Hyperspectral Image Compression with Optimization for Spectral Analysis

Kameron Romines

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2006

Program Authorized to Offer Degree:
Institute of Technology – Tacoma

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Kameron Romines

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Edwin Hong

_____

Donald Chinn

Date: _____

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Kameron Romines

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Edwin Hong

_____

Donald Chinn

Date: _____

In presenting this thesis in partial fulfillment of the requirements for a master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purposes or by any means shall not be allowed without my written permission.

Signature _____

Date _____

University of Washington

Abstract

Hyperspectral Image Compression with Optimization for Spectral Analysis

Kameron Romines

Chair of the Supervisory Committee:
Assistant Professor Edwin Hong
Institute of Technology

Hyperspectral imaging is of interest in a large number of remote sensing applications, such as geology and pollution monitoring, in order to detect and analyze surface and atmospheric composition. The processing of these images, called spectral analysis, allows for the identification of the specific mineralogical and agricultural elements which compose an image. The research presented in this paper takes advantage of details specific to this processing in order to maximize the ability of compression algorithms to operate on the image with minimal loss in image utility. The research begins with the recommendation of new error determination utilities which incorporate spectral analysis techniques in order to model real usage, and then compares the results from these utilities with the commonly used PSNR and MAE error metrics. Then, it continues by building on the results of these utilities with the recommendation of a modification to a commonly used compression algorithm, 3D-SPIHT. Results of this modification show an improvement in the error rate as reported by these utilities, indicating an increase in the ability to analyze hyperspectral images which have been compressed.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

Introduction

Hyperspectral images are widely used in a number of civilian and military applications. The images are acquired from plane or satellite borne spectrometers and cover large tracts of the Earth's surface. Through the analysis of the spectrum of reflected light present in these images, it is possible to identify what materials are present on the land and in the atmosphere. This information has been used for such varied purposes as environmental studies, military surveillance and the analysis and location of mineral deposits.

Hyperspectral images are composed of hundreds of narrow and contiguous bands of data covering a large spectrum of reflected light. Conventional cameras are designed to record data in coarse bands of red, green, and blue, while hyperspectral imagers record much finer wavelengths and with a range far into the ultraviolet and infrared. This spectral information recorded for each physical location in a hyperspectral image can then be compared to known spectral profiles in a library in order to identify material composition of the photographed region. These images are manipulated as a three dimensional cube of data, with two spatial dimensions and the depth dimension containing the spectral signature.

There are a number of different sources for hyperspectral images. The most commonly available are probably those from AVIRIS (Airborne Visible InfraRed Imaging Spectrometer), an instrument which has been flown by NASA over much of the US, Canada, and Europe. Images collected by AVIRIS are quite large, with approximately 140 MB of data for every 10km of flight – or about 16 GB for a day's work (AVIRIS [online]). The Hyperion imager carried on the EO-1 (Earth Observing-1) satellite is also a common source of hyperspectral data. Image compression is particularly important for this application, where the images must be compressed and sent over a limited bandwidth carrier before analysis can take place.

In image compression, there are two primary categories of algorithms: lossless and lossy. Lossless image compression, in which the original image can be reconstructed exactly from the compressed image, is the ideal. Unfortunately, in practical applications this type of image compression can generally only compress images by 2 to 5 times (Keränen 2003). Lossy compression schemes discard some amount of data in an image, but can achieve much higher compression rates – indeed, with many algorithms, the user can choose exactly how large they would like the resulting compressed image to be. The difference between the original image and one reconstructed from lossily compressed data constitutes the compression error, and different applications for the images have different levels of tolerance for amount and type of introduced error.

In Chapter I of this paper, the application of compression to hyperspectral images is discussed, and the basis for this area of research is explored. In Chapter II, the experimentation methodology is covered in detail in order to allow recreation of the discussed results. Chapter III discusses modifications which were made to one compression algorithm, 3D-SPIHT, in order to take advantage of the findings in this paper, and Chapter IV contains the analysis of results. The Appendix contains additional information on the tools and algorithms used for this research as well as definitions for terminology (*italicized* on first use) which are not explained in the body of the text.

Chapter I: Purpose

Problem Statement

The most commonly used metrics for determining error in a lossy image are the Peak Signal to Noise Ratio (PSNR) and, increasingly, Maximum Absolute Error (MAE). These metrics are general purpose, and give a good overall indication of how closely data points in a reconstructed image are to the corresponding points in the original. Unfortunately, these metrics do not necessarily correlate well to the actual ability of the image viewer to process that data, such as in the spectral analysis of hyperspectral images. Further, most compression algorithms, such as 3D-SPIHT, are designed to maximize quality as measured by PSNR, which means that images compressed with these routines may have a greater error when used for actual classification purposes than is necessary.

PSNR is computed from the mean squared error and gives a good estimate of overall image quality (higher values are better), while MAE represents the single largest difference between a reconstructed and an original data point (smaller values are better). Higher PSNR values may imply a closer resemblance between a reconstructed and the original image, but the metric has only an approximate relationship with errors as perceived by humans (Salomon 270). Dragotti notes this effect with regards to compression of hyperspectral images with 3D-SPIHT. He found that subjective comparisons between reconstructed images at moderate compression rates and the originals resulted in very different quality ranking than that produced with mean squared error based metrics such as PSNR (Dragotti et. al. 2000). It is expected that this problem extends to various types of specialized usage, especially those which favor some types of error over others – such as is the case with spectral analysis. Actual correlation between PSNR values and the ability to perform spectral analysis has not been well studied.

The original SPIHT algorithm was intended for two dimensional lossy image compression. It was extended to operate on three dimensional data as 3D-SPIHT (Kim

and Pearlman 1997), which has been used in a number of applications such as video encoding, multispectral image compression, and hyperspectral image compression. In video, the dimensions are height, width, and time; for the images, the third dimension is that of multiple wavelengths of reflected spectral information. 3D-SPIHT is able to exploit the high degree of similarity between layers in this third dimension to achieve much higher compression rates than would be possible if each layer were encoded independently using a two dimensional technique, such as SPIHT or the commonly used JPEG standard. As is commonly the case with techniques used in multiple applications, 3D-SPIHT is designed to produce the best average results – something that PSNR is designed to measure.

The PSNR error metric and the 3D-SPIHT algorithm are both techniques with application to a number of different data sets. As such, they foster familiarity and encourage portability to new applications. However, at least in the case of hyperspectral imaging, it should be expected that more specialized techniques will result in greater utility. Error measuring utilities that accurately model how an image will actually be used (e.g., in spectral analysis) are likely to be more useful in reporting the actual implications involved with the lossy compression of these images than the generic PSNR and MAE metrics. Likewise, modifications to 3D-SPIHT which prioritize the data with greatest impact to the various spectral analysis techniques can be expected to improve the usability of the reconstructed images.

Purpose of Study

Hyperspectral images are typically quite large, as they are essentially a compilation of hundreds of images with each containing data for a specific wavelength range. A single AVIRIS pass can generate in excess of 600 Megabytes of data (AVIRIS [online]). While this may not be problematic in instances where storage for analysis is immediately available for such large amounts of data, it is a problem when the data must be transmitted before analysis can take place. For example, the EO-1 satellite has approximately 3 – 3.5 Mbps of measured effective bandwidth (Johnson 2004). However,

the onboard Hyperion Imager can take over 20MB of uncompressed data every second (242 bands * 256 pixels/line * 223.4 lines/second * 1.5 Bytes/pixel) (Barry 2001), which is compressed via lossless *HDF* compression to approximately 10MB (EO-1 [online]). Clearly, even if all the available bandwidth of the EO-1 satellite is devoted to transmission of Hyperion data, only a small fraction of feasibly available data can be retrieved.

There may be additional gains to be had through the use of a progressive algorithm such as 3D-SPIHT over HDF even if quantity of data is not an issue. In some cases it may be critical to obtain the best possible information as soon as possible; 3D-SPIHT would allow for continual spectral analysis of received data with gradual refinement of detected spectra as more data is obtained. Even if no data truncation is performed on the SPIHT data stream (see the 3D-SPIHT discussion in the Appendix for more information) to limit information loss as much as possible, this progressive transmission may be advantageous. The can be contrasted with conventional compression algorithms in which the entire data set must be received before any of it can be decompressed, such as is the case with the HDF data.

During this course of study, some automated spectral analysis methodologies were studied and incorporated into utilities for the purpose of measuring error introduced by compression. This was expected to provide a mechanism for best examining the implications of compression on spectral analysis. Results from the new error utilities were compared with the conventional PSNR and MAE error metrics to see how well those metrics reflect actual quality for this purpose. Additionally, it was conjectured that a bias for information in the spectral dimension of hyperspectral images during compression would improve the ability to identify materials in the reconstructed images via spectral analysis. As any such introduced bias would be expected to lower the average quality of information in the image, but improve that in the spectral dimension, it was expected that PSNR and MAE would suffer as a result of this sort of modification – but that these new utilities would accurately reflect any gains in the ability to utilize the

reconstructed images. This paper attempts to address these two issues and to specify exactly what kind of gains (if any) might be expected through this sort of technique.

Chapter II: Methodology

Image Preparation

Overview

Images used for this work come from the freely available AVIRIS 1997 runs, which can be downloaded at http://aviris.jpl.nasa.gov/html/aviris.freedata.html. For this research, scene 1 reflectance data from both the Cuprite and Jasper Ridge sites was used (see Table 1). Images were loaded into the commercial software package ENVI (see Appendix A) for viewing and manipulation; ENVI tooling was used to convert the data for use with QccPack (also in Appendix A), which provides an implementation of 3D-SPIHT.

Table 1: Test images

Image	Site	Date	Source
1	Cuprite, NV	June 6, 1997	AVIRIS flight f970619t01p02_r02, Scene 1
2	Jasper Ridge, CA	April 3, 1997	AVIRIS flight f970403t01p02_r03, Scene 1

Selection

The AVIRIS images are available as a series of scenes which can be compiled together in order to obtain one single image of the entire flight line. For the purposes of this testing, only a single scene was chosen in each instance in order to minimize memory and run time constraints. The AVIRIS images are available as both radiance and reflectance data; radiance images contain the data as recorded by the sensors, while reflectance data has been corrected for sensor and atmospheric conditions. The corrected reflectance data was used in this testing in order to best match the available spectral libraries. It is expected that findings presented in this paper will translate to the compression of the radiance data with correction after decompression, but this has not been verified.

