as a window function, except that the windowed sinc design constrains the locations of zero crossings in the windowed impulse response.

The natural choice for a lowpass filter cutoff frequency is the Nyquist frequency corresponding to the lower of the two sampling rates, the target rate for aggregation (downsampling) and the source rate for interpolation (upsampling). The Nyquist frequency is half the sampling rate, and it is the highest frequency which can be represented in the sampled function without aliasing. Filtering at the target Nyquist frequency before downsampling limits aliasing of frequencies present in the source signal which cannot be represented in the aggregated image. Filtering at the source Nyquist frequency before upsampling (the same spatial frequency relative to features in the image as the target Nyquist frequency for downsampling) prevents interpretation of aliased source baseband frequencies as true baseband frequencies in the broader bandwidth interpolated image.

### 4.1 Properties for Practical Filters and Sources

Considering the design of actual anti-aliasing filters and specific functional forms for power spectral densities, added desirable characteristics can be specified. Whereas practical filters have frequency responses with ripples in their passband, it would be desirable to design the interpolating and aggregating filters to have passband ripples  $180^\circ$  out of phase, so that the compound frequency response due to both filters is flat. Furthermore, whereas natural imagery tends to have most of its signal power concentrated around DC (constant intensity specifically and low spatial frequencies generally), having a filter transfer function with zeroes at all frequencies above baseband which correspond to an integer number of wavelengths in the sampling interval would reduce aliasing, since these frequencies are where the periodic spectrum of the sampled signal has the most power above the Nyquist frequency.



**Inference of Downsampling Filter Characteristics From Signal PSD**

The desired downsampling filter characteristic is illustrated above. The desired upsampling filter characteristic has the same passband as the downsampling filter, but the period of the power spectrum of the downsampled image will be shorter. The hard zeroes would therefore be at frequencies above baseband corresponding to an integer number of wavelengths in the subsampled sampling interval (which is broader, so that the hard zeroes are more closely spaced).

In summary, the desired characteristics of the downsampling and upsampling filters are:

- Evenly spaced zeroes in frequency response at nonzero even multiples of the original image Nyquist frequency for the downsampling filter, and at the subsampled image Nyquist frequency for the upsampling filter.
- Sharp transition in frequency response at the Nyquist frequency of the subsampled image for both filters.
- Flat joint passband response for the concatenation of both filters, especially about DC (where most of the signal power is). The filters will have rippled passband response, but the ripples should be designed to cancel out.

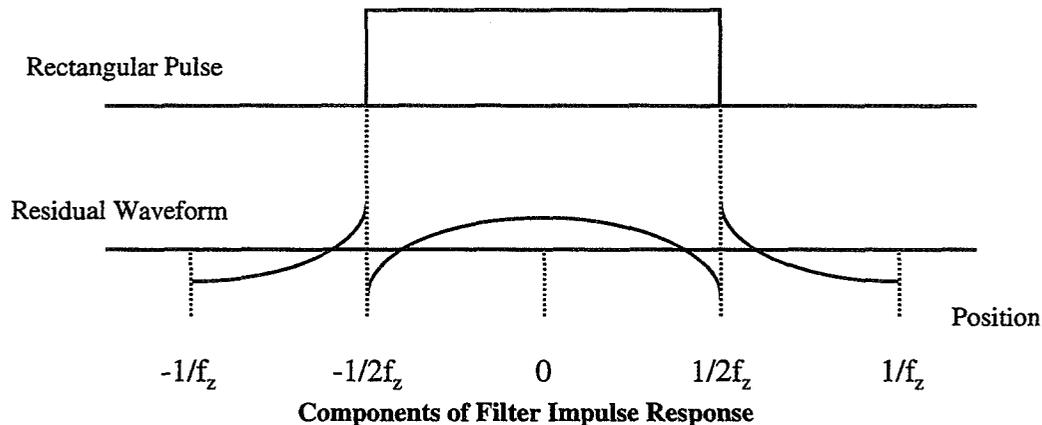
## 5.0 Application of Nyquist Filter Design

The problem of designing a finite-length filter with evenly spaced zeroes in the frequency domain is the Fourier transform dual of the traditional Nyquist waveform design problem from telegraphy. Nyquist [5] was concerned to design waveforms with finite bandwidth and evenly spaced zeroes in time for transmission of information over telegraph wires without interpulse interference. His results can be applied to the design of filters of finite length with evenly spaced zeroes in frequency response, and then extended to add other characteristics (such as a cutoff frequency) to the filters.

Define the frequencies of the first hard zeroes about DC as  $\pm f_z$ . Then following Nyquist, the filter impulse response is broken into two additive components: a rectangular pulse of length  $1/f_z$ , and a residual waveform of length  $2/f_z$ .

The components have the following required properties:

1. Both waveforms must have even symmetry about their midpoints. This causes zero response to centered sine waves of all frequencies.
2. The left and right halves of the residual waveform must have odd symmetry about their midpoints (the quartile points). This forces zero response to cosines of frequencies  $\pm n f_z$ .



Nyquist demonstrated that the family of raised cosines can be decomposed in this manner, and proposed the use of waveforms with raised cosine spectra as telegraph pulses. If the properties of raised cosines were otherwise acceptable then raised cosine impulse responses could be used for downsampling and upsampling filters. However further desired properties have been specified above, and an optimal approximation (in the sense of sharpest cutoff) to the ideal lowpass filter can be explicitly derived for the constraints of the structure above.

The steps of the optimal approximation are:

1. Begin with the ideal  $\text{sinc}(2\pi f_c x)$  waveform, truncated to the interval  $-1/f_z < x < 1/f_z$ .

2. Scale the waveform for DC response of  $1/f_z$ . This is the same DC response as the ideal infinite length sinc waveform for upsampling, and it is the ideal DC response scaled by the sampling ratio for downsampling.
3. Extract and retain the odd component of the positive and negative halves of the truncated scaled waveform.
4. Add a rectangular pulse of length  $2/f_z$  and height  $1/2$  to the extracted waveform (note that this is not the same length as the rectangular pulse in the diagram above).

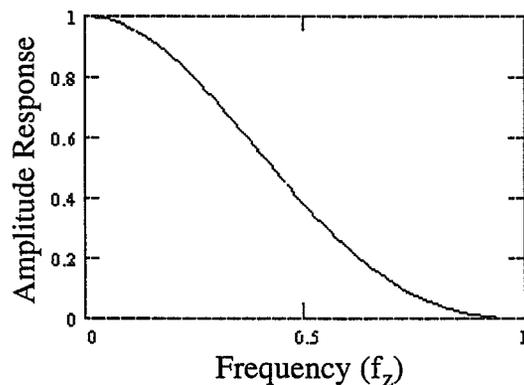
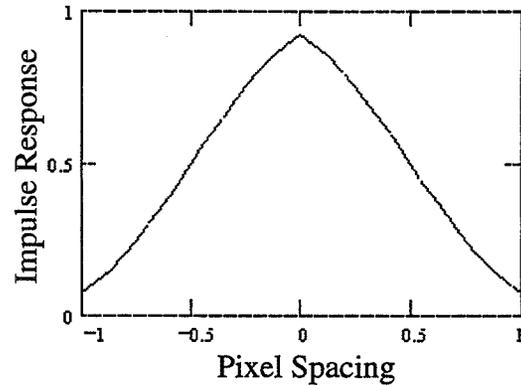
This one step shorter than, but operationally equivalent to subtracting out a shorter rectangular pulse scaled so as to zero out the DC response of the residual, extracting the odd component of the halves of the residual, adding the rectangular pulse back in, and scaling the final waveform for proper DC response.

### 5.1 Illustration of Nyquist Design for Upsampling Filters

The upsampling filters form a suitable example for illustration, since for them the relationship of the filter cutoff frequency  $f_c$  to the first hard zero frequency  $f_z$  is fixed regardless of the downsampling ratio (for all upsampling filters  $f_c = f_z/2$ ). Applying the steps of the optimal approximation above for the upsampling case produces a fairly triangular impulse response with zeroes in the correct positions, but very poor cutoff at the Nyquist frequency of the downsampled image,  $f_c = f_z/2$ . This is due to the lack of freedom to place additional zeroes between  $f_c$  and  $f_z$ , so that the filter must have its steep cutoff near its first zero, which is at twice the frequency of the desired cutoff.

For sharper cutoff additional zeroes are needed above  $f_c$ , but not all the way out at  $f_z$ . This will steepen the transfer function at  $f_c$  as it falls toward the first zero. An alternative for producing steeper cutoff at  $f_c$  in the Nyquist design paradigm would be to increase the length of the filter, thereby increasing the density of zeroes. Increasing the length of the filter by any integer factor less than  $f_z/f_c$  gives discrete choices for the downsampling filter (for which  $f_c < f_z/2$ ).

However in the upsampling case any integer scaling of the length of the filter introduces a zero in the passband, which is obviously not desirable. Designing a matched pair of filters with the Nyquist solution is difficult in general (with the subsampling ratio unknown *a priori*). Constraining the Nyquist solution to raised cosine impulse responses doubles the density of hard zeroes above  $f_z$  (but not at  $f_z/2$ ), so that a raised cosine filter of length  $3/f_z$  would have zeroes at multiples of  $f_z/3$  at and above  $2f_z/3$ , but for raised cosines there are too few free parameters to optimize a frequency cutoff and still match the two filters to one another for flat passband response.



## 6.0 Convolved Truncated Filter Design

A different method of solving Nyquist's problem consists of designing a filter as the convolution of two components, one of which has zeroes in the appropriate positions in its transfer function while the other is arbitrary. In this manner the discrete sizes of Nyquist filters are avoided, as the arbitrary component can have any length.

The natural choice for the zero-fixing component is Nyquist's rectangular impulse response of length  $1/f_z$ , which is nothing more than a nearest-sample filter (a spatially shifted sample-and-hold filter) for a discrete-position sequence. Due to its flat response this choice has the desirable property that computing the convolution in many spatially shifted positions simultaneously by numerical integration is accelerated by reusing integrals over subintervals in all convolutions spanning the subinterval.

Once the hard zeroes have been fixed by convolution with the nearest-sample filter, the remaining desirable characteristics of the arbitrary component are that it provide maximally sharp cutoff and that the passband ripples of the upsampling and downsampling filters be so arranged as to provide a flat composite transfer function, especially at DC (where the signal power is concentrated).

The classical result for maximally sharp cutoff of a lowpass filter is the truncated ideal sinc( $2\pi f_c x$ ) filter [6]. Given truncated sinc filters and a predetermined cutoff frequency (the Nyquist frequency of the downsampled image) the only remaining freedom to match the filters to one another is to truncate them to unequal length, with the lengths so chosen as to produce the desired flat composite transfer function.

## 6.1 Joint Filter Design Strategy

Given filters of different length, the natural location for the shorter filter is at the receiver (for upsampling), for at least the following reasons:

- The receiver is more likely to be in a tactical environment where computing power is scarce, and therefore shorter filters are desirable.
- The upsampler translates imagery to a scale whose Nyquist frequency is above the anti-aliasing filter cutoff frequency, so that further anti-alias filtering can always be performed after upsampling if needed.
- Decompression and upsampling may be done many times for each time an image is compressed, again favoring keeping the upsampling less complex than the downsampling.

For the shorter (upsampling) filter, sharpness of the cutoff of the frequency transfer function is of greater importance than for the longer (downsampling) filter, since the longer filter will always have a sharper cutoff given that both filters are truncated sincs. Thus, the chosen filter optimization strategy will consist of optimizing the upsampling filter independently of the downsampling filter, and then designing the longer downsampling filter to match the optimized upsampling filter. Fortuitously the downsampling filter fitted to the optimal upsampling filter is nearly optimal in its own right by the efficiency criterion used to design the upsampling filter.

## 6.2 Design of the Upsampling Filter

The criterion chosen for optimizing the upsampling filter is the length-normalized derivative of the DC-normalized transfer function evaluated at the cutoff frequency.

Length normalization refers to constructing a figure of merit by dividing the transfer function derivative at the cutoff frequency by the length of the filter, since the derivative at the cutoff is inversely proportional to filter length except for an oscillating term. This normalization identifies filter lengths which are most efficient in terms of cutoff sharpness attained relative to their computational cost.

For a symmetric truncated sinc waveform of length  $\lambda$  the frequency response is

$$\int_{-\frac{\lambda}{2}}^{\frac{\lambda}{2}} \cos(\omega \cdot x) \cdot \frac{\sin(\omega_c \cdot x)}{\omega_c \cdot x} \cdot dx = \int_{\frac{\lambda}{2} \frac{\omega - \omega_c}{\omega_c}}^{\frac{\lambda}{2} \frac{\omega + \omega_c}{\omega_c}} \frac{\sin(\omega_c \cdot x)}{\omega_c \cdot x} \cdot dx$$

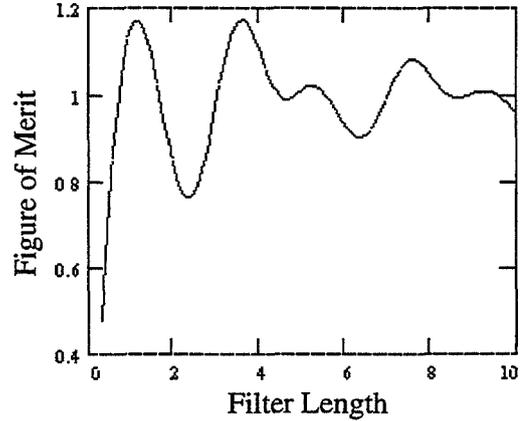
Differentiating and evaluating at the cutoff frequency yields

$$\left. \frac{\sin\left(\frac{\lambda}{2} \cdot (\omega_c + \omega)\right)}{\omega_c \cdot (\omega_c + \omega)} - \frac{\sin\left(\frac{\lambda}{2} \cdot (\omega_c - \omega)\right)}{\omega_c \cdot (\omega_c - \omega)} \right|_{\omega=\omega_c} = \frac{\sin(\lambda \cdot \omega_c)}{2 \cdot \omega_c^2} - \frac{\lambda}{2 \cdot \omega_c}$$

Note that for  $\lambda\omega_c$  equal to even multiples of  $\pi$ , the derivative of the cutoff sharpness at  $\omega_c$  with respect to  $\lambda$  is zero. The trend in the dependency of the cutoff sharpness above on  $\lambda$  is that the derivative grows as  $-\lambda/2\omega_c$ , so that dividing the cutoff sharpness by that expression (which is directly proportional to computational complexity) and also by the DC response (to normalize) produces a figure of merit for comparing filter lengths in terms of their computational efficiency in achieving cutoff sharpness for DC normalized filters.

Expressing the length of the filter in terms of the sampling interval of the downsampled image (also equal to the number of lobes of the sinc included in the truncation, with the main lobe counted as two since the sine in the numerator has two lobes in the main lobe of the sinc), where  $n = \lambda\omega_c/\pi$ , produces the adjacent FOM as a function of  $n$ .

$$FOM = \frac{-\pi}{2 \cdot \lambda} \cdot \frac{\sin(\lambda) - \lambda}{\int_0^{\frac{\lambda}{2}} \frac{\sin(x)}{x} \cdot dx}$$



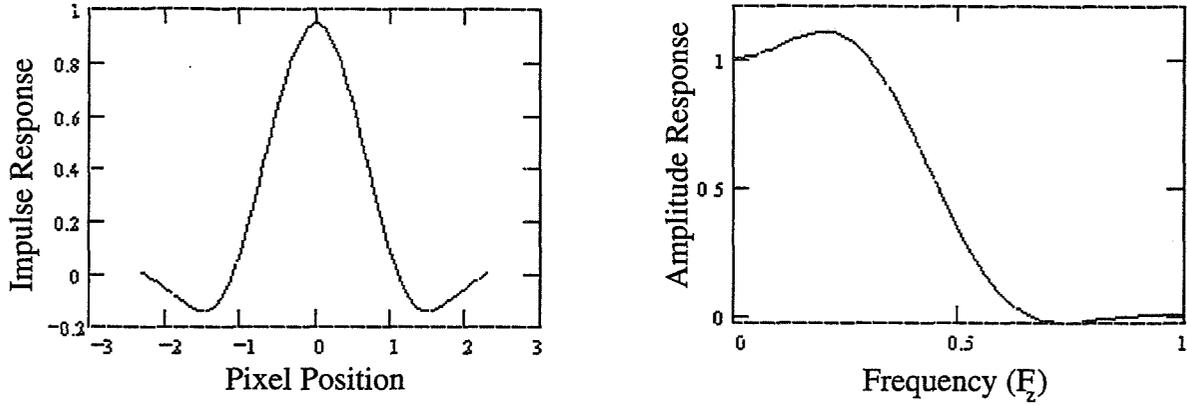
The first two local optima above appear attractive as effective lengths for upsampling filters, but on inspection the first local optimum corresponds to a filter truncated inside the first zero crossings of the sinc ( $n < 2$ ), at a point which is efficient more because it is computationally sparse than because of the sharpness of the filter cutoff. The second local optimum is shorter than a cubic spline interpolator (which is fairly close in shape to an  $n = 4$  truncated sinc) but spans the first zero crossings of the sinc, so it is a plausible upsampling filter length.

The precise locations of the local optima can be found numerically by first differentiating the FOM with respect to  $\lambda$  and then solving for the roots of the numerator. These are the roots of the expression

$$(\lambda \cdot \cos(\lambda) - \sin(\lambda)) \cdot \int_0^{\frac{\lambda}{2}} \frac{\sin(x)}{x} \cdot dx + (\lambda - \sin(\lambda)) \cdot \sin\left(\frac{\lambda}{2}\right)$$

These roots are found at  $n = \{1.171, 3.612, 5.281, 7.615, \dots\}$ . The root chosen for the length of the upsampling filter is  $n = 3.612$ , and it would be desirable to have the downsampling filter length near  $n = 5.281$ .

For fine-tuning the optimization above the effects of the convolution with the nearest-sample filter (a rect of length  $1/f_z$ ) could be included in the figure of merit. For the downsampling filter the ratio of  $f_z$  to  $f_c$  depends on the arbitrary downsampling ratio, but for the upsampling filter being derived above the ratio is known *a priori* ( $f_z = 2f_c$ ). The effects of including the nearest-sample filter in the figure of merit would be a constant scaling of the cutoff slope in the numerator (which has no effect on relative choices of filter length) and a constant increment to computational complexity in the denominator (which favors slightly longer filters).



Upsampling Filter Characteristics

### 6.3 Design of the Downsampling Filter

The downsampling filter was chosen above (section 6.1) to be the longer of the two filters, and therefore its cutoff sharpness is less of a concern for optimization, although it would be desirable to have the downsampling filter length be around  $n = 5.281$ . The selection criterion for the downsampling filter is that its passband transfer function at DC should complement the transfer function of the upsampling filter. Specifically, since both filters will be DC normalized and have zero derivative at DC (due to symmetry), the desired property is that the second derivative of the downsampling filter transfer function at DC cancel out the second derivative of the upsampling filter transfer function. Taking the second derivative of the transfer function (the function and its first derivative are in section 6.2 above) and evaluating at DC,

$$\begin{aligned} & \frac{\frac{\lambda}{2} \cdot \cos\left[\frac{\lambda}{2}(\omega + \omega_c)\right] \sin\left[\frac{\lambda}{2}(\omega + \omega_c)\right]}{\omega_c \cdot (\omega + \omega_c)} - \frac{\frac{\lambda}{2} \cdot \cos\left[\frac{\lambda}{2}(\omega_c - \omega)\right] \sin\left[\frac{\lambda}{2}(\omega_c - \omega)\right]}{\omega_c \cdot (\omega_c - \omega)^2} \\ & + \frac{\frac{\lambda}{2} \cdot \cos\left[\frac{\lambda}{2}(\omega_c - \omega)\right] \sin\left[\frac{\lambda}{2}(\omega_c - \omega)\right]}{\omega_c \cdot (\omega_c - \omega)} - \frac{\frac{\lambda}{2} \cdot \cos\left[\frac{\lambda}{2}(\omega + \omega_c)\right] \sin\left[\frac{\lambda}{2}(\omega + \omega_c)\right]}{\omega_c \cdot (\omega + \omega_c)^2} \\ & = \frac{\lambda}{(\omega_c)^2} \cdot \cos\left(\frac{\lambda}{2} \cdot \omega_c\right) - \frac{2}{(\omega_c)^3} \cdot \sin\left(\frac{\lambda}{2} \cdot \omega_c\right), \quad \omega = 0. \end{aligned}$$

For large values of  $\lambda$  the cosine term dominates, and complementary filter lengths correspond to differences in filter length for which the argument of the cosine is  $\pi$ , which correspond to length differences of  $\Delta\lambda = 2\pi/\omega_c$ , or  $\Delta n = 2$ .

For the general case the filter length must be found numerically as the root of the expression for the second derivative of the compound transfer function at DC, expressed below in terms of  $L_u$  and  $L_d$ , where  $L_u$  and  $L_d$  are the downsampling interval normalized filter lengths  $\lambda_u\omega_c$  and  $\lambda_d\omega_c$  respectively (the subscripts identify up- or downsampling). From the optimization of the upsampling filter in section 6.2 above,  $L_u = 3.61212 \cdot \pi = 11.34781$ .

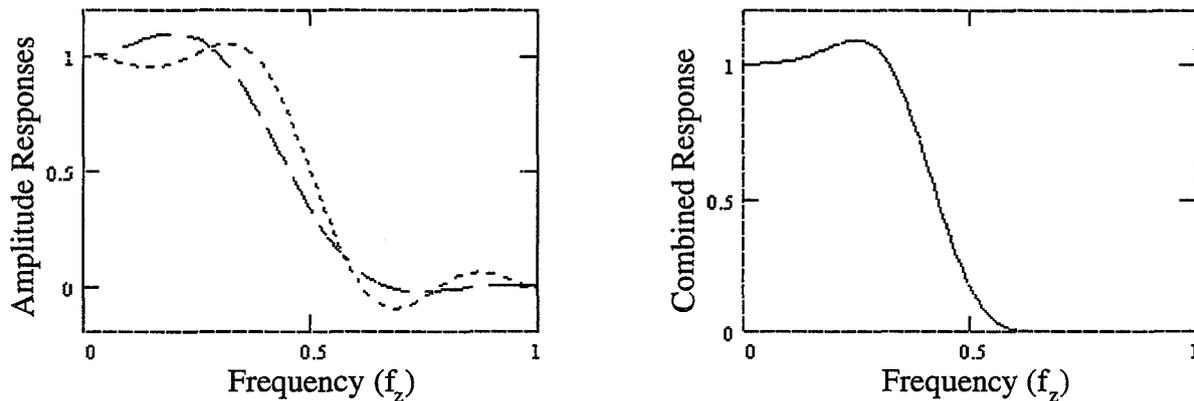
The extra constant in the middle of the expression is the contribution of the nearest-sample filter to the second derivative of the frequency response of the upsampling filter. A similar term for the downsampling filter would be a function of the downsampling ratio, and therefore not known *a priori*.

$$\frac{L_u \cdot \cos\left(\frac{1}{2} \cdot L_u\right) - 2 \cdot \sin\left(\frac{1}{2} \cdot L_u\right)}{\int_0^{\frac{L_u}{2}} \frac{\sin(x)}{x} dx} - \frac{\pi^2}{4} + \frac{L_d \cdot \cos\left(\frac{1}{2} \cdot L_d\right) - 2 \cdot \sin\left(\frac{1}{2} \cdot L_d\right)}{\int_0^{\frac{L_d}{2}} \frac{\sin(x)}{x} dx}$$

Solving numerically for the root of the expression above,  $L_d = 16.41896 \Rightarrow n = 5.2263$ . The precise shape of the downsampling filter and its transfer function still depend on the nearest-sample filter for the original image sampling

interval, but of the four filter components in the system (two truncated sincs and two nearest-sample filters) that will always be the smallest and negligible for design purposes.

Taking the downsampling truncated sinc of length  $n = 5.2263$  and plotting its frequency response together with that of the upsampling filter yields the following results.



**Matching of Filter Passband Responses**

The joint passband response has a broader flat region around DC than either of the individual filters has, with a single positive lobe leading into the cutoff. Since the designs have been optimized for the sharpest possible cutoff the positive lobe is natural and desirable; it pushes the passband response below  $f_c$  higher so that the fall at  $f_c$  is steeper.

## 7.0 Conclusion

The JPEG image compression algorithm implemented in the National Imagery Transmission Format Standard can be efficiently extended to arbitrarily low bitrates by compressing imagery in a compound system where original imagery is reduced in size prior to JPEG compression, and then resized back to the original scale after JPEG reconstruction.

This paper, and the associated presentation and demonstration, describe the filters and procedures used for low-bitrate JPEG implementation in the NITFS low bitrate evaluation. This implementation was competitive in performance and complexity with state-of-the-art wavelet algorithms entered into the evaluation [7].

The author gratefully acknowledges the support of the Central Imagery Office.

## 8.0 References

- [1] ISO Draft International Standard 10918-1, Annex J.
- [2] MIL-STD-188-198A, JPEG Image Compression for the National Imagery Transmission Format Standard.
- [3] Gallager, R. G., *Information Theory and Reliable Communication*, 475-490, Wiley, New York, 1968.
- [4] Shannon, C., "A Mathematical Theory of Communication", *Bell System Tech. J.*, vol. 27, pp. 623-656, 1948.
- [5] Nyquist, H., "Certain Topics in Telegraph Transmission Theory", *Trans. AIEE*, vol. 47, pp. 617-644, 1928.
- [6] Oppenheim, A., and Schaffer, R., *Digital Signal Processing*, 239-250, Prentice Hall, Englewood, 1975.
- [7] McCabe, J., Miettinen, K., and Reitz, J., "The Evaluation of Low Bitrate and Multi-Component Image Compression Algorithms for Inclusion in the National Imagery Transmission Format Standard (NITFS)", Combined Industry, Space and Earth Science Data Compression Workshop, Snowbird, 1996.

34-32  
013175

**THE EVALUATION OF LOW BIT RATE AND MULTI-COMPONENT IMAGE  
COMPRESSION ALGORITHMS FOR INCLUSION IN THE NATIONAL IMAGERY  
TRANSMISSION FORMAT STANDARD (NITFS)**

268898  
10P.

Jill McCabe  
Kristo Miettinen  
Joseph Reitz

Eastman Kodak Company  
1447 St. Paul St.  
Rochester, NY 14653-7208

email: mccabe@felix.kodak.com, miettine@felix.kodak.com, reitz@kodak.com

**Abstract**

The current National Imagery Transmission Format Standard (NITFS) specifies use of the JPEG-DCT algorithm for compression of image data [1]. The JPEG-DCT algorithm has two shortcomings which may warrant the inclusion of several new algorithms to fulfill these needs. Baseline JPEG-DCT compression does not provide adequate quality to users in the NITFS community at bit rates below 0.5 bits per pixel (bpp). In addition, multi-component images can be compressed with higher fidelity by removing the correlation present between the individual image bands.

A call for candidate low bit rate and multi-component compression algorithms was announced in January 1995 and continued through March 1995. These candidate algorithms have undergone two phases of performance testing. This paper summarizes the evaluation procedure used, as well as draws some conclusions on actual algorithm performance.

**Introduction**

The National Imagery Transmission Format (NITF) is the designated format for transmission of digital imagery and image-related products by the Department of Defense and other departments and agencies of the United States Government [3]. The purpose of the NITF Standard (NITFS) is to provide a common standard for transmission of a file composed of an image with subimages, symbols, labels, text, and other information that relates to the image [3].

The primary NITFS compression algorithm, JPEG-DCT, provides adequate compression performance for most users in the NITFS community. There is, however, a fairly large community of NITFS users which must receive imagery products over low bandwidth communications channels using computers of relatively low computational power. Compressing an image to rates below 0.5 bpp allows these users to receive the necessary image products in a timely manner. The JPEG-DCT algorithm does not provide adequate quality to these users at the low bit rates they typically require. An evaluation is currently being conducted to identify

compression algorithms which will satisfy the low bit rate compression requirements of this user community.

The use of an algorithm to reduce the correlations present between components in a multi-component image prior to application of existing spatial compression algorithms can yield dramatic increases in compression performance. Multi-component images are images which are composed of multiple correlated bands, as is the case with multi-spectral images, stereo pairs, sequential spatial slices and MRI images. The current NITFS compression algorithm includes no provision to handle images which contain more than three correlated bands of image data. An evaluation is currently underway to identify compression algorithms which will satisfy the needs of the NITFS community for compression of image data which contains multiple correlated bands of image data.

### **Evaluation Methodology**

A call for candidate low bit rate and multi-component compression algorithms was made through a Commerce Business Daily (CBD) announcement in January 1995. Since the number of participants responding to the CBD announcement was not limited, an initial evaluation was used to select a subset of algorithms to undergo rigorous performance testing in the second phase of the evaluation. An image quality goal was set for the low bit rate algorithms to not exceed a quality loss of more than 0.5 on the National Imagery Interpretability Rating Scale (NIIRS) at 0.5 bpp. For the multi-component evaluation, the image quality goal was set for algorithms to not exceed a quality loss of 0.5 on the Multi-Spectral Image Interpretability Rating Scale (MSIIRS) at a bit rate of 0.25 bits per pixel per band (bpppb). The NIIRS and MSIIRS scales were developed to quantify the utility of imagery based on a user's ability to identify objects of a given size in the scene. Points on each scale define typical objects which can be identified at that quality level. Compression artifacts which distort objects in the scene will have a negative impact on the NIIRS or MSIIRS rating for a given scene.

In phase one of the evaluation there were 17 respondents who expressed interest in the low bit rate evaluation, and 7 respondents who expressed interest in the multi-component evaluation. Of these respondents, 15 low bit rate and two multi-component respondents actually participated in the phase one evaluation.

The 15 low bit rate participants were asked to compress two visible, two radar, and two medical images to various bit rates including 0.5, 0.25, 0.125 and 0.0625 bpp. A visual engineering analysis was conducted to estimate the visual image quality loss induced by each algorithm. The results from this analysis were combined with results from numeric image quality (i.e. Root Mean Square Error), and estimations of computational complexity, bit rate control, vulnerability to bit errors, portability, and compliance to existing standards. From the analysis of these results, four algorithms were chosen to undergo more extensive testing in the second phase of the evaluation. One of the four algorithms chosen in the low bit rate evaluation was the interim NITFS low bit rate algorithm [4].

The two multi-component participants were asked to compress four multi-component images to rates of 1.0, 0.5, 0.25 and 0.125 bpppb. Since there were only two participants in the phase one evaluation, there was no need to select a subset of algorithms to continue in the second phase multi-component evaluation. A brief phase one evaluation and analysis was conducted to

verify the bit rates used in the evaluation as well as verify that the quality of the phase one algorithms was within reasonable bounds. Since there was room in the phase two evaluation for additional algorithms, two baseline algorithms were tested.

The second phase of the evaluation included more rigorous testing than was performed in the phase one evaluation. For the low bit rate evaluation, the participants were asked to compress a total of 74 images which included visible, infrared, radar, fingerprint, map and medical images to bit rates of 0.5, 0.25, 0.125 and 0.0625 bpp. For the multi-component evaluation, participants were asked to compress a total of 27 multi-component images to bit rates of 2.0, 1.0, 0.5, 0.25, and 0.125 bpppb. The 2.0 bpppb rate was added to the multi-component evaluation to provide performance feedback in the near-lossless range of algorithm operation.

### **Phase Two Algorithm Description**

Due to proprietary issues, algorithms are referred to by a specific coded name for each algorithm. For the low bit rate evaluation, algorithms LBR1, LBR2, and LBR3 all implemented wavelet compression variations. The remaining algorithm, the interim NITFS low bit rate algorithm, preprocessed the image prior to JPEG-DCT compression using variable subsampling.

For the multi-component evaluation, algorithms MC1 and MC2 implemented different variations of a Karhunen-Loeve Transform (KLT) / wavelet algorithm, using the KLT for component decorrelation followed by a spatial wavelet compressor. The first of the two multi-component baseline algorithms applied the current single band NITFS compression option (JPEG-DCT) to each band in a multi-component image, applying the interim NITFS low bit rate algorithm to achieve bit rates below 0.5 bpppb. The second baseline algorithm used the standard KLT statistical decorrelation algorithm to decorrelate the image components before applying the JPEG-DCT / interim NITFS compression algorithm to each decorrelated band. The second baseline option reflected the performance gains which could be attained by including an image component decorrelation algorithm as a preprocessor to the current NITFS compression options.

### **Phase Two Visual Image Quality Evaluation**

The phase two visual image quality evaluation was performed at the National Exploitation Laboratory (NEL) by imagery analysts. To limit evaluator fatigue, a subset of the full imagery set was visually evaluated. For the low bit rate evaluation, 7 visible, 7 infrared and 10 radar images were evaluated at all four bit rates (0.5, 0.25, 0.125 and 0.0625 bpp). For the multi-component evaluation, 23 multi-spectral images were evaluated at four bit rates (1.0, 0.5, 0.25 and 0.125 bpppb).

Both evaluations were performed on high quality color displays. Each analyst evaluated the images over a two-day period (one day for the multi-component evaluation, the other for the low bit rate evaluation). The low bit rate evaluation was broken down into three segments, by sensor type (visible, radar, and infrared). The multi-component evaluation included only one segment (multi-spectral). The ordering of the segments in the evaluation was randomized for each participant. The presentation order within a given segment was randomized by scene, compression algorithm, and compression rate.

The evaluator was allowed to flicker between 2X magnifications of the original and compressed / expanded images. The evaluators had the capability to scroll within the scene to examine the effect of compression artifacts. For each image, the evaluator rated the compression loss in NIIRS for the low bit rate evaluation or in MSIIRS for the multi-component evaluation.

Each evaluator was asked to answer a subjective question concerning the effect of artifacts on the utility of the compressed / expanded image using a predefined subjective scale. Tie points were defined on the scale to maintain consistency between evaluators. The rating and definition of each tie point can be found below in Table 1.

<b>Tie Point</b>	<b>Associated Rating</b>	<b>Definition</b>
No Change	0	No noticeable difference between the two images.
Slightly Hinders	2	Slight loss in utility between the two images; Adequate to perform exploitation.
Moderately Hinders	4	Notable loss in utility between the two images; Adequate to perform exploitation, but may seriously affect the accuracy of exploitation.
Significantly Hinders	6	Significant utility loss between the two images; Unusable for exploitation, but usable for briefings, image overlays, and orientation.
Excessively Hinders	8	Severe utility loss between the two images; Unusable for any purpose.

**Table 1 - Subjective Quality Scale**

### **Phase Two Numeric Image Quality Evaluation**

All images in both compression evaluations were tested for numeric accuracy. For the low bit rate compression evaluation, 17 metrics were used. For the multi-component evaluation the low bit rate compression metrics were applied to each band. To study interband relationships, an additional 12 metrics were used across all bands. Several of these metrics, which were specific to multi-spectral images, were applied only to the multi-spectral images in the multi-component evaluation test set. In order to study the numeric image quality evaluation metrics, a correlation test was performed on the data produced from each of the metrics. In the case of multiple correlated metrics, detailed results were produced for just one of the correlated metrics. To simplify the analysis process, RMSE was chosen to be the primary numerical evaluation metric. In the cases where an evaluation metric captured new information regarding numerical accuracy, the results from the metric were analyzed and reported. The combined explanatory power of all the metrics provided a better estimation of the numerical accuracy produced by each compression algorithm.

## Other Evaluation Factors

The other evaluation factors in the NITFS image compression evaluations included, computational complexity, bit rate control, vulnerability to bit errors, portability, and compliance to existing standards. The performance of an algorithm in each of the factor areas was determined from data obtained from a questionnaire which was sent to each participant.

The goal of the computational complexity evaluation was to test the compression algorithm, not the specific implementation. To determine the complexity of a given algorithm, we estimated the relative complexity of each component of the compression algorithm (i.e. the complexity difference in Huffman encoding versus arithmetic encoding). These complexity numbers were verified by two methods. The first was to combine the number of each type of operation with the time required to execute that operation on a given platform to obtain a predicted execution time. The second verification was provided by timing the executable decompressor on a specific platform. Each verification could not be relied on for actual performance numbers since they were determining the complexity of the implementation. The verifications did provide adequate feedback as to whether a component complexity estimation was approximately accurate.

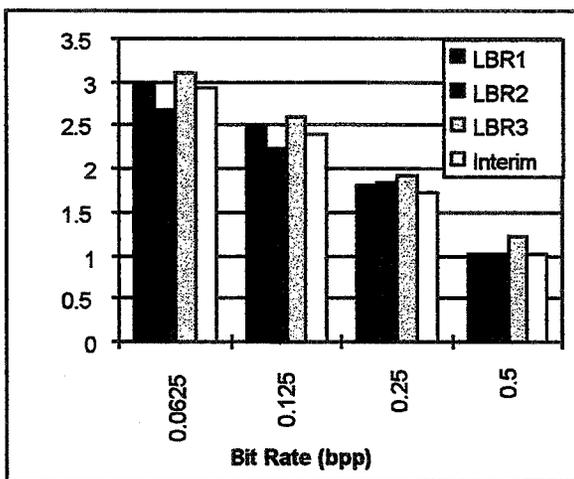
The robustness of each algorithm in the presence of bit errors was determined by evaluating the effect of a channel error on the compressed bit stream. Using the supplied error recovery method from the questionnaire, we were able to estimate the effect of a bit error on the decompressed image. The estimation of algorithm robustness was verified by injecting errors into the compressed bit stream, decompressing the bit stream using the supplied implementation, and observing the propagation of the error through the expanded image. Estimations of relative algorithm robustness using an advanced error recovery method were also determined. This information provided an estimation of algorithm robustness in the event of a bit error in its current and optimized states.

The rankings for each algorithm for the remaining evaluation factors were determined directly from answers provided by the algorithm supplier. Each of the factors was combined with the image quality results to form a matrix for evaluation of algorithm performance. The importance of each of the evaluation factors to users in the NITFS community will be used to judge which algorithms, if any, merit inclusion in the NITFS.

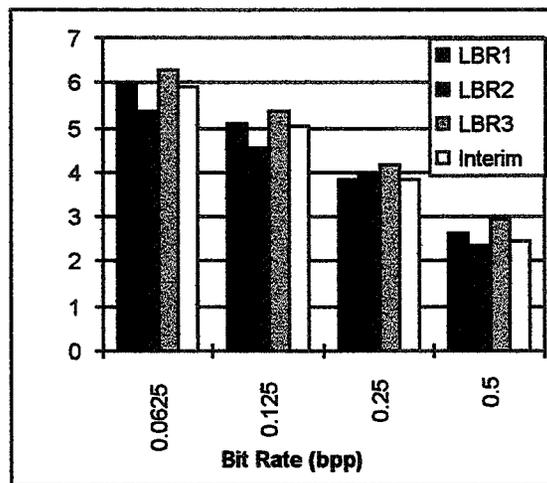
## Results

For the low bit rate visual evaluation, algorithm LBR2 performed slightly better across most image types at the lower evaluation bit rates (less than 0.25 bpp). Algorithm LBR1 and the interim NITFS algorithm performed similarly across most image types and bit rates. Algorithm LBR3 performed slightly worse than the other algorithms across most image types and bit rates. In the numerical evaluation, looking only at RMSE results, LBR1, LBR3 and the interim NITFS low bit rate algorithm performed similarly across most bit rates and image types. Algorithm LBR2 performed slightly better than the other three algorithms at bit rates lower than 0.25 bpp and worse than the other algorithms at the higher bit rates, a trend not shown in the visual evaluation results. Further investigation into this trend showed that at the lower bit rates the LBR2 algorithm introduced a textured artifact which was visually pleasing, but numerically

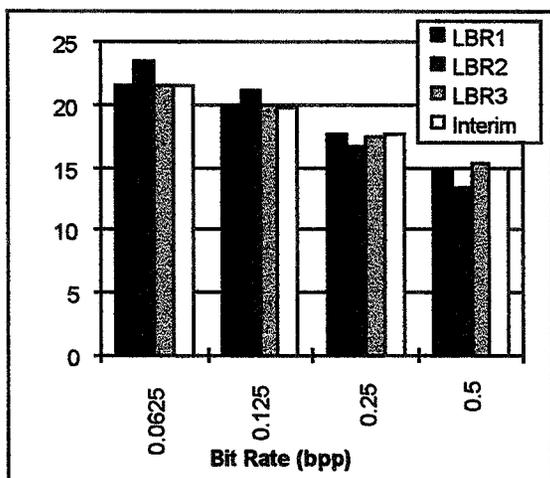
inaccurate. Results for the visible class of imagery can be found in Figures 1, 2 and 3. The magnitude of the ratings varied across the other image types, but the relationships between algorithms remained fairly constant.



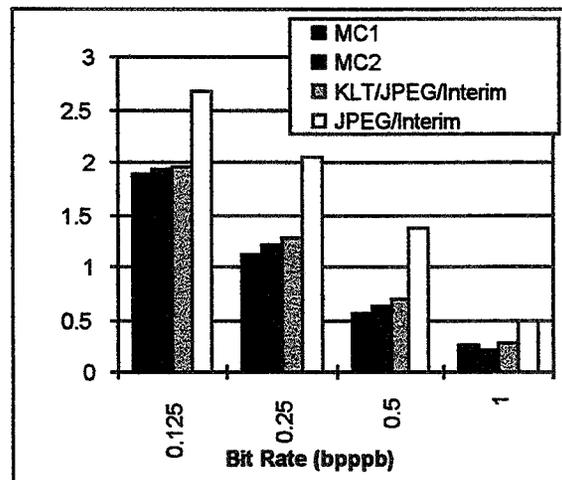
**Figure 1 - Low Bit Rate Evaluation  
Delta NIIRS Ratings for Visible Imagery**



**Figure 2 - Low Bit Rate Evaluation  
Subjective Quality for Visible Imagery**



**Figure 3 - Low Bit Rate Evaluation  
Average RMSE for Visible Imagery**



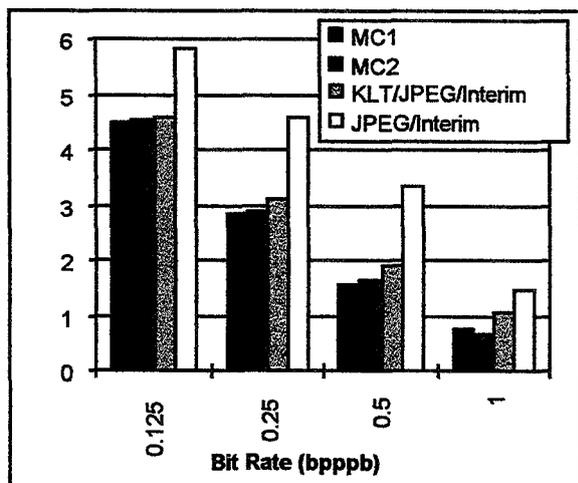
**Figure 4 - Multi-Component Evaluation  
Delta MSIIRS Ratings for 3, 4, 5 Band Images**

Algorithm LBR2 was generally more complex than the other three algorithms across all bit rates. This algorithm used an advanced quantization strategy which is more complex than standard quantization methods. Algorithm LBR3 and the interim NITFS low bit rate algorithm showed a strong complexity dependence on compressed bit rate. This is due to the use of arithmetic encoding by LBR3 and the variable subsampling implemented by the interim NITFS

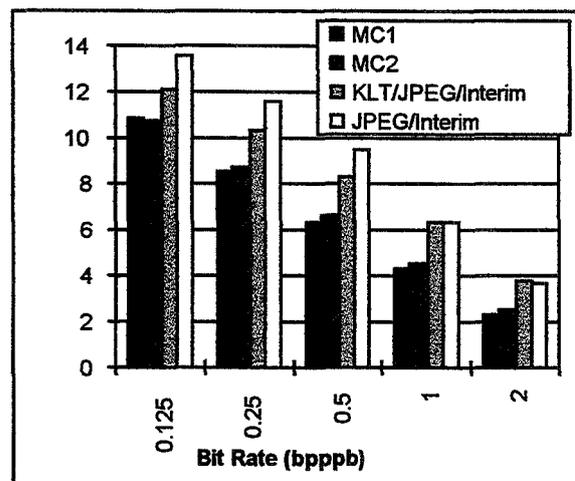
low bit rate algorithm. At the higher evaluation bit rates, these algorithms were more computationally complex than the LBR1 algorithm.

