JPEG-2000 compression using 3D wavelets and KLT with application to HYDICE data

James H. Kasner,[*a] Ali Bilgin,[b] Michael W. Marcellin,[b] Austin Lan,[c]
Bernard V. Brower,[c] Sylvia S. Shen,[a] and Timothy S. Wilkinson,[d]

[a] The Aerospace Corporation, 15049 Conference Center Drive, MS CH1-430, Chantilly, VA 20151
[b] Dept. of Electrical and Computer Eng., University of Arizona, Tucson, AZ 85721
[c] Commercial & Government Systems, Eastman Kodak Co., 1447 St. Paul Street, Rochester, NY 14653
[d] The Aerospace Corporation, 2350 El Segundo Bl., El Segundo, CA 90245

ABSTRACT

JPEG-2000 is the new image compression standard currently under development by ISO/IEC. Part I of this standard provides a "baseline" compression technology appropriate for grayscale and color imagery. Part II of the standard will provide extensions that allow for more advanced coding options, including the compression of multiple component imagery. Several different multiple component compression techniques are currently being investigated for inclusion in the JPEG-2000 standard. In this paper we apply some of these techniques toward the compression of HYDICE data. Two decorrelation techniques, 3D wavelet and Karhunen-Loeve Transform (KLT), were used along with two quantization techniques, scalar and trellis-coded (TCQ), to encode two HYDICE scenes at five different bit rates (4.0, 2.0, 1.0, 0.5, 0.25 bits/pixel/band). The chosen decorrelation and quantization techniques span the range from the simplest to the most complex multiple component compression systems being considered for inclusion in JPEG-2000. This paper reports root-mean-square-error (RMSE) and peak signal-to-noise ratio (PSNR) metrics for the compressed data. A companion paper [1] that follows reports on the effects of these compression techniques on exploitation of the HYDICE scenes.

Keywords: JPEG-2000, compression, hyperspectral, 3D wavelet, Karhunen-Loeve transform, KLT, HYDICE

1. INTRODUCTION

The demand for hyperspectral data and the volume of hyperspectral data that must be stored, transmitted, and analyzed are rapidly increasing. At the same time, limitations on bandwidth and storage space suggest the use of compression technology. Analysts and users of derived products insist on the best possible data fidelity; frequently this translates into a requirement for lossless compression [2]. If system bandwidth cannot accommodate lossless compression, the best lossy compression is needed. Until recently, achieving satisfactory lossy and lossless compression from the same algorithm [3,4,5] seemed a mutually exclusive proposition.

The ISO/IEC JPEG group (ISO/IEC/JTC 1/SC29/WG1) began the development of the successor to the current JPEG algorithm in 1995. This development effort is now commonly know as JPEG-2000. In Part I of the developing JPEG-2000 standard [6], lossy/lossless embedded compression technologies have been developed for grayscale and color imagery applications. Part II of the standard will provide extensions to the processes defined in Part I. These extensions will include more advanced wavelet and quantization options as well as support for multiple component (i.e., multispectral, hyperspectral, medical, and scientific imagery) compression techniques.

In this paper we investigate several of the proposed compression technologies being considered for Part II of JPEG-2000 that would find use in hyperspectral image compression. We concentrate here on classical RMSE and PSNR distortion metrics. A companion paper to this one [1] examines the effects of these compression technologies on the exploitation of hyperspectral

[*] Correspondence: Email: James.H.Kasner@aero.org; Tel: +1-703-633-5353; Fax: +1-703-633-5005

Imaging Spectrometry VI, Michael R. Descour, Sylvia S. Shen, Editors,
Proceedings of SPIE Vol. 4132 (2000) © 2000 SPIE. · 0277-786X/00/$15.00

images. One of the algorithms considered in this study provides lossless and lossy embedded compression in a single bitstream.

This paper is organized as follows: In the next section, characteristics of the HYDICE data sets are briefly discussed. In Section 3, a description of the three compression algorithms that were investigated is given. Section 4 presents experimental results, and Section 5 contains conclusions, observations, and pointers to future work.

2. HYPERSPECTRAL DATA

The HYDICE sensor is an InSB detector array operating in a pushbroom mode. It collects data 320 columns wide in 210 spectral bands spanning a wavelength range of 0.4μm to 2.5μm. The nominal spectral spacing of the bands is 10nm. The detector is sensitized to improve response in different wavelength regions [7]. The scenes used in this study were 300 columns by 960 rows by 210 bands. The original image width was trimmed to eliminate bad pixels. HYDICE is nominally a 12-bit sensor, but post-acquisition calibration procedures change the data's dynamic range.