Importing into ENVI

The AVIRIS images (RFL for reflectance and IMG for radiance data) are stored in a Band Interleaved by Pixel (BIP) format. The pixels are expressed as 16-bit integers. ENVI is able to read these files directly, but the user is required to enter the layout characteristics (interleave method, pixel size, and file dimensions). All of the AVIRIS

scenes consist of 614 samples, 512 lines, and 224 bands except for the last scene in each image series, which may have fewer lines. Wavelength and *Full Width at Half Maximum* (FWHM) data can be loaded into ENVI from the included spectral information file (SPC), and a wavelength unit must be specified (nanometers, in this case).

Resizing

For the purposes of this paper, the tested images were resized such that all dimensions were in multiples of 64. This was done for simplicity in working with QccPack, which is open source software which implements the 3D-SPIHT algorithm. QccPack requires input images to have dimensions divisible by the number of sub-bands for that dimension (a sub-band in this case is a subset data cube compartmentalized by the wavelet transformation step of 3D-SPIHT). For example, in this work the default *dyadic decomposition* of 5 was used, which results in 64 ($2^5 * 2$) sub-bands. QccPack does have the ability to work around this issue through the use of masking images which define the data of interest in an image, but for the purpose of this work it was found appropriate to resize the images.

Both of the used images started as a single AVIRIS scene, with dimensions of 614 x 512 x 224 as noted above. Bands (images in the depth dimension containing the most granular wavelength range) with no data as indicated by the ENVI statistics tool were removed first, and then the trailing bands of the image were removed to achieve 192 bands. The number of samples (the horizontal spatial dimension) was arbitrarily reduced to 576 by trimming the rightmost sample data from the images.

Conversion

QccPack requires images to be in the ICB format, which specifies a Band Sequential (BSQ) pixel layout instead of BIP. Additionally, all pixels must be expressed in the 4 byte float data type. ENVI includes pixel layout conversion and data type promotion tools, the latter of which can be achieved through application of the Apply Gain and Offset feature with a 1.0 gain and Floating Point as the output data type. After these steps, an ICB header was added manually to the ICB format (see Table 2).

Table 2: ICB header

ICB<QccPack Version>
<samples> <lines> <bands>
<minimum pixel value> <maximum pixel value>

Also, the ENVI generated header was manually edited to specify an offset equivalent to the number of bytes in the header. These steps result images which can be loaded by both the QccPack and ENVI tool sets.

PSNR and MAE Metrics

PSNR is a common error metric for quantifying error introduced in a lossy image. It is often chosen for its simplicity and generic applicability to quality comparisons. This calculation is based on the mean squared error for all pixels in an image, as compared to the original. PSNR is expressed in decibels (dB), and has no unit; PSNR values are not meaningful by themselves and are only useful for comparing the amount of loss between images. Typical values for PSNR range from 20 to 40 dB, with higher values representing better quality.

PSNR is derived from the MSE (Mean Squared Error). If p is a source image and P is an image reconstructed by reconstructing the compressed version of p at a given compression rate, MSE and PSNR are calculated in this document as follows

$$MSE = \frac{\sum [p(i, j) - P(i, j)]^2}{N}$$

$$PSNR = 20 \log_{10} (\frac{R}{\sqrt{MSE}})$$

N is the total number of voxels (a voxel is a “volume element”; i.e., a 3D pixel) in the image and R is typically the range of possible pixel values. In this document, 256 is used for R in most instances, except for calculation of the Matched Filtering error where 2 is used. Note that for all results presented here, higher PSNR (and Matched Filtering PSNR, below) values indicate better results.

Maximum Absolute Error (MAE) is simply the largest absolute difference between a value at a location in the reconstructed and the value at the same location in the original image. This indicates the worst case scenario for the reconstructed image. MAE does not necessarily scale smoothly with compression rate for an algorithm such as 3D-SPIHT, which seeks to maximize PSNR, but it does trend towards improvement over a large enough range of compression rates.

SAM-based Error Utility

Spectral Angle Mapping (SAM) is an automated classification technique for classifying the spectra in a hyperspectral image as some type of known spectra. These known spectra might come from a predefined spectral library, such as those supplied by the United States Geological Service, or from user-defined areas of interest on the image itself, called Regions of Interest (ROI). Within ENVI, SAM works on reflectance data – that is, data which has been corrected by a gain factor based upon factors such as topography and shadows in an image. The reference spectra are plotted as vectors in *n*-dimensional space, where *n* corresponds to the number of bands (wavelengths) of which the spectrum is composed (ENVI documentation). The unknown spectra in the image are then plotted in the space as well, and are classified as the reference spectra with the smallest angle between the two vectors. A user-specified error threshold is also specified (0.10 radians for these tests, unless otherwise noted); if this threshold is exceeded for the match, then the spectra is marked as "unclassified."

The SAM technique results in a two dimensional image map, mapping locations of identified materials (as well as any that were not identified, which are marked as unclassified). To determine the effect of compression on this technique, a SAM map was generated for the reference image and for each image reconstructed from a compressed version of the original over a wide range of compression rates. Each of the SAM maps generated from a reconstructed image was then compared to the reference; any difference was flagged as an error. Any spectra which was unclassified in the original but

erroneously classified in the reconstructed map was flagged as a “false positive” (FP) error, and any spectra which was classified in the original but unclassified in the reconstructed image map a “false negative” (FN) error.

Two different test runs were executed using different images and spectral libraries (Table 3), but in both cases the tests were executed against both the entire spectral library and against only a single spectrum. However, in no instance were the results for the entire library significantly different than those for the single element, so only one of each type is shown. For the Cuprite, NV image (Figure 1), all spectra in the USGS Mineral Spectral Library (library A) were used to generated the SAM map. For the Jasper image (Figure 2), only the spectrum for California Valley Oak was used from the Jasper Ridge Spectral Library for Green Vegetation, Dry Vegetation, and Rocks (library B). In both cases, the default error threshold of 0.10 radians was specified for the classification. Both of these libraries are supplied with ENVI. For both resulting SAM maps, unclassified regions are shown in black.

Table 3: Spectral libraries used for testing

Library	ENVI File	Title	Spectra
A	USGS_MIN.SLI	USGS Mineral Spectral Library	481
B	VEG_2GRN.SLI	Jasper Ridge Spectral Library for Green Vegetation, Dry Vegetation, and Rocks	25

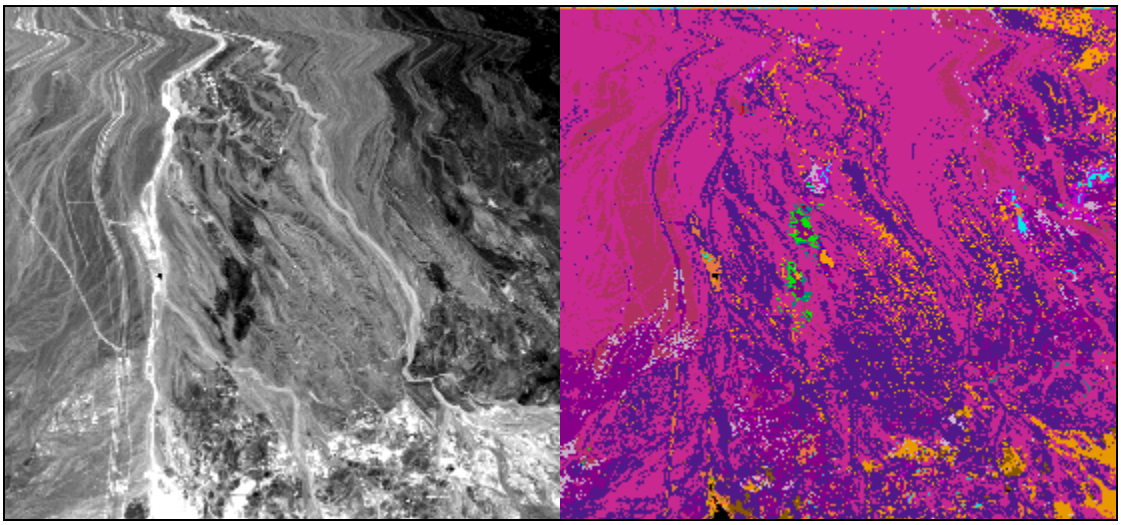

Figure 1: Band 40 of image 1, Cuprite, NV (left) and SAM map (right)

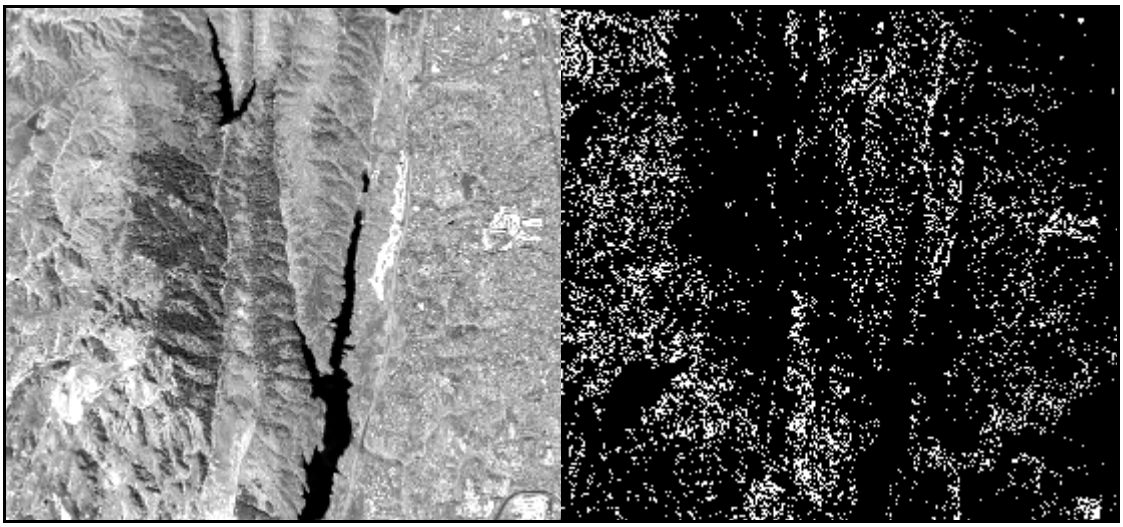

Figure 2: Band 40 of image 2, Jasper Ridge (left) and SAM map for California Valley Oak (right)

Matched Filter-based Error Utility

Matched Filtering is also an automated spectral classification technique. Similar to SAM, an image is compared with reference spectra, which may be loaded from a spectral library or selected within the image itself as Regions of Interest (ROI). However, unlike SAM which assumes that a single image pixel will be predominantly of a single active material, matched filtering can detect multiple materials which contribute to a single image pixel. It is often used as part of a hybrid method called Mixture-Tuned Matched Filtering which attempts to reduce the number of false alarms found with Matched

Filtering alone (ENVI [online]), though the hybrid method was not used for the purposes of this error utility. The result of a Matched Filtering analysis in ENVI is a three dimensional file, with one band for each reference spectra used. Each band is a gray-scale image with a floating point value for each pixel which indicates the relative degree of match to the reference spectra, where 1.0 is a perfect match.