In their current implementations, algorithms LBR1 and LBR2 generally did not recover in the presence of a bit error. Algorithm LBR3 recovered from a bit error, but stopped decompression at the location of the bit error. This generally produced an image whose quality was too low for practical use. The interim NITFS low bit rate algorithm recovered from a bit error, but generally all data between the location of the bit error and the next restart marker was lost. This produced unacceptable results in horizontal stripes of the image affected by the error, but the remainder of the image was left uncorrupted.

For the multi-component evaluation, algorithms MC1, MC2 and the KLT baseline algorithm all showed a dependence between image quality and number of bands. The results for the multi-component evaluation have been separated into two groupings by band number (images which contained 3, 4 or 5 bands, and images which contained 8, 12 or 16 bands). For the visual evaluation, algorithms MC1 and MC2 performed similarly across the various bit rates and band groupings. The KLT baseline algorithm performed slightly worse than the MC1 and MC2 algorithms at the lower bit rates, especially for images which contained the most bands. The baseline algorithm which performed no component decorrelation, performed substantially worse than the three other algorithms, especially for images with the most bands. At the higher bit rates, the KLT baseline algorithm performed similarly to the MC1 and MC2 algorithms. Figures 4,5,7 and 8 graphically depict the multi-component visual evaluation results.



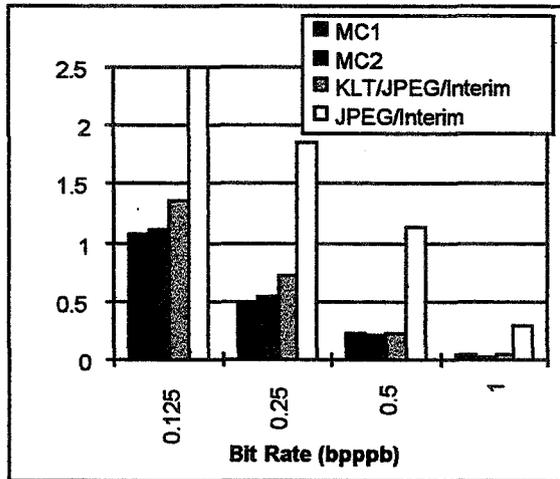
**Figure 5 - Multi-Component Evaluation Subjective Quality for 3, 4, 5 Band Images**



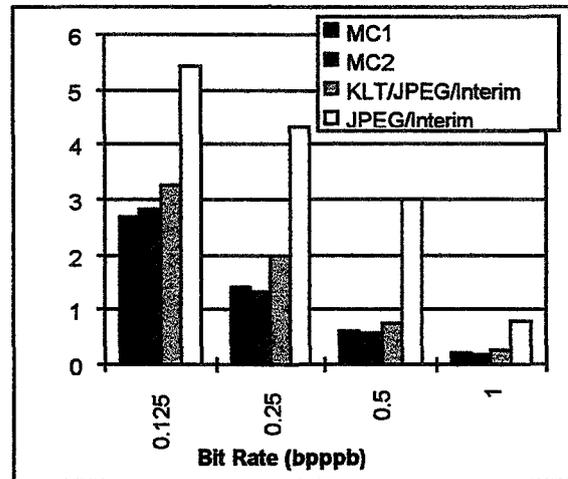
**Figure 6 - Multi-Component Evaluation Average RMSE for 3, 4, 5 Band Images**

In the numerical evaluation, algorithms MC1 and MC2 performed similarly across most bit rates and band groupings. The KLT and the non-decorrelating baseline algorithms performed worse than the MC1 and MC2 algorithms across all bit rates and band groupings. Additionally, these results showed that at the highest bit rate, the KLT baseline algorithm actually performed worse than the non-decorrelating algorithm. Further investigation showed that the process of integerizing the KL transformed bands for compatibility with JPEG contributed significantly to this inefficiency. Figures 6 and 9 show the results, separated by band grouping, for the analysis of RMSE.

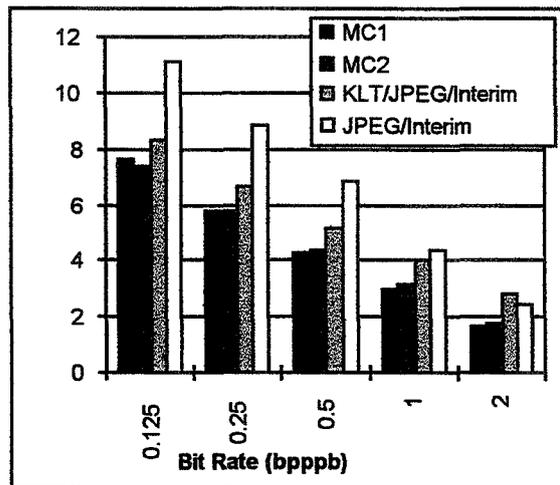
Looking at supervised classification accuracy, we see that the KLT baseline algorithm performed better than any of the other three algorithms at the lower bit rates. Visual examination of the results from the MC1 and MC2 algorithms showed a tendency to induce spectral distortions in localized areas of the image. These distortions would negatively impact a machine-based exploitation as was shown with the supervised classification accuracy metric. At the higher bit rates, algorithms MC1 and MC2 show advantages in supervised classification accuracy. Figure 10 shows the supervised classification accuracy results for a single 16 band image.



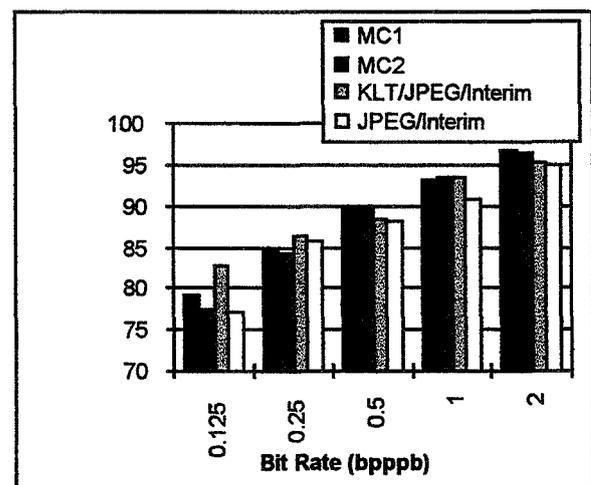
**Figure 7 - Multi-Component Evaluation Delta MSIIRS for 8, 12, 16 Band Images**



**Figure 8 - Multi-Component Evaluation Subjective Quality for 8, 12, 16 Band Images**



**Figure 9 - Multi-Component Evaluation Average RMSE for 8, 12, 16 Band Imagery**



**Figure 10 Multi-component Evaluation Classification Accuracy for a 16 Band Image**

Computationally, the non-decorrelating baseline algorithm performed the best. The remaining three algorithms were all slowed by the time required to perform the KL transform.

This was especially true for the band grouping which contained the most bands, since the KLT complexity is related to the square of the number of bands. At the higher bit rates, the KLT-JPEG baseline algorithm performed next best, followed by the MC2 algorithm. A bit rate dependence forced the MC1 algorithm to be more computationally complex than the MC2 and the baseline KLT-JPEG algorithms at the higher bit rates. This was largely due to the use of arithmetic coding by the MC1 algorithm. At the lower bit rates, the three algorithms performed similarly.

Algorithms MC1 and MC2 did not recover from a bit error in their current implementation. The two baseline algorithms recovered with the same mechanism as the interim NITFS low bit rate algorithm. For the non-decorrelating baseline algorithm, the error was localized to the band in which it occurred. For the KLT baseline algorithm, a single bit error affected all image bands, although the magnitude of the effect varied according to the original energy content of the area in the transformed band which the bit error affected. In algorithms MC1 and MC2, the same holds true, although the effect of a bit error would generally be less catastrophic if an advanced recovery method is used. This is due to the ability of the wavelet algorithm to maintain some degree of accuracy in the region affected by the error depending on the location of the error in the data stream.

## Conclusions

With experimental error included, all low bit rate algorithms only met the image quality criterion for the infrared imagery type. The interim NITFS low bit rate algorithm produced results which were similar in image quality to the submitted algorithms. Implementation of shorter length filters in the interim NITFS low bit rate algorithm should produce similar image quality at a lower computational complexity.

In the multi-component evaluation, algorithms MC1 and MC2 met the image quality criterion only for the band grouping which contains the most bands. The MC1 and MC2 algorithms induced spectral distortions at the lower bit rates which affected the quality of machine based exploitation tasks. The performance of these algorithms needs to be improved if machine based exploitation is important at these bit rates.

The use of the KL transform produced dramatic increases in quality for images which contained a large number of correlated bands. However, the use of the KL transform can increase complexity beyond what many users are capable of handling. A system which has a large amount of bandwidth available for transmission and uses computers of low computational power may find it faster to send each band at a higher quality level with no component decorrelation.

All algorithms in both evaluations which were not JPEG based did not produce acceptable results in the presence of bit errors in their current implementation. More work must be done on these algorithms to improve their performance for implementation in noisy channels. The implementation of advanced recovery methods with these algorithms should increase their performance in noisy channels beyond the current NITFS JPEG-DCT implementation with some image quality loss.

The final results from this evaluation will be used to decide which, if any, of the evaluated algorithms merit inclusion in the NITFS. This evaluation effort is still ongoing, but should be completed in the Spring of 1996.

### **Acknowledgments**

This work was performed under contract to the Central Imagery Office. The authors of this paper are grateful for their support.

### **References**

1. MIL-STD-188-198A, "Joint Photographic Experts Group (JPEG) Image Compression for the National Imagery Transmission Format".
2. MIL-STD-2500A, "National Imagery Transmission Format (NITF) for the National Imagery Transmission Format Standard (NITFS)".
3. MIL-HDBK-1300A, "National Imagery Transmission Format Standard Handbook".
4. K. Miettinen, "Low Bitrate Compression with Downsampling and JPEG" preprint of paper to be included in the proceedings of the 1996 Combined Industry / NASA Workshops.
5. K. Riehl Jr., C. Erdman, "An assessment of the Effects of Data Compression on Multispectral Imagery", 1994 Industry Workshop.
6. A. Eskicioglu and P. Fisher, "A Survey of Quality Measures for Gray Scale Image Compression", proceedings of 1993 Space and Earth Science Data Compression Workshop.

85-82  
013176  
568899

# Live Video Communication over Computer Networks Using MPEG\*

10P.

**Ciro Aloisio Noronha Jr.**

**Feiling Jia**

Optivision, Inc.  
3450 Hillview Ave.  
Palo Alto, CA 94304  
email: noronha,jia@optivision.com

## Abstract

The advent of MPEG-1 has resulted in a standard method for compressing audio and video, producing VCR-like video quality at data rates in the range of 1.5 Mb/s. A natural extension of these techniques is to use MPEG compression to transmit live audio and video over computer networks. With MPEG, it becomes possible to deliver good quality video and audio at reasonable data rates, using existing networking infrastructures (Ethernet/Token Ring in the local area, T1 and T3 pipes in the wide area). Moreover, the new MPEG-2 encoding scheme allows delivery of broadcast-quality video at rates between 4 and 15 Mb/s, using network technologies such as ATM or Fast Ethernet.

This paper addresses the issues related to transmitting live MPEG streams over traditional packet switched computer networks. After a brief introduction, we discuss the three main issues in transporting live MPEG, namely, encoder/decoder synchronization, error control, and latency (which is important in interactive applications such as videoconferencing). Finally, we discuss issues related to transporting MPEG streams in current network infrastructures.

## 1. Introduction

The advent of MPEG-1 has resulted in a standard method for compressing audio and video, producing VCR-like video quality at data rates in the range of 1.5 Mb/s. Moreover, with MPEG-2, it is possible to produce broadcast-quality video in the 4 to 15 Mb/s range. Once real-time MPEG encoders and decoders are available, a natural extension of these techniques is to use MPEG compression to transmit live audio and video over traditional computer networks, as depicted in Figure 1. The important issues related to transporting live MPEG streams over computer networks are:

---

\* This work was funded in part by NSF under SBIR grant number DMI-9460280.

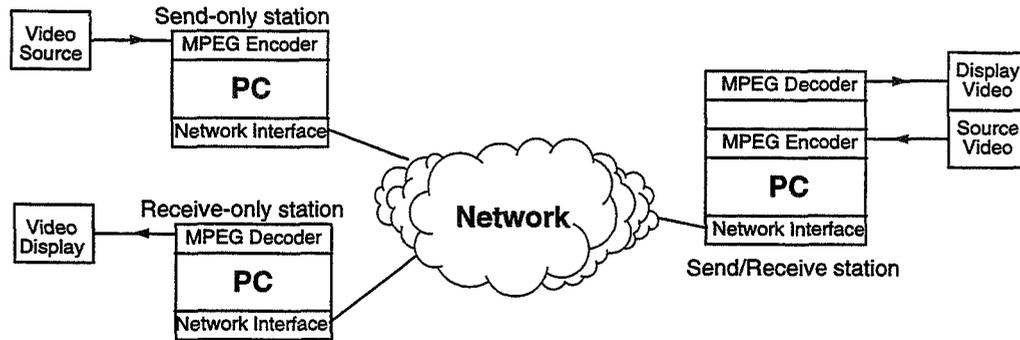


Figure 1: Transmission of Live MPEG Streams over a Computer Network

- *Encoder/Decoder clock synchronization:* The encoder produces data according to its own internal clock at a certain nominal rate. The decoder (which may be located on another city or even on another country) consumes data at the same nominal rate. However, if their clocks are not synchronized, the decoder will eventually overflow or underflow due to the mismatch of rates.
- *Error Recovery and Control:* Typical computer networks operate in a “best-effort” mode: they will try their best to deliver the data, but no guarantees are given - a packet can be silently dropped (for example, in case of congestion). Excessive loss, however, will degrade the video quality to unacceptable levels. This is specially true of compressed video, since loss of data in a frame might cause a glitch which will persist for several frames.
- *Latency:* Latency is defined as the time between the moment a frame of video enters the encoder, and the moment the same frame is displayed at the decoder. Some iterative applications, such as video-conferencing, require latency to be low, typically under 250 ms. Other applications, such as remote teaching, where the interaction between the participants is more structured, can tolerate higher latencies. Latency is an issue in MPEG, due to the processing and buffering required, and due to the fact that, to compress some frames, the algorithm requires knowledge of future frames. Therefore, the encoder must wait for this future frame before it can encode the current frame. A similar comment can be made for the decoder.

## 2. Encoder/Decoder Synchronization

As pointed out in Section 1, the clock used in the decoder to control the playback of the video data must be synchronized with the encoder clock used to generate it. If it is not, the decoder will either overflow or underflow. Consider, for example, a case where the decoder clock is running at a rate slightly lower than that of the encoder clock. MPEG data will slowly “accumulate” at the decoder, until its buffer overflows. Similarly, an underflow situation will occur when the decoder clock is running slightly faster than the encoder clock. Therefore, the two clocks must be kept synchronized. There are two options to achieve this synchronization:

- *Use a network-wide clock for synchronization:* This approach is, in principle, feasible when using public communication networks, since they are all synchronized to the same master clock. However, in LAN environments, this kind of network clock is not available.

- *Use time-stamps the time stamps in the stream and a PLL to recover the clock [1]:* In this approach, a PLL (Phase-Locked Loop) is used to keep the local decoder clock synchronized to the encoder clock. This synchronization is achieved by comparing the time-stamps in the received MPEG stream with time-stamps from a local register, which is driven by the decoder clock. From this comparison, a phase error can be derived to adjust the decoder clock, as shown in the block diagram of Figure 2. For this technique to work well, the delay in the network must be constant. Small variations (delay jitter) can be accommodated by the loop filter. Delay jitter, however, may be a problem in congested networks.

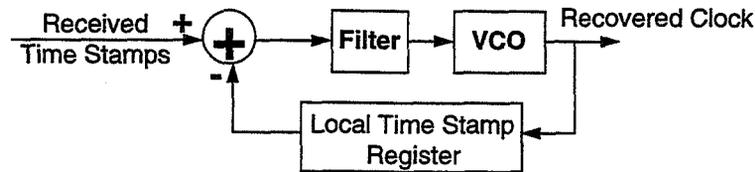


Figure 2: Clock Recovery at the decoder

### 3. Error Control

No computer network is perfect: at one time or another, data will be lost or corrupted. Data can be lost in two ways: the first is congestion in the network. Traditional data traffic is bursty; the backbones of most networks are designed taking this fact into account. Therefore, if a large number of computers decide to transmit at the same time, there will not be enough capacity in the network and data will be lost, either by contention or by buffer overflow. The second loss mechanism is link error: bits may be corrupted in the network links due to noise. Virtually all network protocols add CRCs and checksums to packets; a packet with invalid CRC is typically discarded by the network hardware. Therefore, link errors are equivalent to data loss<sup>1</sup>.

When communicating MPEG streams through a computer network, it is necessary to deal with data loss in the received stream. There are two parts to this issue: first, if data is lost, how to recover it; second, what to do if the lost data is not recovered. In the remainder of this section, we address these two questions.

#### 3.1 Recovering Lost Data: Retransmission or Forward Error Correction?

There are basically two ways of recovering from lost data: (i) the receiver can request a retransmission of the lost data, or (ii) the transmitter can include redundant information in the data stream, from which the receiver may be able to restore the lost data (FEC - Forward Error Correction; more specifically, Erasure Codes). Both methods have advantages and disadvantages. Retransmission has the advantage that no additional data needs to be transmitted in the

<sup>1</sup> Since in today's optical networks the bit error rates are very low, it has been proposed that errored packets be passed to the higher level software with an appropriate indication, since most of the data is still usable.

absence of errors, but it requires that data corresponding to at least one round-trip delay be buffered both at the sender and at the receiver (to allow the receiver time to request the lost packet and receive it before the time that packet is needed to play the stream), thereby increasing the delay and eventually making this method unusable for real-time transmission. Moreover, in a point-to-multipoint scenario (one sender, multiple destinations), the sender might be swamped by requests from the destinations if many of them miss a packet. Erasure codes, on the other hand, introduce very little latency, but the transmission overhead is used all the time. In this section, we present a performance comparison between retransmission and two selected erasure codes in terms of error correction capability and overhead introduced. For the performance evaluation, we chose the Parity Erasure Codes described in [2]; the reader is referred to [6] for other codes, especially chosen for MPEG. We selected two specific codes, which we will denote by PE-SP and PE-TP. In PE-SP, one parity packet is added to each group of  $G$  packets; a single packet loss within these  $G + 1$  packets can be recovered. In PE-TP, two parity packets are added to the group of  $G$  packets, and two losses out of the  $G + 2$  packets can be recovered. For implementation details, see [2,3].

For the analysis, we make the following assumptions:

1. Bit errors are an independent, identically distributed process.
2. Bits are arranged into packets of a given size. Any bit error in a packet will cause that packet to be lost. The probability of undetected bit errors in a packet is assumed to be zero.

The justification for assumption (2) is that packets are protected by a hardware-computed CRC at layer 2, a header checksum at layer 3, and a payload checksum at layer 4. A packet with incorrect checksum or CRC is dropped. Since it would take a very specific pattern of errors for a packet with bit errors to appear to be a valid packet, we assume that the probability of that event is, for all practical purposes, zero.

Definitions:

- $B$  : packet size, in bits
- $G$  : group size, in packets
- $R$  : MPEG stream data rate, in bits/second
- $A$  : aggregate stream data rate (data plus parity)
- $K$  : number of parity packets per group
- $p$  : bit error probability
- $P$  : probability that there is at least one error in a packet (causing it to be lost)
- $P_C$  : packet loss probability *after* error correction (i.e., probability that an error escapes the error correction mechanism)

Our objective, in this analysis, is to find  $P_C$  and  $A$  as a function of  $p$  for:

- retransmission (denoted by  $P_{C,s}$  and  $A_s$ );
- 1-packet erasure code, PE-SP (denoted by  $P_{C,1}$  and  $A_1$ ); and
- 2-packet erasure code, PE-TP (denoted by  $P_{C,2}$  and  $A_2$ ).

If the bit error probability is  $p$ , the probability that a packet contains one or more errors (thus causing it to be lost) is:

$$P = 1 - (1 - p)^B \quad (1)$$

If we use the 1-packet erasure code, uncorrected packet loss will happen if there are 2 or more losses in a group of  $G + 1$  packets:

$$P_{C,1} = 1 - (1 - P)^{G+1} - (G + 1)P(1 - P)^G \quad (2)$$

Similarly, if we use the 2-packet erasure code, uncorrected packet loss will happen if there are 3 or more losses in a group of  $G + 2$  packets:

$$P_{C,2} = 1 - (1 - P)^{G+2} - (G + 2)P(1 - P)^{G+1} - \frac{(G + 2)(G + 1)}{2} P^2(1 - P)^G \quad (3)$$

In the case of retransmission, uncorrected packet loss will happen if both the original packet and the retransmission are lost. Since the losses are independent, the probability of this event is just  $P^2$ . However, to compare the loss probability under retransmission to the numbers in equations (2) and (3), we need to artificially divide the packets in groups of  $G$  and compute the probability that there is one or more (uncorrected) losses in this group of  $G$  packets. Since the losses are independent, this probability is:

$$P_{C,S} = 1 - (1 - P^2)^G \quad (4)$$

The aggregate data rates under erasure codes are not a function of the bit error probability, since  $K$  packets are added to each group of  $G$  packets:

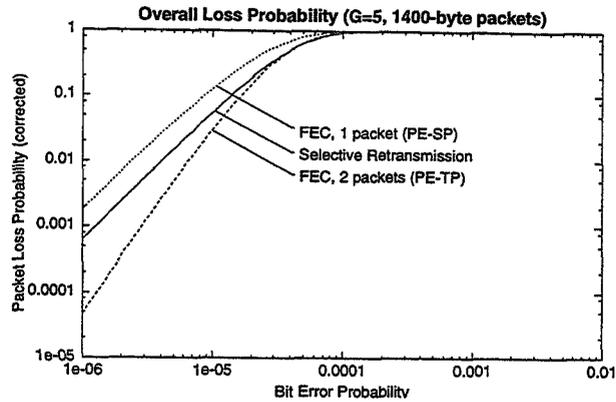
$$A = \frac{G + K}{G} R$$

For 1-packet erasure code:  $A_1 = \frac{G + 1}{G} R \quad (5)$

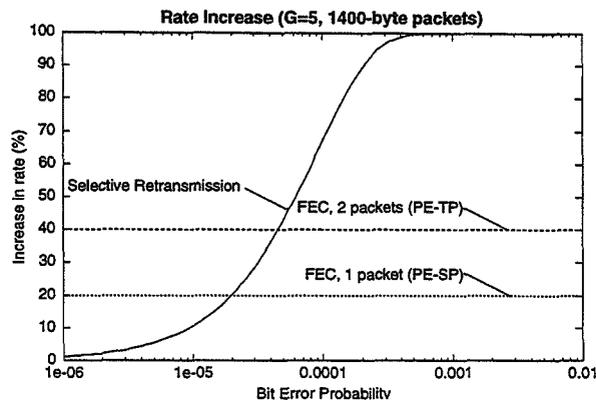
For 2-packet erasure code:  $A_2 = \frac{G + 2}{G} R \quad (6)$

Under retransmission, the aggregate data rate is a function of the bit error probability, since the packets are retransmitted “on demand”:

$$A_S = (1 + P)R \quad (7)$$



(a) Packet loss probability



(b) Aggregate rate

Figure 3: Performance for  $G=5$ , 1400-byte packets

For a given bit error probability  $p$ , the performance of the error correction schemes (as measured by  $A$  and  $P_c$ ) is a function of the packet size  $B$  and the group size  $G$ . For the numerical evaluation, we chose a packet size of 1400 bytes (roughly equivalent to the Ethernet maximum packet size) and a group size  $G$  of 5 packets. Results for other packet sizes and group sizes are similar. Figures 3 (a) and (b) show the packet loss and aggregate rates for the three error control schemes, when the group size is 5 packets. The plots indicate that packet loss probability of the retransmission scheme lies between that of the 1-packet correction scheme and of the 2-packet correction scheme. However, for reasonable bit error probabilities (i.e., in the range of  $10^{-3}$  or less), the overhead associated with retransmission is less than that of the error correction schemes. Note that this analysis assumes only one destination. If there are more destinations, there is no change in the Erasure Code curves, but the overhead for retransmission (Figure 3(b), equation 7) will increase with the number of destinations.

### 3.2 What if lost data is not recovered?

No error correction mechanism is perfect. Therefore, there will occasions when data is lost and cannot be recovered. In this section, we discuss the recommended procedure at the re-

ceiver in such situations. To accomplish that, it becomes necessary to characterize the impact of such errors in the image quality. This evaluation is, by nature, subjective; what one viewer may consider acceptable quality, another may deem unacceptable. Therefore, the conclusions presented in this section reflect the opinions of the authors on what is acceptable video quality.

Data loss will manifest itself as a glitch. If audio data is lost, the audible result is a “click”. There is very little that can be done in case of audio loss; therefore, we will consider only the video. The duration of a glitch can go all the way from a single frame (33 milliseconds) to a whole GOP (0.5 second). In order to characterize the glitches as seen by the viewer, we took MPEG-encoded streams (video rate 1.1 Mb/s, audio rate 128 kb/s) and simulated packet loss when playing them. We assumed 1400-byte packets, and that an errored packet would be deleted from the stream. To the viewer, the glitches appear as:

1. rectangles of a different color briefly appearing at certain points of the screen;
2. part of a frame freezing momentarily; and
3. tiling effects.

We varied the packet loss ratio from under 1% up to 10%. The main conclusion is that the video quality for a packet loss rate in the range of 0.1% is acceptable. Moreover, the MPEG decoder tested (Optivision’s decoder) was very resilient to loss. Therefore, our conclusion is: *for packet losses on the order of 0.1%, no special error handling and concealment is necessary.*

## 4. Latency Issues

As pointed out before, one of the requirements for interactive communications is low latency. Subjective studies made for voice have found that the maximum latency human beings can tolerate is on the order of 250 ms; the same figure also applies to live video communications. The 250 ms figure applies to the end-to-end latency; i.e., the time between the instant a frame of video is grabbed at the encoder and the instant that same frame is displayed at the decoder. Figure 4 shows the components of the end-to-end latency in a MPEG communications system.

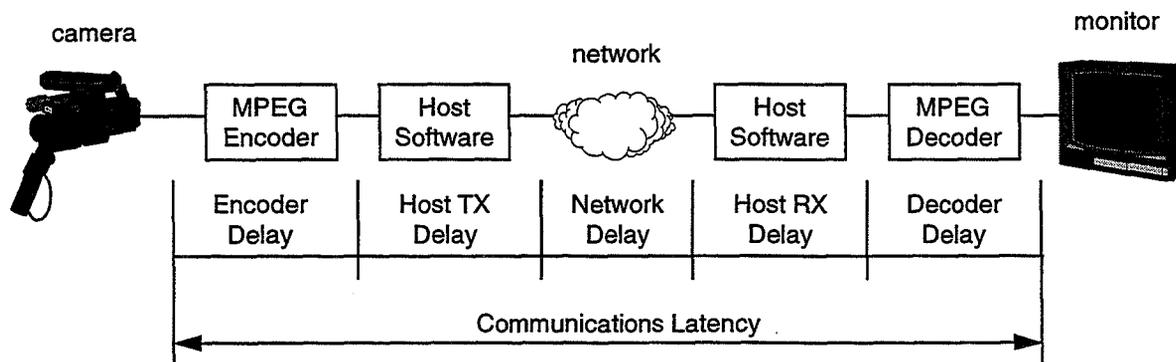


Figure 4: Components of the End-to-End Latency

The components in the communications latency depicted in Figure 4 are:

**Encoder Delay:** this is the delay through the encoder hardware. It is a function both of the software implementation and the frame structure of the compressed signal. Intra-coded frames (I-frames) can be encoded without reference to any other frames; in theory, they can be encoded as they are digitized. Predicted frames (P-frames) need data from the *previous* I-frame or P-frame to be encoded; since this information is already at hand when the frame is being digitized, the theoretical absolute minimum latency is also (almost) zero. However, Bidirectional frames (B-frames) need data both from the *previous* I- or P-frame, as well as the *next* I- or P-frame. Therefore, a B-frame cannot be encoded until the next I or P frame is captured, which means a minimum theoretical latency of two frame times (67 ms at 30 frames/second). Of course, the software implementation adds additional latency over the minimum.

**Host Software Delay (TX):** this is the delay between the moment the video encoder passes the data to the host and the moment that data is transmitted through the network interface. It has two components: (1) processing on the host, and (2) network transmission delay, which is a function of the available network data rate. The different frame types (I, P, B) have different average sizes, but they are still generated at 30 frames/second. Therefore, the intrinsic data generation process at the encoder is inherently bursty (although the average data rate is very precisely controlled). If the network requires a constant fixed data rate, a buffer must be provided to “smooth out” the burstiness; this buffer will introduce additional delay. For example, the average size of an I-frame at 1.15 Mb/s is about 13,000 bytes; if the network can only support constant data rate (as, for example, in a T-1 link), the delay on this buffer would correspond to about 95 ms. On the other hand, if the network is a 10 Mb/s Ethernet segment, the data can be transmitted in a burst at about 10 times that speed, bringing the delay down to about 10 ms.

**Network Delay:** this delay is a function of the network infrastructure, and the distances involved. If the encoder and decoder machines are in the same Ethernet segment, this delay is negligible. On the other hand, if they are in different cities, there may be a significant delay in the network.

**Host Software Delay (RX):** this delay has two components: (1) the network buffer delay, and (2) the host processing delay. The network buffer is responsible for smoothing out the delivery of data to the video decoder by removing the network jitter, and buffering the data as needed for the network protocol. For example, if the system uses a retransmission strategy, the network buffer delay must be at least equal to one round-trip delay. However, if the system uses forward error correction, the network buffer needs only to contain one block of data. For example, if the system uses 1400-byte packets and uses a block size of 5 packets, this delay would correspond to about 49 ms at 1.15 Mb/s. The additional host processing delay corresponds to the time required to demultiplex the audio and the video.

**Decoder Delay:** The decoder delay has similar components to the encoder delay. First of all, if B frames are used, there is a minimum 2-frame delay while the decoder is waiting for the information to decode the frame. Other implementation-specific delays are:

- VBV (Video Buffer Verifier) Delay: this is similar to the network transmission delay; typically, the decoder is not aware of the network bandwidth, and will assume the worst case (i.e., constant bit rate delivery). This means that the decoder will pre-buffer a certain

amount of data prior to start decoding; the latency introduced is equivalent to the network transmission latency discussed for the host software in the transmitter.

- Video Output Delay: Current decoders pipeline decoding and displaying. This pipeline may introduce one to two frames of delay.

As indicated by the above discussion, the latency can easily exceed 250 ms in the worst case, without even considering the network delay. The keys to reducing the latency in MPEG are: (1) do not use B-frames; (2) use the highest available data rate, to reduce buffer latencies.

## **5. Network Infrastructure Issues**

### **5.1 Protocol Issues**

If traditional network infrastructures are to be used to transport MPEG streams, the stream must be packetized and use the same protocols as data in the network. The most common network layer protocol today is the IP protocol, which is supported by most network devices available today. At the transport layer, MPEG streams should use UDP rather than TCP, because the flow control and recovery mechanisms of the latter are unsuitable for video. The industry seems to be moving towards putting MPEG directly in the raw UDP payload; Optivision's transmission systems support this format. Optivision also uses a proprietary format where a sequence number is added for error control and recovery. For ATM networks, the ATM Forum is proposing the use of MPEG-2 transport packets directly as AAL-5 PDUs; in particular, 2 transport packets (376 bytes) fit exactly into an 8-cell AAL-5 PDU with no padding.

### **5.2 Local Area Networks**

The most common local area networks are Ethernet (10 Mb/s) and Token Ring (4 or 16 Mb/s). These networks have the capacity to support a small number of MPEG streams, depending on the individual stream bandwidth [4]. The main problem with these networks is giving priority for video over data; Ethernet does not have any mechanism to implement priorities, and while mechanisms exist in Token Ring, they are typically not implemented in most devices. Our experience is that, although it is possible to mix video and data in the same network, performance for data users will be greatly degraded. In these networks, video and data should be kept in different segments.

### **5.3 Wide Area Networks**

By and large, wide area networks are point-to-point links that connect switching devices such as routers. It is possible to use these wide-area links either by injecting the MPEG streams in a LAN connected to a WAN router, or by having the MPEG encoder/decoder directly terminate the WAN link. Optivision's MPEG transmission systems support both options.

## 5.4 ATM

ATM is intended to be a single standard both for the local and the wide-area networks. Since it is switch-based and not shared-medium, the available bandwidth increases linearly with the number of ports. Moreover, ATM has a signaling protocol that allows for priorities and resource reservation. During the current transition period, ATM will be adopted first in the backbone and for devices that need a high-speed connection, and then will migrate to the desktop [5]. As pointed out in Section 5.1, the ATM Forum is in the process of standardizing the transport of MPEG-2 over ATM networks.

## 6. References

- [1] S. Dixit and P. Skelly, "MPEG-2 over ATM for Video Dial Tone Networks: Issues and Strategies," *IEEE Network*, Sept/Oct 1995, Vol. 9, No. 5, pp 30-40.
- [2] N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management," *Proc. INFOCOM'90*, pp. 124-131, San Francisco, CA, June 1990.
- [3] Optivision, Inc., "Multicast Transport of Compressed Audio and Video in Computer Networks," *NSF Phase I Final Report*, August 1995.
- [4] I. Dalgıç, W. Chien and F. Tobagi, "Evaluation of 10Base-T and 100Base-T Ethernets Carrying Video, Audio and Data Traffic," *IEEE INFOCOM 94*, Toronto, Canada, June 1994, pp 1094-1102.
- [5] C. Noronha and F. Tobagi, "The Evolution of Campus Networks Towards Multimedia," *Proceedings of COMPCON Spring'93*, San Francisco, CA, February 1993.
- [6] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *35th Annual Symposium on Foundations of Computer Science*, IEEE Computer Science Press, 1994.

# Global and Local Distortion Inference During Embedded Zerotree Wavelet Decompression

A. Kris Huber and Scott E. Budge  
Electrical and Computer Engineering Dept.  
Utah State University  
Logan, Utah 84322-4120  
e-mail: kris@ece.usu.edu & scott@goga.ece.usu.edu

56-82  
013177  
268901  
10 r.

## Abstract

This paper presents algorithms for inferring global and spatially local estimates of the squared-error distortion measures for the Embedded Zerotree Wavelet (EZW) image compression algorithm. All distortion estimates are obtained at the decoder without significantly compromising EZW's rate-distortion performance. Two methods are given for propagating distortion estimates from the wavelet domain to the spatial domain, thus giving individual estimates of distortion for each pixel of the decompressed image. These local distortion estimates seem to provide only slight improvement in the statistical characterization of EZW compression error relative to the global measure, unless actual squared errors are propagated. However, they provide qualitative information about the asymptotic nature of the error that may be helpful in wavelet filter selection for low bit rate applications.

## 1 Introduction and Motivation

An image compression algorithm encodes data from a two-dimensional information source for efficient transmission and later reconstruction by a decoder for some intended receiver. The receiver performs some sort of analysis of the image. Lossy compression algorithms produce reconstructed images that are not in general equal to the input image. Any measure of the error or the nature of the error induced by a compression algorithm is commonly termed a *distortion measure* by data compression technologists. In this context, distortion is a more general term for error which must not be confused with the image distortion of optical aberration theory that refers to the spatial variation of the magnification from input to output of an optical system (e.g. pincushion or barrel distortion) [1]. Many distortion measures for image compression have been devised and studied [2]. Probably the most commonly used distortion measures are those related to the sum-of-squares of the errors, such as the peak signal-to-noise ratio (PSNR), mean squared error (MSE), and root-mean-squared error (RMS error). These measures fit in nicely with the standard error analysis methodologies used in most science and engineering work [3, 4]. The distortion measures that may be inferred using the algorithms presented here are these squared-error measures. For convenience, only MSE will be presented in the graphs and tables.

Compression with global and local distortion inference is a step toward the design of an image communication system in which the compression algorithm is optimized for a receiver intending to perform an uncertainty analysis based upon the data. For example, a scientist (or his computer program) may be measuring some parameters of his hypothetical model based upon an EZW-compressed image. A typical approach to this problem is to perform Monte Carlo simulations of the compression/analysis system and generate uncertainty estimates in the derived results on the basis of simulation. Error estimates based on how well the data fits the model may also be generated. Distortion estimates obtained by global and/or local distortion inference can be used to supplement this approach and may lead to more accurate uncertainty estimates for the model parameters. In addition, the inferred compression distortion may assist in determining when and where deviations from the simulated conditions occur.

Section 2 describes the EZW algorithm. Section 3 explains two methods for global distortion inference and shows the measured and inferred distortion-rate curves for five test images. Section 4 describes two local distortion inference methods, presents a comparison to global distortion inference based on error coverage

factor, and discusses results. Images are shown that illustrate qualitative results. Section 5 is a summary and conclusion.

## 2 Description of the EZW Algorithm

The Embedded Zerotree Wavelet Algorithm was introduced by J. Shapiro [5]. It simultaneously possesses several desirable features including progressive transmission (the image can be reconstructed at intermediate points during decompression), embedded coding (different bit rates and distortions can be obtained by simply truncating the compressed file), and fast speed (relative to the standard DCT-based algorithms) due to the wavelet transform. Its principal disadvantage is a large memory buffer. The basic operation of the EZW image compression algorithm can be explained as follows:

Transform the input image hierarchically with a discrete orthonormal wavelet of choice. Consider the resulting wavelet coefficients to be linked in a tree structure based on spatial location. Consider each top-level (LL subband) coefficient as the parent of three coefficients (one each in the top-level LH, HL, and HH subbands). Consider the remaining coefficients to be linked with four children per parent, all from subbands of the same orientation (e.g. HL parent has four children in the HL subband on the next level down). Subtract the mean of the LL coefficients and code it in a header with high accuracy. It is helpful to apply the filters with shifts based on the *center of energy* of the filters [6] so that the wavelet coefficients are stored in locations that correspond well to the location of the spatial-domain data upon which they most strongly depend. See [7] or [6] for separable wavelet transformation details.

Regard wavelet coefficients as *insignificant* with respect to a threshold  $T$  if they are less than  $T$ , and *significant* otherwise. Consider the set of known significant coefficients to belong to the *subordinate list* and the rest of the coefficients to belong to the *dominant list*. Initialize  $T$  large enough that all coefficients belong to the dominant list. Consider subtrees of wavelet coefficients to be *zerotrees* with respect to  $T$  and the top-most parent of a zerotree to be the *zerotree root* whenever it and all its descendants are insignificant or on the subordinate list. Consider an insignificant coefficient to be an *isolated zero* if it is insignificant, but at least one of its descendants is not.

**Dominant pass:** Use a four-symbol adaptive arithmetic coder [8] to entropy code one of the following symbols: zerotree root, isolated zero, positive significant, and negative significant. Code a symbol only if the coefficient is on the dominant list and if its insignificance is not already known due to a previously coded zerotree root symbol. Move significant coefficients to the subordinate list as they are found. Divide  $T$  by 2 after completing the pass.

**Subordinate pass:** Use a two-symbol arithmetic coder to refine the significant coefficients by encoding the next-most-significant bit.

If the the target bit rate or distortion is reached at any time during dominant or subordinate passes, terminate the coding. Otherwise continue on to the next dominant pass.

Decoding is the inverse of this process. Dominant list coefficients are inverse-quantized to zero. Subordinate list coefficients are inverse quantized by inserting a 1 bit in the position below the least significant bit decoded and multiplying by the value of  $T$  at the time the subordinate coefficient was last refined or was moved to the subordinate list. This inverse quantization method places reconstruction points in the center of quantization bins of width  $2T$  and  $T$  for dominant and subordinate list coefficients, respectively.

A more detailed description is given in the Appendix.

## 3 Global Distortion Inference

The wavelet transform assumed here is orthonormal and separable. Low-pass filter coefficients therefore sum to  $\sqrt{2}$  and the sum of squares of the coefficients is 1. The sum of high-pass filter coefficients is 0. Orthonormal wavelet decomposition is a unitary transformation. As such, energy is preserved across the transformation, i.e. the sum of squares of the wavelet coefficients is equal to the sum of squares of the input image. Moreover quantization error in the wavelet domain leads to an image reconstruction exhibiting the same error energy. Because of this it is possible to know the squared-error distortion (and its relatives PSNR, MSE and RMS error) precisely in the encoder by maintaining running sums of dominant and subordinate list quantization

## MSE Distortion vs. Rate, Measured and Inferred

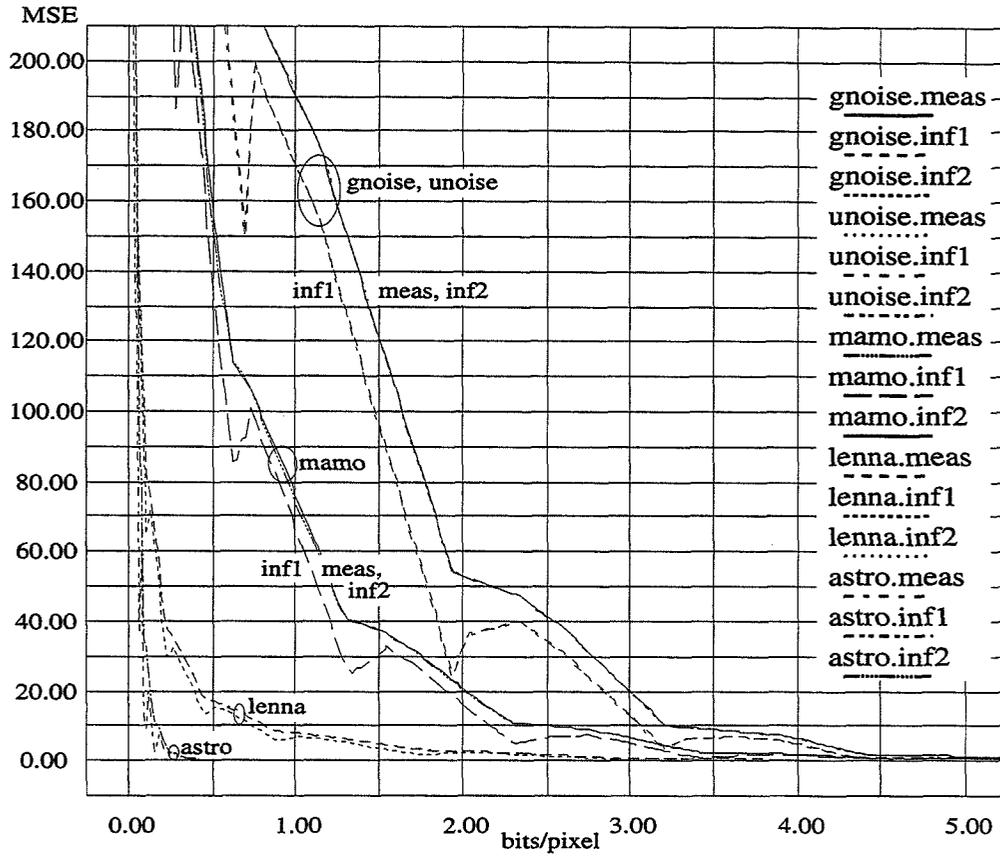


Figure 1: MSE distortion vs. rate curves for five test images. Measured (meas), global inference method 1 (inf1), and global inference method 2 (inf2) curves are indicated.

error energies. The distortion for each bit rate can be inferred at the decoder from estimates of the wavelet coefficient error energy.