In the spectral bands corresponding to atmospheric water absorption, bands [102,109] and [137,155] (around 1.4μm and 1.9μm), the data are best treated as signed integers. The dynamic range of these bands is approximately [-50, 50] with a roughly symmetrical distribution around zero. A water-absorption band will therefore have tens of thousands of small negative values. The other spectral bands do not display this characteristic and are best treated as 16-bit unsigned integers. The dynamic range of these bands will exceed 2^{15}-1, but relatively few of the band values have magnitudes this large.

For the purposes of compression, it is important to properly interpret the data types of the bands. Treating the data as 16-bit unsigned integers is inappropriate. The small magnitude negative values in the water-absorption bands will "wrap around" to large positive integers. This creates thousands of large values in these bands and greatly increases their variance. These large values will adversely affect decorrelation and rate allocation procedures and lead to poor compression performance. In the exploitation study performed in a companion paper [1], the water-absorption bands were not used, so the negative values in these bands have been zeroed out for this paper and the exploitation study. All negative values in the range [-149, -1] have been set to zero. This range was chosen to preserve values in non-water-absorption bands that exceed 2^{15}-1 that will map to large magnitude negative numbers when interpreting the data as signed 16-bit integers. For applications where the water-absorption information is useful, such as atmospheric modeling, this procedure should not be applied, and the water-absorption bands should be treated as signed integers.

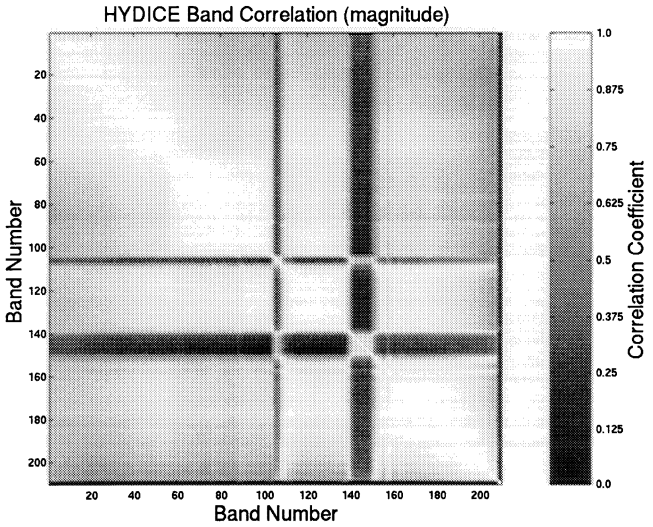

Figure 1. Band Correlation (desert scene)

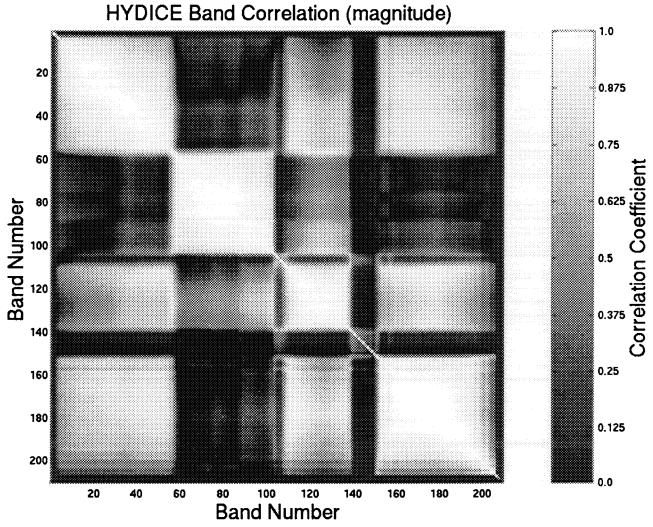

Figure 2. Band Correlation (forest scene)

Figures 1 and 2 show the magnitude of the correlation coefficients between bands in the two HYDICE scenes. The gaps in the correlation structure due to the water bands are evident in both images. The correlation coefficients were predominantly positive in value, with no significant negative correlations present. It is not known what effect calibration procedures had on

the correlation structure without access to data straight from the sensor. For both scenes, significant correlation structure still exists and may be exploited by the compression algorithms.

3. COMPRESSION ALGORITHMS

The algorithms employed in the compression of the HYDICE scenes utilized proposed extensions to the "baseline" JPEG-2000 compression algorithm described in [8,6]. All experiments were conducted using Verification Model software (VMs 7.x) [9,10,11]. The VM software provides a common framework in which all contributing members of the JPEG committee perform core experiments. During the course of this study, new versions of the VM software were released that addressed minor bugs or improved ease of use. For each experiment performed, it was ascertained that the VM software being used was operating properly.