For this error utility, PSNR and MAE on the Matched Filtering output is used to determine the relative degree of similarity between the reference result and the results for each reconstructed image. Note this should not be confused with using PSNR or MAE to compare the images themselves per se; rather, they are used to quantify the difference in floating point values between two sets of spectral analysis results.

For this testing, seven points were chosen by hand on the reference image (Figure 2 – Jasper Ridge) to serve as Regions of Interest (ROI). The spectra for these ROI were used as input to ENVI's Matched Filtering routine, as the reference spectra to which all other pixels were compared. As noted previously, the result image from this operation has the same spatial dimensions as the input image, but only seven bands in the third dimension – one for each input spectra. For these output bands as displayed in Figure 3, brighter pixels indicate a greater degree of match to the reference.

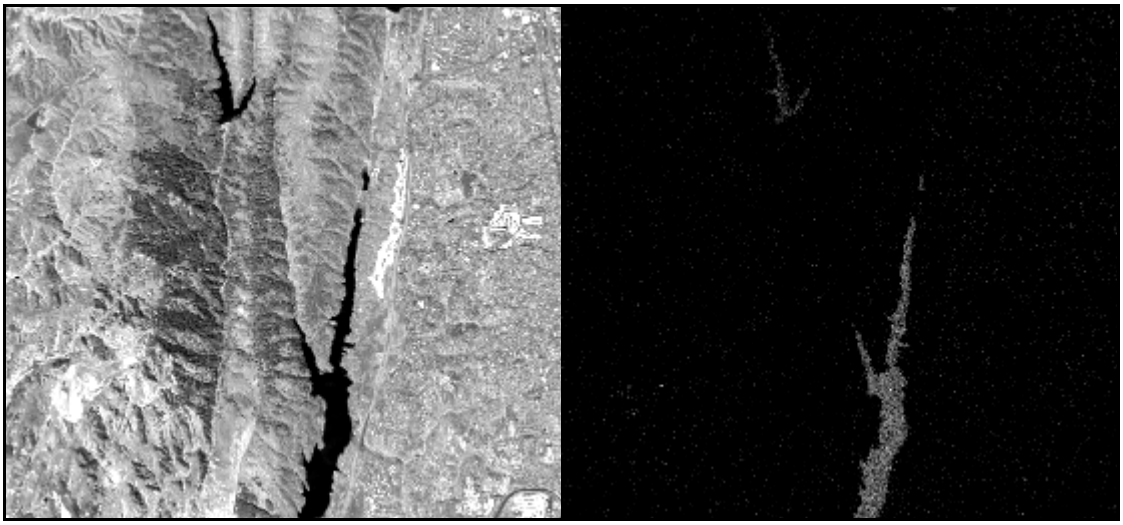

Figure 3: Band 40 of image 2, Jasper Ridge (left) and MF results for a sample ROI (right, pixels with magnitude of 0.1-0.65)

Chapter III: 3-D SPIHT Modifications

The 3D-SPIHT compression algorithm at its simplest is the application of two discrete processing steps to an image. First, the wavelet transform is applied (repeatedly and typically in multiple dimensions) in order to concentrate the most important information in a small portion of the image. Then, the SPIHT algorithm processes the wavelet transformed data by bit plane, efficiently encoding the data such that the most important data is encoded first (as covered in more detail in the Appendix).

Several modifications were tried at various points of the algorithm, but the one with perhaps the most interesting results was the application of a predetermined multiplier for each wavelet transformed data point. In this technique, a multiplier is chosen depending on the sub-band in which a data point is located; every data point in the sub-band is multiplied by this value after the wavelet transform is complete and before the SPIHT processing into *zero-trees*. During the decoding step, the multiplier is reversed after the SPIHT processing and before the reverse wavelet transform. This requires that both encoder and decoder match in order to apply and reverse, respectively, the same predetermined multiplier at each location. However, it does not require any other extra

data to be sent with the encoded data nor does it result in an additional source of error, other than that introduced by floating point rounding.

This step is intended to introduce a bias affecting the relative composition of spectral to spatial information in the 3D-SPIHT encoded bit stream. The result of the wavelet encoding step is a data cube in which the sub-band planes formed by three of the faces have the highest amount of relative energy for their respective dimensions. For example, refer to Figure 4; the frontmost plane contains the most spectral energy (which is expressed in the depth dimension), with each successive plane containing correspondingly less. The 3D-SPIHT algorithm takes advantage of this in order to efficiently encode the data into its zero trees. To introduce the bias and enhance the relative magnitude of data for a given dimension, all of the sub-band cubes in the plane with the most energy are multiplied by the largest factor, with each successive plane receiving a smaller one. This mechanism, from at least an intuitive point of view, allows for manipulation of the relative balance of spectral versus spatial information in the encoded 3D-SPIHT data stream. Technically this is not an entirely accurate description, as indeed the lowest frequency spatial information is weighted just as highly as the lowest frequency spectral information (e.g., in the lowest level sub-band cube). However, the simplicity of this description is appealing and appears to match the observations discussed later in this text.

For the purposes of this research, the planes of decreasing spectral energy were numbered beginning with the lowest level sub-band plane. Then, the multiplier for each of the data points in each plane was determined as follows:

1. M_N is the multiplier for every coefficient (wavelet transformed data element) in Plane N
2. P = number of planes; P_N is Plane N, where $N = \{1 \ldots P\}$

3. $M_N = 1 + (M_1 - 1) * \left(1 + \frac{1 - N}{P}\right)$

Figure 4 shows an example sub-band cube resulting from the application of two levels of wavelet decomposition. The spectral planes are numbered as noted above with an initial multiplier of 4.

Table 4 lists the various biases tested during this research and the multiplier (M_1) applied to the first sub-band plane. Note, results for bias 3, 4, and 6 are not shown in any of the supplied charts; 3 and 4 were not found to add interesting data for the purposes of this paper and bias 6 was used as a control (and was found to have no deviation from the unmodified algorithm, as expected). Bias 7 is different from the others in that M_1 is less than 1; note that this has the opposite effect of biases 1-5 by gradually *increasing* the multiplier in planes of *decreasing* amounts of spectral information, with the multiplier for the final plane approaching 1 for a large number of planes.

Table 4: Types of scalar multipliers tested

Bias	Type	Initial Multiplier (M_1)
1	Linear Multiplier	2
2	Linear Multiplier	4
3	Linear Multiplier	16
4	Linear Multiplier	32
5	Squared Multiplier (Bias #2 values squared)	16
6	Flat Multiplier (all values multiplied by 2 regardless of plane)	2
7	Linear Multiplier	0.5

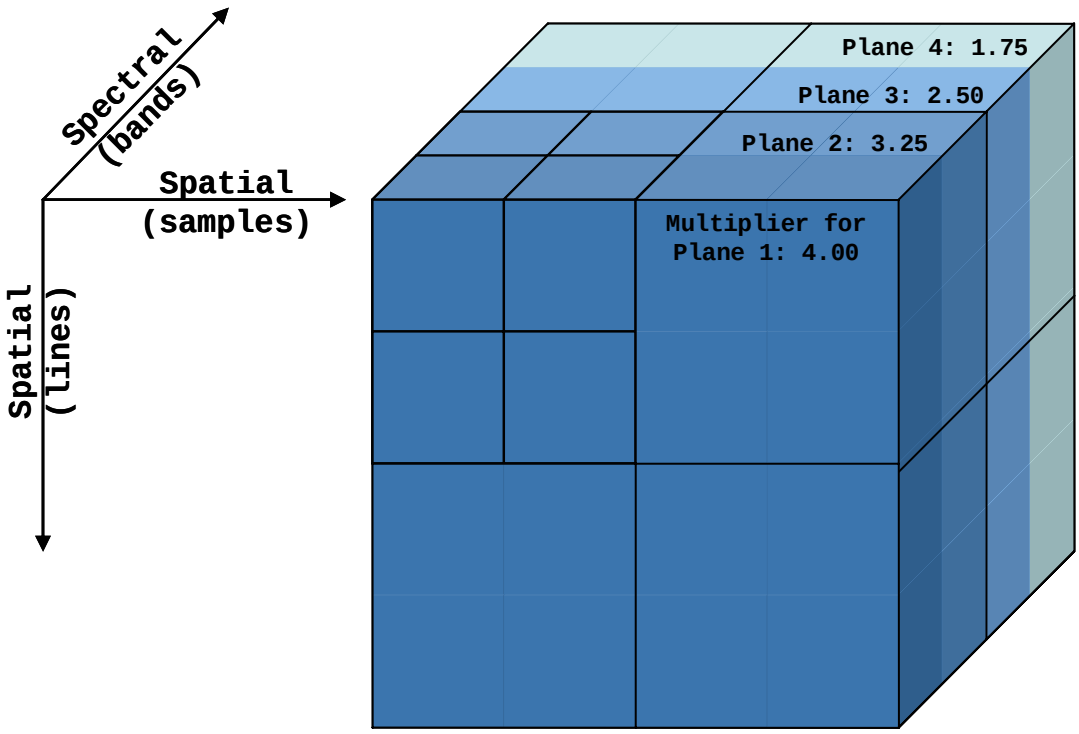

Figure 4: Example sub-band matrix with planes of decreasing spectral energy (and corresponding multipliers)

Chapter IV: Findings and Discussion

SAM-based Error Utility on Unmodified 3D-SPIHT for Image 1

In Figure 5 and Figure 6, we see a typical graph of PSNR and MAE for the image of Cuprite, NV. Both improve rapidly at first, through perhaps 0.3 bits per voxel, and then begin to level off as the refinement process takes more and more data for additional improvement. These can be compared with Figure 7, which shows the results for the SAM error utility.