At all times during the EZW coding total energy remains constant ( $E_{tot} = E_{DL} + E_{SL}$ , where  $E_{tot}$  is the total energy of the image,  $E_{DL}$  and  $E_{SL}$  are the energy of coefficients on the dominant and subordinate lists, respectively). The error energy can be written  $E_{err} = E_{DL} + E_{err,SL}$ , where  $E_{err}$  is the total error energy in the reconstruction, and  $E_{err,SL}$  is the error energy in the subordinate list coefficients. Dominant list coefficients are inverse quantized to 0, therefore  $E_{DL}$  contributes directly to the error energy.

With negligible increase in the number of compressed bits ( $\approx 40$ ) it is possible to accurately encode the total energy,  $E_{tot}$ . Energies may then be estimated in the decoder using

$$\hat{E}_{DL} = E_{tot} - \hat{E}_{SL} \quad (1)$$

$$\hat{E}_{err} = \hat{E}_{DL} + \hat{E}_{err,SL}. \quad (2)$$

The following method of estimating  $E_{err,SL}$  works well. Set  $\hat{E}_{err,SL}$  to 0 (the subordinate list is initially empty). As each significant coefficient is found during a dominant pass, update  $\hat{E}_{err,SL}$  based on the assumption that the quantization error of the subordinate coefficients is uniformly distributed. The energy of each subordinate coefficient is therefore  $\frac{T^2}{12}$  (since the significance threshold,  $T$ , is the current quantization bin width for significant coefficients). The update equation for the dominant pass is therefore

$$\hat{E}_{err,SL} = \hat{E}_{err,SL} + \frac{T^2}{12}. \quad (3)$$

This is performed every time a significant coefficient is found during a dominant pass. Prior to a subordinate pass,  $T$  is halved, resulting in the update equation

$$\hat{E}_{err,SL} = \hat{E}_{err,SL} - \frac{T^2}{4}. \quad (4)$$

To estimate subordinate list energy, begin by initializing  $\hat{E}_{SL}$  to 0. Basing subordinate list energy on the sum of squares of the (would-be) inverse-quantized wavelet coefficients leads to the update equation

$$\hat{E}_{SL} = \hat{E}_{SL} + \frac{9}{2}T^2 \quad (5)$$

for the dominant pass, and for each coefficient refined during the subordinate pass

$$\hat{E}_{SL} = \hat{E}_{SL} + \begin{cases} \frac{1}{4}T^2 + Tw & \text{if current LSB (least significant bit) = 1} \\ -\frac{3}{4}T^2 - Tw & \text{otherwise,} \end{cases} \quad (6)$$

where  $w$  is the (integer) word containing the bits that have been decoded for the current significant coefficient. In the above, the current bit is already assumed to be shifted into  $w$ .

The above estimate for  $\hat{E}_{SL}$  is inadequate for reasons explained in the next section. A second method that gives better distortion inferences is now explained. Before each dominant pass, the encoder transmits the fraction of energy that will remain on the dominant list at the end of the pass, along with the number of coefficients that will be found during the pass,  $\Delta n$ . From this, the decoder determines an energy change,  $\delta_E = \frac{\Delta E_{DL}}{\Delta n}$ , to use during the current dominant pass. It adds  $\delta_E$  to  $\hat{E}_{SL}$  every time a coefficient is moved to the subordinate list:

$$\hat{E}_{SL} = \hat{E}_{SL} + \delta_E, \quad (7)$$

and performs no update during the subordinate pass. The encoder can calculate the energies for all passes in the process of calculating  $E_{tot}$ , thus little computational burden is added to the encoder. This approach assumes that the increase in subordinate list energy during a dominant pass is directly proportional to the number of coefficients found significant since the beginning of the pass. This method differs from the first method only in the way the subordinate list energy is estimated and the small additional amount of overhead information (less than  $\approx 60$  extra bits per bitplane).

### 3.1 Global Distortion Inference Results

Five test images were compressed to evaluate the accuracy of distortion inferences using the methods described above. Four stages of wavelet decomposition were performed using the Beylkin 18-tap filter [6] with periodic extension used at image boundaries. The adaptive arithmetic coder used a maximum frequency of 127 and a 9-bit code point [8]. A constant 73-bit coding delay between encoder and decoder was empirically determined, and the rate measured in the encoder was increased accordingly.

The measured and inferred MSE distortion vs. rate curves are shown in Figure 1. The images were: gnoise (i.i.d. Gaussian noise with  $\mu=128$  and  $\sigma=20$ ), unoise (i.i.d. uniformly distributed noise with  $\mu=128$  and  $\sigma=20$ ), mammo (a mammogram image), lenna (ftp://ipl.rpi.edu/pub/image/still/usc/gray/lenna512.ras), and astro (a high-resolution astronomical image from simulation work of the first author [9]). All images were 512x512 pixels stored in floating-point format. It is interesting that the EZW performance was insensitive to the distribution shape for the noise files.

Figure 1 shows that Method 1 yields poor global inferences of the MSE. This is because, first, at interesting bit rates there are few coefficients on the subordinate list, yet their sum-of-squares,  $\hat{E}_{SL}$ , constitutes a large fraction of the total energy. Both  $E_{tot}$  and  $\hat{E}_{SL}$  are very large numbers, the difference of which must be accurately known to infer  $\hat{E}_{DL}$ . Appropriately small errors in subordinate list coefficients lead to large enough errors in their squares to render  $\hat{E}_{DL}$  nearly useless. Second,  $\hat{E}_{err}$  is consistently too low, due to positive bias in  $\hat{E}_{SL}$ . The bias is probably caused by use of the quantization bin center as the reconstruction point, in spite of the slight slope of the typical wavelet coefficient probability density function across a quantization bin. For high bit rates and at the end of each of the subordinate passes, Method 1 gives its best performance and is perhaps acceptable for some applications. The Method 2 distortion inference curves shown in Figure 1 are very good. For each image the inferred distortion-rate curve is difficult to distinguish from the measured curve.

## 4 Local Distortion Inference

The work in this section was motivated by the observation that much of the structure of an image is typically visible in the error image, thus the compression error is spatially nonuniform. This can be demonstrated very easily by compressing an image at a moderate-to-low bit rate with most any compression algorithm, and viewing the difference image between input and output. The structure of the input image will appear in the error image as regions of larger errors in the locations where edges and fine details occur. This is so because of the low-pass nature of nearly all lossy compression algorithms. The desire to reduce the spatial nonuniformity of error without penalizing compression performance led to the analysis and algorithms described below.

A method of spatially whitening the compression error is to add a dither sequence prior to quantization and subtract the dither after inverse quantization [10]. Pondering the application of this technique to EZW, however, one realizes that the dither sequence will appear as noise to the coder whether it is added in the spatial or wavelet domain. One will actually expend more bits to produce a reconstruction that has higher average error than the reconstruction obtained without dithered quantization.

Since the relatively low error in smoother regions of an image is available *for free* using the EZW algorithm, a more efficient approach is to allow the nonuniformity to occur and to model it. A method of doing this, here termed *local distortion inference*, is to use what information is known in the decoder about the accuracy of wavelet coefficients to estimate errors in the decoded image. The resulting noise image is then propagated into the subsequent image analysis, leading to improved error characterization for the resulting data products.

Both methods of local distortion inference described next use an estimate of error energy for each wavelet coefficient, referred to collectively as the wavelet-domain *noise image*. This can be calculated by allocating the appropriate portion of  $E_{tot}$  to each coefficient depending upon its membership in the dominant list, the subordinate list, and the final state of the decoder. If termination occurred partway through a subordinate pass, each coefficient that was refined should be allotted 1/4 of the error energy allocated to each remaining subordinate list coefficient. If the algorithm terminated during a dominant pass, all coefficients examined directly or indirectly (by being descendants of zerotree roots that were decoded) are allocated 1/4 of the energy of the coefficients not yet significance-tested.

### 4.1 Error Propagation Transform for Local Distortion Inference

The filters involved in inverse wavelet filtering are especially designed to cancel the aliasing distortion due to decimation of the forward-transform filter outputs. But this cancellation is complete only in the absence of quantization error. One way to model the hierarchical inverse wavelet transform in the presence of significant quantization error is to simply regard it as the application of ordinary linear finite impulse response (FIR) filters at the various resolutions involved. A method of inferring distortion locally in an EZW reconstruction can be based on this view by applying the error propagation equation common in measurement uncertainty analysis [4] to the inverse filtering process.

To apply this approach, error energy estimates are first obtained for each wavelet coefficient based on precision and significance list membership information as discussed previously. For each FIR filter application  $x_j = au_m + bu_n + \dots$ , where  $j$ ,  $m$ , and  $n$  represent appropriate position indices, error variance and covariance estimates  $\sigma_{u_m}^2, \sigma_{u_n}^2, \sigma_{u_m u_n}^2, \dots$ , for the filter inputs can be propagated to the filter output using

$$\sigma_{x_j}^2 \simeq \sigma_{u_m}^2 \left( \frac{\partial x_j}{\partial u_m} \right)^2 + \sigma_{u_n}^2 \left( \frac{\partial x_j}{\partial u_n} \right)^2 + \dots + 2\sigma_{u_m u_n}^2 \left( \frac{\partial x_j}{\partial u_m} \right) \left( \frac{\partial x_j}{\partial u_n} \right) + \dots \quad (8)$$

$$= a^2 \sigma_{u_m}^2 + b^2 \sigma_{u_n}^2 + \dots + 2ab\sigma_{u_m u_n}^2 + \dots \quad (9)$$

This equation is based upon consideration of the spread of the values of  $x_j$  resulting from combining the individual “measurements” (inverse-quantized wavelet coefficients  $u_m, u_n, \dots$ ) about an assumed most probable value  $\bar{x}_j = f(\bar{u}_m, \bar{u}_n, \dots) = a\bar{u}_m + b\bar{u}_n + \dots$  [4].

Application of this error-propagation formula, neglecting the covariances,  $\sigma_{u_m u_n}^2, \dots$ , can be implemented very simply: inverse transform the wavelet-domain noise image using the squares of the inverse wavelet filter coefficients ( $a^2, b^2, \dots$ ), rather than the filter coefficients themselves ( $a, b, \dots$ ). Any explicit subtractions must be replaced with additions as well. This *error-propagation transform* has equal complexity to the inverse wavelet transformation. Discussion of the effects of ignoring the covariance terms is deferred to Section 4.3.

Image	Filter	Rate	MSE	Global Inference			Local Inference		Prop. Error	
				MSE	$k_{95.5\%}$	$k_{99.7\%}$	$k_{95.5\%}$	$k_{99.7\%}$	$k_{95.5\%}$	$k_{99.7\%}$
astro	H2	0.077	217.5	216.6	2.073	4.655	2.024	4.051	1.909	2.513
astro	D12	0.060	98.0	97.4	2.033	3.565	2.017	3.434	2.245	3.540
astro	B18	0.063	93.4	92.8	2.026	3.474	2.012	3.410	1.994	3.004
mamo	H2	0.103	576.0	573.4	2.147	3.567	2.138	3.536	1.932	2.568
mamo	D12	0.118	480.2	477.6	2.138	3.536	2.126	3.508	2.092	3.176
mamo	B18	0.117	484.8	483.0	2.139	3.585	2.123	3.521	1.961	2.903
lenna	H2	0.578	22.2	22.1	2.139	3.607	2.136	3.625	1.889	2.498
lenna	D12	0.455	17.9	17.8	2.129	3.732	2.117	3.698	1.972	2.871
lenna	B18	0.233	37.9	37.8	2.183	4.056	2.139	3.910	1.944	2.887
unoise	H2	6.517	0.093	0.093	1.983	2.887	1.972	2.769	1.895	2.434
unoise	D12	6.516	0.093	0.093	1.988	2.939	1.983	2.887	1.950	2.753
unoise	B18	6.518	0.093	0.093	1.992	2.957	1.987	2.917	1.965	2.817
gnoise	H2	6.525	0.093	0.093	1.982	2.885	1.974	2.781	1.893	2.431
gnoise	D12	6.525	0.093	0.093	1.986	2.940	1.980	2.875	1.947	2.738
gnoise	B18	6.526	0.093	0.093	1.989	2.965	1.987	2.917	1.967	2.820

Table 1: Error coverage factors for global and local distortion inferences, and error propagation transform of wavelet-domain squared error for three filters (H2=Haar, D12=Daubechies, B18=Beylkin).

## 4.2 An Iterative Method of Local Distortion Inference

Inverse hierarchical wavelet transformation is a linear operation. This means that for any two images of wavelet coefficients, the inverse wavelet transform of the sum of the coefficients equals the sum of the inverse transforms of the two wavelet coefficient images. Considering the quantization error added to wavelet coefficients by lossy compression as one of these images, one realizes that for any such quantization-error image, the quantization error image in the spatial domain can be obtained through inverse transforming the wavelet-domain error image.

The iterative method for local distortion inference is to update the necessary statistic (the sum of squares) in the spatial domain after each of many samples of pseudorandom error images are inverse transformed. All pseudorandom images are generated to be consistent with what is known about coefficient precisions and list membership as described in Section 4. After completing the desired number of iterations, the *accumulator image* containing the spatial domain energy resulting from all iterations, is divided by the number of iterations to obtain a spatial-domain noise image. This method has the advantage that any desired noise distributions and correlation models can be simulated by modifying the random number generation process. It is also possible to accumulate other statistics, such as the absolute sum which would give a local estimate of the mean absolute error (MAE).

## 4.3 Local Distortion Inference Results

Table 1 gives the results of an experiment to investigate the local distortion inference performance using the error propagation transform of Section 4.1. The eight most significant bitplanes of quantized wavelet coefficients were encoded using the EZW algorithm for each of the five test images. Three filters were used, the Haar, Daubechies 12-tap, and Beylkin 18-tap filters [6]. Method 2 for global distortion inference was used. Since coding was halted after a complete bitplane (after the eighth dominant pass), distributing error energy was easier to implement than in the general case: the subordinate list error energy,  $\hat{E}_{err,SL}$ , and the dominant list energy,  $\hat{E}_{DL}$ , were divided evenly among subordinate and dominant list coefficients, respectively. The error propagation transform was then applied to these energies to produce the *local inference* image. A *propagated error* image was also generated by applying the transform to the actual wavelet-domain squared errors. Although actual error is not normally available at the decoder, the performance of this prediction gives some indication of a limit to the achievable performance of more sophisticated techniques of estimating

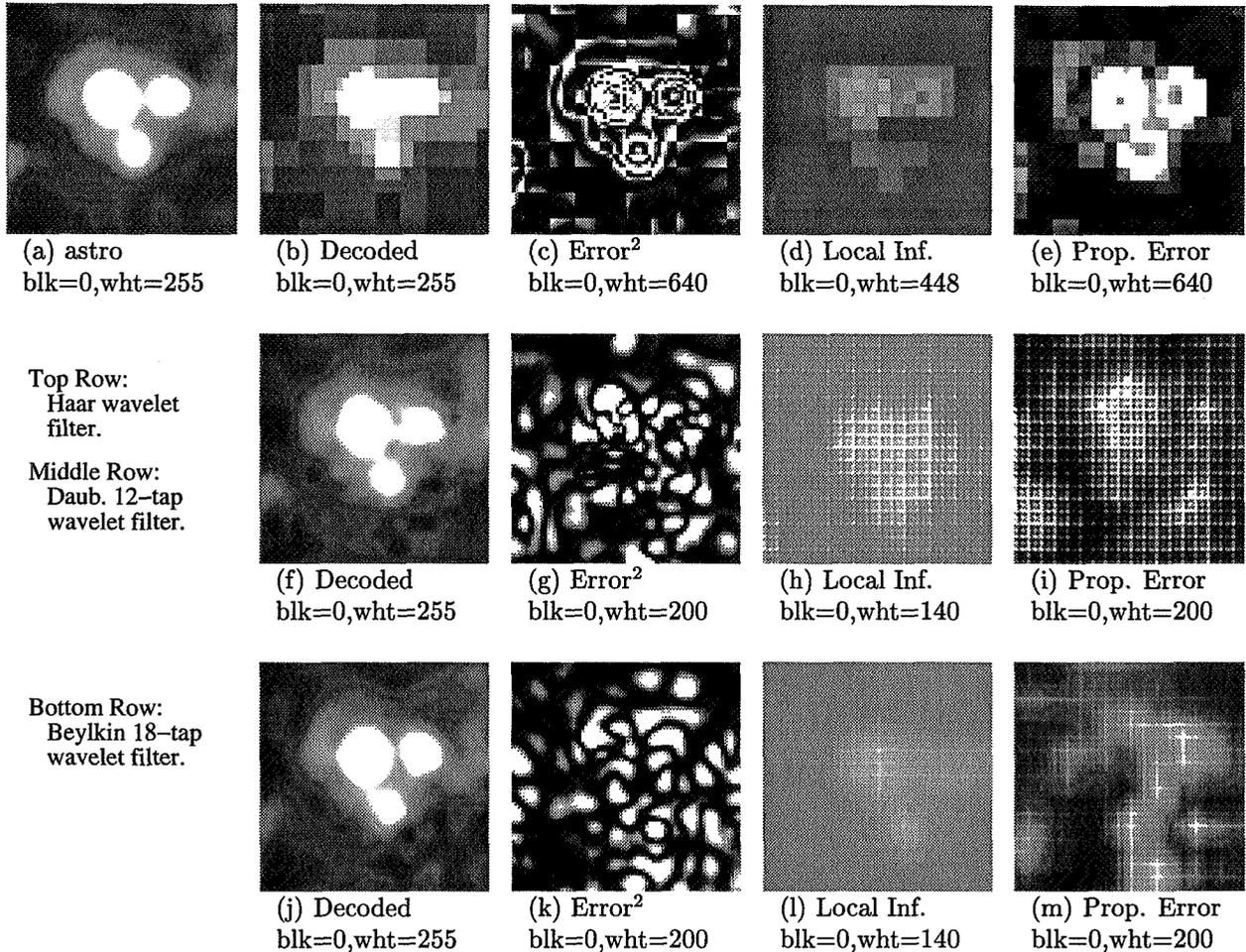


Figure 2: 72x72 pixel subimages associated with results in Table 1 for astro. Data numbers corresponding to black (blk) and white (wht) indicate scaling of images for display purposes.

the wavelet-domain error energies.

In addition to bit rates and MSE measures (actual and the global inference), the 95.5% and 99.7% coverage factors are given in Table 1. Coverage factor,  $k_p$ , refers to the factor by which one multiplies the uncertainty  $u$  ( $\approx$ RMS error) in order to have confidence level  $p$  that the true value is within  $\pm k_p u$  of the estimate [3]. For a Gaussian error distribution,  $u$  is equal to the standard deviation  $\sigma$ ,  $k_{95.5\%}=2$ , and  $k_{99.7\%}=3$ . The coverage factors in Table 1 were calculated from histograms of uncertainty-normalized error images. The normalization was done by dividing each error image pixel by the RMS error predicted by the distortion inference. Lower values of coverage factor are indications of improvement in the error estimate. The comparison of global and local distortion inference by looking at coverage factors is “fair” in the sense that all distortion estimates have the same total energy.

Table 1 shows that  $k_{95.5\%}$  and  $k_{99.7\%}$  were reduced on the order of 1% by using the local distortion inference. An exception was a 0.5% increase in  $k_{99.7\%}$  for the Haar-filtered Lenna image. The most dramatic improvement occurred for the Haar-filtered astronomical image, for which local distortion inference reduced the coverage factors by 2.4% and 13%, respectively. The error propagation transform of the actual squared errors reduced  $k_{95.5\%}$  on the order of 5% for both filters. Reductions in  $k_{99.7\%}$  occurred on the order of 15% for the Haar and 25% for the Beylkin filter. The most significant reductions occurred for the files that compressed to lower rates.

Portions of the decoded, squared error, local inference, and propagated error images are shown in Figure 2 for the astronomical test image. There was a qualitative difference in the predicted squared error that

depended upon the filter. Blocky artifacts appeared when the Haar filter was used. Grid- and streak-like patterns such as those in Figures 2(h), 2(i), 2(l), and 2(m) appeared when other wavelet filters were used. The tendency of higher or lower error predictions along particular rows and columns to form a grid pattern was unexpected and quite puzzling. Such grid patterns appeared in the predicted squared error images for all image types when compression was performed at sufficiently low bit rates. As the bit rate was increased, the grid patterns in the propagated error images became less coarse until they vanished altogether.

Based on a suspicion that such patterns were not valid predictions, an experiment was performed using the iterative method for error propagation of Section 4.2. For the Haar filter, 16,384 iterations of pseudorandom noise generation, inverse wavelet transformation, and spatial-domain error energy accumulation was sufficient for the resulting image to converge very closely to the image calculated with the error propagation transform. For the Beylkin filter, 16,384 iterations produced an image that had not yet converged to the local inference image, but streaks had formed along the same grid lines. Thus the unusual patterns in local distortion inferences appear to be valid indicators of an asymptotic tendency of the spatial error energy distribution of statistically independent wavelet-domain quantization errors.

It was found that filters for which the sum of squares of even- and odd-tap coefficients are equal seem to produce lower-contrast grid patterns in the local inference and propagated error images. The Beylkin 18-tap filter is good and the Haar is perfect in this regard, while the Daubechies 12-tap and most other wavelet filters are poor in this regard. Further study along these lines may yield interesting if not important filter design principles for low bit rate wavelet compression applications. At high bit rates, of course, the error image and any associated pattern approach zero. The patterns are only significant at low bit rates where the assumption of independent errors is perhaps not valid.

The error propagation transform neglects contributions to the error due to *cross terms* of the wavelet filter coefficients (the terms multiplying the covariances in Equation 9). For example, consider the errors in a pair of adjacent pixels reconstructed using the Haar transform. The errors are the inverse transform of two wavelet-domain quantization errors  $e_1$  and  $e_2$ . Some algebra shows that the errors in the spatial domain will be  $\frac{e_1^2+e_2^2}{2} + e_1e_2$  and  $\frac{e_1^2+e_2^2}{2} - e_1e_2$ , whereas the error propagation transform predicts an error of  $\frac{e_1^2+e_2^2}{2}$  for both samples. The difference can clearly be substantial. The situation is more complex with longer filters (e.g. the Daubechies 4-tap filter [6] has 6 cross terms), but the cross terms continue to be significant. In a hierarchical wavelet decomposition, cross terms partially cancel due to contributions of opposite sign, but even if quantization error is independent, the cancellation will not be complete for the individual samples of reconstruction error that are of the greatest interest from a local distortion inference standpoint. Efficient use of covariance or other techniques to model effects due to cross terms remains a topic for further study.

## 5 Summary and Conclusions

Two methods of global distortion inference during decompression of EZW-coded images were described. Both required transmission of the total error energy from the encoder to the decoder. The second method, which required slightly more overhead, yielded very accurate distortion-rate curves.

Two local distortion inference methods were presented. Reductions on the order of 1% in error coverage factor were obtained for the local distortion inference using the error propagation transform relative to the global distortion estimate. While not dramatic, they were realizable local error characterizations that require no rate overhead beyond that of the global distortion inference. The propagation of actual wavelet-domain squared errors gave significant improvement in terms of coverage factor. Although unrealizable in a practical decoder, it indicated that maximum coverage factor reductions of no more than 5-25% are to be expected of more sophisticated schemes of wavelet-domain error energy estimation.

Images were shown that demonstrated the ability of the local distortion inference using the error propagation transform to give qualitative information about the asymptotic nature of the compression error at low bit rates. The occurrence of grid- and streak-like patterns in the resulting local squared-error predictions was discussed and the iterative method for local distortion inference confirmed their existence.

The main contribution of this paper was to study a wavelet-based compression algorithm in the context of uncertainty analysis. This led to a natural way of viewing the wavelet reconstruction process in the presence of quantization error. Further investigation of wavelet based compression in this context may lead to improved filter designs and compression algorithms, particularly for low bit rate scientific applications.

## Acknowledgment

This work was supported under the Space Dynamics Laboratory/Utah State University subcontract 959786 with the California Institute of Technology Jet Propulsion Laboratory under NASA Contract NAS7-918.

## References

- [1] F. L. Pedrotti and L. S. Pedrotti, *Introduction to Optics*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [2] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Trans. Commun.*, vol. 43, no. 12, pp. 2959–2965, Dec. 1995.
- [3] ISO/IEC/OIML/BIPM, *Guide to the Expression of Uncertainty in Measurement*. ISO/TAG 4/WG 3, June 1992.
- [4] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*, 2nd ed. New York, NY: McGraw-Hill, 1992.
- [5] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
- [6] M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*. Wellesley, MA: A. K. Peters, 1994.
- [7] H. J. Barnard, J. H. Weber, and J. Biemond, "Efficient signal extension for subband-wavelet decomposition of arbitrary length signals," *Proc. SPIE*, vol. 2094, pp. 966–975, Oct. 1993.
- [8] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, June 1987.
- [9] J. C. Kemp, R. W. Esplin, and P. B. Hacking, "Sensitivity model for the wide-field infrared explorer mission," *Proc. SPIE*, vol. 2553, pp. 26–37, Sept. 1995.
- [10] L. G. Roberts, "Picture coding using pseudo-random noise," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 145–154, Feb. 1962.
- [11] V. R. Algazi and R. R. Estes, Jr., "Analysis based coding of image transform and subband coefficients," *Proc. SPIE*, vol. 2564, pp. 11–21, Aug. 1995.

## Appendix

A more detailed description of the authors' implementation of the EZW algorithm is given below.

1. Wavelet transform the input image, remove the mean of the top-level coefficients, and consider the coefficient relationships as described in Section 2.
2. Scalar quantize the wavelet coefficients by dividing by a small threshold,  $T_{min}$ , and keeping the integer portion. Represent negative integers using signed-magnitude format.
3. Place the quantized coefficients from each subband in a Peano-scan order (consecutive coefficients spatially adjacent) in a 1-dimensional array ( $qc$ ). Make sure the order is consistent from level to level so that the children corresponding to the same parent are consecutive as well. See [11] for an illustration of the Peano-scan ordering and alternative orderings.
4. Create a significance map ( $sig\_map$ ), an array of integers. Initialize it by placing a 1 bit in each  $sig\_map$  entry in the position of the most significant magnitude bit of the corresponding quantized coefficient. For each level of the pyramid, bitwise OR each group of four child entries with their immediate parent entry and store the result in the parent entry.

5. Consider all coefficients to belong to the *dominant list* unless pointed to by an entry on the *subordinate list* (*sub\_list*). As coding progresses, mark *sub\_list* coefficients by setting the sign bit of the corresponding significance map (*sig\_map*) entry.
6. Initialize *slice\_num* to the bit position of the most significant bit (MSB) of the largest coefficient magnitude. Set  $slice\_mask = 1 \ll slice\_num$  for testing and setting bits on the current bitplane. These variables represent the descending threshold  $T$ . The relationship is  $T = slice\_mask \times T_{min}$ .
7. Dominant pass: Encode the sign and position of the quantized wavelet coefficients having MSBs at the current bitplane. To do this scan through the *sig\_map* and *qc* arrays, performing actions based on the value of parent and child *sig\_map* and *qc* entries. For conciseness of explanation, consider a vector,  $v$ , to be comprised of four bits from the current *sig\_map* and *qc* entries such that  $v = (MSB_{sig\_map}, CBB_{sig\_map}, SGN_{qc}, CBB_{qc})$ . Here SGN indicates the sign bit and CBB means the “current bitplane bit” which may be tested by ANDing the entry with *slice\_mask*. Proceed as follows:
  - (a) Check the contents of the *sig\_map* entries of all parents (grandparents, etc.), beginning at the top of the tree structure. If a zero is found in the *slice\_num*th bit position of any parent, the current wavelet coefficient (and its siblings) are *predictably insignificant* since they are part of a previously coded zerotree. Advance past them to the next entry.
  - (b) If  $v=(1,x,x,x)$ , where ‘x’ indicates a *don’t care* condition, then the coefficient is already on the subordinate list. Proceed to the next entry.
  - (c) If  $v=(0,1,x,1)$  a new significant coefficient has been found. Code it with a Positive Significant or Negative Significant symbol according to the value of the  $SGN_{qc}$  bit, then set all the bits of the current *sig\_map* entry. This both marks the coefficient as moved to the subordinate list, and prevents it from producing extra Zerotree Root and/or Isolated Zero symbols when visited on successive bitplanes. Effectively move the quantized coefficient to the subordinate list by setting the next pointer on *sub\_list* to the address of the quantized coefficient entry.
  - (d) If  $v=(0,0,0,0)$  then code a Zerotree Root symbol. All children (grandchildren, etc.) of this coefficient are either already on the subordinate list or are not significant at the threshold corresponding to the current bitplane. No bits will be spent to encode them during the rest of the dominant pass.
  - (e) If  $v=(0,1,0,0)$  then code an Isolated Zero symbol. This indicates that the current coefficient is insignificant, but has a significant descendent.
  - (f) If a symbol was required above, code it using an arithmetic coder [8] of alphabet size 3 or 4 symbols according to the number of possible symbols (e.g. bottom-level coefficients have no children and therefore can’t be Isolated Zeroes). Count the number of arithmetic coded symbols for later inclusion in the header.
8. Shift *slice\_mask* right by 1 bit, decrement *slice\_num*.
9. Subordinate pass: Refine each coefficients on the subordinate list by one bit. The arithmetic coder is used with a two-symbol alphabet for 0 and 1. The pointers in *sub\_list* are used to access the coefficients in the order that they were found significant, which is roughly in reverse order of their magnitude.
10. Go to step 7. If at any time during steps 7-9 the target bit rate is reached, go on to step 11.
11. Write a header (number of compressed bits, number of arithmetic-coded symbols, *num\_slices*, and mean) followed by the compressed data.
12. Decoding consists of the inverse of these steps. At each level a bitplane of *sig\_map* is built up, allowing the same efficient determination of zerotree membership by checking the *sig\_map* entries of the parents. Prior to inversion, each coefficient on the subordinate list is inverse-quantized by adding  $slice\_mask/2$  and multiplying the result by  $\pm T_{min}$  (depending on the sign of the coefficient). This places the reconstructed value in the center of the quantization bin.

57-82  
013178  
268902  
108

# Experimental Studies on a Compact Storage Scheme for Wavelet-Based Multiresolution Subregion Retrieval\*

A. Poulakidas, A. Srinivasan, O. Egecioglu, O. Ibarra, T. Yang  
Department of Computer Science  
University of California  
Santa Barbara, CA 93106

## Abstract

Wavelet transforms, when combined with quantization and a suitable encoding, can be used to compress images effectively. In order to use them for image library systems, a compact storage scheme for quantized coefficient wavelet data must be developed with a support for fast subregion retrieval. We have designed such a scheme and in this paper we provide experimental studies to demonstrate that it achieves good image compression ratios, while providing a natural indexing mechanism that facilitates fast retrieval of portions of the image at various resolutions.

## 1 Introduction

Wavelet based algorithms have become popular for image compression. These typically involve three steps: (i) the application of a wavelet transform to an image to create coefficient matrices (ii) the quantization of these floating point matrices to create integer matrices, and (iii) encoding the quantized matrices to obtain the compressed image. The original image can be reconstructed by reversing the steps mentioned above. Some loss in the quality of the image results from the fact that the second step cannot be inverted exactly. The three steps are discussed briefly below.

A 2-dimensional wavelet transform maps a given image to another that resembles the original, with half the resolution. Three coefficient (or detail) matrices are also created. This process is referred to as *decomposition*. The detail matrices can be combined with the resembling image, through the application of a 2-dimensional inverse wavelet transform, to reproduce the original image. This inverse transform is also referred to as *reconstruction*. This process is illustrated in Figure 1. The decomposition described above can be recursively applied to the resembling image to produce smaller images at lower resolutions, as shown in Figure 2. This recursive process is carried out to a prescribed maximum depth, which depends on the application. The smallest resulting image obtained is called the *thumbnail*.

---

\*Supported by NSF/NASA/ARPA Grant No. IRI94-11330

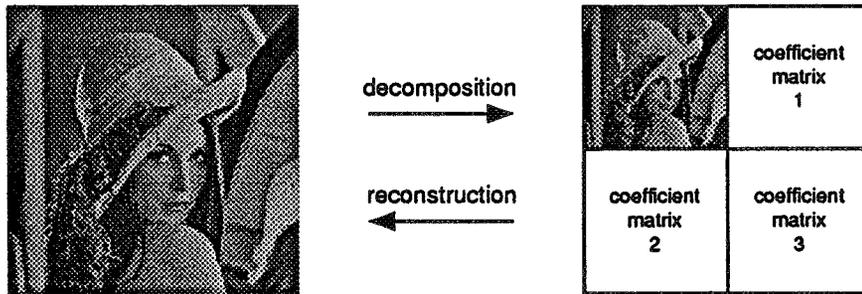


Figure 1: Image decomposition and reconstruction.

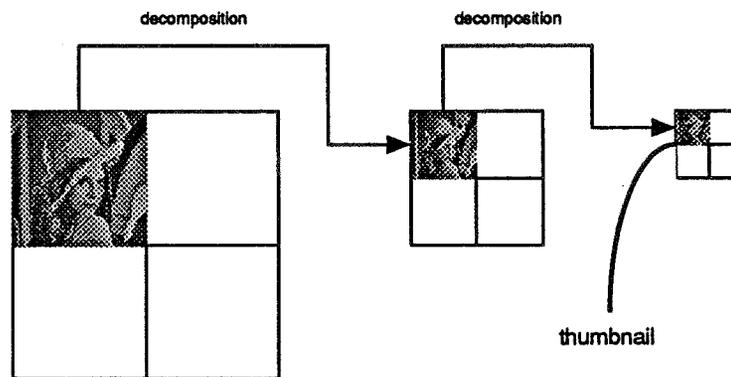


Figure 2: Multi-level decomposition of an image.

The coefficient matrices obtained as a result of the previous step consist of, in general, floating point numbers. These floating point numbers are then quantized to integers in a smaller range. This step enables efficient compression of the coefficient matrices.

In the final step, further compression of the coefficient matrices is achieved by encoding the quantized matrices. The encoding schemes usually utilize the fact that the histogram of the quantized coefficients typically has a narrow distribution.

In the Alexandria Digital Library (ADL) project at UC Santa Barbara, wavelets are used to support the hierarchical image storage and browsing: A typical ADL user will issue a query to access a set of thumbnails. Then the user will browse through the thumbnails, select the ones that he/she wants, and enlarge them to some desired resolution (or, more often, access just a subregion of a thumbnail of interest), using the wavelet reconstruction. We briefly discuss below the utility of a wavelet based compression scheme in the ADL environment.

1. The image collection includes a large number of satellite and aerophoto images with an average size of 30MB. The storage scheme must be able to achieve a good *compression* ratio in order to store these images efficiently. Apart from influencing the storage requirement, effective compression also reduces the time required for reconstructing the image since the time taken for reading from the disk is reduced if the image is stored in less space. Wavelet based schemes have proven effective in yielding substantial compression.
2. The size of the images are usually too large to display in their entirety at the highest

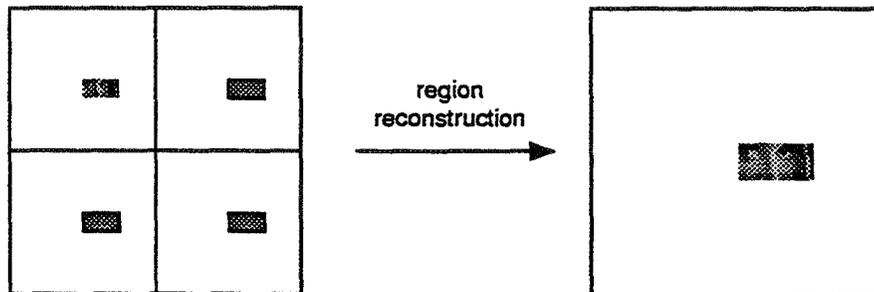


Figure 3: Region reconstruction.

possible resolution. Therefore the user typically either views the whole image at a lower resolution, or views a subregion of the image at a possibly high resolution, as shown in Figure 3. Retrieving a portion of an image from compressed data requires a sophisticated indexing scheme to make access fast. The challenge is to design a storage scheme with a fast indexing mechanism such that the compression ratio is not compromised while trying to reduce the time to access a subregion.

There is extensive literature available on selecting the appropriate wavelet transform (e.g., [1, 7]), and for the quantization that will provide compression by minimizing the bit allocation, while preserving much of the image quality (e.g., [9, 3, 4, 2]). To further reduce storage requirements, often an extra phase of compression is applied to the quantized values, typically using entropy coding, hybrid schemes using run-length/Huffman coding [1], or other methods [8, 6]. In this paper, we describe a scheme that provides an alternative technique for this phase, which results in similar compression performance but with superior subregion retrieval speed.

The storage scheme we implemented works on any naturally quantized coefficient matrix as input. Furthermore, care has been taken in the implementation of the storage scheme to allow flexibility on the data placement. In the experiments reported in this paper, all data are stored on a single disk but the algorithm can be easily extended to multiple disks or on hierarchical storage media.

We describe the storage scheme in section 2 and report on the experimental results in section 3.

## 2 The storage scheme

The effective compression ratio of our scheme depends on the histograms of coefficient matrices. Theoretically speaking, any distribution of histograms is possible; however, in practice they are symmetrical peaks centered at zero [7], similar to the one shown in Figure 4. Using an appropriate quantization, the resulting matrices will contain many zeros. Furthermore, because of the locality observed in most images and the fact that coefficient matrices consist of difference values among weighted neighboring pixels, it is expected that very often many of the zeros in the quantized matrices will be adjacent and grouped to form 2-dimensional regions consisting of zeros.

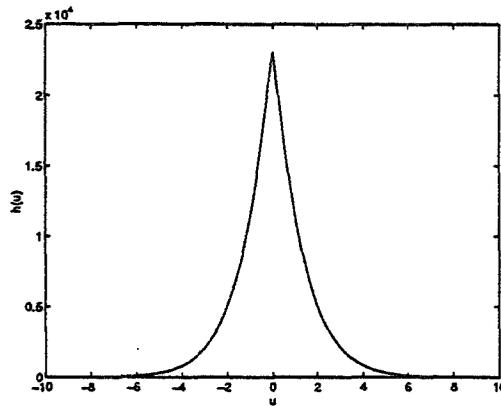


Figure 4: A typical histogram of a coefficient matrix produced by wavelet decomposition.

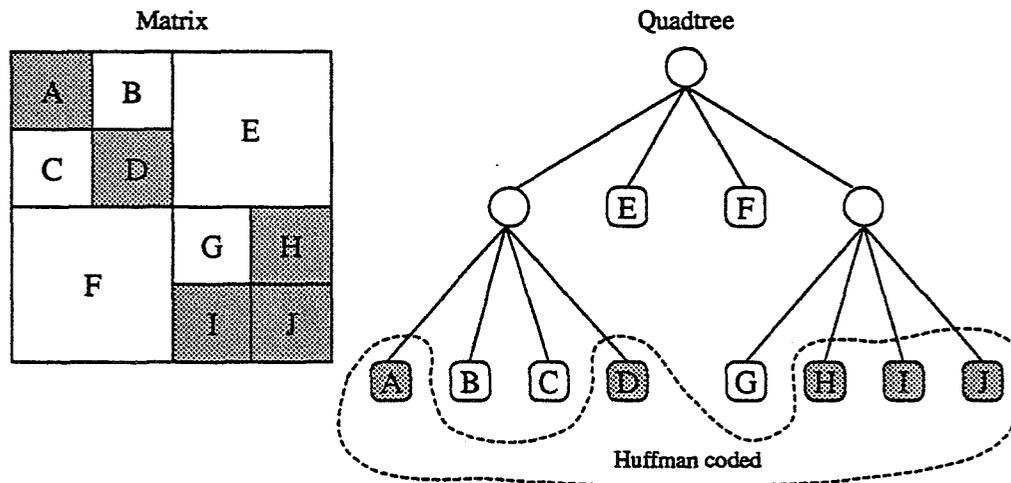


Figure 5: The combination of quadtrees and Huffman coding in our storage scheme.

A classical solution is to use *quadtrees* to compress coefficient matrices. Quadtrees are useful in eliminating unnecessary storage of 2-dimensional blocks of zeros. Furthermore, they provide a natural indexing mechanism to easily access image subregions as depicted in Figure 5. However, quadtrees built on image data usually are large, overhead for accessing and composing subregions is high and also compression ratio is not as effective as other methods such as Huffman coding.

The peaked histograms of coefficient matrices suggest that the non-zero data in quantized matrices will compress well if *Huffman coding* is used<sup>1</sup>. For this reason we have used a hybrid method that takes advantages of both quadtree and Huffman coding: 1) Use the quadtree method to partition an image and 2) compress all the non-zero leaf areas of the quadtree using Huffman coding (see Figure 5).

The peaks of the histograms of coefficient matrices become shorter and wider as the level

<sup>1</sup>Actually, this is not necessarily true if the quantized scheme does not use a constant bin size for each coefficient matrix. We have not experimented with quantization schemes that use many bin sizes for each coefficient matrix.

of the decomposition increases. However, this does not affect the performance of our storage scheme, since the size of coefficient matrices becomes smaller as the level of decomposition increases. Note that just the matrices at the first level of decomposition add up to more than 75% of the total space size of coefficient matrices.