Figure 3 is a block diagram of the VM 7.2 encoder. Dashed boxes represent portions of the encoder that are Part II specific (note that the trellis-coded quantizer (TCQ) is Part II specific as well). Images with more than one band are referred to as "multiple component" in JPEG-2000. An input multiple component image may be optionally processed by a component decorrelation transform. Allowed transforms include linear mixing of image components and wavelet transforms. In this work the Karhunen-Loeve Transform (KLT) served as a linear mixing transform and the reversible (5,3) integer wavelet [6] served as the wavelet transform in the spectral dimension.

After processing by the component decorrelation, the transformed components were wavelet processed in the spatial dimension. Depending upon which algorithm was used in a given experiment, either the (5,3) wavelet or a (9,7) floating-point wavelet [6] was employed. The wavelet coefficients were then quantized using either a deadzone scalar quantizer or TCQ. The resulting quantization indices in each wavelet subband were then collected into blocks. Each block of quantization indices was encoded using adaptive arithmetic bitplane coding. Blocking the quantization indices within each wavelet subband allows for more flexibility and better control over the rate-distortion progression in the bitstream, at the cost of some small signaling overhead and possibly coding efficiency.

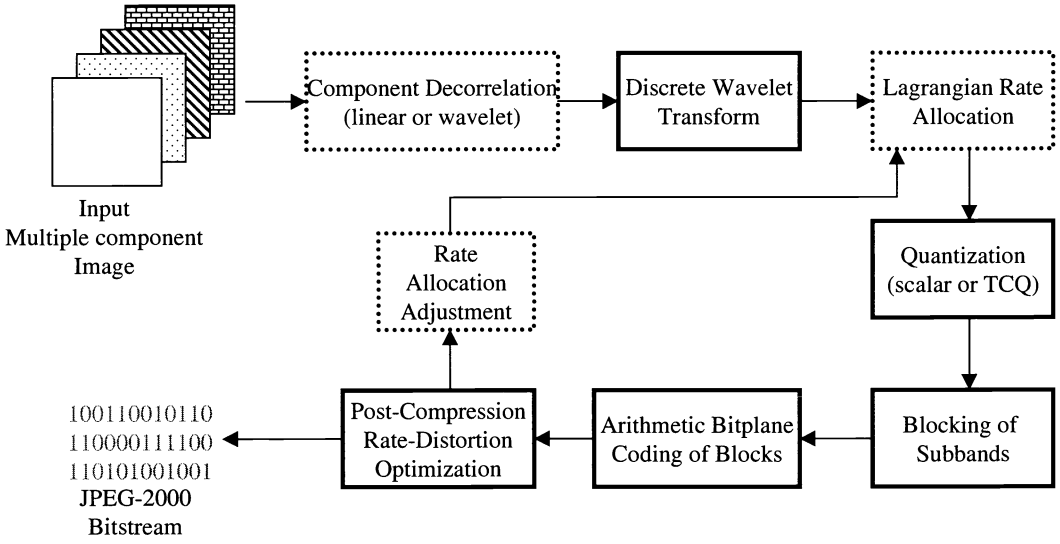

Figure 3. Block Diagram of VM 7.2 Encoder

The arithmetic bitplane coder uses fractional bitplane techniques [12,13,4] to encode a given bitplane in three coding passes. Bitplanes are encoded from most significant bit (MSB) to least significant bit (LSB). The three coding passes are called: *Significance Propagation*, *Magnitude Refinement* and *Cleanup*. The *Significance Propagation* pass encodes bits that have significant neighbors in an eight-connected neighborhood, but that are not yet significant themselves (i.e., their most significant "1" bit has not been encountered yet). The *Magnitude Refinement* pass encodes bits in indices that have already been deemed significant. Finally, the *Cleanup* pass encodes all other bits in the bitplane. All passes use context-based

adaptive arithmetic encoding. The *Significance Propagation* and *Cleanup* passes use run-length encoding to reduce the number of arithmetic symbols encoded.

After compression of the data, the bitstream contributions from each block are ordered and truncated. This is done in an optimal fashion to minimize distortion subject to a rate constraint. If Lagrangian rate control is used, the bitrate of the encoded file is checked against the desired bitrate. If they do not match within user-specified tolerances, the rate allocation is adjusted, and the data quantized and encoded again. Other rate control procedures in VM 7.x do not need this iteration step. The following sections describe in more detail the three algorithms investigated in this work.