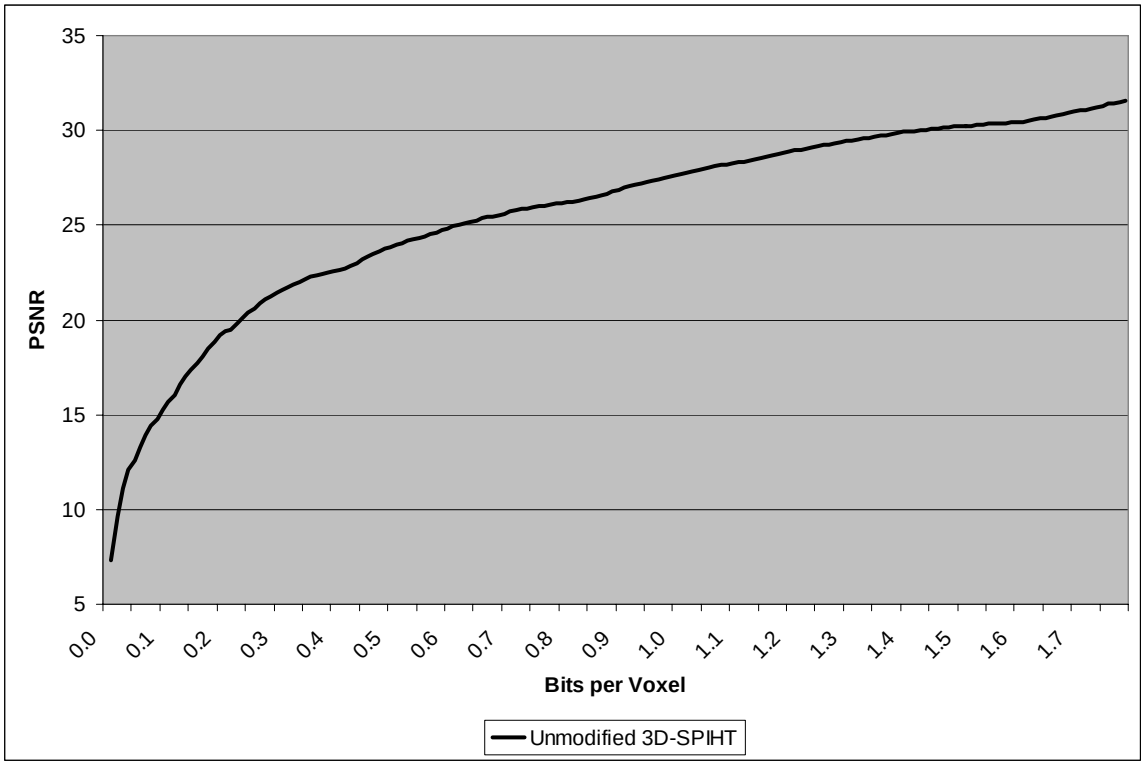

Figure 5: PSNR vs. compression rate for image 1

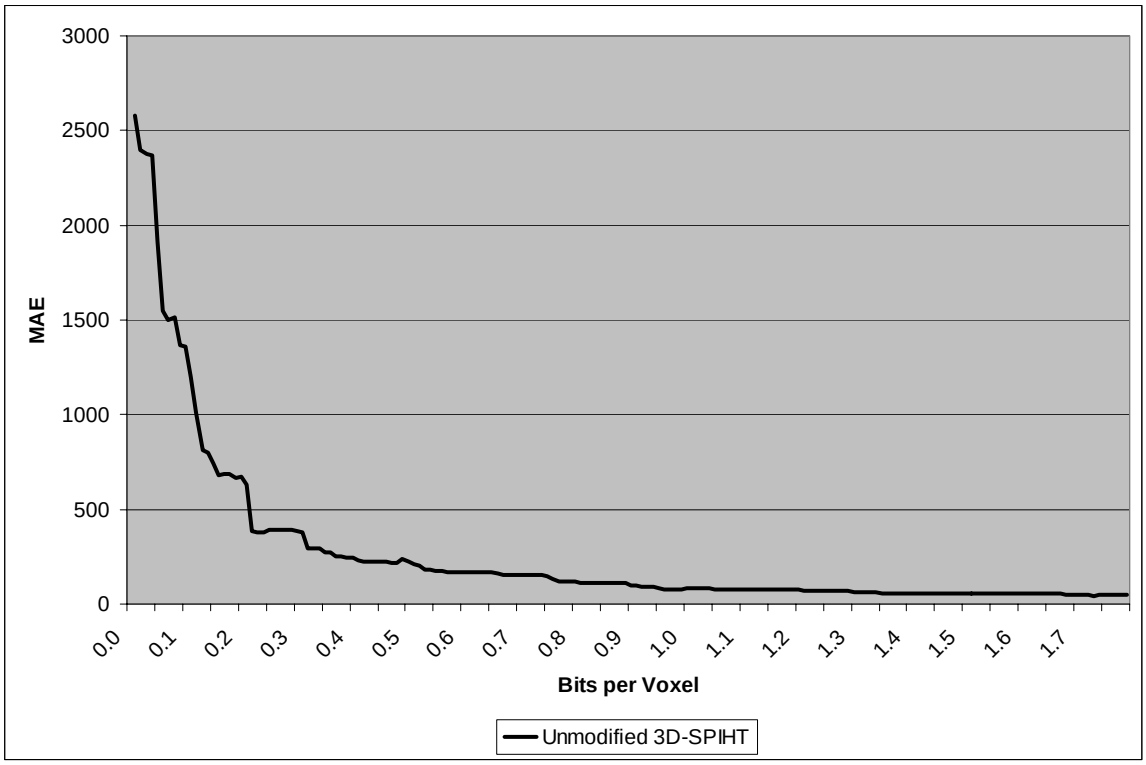

Figure 6: MAE vs. compression rate for image 1

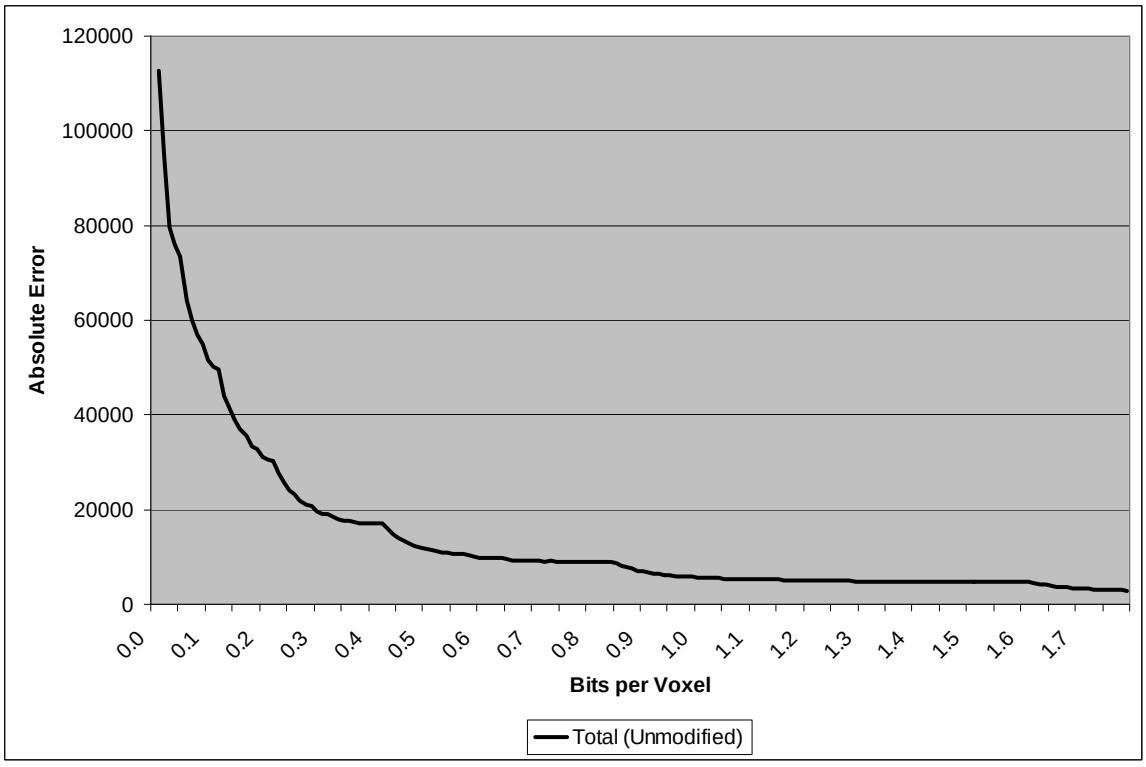

Figure 7: SAM error vs. compression rate for image 1

The most noticeable characteristic of the SAM graph is the sudden reductions in error rate followed by increasingly long periods of apparent stasis, which become apparent at around the 0.3 bits per voxel rate. Referring to Figure 8 it can be seen why this is the case – the sudden improvements in quality correspond to the start of the 3D-SPIHT refinement pass. In this portion of the algorithm, the next bit plane of the known significant digits is transmitted, which reduces the possible range of values for that voxel by half. There is no overhead involved with this pass in 3D-SPIHT; both encoder and decoder "know" exactly how many bits should be processed and do not need and "flow and control" or similar bits which pad out the data stream. In the significance pass, voxels currently marked as insignificant are processed via the zero-tree method. While 3D-SPIHT is remarkably efficient in this encoding, there is still overhead in which the encoder must communicate to the decoder which voxels are becoming newly significant.

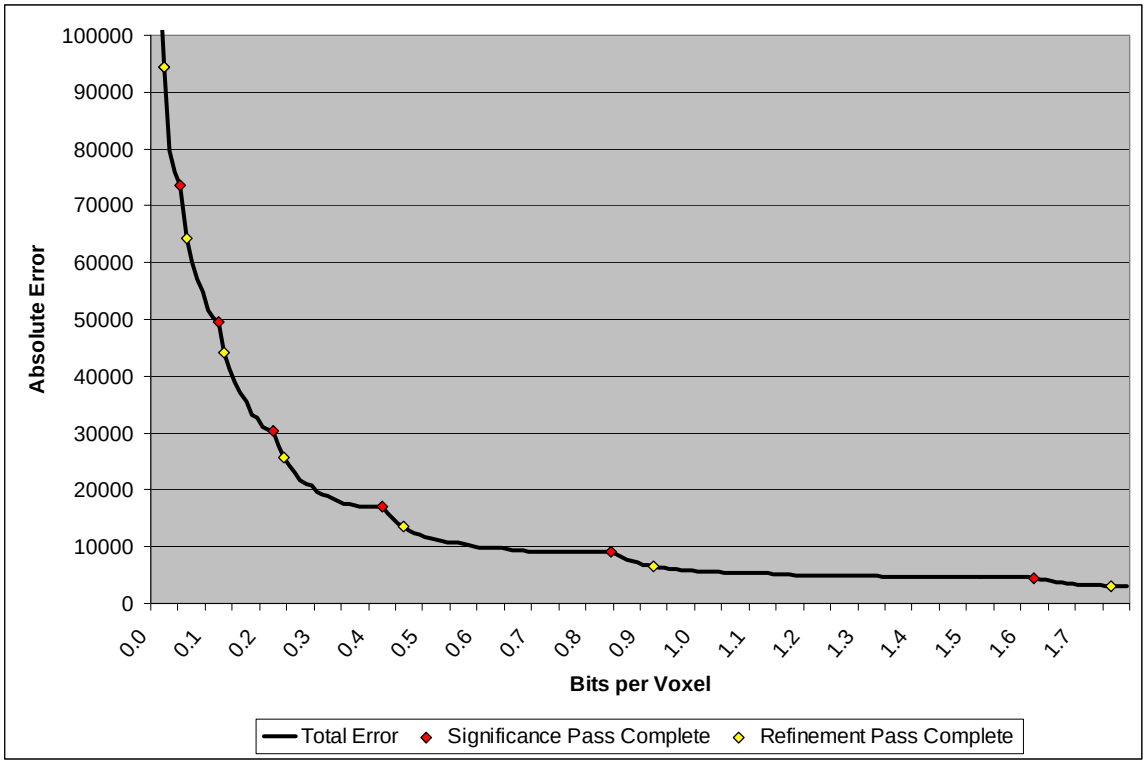

Figure 8: Affect of compression stages on SAM error, image 1

Clearly, for images which show this sort of progression that will be used in a SAM type analysis, there is little advantage to continuing the transmittal or storage of bits generated

in the significance pass if no refinement data will be processed for that bit plane as well. This also suggests that perhaps 3D-SPIHT and similar algorithms would be better served with transmitting initial refinement data as soon as a voxel is found significant.