## 2.1 Remarks on the storage scheme

- We needed to eliminate internal fragmentation at the byte level, since often the compressed size of the non-zero leaf areas is not a multiple of 8 bits.
- A compact representation of a quadtree is used. This is important since in many cases the size of the quadtree takes more than 50% of the total storage required.
- An adaptive control mechanism is provided that automatically prunes a quadtree to improve the compression, while making sure that the desired trade-off with the region retrieval speed is not violated. There is a trade-off in those coefficient matrices that do not compress well under quadtrees, so they require a minimal quadtree in order to compress well. But if the quadtree is too small, insufficient indexing may lead to a slow region retrieval. The desired trade-off is determined by a parameter that specifies the maximum non-zero area that is allowed to be a quadtree leaf.
- Buffering is used to reduce the number of disk reads when retrieving many small non-zero areas.

## 3 Experiments

We demonstrate the performance of our scheme through experimental results. In particular, we test our scheme against the following three criteria: compression ratio, reconstruction quality, and region retrieval speed.

The wavelet filters used in our experiments were designed to be very fast in reconstruction, since they are just 3-tap and only require integer operations<sup>2</sup>. But they violate the QMF criteria, and provide a slight degradation in reconstruction quality [1].

We tested our scheme with a very simple quantization scheme: the bin sizes are the same for all coefficient matrices at any given level of decomposition, and they have half the width of the bin sizes at the previous level of decomposition. The bin size at the first level of decomposition is user-specified. A more sophisticated quantization should provide better reconstruction quality for a given compression ratio.

We compare the performance of our scheme with that of a typical storage scheme that uses a combined run-length/Huffman entropy coder. Of course, for the comparisons to be legitimate, this scheme is using the same wavelets and quantization scheme as our storage scheme.

We have experimented extensively with several representative images from the ADL (in particular, aerophoto and satellite images) and texture image collections. For the time being,

---

<sup>2</sup>The reconstruction filters are  $[1 \ 2 \ 1]$  and  $[-1 \ 2 \ -1]$ .

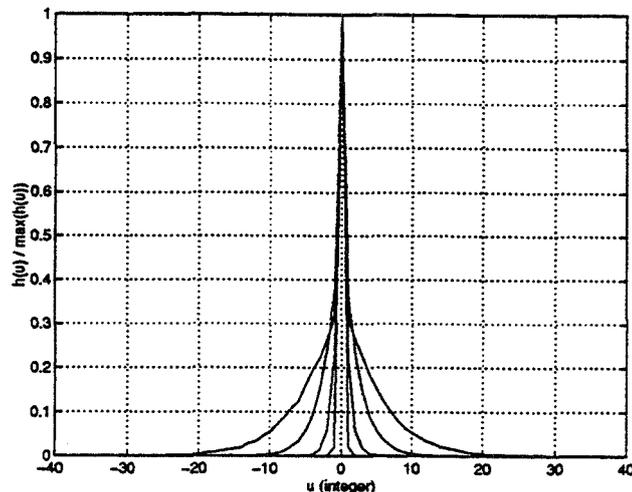


Figure 6: Typical observed histograms of the quantized coefficients, for varying levels of decomposition. The wider the histogram the larger the level of decomposition. The histograms have been normalized so that their height is 1.

we considered images that allocate one byte per pixel, and have sizes varying from 65KB to 4MB.

### 3.1 Compression ratios

The bin size used during quantization were set such that the distortion in the reconstructed regions was not visually disturbing. In Figure 6 the typical peaks of the observed quantized coefficient matrices are shown. The narrower the peak the better it compresses. Note that although the peaks at deeper levels of decomposition become wider, the size of the corresponding matrices becomes insignificant —the first decomposition level coefficients are more than 75% of the total.

The typical compression we have obtained ranged from 75% to 90%. The worst compression experienced was 72% for some texture images. These images have been shown hard to compress by other research groups and we expect around 70% to be very close to the worst case compression. The compression ratios achieved when using run-length/Huffman were similar. Furthermore, as one might expect, the compression ratio seems to improve as we consider larger images.

### 3.2 Reconstruction Quality

We measure the reconstruction quality using the peak-signal-to-noise ratio (PSNR) criterion. The PSNR for an  $N \times M$  8-bit image  $I$  and its reconstructed equivalent  $\hat{I}$  is given in dB by

$$\text{PSNR} = 20 \log_{10} \left( \frac{255}{\sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M [I(i, j) - \hat{I}(i, j)]^2}} \right)$$

The higher the PSNR the higher is the reconstruction quality. The compression ratios reported in the previous subsection were achieved with PSNR greater than 32 dB.

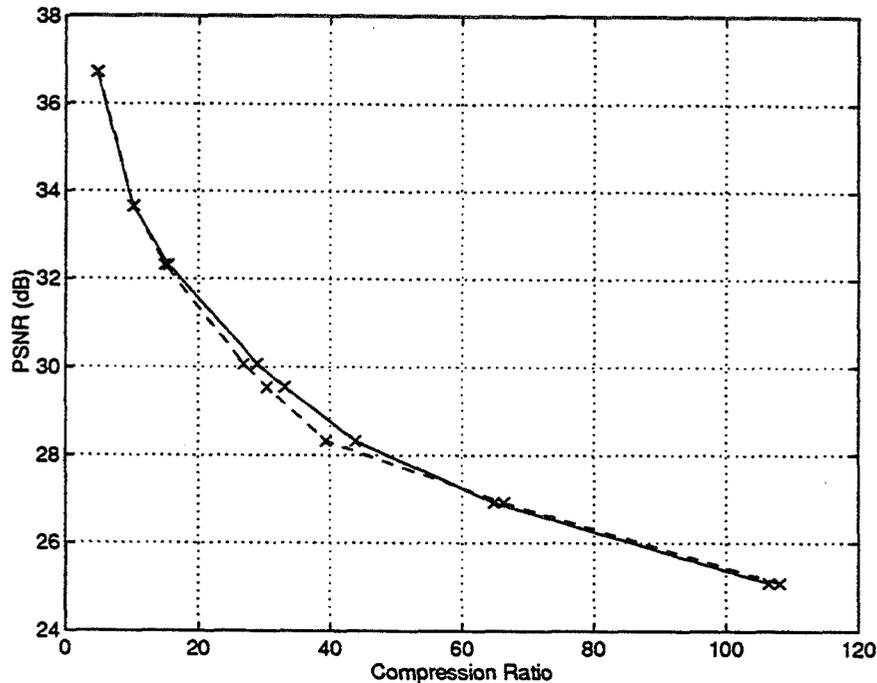


Figure 7: A comparison of the image reconstruction quality versus the compression ratio. The tests were performed on the  $512 \times 512$ , 8-bpp Lena image. The solid line refers to our method and the dashed to the run-length/Huffman based method.

The effect of changing the bin size in compression and reconstruction quality is shown in Figure 7, compared with run-length/Huffman. [5] provides results for other methods.

### 3.3 Region Retrieval Speed

We run our experiments on a SUN SPARC 5 (75Mhz) workstation with its own local SCSI-2 disk.

To test our region retrieval time we compared with the same run-length/Huffman coding scheme. Its disadvantage lies in that for each region retrieval from a matrix, this scheme requires the whole matrix to be loaded and uncompressed. However, while data loaded are more than that in our scheme, the number of disk seeks required is minimal. Once data is loaded, only those portions needed to reconstruct the desired region are actually used, and the computation is the same for both methods.

In Figures 8 and 9 the two schemes are compared. Figure 8 shows the running times when fully reconstructing  $128 \times 128$ ,  $256 \times 256$ ,  $512 \times 512$  and  $1024 \times 1024$  regions. Figure 9 shows the running times for the reconstruction of a region of typical size ( $512 \times 512$ ) for 1 to 5 levels.

## 4 Current Work

We have been extending our scheme to a client/server environment, where coefficient data are progressively delivered by the server to the client, where the reconstruction is performed.

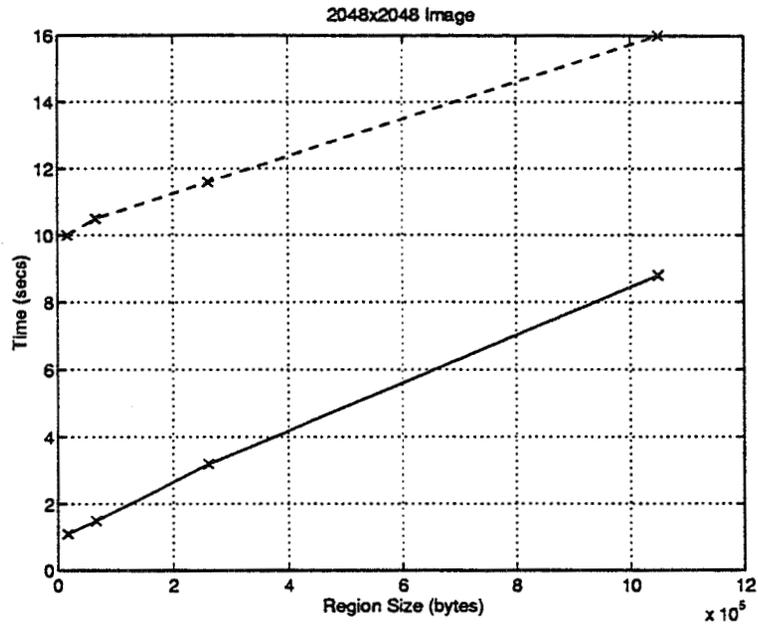


Figure 8: Retrieval+Reconstruction times for various region sizes. The regions were re-constructed for 5 levels. The solid line refers to our method and the dashed to the run-length/Huffman based method.

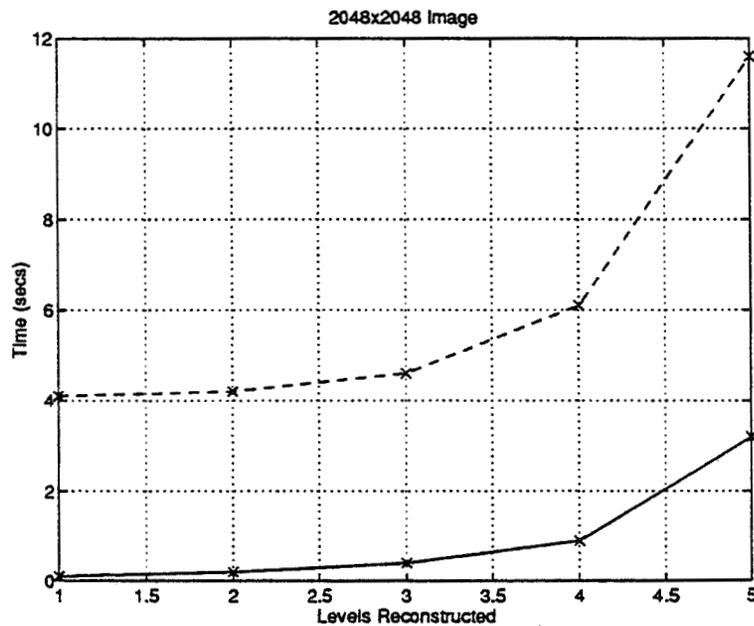


Figure 9: Retrieval+Reconstruction times for a 512x512 region for various levels of reconstruction. The solid line refers to our method and the dashed to the run-length/Huffman based method.

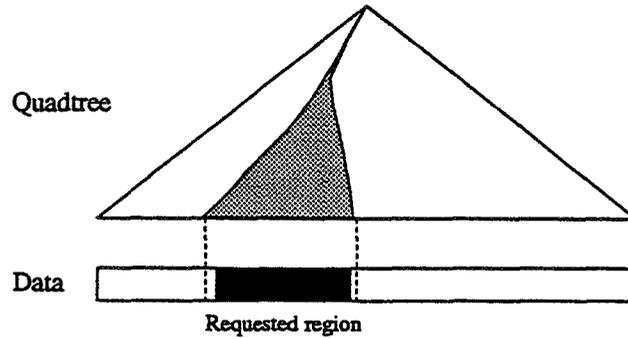


Figure 10: One need only have access to the subtree of the quadtree that “covers” the requested region, in order to be able to reconstruct it. This is demonstrated here for an 1-D region. The 2-D region is a direct generalization.

Our scheme adapts very naturally to this environment, since the server does not have to uncompress the data before sending them. As indicated in Figure 10, only the part of the quadtree that “covers” the requested region and the Huffman-compressed non-zero areas at the region need to be sent to the client. Since the data are sent compressed to the client, the server work is less and network bandwidth is saved.

## References

- [1] E. H. Adelson and E. Simoncelli, “Subband Image Coding with Three-tap Pyramids,” *Picture Coding Symposium*, 1990, Cambridge, MA.
- [2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image Coding Using the Wavelet Transform,” *IEEE Transactions on Image Processing*, 1(2), April 1992.
- [3] M. Barlaud, P. Sole, T. Gaidon, M. Antonini, and P. Mathieu, “Pyramidal Lattice Vector Quantization for Multiscale Coding,” *IEEE Transactions on Image Processing*, 3(4):367–381, July 1994.
- [4] R. A. DeVore, B. Jawerth, and B. J. Lucier, “Image Compression Through Wavelet Transform Coding,” *IEEE Transactions on Information Theory*, 38(2):719–746, March 1992.
- [5] M. L. Hilton, B. D. Jawerth, A. Sengupta, “Compressing Still and Moving Images with Wavelets,” *Multimedia Systems*, 2(3).
- [6] A. S. Lewis and G. Knowles, “Image Compression Using the 2-d Wavelet Transform,” *IEEE Transactions on Image Processing*, 1(2):244–250, April 1992.
- [7] S. G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

- [8] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.
- [9] N. Strobel, S. K. Mitra, and B. S. Manjunath, "An Approach to Efficient Storage, Retrieval, and Browsing of Large Scale Image Databases," *Photonics East, Proceedings of the SPIE — The International Society for Optical Engineering*, to appear.

98-82  
013179  
068903  
10P.

# JPEG Lossless Image Compression and High Compressibility

Xiaolin Wu	Nasir Memon	Glen Langdon	Don Speck
Univ. of Western Ontario	Northern Illinois Univ.	Theodore Goodman	Jim Spring
London, Ontario, Canada	De Kalb, IL	Univ. of California	Santa Cruz, CA

## Abstract

The JPEG international standards effort, under the auspices of ISO/IEC Joint Technical Committee 1, is currently revisiting the lossless image compression standard under a new work item. Two thrusts have emerged: high compressibility and low complexity. Several fine algorithms have been proposed for both thrusts, and a baseline has been determined. The current task is algorithm convergence for scalability to high compressibility. We present experimental results and discuss key ideas behind the results, and point out common points of the approaches for the prediction, error handling, local contexts and coding.

## 1 Introduction

A current agenda of the JPEG international standards activity is a new work item [1], involving the evaluation of new lossless and near-lossless image compression algorithms for possible standardization. Prior to the new work item, the state of the JPEG algorithm set is available in book form [2]. After three international meetings (July 1995, November 1995, and February 1996), the JPEG low-complexity group has converged on a baseline algorithm for the new lossless image compression algorithm. Here we focus on the high-compressibility extensions that would employ more complex models coupled with an arithmetic code. Modeling includes determining a context, and then the probability-related parameters of the context's distribution that belongs to the next event. The coding part encodes statistical events of a designated probability distribution by operations that create the code string. Currently the work item calls for a near-lossless capability, where the peak loss is bounded by an absolute value ( $\pm 1$ , and  $\pm 3$ ).

## 2 JPEG Lossless Image Compression Contributions

The Call for Contributions resulted in algorithms from eight organizations. The organizations and their algorithms (in parentheses) are: Ricoh (Crew), Mitsubishi (Clara), UCSC (Jslug and ALCM), NEC (LTC), Canon (APEC), WMS (CALIC), Hewlett-Packard (LOCO), and Kodak (DARC). The WMS contribution (CALIC) is from three individuals affiliated with three universities (X. Wu of the University of Western Ontario, N. Memon of Northern Illinois University, and K. Sayood of the University of Nebraska). The algorithm LOCO

(LOW COmplexity) was the contender for the low complexity end, and CALIC the contender for the high compression end.

A goal is to make the complexity *scalable*, through a choice of parameters and options. We suggest addressing certain image classes directly through special parameter sets. For example, medical images fall into different modalities; and the following have different properties: computer-generated artwork, photographs, and composite images.

For the second JPEG lossless WG1 meeting in Dallas, Texas in November 1995, ideas from other algorithms presented in Epernay were incorporated in a move toward convergence. CALIC performed best in the lossless category again, with ALCM in third place. Good ideas from the original proposals were incorporated into the other proposals. Further convergence efforts took place at the third meeting of the JPEG lossless work item in Geneva in February 1996. The new lossless baseline (low-complexity) algorithm was defined there, while the high-compressibility algorithms also continue to improve and further explore complex ideas. Intuitively it seems easier to describe and investigate low-complexity algorithms than to explore the solution space of high-complexity algorithms.

This present work focuses on the authors' interests in higher compression, and thus includes ideas taken from the paper submissions of CALIC, ALCM, and Jslug to the November 1995 WG1 meeting, including material from our meeting-agenda reports.

### 3 Brief Overview of CALIC, ALCM, and Jslug

#### 3.1 CALIC

The CALIC proposal stretched the technology in several directions. The proposal that favors the compression performance the most under the two conflicting considerations of high coding efficiency and low computational complexity was contributed by X. Wu, N. Memon, and K. Sayood. This lossless image coding scheme, which became later known as CALIC (context-based, adaptive lossless image codec), puts heavy emphasis on image data modeling.

Instead of using a static (fixed) predictor and a large collection of statistical contexts in which conditional probabilities are estimated for entropy coding, CALIC provides a large collection of predictor contexts and relatively few conditioning contexts for entropy coding. The reason for this is to alleviate the sparse context problem. Local gradient information provides the basis for the prediction, hence the name Gradient-Adjusted Predictor (GAP). *Local differences* in the causal template provide the gradients. The GAP predictor is further made adaptive to the recent source history via context-based error modeling and error feedback.

The conditioning context for the error distribution has components of local texture and error strength. The error distribution has an innovative treatment with a context-sensitive error-feedback technique. Also, twin error distributions that are asymmetric about zero but have means of opposite sign undergo a *sign-flipping* operation to combine them into a single distribution. This cuts down on the number of conditional probabilities used in entropy coding by half with a negligible loss in coding efficiency. The error values are encoded with an adaptive m-ary arithmetic code from the CACM++ package that was made publicly available by Moffat, Neal, Witten, Carpinelli and Salamonsen.

For treating compound images, the CALIC system has two mutually exclusive modes: *binary mode* and *continuous-tone mode*. The binary mode is for situations in which the current locality of the input image has no more than two distinct intensity values. The system enters one of the two modes depending on the context of the current pixel. In the binary mode, a context-based adaptive ternary arithmetic coder is used to code three symbols: the two observed pixel values in a fixed neighborhood, and an escape symbol in case the current pixel is not one of the two said values. The mode selection is automatic, and completely transparent to the user. No side information about mode switching is required.

For the November 1995 WG1 meeting, the binary mode decision was improved, a simpler energy estimate was employed, and the GAP predictor was simplified. Also, a scalable texture pattern was devised. Several CALIC ideas presented at Epernay were tried by others and presented at the Dallas meeting.

### 3.2 ALCM

The ALCM proposal uses an adaptive predictor based on 5 neighborhood pixel values (a 5-point adaptive predictor). Prediction errors are encoded in one of several statistical contexts depending on local activity. The activity, roughly speaking, measures the presence or strength of an edge. The treatment of the prediction error is influenced by the prediction error model of Rice and Plaunt [7]. The idea is to interleave negative prediction errors with positive prediction errors. The effect is to convert a double-sided exponential distribution (such as the Laplace) to a single-sided exponential distribution resembling the geometric distribution. In the data model of [7], the interleaved prediction error position number is a run length, and so is run-length encoded with a Golomb code [10] of an appropriate parameter. ALCM is not the only WG1 proposal that employs the prediction error model of [7]. However, ALCM achieves further compression by using an adaptive binary arithmetic code on the bitstring generated by the Golomb code: the more efficient the Golomb code, the fewer the required binary decisions. For the Dallas WG1 meeting results, a 6-point neighborhood adaptive predictor replaced the 5-point predictor of the July 1995 WG1 results.

### 3.3 Jslug

The Jslug algorithm submitted to the July 1995 WG1 meeting, except for the ALCM predictor, was essentially the model of [8], which was tuned to the earlier JPEG test image set. The algorithm used bucketed prediction error contexts of the W, N, and NE pixel locations. The W and N each used 7 buckets and the NE location used 3 buckets. The adaptive binary arithmetic coder was programmed by Speck, used scaled-counts of 0s and 1s, and counts were halved when the total count reached 128. Subsequent work discovered the number of contexts could be greatly increased. Also, a simple gradient detector for the Horizontal and Vertical edges was devised, with good improvements.

## 4 Model Variations

### 4.1 Predictors

#### ALCM

Among the new predictors, ALCM proposed a single-context adaptive predictor [3] also used by Jslug. Each of the 5 or 6 pixel locations has a weight assigned, and the weights add up to a convenient power-of-two, 256. The prediction is formed by first forming the products of each sample value by the weight of its current location relative to the pixel location being predicted. The products are summed, then shifted right by the power-of-two representing the sum of the weights.

For each prediction, if the prediction is correct, no weights are changed. If the prediction is too high, then the weight corresponding to the minimum in the prediction window has its weight increased by one. Similarly, the weight relative the maximum in the prediction window has its weight decreased by one. Since the minimum (or maximum) value may be shared by more than one location, a priority scheme on the locations breaks the ties.

#### CALIC

For the continuous-tone mode of CALIC, a context-sensitive error modeler using feedback improves the prediction. CALIC also does context-merging of pairs of distributions that are asymmetric about zero but with positive and negative means with a similar degree of bias by a “sign-flipping” technique.

The most important contribution of CALIC is the idea of using a large number of modeling contexts to estimate some parameters of prediction errors such as mean, median, or variance. These estimated parameters are then used to improve a so-called gradient-adjusted predictor (GAP) via a closed error feedback loop, and to shape the error probability mass functions so that the underlying entropies are reduced.

The gradient is determined using the technique of absolute *local differences*, without regard to sign, which drives an if-then clause that chooses the predictor function. This, plus the previously mentioned context-dependent constants and error feedback adapts the predictor to the local neighborhood. In contrast, the UCSC predictor [3] dynamically adjusts its constants to local changes, but uses one set of constants.

In CALIC, local intensity gradients are estimated in the horizontal, vertical, and two diagonal directions. These estimates detect the magnitudes and orientations of edges at or near the current pixel, if the edges exist. Based on the gradient estimates, a gradient-adjusted prediction (GAP)  $\hat{I}$  of  $I$  is made. The GAP predictor  $\hat{I}$  is more robust than the existing JPEG (and many other) DPCM predictors, particularly at the presence of strong image edges.

Gradients alone cannot adequately characterize some of more complex relationships between the predicted pixel and its surroundings. Since the GAP predictor  $\hat{I}$  cannot completely decorrelate the neighboring pixels, there still exist some structures of the prediction error associated with the context of the predicted pixel. Therefore, context modeling of prediction errors can exploit higher-order structures such as *texture patterns* in the image for further compression gains, after the lower-order structures such as smoothness are captured and exploited by the GAP predictor.

Specifically, the prediction value  $\hat{I}$  is used to threshold the preceding pixel values in a causal templates into 1 or 0 depending on whether they exceed  $\hat{I}$  or not. This binarization quantizes causal template of  $k$  pixels into  $2^k$  texture patterns. In addition, CALIC estimates the energy of the next prediction error by a linear regression of estimated gradients (local activity levels) and preceding prediction errors. The error energy estimator is quantized into 8 levels and used for both context-based error modeling and entropy coding.

Combining the binary texture patterns with quantized error energy estimate via Cartesian product yields the so-called *compound modeling contexts*. These compound contexts depict both the waveform and energy level of the intensity function. The prediction errors are then modeled under different compound contexts. Given a compound context, the conditional sample mean  $\bar{e}$  of the prediction errors under this context is fed back to  $\hat{I}$  to generate a second and improved prediction  $\check{I} = \hat{I} + \bar{e}$ . The modified predictor  $\check{I}$  is a context-based, adaptive, non-linear predictor that can correct itself by learning from its own mistakes made in the past and in the same context.

#### **Prediction Error Treatment: Error Feedback**

A consensus seems to be reached favoring use of error-feedback. Following a July 1995 meeting, other proposer studied CALIC's scheme of context-based error modeling and error feedback. LOCO employs the error feedback notion, and Canon implemented a version called Sign Prediction.

Context-based error modeling and error feedback did improve coding efficiency by 2-3% on a majority of the ISO/ETC test images and with all prediction schemes that were under the proposers' investigation. The improvement was more significant when some particular predictors mismatch the image data. Thus context-based error modeling and error feedback appears to be robust.

An exception to the observations made above is that error feedback could hurt coding efficiency when predictive coding is used to compress some of the compound images. Prediction schemes (as a form of data fitting) model the image smoothness and so break down on images or subimages lacking in smoothness. Here, the error feedback attempts to cancel the bias in the prediction value could cause the adjusted prediction value to alternately jump between a few widely-separated pixel values, increasing the entropy of the resulting error pmf. Having a binary mode could solve the problem. Here, the pixel values rather than prediction errors are coded.

#### **Data Modeling the prediction errors: Rice and Plaunt**

Another consensus is the data model for prediction errors, e.g., LOCO, ALCM, and CALIC, that use an approach developed by Rice and Plaunt [7].

## **4.2 Local Contexts**

The *local gradient* (local differences between adjacent pixels in the causal plane) is viewed as the best so far for local context. This represents another converging position. LOCO [5] employs the local gradient to advantage to condition error distributions and CALIC employs them in its gradient-adjusted predictor to "choose" the equation that determines prediction

(as in a switched predictor). Recent experiments with the Jslug model at UCSC also show that local differences provide error distributions with lower entropy.

Another context or context component is some measure of the dispersion of an error distribution. Both CALIC and ALCM used a type of “activity measure” based on either a large difference between pixels or value of an error distribution variance.

CALIC uses only a modest number (eight) of contexts in entropy coding. This ensures that good estimates of conditional probability mass functions (pmf) can be obtained via counting statistics to drive the entropy coder. This strategy of using a much larger number of contexts for error modeling than for entropy coding alleviates the problem of context dilution (insufficient number of samples in a given context for reliable estimation of error pmfs), because the rate of convergence of statistical estimates is higher for parameters of a pmf than for the pmf itself.

### 4.3 Coding

The Rice method [7] of treating prediction errors (called the Fundamental Sequence) permits the use of a Golomb code; a prefix code developed by Golomb [10] for run-length coding.

CALIC has a clear separation of the error modeling module and entropy coding module, permitting easy interfacing of the CALIC modeler with either arithmetic or Huffman coding, allowing a choice between compression ratio and compressor throughput. The arithmetic coding version of CALIC obtains 3.6 per cent higher compression than its one-pass Huffman counterpart on the ISO/IEC set of test images for the purpose of evaluating candidate proposals, while the former runs two times slower than the latter in the current C implementation of CALIC. CALIC employed an M-ary code (where M is the pixel depth) to encode the Fundamental Sequence.

#### Binary arithmetic coding and the QM-coder

Experiments performed with the Jslug algorithm, and reported at the February 1996 WG1 meeting show that the scaled-count binary arithmetic code used outperforms the QM-coder on all but a few images. The performance for the scaled-count estimator on the test set was 3.61 bits/sample, while the QM-coder version of Jslug yielded 3.73 bits/sample. Thus the QM-coder is 3.325% worse. (While 3.3% may not seem much, consider the following scenario. A 3.3% bandwidth increase on a \$100,000 annual communications expense in exchange for a capital investment of an extra \$1,000 investment for a high-compression option.)

#### Binary arithmetic coding and the CJ-coder

The Jslug and ALCM coders are binary. The coder is a direct implementation and specialization of the generalized arithmetic coder algorithm reported by Christopher Jones [6]. The Jones algorithm, a pseudo-code version of which appears in the Appendix of his paper, is highly parameterized. The so-called CJ binary coder is the specialization to the binary source alphabet and radix-255 (*not* radix-256) alphabet.

## Byte-stuffing

The byte value 255 (0xFF) signals a marker byte in the JPEG bitstream. In the current JPEG coded bit streams, the 0xFF byte naturally occurs, on average, once for every 256 bytes. When a 0xFF byte is emitted, the coder must follow it with a 0x00 byte (the stuff-byte) which the decoder must subsequently discard. The term employed for the insertion is *byte-stuffing*. The inefficiency of an extra byte every 256 bytes is approximately 0.00392, so the coding part loses about 0.4% in addition to other coding inefficiencies.

The CJ coder uses radix-255 (0xFE in hex) instead of radix-256 (0xFF). The largest number in radix-255 system is 254 (0xFE in hex), therefore byte 254 (0xFE) is the largest-valued byte that can occur: no *byte-stuffing*! Thus the byte value 0xFF (hexadecimal) never appears in the code string. The application of radix 255 code alphabets remove the need for byte-stuffing and unstuffing in the coded bitstreams.

The 8-bit code symbol alphabet codes  $\log_2 256$  or 8-bits of coded information. On the other hand, using 255 code symbols provides  $\log_2 255$  bits, or 7.99435-bits of information. The redundancy over 8 bits is 0.0056766, which yields a redundancy of 0.0007058 bits per coded bit, or an overhead of 0.07% in inefficiency.

Moreover, the Jones algorithm provides a solution to the carry-over problem that is in the public domain. If digits 254 are presented for output, and are subjected to a carry, the 254 is converted to 0 and the carry is propagated. To have something to carry into, a “guard byte” of value 253 or less is retained. Digits 254 that are subsequently output from the coder are simply counted in counter CTR until one of two things happens. If a carry-over occurs, the carry is propagated to the *guard byte*, and followed by the number of 0x00 bytes indicated by CTR. If instead, another non-254 value is emitted from the coder, then the first guard byte, followed by the number CTR of 0xFE bytes, is output. The newcomer non-254 byte becomes the new guard byte.

The Jones carry-over control method is very similar to, and a generalization of, that of Langdon’s for the binary code alphabet [9], where a blocking 0 is retained in case of a carry-over, and a count of the number of bits in the carry-over string is maintained. Similarly, if the arithmetic coding interval is delimited by values *low* and *high* as an alternative to delimiting method *low* (C-register) and *high - low* (A-register), then the technique that handles the carry-over problem in [12] applies. In other words, the method of Jones for carry-over control is the analog of the method published in [12]. Knowing one of the carry-over technique for one of these interval delimiting methods has an obvious conversion to the other method of delimiting the arithmetic coding interval.

## Binarization of Prediction Errors

The “binarization” of each have different approaches. Jslug decomposes the error values into zero or non-zero, and then decomposes the nonzero errors in the center of the distribution with a binary tree [8]. The tails employ an idea described in [11]. The position of the most significant bit of the error is encoded, which tells the decoder how many “extrabits” follow. Jslug encodes the extrabits using their position as context. A similar treatment was used in the arithmetic coding version of the current lossless JPEG algorithm [2].

A recent experiment replaced the adaptive binary coder in ALCM with an M-ary coder described in Jones [6]. The binary coder offered better performance than the M-ary coder on *every* test image.

## 5 Future Work

A mixture of interesting ideas for doing still image compression was motivated by the JTC1 new work item [1]. The algorithm convergence phase has created the need to evaluate the merits of various techniques and how well they work together. The question “*Where is the compression coming from?*” needs to be answered for each contribution that does best on certain images. We also note the trend toward simplification with little loss in compression.

The task is to build upon the baseline algorithm, and increase the compression performance in a scalable way. We have already indicated several of the features to build upon: predictive coding, Rice’s data model for the error, and use of local differences for a context component. From the beginning of the project, the dream was to have a Huffman code and arithmetic coding version, and a selection of predictors (as in the current lossless JPEG).

### 5.1 Prediction

For the *prediction component*, we are considering hybrid predictors using ideas from various contributions. Also, work needs to be done relative to knowledge available to the decoders, such as bits per pixel, and number of samples in the component of the image. For example, with large images the sparse context problem is of lesser concern but with 12-bits per sample the error distribution tails are expected to be sparse. Test image *ct* is such an example.

### 5.2 Use of Lookahead

CALIC has kept two scanlines of history data. Suppose instead we maintain one scanline of history, and one scanline of lookahead. Suppose we had an additional 16 bits of *side information* per 500 samples. The overhead would be 0.032 bits per sample. For images compressed to 3 bits per pixel, this is not a significant difference. Can representing knowledge so obtained be used to advantage? Can we identify two-toned or “few toned” regions, or large edges, etc.?

### 5.3 Looking at the error distributions

Several approaches *model the error distributions* with the Rice model, and with success. A closer look at what helps shape the distributions is an interesting project.

### 5.4 Binary arithmetic coding

A preliminary result based on ALCM favors binary arithmetic coding over M-ary. We have not experimented with the scaled-count part of the algorithm, or looked into an early attack (fast attack) to learn the statistics quickly at the beginning by using less inertia.

### 5.5 Polarity of Local Differences

A preliminary study indicated that local differences for conditioning the error distributions give an improvement. So far, we have not considered the context component of the sign of the local differences; i.e., differentiating positive edge from negative edges.

## 6 Results

In this section we show how the algorithms under consideration have evolved. See Table 1, which is divided vertically in fourths; first Jslug, then ALCM, then CALIC. The last division shows a comparison with the current arithmetically encoded QM-JPEG lossless algorithm. The test images represent a selection from those used for the new work item. Hotel and Gold are the three-component images from the first JPEG test image set. The table entries for performance are in *bits per sample*. In the multicomponent (multiband) images, the compression performance value is calculated by dividing the code string length in bits, by the total number of scalar intensity values in the multicomponent image.

The leftmost column names the test image. The three marked columns for Jslug, ALCM, and CALIC show the respective improvement of the current version, with the most recent CALIC results unknown at this writing. The third column within each division shows the  $\delta$  improvement in bits/sample. The table shows only the lossless case.

## 7 Summary and Conclusions

This paper describes work-in-progress of the authors toward the definition of features beyond the baseline for lossless and near-lossless JPEG [1]. Topics include context-based error-feedback to correct the predictor bias, use of local differences for gradients as predictor and/or conditioning contexts, texture patterns, interesting adaptive predictors, new ways to model and shape error distributions, and new techniques to alleviate sparse context problem.

We also offer a solution to the JPEG marker and carry-over problems.

## References

- [1] *Call for Contributions - Lossless Compression of Continuous-tone Still Pictures*, ISO/IEC JTC 1/SC 29/WG 1, March 1994. concerning Work Item "Next Generation Lossless Compression of Continuous-tone Still Pictures". ISO/IEC JTC1 document number N2395. Widely distributed document.
- [2] W. Pennebaker and J. Mitchell, *JPEG Still Image Compression Standard*, Van Nostrand Reinhold, 1993.
- [3] Don Speck, "Fast Robust Adaptation of Predictor Weights from Min/Max Neighborhood Pixels for Minimum Conditional Entropy", *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove CA, Oct 30 - Nov 1, 1995.
- [4] Xiaolin Wu, "An Algorithmic Study on Lossless Image Compression", sister manuscript on Wu's ideas related to CALIC; off-line optimization, context hierarchy, and pessimistic empirical evidence related to further compression gains, submitted to DCC96. Does not discuss WG1 standards effort.
- [5] Marcelo Weinberger and Gadiel Seroussi, submission to WG1 describing Low Complexity (LOCO) WG1 algorithm contribution.
- [6] Christopher B. Jones, "An Efficient Coding System for Long Source Sequences", *IEEE Trans on Information Thy.*, vol. IT-27, 280-291, May 1981.

Table 1: Compression Progress and Comparison with QM-JPEG

Image	bits/sample: original version, current version, improvement			$\Delta$ bps from QM-JPEG					
	Jslug1	Jslug3 $\Delta$	ALCM1	ALCM2 $\Delta$	CALC1	CALC2 $\Delta$	Jslug3	ALCM2	CALC2
hotel	3.87	3.77 = 0.10	3.88	3.81 = 0.06	3.71	3.68 = 0.03	0.379	0.340	<b>0.475</b>
gold	3.93	3.89 = 0.04	3.88	3.83 = 0.05	3.83	3.82 = 0.01	0.244	0.300	<b>0.313</b>
bike	3.66	3.55 = 0.11	3.63	3.58 = 0.05	3.50	3.50 = 0.00	0.371	0.335	<b>0.417</b>
woman	4.17	4.10 = 0.07	4.15	4.09 = 0.06	4.05	4.03 = 0.01	0.367	0.379	<b>0.436</b>
cafe	4.93	4.74 = 0.18	4.86	4.80 = 0.06	4.69	4.68 = 0.01	0.604	0.543	<b>0.665</b>
tools	5.17	4.97 = 0.20	5.06	4.99 = 0.07	4.95	4.92 = 0.03	0.501	0.479	<b>0.544</b>
bike3	4.58	4.25 = 0.33	4.48	4.35 = 0.13	4.23	4.21 = 0.02	0.533	0.430	<b>0.570</b>
water	1.81	1.76 = 0.05	1.84	1.73 = 0.11	1.74	1.73 = 0.00	0.112	<b>0.149</b>	0.140
cats	2.57	2.53 = 0.04	2.57	2.49 = 0.08	2.51	2.49 = 0.02	0.216	0.250	<b>0.252</b>
aerial1	8.59	8.44 = 0.15	8.36	8.34 = 0.02	8.31	8.26 = 0.05	0.553	0.652	<b>0.726</b>
aerial2	4.18	3.71 = 0.47	4.11	4.02 = 0.10	3.83	3.80 = 0.03	<b>0.428</b>	0.122	0.337
cmpnd1	1.59	1.22 = 0.37	1.52	1.38 = 0.15	1.24	1.20 = 0.04	0.277	0.124	<b>0.298</b>
cmpnd2	1.56	1.25 = 0.30	1.47	1.37 = 0.10	1.24	1.22 = 0.02	0.282	0.162	<b>0.315</b>
chart	1.37	1.19 = 0.18	1.36	1.29 = 0.08	1.28	1.28 = 0.00	<b>0.261</b>	0.167	0.175
chart_s	2.89	2.63 = 0.26	2.88	2.77 = 0.11	2.66	2.66 = 0.00	<b>0.437</b>	0.295	0.408
finger	5.48	5.55 = -0.07	5.46	5.42 = 0.04	5.47	5.39 = 0.07	0.301	0.428	<b>0.457</b>
x_ray	5.95	5.92 = 0.03	5.99	5.90 = 0.09	5.83	5.86 = -0.03	0.320	0.346	<b>0.379</b>
cr	5.21	5.17 = 0.03	5.21	5.16 = 0.05	5.17	5.13 = 0.03	0.259	0.267	<b>0.300</b>
ct	3.56	3.68 = -0.12	4.08	3.62 = 0.46	3.63	3.59 = 0.04	0.405	0.461	<b>0.498</b>
us	2.70	2.41 = 0.29	2.63	2.51 = 0.11	2.55	2.45 = 0.10	<b>0.463</b>	0.355	0.421
mri	5.91	6.00 = -0.09	5.73	5.70 = 0.03	5.73	5.67 = 0.06	0.379	0.681	<b>0.710</b>
faxb	0.79	0.55 = 0.24	0.82	0.72 = 0.10	0.75	0.71 = 0.03	<b>0.292</b>	0.123	0.129
<i>Average</i>	3.84	3.69 = 0.15	3.82	3.72 = 0.10	3.68	3.65 = 0.03	0.363	0.336	<b>0.408</b>

- [7] Robert F. Rice and James R. Plaunt, "Adaptive variable-length coding for efficient compression of spacecraft television data", *IEEE Trans. Commun.*, vol COM-19, no 6, 889-897, December 1971.
- [8] G. Langdon and C. Haidinyak, "Experiments with Lossless and Virtually Lossless Image Compression Algorithms", *Proc. SPIE*, vol 2418, 21-27, February 1995.
- [9] G. Langdon, "Method for Carry-over Control in a FIFO Arithmetic Code String", *IBM Technical Disclosure Bulletin*, vol 23, no 1, pp 310-312, June 1980.
- [10] S. W. Golomb, "Run-length encodings", *IEEE Trans. Inform. Theory*, vol. IT-12, no. 7, pp. 399-401, July 1966.
- [11] G. Langdon, "Sunset: a hardware-oriented algorithm for lossless compression of gray scale images", *Proc. SPIE, (Medical Imaging V: Image Capture, Formatting, and Display)*, vol 1444, 272-282, March 1991.
- [12] I. Witten and R. Neal and J. Cleary, "Arithmetic Coding for Data Compression", *Comm. ACM*, 520 - 540, June 1987.
- [13] G. Langdon, "Method for Carry-over Control in a FIFO Arithmetic Code String", *IBM Technical Disclosure Bulletin*, vol 23, no 1, 310-312, June 1980.

# COMPRESSION OF AERIAL ORTHO IMAGES BASED ON IMAGE DENOISING

A. Langi and W. Kinsner

Department of Electrical and Computer Engineering, Signal and Data Compression Laboratory,  
University of Manitoba, Winnipeg, Manitoba, Canada R3T-5V6,  
email: {kinsner|langi}@ee.umanitoba.ca, and  
TRLabs (Telecommunications Research Laboratories)  
10-75 Scurfield Boulevard, Winnipeg, Manitoba, Canada R3Y-1P6

## ABSTRACT

This paper deals with compression of an important class of computer images, called *aerial ortho images*, that result from geodetic transformations. The associated computations introduce numerical noise, making the images nearly incompressible losslessly because of their high entropy (e.g., 7.65 bpp). The use of classical lossy compression schemes is not desirable because their effects on the original image are unknown, e.g., it is possible that the actual image is altered, while the noise is not removed. We then propose the use of image denoising capable of preserving selected image features, and coupled with lossless image compression. This paper compares two denoising schemes applied to the compression of aerial ortho images. The first scheme is based on a Donoho's wavelet shrinking scheme that preserves image *smoothness*. The second scheme is based on preserving pixel *predictability*, leading to a variant of planar predictive coding. The paper shows the effect of various shrinking parameter values on the compression ratio and image quality. The paper also describes two different predictive coding schemes, achieving a compression ratio of 2:1 at 49.9 dB and 51.2 dB peak signal-to-noise ratio, with a difference between the original and the denoised images not exceeding one grayscale level.

## I. IMAGE NOISE AND PROBLEMS IN COMPRESSING AERIAL ORTHO IMAGES

Aerial ortho images play an increasingly important role in regional development and utility planning because they provide accurate and up-to-date land-related information of *rural* and *urban* areas [Oswa95]. Such images are large with typically 5000×5000 pixels each and 8 bits per pixel (bpp) grayscale, thus requiring 25 Mbytes of storage. This large size is due to high-precision required over a large area. For example, a 512×512 zoomed-in portion of the urban image in Fig. 1 (which is used as the *working image*) still contains many details that are clearly visible [Kins94a]. Such large images are difficult to manage (i.e., to store and/or to distribute) without image compression. If the nature of the applications is not known, lossless approach should be used to preserve all the details in the images.