3.1. 3D Wavelet

The 3D wavelet algorithm [14] employs the reversible (5,3) integer wavelet transform in the spatial and spectral dimensions. For the (5,3) filter, the quantization step size is set equal to one (no quantization). This algorithm achieves "quantization" by simple truncation of the encoded bitstream. The bitstream generated by the 3D wavelet algorithm is fully embedded and optimized for SNR scalability. This means that collections of bitplanes from subband blocks are ordered in the output bitstream in such a fashion as to maximize the SNR of the decoded image when the bitstream is truncated. Specifically, we used the "layer-resolution-component-position progression" (LRCP) to order the data [6]. The 3D wavelet algorithm is the only algorithm of the three investigated in this work that allows for lossless compression. In terms of computational complexity, it is the least complex of the three. It is also the simplest 3D compression algorithm in the JPEG-2000 VM.

3.2. KLT-Scalar

The KLT-scalar algorithm uses a KLT for spectral decorrelation and deadzone scalar quantization. The floating-point (9,7) wavelet filter is used for wavelet analysis in the spatial dimensions. The KL transform matrices were externally generated and supplied to the VM software. The inverse transform matrix (transpose of the forward transform matrix) was sent as four-byte floating-point numbers in a comment tag in the encoded bitstream as overhead. Its size has therefore been included in all rate computations. The deadzone scalar quantizer has a central "deadzone" bin whose width is twice that of all other quantization bins. When encoding an image, a single quantization step size is specified for the entire file. This single quantization step size is adjusted for each subband in the wavelet decomposition to take into account how quantization error in each subband is amplified during the synthesis process. The KLT-scalar algorithm generates an embedded scalable bitstream but it is not lossless.

3.3. KLT-TCQ

The KLT-TCQ algorithm is by far the most complex algorithm in the JPEG-2000 VM. Wavelet coefficients are analyzed to determine which of five different generalized-Gaussian densities best approximate their statistics. This information, along with coefficients variances, relative subband sizes, subband energy weights, and quantizer rate-distortion models, is used by the Lagrangian rate allocator to determine quantizer step sizes for each subband that minimize distortion subject to a rate constraint [15]. The trellis quantizer is a more advanced type of quantizer that utilizes four scalar quantizers with memory [13, 15]. A rate iteration control loop is used to adjust the rate allocation until the desired output bitrate is met within user-specified tolerances.

4. EXPERIMENTAL RESULTS

An uncompressed data cube for either HYDICE scene is 120,960,000 bytes (115.36 Mbytes) of data. Compressed and decoded hyperspectral cubes were generated at bitrates of 4.0, 2.0, 1.0, 0.5, and 0.25 bits/pixel/band (bpppb) for both HYDICE scenes (desert and forest) and for each compression algorithm, a total of 30 cubes. These bitrates correspond to compression ratios of 3, 6, 12, 24, and 48:1 relative to the image bitdepth of the sensor. They were chosen for comparative purposes [1] with prior exploitation studies [16][1]. Calibration procedures applied to the sensor data slightly increases the bitdepth. Since it is difficult to truly estimate the bitdepth of the data due to the calibration procedures, we consider the data as 12-bit when giving compression ratios. The data is processed as 16-bit signed integers (there are a few values negative values) within the VM software.

In experiments using the 3D wavelet algorithm, a lossless encoded file was generated for each scene. Each encoded file was truncated to five different lengths corresponding to the desired bitrates and then decoded. This very desirable feature is the

[1] Note that in the companion paper to this work [1], only the 3D wavelet and KLT-TCQ algorithms were compared in the exploitation study.

chief advantage of this technique. The KLT-Scalar algorithm was used to create two cubes as well. A base quantization step size of 2^{-16} was used for both scenes resulting in compressed files of 5.80 bpppb (43,811,370 bytes) and 5.48 bpppb (41,418,636 bytes) for the desert and forest scenes respectively. These files were also truncated to the appropriate lengths for the desired bitrates and decoded to generate 10 cubes. The KLT-TCQ algorithm was used to generate 10 encoded files one for each desired rate and scene. The file size tolerances were set such that the resulting output file size was never larger than that corresponding to the desired rate. The rate iteration control loop was able to create files whose rates were within 0.01 bpppb of the desired rate.