PSNR and MAE results for image 2 have been omitted from this section due to the large degree of similarity with the results for image 1. However, the SAM error and PSNR graphs for the unmodified algorithms may be found in Figures 13 and 14, respectively, in comparison with the modified algorithm results.

MF-based Error Utility on Unmodified 3D-SPIHT for Image 2

The Matched Filtering results are an interesting contrast to those for the SAM images. Like the SAM images, analysis fidelity increases with the compression rate as should be expected (note for this error metric, higher values indicate better results). However, the most improvement seems to be made not during the refinement pass, but rather during the significance phase of the algorithm (Figure 9). Further exploration shows that this occurs during the processing of the List of Insignificant Sets (LIS) portion of the significance pass (not shown), but little explanation is forthcoming for this behavior. Regardless, a similar conclusion to that for the SAM analysis can be reached, which is that transmitting refinement data as soon as possible should result in a more linear reduction of error.

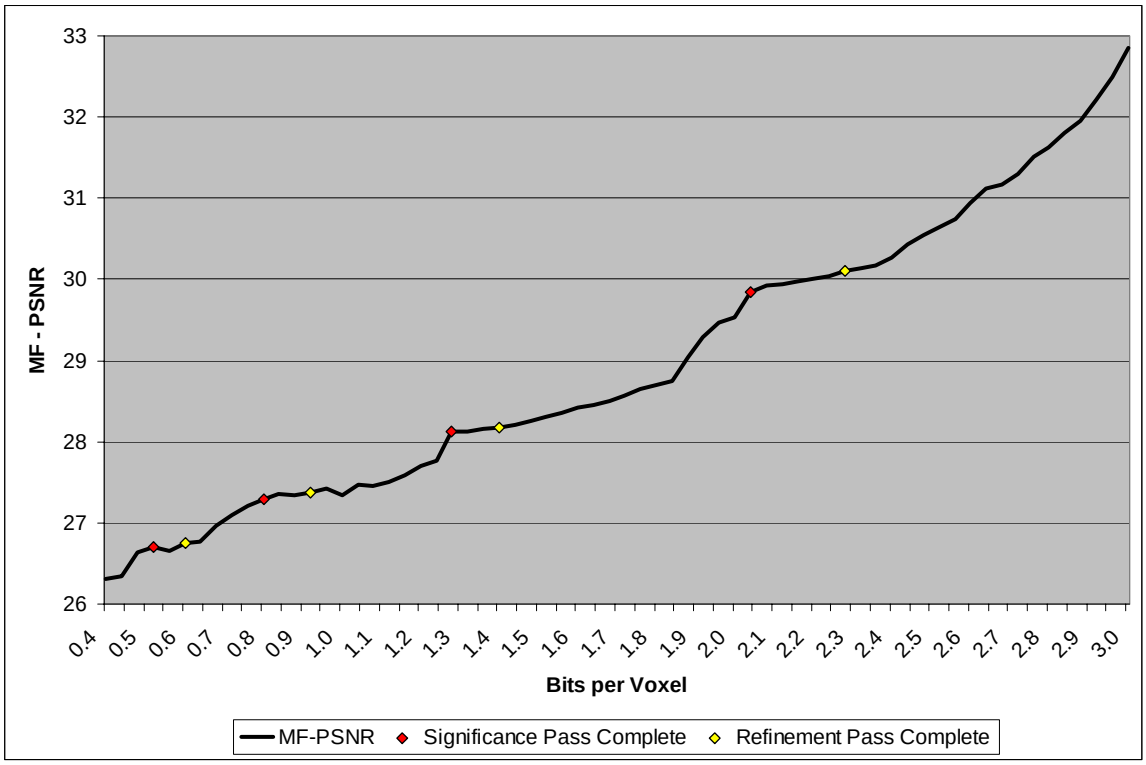

Figure 9: Affect of compression stages on MF error, image 2, California Valley Oak

3D-SPIHT Optimizations

Examining Figure 10, it is readily apparent that both of the displayed introduced biases have a direct and significant improvement to the error rate as measured by the SAM utility. Bias 5 clearly has the most substantial improvement; in it, the most significant data in the spectral dimension is encoded first, up to 4 bit planes (a factor of 16) sooner than in the unmodified algorithm. With the scalar factors being squared in this bias, the more important spectral sub-band matrix planes are much more heavily weighted than the less important ones. Note that Bias 2 is not much less efficient, and still significantly better than the unmodified 3D-SPIHT. Compare these results to those for PSNR (Figure 11) and MAE (Figure 12). Bias 5 performs significantly worse than the others for these tests, while Bias 2 is (comparatively) only slightly worse than the unmodified 3D-SPIHT.