Unfortunately, lossless compression of aerial ortho images is difficult because their first-order entropy is very high (typically 7.65 bits per pixel, bpp). This is due to geodetic transformations which compensate for the earth surface curvature and the camera movement. Such transformations introduce *numerical noise* (e.g., finite-register-length effects) due to extensive numerical computations. Lossless compression using arithmetic coding (AC) [Kins91], Gnu ZIP (GZIP), and lossless joint photograph expert group (LJPEG) [HuSm94] result in ratios of 1.06:1, 1:08:1, and 1.62:1, respectively, and none can achieve a required compression ratio of 2:1 or better.

Such higher compression ratios could only be achieved by *noise reduction* techniques. However, traditional noise reduction schemes based on *hard thresholding* (such as truncation of bits either in time/space domain or transform domain) are unacceptable because they affect the signal unconditionally. Other American, European, and Russian more refined approaches to signal extraction from noisy data by *semi-hard thresholding* include [Dono92a] (i) convolutional smoothers, (ii) stiffness-penalized splines, and (iii) Fourier-domain damping [WaWo75]. These methods achieve noise suppression only by broadening (or even erasing) certain features from the data.

Consequently, we should use a newer approach capable of separating the useful signal from the unwanted noise, and suppressing the noise without altering the signal. This approach, called *denoising*, can be realized by several fundamental techniques including (i) preservation of the smoothness of the signal and (ii) prediction of singularities through local dynamical behaviour. Both are *soft thresholding* techniques.

59-82  
013180  
068904  
101



Fig. 1. A 512×512 portion of an urban area image, selected as the working image.

The first denoising technique is achieved by shrinking noisy wavelet coefficients via *soft thresholding*, as introduced by Donoho [Dono92b], [DoJo92a-d], [KrPi92]. This new approach offers alternatives to the existing methods, and has theoretical properties of adaptive minimaxity that far surpasses anything previously known. For example, it attains a root-mean-square error (rms) of size  $((\log N)/N)^{2/3}$ , while the traditional linear or adaptive linear techniques attain only  $N^{-1/2}$ , for  $N$  estimation observations [Dono92a].

Other soft thresholding techniques are being developed by S. Mallat *et al.* (at Courant Institute) through singularity preservation using wavelet maxima [MaHw92], R. Coifman *et al.* (Yale) through adaptive selection of orthonormal basis from a library of bases [CoMW92] (and also [DoJo94b]), Healy *et al.* (Darmouth), and R. A. DeVore (South Carolina) and B. J. Lucier (Purdue) through smoothness preservation [DeJL92].

The second class of denoising techniques is based on *dynamics* preservation [KoSc93], which can also be seen as another kind of soft thresholding. Signals are generally modeled as observations (or realizations) of the state of signal sources [OpWY83]. For time/space varying signals (which are of interest in this paper), the sources are dynamical systems having certain degree of freedom (called *dimension*). On the other hand, the noise is modeled as a random process, produced by chance or by a dynamical system with a very-high dimension. Simple signals imply their sources are of low dimension. Furthermore, noise sources always generate realizations having complex behaviour. However, it has been shown that complex signals can also come from low dimensional systems, e.g., chaotic systems having fractional dimensions (called *fractal dimensions*) [Kins94b].

One way to estimate the source dimensions of a given signal is by using a successive measurement of *scaling* behaviour of some measures associated with the signals over some time/spatial scales. When a signal is contaminated by noise, the resulting signal shows signal scaling behaviour at coarse scales and noise scaling behaviour at finer scales. Hence, denoising schemes that preserve dynamics alter the contaminated signal to extend the signal scaling behaviour to finer scales in the successive measurement [KoSc93]. In practice, this means a denoising process must preserve (fractal) dimensions associated with the original signal.

Different techniques for extending the signal scaling behaviour to finer scales have been suggested, including techniques by Kostelich and Yorke [KoYo90], Schreiber and Grassberger [ScGr91], Farmer and Sidorowich [FaSi91], Sauer [Saue92], and Cawley and Hsu [CaHs92]. One technique that is of particular interest to us is based on signal *prediction* [KoSc93] because of its ability to extend signal scaling behaviour in a relatively simple scheme and its immediate applicability to compression. In fact, this paper shows that the well-known planar predictor [ChRa94] is a special case and one of the simplest implementations of the prediction-based denoising.

It should be noted that most of the denoising work is directed toward improving data analysis [Lin94], and not for compression. Some exceptions include Saito's work [Sait94] that extends the work of Coifman to

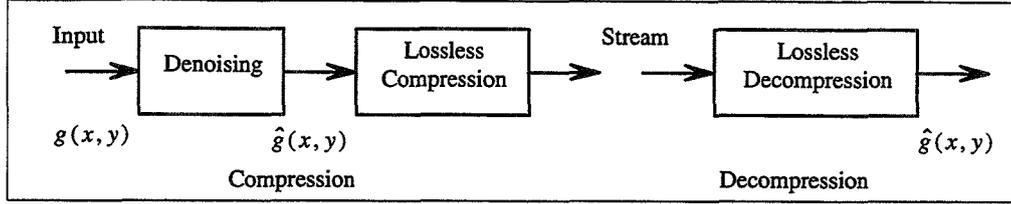


Fig. 2. A basic denoising compression scheme.

perform denoising for signal compression, and shows how denoising improves compressibility. However, Saito uses minimum description length as the soft thresholding criterion whose ability in preserving signal dynamics (i.e., extending the signal scaling behaviour) is unclear.

Although wavelets play an increasingly important role in signal denoising based on soft thresholding because of their ability to provide (i) analysis framework for signal and noise characterizations and (ii) simple algorithms to implement the denoising, we also believe that the concept of fractality and multifractality plays an even more central role in the design and evaluation of denoising schemes because of its practical ability in distinguishing complex signals from noise through the preservation of signal dynamics. The two concepts of wavelets and fractality have been shown indeed to be closely related ([Lang96], [Kins94b], [MuBA93], and [MaHw92]).

The remaining part of this paper describes the selected denoising schemes for compression of aerial ortho images. Section II formulates the denoising problem for compression. Section III describes a compression scheme using a wavelet-based denoising algorithm that preserves signal *smoothness*. Section IV describes the second scheme using a predictor that preserves signal *predictability*. Although both schemes remove data (i.e., noise), they should be considered as near-lossless or lossless signal compression schemes because of the explicit effort in removing data that are not parts of the signal. This approach has been implemented in the laboratory and then converted into a product for Linnet Geomatic International [Oswa95].

## II. COMPRESSION THROUGH DENOISING

The purpose of denoising in this work is to reduce image entropy to obtain 2:1 compression, while preserving the image details. Let  $f(x,y)$  be a pixel value of an actual aerial image with  $N$  pixels, where integers  $x$  and  $y$  represent the row and column positions of the pixel. Let  $n(x,y)$  be the value of noise added to the pixel  $f(x,y)$  by the geodetic transformations. The resulting aerial ortho image has pixels  $g(x,y)$  defined as

$$g(x,y) = f(x,y) + n(x,y) \quad (1)$$

The denoising problem becomes: Given  $g(x,y)$ , minimize  $n(x,y)$  into  $\hat{n}(x,y)$  such that

$$\hat{g}(x,y) = f(x,y) + \hat{n}(x,y) \quad (2)$$

has a reduced entropy for the required 2:1 compression. The image  $\hat{g}(x,y)$  is the denoised image. Once the entropy has been reduced, a lossless compression (such as Huffman, AC or LJPEG) can be applied to the denoised image  $\hat{g}(x,y)$  to obtain the required compression ratio, as in the scheme shown in Fig. 2. We argue that this compression approach can be considered lossless because  $n(x,y)$  is affected mostly while  $f(x,y)$  is not.

In general, denoising is a difficult problem because the characteristics of  $f(x,y)$  and  $n(x,y)$  are not known. However, aerial ortho images  $g(x,y)$  are special cases because their  $f(x,y)$  represent natural images of landscapes with many uniform (coherent) subregions. The images (such as in Fig. 1) contain many man-made, regular objects, such as roads, houses, and parks, which have very different characteristics from highly irregular numerical noise  $n(x,y)$ . Denoising methods are then based on removing parts of the signal data that have noise characteristics and do not have signal characteristics.

Successful denoising is signified by several indicators [KoSc93], [Kins94b]:

- The so-called *log-log plot* of measures such as *correlation-sum* [Kins94b] of the denoised signal  $\hat{g}(x,y)$  has a consistent slope at a wide range of the plot's  $x$ -axis. This means the denoising preserves fractal dimensions of  $f(x,y)$ . Furthermore, the *multifractal* measure (the generalized dimension  $D_q$  vs.  $q$ ) of the removed signal (i.e.,  $g(x,y) - \hat{g}(x,y)$ ) represents a flat line approaching the dimension of image support  $(x,y)$ . As explained in

[Kins94b], noise has fractal dimensions as high as the dimension of the support, e.g., 2D white noise in the case of image denoising has dimensions of two because it fills the area  $(x,y)$ .

- The  $\hat{g}(x,y)$  is more *predictable* (explained later in the following section). This means one pixel has relationships with its neighbouring pixels, and they are not independent.
- The power spectrum density of  $\hat{g}(x,y)$  is still similar to that of  $g(x,y)$ , at least in the frequency range and spectrum shape. This means there is no frequency filtering taking place, as in the traditional denoising. Furthermore, this ensures that the fractal dimensions of fractional Brownian motion parts in the image are preserved [Kins94b].
- The removed signal (i.e.,  $g(x,y) - \hat{g}(x,y)$ ) should behave like a random process whose statistical behaviour matches that of the assumed noise model. For example, the *cross-correlation* between the removed signal and  $\hat{g}(x,y)$  should be low. This is a logical consequence of removing additive noise.

### III. THE WAVELET SMOOTHING TECHNIQUE

#### A. Smooth Signals and Noise

A signal can be separated from noise based on its membership on a set of *smooth* prototype signals.

Smooth prototype signals are first defined as members of a *Besov* space  $B_p^\alpha$ , which is a space of signals having  $\alpha$  derivatives in a space  $L^p$  [DeJL92]. Two signals are said to be of similar smoothness if they belong to a Besov space with the same  $\alpha$  and  $p$ . Images can now be characterized based on their membership into Besov spaces. More specifically, let the noise be

$$n(x,y) = \sigma z(x,y) \quad (3)$$

where  $\sigma$  is a gain factor (or *noise level*) and  $z(x,y)$  is a normalized (i.e., unit variance) 2D Gaussian noise. An image  $f(x,y)$  may belong to a Besov space, while  $n(x,y)$  should not because  $z(x,y)$  does not have any derivative. This also applies to non-Gaussian noise.

#### B. A Denoising Algorithm based on Wavelet Shrinking

A *wavelet shrinkage* algorithm [Dono92b] extracts the denoised  $\hat{g}(x,y)$  from  $g(x,y)$ , using wavelet transforms and soft thresholding. The technique is based on minimization of the mean-squared-error expressed by

$$e_s^* = \left( \frac{1}{N} \sum_{x,y} |\hat{g}(x,y) - f(x,y)|^2 \right)^{1/2} \quad (4)$$

subject to a condition that the membership of  $\hat{g}(x,y)$  in the Besov space tends to be the same as that of  $f(x,y)$  with a probability of almost 1 (where  $N$  is the total number of pixels). In other words, since  $\hat{g}(x,y)$  is as smooth as  $f(x,y)$ , the noise has been removed or reduced.

The algorithm first obtains a 2D wavelet transform  $G_j(x,y)$  of the normalized input image  $g(x,y)/\sqrt{N}$ , where  $j$  is the wavelet *scale*, and  $(x,y)$  are the 2D spatial indices in the wavelet domain (see [LaKi94]). It then shrinks the value of  $G_j(x,y)$  to become  $\hat{G}_j(x,y)$  according to

$$\hat{G}_j(x,y) = \begin{cases} G_j(x,y) - T; & G_j(x,y) > T \\ 0; & -T \leq G_j(x,y) \leq T \\ G_j(x,y) + T; & G_j(x,y) < -T \end{cases} \quad (5)$$

where the reduction threshold  $T$  depends on the noise level  $\sigma$  and the size of the image according to

$$T = \sigma \frac{\sqrt{2\log(N)}}{\sqrt{N}} \quad (6)$$

and is related to Stein's Unbiased Estimate of Risk [Ste181]. Notice that, in general, the noise level  $\sigma$  is not known. A plausible statistical estimation of  $\sigma$  suggested in [DoJo94a] is

$$\sigma = \text{median}\{|G_1(x,y)|\} \quad (7)$$

Finally, an inverse wavelet transform  $\hat{G}_j(x,y)$  and denormalization are performed to obtain the desired  $\hat{g}(x,y)$ . Notice that this algorithm was not developed originally for signal compression.

### C. Compression Results

The denoising scheme has been tested using a Daubechies-8 wavelet on the working image shown in Fig. 1 using a C program running on a SUN Sparcstation 5 under UNIX operating system. The denoised image  $\hat{g}(x,y)$  has been compressed into a data stream, using a predictive-Huffman encoder, (e.g., [HuSm94]). The decompression is simply a predictive-Huffman decoder which recovers  $\hat{g}(x,y)$  from the data stream. Both the compression and decompression of the working image take less than 1 minute.

The performance of any compression scheme must address two aspects: (i) compression ratio and (ii) peak signal-to-noise ratio (PSNR). In this scheme, the compression parameter is the wavelet reduction soft threshold  $T$  (see Eq. (5)). As shown in Table 1, the threshold determines how much noise is removed from the image. It is seen that a higher threshold removes more noise, resulting in a higher compression ratio, but a lower PSNR. The threshold estimated from Eq. (7) is  $3.76 \times 10^{-3}$ . It should be noted that PSNR is not the best indication of compression performance in this case. The compressed images with PSNR as low as 35.58 dB (such as the one shown in Fig. 3) have edges as sharp as the original images, which is required for geodetic images.

Table 1. Compression performance of wavelet denoising scheme.

Soft Threshold $T$	Compression Ratio	PSNR (dB)
$2.41 \times 10^{-6}$	1.64 : 1	51.06
$3.76 \times 10^{-3}$	1.75 : 1	42.88
0.006	1.84 : 1	39.34
0.01	1.99 : 1	36.13
0.011	2.03 : 1	35.58
0.05	2.61 : 1	28.72
0.07	2.69 : 1	27.42
0.1	2.78 : 1	21.50
1	3.09 : 1	19.12



Fig. 3. A decompressed image with  $T = 0.011$ .

## IV. THE PREDICTABILITY TECHNIQUE

Kostelich and Schreiber [KoSc93] suggest an alternative approach based on preserving signal *dynamics*. The approach is powerful because it preserves *fractal* complexity of the signal, hence can accommodate natural signals having noise-like appearances but are not noise (e.g., textures). The approach extracts  $\hat{g}(x,y)$  so that the

mean square error is minimized and its fractal dimensions are similar to those of  $f(x,y)$ . Prediction is now widely used in various compression methods [WMSL96].

### A. Predictable Signals and Noise

One dynamic-preserving approach used in our work separates signals from noise based on their membership on a set of *predictable* signals. The concept starts by defining predictable signals as signals whose elements (or pixels) can be obtained or approximated using their neighbouring elements through linear or nonlinear functions called *predictors* [KoSc93]. In other words, the elements are self-consistent and have correlations among them. Intuitively, natural images containing natural objects must have such predictability characteristics because pixels of an object are related to one another, while noise pixels are independent of one another. Let the predicted pixel value at a position  $(x,y)$  be  $\hat{g}(x,y)$ . The predictable pixels  $\hat{g}(x,y)$  resemble the denoised image. For predictable signals, the predictor  $\hat{g}(x,y)$  minimizes the prediction error  $e_p^*$  as

$$e_p^* = \left( \frac{1}{N} \sum_{x,y} |\hat{g}(x,y) - g(x,y)|^2 \right)^{1/2} \quad (8)$$

White noise always results in high  $e_p^*$  because its pixels are either uncorrelated or just partly correlated (fractional noise). Therefore, there is no predictor for white noise that can result in an arbitrarily low  $e_p^*$ . The denoising scheme is then designed to translate parts of  $g(x,y)$  into  $\hat{g}(x,y)$  with a low  $e_p^*$ . This means that one must design a predictor capable of minimizing  $e_p^*$ . There can be more than one predictor that minimize  $e_p^*$ . In our case, we should use one that results in a low entropy of  $\hat{g}(x,y)$ .

#### 1. A Linear Planar Predictor

We have selected a linear planar predictor which is well known for its ability to predict natural images [ChRa94]. The predictor is defined as

$$\hat{g}(x,y) = a_1 \hat{g}(x,y-1) + a_2 \hat{g}(x-1,y) + a_3 \hat{g}(x-1,y-1) + \hat{e}(x,y) \quad (9)$$

or equivalently

$$\hat{g}(x,y) = \tilde{g}(x,y) + \hat{e}(x,y) \quad (10)$$

where

$$\tilde{g}(x,y) = a_1 \hat{g}(x,y-1) + a_2 \hat{g}(x-1,y) + a_3 \hat{g}(x-1,y-1) \quad (11)$$

The  $\tilde{g}(x,y)$  term represents the *pure* prediction of the pixel in location  $(x,y)$  by its three neighbours, with gain factors  $a_1$ ,  $a_2$ , and  $a_3$ , respectively. (We assume that the pixels outside image boundaries are zero.) The  $\hat{e}(x,y)$

term represents a bias to force the value of  $e_p^*$  to zero (perfectly predictable) by compensating the inability of the pure predictor to represent unpredictable parts contained in the image. This bias term turns out to be a critical part in characterizing predictability, as shown next.

#### 2. Characterizing Predictability Through Predictor Residues

The planar predictors allow us to characterize predictability through characterization of predictor bias  $\hat{e}(x,y)$  instead of  $\hat{g}(x,y)$  as required by Eq. (8). In general, all the gain factors  $a_i$  are fixed, implying that  $\hat{e}(x,y)$  fully characterizes  $\hat{g}(x,y)$ . Let us now have the same structure for  $g(x,y)$  as for  $\hat{g}(x,y)$  in Eq. (10) by introducing  $e(x,y)$  according to

$$g(x, y) = \tilde{g}(x, y) + e(x, y) \quad (12)$$

Clearly the denoising produces the residual (i.e., removed noise) image  $r(x, y)$  as

$$r(x, y) \equiv g(x, y) - \hat{g}(x, y) = e(x, y) - \hat{e}(x, y) \quad (13)$$

Equation (8) becomes

$$e_p^* = \left( \frac{1}{N} \sum_{x,y} |\hat{e}(x, y) - e(x, y)|^2 \right)^{1/2} = \left( \frac{1}{N} \sum_{x,y} |r(x, y)|^2 \right)^{1/2} \quad (14)$$

In other words, we can now remove the noise by processing (i.e., mapping)  $e(x, y)$  into  $\hat{e}(x, y)$  as long as the following two requirements are satisfied: (i) Eq. (14) is minimized to satisfy the predictability requirement in Eq. (8) and (ii) the residue  $r(x, y)$  behaves as noise as required by Eq. (13). If they are satisfied, the predictability of the denoised image is guaranteed by Eq. (10).

Given a projection from  $R^N$  to  $\hat{R}^N$  that satisfies both requirements, we can develop a denoising scheme as shown in Fig. 4. The scheme consists of (i) a projection, and (ii) a predictor defined by Eq. (10). Given an input image  $g(x, y)$ , the algorithm calculates the image  $e(x, y)$  recursively according to Eq. (13). At the beginning,  $\hat{g}(x, y)$  is zero, hence  $\tilde{g}(x, y)$  is also zero. The projection block obtains  $\hat{e}(x, y)$  from  $e(x, y)$  and passes the resulting  $\hat{e}(x, y)$  to the predictor. The predictor then generates the denoised image  $\hat{g}(x, y)$  according to Eq. (10).

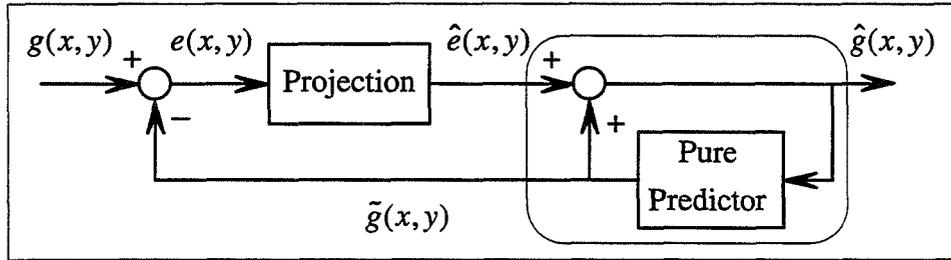


Fig. 4. The predictive denoising scheme.

### 3. Two Simplest Appropriate Selections of $\hat{R}^N$

We now focus on obtaining the mapping by defining a space  $\hat{R}^N \subset R^N$  where images  $\hat{e}(x, y) \in \hat{R}^N$  and images  $e(x, y) \in R^N$  for all possible input images. We can now define predictable images as those whose residues are *all* in  $\hat{R}^N$  and noise images are those having residues outside  $\hat{R}^N$ . It can be shown that orthogonal projection from  $R^N$  to  $\hat{R}^N$  minimizes Eq. (14) [Krey78], thus satisfying the first requirement. Furthermore, projections in a form of vector quantization (VQ) minimize Eq. (14) and produce noise residues, satisfying both requirements. Therefore, a VQ on  $e(x, y) \in R^N$  into  $\hat{e}(x, y) \in \hat{R}^N$  results in a denoising that preserves the predictability.

One disadvantage of using VQ as the projection is that the computational cost is high. We then study the use of scalar quantization (SQ) as the projection. The SQ can be seen as a simplest form of VQ, hence it satisfies both projection requirements. Although its performance (i.e., the quality vs. bit rate) is inferior to that of vector quantization, SQ can be extremely fast. This fact is especially important, because aerial ortho images are large. We then use two SQ projections (Projections 1 and 2) with their corresponding  $\hat{R}^N$  (which now can be represented by  $\hat{R}$  because the  $N$  pixels are projected individually). Both projections result in the well-known differential predictive coding modulation (DPCM) scheme, with their corresponding  $\hat{R}$  are  $\{\dots, -9, -6, -3, 0, 3, 6, 9 \dots\}$  and  $\{\dots, -6, -4, -2, 0, 2, 4, 6 \dots\}$  as in [ChRa94]. Hence, both Projections 1 and 2 guarantee that  $|r(i, j)| \leq 1.5$  and  $|r(i, j)| \leq 1$ , respectively.

## B. Compression Results

Instead of compressing  $\hat{g}(x, y)$  directly, the compression scheme is based on efficient representation of  $\hat{e}(x, y)$ , because it can be used to generate  $\hat{g}(x, y)$  using Eq. (10). Furthermore, the set  $\hat{R}$  contains fewer integers than the set of  $\hat{g}(x, y)$  values, and the pixel distribution is not as uniform as that of  $\hat{g}(x, y)$ . The compression scheme operates on images line-by-line (i.e., the procedure is repeated for each index  $x$ ). It first produces  $\hat{e}(x, y)$  using either Projection 1 and 2 and compresses it into data stream using an arithmetic encoder. The decompression recovers  $\hat{e}(x, y)$  from the data stream, and produces  $\hat{g}(x, y)$  using the predictor in Eq. (10).

The scheme has been tested using the working image in Fig. 1, and is successful in attaining the target of 2:1 compression ratio. A C language implementation on an IBM PC 486-DX33 uses CPU processing times of 7.1 s and 6.7 s for both the compression and decompression, respectively. The resulting decompressed images are indistinguishable from the original one, as shown in Fig. 5 using Projection 2. Tables 2 and 3 compare the performance of both SQ projections. Both have extremely high PSNR values for a 2:1 compression ratio. We have also generated two images compressed at approximately 2:1 compression ratios using lossy JPEG coding implemented by a popular Unix program called *xv* [Brad94]. Both images (called JPEG 1 and JPEG 2) are compressed using 96% and 97% quality settings, resulting in compression ratios of 2.08:1 and 1.87:1, respectively. As shown in Table 3, the quality of the lossy JPEG compression is inferior to that of the denoising schemes despite of having more representation bits. Furthermore, the fact that  $|r(x, y)| \leq 1$  is also shown in the table, where the measured maximum error is 1 grayscale level. As shown in Fig. 5c, the residue resembles noise, where it has three pixel values, -1, 0, and 1, shown as black, grey, and white pixels. Although Projection 2 gives compression ratio slightly below 2:1 for this test image, its compression ratio for the complete original image is 2.20:1. This is because if the images are larger, the arithmetic encoder can optimize the compression ratio better due to the more complete knowledge of residue statistics.

We have verified that the prediction-based denoising preserves fractal and multifractal characteristics. The Rényi generalized dimensions  $D_q$  have been calculated for the original, denoised, and residue images, as well as the JPEG 1 and JPEG 2 images. The results are used to calculate the spectra of singularities  $f(\alpha)$  through a

Table 2. Compression performance comparison for both Projections 1 and 2. The objective quality is very high, and the original and compressed images are visually indistinguishable.

Aspects	Projection 1	Projection 2
Compression Ratio	2.2220 : 1	1.9696 : 1
Peak SNR (dB)	49.9	51.1675
RMS	0.815721	0.704959
Error Variance	0.222643	0.249992
Error Mean	0.665401	0.496967
Error Max	1	1

Table 3. Compression performance of the new denoising schemes as compared to other traditional compression schemes. Note that 8 bpp grayscale digitized images (such as the original image) is considered to have quality equivalent to 54 dB PSNR.

Schemes	Compression Ratio	PSNR (dB)
Predictive Denoising (Projection 1)	2.22 : 1	49.9
Predictive Denoising (Projection 2)	1.97 : 1	51.17
Wavelet Smoothness	2.03 : 1	35.5
Lossy JPEG 1	2.08 : 1	44.7
Lossy JPEG 2	1.87 : 1	47
Lossless JPEG	1.62 : 1	n/a
GZIP	1.08 : 1	n/a
Arithmetic Coding	1.06 : 1	n/a



(a) original image      (b) denoised image      (c) residue image

Fig. 5. Denoising results on a portion of the original image. The original image in (a) and the denoised image in (b) are indistinguishable, and the residue image in (c) resembles noise.

Legendre transform [Kins94b]. More than 600 points of  $q$  over an interval of  $[-30, 30]$  have been used in the calculation. The values of  $D_q$  and  $f(\alpha)$  of the original and denoised images are extremely close. The average of the absolute differences of the  $D_q$  is  $5.71 \times 10^{-5}$ . The same measurement on JPEG 1 and JPEG 2 images results in averages of  $3.25 \times 10^{-3}$  and  $2.23 \times 10^{-3}$ , respectively, which is two order of magnitude worse. Figure 6 shows that the singularity spectra of the denoised and the original images coincide through out the range of  $\alpha$ , while the JPEG images deviate significantly at high values of  $\alpha$ .

## V. DISCUSSION

The denoising approaches discussed in this paper are best suited for compression of signals whose entropy is very high due to an unwanted noise. Although the original images and reconstructed images have differences, it has been shown that the differences do not belong to the expected class of signal and they have noise characteristics. Hence, it can be argued that the denoising schemes preserve the actual image losslessly. For example, the wavelet denoising approach preserves high-frequency information, so that sharp edges do not become blurred as in classical filtering methods. This is critically important, because the main feature of ortho images is in its flatness and its precision of edge position. The prediction-based denoising also preserves the edges very well. In addition, pixel deviation between the original and denoised images is limited within one

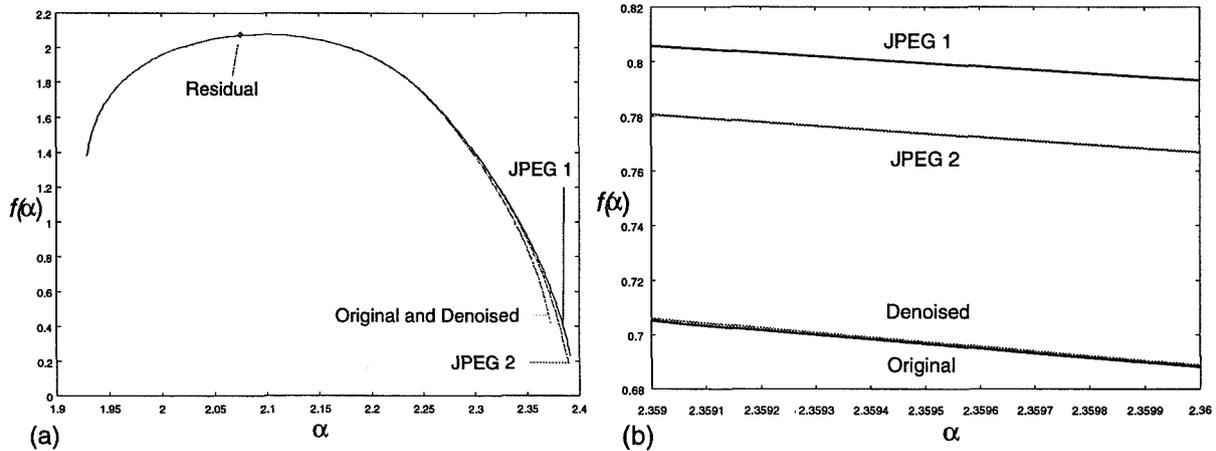


Fig. 6. (a) Complete and (b) zoomed-in plots of the spectra of singularities of the original, denoised, residual, JPEG 1, and JPEG 2 images. While the singularity spectra of the original and denoised images coincide, those of the JPEG images deviate at high singularities  $\alpha$ .

grayscale level (known as near-lossless reconstruction). This results in extremely high PSNR and superior fractal and multifractal preservation. In conclusion, wavelet and fractal denoising improves significantly the performance of high-quality image compression of aerial ortho images.

## REFERENCES

- [Brad94] J. Bradley, *XV v.3.10a* (a Unix program). Available at xv@devo.dccs.upenn.edu, 1994.
- [CaHs92] R. Cawley and G. H. Hsu, "Local-geometric-projection method for noise reduction in chaotic maps and flows," *Phys. Rev. A*, vol. 46, pp. 3057-3082, 1992.
- [ChRa94] K. Chen and T. V. Ramabadran, "Near-lossless compression of medical images through entropy-coded DPCM," *IEEE Trans. Medical Imaging*, vol. 13, no. 3, pp. 538-548, Sept. 1994.
- [CoMW92] R. R. Coifman, Y. Meyer and M. V. Wickerhauser, "Wavelet analysis and signal processing," in *Wavelet and Their Applications*, M. Ruskai (ed.), Boston: Jones and Bartlett, pp. 153-178, 1992.
- [DeJL92] R. A. DeVorge, B. Jawerth, B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Information Theory*, vol. 38, no. 2, pp. 719-746, Mar. 1992.
- [DoJo92a] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Technical Report*, Dept. of Statistics, Stanford University, 1992. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [DoJo92b] D. L. Donoho and I. M. Johnstone, "Minimax estimation via wavelet shrinkage," *Technical Report*, Dept. of Statistics, Stanford University, 1992. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [DoJo92c] D. L. Donoho and I. M. Johnstone, "New minimax theorems, thresholding, and adaptation," *Technical Report*, Dept. of Statistics, Stanford University, 1992. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [DoJo92d] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness by wavelet shrinkage," *Technical Report*, Dept. of Statistics, Stanford University, 1992. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [DoJo94a] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Technical Report*, Dept. of Statistics, Stanford University, 28 pp., 1994. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [DoJo94b] D. L. Donoho and I. M. Johnstone, "Ideal denoising in an orthonormal basis chosen from a library of bases," *Technical Report*, Dept. of Statistics, Stanford University, Sep. 1994. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [Dono92a] D. L. Donoho, "Wavelet shrinkage and W.V.D.: A 10-minute tour," *Technical Report*, Dept. of Statistics, Stanford University, 12 pp., 1992. (available in ftp://playfair.stanford.edu/pub/donoho).
- [Dono92b] D. L. Donoho, "De-noising via soft-thresholding," *Technical Report*, Dept. of Statistics, Stanford University, 37p., 1992. (available through WWW: ftp://playfair.stanford.edu/pub/donoho).
- [FaSi91] J. D. Farmer and J. Sidorowich, "Optimal shadowing and noise reduction," *Physica D*, vol. 47, pp. 373-392, 1991.
- [HuSm94] K. Huang and B. Smith, "Lossless JPEG codec," Dept. Computer Sciences, Cornell University, 1994. (available through the Internet at ftp://ftp.cs.cornell.edu/pub/multimedia/ljpeg).
- [Kins91] W. Kinsner, "Review of data compression methods, including Shannon-Fano, Huffman, Arithmetic, Storer, Lempel-Ziv-Welch, Fractal, Neural Network, and Wavelet Algorithms," *Technical Report*, DEL91-1, University of Manitoba, 157 pp, 1991.
- [Kins94a] W. Kinsner, "Compression of digital ortho images for Linnet Geomatics International," *Progress Report I*, University of Manitoba, 17 pp., Dec. 1994.
- [Kins94b] W. Kinsner, "Fractal dimensions: Morphological, entropy, spectrum, and variance classes," *Technical Report*, DEL94-4, University of Manitoba, 146 pp, 1994.
- [KoSc93] E. J. Kostelich and T. Schreiber, "Noise reduction in chaotic time-series data: A survey of common methods," *Physical Review E*, vol. 48, no. 3, pp. 1752-1763, Sep. 1993.
- [KoYo90] E. J. Kostelich and J. A. Yorke, "Noise reduction: finding the simplest dynamical system consistent with the data," *Physica D*, vol. 41, pp. 183-196, 1990.
- [Krey78] E. Kreyzig, *Introductory Functional Analysis with Applications*. New York: John Wiley & Sons, 688 pp., 1978.
- [KrPi92] G. Kerkyacharian and D. Picard, "Density estimation in Besov spaces," *Statist. and Prob. Lett.*, no. 13, pp. 15-24, 1992.
- [LaKi94] A. Langi and W. Kinsner, "Wavelet compression for image transmission through bandlimited channels," *ARRL QEX Experimenters's Exchange*, no. 151, pp. 12-21, Sep. 1994.
- [Lang96] A. Langi, "Wavelet and fractal processing of nonstationary signals," *Ph.D. Thesis*, University of Manitoba, 1996.
- [Lin94] B. Lin, "Wavelet phase filter for denoising in tomographic image reconstruction," *Ph.D. Thesis*, Illinois Institute of Technology, 1994.
- [MaHw92] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 617-643, 1992.
- [MuBa93] J. F. Muzy, E. Bachry, and A. Arneodo "Multifractal formalism for fractal signals: the structure-function approach versus the wavelet-transform modulus-maxima method," *Physical Review E*, vol. 47, no. 2, pp. 875-884, 1993.
- [OpWY83] A. V. Oppenheim, A. S. Wilsky, I. T. Young, *Signals and Systems*. Englewood: Prentice-Hall, 796 pp., 1983.
- [Oswa95] R. Oswald, "Manitoba land related information system: the information utility," *IEEE Wescanex 95 Proc.*, pp. 252-257, May 1995.
- [Sait94] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion," in *Wavelets in Geophysics*, E. Foufoula-Georgiou and P. Kumar (eds.), Academic Press. Inc., (preprint), 25p., 1994.
- [Sae92] T. Sauer, "A noise reduction method for signals from nonlinear systems," *Physica D*, no. 58, pp. 193-201, 1992.
- [ScGr91] T. Schreiber and P. Grassberger, "A simple noise-reduction method for real data," *Physics Letter A*, vol. 160, pp. 411-418, 1991.
- [Ste81] C. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, no. 9, pp. 1135-1151, 1981.
- [WaWo75] G. Wahba and S. Wold, "A completely automatic French curve," *Communic. Statist.*, vol. 4, pp. 1-17, 1975.
- [WMSL96] X. Wu, N. Memon, D. Speck, F. Langdon, T. Goodman, and J. Spring, "The JPEG lossless image compression project and convergence to high compressability," *Proc. NASA/Industry Data Compression Workshop*, Snowbird, UT, April 4, 1996.

510-82  
013181  
268905  
10P

# Wavelet Coding for Remote Sensed Data

Zia-ur Rahman<sup>1</sup>

Science and Technology Corporation, Hampton, Virginia 23666

## Abstract

A new algorithm for wavelet compression is presented. This algorithm was developed specifically for the Small Spacecraft Technology Initiative (SSTI) Clark mission which requires 3m ground resolution data to be transmitted down over a relatively slow data link. Hence the data needs to be reduced by at least an order of magnitude, yet has to maintain the high resolution as much as possible. Our algorithm is based on the analysis of the communication channel as a whole starting at the image acquisition and ending at image restoration. The quantization maps for the wavelet coded data are developed using the information theoretical metrics that arise from this particular analysis of the visual communication channel.

## 1. Introduction

The demand for high resolution imagery cannot be overemphasized in current space missions. This demand creates the need for high data compression because of the limited bandwidth of the communication channels. The problem is that achieving high compression rates and preserving high resolution is inherently contrary: high resolution generally requires more data, and high data compression results in lower resolution. Hence a careful balance must be achieved between the demands of high data compression and high resolution, and it is only possible to achieve a certain resolution given a certain bandwidth. What imposes these constraints is the visual communication channel itself—starting at image acquisition and ending at image display. An information theoretical analysis of the channel is essential in providing a rigorous, quantitative characterization of its performance, and in optimizing its throughput. An optimal communication channel would be able to transmit the best possible image at the lowest possible data rate, where “best” is defined in terms of some metric. Previously,<sup>1-5</sup> we have combined Shannon’s information theory<sup>6</sup> with Wiener’s restoration filter<sup>7</sup> and with the critical limiting factors that affect a visual communication channel to provide rigorous quantitative metrics for characterizing its design and evaluating its performance in terms of restorability. We now integrate lossy data compression into this framework to optimize data rates<sup>8</sup> and evaluate its effects on image resolution.

Figure 1 shows the visual communication channel. At the head of the communication channel is an image-gathering device which consists of a lens, a photodetector array, and an analog-to-digital (A/D) converter. This device converts the radiance field incident on it into a quantized, digital signal which is then transmitted. At the tail of the channel is a receiver which resolves the signal and provides the information to an image display device (e.g. video monitor, or a printer) which, in turn, represents this information in a form suitable for interpretation by an observer. Between the output of the A/D converter and the receiver, any number of digital image processing

<sup>1</sup>Contact address: MS 473, NASA Langley Research Center, Hampton, VA 23681. Phone: (804) 864-6556. FAX: (804) 864-7891. e-mail: z.rahman@larc.nasa.gov

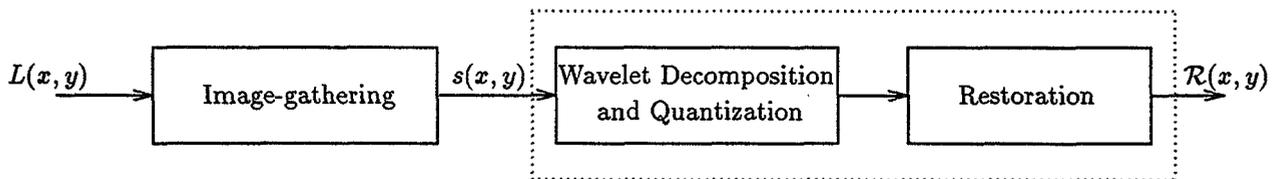


Figure 1: The visual communication channel

algorithms can be applied for image enhancement and efficient data representation and transmission. Traditional analysis of image compression and restoration algorithm is generally restricted to this stage only, neglecting the analog-to-digital conversion at image acquisition and digital-to-analog conversion at image display. This leads to an inherently incomplete model which results in restorations and compression rates which could have been better if the conversions at acquisition and display had been properly taken into account.

The imaging process injects errors in the original information that is incident on the image gathering device. The combined spatial frequency response (SFR) of the lens and the photosensor array blurs the radiance field; the photosensor array and the A/D converter introduce electronic and quantization noise respectively; and the SFR of the display spot of the image-display device blurs the resolved signal. These errors result in a reduction in the amount of information that is received by the observer when compared with the information which was present at the beginning of the imaging process. Since for a lot of applications (e.g. remote sensing) the *only* information the observer has is this received information, it is advantageous to minimize the channel effects so as to maximize information. A rigorous mathematical analysis provides the framework to evaluate the performance of the communication channel and can thus be used to informationally optimize its design in terms of both resolution (restoration) and compression. We feel that this is *essential* in designing a channel which provides the most information, and hence the highest resolution, for the least data.

## 2. Mathematics of the visual communication channel

The visual communication channel is divided into five stages: image-gathering, decomposition, quantization, synthesis, and restoration. Though we present the mathematics of each stage individually, each stage builds on the preceding stages and the restoration filters depend upon the end-to-end process.

### 2.1. Image gathering

The image gathering device converts the incident continuous radiance field  $L(x, y)$  into the discrete signal  $s(x, y)$  (Figure 2). The combined SFR of the device optics and photosensor array aperture,  $\tau_d(x, y)$ , blurs the input  $L(x, y)$ , which is sampled by the rectangular unit sampling lattice,  $\text{III}(x, y)$ , and corrupted by the additive noise due to the analog-to-digital (A/D) conversion,  $N_{a/d}(x, y)$ , and the electronics,  $N_e(x, y)$ , producing  $s(x, y)$

$$s(x, y) = [KL(x, y) * \tau_d(x, y)] \text{III} + N_e(x, y) + N_{a/d}(x, y), \quad (1a)$$

where  $K$  is the steady-state linear radiance-to-signal conversion gain. The sampling lattice is  $\text{III} = \sum_{m,n=-\infty}^{\infty} \delta(x - m, y - n)$ . Rewriting in Fourier domain,

$$\tilde{s}(v, \omega) = [K\hat{L}(v, \omega)\hat{\tau}_d(v, \omega)] * \text{III} + \tilde{N}_e(v, \omega) + \tilde{N}_{a/d}(v, \omega), \quad (1b)$$

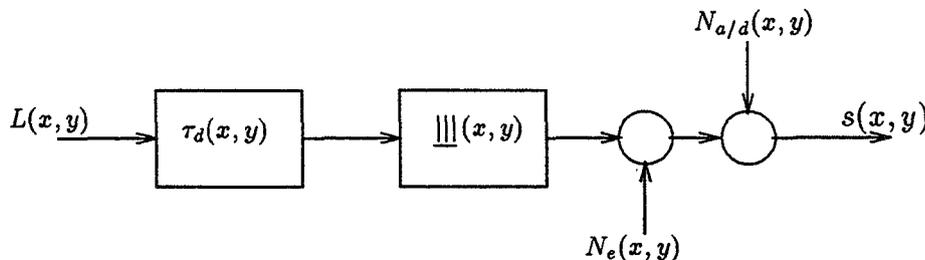


Figure 2: Image gathering: The radiance field  $L(x, y)$  is converted to signal  $s(x, y)$ .

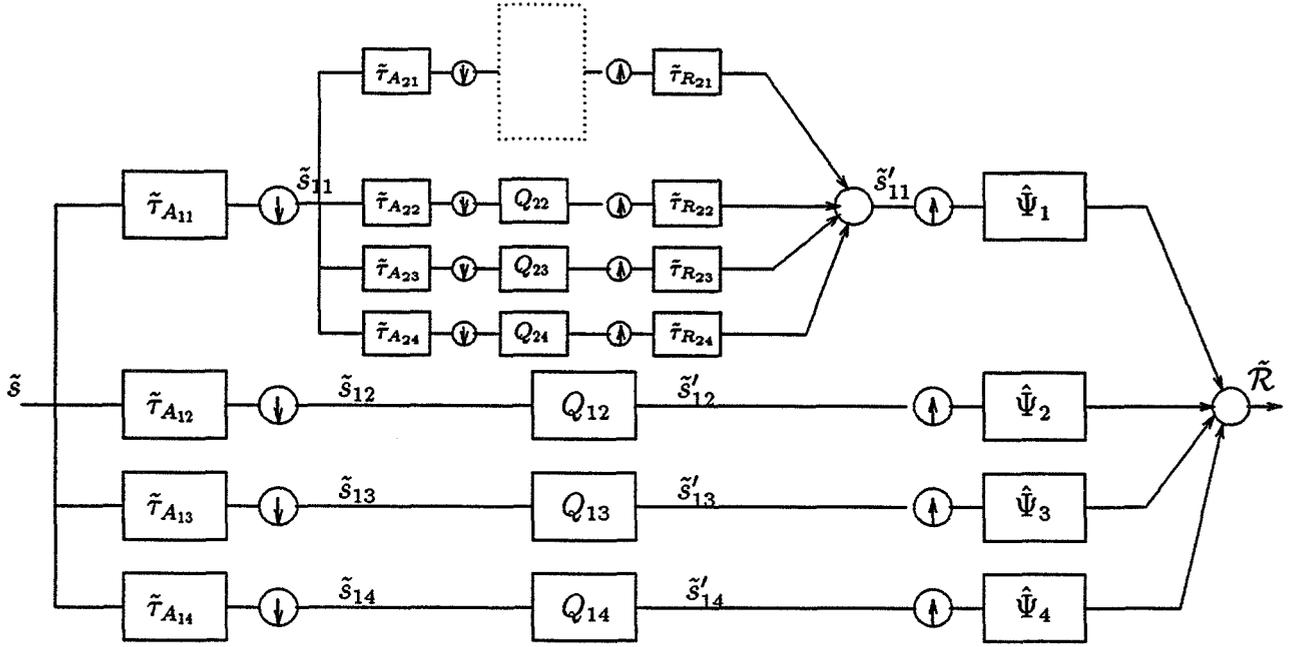


Figure 3: Wavelet decomposition, quantization, and restoration

where the notation  $\hat{p}(v, \omega)$  refers to the *continuous* Fourier transform of a function  $p(x, y)$  and  $\tilde{q}(v, \omega)$  refers to the *discrete* Fourier transform of a function  $q(x, y)$ . The Fourier transform of  $\underline{\underline{\underline{q}}}(x, y)$  is given by  $\underline{\underline{\underline{q}}} = \sum_{m, n=-\infty}^{\infty} \delta(v - m, \omega - n)$ , with the associated sampling passband  $\hat{B} = \{v, \omega : |v|, |\omega| \leq 0.5\}$ . The probability density function of the noise  $N_{a/d}(x, y)$  can be written as

$$p[N_{a/d}(x, y)] = \frac{\kappa}{s_{p_{max}}(x, y) - s_{p_{min}}(x, y)} = \frac{\kappa}{2\sigma_s}, \quad (2)$$

for  $s_{p_{max}} = k\sigma_s$  and  $s_{p_{min}} = -k\sigma_s$  which specify the range of the signal;  $\kappa$  is the number of quantization levels of the A/D converter; and  $\sigma_s^2 = \iint_{\hat{B}} \tilde{\Phi}_s(v, \omega) dv d\omega$ . The power spectral density (PSD) of the signal  $\tilde{\Phi}_s(v, \omega)$  prior to quantization is

$$\begin{aligned} \tilde{\Phi}_s(v, \omega) &= E[\tilde{s}(v, \omega)\tilde{s}^*(v, \omega)] \\ &= [K^2\hat{\Phi}_L(v, \omega)|\hat{\tau}_d(v, \omega)|^2] * \underline{\underline{\underline{q}}} + \tilde{\Phi}_{N_e}(v, \omega), \end{aligned} \quad (3)$$

where  $E[\cdot]$  is the expectation operator, and  $*$  indicates complex conjugation. Assuming that the error within each quantization interval is uncorrelated with errors within other intervals,

$$\tilde{\Phi}_{N_{a/d}}(v, \omega) = \sigma_{N_{a/d}}^2 = \frac{1}{3} \left( \frac{k\sigma_s}{\kappa} \right)^2. \quad (4)$$

## 2.2. Decomposition

Figure 3 shows signal decomposition, and synthesis and restoration. The signal  $s(x, y)$  is decomposed into  $N$  parts, at  $L$  levels using the discrete wavelet transform.

$$s_{0,1}(x, y) \equiv s(x, y) \quad (5a)$$

$$s_{l,\beta}(x, y) = [s_{l-1,1}(x, y) * \tau_{A_{l,\beta}}(x, y)] \underline{\underline{\underline{q}}}, \quad \beta = 1, \dots, N, l = 1, \dots, L, \quad (5b)$$

where  $s_{l-1,1}(x, y)$  is the “low frequency” band from the previous level,  $\tau_{A_{l,\beta}}(x, y)$  are the wavelet analysis filters at the current level and  $\Downarrow_w = XY \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(v - mX, \omega - nY)$  is the downsampler by  $X = Y = \sqrt{N}$ . In the frequency domain this can be seen as dividing the passband into  $N$  segments each occupying  $1/N$ th the original bandwidth.

$$\tilde{s}_{0,1}(v, \omega) \equiv \tilde{s}(v, \omega) \quad (6a)$$

$$\tilde{s}_{l,\beta}(v, \omega) = [\tilde{s}_{l-1,1}(v, \omega) \tilde{\tau}_{A_{l,\beta}}(v, \omega)] * \Downarrow_w, \quad \beta = 1, \dots, N, l = 1, \dots, L, \quad (6b)$$

where the  $\tilde{\tau}_{A_{l,\beta}}(v, \omega)$  are, generally, orthogonal, and  $\Downarrow_w = \sum_{m=0}^{X-1} \sum_{n=0}^{Y-1} \delta(v - \frac{m}{X}, \omega - \frac{n}{Y})$ . The signals  $\tilde{s}_{l,\beta}$  occupy different frequency bands. Each signal can possess quite different characteristics and hence be amenable to different methods and rates of quantization. This provides a versatile method for efficient signal representation.

Equations 5 show the relationship between the signal  $s(x, y)$  and the decomposed signals  $s_{l,\beta}(x, y)$  in terms of the wavelet analysis filters and the downsampler. More explicitly, using Equations 1,

$$s_{1,\beta}(x, y) = \left\{ [KL(x, y) * \tau_d(x, y)] \Downarrow + N_e(x, y) + N_{a/d}(x, y) \right\} * \tau_{A_{1,\beta}}(x, y) \Downarrow_w, \quad (7a)$$

$$\tilde{s}_{1,\beta}(v, \omega) = [K\hat{L}(v, \omega) \hat{\tau}_d(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega)] * \Downarrow' + [\tilde{N}_e(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega) + \tilde{N}_{a/d}(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega)] * \Downarrow_w, \quad (7b)$$

where  $\Downarrow' = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(v - \frac{m}{X}, \omega - \frac{n}{Y})$ , is the combination of the uniform rectangular sampling lattice of the image-gathering device and the downsampler. It has sampling intervals of  $(X, Y)$  and associated passband  $\hat{B}_l(v, \omega) = \{v, \omega : |v|, |\omega| \leq 2^{-(l+1)}\}$ .

### 2.3. Quantization

The wavelet coefficients are quantized in the spatial-frequency domain. Each coefficient,  $\tilde{s}_{l,\beta}(v, \omega)$  contains a measurable amount of information  $\mathcal{H}_{l,\beta}(v, \omega)$  (§3). McCormick<sup>9</sup> et al., and Huck<sup>3, 10</sup> have shown that maximizing information leads to maximizing the image quality. Hence the coefficients are quantized under the constraint of preserving a certain percentage of maximum realizable information. The quantization transforms the wavelet coefficient  $\tilde{s}_{l,\beta}(v, \omega)$  into a representation  $Q[\tilde{s}_{l,\beta}(v, \omega)]$ . The inverse process transforms the  $Q[\tilde{s}_{l,\beta}(v, \omega)]$  back to the coefficients plus, perhaps, a noise term.

$$\tilde{s}'_{l,\beta}(v, \omega) = \tilde{s}_{l,\beta}(v, \omega) + \tilde{N}_{Q_{l,\beta}}(v, \omega), \quad \beta = 2, \dots, N \quad (8a)$$

$$\tilde{s}'_{l,1}(v, \omega) = \sum_{\beta=1}^N \tilde{s}'_{l+1,\beta}(v, \omega) \tilde{\tau}_{R_{l+1,\beta}}(v, \omega) \quad (8b)$$

where  $\tilde{s}'_{l,\beta}(v, \omega)$  are the dequantized coefficients, and  $\tilde{N}_{Q_{l,\beta}}(v, \omega)$  represents the “noise” due to quantization/dequantization process. The initial conditions are given by

$$\begin{aligned} \tilde{s}'_{1,\beta}(v, \omega) &= \tilde{s}(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega) \\ &= [K\hat{L}(v, \omega) \hat{\tau}_d(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega)] * \Downarrow' + [\hat{N}_e(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega)] * \Downarrow_w \\ &\quad + [\hat{N}_{a/d}(v, \omega) \tilde{\tau}_{A_{1,\beta}}(v, \omega)] * \Downarrow_w + \tilde{N}_{Q_{1,\beta}}(v, \omega). \end{aligned} \quad (8c)$$

Assuming uniform quantization for each wavelet coefficient between the range  $\pm k\sigma_{l,\beta}(v, \omega)$  we can write the PSD of the wavelet quantization noise as

$$\tilde{\Phi}_{N_{Q_{l,\beta}}}(v, \omega) = \sigma_{N_{Q_{l,\beta}}}^2(v, \omega) = \frac{1}{3} \left( \frac{k\sigma_{l,\beta}(v, \omega)}{\kappa_{l,\beta}(v, \omega)} \right)^2, \quad (9)$$

where  $\sigma_{l,\beta}(v, \omega)$  are the ensemble standard deviations for the coefficient at frequency  $(v, \omega)$  in band  $\beta$  and level  $l$ , and  $\kappa_{l,\beta}(v, \omega)$  is the number of quantization levels at that location.

## 2.4. Synthesis

The synthesis filters reconstruct the decomposed signals. When the coefficients are not quantized, this can be done perfectly.<sup>11, 12, 13</sup> With lossy quantization, however, the synthesis filters are generated by minimizing the minimum mean square error  $\epsilon^2(v, \omega)$ , where

$$\epsilon^2(v, \omega) = E [|\tilde{s}_{l,1}(v, \omega) - \tilde{s}'_{l,1}(v, \omega)|^2],$$

$\tilde{s}_{l,1}(v, \omega)$  is the input and  $\tilde{s}'_{l,1}(v, \omega)$  is the output to level  $l + 1$ . Using Equations 6 and 8,

$$\tilde{r}_{R_{l+1},\beta}(v, \omega) = \frac{\tilde{\Phi}_{S_{l,1}}(v, \omega) \tilde{r}_{A_{l+1},\beta}^*(v, \omega)}{\left[ \tilde{\Phi}_{S_{l,1}}(v, \omega) |\tilde{r}_{A_{l+1},\beta}(v, \omega)|^2 \right] * \hat{\underline{\underline{\downarrow}}}_w + \tilde{\Phi}_{N_{Q_{l+1},\beta}}(v, \omega)}, \quad (10)$$

where  $\tilde{\Phi}_{S_{l,1}}(v, \omega)$  is the PSD of  $\tilde{s}_{l,1}(v, \omega)$ , and  $\tilde{\Phi}_{N_{Q_{l+1},\beta}}(v, \omega)$  is the PSD of the quantization noise in band  $l + 1$ . The  $\tilde{\Phi}_{S_{l,\beta}}(v, \omega)$  can be defined by a set of iterative equations

$$\tilde{\Phi}_{S_{0,1}}(v, \omega) = \left[ K^2 \hat{\Phi}_L(v, \omega) |\hat{r}_d(v, \omega)|^2 \right] * \hat{\underline{\underline{\downarrow}}} + \tilde{\Phi}_{N_{a/d}}(v, \omega) + \tilde{\Phi}_{N_c}(v, \omega) \quad (11a)$$

$$\tilde{\Phi}_{S_{l,\beta}}(v, \omega) = \tilde{\Phi}_{S_{l-1,1}}(v, \omega) |\tilde{r}_{A_{l,\beta}}(v, \omega)|^2 * \hat{\underline{\underline{\downarrow}}}_w \quad (11b)$$

$$\approx \left[ \tilde{\Phi}_{S_{0,1}}(v, \omega) |\tilde{r}_{A_{l,\beta}}(v, \omega)|^2 \prod_{k=1}^{l-1} |\tilde{r}_{A_{k,1}}(v, \omega)|^2 \right] * \hat{\underline{\underline{\downarrow}}}_w, \quad (11c)$$

where  $\hat{\underline{\underline{\downarrow}}}_l = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(v - m/X^l, \omega - n/Y^l)$ .

## 2.5. Restoration

The Wiener-matrix restoration filters  $\hat{\Psi}_\beta(v, \omega)$ <sup>5, 14</sup> synthesize the  $N$  level-1 outputs  $\tilde{s}'_{1,\beta}(v, \omega)$  into a single *continuous* image  $\mathcal{R}$  given by

$$\mathcal{R}(v, \omega) = \sum_{\beta=1}^N K^{-1} \hat{\Psi}_\beta(v, \omega) \tilde{s}'_{1,\beta}(v, \omega) \quad (12)$$

where

$$\hat{\Psi}_\beta(v, \omega) = \hat{\Phi}_L(v, \omega) \hat{r}_d^*(v, \omega) \sum_{\alpha=1}^N \tilde{r}_{A_{1,\alpha}}^*(v, \omega) \left[ \hat{\mathcal{T}}^{-1}(v, \omega) \right]_{\alpha\beta}, \quad (13)$$

$$\left[ \hat{\mathcal{T}}(v, \omega) \right]_{\beta\alpha} = \left[ \hat{\Phi}_L(v, \omega) |\hat{r}_d(v, \omega)|^2 \tilde{r}_{A_{1,\beta}}(v, \omega) \tilde{r}_{A_{1,\alpha}}^*(v, \omega) \right] * \hat{\underline{\underline{\downarrow}}}' + \left[ \tilde{\Phi}_{N_c}(v, \omega) \tilde{r}_{A_{1,\beta}}(v, \omega) \tilde{r}_{A_{1,\alpha}}^*(v, \omega) \right] * \hat{\underline{\underline{\downarrow}}}_w \quad (14)$$

$$+ \left[ \tilde{\Phi}_{N_{a/d}} \tilde{r}_{A_{1,\beta}}(v, \omega) \tilde{r}_{A_{1,\alpha}}^*(v, \omega) * \hat{\underline{\underline{\downarrow}}}_w \right] + \tilde{\Phi}_{N_{Q_{1,\beta}}}(v, \omega) \delta(\beta, \alpha),$$

$\hat{\Phi}_L(v, \omega)$ ,  $\tilde{\Phi}_{N_c}(v, \omega)$ ,  $\tilde{\Phi}_{N_{a/d}}(v, \omega)$  and  $\tilde{\Phi}_{N_{Q_{1,\beta}}}(v, \omega)$  represent the power spectral density (PSD) of the radiance field, the photodetector noise, the A/D quantization noise, and the level-1 quantization noise respectively. When  $\tilde{r}_{A_{1,\beta}}(v, \omega) \tilde{r}_{A_{1,\alpha}}^*(v, \omega) = |\tilde{r}_{A_{1,\beta}}(v, \omega)|^2 \delta(\beta, \alpha)$ , the Wiener matrix filter reduces to

$$\hat{\Psi}_\beta(v, \omega) = \frac{\hat{\Phi}_L(v, \omega) \hat{r}_d^*(v, \omega) \tilde{r}_{A_{1,\beta}}^*(v, \omega)}{\tilde{\Phi}_s(v, \omega) |\tilde{r}_{A_{1,\beta}}(v, \omega)|^2 * \hat{\underline{\underline{\downarrow}}}_w + \tilde{\Phi}_{N_{Q_{1,\beta}}}(v, \omega)}. \quad (15)$$

The formulation of the Wiener-matrix filter takes into account the degradations due to not only the blurring and noise<sup>15, 16</sup> but also due to insufficient sampling, the analysis filter response, and quantization. It also suppresses the blurring and raster effects in image display by interpolating the image-gathering lattice on an at least 4 times finer image-display lattice.

### 3. Information

The information rate,  $\mathcal{H}$ , is used to evaluate the performance of the visual communication channel. The only information the observer has about the incident radiance field is that contained in the restored image. The degradations due to aliasing and various noise sources appear as artifacts. The information at each level in each decomposed band  $\mathcal{H}_{l,\beta}$  is

$$\mathcal{H}_{l,\beta} = \iint_{\tilde{B}_l} \mathcal{H}_{l,\beta}(v, \omega) dv d\omega = \frac{1}{2} \iint_{\tilde{B}_l} \log_2 \left[ 1 + \frac{\tilde{\Phi}_{S_{l-1,1}}(v, \omega) |\tilde{r}_{A_{l,\beta}}(v, \omega)|^2}{\tilde{\Phi}_{S_{l-1,\beta}}(v, \omega) |\tilde{r}_{A_{l,\beta}}(v, \omega)|^2 * \underbrace{\text{III}}_a + \tilde{\Phi}_{Q_{N_{l,\beta}}}(v, \omega)} \right] dv d\omega. \quad (16)$$

where  $\underbrace{\text{III}}_a = \sum_{m=0}^{X-1} \sum_{n=0}^{Y-1} \delta(v - \frac{m}{X}, w - \frac{n}{Y})$ ,  $m = n \neq 0$ , are the sidebands of the down sampler, and the PSDs  $\tilde{\Phi}_{S_{l,1}}(v, \omega)$  are given by Equation 11. The total information for the visual communication channel is

$$\mathcal{H} = \sum_{l=1}^{L-1} \sum_{\beta=2}^N \mathcal{H}_{l,\beta} + \sum_{\beta=1}^N \mathcal{H}_{L,\beta}.$$

### 4. Algorithm and Simulation Setup

Since Equation 16(a) provides a measure for the contribution of each wavelet coefficient to the total information, it can be used to devise the quantization scheme. The scheme is as follows:

1. Quantize all the coefficients at the maximum rate.
2. Determine the loss in information due to a reduction in the quantization rate for each coefficient.
3. Reduce the quantization rate of the coefficient suffering the least loss.
4. Iterate this process until either the bit bank is empty, or the required information tolerance<sup>2</sup> is reached.

Because this process uses a statistical measure, the quantization table thus obtained can be used for a wide range of input scenes, producing excellent results for input radiance fields which closely match the assumed  $\Phi_L$  and good results for others.

Since it is virtually impossible to successfully estimate all the parameters of an image-gathering device from the received signal, we use simulated imagery to test our algorithm. We use targets made up of randomly generated polygons whose mean spatial detail—the average distance between edges— $\mu$ , is Poisson distributed and intensity levels are Gaussian distributed with standard deviation  $\sigma_L$ . The targets with mean spatial details  $\mu = 1$  and 3 are shown in Figure 4. The associated radiance field  $L(x, y)$  is stationary and Gaussian. We assume that both the white  $N_e(x, y)$ , and the  $N_{a/d}(x, y)$  due to 8-bit A/D conversion, are uncorrelated with the radiance field. In addition, we model the SFR of the image gathering device with a Gaussian,  $\hat{r}_d(v, \omega) = e^{-\left(\frac{\rho}{\rho_c}\right)^2}$ , where  $\rho = v^2 + \omega^2$ , and  $\rho_c$ , which controls the width of the response, and hence the tradeoff between aliasing and blurring, is the point where  $\hat{r}_d(v, \omega) = 1/e \approx 0.37$  (Figure 5(b)). The restoration filters  $\Psi_\beta$  are generated at at least 4 times finer density than the sampling lattice to suppress the raster effects of the display device<sup>9</sup>.

### 5. Results

Figure 5(a) shows the maximum achievable information for a channel as a function of  $K\sigma_L/\sigma_{N_e}$  and  $\hat{r}_d(v, \omega)$  in the absence of compression. This set of curves allows us to design channels that maximize information for the given physical constraints. Since these designs provide the best image quality, they are used in investigating the effects of image compression on information and image quality. Table 1 summarizes three selected designs representing high, average, and low SNRs.

<sup>2</sup>Tolerance,  $\mathcal{T}$  is defined as  $\mathcal{T} = 100 \times \mathcal{H}(\eta)/\mathcal{H}_{max}$ , where  $\mathcal{H}_{max}$  is the maximum achievable information and  $\mathcal{H}(\eta)$  is the information at a given number of bits.

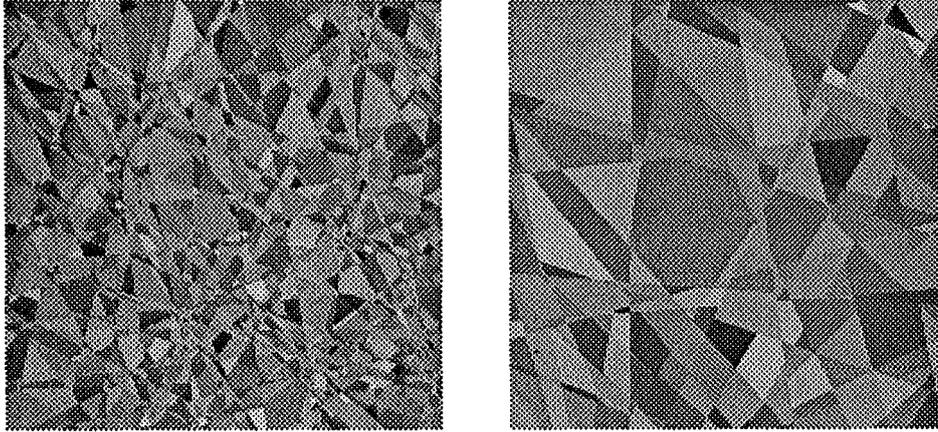


Figure 4: Random polygon targets: (a)  $\mu = 1$  (b)  $\mu = 3$

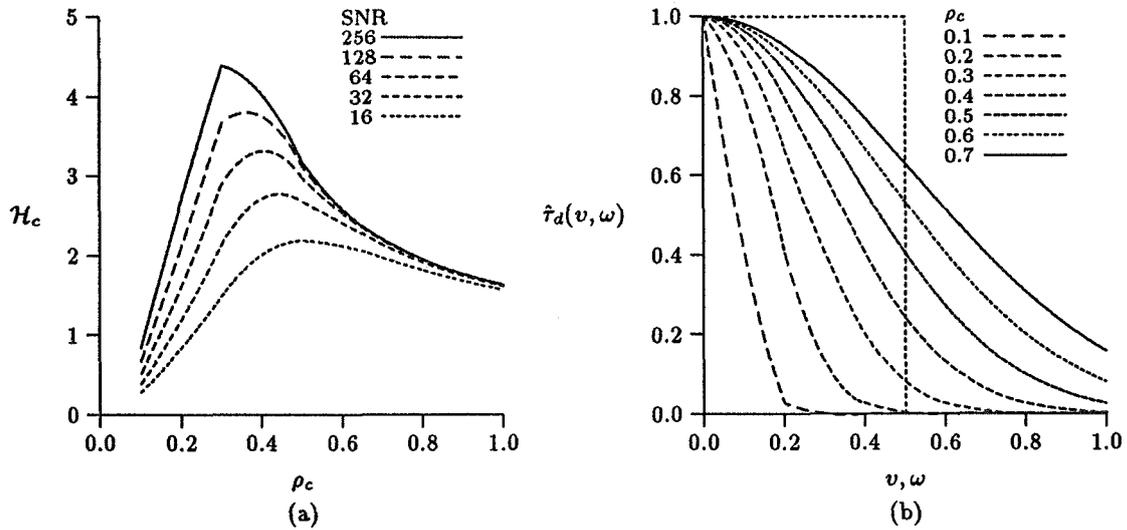


Figure 5: (a)  $\mathcal{H}$  as a function of  $\rho_c$  for several SNRs  $K\sigma_L/\sigma_{N_e}$ . The optimized design maximizes  $\mathcal{H}$  for the given  $\hat{\tau}_d(v, \omega)$  parameterized by  $\rho_c$ , and SNR  $K\sigma_L/\sigma_{N_e}$ . Information is maximum when mean spatial detail  $\mu = 1.3$  (b) The SFR  $\hat{\tau}_d(v, \omega)$  of the image-gathering device for several values of the parameter  $\rho_c$  which controls the width of the response and, hence, the tradeoff between aliasing and blurring.

Design	$\rho_c$	$K\sigma_L/\sigma_{N_e}$	$\mathcal{H}_{max}$
1	0.3	256	4.4
2	0.4	64	3.3
3	0.5	16	2.1

Table 1: Characteristics of informationally optimized image gathering (from Figure5).

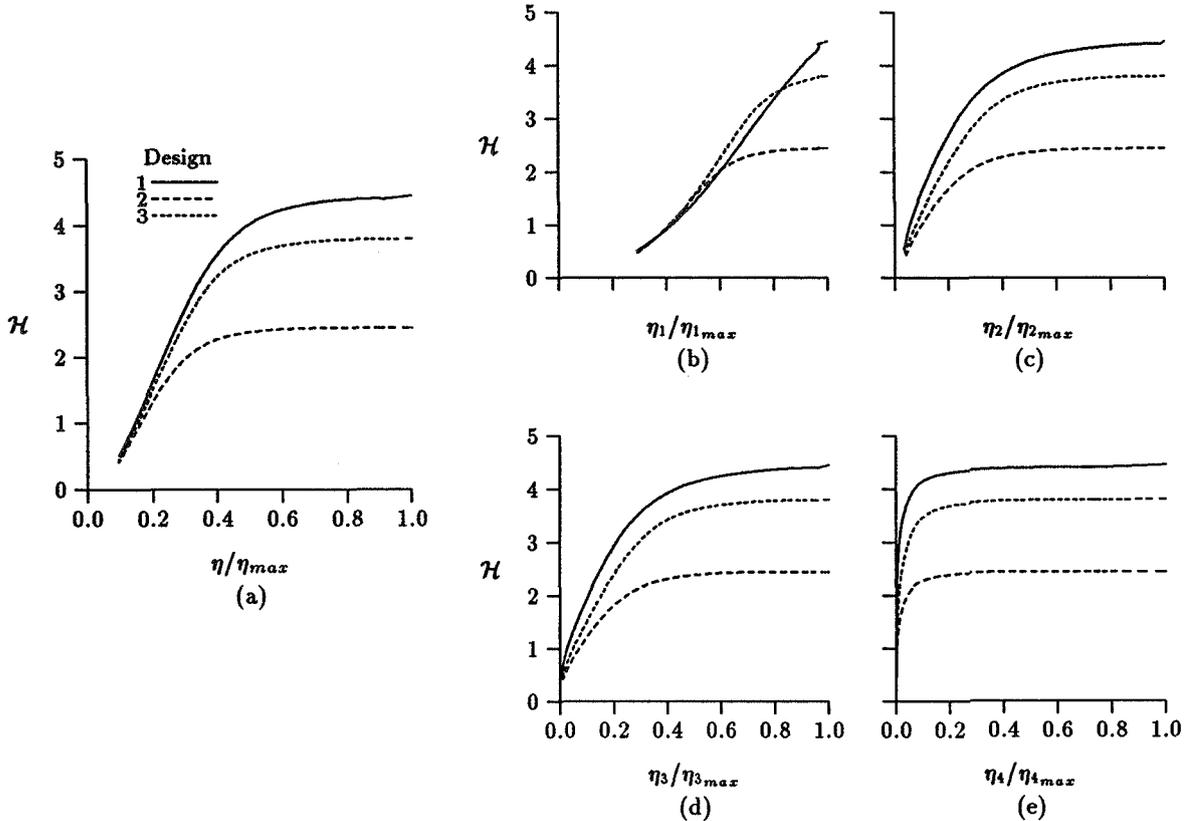


Figure 6: (a) Information  $\mathcal{H}$  as a function of the ratio  $\eta/\eta_{max}$  of the total number of bits for level-1 decomposition. (b)–(e)  $\mathcal{H}$  as a function of the number of bits in each of the decomposed band.  $\mathcal{H}$  is most affected by changing the number of bits in band 1 and least by changing the number of bits in band 4. The number of bits is decremented using the algorithm outlined in §4.

Next we evaluate how wavelet decomposition and quantization affects the total information in the restored image. From Equations 13–16, the only additional component in this calculation is the quantization noise  $\tilde{N}_{Q_{1,\beta}}(v,\omega)$ . Figure 6(a) shows how the quantization affects the total information. The information  $\mathcal{H}$  is plotted as a function of the number of bits being used for quantization. Predictably, the change in information is slow at first but then picks up momentum. Because the bit allocation algorithm (§3) reduces the number of bits for the coefficient based upon the amount of information  $\mathcal{H}(v,\omega)$  it contains, the initial reduction in the number of bits does not have a significant impact upon the total information. However, as more and more bits are discarded, the information content of the affected coefficients is higher, and the rate of information reduction increases. It is also interesting to look at the reduction in total information as a function of the reduction in the number of bits in each band (Figures 6(b)–(e)). The information content of coefficients in band 4 is very low, hence the number of bits can be reduced by about 15% before any impact is felt on the total information; coefficients in bands 2 and 3 have higher information content and have more of an impact on the total information; and the coefficients in band 1 have the highest information content as is evident by the sharp reduction in  $\mathcal{H}$  as the total number of bits for band 1 decreases.

A second point of interest is the effect on  $\mathcal{H}$  as a function of the channel design. Figure 6 shows that the total information for designs 1, 2, and 3 follows the same pattern as that in Figure 5(a) for all bit rates, though the difference becomes increasingly smaller at lower values. Thus, for a given number of bits, the reduction in information is highest for Design 1, and the lowest for Design 3 which suggests that greater data compression can be achieved for the channel that have lower SNRs.

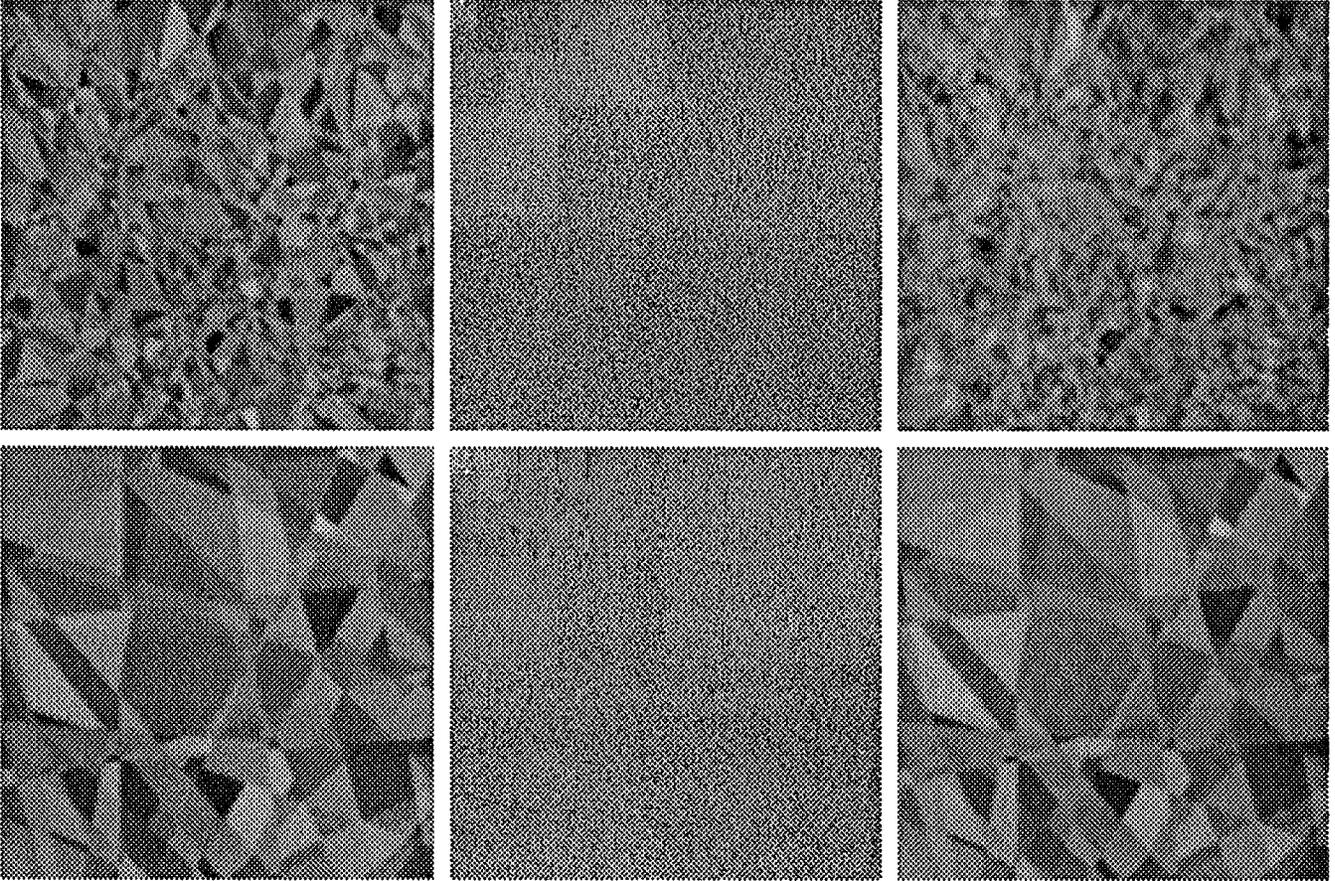


Figure 7: (a) The blurred and noisy signals (output of the image gathering stage). (b) the decomposed signals (output of the wavelet decomposition stage) at  $L = 3$ , and (c) The expected restored images. Results are shown for restoration done from level-1 components. The original scenes are shown in Figure 4.

These curves also determine the number of bits required (i.e. the compression ratio) for a given  $\mathcal{T}$ . For example, if the goal is to achieve the maximum possible compression without letting the overall information rate fall below, say,  $\mathcal{T} = 90$  then the compression ratio can be determined from these curves. Conversely, given a target compression ratio, the cost in terms of information loss can be determined as well.

Figure 7(b) shows the decomposed images of the targets shown in Figure 4, and Figure 7(c) shows the restored images. The results are shown for a tolerance level of  $\mathcal{T} = 90$ . The restorations show some obvious artifacts, more so in the  $\mu = 3$  image than in the  $\mu = 1$  image. This is because the aliasing artifacts and colored noise are, to a certain extent, masked by the detail in the scene. Conversely, one sees loss of detail in the  $\mu = 1$  scene due to the loss of high-frequency information in the quantization process. These results point to the necessity of developing better filter banks for the restoration of images.

## 6. Conclusions

A new algorithm is presented for developing quantization masks based on the total information content of a signal, and the individual contribution of the coefficients to this total. The results are encouraging in that respectable data compression is achieved even before entropy coding. The ever present tradeoff between image quality and data compression is analyzed in terms of the tradeoff between data compression and the amount of information the channel transmits. We show that the total information of the visual communication channel is a monotonically

the number of bits for transmission, does not increase the overall information in the channel.

Though the algorithm outlined here is simple conceptually, it is computationally intensive and for that reason the results presented here are for the simplest case where the analysis filters are orthogonal. This reduces the number of computations significantly but at the same time provides good insight into the results which can be expected from this approach. But this also does not fully achieve the image quality that is expected of this algorithm. Current research is looking at improving the analysis/synthesis filters, as well as improving the robustness of the restoration filters. This will lead to both better image quality in terms of human perception, but also restorability in terms of the amount of detail resolved in the displayed images.

## REFERENCES

1. C. L. Fales and F. O. Huck, "An information theory of image gathering," *Information Sciences*, vol. 57-58, pp. 245-285, 1991.
2. F. O. Huck, C. L. Fales, R. Alter-Gartenberg, and Z. Rahman, "Visual communication: Information and data transmission," *Journal of Visual Communication and Image Representation*, vol. 5, no. 3, 1994.
3. F. O. Huck, C. L. Fales, R. Alter-Gartenberg, and Z. Rahman, "On the assessment of visual communication," in *Handbook of Statistics* (N. Bose and C. R. Rao, eds.), vol. 10, Elsevier Science Publishers, 1993.
4. F. O. Huck, C. L. Fales, R. Alter-Gartenberg, Z. Rahman, and S. E. Reichenbach, "Visual communication: Information and fidelity," *Journal of Visual Communication and Image Representation*, vol. 4, no. 2, 1993.
5. C. L. Fales, F. O. Huck, R. Alter-Gartenberg, and Z. Rahman, "Multiresolution image gathering and restoration," *Journal of Visual Communication and Image Representation*, vol. 3, no. 4, pp. 356-363, 1992.
6. C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1964. Originally published in the *Bell System Technical Journal*, 27:379-423 and 28:623-656, 1948.
7. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Cambridge, MA: MIT Press, 1949.
8. Z. Rahman, "Information theoretic assessment of visual communication with wavelet coding," in *Visual Information Processing IV* (F. O. Huck and R. D. Juday, eds.), Proc. SPIE, 1995.
9. J. A. McCormick, R. Alter-Gartenberg, and F. O. Huck, "Image gathering and restoration: Information and visual quality," *Journal of the Optical Society of America A*, vol. 6, no. 7, pp. 987-1005, 1989.
10. F. O. Huck, C. L. Fales, J. A. McCormick, and S. K. Park, "Image-gathering system design for information and fidelity," *Journal of the Optical Society of America A*, vol. 5, no. 3, pp. 285-299, 1988.
11. P. P. Vaidyanathan, "Quadrature mirror filter banks, m-band extensions and perfect reconstruction techniques," *IEEE Signal Processing Magazine*, no. 7, 1987.
12. J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1278-1288, 1986.
13. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition*. Orlando, FL: Academic Press, 1992.
14. R. Alter-Gartenberg and Z. Rahman, "Multiresolution imaging: Information efficiency and visual quality," in *Visual Information Processing* (F. O. Huck and R. D. Juday, eds.), Proc. SPIE 1705, 1992.
15. M. Ozkan, T. Erdem, I. Sezan, and M. Telkap, "Efficient multiframe Wiener restoration for blurred and noisy image sequences," *IEEE Transactions on Image Processing*, vol. 1, pp. 453-476, 1992.
16. R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, second ed., 1987.

511-82  
013 182  
268906

91.

# Phase-preserving Compression of Seismic Data using the Self-adjusting Wavelet Transform

H. Guo and C.S. Burrus\*  
ECE Department, Rice University  
Houston, TX 77251-1892

## Abstract

Phase information is crucial for seismic data processing. However, traditional compression schemes do not pay special attention in preserving the phase of the seismic data, resulting in the loss of critical information. In this paper, we propose a lossy compression method that preserves the phase as much as possible. The method is based on the self-adjusting wavelet transform that adapts to the locations of the significant signal components. The elegant method of embedded zero-tree wavelet compression is modified and incorporated into our compression scheme. Our method can be applied to both one dimensional seismic signals and two dimensional seismic images.

## 1 Introduction

The seismic method plays a prominent role in the search for hydrocarbons. Seismic exploration consists of three main stages: data acquisition, processing, and interpretation. Due to the massive data acquisition activities, the need to compress the seismic data is paramount. Unlike other images, the seismic data are further processed, e.g. deconvolution, stacking, and migration, then are interpreted by geologists. So the goal of compression is not only to reduce the storage requirements, but also to avoid any processing artifacts, and eventually lead to accurate geological interpretations.

Seismic data are mostly floating point, thus less compressible compared to text when lossless compression method (Ziv-Lempel, run-length coding, etc.) is used. To achieve higher compression ratio, lossy compression method has to be used. Seismic data are well characterized as a series of transients (e.g. reflections) and thus the wavelet transform has been shown to be effective in this data domain. Recent developments in wavelet based image compression (e.g. embedded zero-tree wavelet compression[11]) have shown promising results.

Phase information is crucial for seismic data processing. However, traditional compression schemes do not pay special attention in preserving the phase of the seismic data, resulting in the loss of critical information. Many of the seismic processing methods are phase sensitive. Loss of the phase information can be easily seen after such processing, e.g. deconvolution. The loss of phase information is extremely undesirable.

Recently, the self-adjusting wavelet transform has been developed [8]. It can adapt to the locations of significant signal components, thus result in better representation of the signal and

---

\*This research is supported in part by TI, BNR, and Texas ATP grant.

accurate phase information. In this paper, we will present a phase-preserving compression method for seismic data by combining the power of the self-adjusting wavelet transform and the embedded zero-tree coding scheme.

## 2 Seismic Data

Seismic data are gathered by making man-made seismic event, then collecting the reflections from earth. The primary impulse response of the earth is,

$$e_p(t) = \sum_i c_i \delta(t - t_i), \quad (1)$$

where  $c_i$  is the reflection coefficient at the  $i^{\text{th}}$  layer. For many situations,  $c_i$  can be approximated by,

$$c_i \approx \frac{v_i - v_{i-1}}{v_i + v_{i-1}}, \quad (2)$$

where  $v_i$  is the velocity of wave propagation in the  $i^{\text{th}}$  layer. A synthetic model of water bottom is shown in Fig. 1(a). The impulse response of the earth also includes multiple reflections, e.g. surface, inter-bed, and intra-bed multiples. The strong multiple in the water bottom example is shown in Fig. 1(b). As we can see, the primary reflection is less visible due to multiples. The seismogram  $x(t)$  that we collect is

$$x(t) = w(t) * e(t) + n(t), \quad (3)$$

where  $w(t)$  is man-made signature wave (also called wavelet in seismic literature),  $e(t)$  is the impulse response of the earth including multiples, and  $n(t)$  is the noise. The synthetic seismogram of the water bottom example is shown in Fig. 1(c). The primary reflection is even less visible than in Fig. 1(b). An example of real world seismogram is shown in Fig. 4(a).

The time domain impulse response of the sampled signature wave used in the synthetic seismogram is shown in Fig. 2(a). The energy of the signature wave is concentrated at the beginning, i.e. it has minimum energy delay. The roots of the signature wave are inside the unit circle, c.f. Fig. 2(b). This type of signature wave is *minimum phase*.

## 3 Seismic Data Processing

The goal of seismic data processing is to deduce the underground structure from collected seismograms [14]. One of the most important seismic data processing techniques is *deconvolution*.