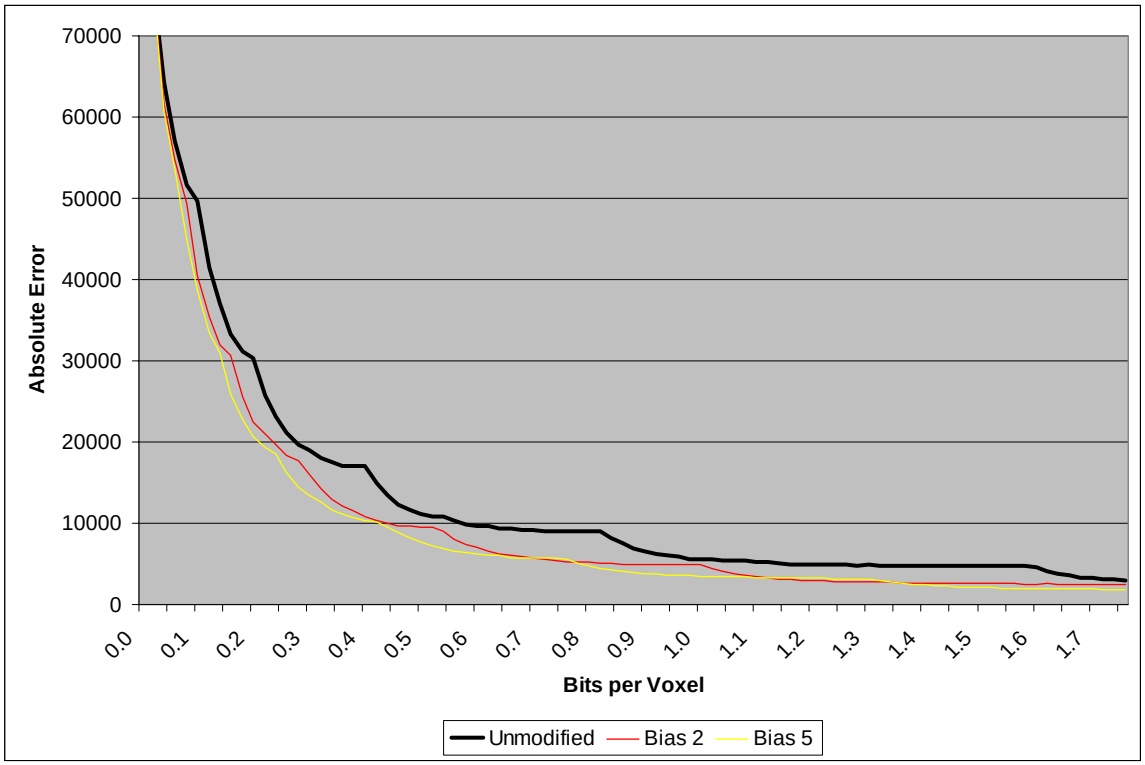

Figure 10: Comparison of SAM error for sample biases on image 1

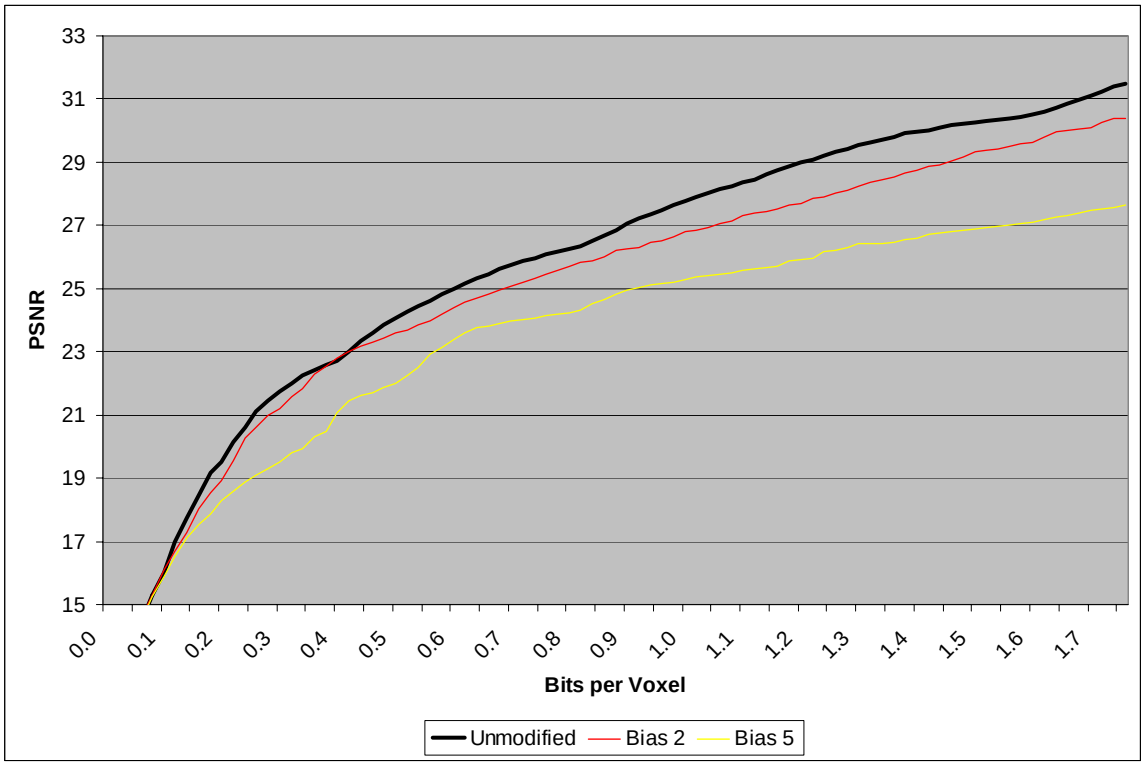

Figure 11: Comparison of PSNR for sample biases on image 1

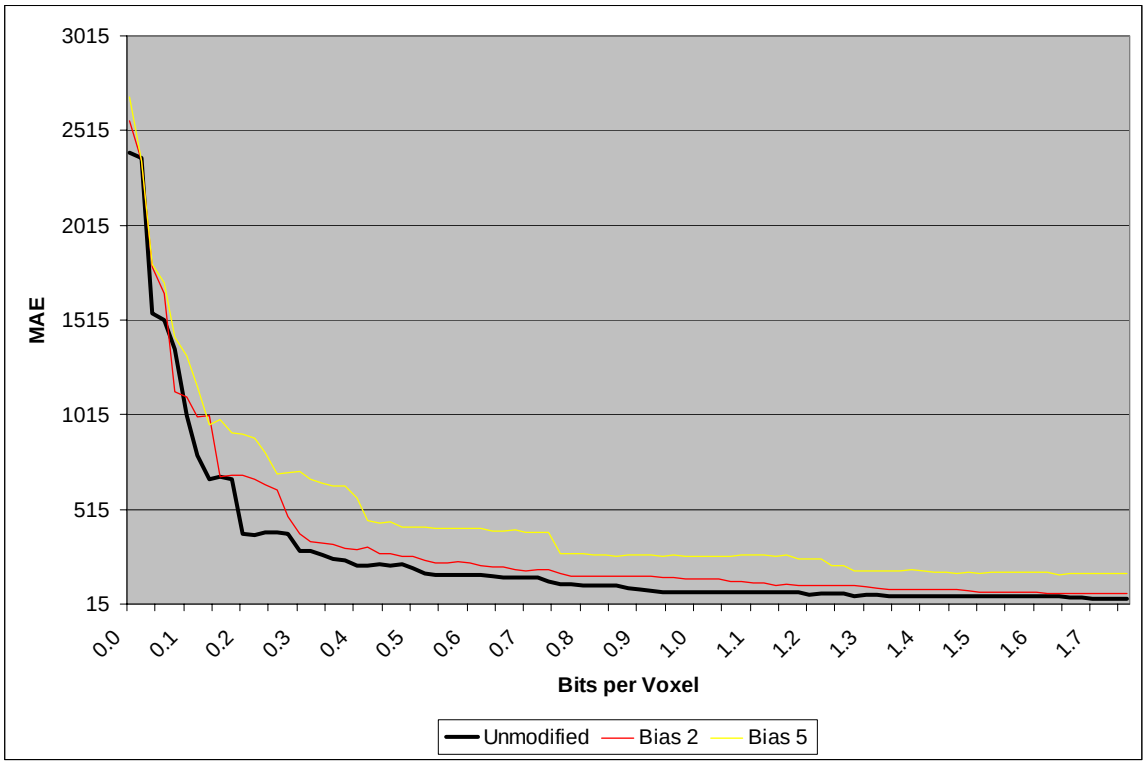

Figure 12: Comparison of MAE for sample biases on image 1

The explanation for these behaviors becomes clear when we consider the tradeoff that is being made with these 3D-SPIHT modifications. PSNR is an equal opportunity difference engine; all bits are equally important, and the metric gives a measure of average quality. The new error utilities and optimizations discussed in this paper however are biased to information in the spectral dimension – bits that contribute most to the spectral signature are valued the most highly. 3D-SPIHT and indeed almost all compression algorithms are designed to maximize the average image quality and thus PSNR; this of course is the only rational way to design the algorithms in order for them to be most flexible, as they are intended for use on a wide range of data sets. Considering this, it is only to be expected that PSNR must decrease as we move away from a balance of spectral and spatial information to one which favors data in the spectral dimension.

The next set of tests show the same metrics on a different hyperspectral image, from Jasper Ridge, CA. Jasper Ridge is a biological preserve owned by Stanford University.

In Figure 13 and Figure 14, it can be seen that both bias 1 and bias 2 of the SAM utility return lower error rates than the unmodified algorithm at almost all compression rates. Bias 5, which was previously significantly better than the others, can be seen to perform much worse than even the unmodified SPIHT.

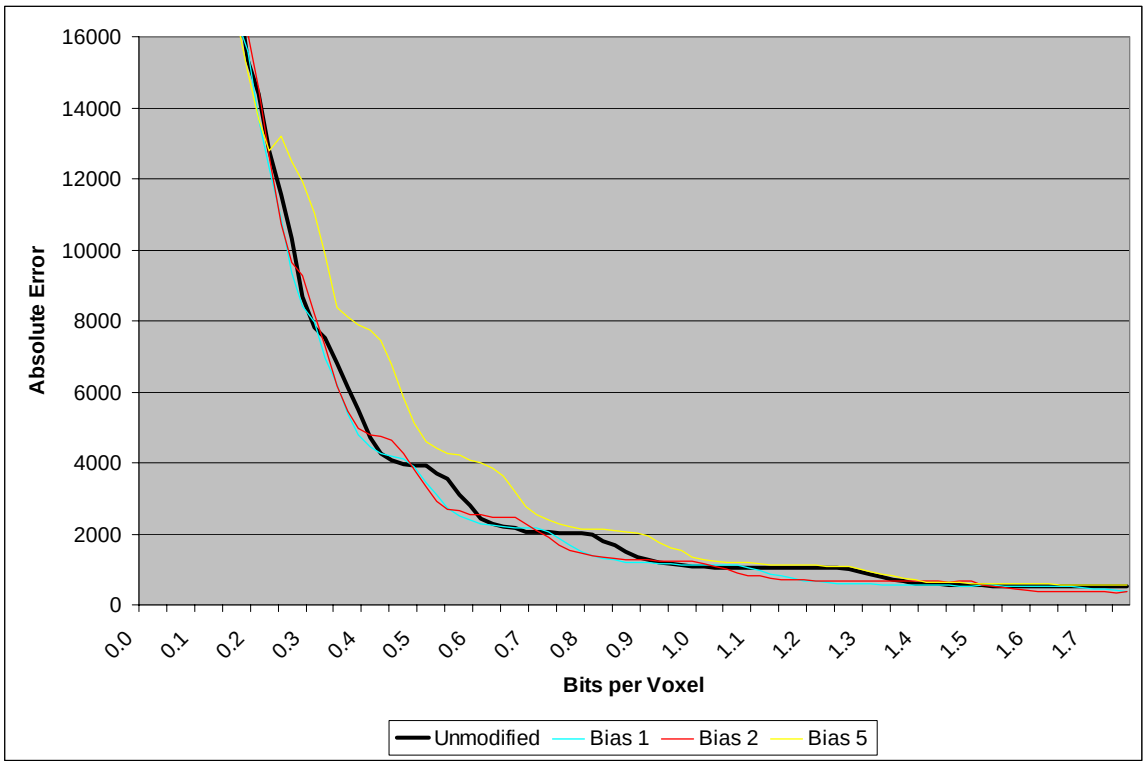

Figure 13: Comparison of SAM error for sample biases on image 2, California Valley Oak

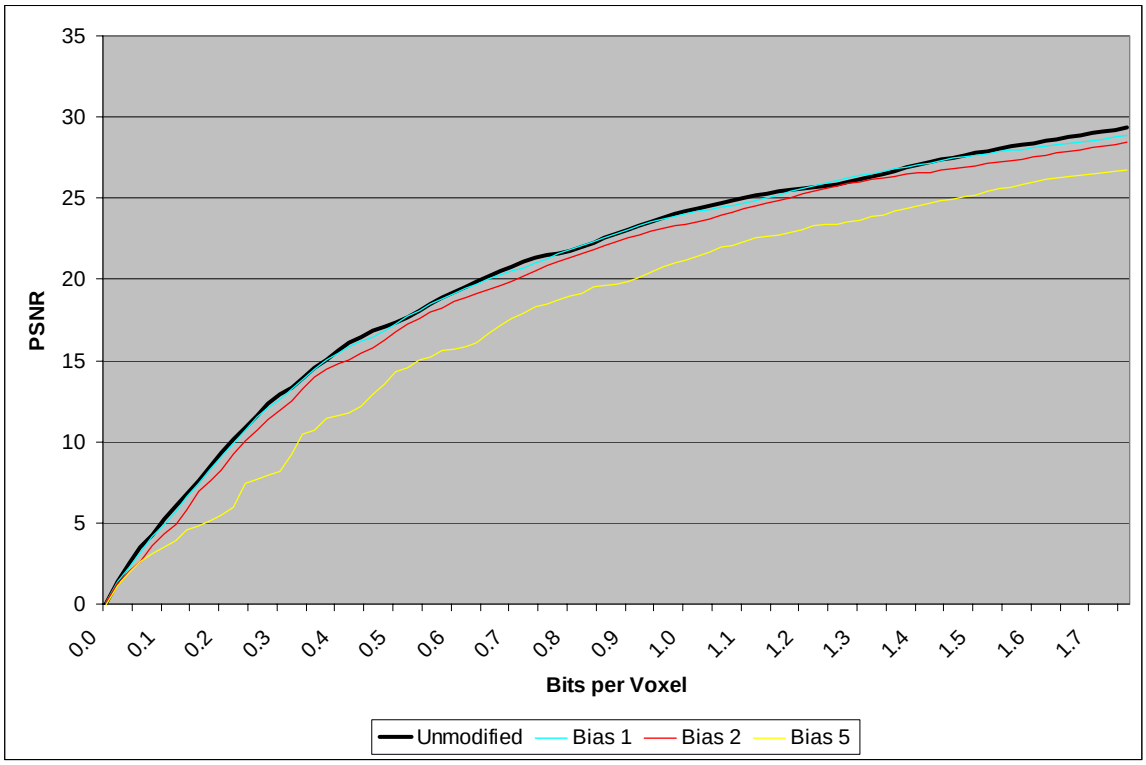

Figure 14: Comparison of PSNR for sample biases on image 2, California Valley Oak

Rationale for this can be found by examining the SAM maps of the two respective source images. For the Cuprite image, the spectra used for the error utility are for minerals, which by their nature tend to be found in patches on the image. As a result of this occurrence pattern, spatial features are less important for the deconstructed image – neighboring pixels are likely to be of the same material, and so a greater value is placed on the information in the spectral dimension. For the Jasper ridge image however, this is much less the case. The spectrum here is that of the California Valley Oak, which occurs more evenly throughout the picture and in smaller patches (i.e., as single trees, typically on the order of a single pixel). Here, the spatial information is more important in order to distinguish between the areas of "oak" and "not oak", and so the larger spectral biases prove to be overwhelming in the sense that they give too low a priority to the spatial information.