### 3.1 Spiking Deconvolution

Spiking deconvolution is a process that improves the temporal resolution of seismic data by compressing the basic signature wave. First we find  $a(t)$  such that,

$$\delta(t) = w(t) * a(t). \quad (4)$$

Then convolve the seismogram with  $a(t)$ ,

$$\hat{e}(t) = x(t) * a(t), \quad (5)$$

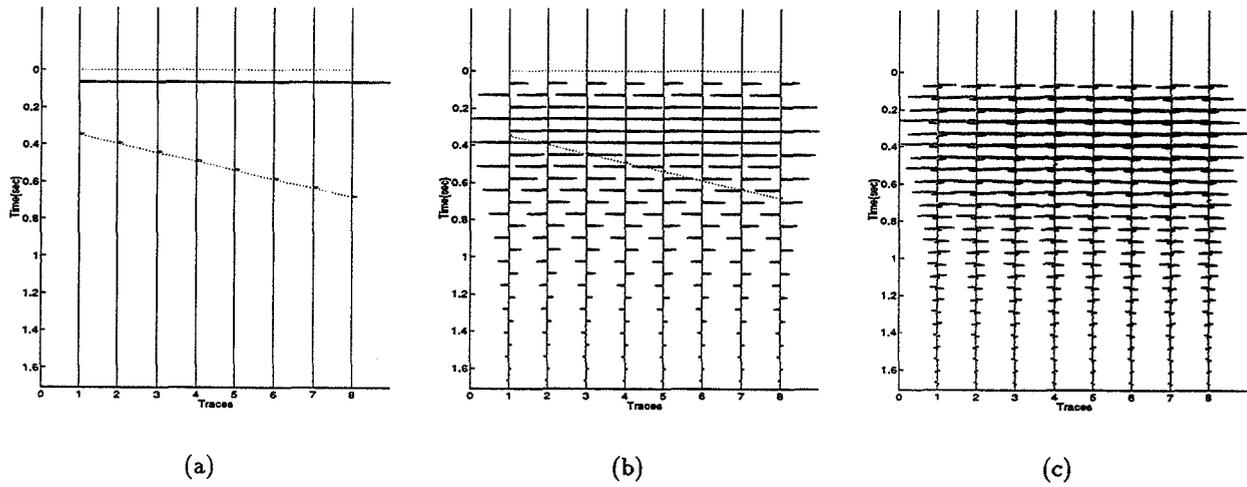


Figure 1: The water bottom example. (a) Model with only primary reflections. (b) Model with both primary and multiple reflections. (c) Synthetic seismogram.

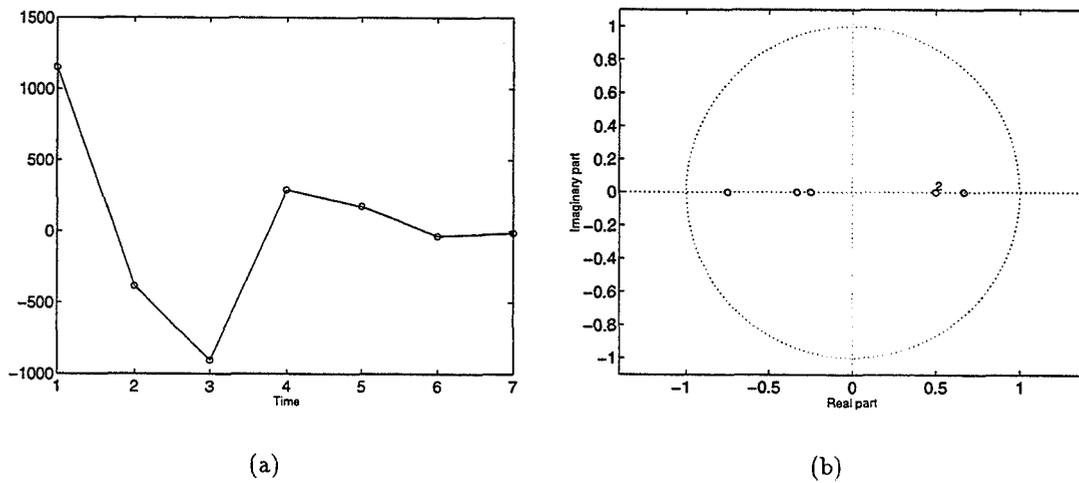


Figure 2: The signature wave for the water bottom example. (a) The time domain impulse response. (b) Zero locations.

to get an estimate of the impulse response of the earth.  $a(t)$  can be found by solving the Wiener-Hopt equation. The key requirement for  $a(t)$  to be stable is that  $w(t)$  must be minimum phase. The spiking deconvolution result for the water bottom example is shown in Fig. 3(a), and the result of the real world data is shown in Fig. 4(b). Interpreters prefer the crisp, finely detailed appearance of the deconvolved section as opposed to the blurred, ringy appearance of the section without deconvolution.

### 3.2 Predictive Deconvolution

Multiples occur at fixed intervals, so they are predictable. The method of predictive deconvolution first finds  $b(t)$ , such that

$$\hat{e}(t + \Delta t) = \hat{e}(t) * b(t). \quad (6)$$

Then calculates the prediction error,

$$\hat{e}_p(t) = \hat{e}(t) - \hat{e}(t - \Delta t) * b(t). \quad (7)$$

This method is also called the predictive error filtering (PEF). PEF can reduce multiple as shown in Fig. 3(b).

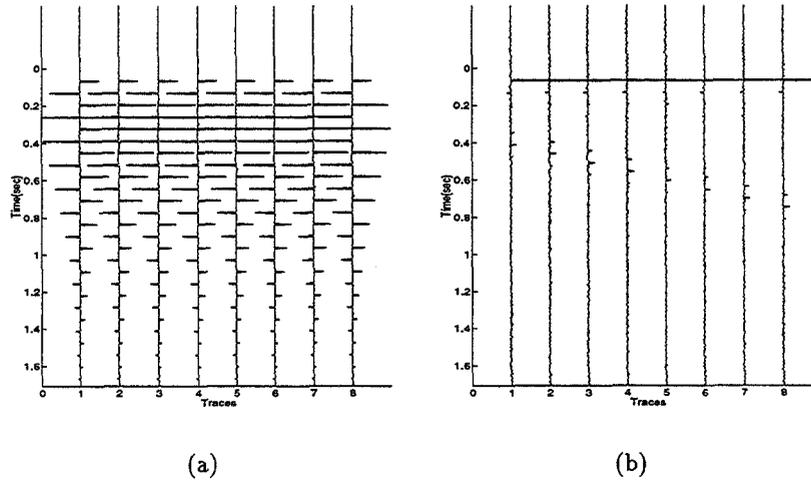


Figure 3: Processing results for the water bottom example. (a) The spiking deconvolution result based on the original data. (b) The predictive deconvolution result based on the original data.

## 4 Compression Distortion and its Effects

We use the 1D embedded zero-tree wavelet (EZW) [11] compression algorithm to compress the seismograms. Almost no visual differences can be detected. For real world pre-stack seismic images, 30:1 compression could achieve no visual differences. If we apply the same predictive deconvolution algorithm on the compressed and then decompressed data, the results are shown in Fig. 5 and

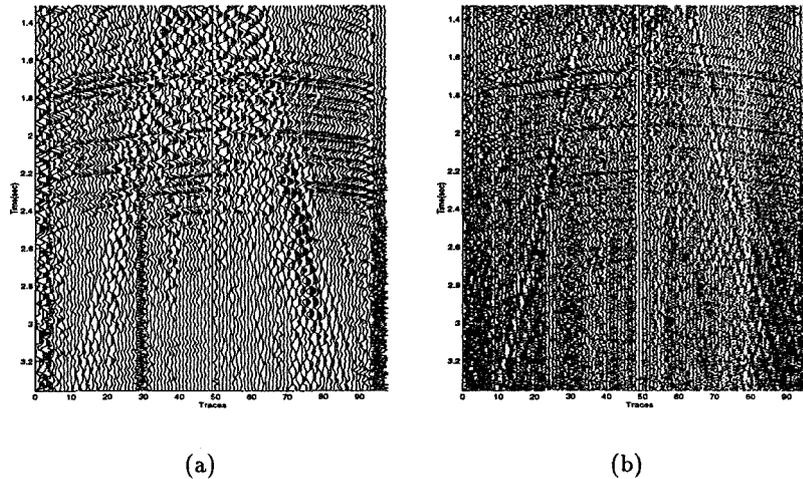


Figure 4: (a) Real world seismic data. (b) The spiking deconvolution result based on the original data.

Fig. 6. As we can see that it is far less crisp, and has much more ringings. From the theory of deconvolution, we know that the stable deconvolution requires that the wave is minimum phase. Non-minimum phase or near non-minimum phase will result in non-stable or near non-stable deconvolution, e.g. ringings in the results. This suggests that the phases of the waves in the compressed image have been changed. Also from Fig. 5 and Fig. 6, we can see that using different wavelets and different relative locations of wavelets can change the PEF results dramatically.

Similar experiments on real world seismic data suggest that, although there is no visual differences between the compressed and the original data, the phase of the signal has been changed.

## 5 Self-adjusting Wavelet Transform

The wavelet transforms not only provide a powerful multiresolution analysis of the signals, they also have built-in structures that allow us to construct huge amount of basis that have different time/space and frequency/scale characteristics. Due to the inherent hierarchical structure of the wavelet transforms, fast algorithms that expand signals onto those basis, and efficient methods that find the best basis within the huge set, exist and are practical [9, 6, 13, 12, 5, 4, 10, 3, 2].

One of the main ingredients in the wavelets transform is the downsampling at each scale. Although the downsampling reduces the output data rate and results in compact representation, it also introduces one artifact - *shift-variance*. The wavelet transform of a signal and the wavelet transform of a shifted version of the same signal is drastically different. The lack of shift invariance is one well known disadvantage of the discrete wavelet transform.

For the point of seismic signal processing, the down-sampling means that the basis functions are at some fixed locations. If the location of the significant signal component does not coincide with the location of one basis function, it will be represented by nearby basis function, thus results in phase shift.

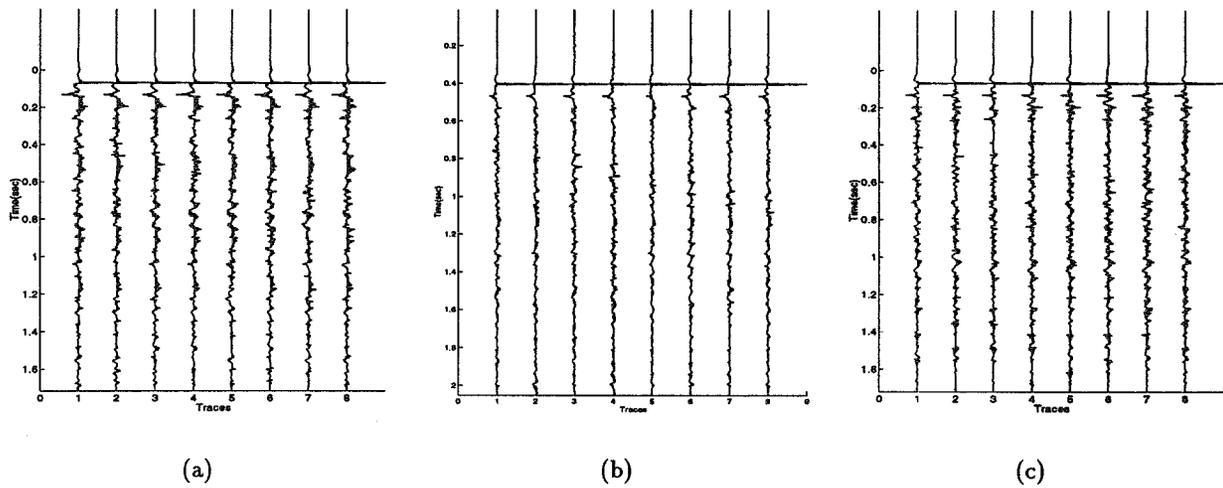


Figure 5: (a) The predictive deconvolution result based on the water bottom data compressed at 2.5 bps using using EZW algorithm, standard wavelet transform, and the Daubechies 7-9 biorthogonal wavelets. (b) The predictive deconvolution result based on the water bottom data compressed at 2.5 bps using EZW algorithm, standard wavelet transform, and the Daubechies length 6 wavelets. (c) The predictive deconvolution result based on the shifted water bottom data compressed at 2.5 bps using EZW algorithm, standard wavelet transform, and the Daubechies length 6 wavelets.

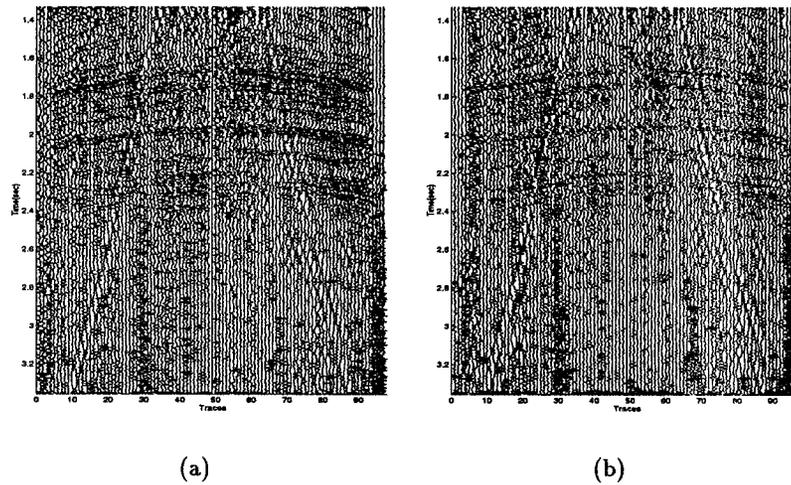


Figure 6: (a) The spiking deconvolution result based on the real world data compressed at 1.0 bps using using EZW algorithm, standard wavelet transform, and the Daubechies 7-9 biorthogonal wavelets. (b) The spiking deconvolution result based on the real world data compressed at 1.0 bps using using EZW algorithm, standard wavelet transform, and the Daubechies length 16 wavelets.

The basic DWT building block has two downsampling blocks. We realize that we can either take the even or odd indexed downsamples, and are able to reconstruct the original signal. Equivalently, we can use all the basis function at even locations to represent the signal or use all the basis function at odd locations to represent the signal. Clearly, we need to chose one that preserve the phase of the signal.

For a multiscale wavelet transform, we need to choose a path from root to the leave from a full binary tree of all possible paths. In order to find the best cost and the best tree shape, dynamic programming method [1] needs to be utilized. The dynamic programming theory requires that any subpath of the optimal solution is optimal for the subproblem. The optimal solution can be found in  $O(N \log N)$  time.

However, this is still not good enough, because one significant component could be better represented by even basis functions, while another significant component could be better represented by odd basis functions. we need to introduce the self-adjusting wavelet systems. The idea is quite simple, we can cut the input signal into several non-overlapping segments, and use different wavelet transforms on different segments. However, the task of finding the best time segmentations and the best wavelet transform on each segment is very hard, since the number of possible choices is huge. For example, for a length  $N$  signal, the number of different ways of segmenting the signal is  $2^{N-1}$ . Fortunately, if the cost function is additive, we can exploit the structure of the wavelet transform and construct a fast algorithm that finds the self-adjusting wavelet transform efficiently [7].

## 6 Phase-preserving Compression

Because of the structure of the wavelet transform and the zero-tree algorithm, our phase-preserving compression algorithm can be easily incorporated into the embedded wavelet compression scheme. The outlines of the phase-preserving compression algorithm are,

1. Find the self-adjusting wavelet transform.
2. Modified embedded wavelet compression.
  - (a) Initialization.
 

Determine and output the initial quantization-step and the initial phase.
  - (b) Dominated pass.
    - i. Determine the type of the coefficient, e.g positive significant, negative significant, isolated zero, or zero-tree root.
    - ii. If the phase has changed from previous coefficient *and* the coefficient is significant, output a phase change symbol.
    - iii. Output the type of the coefficient
  - (c) Subordinate pass.
 

Refine the values of the significant coefficients.
  - (d) Half the quantization-step and go to 2(b).
3. Adaptive arithmetic coding of the output symbol.

The algorithm terminates when the available bits run out. Notice that the phase information is transmitted in a progressive and embedded way. Because of the localization of seismic events, the localization of wavelet basis, and the localization ability of the zero-tree structure, the above phase-preserving compression algorithm works very well. As we can see from the results in Fig. 7, both amplitude and phase information have been preserved.

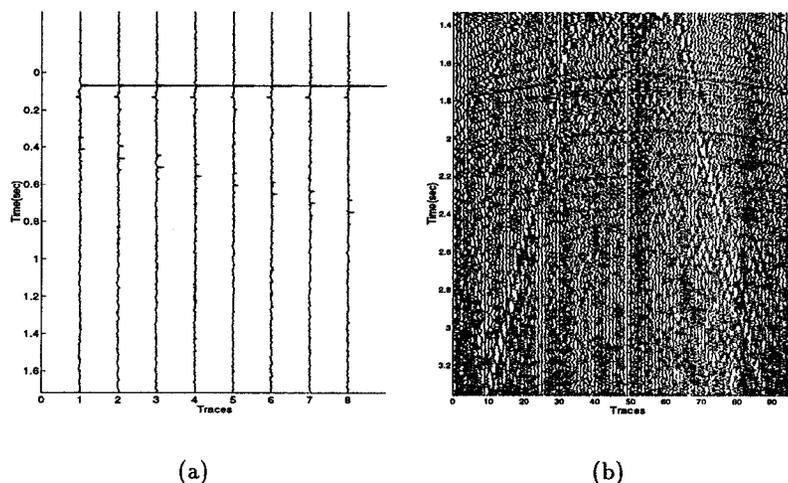


Figure 7: (a) The predictive deconvolution result based on the water bottom data compressed at 2.5 bps using EZW algorithm, self-adjusting wavelet transform, and the Daubechies length 6 wavelets. (b) The predictive deconvolution result based on the real world data compressed at 1.0 bps using EZW algorithm, self-adjusting wavelet transform, and the Daubechies length 16 wavelets.

## 7 Summary

Phase information is crucial for seismic data processing. However, the traditional compression schemes do not pay special attention in preserving the phase of the seismic data, resulting in the loss of this critical information and artifacts in subsequent processing. In this paper, we study the deconvolution and propose a lossy compression method that preserves the phase as much as possible. The method is based on the self-adjusting wavelet transform that adapts to the locations of the significant signal components. The elegant method of embedded zero-tree wavelet compression is modified and incorporated into our compression scheme.

## References

- [1] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [2] I. Cohen, S. Raz, and D. Malah. Shift invariant wavelet packet bases. In *IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing*, volume 4, pages 1080–1084, Detroit, MI, 1995. IEEE.

- [3] S. del Marco, J. Weiss, and K. Jager. Wavepacket-based transient signal detector using a translation invariant wavelet transform. In *Wavelet Applications in Signal and Image Processing*, volume 2242, pages 154–165, San Diego, CA, July 1994. SPIE.
- [4] B. Deng, B. Jawerth, G. Peters, and W. Sweldens. Wavelet probing for compression-based segmentation. In *Wavelet Applications in Signal and Image Processing*, San Diego, CA, July 1993. SPIE.
- [5] D. L. Donoho. On minimum entropy segmentation. Technical report, Stanford University, Department of Statistics, March 1994.
- [6] R. A. Gopinath and C. S. Burrus. Factorization approach to unitary time-varying filter banks and wavelets. Technical report, Rice University, Houston, TX, December 1992.
- [7] H. Guo. Redundant wavelet transform and denoising. Technical Report CML TR94-17, Rice University, Houston, TX, December 1994.
- [8] H. Guo. Theory and applications of the shift-invariant, time-varying and undecimated wavelet transform. Master's thesis, Rice University, Houston, TX, May 1995.
- [9] C. Herley, J. Kovačević, Kannan Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Transactions on Signal Processing*, 41(12):3341–3359, December 1993. Special issue on wavelets, Also Tech. Report CU/CTR/TR 315-92-25.
- [10] J. Liang and T. W. Parks. A two-dimensional translation invariant wavelet representation and its applications. In *IEEE Proc. Int. Conf. Image Processing*, volume I, pages 66–70, Austin, TX, November 1994. IEEE.
- [11] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. SP*, 41:3445–3462, 1993.
- [12] M. A. Sola and S. Sallent. Best progressive tiling of the time-frequency plane based on fast time splitting and wavelet transform. In *Proc. IEEE Conf. on Time-Frequency and Time-Scale Analysis*, pages 132–135. IEEE, 1994.
- [13] Z. Xiong, C. Herley, K. Ramchandran, and M. T. Orchard. Flexible time segmentations for time-varying wavelet packets. In *Proc. IEEE Conf. on Time-Frequency and Time-Scale Analysis*, pages 9–12. IEEE, 1994.
- [14] O. Yilmaz. *Seismic Data Processing*. Society of Exploration Geophysicists, Tulas, OK, 1987.

212 82...  
212 1831 /  
268909 '  
10 P.

## The Effects of Wavelet-based Data Compression on Flat Field Calibration for Remote Sensing Applications

Susan P. Hagerty  
shagerty@ball.com  
Ball Aerospace

Jerome M. Shapiro  
shapiro@aware.com  
Aware, Inc.

### Abstract

Remote sensing systems typically employ pushbroom sensors (linear arrays “swept” along by the satellite) to collect imagery. These line scan arrays contain pixel-to-pixel nonuniformity errors due to variations in the active area, thickness, doping, and temperature of each detector in the array. These errors are estimated and corrected using flat field calibration. Assuming the detectors exhibit a nominally linear response, this technique inputs flat radiance fields at two radiance levels, estimates the gain and offset of each pixel, determines correction factors and applies them to each pixel. The purpose of this paper is to describe the effects of performing this calibration through wavelet-based data compression. The two major steps involved in this calibration process—computing the correction coefficients and applying the correction coefficients—can be done *prior to* or *after* the data compression. To minimize on-board hardware, it is preferable to compute and apply the correction factors after the compression, i.e., on the ground. We determined the feasibility of using this desired approach and compared it to: (1) the more ideal approach of computing and applying the correction factors prior to any data compression, and (2) an alternative approach of computing the coefficients prior to compression and applying the coefficients after compression.

### 1. Introduction

Data compression is an enabling technology in the remote sensing arena. Current high-end sensor data rates of remote sensing systems are pushing 2 to 4 Gbits/sec. Data rates in next-generation systems are growing due to the desire for larger ground coverage and swath widths, increased number of spectral bands, and multiple on-board sensors. Current commercial downlinks are only around 250 Mbits/sec. Therefore, assuming 12-bit input data, high-end systems will require a bit rate of 0.75 bits per pixel (bpp) (16:1) to 1.5 bpp (8:1) with next-generation systems demanding 0.24 bpp (50:1) to 0.6 bpp (20:1). Not only must this range of compression ratios be obtained but they must be supplied with negligible discernible degradation of image quality. Cost-effective hardware for these data compression systems must be available in a timely manner, and with a reasonable size, weight, power, and throughput configuration. Given these considerations, we determined that it is highly desirable to put a wavelet-based compression system on-board the satellite to realize the full potential of next-generation remote sensing systems. Wavelet-based compression provides low bit rate compression with high image quality and reasonable computational complexity. It offers better performance at low bit rates than other common techniques such as JPEG, vector quantization, and DPCM [1] yet can still be implemented in a real-time system with realistic on-board size, weight, and power constraints. Ball and Aware are currently developing hardware towards this end.

An important parameter in the overall design of a remote sensing system as well as to the remote sensing community, is to understand and assess the impacts of adding data compression to the satellite payload. In particular, it is important to address system issues such as end-to-end performance, bit error susceptibility, and flat field calibration. This paper addresses flat field calibration. Typical remote sensing systems use a line-scan array oriented in the cross-track direction (perpendicular to the direction of travel) that is “pushed” along by the satellite. This sensor configuration is commonly called a “pushbroom sensor”. Each detector in the array has a different gain and offset due to variations in the active area, thickness, doping, and temperature resulting in a different response to a uniform input. Without correction, this phenomenon would result in striping in the image (shown later in Figure 5). Thus, a necessary requirement for a remote sensing system is to correct for this striping. This correction process is straightforward without data compression in the system. However, this is not the case for a system with data compression and therefore we have undertaken a study to investigate the effects of wavelet-based data compression on this flat field calibration process. The purpose of this paper is to describe the results of this

study. Section 2 of this paper discusses flat field calibration for remote sensing applications. In section 3, a brief overview of wavelet-based compression is provided. Section 4 presents results of the study, while section 5 offers some conclusions.

## 2. Flat Field Calibration

Flat field calibration is performed on focal plane arrays to try to mitigate the effects of pixel-to-pixel nonuniformities. For a line scan array this calibration process is implemented by: (1) inputting two known radiance values to all pixels and obtaining the pixel responses, (2) spatially averaging each pixel along the in-track direction (same as direction of travel), (3) determining the gain and offset for each pixel from the averaged responses, (4) computing the correction factors required to give all pixels the same nominal gain and offset value, and (5) applying these correction factors on a line-by-line basis to the array.

Note that the flat field calibration data can be taken with or without data compression in the system (since real-time data requirements can be relaxed when gathering calibration data). Also, assuming the real-time system (for collecting remotely sensed imagery) will contain data compression, the correction factors can be applied *prior to* or *after* the data compression. To minimize on-board hardware, it is preferable to apply the correction factors after the compression, i.e., on the ground. This paper determines the feasibility of this desired approach and compares it to the more ideal approach of computing and applying the correction factors prior to any compression of the data. The performance of the calibration is evaluated on both flat field images and images representative of remotely sensed imagery.

### 2.1 Simulation of Calibration Flat Field Images

Parameters from a typical remote sensing line-scan sensor were used for the simulation. The pixels on each line have normally distributed gains,  $\mathbf{m}$ , with a mean of 1 and standard deviation of 5%. They also have normally distributed offsets,  $\mathbf{b}$ , with a mean of 1300 e- and a standard deviation of 15% of 1300. The LSB of the analog-to-digital converter is equal to 100 e-, i.e., there are 100 e-/count. The low radiance calibration value,  $x_0$ , is 3,800 e- and the high radiance calibration value,  $x_1$ , is 150,000 e-. Temporal noise,  $\mathbf{n}$ , was added which includes shot noise<sup>1</sup> and other noise sources equal to the least significant bit, i.e., this noise is modeled by a normally distributed random variable with zero mean and variance of pixel value + 100<sup>2</sup>. The resulting data values are 12-bits wide and are stored in 16-bit integers. Taking into account the above information, the flat field images (at low ( $k=0$ ) and high ( $k=1$ ) radiance values) were generated by

$$\mathbf{y}_k = \text{round}\left(\left(\mathbf{m}x_k + \mathbf{b} + \mathbf{n}\right) / c\right) \quad \text{where } c \text{ represents counts/e- (nom. = 100)} \quad (1)$$

or

$$\mathbf{y}_k = \left(\mathbf{m}x_k + \mathbf{b} + \mathbf{n}\right) / c + \mathbf{n}_q \quad \text{where } \mathbf{n}_q \text{ is the quantization noise, } U[-.5,.5] \quad (2)$$

For an individual pixel  $i$ , we have

$$\mathbf{y}_{k,i} = \left(m_i x_k + b_i + \mathbf{n}\right) / c + \mathbf{n}_q \quad \text{where } m_i, b_i \text{ are the gain and offset for pixel } i \quad (3)$$

### 2.2 Determining Correction Factors

Ideally, we want each pixel to have the same response:

$$\hat{\mathbf{y}}_{k,i} = \left(\hat{m}x_k + \hat{b}\right) / c \quad (4)$$

---

<sup>1</sup> Shot noise is Poisson distributed but can be approximated by a Gaussian distribution for large values

where  $\hat{m}$  and  $\hat{b}$  are the sample means of the gains and offsets, respectively

Estimating  $x_{k,i}$  from (3), we have

$$\hat{\mathbf{x}}_{k,i} = \left( c(\mathbf{y}_{k,i} - \mathbf{n}_q) - b_i - \mathbf{n} \right) / m_i \quad (5)$$

Note that

$$E\hat{\mathbf{x}}_{k,i} = \left[ c(m_i x_k + b_i) / c - b_i \right] / m_i = x_k \quad (6)$$

Plugging (5) into (4), we get the corrected response for each pixel

$$\begin{aligned} \hat{\mathbf{y}}_{k,i} &= \frac{1}{c} \left[ \hat{m} \left[ c(\mathbf{y}_{k,i} - \mathbf{n}_q) - b_i - \mathbf{n} \right] / m_i + \hat{b} \right] \\ &= \frac{\hat{m}}{m_i} \mathbf{y}_{k,i} - \frac{\hat{m}}{m_i} \mathbf{n}_q - \frac{\hat{m}}{m_i c} (b_i + \mathbf{n}) + \frac{\hat{b}}{c} \end{aligned} \quad (7)$$

The expected value of  $\hat{\mathbf{y}}_{k,i}$  is

$$E\hat{\mathbf{y}}_{k,i} = \frac{\hat{m}}{m_i} E\mathbf{y}_{k,i} + \frac{1}{c} \left[ \hat{b} - \frac{\hat{m}}{m_i} (b_i) \right] \quad (8)$$

and therefore, the gain and offset correction factors are, respectively

$$\frac{\hat{m}}{m_i} \quad \text{and} \quad \frac{1}{c} \left[ \hat{b} - \frac{\hat{m}}{m_i} (b_i) \right] \quad (9)$$

To check that these correction factors give the desired result, we further expand and simplify the expected value of  $\hat{\mathbf{y}}_{k,i}$  :

$$E\hat{\mathbf{y}}_{k,i} = \frac{\hat{m}}{m_i} \left[ (m_i x_k + b_i) / c \right] + \frac{1}{c} \left[ \hat{b} - \frac{\hat{m}}{m_i} (b_i) \right] = (\hat{m} x_k + \hat{b}) / c \quad (10)$$

which agrees with our idealized response in (4).

### 3. Wavelet-based Compression

The first step in wavelet-based data compression is wavelet decomposition. This consists of decomposing the input image into different frequency subbands at different spatial resolutions via multiresolution analysis (MRA). In 1-D, an MRA consists of a pyramidal algorithm where level  $j$  of the pyramid contains a  $2^j$  resolution smooth approximation of the input signal as well as a  $2^j$  resolution error or detail signal. The detail signal represents the difference in information between the signal approximations at resolutions  $2^{j+1}$  and  $2^j$ . This detail information can be extracted by decomposing the signal on a wavelet orthogonal basis[2]. The smooth and detail signals at resolution  $2^j$  are simply computed by convolving the smooth signal at resolution  $2^{j+1}$  with a lowpass filter (dilated and translated version of the “scaling” function) and a bandpass filter (dilated and translated version of the “wavelet” function), respectively, and decimating each filtered signal by 2[3]. Thus, for a 1-D signal, the smooth approximation at level  $j$  will contain half the pixels as the smooth approximation at level  $j+1$ . The “other half” of

the pixels will contain the detail information. Extending this to 2-D images (which are treated in a separable manner), we have that the number of pixels to represent the smooth approximation at level  $j$  is  $1/4$  of the number of pixels required to represent the previous smooth approximation at level  $j+1$  ( $(1/2$  due to vertical decimation by  $2$ )  $\times$   $(1/2$  due to horizontal decimation by  $2$ )). The other  $3/4$  of the pixels that comprised the smooth approximation at level  $j+1$  now contain detail information at level  $j$ . Hence if the error (detail) between approximations at successive levels is small, the overall wavelet decomposition will be sparse (see Figure 1).

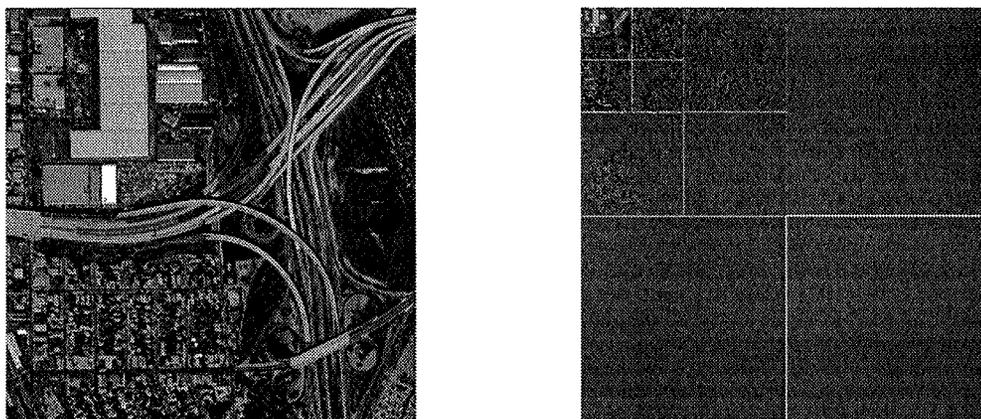


Figure 1. Sample input image and 3-level wavelet decomposition. Note sparseness of decomposition with exception of upper left block.

This sparse representation of the image allows many coefficients to be discarded (or allocated a small number of bits) after quantization. The steps following computation of the wavelet decomposition are quantization and entropy encoding. The quantization step consists of intelligently quantizing the wavelet coefficients so that the target bit rate is achieved with minimum degradation in the image quality of the reconstructed image. Techniques for quantization come from the rate-distortion branch of information theory[4]. Some examples of quantization techniques as applied to data compression are contained in [5][6][7]. Entropy encoding provides additional lossless compression and generates the bitstream that is transmitted to the ground and/or on-board storage device. Common entropy encoding techniques employed in lossy data compression schemes include Huffman coding and arithmetic coding (see [4]).

Wavelet-based compression provides high image quality at low bit rates due to: (1) the sparse nature of the wavelet representation, (2) the optimized allocation of bits, and (3) the absence of objectionable blocking artifacts (as observed in JPEG) since processing operates on the entire image (or very large sections of an image). This technique also exhibits reasonable computational complexity since the bulk of the processing is in the wavelet decomposition step which simply consists of a finite number of convolutions and decimations of the input image with small separable kernels (on the order of 2-20 pixels). Also, Ball/Aware-developed hardware for implementing wavelet-based compression in real-time systems will be available in the near future. These attributes make wavelet-based compression very attractive to incorporate into a remote sensing system. The results presented in this paper are obtained using a software simulator of the Ball/Aware-developed wavelet compression hardware.

#### 4. Simulation Results

The simulation results are split into two sections: one for correcting flat field images, and one for correcting representative images. For both sections, simulation results are presented for three cases: i) computing and applying the correction factors prior to compression (*Prior*), ii) computing the correction factors prior to compression and applying the correction factors after compression (*Mixed*), and iii) computing and applying the correction factors after the compression (*Post*). Cases (i) and (ii) require either a compression bypass option or a lossless compression option in calibration mode (non real-time) so that calibration data can be taken and correction factors computed without compression. For cases (ii) and (iii), where the correction factors are applied after compression, the calibration hardware for applying these corrections may reside on the ground. For case (i), where

the correction factors are applied prior to the on-board compression, calibration hardware must be included on-board.

#### 4.1 Correcting flat field images

The objective in trying to correct non-uniform “flat field” images is to obtain, as closely as possible, a true flat field, i.e., for a constant input radiance, all responses ideally are identical. To evaluate the performance of the correction, we wish to compute the standard deviation across the detectors in the line array. Prior to computing these statistics, the pixels are averaged in the in-track direction to reduce the effects of noise. To evaluate the performance of the corrections, a third radiance level ( $x_2 = 75,000 e^-$ ) is also corrected which is not used in the computation of the correction values. This gives an independent test of how well the correction performs on values not used during the calibration process.

One way to evaluate the performance of the correction is to look at detector response versus flux level parameterized over individual pixels. Plots of detector response versus flux level for the the uncorrected pixel responses and corrected pixel responses (for the *Prior*, *Mixed*, and *Post* cases as defined above) are shown in Figure 2. One hundred pixels are plotted in each graph. The plots for the *Mixed* and *Post* cases are for a bit rate of 1.2 bpp. In the ideal corrected case, every pixel would exhibit the same response. It is clear that the *Prior* case performs best, followed by the *Post* case, and then the *Mixed* case.

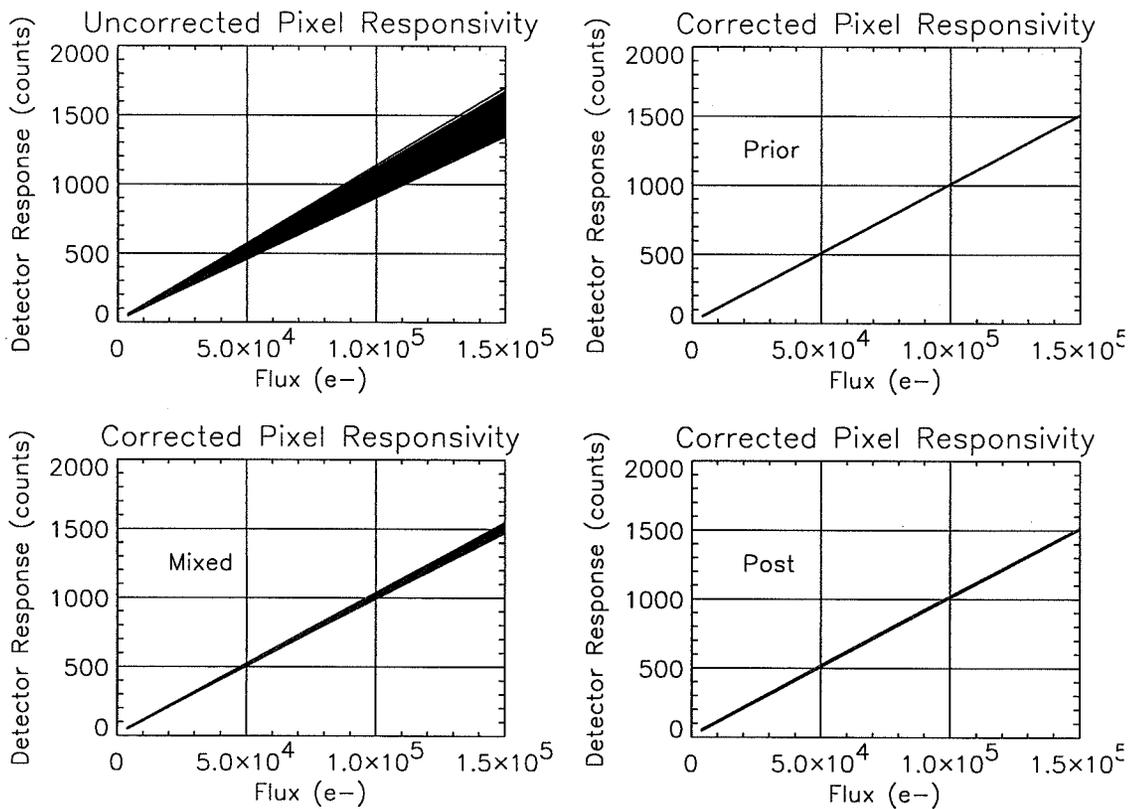


Figure 2. Detector Response vs. Input Flux for uncorrected case (upper left) and *Prior* (upper right), *Mixed* (bottom left), and *Post* (bottom right) corrected cases

More quantitative results are shown in Figure 3. Here plots of the line-to-line standard deviation versus bits per pixel are shown for the high, medium, and low input radiance levels. By line-to-line standard deviation we mean the standard deviation from detector to detector in the corrected image after averaging in the in-track direction. These plots are parameterized over correction technique (*Uncorrected*, *Mixed*, *Post*). Note that, since the *Prior*

case is computed prior to any compression, this case is implicitly contained in these plots as the values at the original bit rate of 12 bpp.

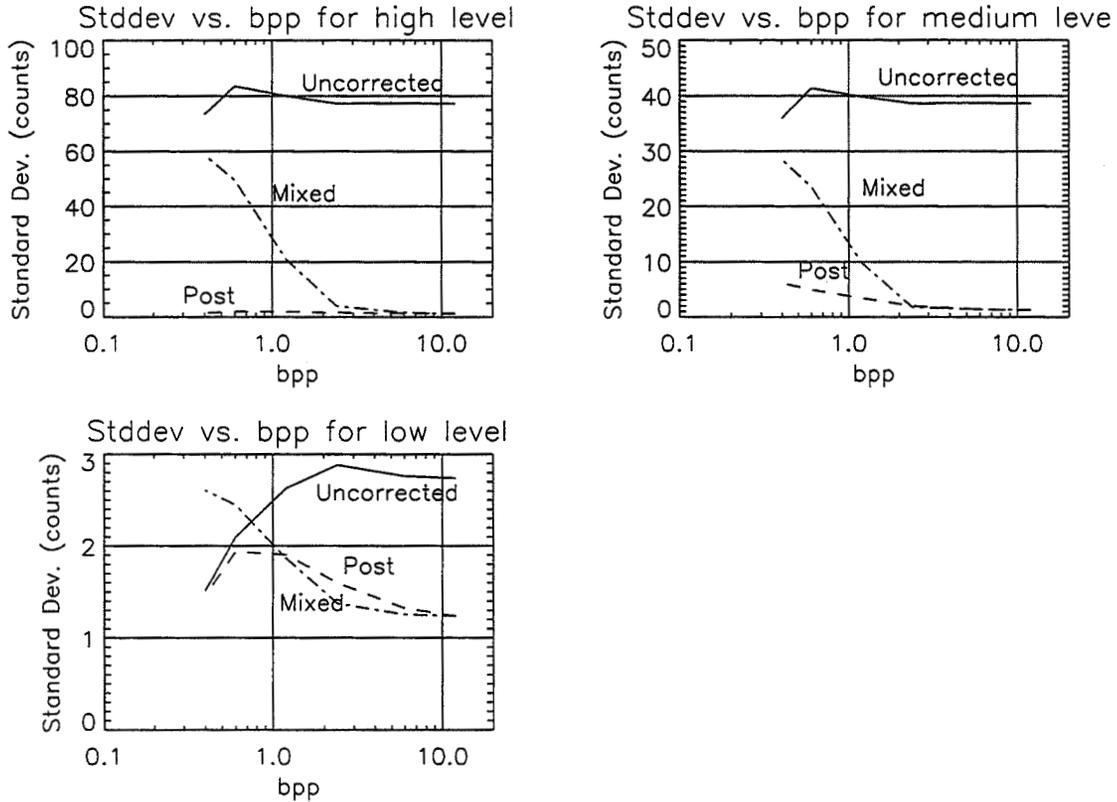


Figure 3. Line-to-line standard deviation vs. bit rate for the low, medium, and high radiance levels.

These plots show that the performance of the correction is best when the correction coefficients are both computed and applied prior to compression (*Prior* case). Also, the performance of the correction is superior, in general, when the correction coefficients are both computed and applied after compression (*Post* case) as opposed to computed prior to compression and applied after compression (*Mixed* case). To explain this behavior, we offer the following argument. Prior to compression we have

$$y_{k,i} = (m_i x_k + b_i + \mathbf{n}) / c + \mathbf{n}_q \quad (11)$$

with gain and offset correction factors (also computed prior to compression) of

$$\frac{\hat{m}}{m_i} \quad \text{and} \quad \frac{1}{c} \left[ \hat{b} - \frac{\hat{m}}{m_i} (b_i) \right] \quad (12)$$

If we compress the images prior to computing the correction coefficients, we can model the compressed images similarly to the uncompressed images, i.e., as a linear model with additive noise (plots of pixel response vs. radiance level in the compressed flat field images averaged in the in-track direction show good linear behavior) :

$$\tilde{y}_{k,i} = (\tilde{m}_i x_k + \tilde{b}_i + \tilde{\mathbf{n}}) / c + \tilde{\mathbf{n}}_q \quad (13)$$

The correction factors for this case are then

$$\frac{\hat{m}}{\hat{m}_i} \quad \text{and} \quad \frac{1}{c} \left[ \hat{b} - \frac{\hat{m}}{\hat{m}_i} (\tilde{b}_i) \right] \quad (14)$$

These are matched sets, i.e., the correction factors computed on the uncompressed calibration images will perform best when applied to uncompressed flat field images while the correction factors computed on the compressed calibration images will perform best when applied to compressed flat field images. The worst performance will be observed in the unmatched case, i.e., when the correction factors are computed with the uncompressed calibration images and applied to compressed flat field images. These observations agree with the data.

Another question to answer about Figure 3 is why does the line-to-line standard deviation of the uncorrected image first increase and then decrease with decreasing bit rate? To help answer this question we plotted cross-track slices of the high and low uncorrected images at bit rates of 12, 2.4, 1.2, 0.6, and 0.4 bpp (see Figure 4). They were plotted with successive offsets between curves for easier viewing. The top curve represents 12 bpp and the bit rate decreases down to the bottom curve which represents 0.4 bpp. These plots show two trends: (1) the low amplitude noise tends to be increasingly spatially averaged (high frequency information was lost during the quantization process) for lower bit rates which reduces the line-to-line standard deviation, and (2) the high amplitude noise (which the quantizer decides should be preserved) tends to be amplified which increases the line-to-line standard deviation. At the higher bit rates, the second trend dominates, whereas at the lower bit rates, the first trend dominates, thus resulting in the behavior seen in Figure 3.

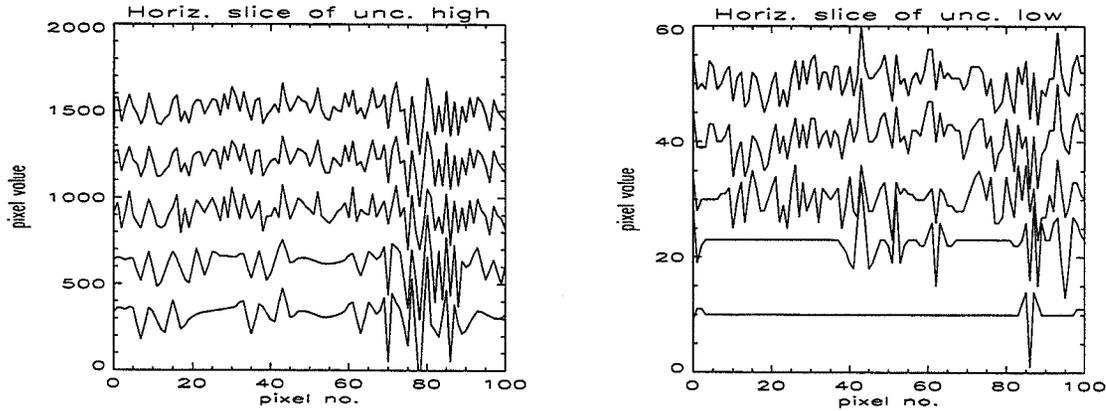
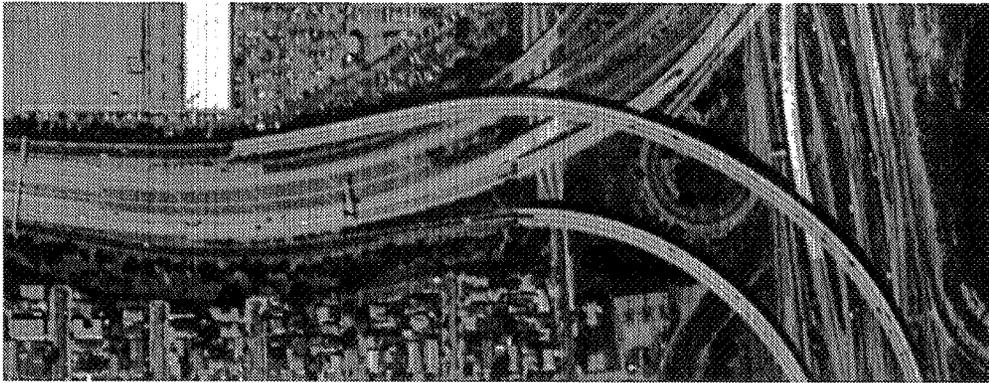


Figure 4. Cross-track slices of the uncorrected flat field image at high and low radiance levels for bit rates of 12, 2.4, 1.2, 0.6, and 0.4 bpp.

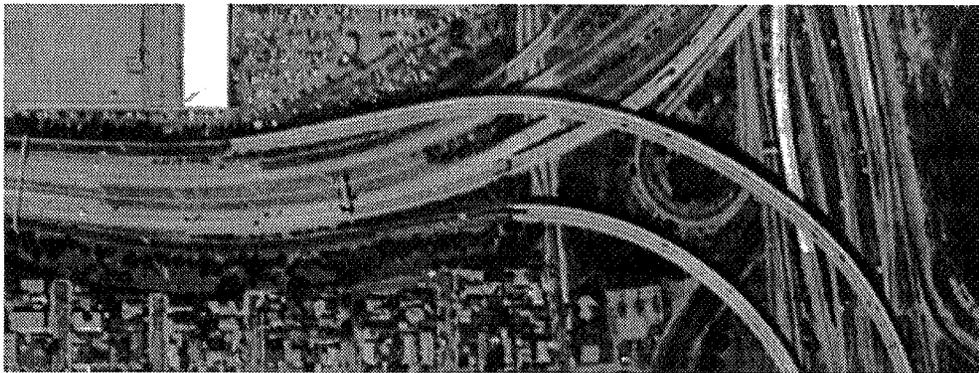
## 4.2 Correcting representative images

The raw image containing non-uniformities and the corrected images (for the *Prior*, *Mixed*, and *Post* cases) are shown in Figure 5 for a representative image. The striping effect due to pixel-to-pixel nonuniformity is very apparent in the uncorrected image. The correction using the *Prior* technique virtually eliminates all visible striping. Correction using the *Mixed* and *Post* techniques still exhibit a fair amount of striping.

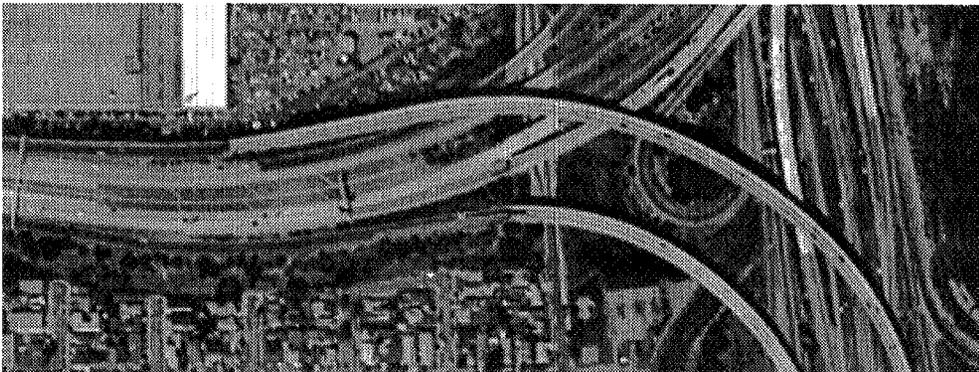
On a more quantitative note, it is desirable to compare the corrected images to the ideal corrected image (no compression). Figure 6 shows the mean squared error between the ideal and non-ideal corrected images vs. bit rate for the *Prior*, *Mixed*, and *Post* cases.



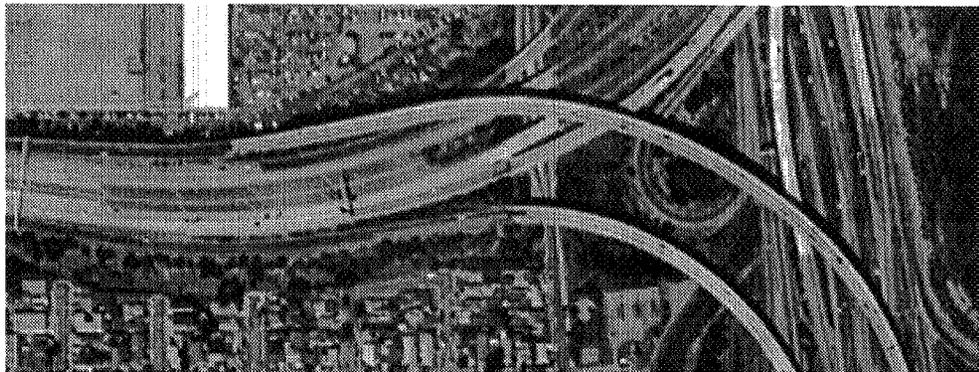
uncorrected



(i) Prior



(ii) Mixed



(iii) Post

Figure 5. Uncorrected and Prior-, Mixed-, and Post-Corrected Images for the Representative Image

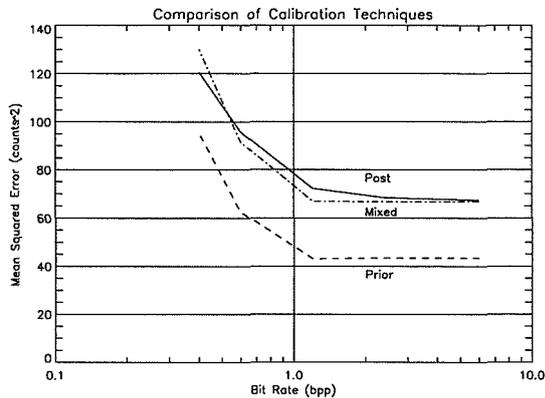


Figure 6. Comparison of Calibration Techniques for prior, mixed, and post algorithms

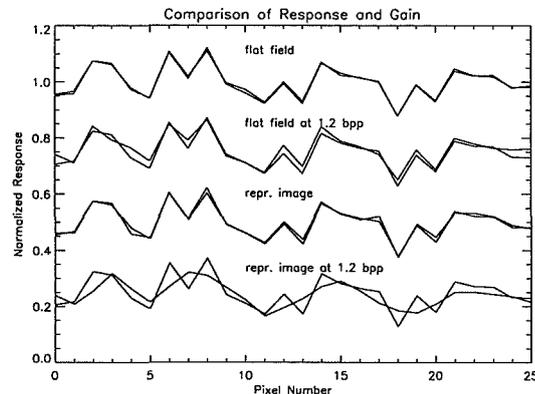


Figure 7. Comparison of pixel response and gain values for the flat field (uncompressed and compressed at 1.2 bpp) and representative (uncompressed and compressed at 1.2 bpp) images.

Figure 6 shows that the best performance is clearly obtained when the correction coefficients are both computed and applied prior to compression (there is an increase in RMSE of between 12% and 29%, depending on the bit rate, for the *Post* case as compared to the *Prior* case). This is because the *Prior* case is now the only matched case. Compression of flat field images behaves much differently than compression of representative images (frequency components in the flat field images will, in general, have different amounts of quantization than corresponding frequency components in the representative images due to different wavelet coefficient statistics), so the *Post* case is no longer a matched case. Therefore, the performance of the *Post* case will be worse than the *Prior* case for representative images. To illustrate this point, we plotted four pairs of curves, shown in Figure 7. One curve in each pair represents the pixel-to-pixel gains of the detector responses (note that pixel response is dominated by gain, not offset). The other four curves (one in each pair) represent a cross-track slice of the medium radiance flat field image, uncompressed (top curve) and compressed at 1.2 bpp (second curve); and a cross-track slice of the representative image in a uniform part of the image (white roof in the top left of image), uncompressed (third curve) and compressed at 1.2 bpp (bottom curve). The flat field images are normalized by the mean of the slice and the representative images are normalized by the mean of the slice as well as the ideal normalized value for each pixel. The bottom three pairs of curves are offset for easier viewing.

The first set of pairs for the flat field image, match very well, i.e., the normalized detector response is very close to the detector gains, as expected. The slight differences are due to detector offsets and noise. The second set of pairs for the compressed flat field image are still close, but not as close due to the effects of the compression. One can see that the effective gains of this equivalent system will be close to the original gains. The third set of curves for the representative image, are also very close to each other. Again, this simply illustrates that the detector response is dominated by gain and verifies that the gains on the flat field image are the same as the gains on the representative image. The fourth set of pairs, for the representative image compressed at 1.2 bpp, are not very close. The compression process on the representative image smoothed out the pixel-to-pixel variations. If we look at this smoother curve as having the same input radiance as the uncompressed representative image, then it is apparent that it has very different effective gains than the gains in the uncompressed system and even the gains computed in the compressed flat field system. Therefore, correction coefficients that were computed from either the uncompressed or compressed *flat field* images will not correct the compressed *representative* images as well as flat field images since the flat field and representative images have different effective gains.

## 5. Conclusions

Line scan sensors used in remote sensing systems exhibit striping artifacts due to detector-to-detector nonuniformities. These nonuniformities are generally characterized by a gain and offset for each detector in the array and are corrected via flat field calibration. Current high-end and next-generation remote sensing systems are demanding incorporation of high quality, low bit rate data compression onto the spacecraft. A good candidate to apply in this arena is wavelet-based data compression due to its high image quality at low bit rates with reasonable

computational complexity and soon-to-be-available hardware. This paper described our understanding of the effects of wavelet-based data compression on flat field calibration. Both corrections of flat field images and representative remote sensing imagery are evaluated. Three calibration/compression approaches were studied: (i) computing and applying correction coefficients prior to compression, (ii) computing correction coefficients prior to compression and applying correction coefficients after compression, and (iii) computing and applying correction coefficients after compression. Cases (i) and (ii) require a lossless or no compression mode for gathering calibration data and computing correction factors without compression. For case (i), where the correction factors are applied prior to the on-board compression, calibration hardware must be included on-board, whereas for cases (ii) and (iii), where the correction factors are applied after compression, the calibration hardware for applying these corrections may reside on the ground.

The previous section explained that approaches (i) and (iii) work well when correcting flat field images because the computation and application of the correction factors operate on matched sets of data, i.e., the computation and application of the corrections are either both applied to uncompressed data or both applied to compressed data at the same compression level. However, when trying to correct representative images using correction factors computed from the flat field calibration data, we observed that approach (i) performs better than approach (iii). There is an increase in RMS error of between 12% and 29% observed in case (iii) as compared to case (i). This is because approach (iii) is no longer a matched case due to the fact that compression of flat field images behaves differently than compression of representative imagery. If one chooses to implement approach (i) due to the improved performance provided, then the spacecraft must have: (a) a compression bypass option or a lossless compression option in calibration mode, and (b) additional on-board hardware to apply the gain and offset correction factors to every frame of collected imagery. If one chooses to implement approach (iii), then: (a) no compression bypass or lossless mode is required, and (b) all calibration can be performed on the ground. Clearly, there is a tradeoff between calibration performance and increased hardware/complexity requirements of the on-orbit system. Future work entails a system-level evaluation of this trade as well as continued development of data compression hardware for realization on next-generation space-based remote sensing systems.

## References

1. Internal study, Ball Aerospace, December 1994
2. Stephane G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989.
3. Marc Antonini, Michel Barlaud, Pierre Mathieu, Ingrid Daubechies, "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, Vol. 1, No. 2, April 1992.
4. Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., New York, 1991.
5. Roger C. Wood, "On Optimum Quantization", IEEE Transactions on Information Theory, Vol. IT-15, No. 2, March 1969.
6. J.M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", IEEE Transactions on Signal Processing, Vol. 41, December 1993
7. Z. Xiong, K. Ramchandran, M.T. Orchard, "Joint Optimization of Scalar and Tree-Structured Quantization of Wavelet Image Decompositions", Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, 1993

519-82  
013184

close

268908  
10P.

# PERCEPTUALLY LOSSLESS WAVELET COMPRESSION

ANDREW B. WATSON<sup>1</sup>, GLORIA Y. YANG<sup>2</sup>,  
JOSHUA A. SOLOMON<sup>1</sup> & JOHN VILLASENOR<sup>2</sup>

<sup>1</sup>NASA AMES RESEARCH CENTER, MOFFETT FIELD, CA,  
<sup>2</sup>DEPARTMENT OF ELECTRICAL ENGINEERING, UCLA

## ABSTRACT

The Discrete Wavelet Transform (DWT) decomposes an image into bands that vary in spatial frequency and orientation. It is widely used for image compression. Measures of the visibility of DWT quantization errors are required to achieve optimal compression. Uniform quantization of a single band of coefficients results in an artifact that is the sum of a lattice of random amplitude basis functions of the corresponding DWT synthesis filter, which we call *DWT uniform quantization noise*. We measured visual detection thresholds for samples of DWT uniform quantization noise in Y, Cb, and Cr color channels.

The spatial frequency of a wavelet is  $r 2^{-L}$ , where  $r$  is display visual resolution in pixels/degree, and  $L$  is the wavelet level. Amplitude thresholds increase rapidly with spatial frequency. Thresholds also increase from Y to Cr to Cb, and with orientation from low-pass to horizontal/vertical to diagonal.

We propose a mathematical model for DWT noise detection thresholds that is a function of level, orientation, and display visual resolution. This allows calculation of a "perceptually lossless" quantization matrix for which all errors are in theory below the visual threshold. The model may also be used as the basis for adaptive quantization schemes.

## I. INTRODUCTION

NASA's extensive image compression requirements may be met in part by the use of wavelet techniques [3]. Wavelets form a large class of signal and image transforms, generally characterized by decomposition into a set of self-similar signals that vary in scale and (in 2D) orientation [4]. The Discrete Wavelet Transform (DWT) is a particular member of this family which operates on discrete sequences, and which has proven to be an effective tool in image compression [1, 2, 7, 8]. In a typical compression application, an image is subjected to a two-dimensional DWT whose coefficients are then quantized and entropy coded.

DWT compression is lossy, and depends for its success upon the invisibility of the artifacts. The purpose of this paper is to provide information on the visibility of DWT artifacts, and to show how it may be used in the design of wavelet compression

systems. In this research we have generally followed earlier work on the Discrete Cosine Transform [5, 9, 11], with some important differences that will be discussed below.

## II. BACKGROUND

### A. Discrete Wavelet Transform

Figure 1 illustrates the elements of a one-dimensional, two-channel perfect-reconstruction filter bank. The input discrete sequence  $x$  is convolved with high-pass and low-pass analysis filters  $a_H$  and  $a_L$ , and each result is down-sampled by two, yielding the transformed signals  $x_H$  and  $x_L$ . The signal is reconstructed through up-sampling and convolution with high and low synthesis filters  $s_H$  and  $s_L$ . For properly designed filters, the signal  $x$  is reconstructed exactly ( $y=x$ ).

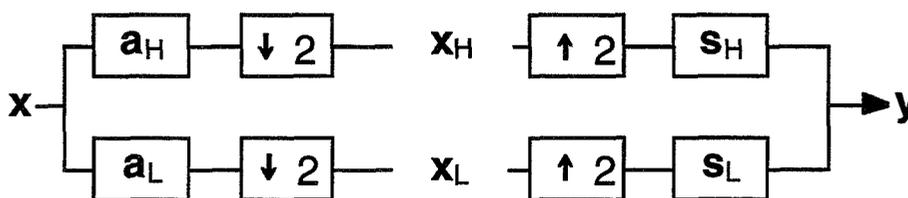


Figure 1. A two-channel perfect-reconstruction filter bank.

A DWT is obtained by further decomposing the low-pass signal  $x_L$  by means of a second identical pair of analysis filters, and, upon reconstruction, synthesis filters, as shown in Figure 2. This process may be repeated, and the number of such stages defines the *level* of the transform.

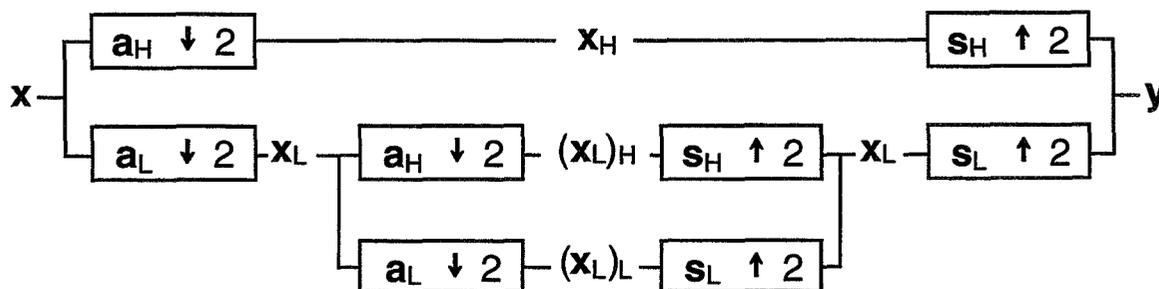


Figure 2. Two-level 1D Discrete Wavelet Transform.

With two-dimensional signals such as images, the DWT is typically applied in a separable fashion to each dimension. Now each filter is two-dimensional, with the subscript indicating the separable horizontal and vertical components, and the downsampling operation is applied in both dimensions. As in the one-dimensional

case, the process may be repeated a number of times, in each case by applying the component  $x_{LL}$  as input to a second stage of identical filters.

Here we adopt the term *level* to describe the number of 2D filter stages a component has passed through, and the term *orientation* to identify the four possible combinations of low-pass and high-pass filtering the signal has experienced. We index orientations as follows: {1,2,3,4} = {LL,HL,HH,LH} where low and high are in the order horizontal-vertical. Each combination of level and orientation {L,O} specifies a single *band*. For the purpose of this research we selected the 9-7 tap Antonini biorthogonal filters [1].

#### D. DWT Quantization Matrix

Compression of the DWT is achieved by quantization and entropy coding of the DWT coefficients. Typically a uniform quantizer is used, implemented by division by a factor  $Q$  and rounding to the nearest integer. The factor  $Q$  may differ for different bands. It will be convenient to speak of a *quantization matrix* to refer to a set of quantization factors corresponding to a particular matrix of levels and orientations.

A particular quantization factor  $Q$  in one band will result in coefficient errors in that band that are approximately uniformly distributed over the interval  $[-Q/2, Q/2]$ . The error image will be the sum of a lattice of basis functions with amplitudes proportional to the corresponding coefficient errors. To predict the visibility of the error due to a particular  $Q$ , we must measure the visibility thresholds for individual basis functions and error ensembles.

#### E. Display Visual Resolution

Visibility of DWT basis functions will depend upon display visual resolution in pixels/degree. Given a viewing distance  $v$  in cm and a display resolution  $d$  in pixels/cm, the effective display visual resolution (DVR)  $r$  in pixels/degree of visual angle is

$$r = d v \tan(\pi/180) \approx d v \pi/180 \quad (1)$$

#### F. Wavelet Level and Spatial Frequency

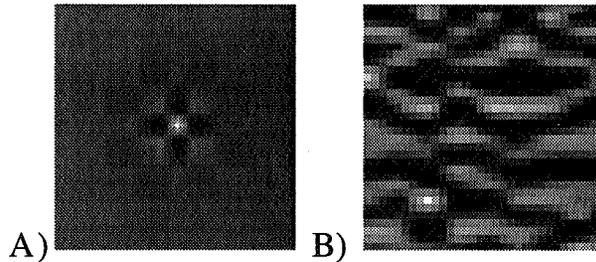
A single basis function encompasses a band of spatial frequencies. We take the Nyquist frequency of the display resolution as the nominal spatial frequency of the first DWT level, and the frequency of each subsequent level will be reduced by a factor of two. Thus for a display resolution of  $r$  pixels/degree, the spatial frequency  $f$  of level  $L$  will be

$$f = r 2^{-L} \text{ cycles/degree} \quad (2)$$

### III. METHODS

Stimuli were modulations of either Y, Cb, or Cr channels of a color image. These produce images that are black/white, yellow/purple, and red/green respectively. All modulations were added to an otherwise uniform (YCbCr = {128,0,0}) image of size 1024x1024 pixels. Modulations were either single DWT basis functions or samples of DWT uniform quantization noise, as shown in Figure 3. In either case, individual modulation images were scaled to produce amplitudes in the range of [0,126]. The amplitude of the modulated signal is our measure of stimulus intensity. The modulated channel, plus the two remaining unmodulated channels, were then transformed to R'G'B' using the rule

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.3441 & -0.7141 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b - 128 \\ C_r - 128 \end{bmatrix} \quad (3)$$



**Figure 3. Example stimuli: a) basis function , b) DWT noise.**

To vary the display visual resolution we pixel-replicated the stimuli by factors of 1 (no replication), 2, or 4 in both dimensions, yielding visual resolutions of 64, 32, and 16 pixels/degree. For all stimuli, the duration was 16 frames in duration at a frame rate of 60 Hz, or 267 msec. The time course was a Gaussian  $e^{-\pi(f/8)^2}$  where  $f$  is in frames. To measure detection thresholds for individual stimuli we used a two-alternative forced-choice (2AFC) procedure. Three observers took part in the experiments.

### IV. RESULTS

#### A. Effect of DWT Level

Thresholds for display resolutions of 16, 32, and 64 pixels/degree, all at orientation 4, are shown in Figure 4. In general thresholds are largely unaffected by resolution, once they are expressed as a function of spatial frequency in cycles/degree. Additional data from observer sfl confirm this observation.

#### B. Single Basis Functions vs Noise Images

Figure 5 plots the difference between log thresholds for single basis functions and for noise. As expected, basis function thresholds are uniformly higher than noise thresholds.

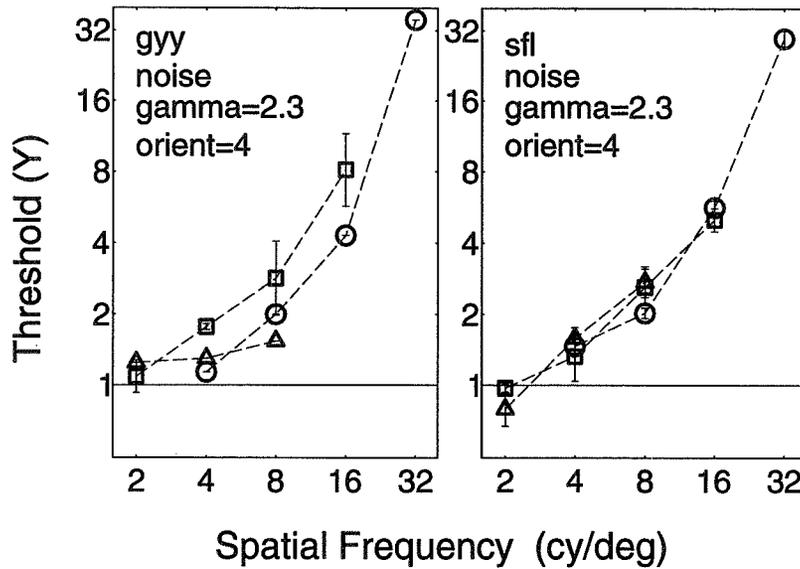


Figure 4. Thresholds at display resolutions of 16 (triangles) 32 (squares) and 64 pixels/degree (circles), for orientation 4.

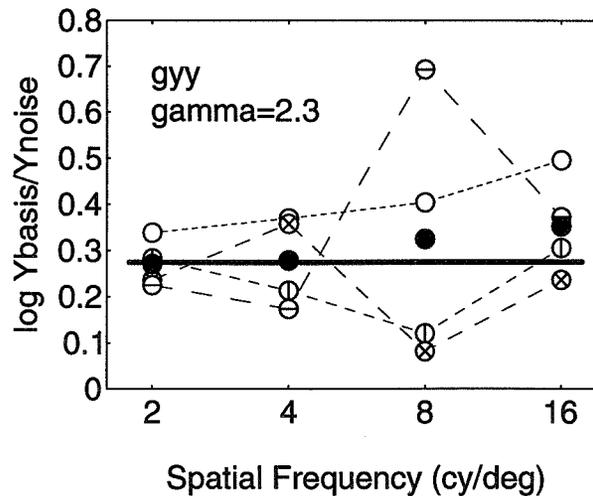


Figure 5. Difference between log thresholds for DWT noise and basis functions. Open symbols show data for individual orientations, solid symbols are means. Heavy line is the probability summation prediction.

We considered a simple spatial probability summation model to account quantitatively for the difference between basis function and noise thresholds [6, 10]. In this context, this model asserts that the Minkowski sum over individual basis functions amplitudes is equal for all basis functions ensembles at threshold. This prediction is plotted as the horizontal line in Figure 5. It is clear that probability summation provides an excellent account of the difference between basis and noise thresholds.

### C. Grayscale Model

One model that provides a reasonable fit to thresholds for grayscale DWT is

$$\log Y = \log a + k (\log f - \log g_0 f_0)^2 \quad (4)$$

This is a parabola in  $\log Y$  vs  $\log f$  coordinates, with a minimum at  $g_0 f_0$  and a width of  $k^{-2}$ . The term  $g_0$  shifts the minimum by an amount that is a function of orientation, and where  $g_2 = g_4 = 1$ . The term  $a$  defines the minimum threshold. The optimized parameters and rms error (of  $\log Y$ ) are given in the first line of Table 3. The fit is shown in Figure 6.

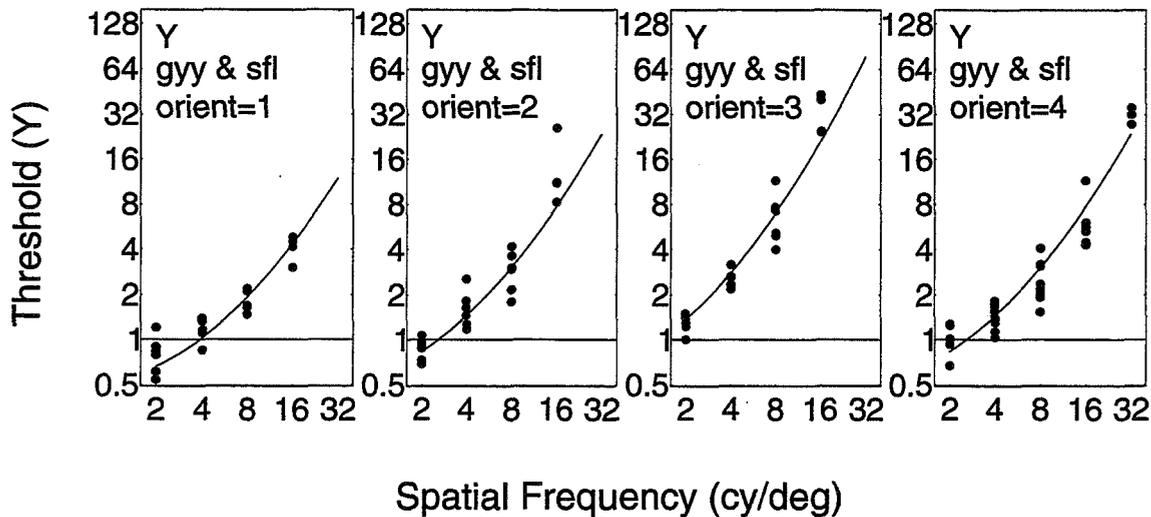


Figure 6. Fit of the threshold model to grayscale data of gyy and sfl.

### D. Color Results and Model

Figure 7 shows results for observers sfl and abw at orientations 1, 3, and 4 for a DWT noise pattern and display gamma of 2.3. We have applied the same model used for grayscale thresholds to the color thresholds in Figure 7. The solid curves therein show the various fits. The parameters are in Table 3, along with the Y parameters.

Color	Observer	rms	a	k	$f_0$	$g_1$	$g_3$
Y	gyy & sfl	0.134	0.495	0.466	0.401	1.501	0.534
Cr	sfl	0.113	0.944	0.521	0.404	1.868	0.516
	abw	0.127	0.803	0.539	0.328	2.017	0.589
Cb	sfl	0.145	1.633	0.353	0.209	1.520	0.502
	abw	0.093	2.432	0.520	0.269	1.706	0.599

Table 3. Parameters for DWT YCbCr threshold model.

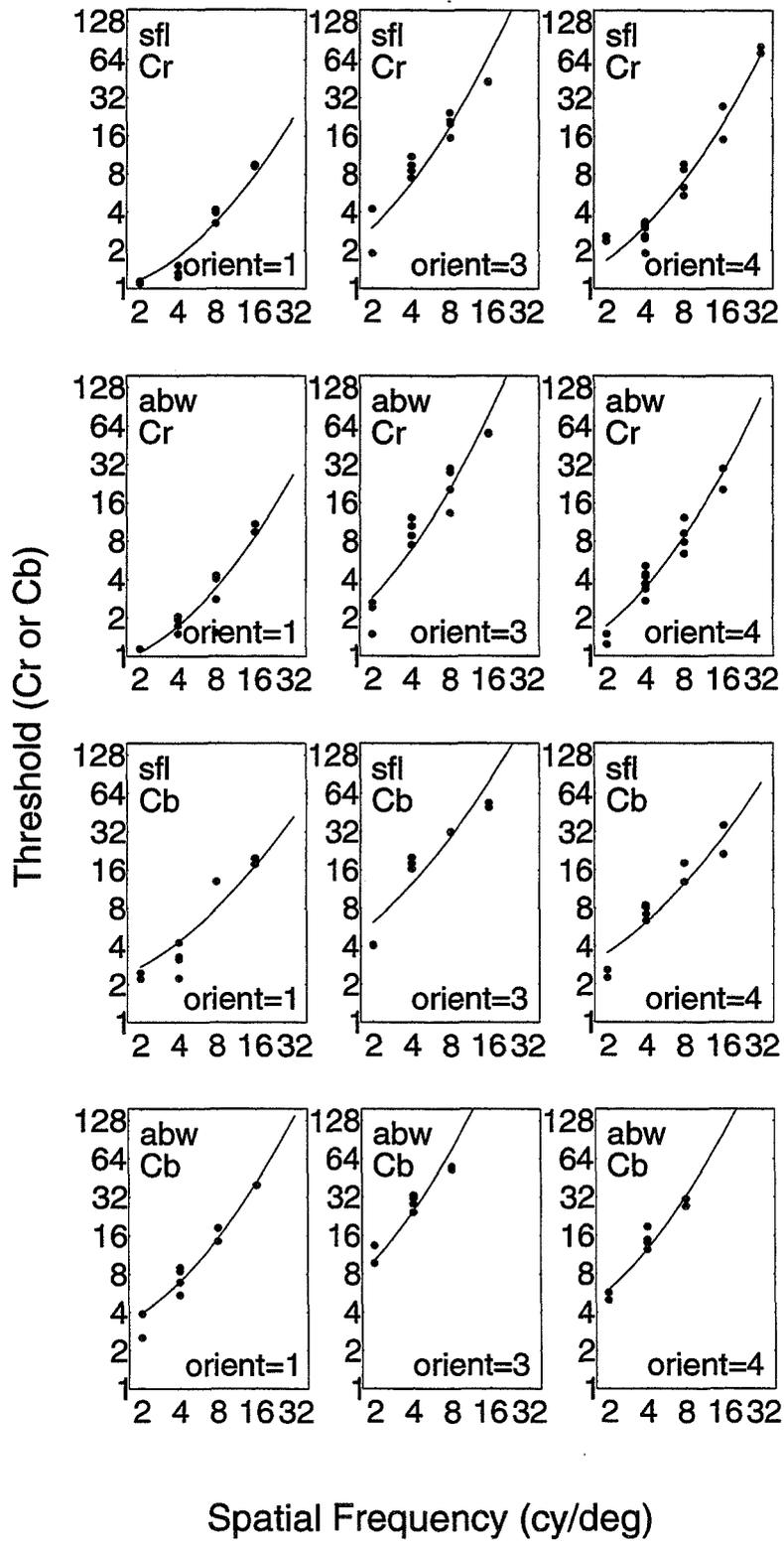


Figure 7. Thresholds for DWT uniform noise in Cr and Cb channels.

## V. QUANTIZATION MATRICES

We now use the model developed above to compute a "perceptually lossless" quantization matrix, by using a quantization factor for each level and orientation that will result in a quantization error that is just at the threshold of visibility. For uniform quantization and a given quantization factor  $Q$ , the largest possible coefficient error is  $Q/2$ . The amplitude of the resulting noise is approximately  $A_{L,O} Q/2$ . Thus we set

$$Q_{L,O} = 2 Y_{L,O} / A_{L,O} . \tag{5}$$

The basis function amplitudes  $A_{L,O}$  are given for six levels in Table 4.

Orientation	Level					
	1	2	3	4	5	6
1	0.62171	0.345374	0.18004	0.0914012	0.0459435	0.0230128
2	0.672341	0.413174	0.227267	0.117925	0.0597584	0.0300184
3	0.727095	0.494284	0.286881	0.152145	0.0777274	0.0391565
4	0.672341	0.413174	0.227267	0.117925	0.0597584	0.0300184

**Table 4. Basis function amplitudes  $A_{L,O}$  for a six-level Antonini DWT.**

Combining (4), (5), and (2),

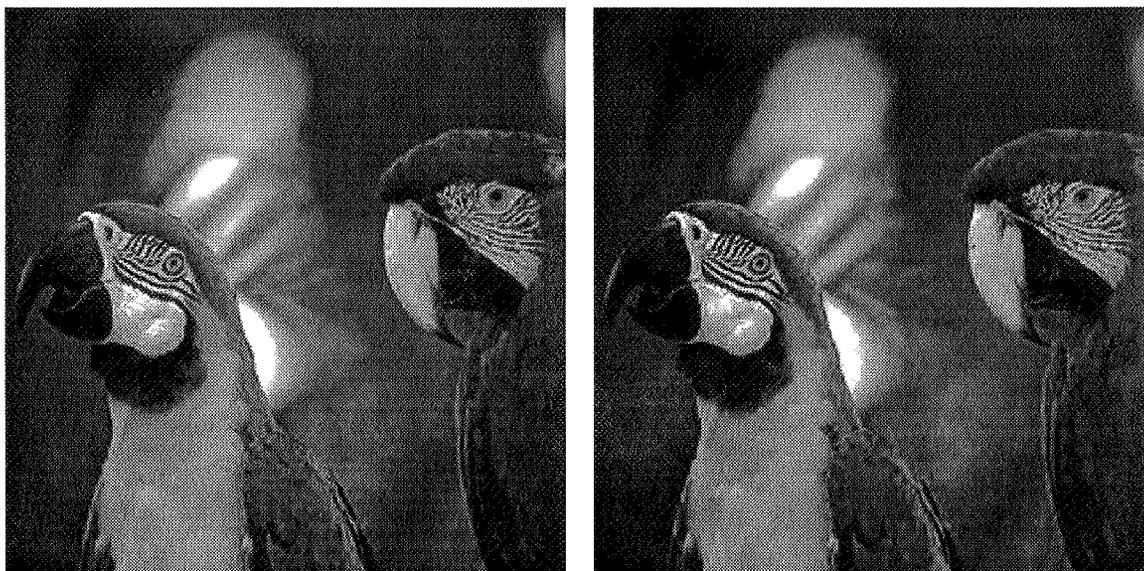
$$Q_{L,O} = \frac{2}{A_{L,O}} a 10^{k \left( \log \frac{2^L f_0 g_0}{r} \right)^2} \tag{6}$$

Table 5 shows example matrices computed from this formula.

Color	Orientation	Level			
		1	2	3	4
Y	1	14.049	11.106	11.363	14.5
	2	23.028	14.685	12.707	14.156
	3	58.756	28.408	19.54	17.864
	4	23.028	14.685	12.707	14.156
Cb	1	55.249	46.559	48.45	59.988
	2	86.789	60.485	54.571	60.476
	3	215.84	117.45	86.737	81.231
	4	86.789	60.485	54.571	60.476
Cr	1	25.044	19.282	19.665	25.597
	2	60.019	34.335	27.276	28.55
	3	184.64	77.569	47.441	39.468
	4	60.019	34.335	27.276	28.55

**Table 5. Quantization factors for four-level Antonini DWT for  $r=32$  pixel/degree.**

Figure 8 shows an example image compressed using the quantization matrix of Table 6, and twice that matrix. Only the grayscale image is shown in this printed paper, though the actual compression was done on a color image. Viewed from the appropriate distance (24 inches, approximately arm's length) the quantization artifacts should be invisible for the left image, and visible for the right. Using typical entropy coding techniques, the resulting bitrates for these two examples are 1.05 and 0.67 bits/pixel.



**Figure 8. Image compressed with perceptually lossless DWT quantization matrix (left) and twice that matrix (right). Image dimensions are 256x256 pixels. Quantization matrix is designed for a viewing distance of 24 inches. Actual images are in color; only grayscale is shown here.**

## IX. CONCLUSIONS

We have measured visual thresholds for samples of uniform quantization noise of a DWT based on the Antonini wavelet. Thresholds were collected for gamma-corrected signals in the three channels of the YCbCr color space. We propose a mathematical model for the thresholds, which may be used to design a simple "perceptually lossless" quantization matrix, or which may be used to weight quantization errors or masking backgrounds in more elaborate adaptive quantization schemes. These perceptual data, models, and methods may enhance the performance of wavelet compression schemes.

## ACKNOWLEDGMENTS

This work was supported by NASA Grant 199-06-12-39 to Andrew Watson from the Life and Biomedical Sciences and Applications Division. We thank Albert J. Ahumada and Heidi A. Peterson for useful discussions.

## REFERENCES

- [1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205-220, 1992.
- [2] T. Hopper, C. Brislawn, and J. Bradley, "WSQ grey-scale fingerprint image compression specification, Version 2.0," Criminal Justice Information Services, Federal Bureau of Investigation, Washington, DC. February 1993.
- [3] A. Jaworski, "Earth Observing System (EOS) Data and Information System (DIS) software interface standards," AIAA/NASA Second International Symposium on Space Information Systems, Proceedings vol. AIAA-90-5075, 1990.
- [4] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 2091-2110, 1989.
- [5] H. Peterson, A. J. Ahumada, Jr., and A. Watson, "An Improved Detection Model for DCT Coefficient Quantization," *SPIE Proceedings*, vol. 1913, pp. 191-201, 1993.
- [6] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vision Research*, vol. 21, pp. 409-418, 1981.
- [7] J. M. Shapiro, "An embedded wavelet hierarchical image coder," ICASSP, Proceedings , pp. 657-660, 1992.
- [8] J. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for efficient image compression," *IEEE Transactions on Image Processing*, vol. 4, pp. 1053-1060, 1995.
- [9] A. B. Watson, "DCT quantization matrices visually optimized for individual images," Human Vision, Visual Processing, and Digital Display IV, Proceedings vol. 1913, pp. 202-216, 1993.
- [10] A. B. Watson, "Probability summation over time," *Vision Research*, vol. 19, pp. 515-522, 1979.
- [11] A. B. Watson, J. A. Solomon, and A. J. Ahumada, Jr., "The visibility of DCT basis functions: effects of display resolution," in *Proceedings Data Compression Conference*, J. A. Storer and M. Cohn, Eds. Los Alamitos, CA: IEEE Computer Society Press, 1994, pp. 371-379.