The final set of tests, run on Image 2 with custom Regions of Interest, uses the alternate error utility involving Matched Filtering (Figure 15). Here, Bias 2 performs particularly poorly when compared with the unmodified algorithm. Bias 7 however does quite well, outperforming the unmodified algorithm at a majority of the compression rates. Recall that Bias 7 applies the smallest multiplier (0.5) to the plane of highest spectral energy – effectively reducing the magnitude of that information, and thus introducing a prioritization bias for the spatial information in the image. This implies that at least for the images used in this research, the spatial information is of more relative importance to the identification of materials than the spectral.

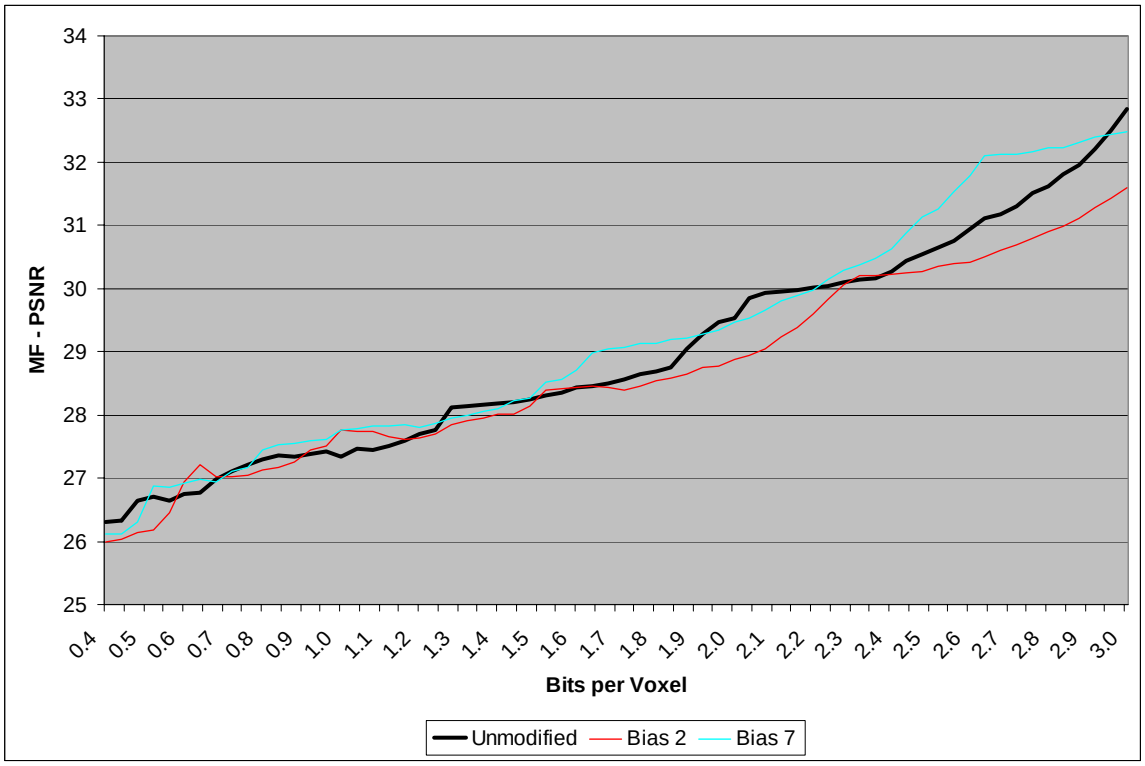

Figure 15: Comparison of MF error for sample biases on select ROI in image 2

Careful examination of the Matched Filtering method helps to explain why this may be the case. The method attempts to reject undesired and interfering signatures in the image by filtering out random noise, which is assumed to be an "independent, identically distributed Gaussian process with zero mean and covariance matrix" (Harsanyi 1994).

Then, the selected endmembers (the chosen ROI) are identified with the expectation that the remaining signal energy is a linear combination of those and other spectrally distinct materials. It can be surmised that the compression affects the assumption of the independence of the contributing variable spectra, with a corresponding loss in the ability to remove the noise. On the other hand, the selected ROI are input to the system as locations on the input image itself, and the spectra for them come from the same reconstructed image that is being analyzed. Therefore, these desired signatures already partially account for the compression which has been applied. These factors, at least for the image and ROI used in these tests, appear to tip the balance of spectral versus spatial information in the direction of increased spatial resolution. This is in direct contrast to the earlier SAM tests, where the matching spectra come from a library and spectral faithfulness to those signatures was the top priority.

Future Work

There are a number of areas in which this research could be continued in order to provide a usable mechanism for compression of hyperspectral images. One such would involve further exploration into the effect of various types of bias introduced in the spectral dimension and modification of the compression algorithm to accommodate a choice of bias, to be encoded with the compressed file. Another should involve additional spectral analysis tools and the compression of radiance data in order to verify the results as discussed here are applicable to a wide range of usage patterns. Additional compression algorithms should be examined to see how a spectral bias might be applied and tested with the error utilities presented here, as well.

In some cases, the introduced modifications to 3D-SPIHT gave very significant performance improvements. For example, if given a quality target of 98.5% correct spectral classifications for image 1, bias 5 reaches that goal with about half the amount of data (0.86 vs. 1.64 bits per voxel). However, as shown with image 2, this is not a uniform improvement for all images. The relationship between the importance of spectral and spatial information in an image warrants further investigation; ideally, a step

involving minimal processing would be able to select the ideal bias to introduce for the image. The difference in performance depending on the spectral analysis technique used complicates this issue, as the biases covered here did not uniformly improve results across the different techniques. This is likely to be a limiting factor in the use of the discussed modification without further research into how various methods are affected.

Additionally, similar error metrics can be created with other types of spectral analysis. As a sanity check, it would be very insightful for someone skilled with manual spectral analysis techniques to analyze the images at the different compression rates and see how those correspond to the results with the automated techniques. Human analysis appears to be quite common, especially for imagery of high importance, and it may be that a human operator is more dependent on spatial features and would not fare as well with compression which obscures that data. On a related note, it would be interesting to test the compression on radiance images (reflectance imagery was used here), with correction for sensor skew, atmospheric conditions, and other factors performed only after image reconstruction. Depending on the type of correction needed, this could be an important factor; it is likely that compression of the radiance imagery would be more useful and common in the field.

One unexplained finding of interest was the impact of the significance pass versus the refinement pass on improvements in the SAM utility and the MF utility. This is something which certainly warrants further investigation. Other compression schemes which merge these passes would be interesting to study here as well, to see how they fare. While 3D-SPIHT was used exclusively for these tests, there is no reason why other algorithms might not benefit from some sort of spectral bias, though the application of such is likely to be significantly different than was done here. As the error utilities discussed here are completely independent of compression method, results would be directly comparable.

Summary and Conclusions

The primary purpose of this study was to explore the applicability of the commonly used error metrics PSNR and MAE to the spectral analysis of compressed hyperspectral images. This was accomplished through the creation of new error utilities based upon actual spectral analysis techniques which would reliably reflect this application, the results of which were compared with PSNR and MAE. A modification to 3D-SPIHT was made to permit the application of a variable bias to the compression of the images, allowing a shift in the balance of spatial versus spectral priority in the encoded data stream. Biases were tested which resulted in an improvement in the ability to spectrally analyze reconstructed hyperspectral images (as measured by these new utilities), but that caused a corresponding decrease in PSNR and MAE scores. Therefore, it can be concluded that PSNR and MAE do not accurately model compression loss with regards to the hyperspectral analysis methods of Spectral Angle Mapping and Matched Filtering. It is expected that other lossy compression techniques which utilize wavelet transforms will show similar results when a bias is introduced for the spectral dimension versus the spatial ones.

End Notes

1. AVIRIS [Online]. Available at http://aviris.jpl.nasa.gov/html/aviris.concept.html

2. Barry, P.S. and Pearlman, J. "The EO-1 Mission: Hyperion Data." Available at http://www.eoc.csiro.au/hswww/eo1_cm/docs.htm (August 2001)

3. Dragotti, P. L. et al. "Compression of Multispectral Images by Three-Dimensional SPIHT Algorithm." IEEE Transactions on Geoscience and Remote Sensing, 38.1 (January 2000): 416-428.

4. EO-1 [Online]. Available at http://eo1.usgs.gov

5. ENVI [Online]. Available at http://www.rsinc.com/envi/

6. ENVI tutorial and reference guides (not publicly available)

7. Harsanyi, J.C. and C.I. Chang. "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach." IEEE Transactions on Geoscience and Remote Sensing, 32.4 (July 1994): 779-785

8. Johnson, M et al. "Networking Technologies Enable Advances in Earth Science." Computer Networks, 46.3 (October 2004): 423-435

9. Keränen, Pekka et al. "Spectral similarity measures for classification in lossy compression of hyperspectral images." Proceedings of SPIE, 4885 (2003): 285-296

10. Kim, B. and W. A. Pearlman. "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical tree." IEEE Data Compression Conference (March 1997): 251-260

11. QccPack [Online]. Available at http://qccpack.sourceforge.net/

12. Said, A. and Pearlman, W. A. "New, fast, and efficient image codec based on set partitioning in hierarchical trees." IEEE Transactions on Circuits and Systems for Video Technology. 6.3 (June 1996): 243-250

13. Salomon, David. Data Compression: The Complete Reference, Third Edition, New York: Springer-Verlag (2004)

Bibliography

1. AVIRIS [Online]. Available at http://aviris.jpl.nasa.gov/html/aviris.concept.html

2. Barry, P.S. and Pearlman, J. "The EO-1 Mission: Hyperion Data." Available at http://www.eoc.csiro.au/hswww/eo1_cm/docs.htm (August 2001)

3. Dragotti, P. L. et al. "Compression of Multispectral Images by Three-Dimensional SPIHT Algorithm." IEEE Transactions on Geoscience and Remote Sensing, 38.1 (January 2000): 416-428.

4. EO-1 [Online]. Available at http://eo1.usgs.gov

5. ENVI [Online]. Available at http://www.rsinc.com/envi/

6. ENVI tutorial and reference guides (not publicly available)

7. Goetz, et al. "Introduction to the proceedings of the Airborne Imaging Spectrometer (AIS) data analysis workshop: in Proceedings of the Airborne Imagine Spectrometer Data Analysis Workshop." JPL Publications (1985): 1-21

8. Harsanyi, J.C. and C.I. Chang. "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach." IEEE Transactions on Geoscience and Remote Sensing, 32.4 (July 1994): 779-785

9. Johnson, M et al. "Networking Technologies Enable Advances in Earth Science." Computer Networks, 46.3 (October 2004): 423-435

10. Keränen, Pekka et al. "Spectral similarity measures for classification in lossy compression of hyperspectral images." Proceedings of SPIE, 4885 (2003): 285-296

11. Kim, B. and W. A. Pearlman. "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical tree." IEEE Data Compression Conference (March 1997): 251-260

12. Lim, Sunghyun et. Al. "Compression for Hyperspectral Images Using Three Dimensional Wavelet Transform." Geoscience and Remote Sensing Symposium. 1 (July 2001): 109-111

13. QccPack [Online]. Available at http://qccpack.sourceforge.net/

14. Said, A. and Pearlman, W. A. "New, fast, and efficient image codec based on set partitioning in hierarchical trees." IEEE Transactions on Circuits and Systems for Video Technology. 6.3 (June 1996): 243-250

15. Salomon, David. Data Compression: The Complete Reference, Third Edition, New York: Springer-Verlag (2004)

16. Schiewe, J. "Effect of lossy data compression techniques on geometry and information content of satellite imagery." ISPRS Commission IV Symposium on GIS – Between Visions and Applications, 32.4 (1998): 540-544

17. Shapiro, J. M. "Embedded image coding using zerotrees of wavelet coefficients." IEEE Transactions on Signal Processing, 41.12 (December 1993): 3445-3462

18. USGS Spectroscopy Lab [Online]. Available at http://speclab.cr.usgs.gov/

19. Yu, Shanshan and Zhang, Ye. "Compression of Hyperspectral Image Based on Three-Dimensional SPIHT Algorithm." Proceedings of SPIE, 4875 (July 2002): 445-450

20. Yuhas, R. H. and A. F. H. Goetz. "Monitoring and Modeling Semi-Arid Landscape Response to Climate Change." Geoscience and Remote Sensing Symposium, 2 (August 1994): 1036-1038

Appendix A: Tools and Algorithms

3D-SPIHT

SPIHT (Set Partitioning In Hierarchical Trees) is an embedded wavelet coding algorithm used to compress images. Key attributes of this algorithm are its low complexity and progressive nature, allowing low-resolution rendering of an image before all image data has been received. That is, the best possible image for a given number of bits can be reproduced given that number of prefix bits from a SPIHT encoded data stream. In other words, an image compressed with SPIHT can be truncated at any point, and that subset of information can be reconstructed into the best possible representation of the original that SPIHT can produce with that amount of data. This can be compared to many other algorithms, in which a fixed size image thumbnail is generated – or with which all data must be received in order to create any image. This also means that it is possible to achieve any compression ratio (uncompressed to compressed size) by only storing or transmitting that particular amount of data; for example, from one compressed file, a highly compressed data stream could be sent to a portable device and a near-lossless stream could be generated to a fully equipped PC.

SPIHT first specifies the application of wavelet transforms to the data, such that lower frequency information (and thus higher energy) is accumulated in a subset of the data. If a wavelet transform is applied in one dimension to some data, the data is split into low frequency (L) and high frequency (H) sections. For multi-dimensioned data, the transform is simply applied sequentially in each dimension; so for two dimensioned data such as most images, the result is a block of four subsets (AKA sub-bands): LL, LH, HL, and HH, with LL having the lowest frequency information. This idea can be simply extended to three (or even more) dimensions, so that one ends up with LLL, LLH, LHL, LHH sub-bands, and so forth. In dyadic wavelet transform, the lowest level sub-band is recursively transformed as desired in order to further concentrate the lowest frequency data; repetition count is limited by some minimal sub-band size (which depends on the wavelet transform), and some function of the length of the shortest dimension. The

Daubechies 9/7 biorthogonal transform is the most commonly used in practice, though no particular transform is specified or required by the SPIHT algorithm. The wavelet transformation step is reversed when the data stream is uncompressed, and does not contribute significantly to any data loss.

SPIHT takes advantage of this frequency ordering with spatial orientation trees (or zero-trees). A zero-tree is used under the assumption that lower frequency data is more likely to be significant (that is, have a higher value) than higher frequency data. A zero-tree is a structure in which each level of the tree contains pixels of a different sub-band, with the root being in the lowest sub-band and each additional level coming from the next lowest sub-band. Each of those additional levels serves to add further detail to the lower frequency information in their parent node. SPIHT processes these coefficients in bit order: that is, the current highest significant bit of every (significant) coefficient is transmitted before the next significant bit of any coefficient is processed. Thus, if an entire tree is insignificant at a certain bit level, one bit can be transmitted to indicate that, which is quite efficient. Bits resulting from this method of zero-tree processing can be further compressed by various common lossless algorithms for some small improvement in compression performance, though the gain is nominal (Salomon 604).

SPIHT was expanded to operate on three dimensional data by Kim and Pearlman (Kim and Pearlman 97), specifically for use in video encoding. For this use, it has been shown to produce significantly higher quality video than the MPEG-2 standard at the same bit rate, both as measured by PSNR and through visual inspection. It has also been adopted for use with hyperspectral images, where it is one of the commonly used algorithms today.

ENVI

ENVI is a commercial software package for image processing and analysis. With respect to hyperspectral images, it performs a number of functions, but perhaps most relevant to this paper is the ability to compare spectral signatures to those in an included library for

determination of image content (ENVI [online]); for example, identification of mineral deposits. ENVI is built on an underlying language, IDL (Interactive Data Language), with which custom routines for batch processing and image manipulation can be written by the user.

QccPack

QccPack is an open source library of routines and utilities specifically for the quantization, compression, and coding of data. These include implementations of SPIHT, 3D-SPIHT, entropy coding (Huffman, Arithmetic, and Golomb algorithms are included), and a number of popular wavelet transforms (QccPack [online]). Though SPIHT and 3D-SPIHT are patented algorithms, a license is provided which does not restrict any sort of academic or non-commercial use. For this paper the most recently available QccPack, version 0.51, was used.

3D-SPIHT Modification

For the optimization modification as discussed in Chapter III, only the single source file spiht3d.c was modified. A single new function was added in order to handle both the application and removal of the multiplier, which was called from the QccSPIHT3DEncode and QccSPIHT3DDecode methods in between the wavelet transform application/removal and SPIHT processing. Refer to Chapter III for logic details.

Error Utility Automation

Both the SAM and MF based error determination utilities were automated, as these tasks were quite time consuming (data for the highest bit rates took approximately 20 minutes to collect on the available machine, a dual processor Apple G5 with 6GB of RAM); this allowed for much greater data collection and more trials than would have otherwise been possible.

Two test scripts were used to control task execution. For each modification to SPIHT, these scripts were re-run to collect data. The first, INIT.SH, resampled the three input spectral libraries to match the source image (RESAMP_SPEC_LIBS.PRO), generate the SAM and Matched Filter maps for the source, and initialize the environment for the RUN.SH script. The second script, RUN.SH, generated a "master" compressed image at a slightly higher bit rate than the maximum, and then performed the following steps for each of the specified test bit rates.

1. Compress the source image at the specified bit rate. This operation was simulated by creating a new file containing only the first N bytes of the "master" compressed image, where N is the number of bytes expected for a given compression rate. This is possible since SPIHT is a progressive compression algorithm; the first N bytes of any two versions of a single SPIHT compressed image are identical. This was done since SPIHT compression was one of the lengthiest operations in this process (25 or more minutes for the highest bit rates), while this method executed in much less than a second in every case. Note, *arithmetic coding* was disabled for all compression.
2. Decompress the image generated in 1. This operation was the lengthiest which had to be done in the process, and there was unfortunately no shortcut as for 1; this step took up to 19 minutes at the highest bit rates to complete.
3. Obtain MSE and MAE data for the reconstructed image. For this, the tooling included with QccPack was used (ICBDIST).
4. Execute the SAM error metric in ENVI with the reconstructed image (SAM.PRO). This process generated a SAM map for both the entire supplied spectral library and the subset specified for the trial run (e.g., Cuprite).
5. Execute the Matched Filter error metric in ENVI with the reconstructed image (MF.PRO). This process generated the MF map for the specified set of spectra.
6. Calculate the error rate for the SAM maps. This was a simple Java program (SamDiff.java) which counted total error, false positives, and false negatives between the SAM map generated from the source image and those generated from

the reconstructed images. Again, this was done for both the entire supplied library and the specified subset.

7. Calculate the PSNR and MAE for the Matched Filter map for the reconstructed image as compared to that for the source MF map.

Appendix B: Terminology

Arithmetic Coding

Arithmetic Coding is a lossless data compression technique which is often used to further compress SPIHT data streams for a minor improvement in compression rate.

Dyadic Decomposition

In dyadic decomposition, wavelet transforms are applied in every dimension at every level of wavelet decomposition. For example, 5 levels of decomposition applied to a three dimensional image results in 64 sub-bands for each dimension, or 262,144 total sub-band cubes.

Full Width at Half Maximum (FWHM)

For a given wavelength interval, FWHM is the width of the waveform at one-half its maximum value.

Hierarchical Data Format (HDF)

HDF is a freely available multi-object file format for the transfer of graphical and numerical data (http://hdf.ncsa.uiuc.edu/).

Zero-Tree

A zero-tree, or spatial orientation tree, is a tree structure which defines the spatial relationship between sub-bands of wavelet transformed data into a hierarchical pyramid (Said and Pearlman 1996). Each node in the tree corresponds to a pixel and has descendants corresponding to pixels at the same spatial orientation in the next finer level of detail in the sub-band pyramid.