The VM software does not provide the capability to create KLT matrices; these matrices were generated externally and provided to the VM software for use in its component mixing stage. The eigenvector matrix of the KLT was derived from the spectral covariance matrix of all 210 spectral bands. Deriving such large KLT matrices may not be practical in many situations, and future work in this area will include spectral segmentation prior to KLT generation in order to reduce computational complexity. Figure 4 shows the eigenvalues (eigenband variances) for both the desert and forest scenes. In the desert scene, 99% of the spectral energy is contained in the first 5 eigenbands; in the forest scene, 99% of the spectral energy is contained in the first 7 eigenbands.

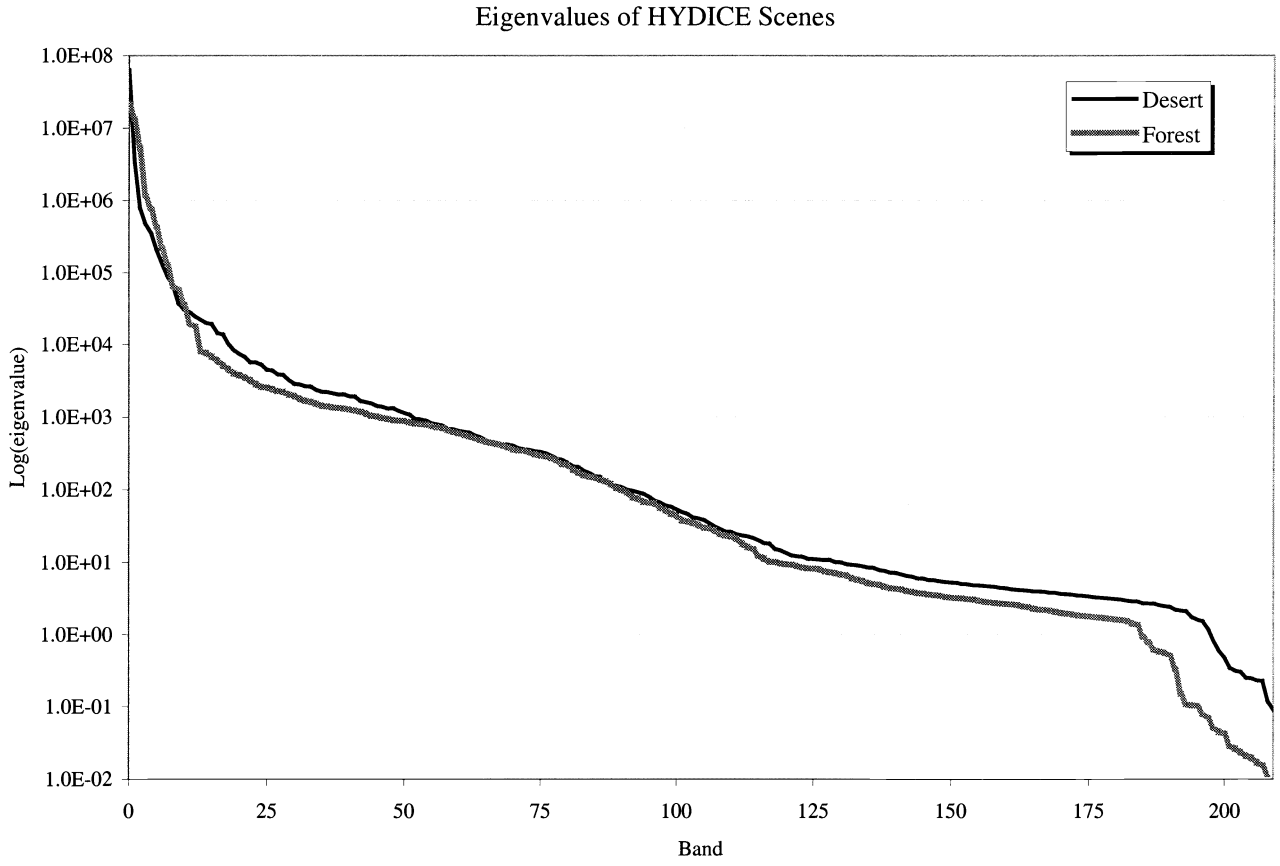

Figure 4. Eigenvalues of the HYDICE Desert and Forest Scenes

Figures 5 and 6 show the per-band RMSE and PSNR performance of the 3D wavelet algorithm for both HYDICE scenes. The PSNR figures were computed as:

$$PSNR = 10\log_{10}\left(\frac{2^{16}-1}{\sigma_e^2}\right)$$

where, σ_e^2, is the variance of the error between the original data and its lossy reconstruction. Again it is difficult to know the proper bitdepth to use in this calculation. Given that there are data samples in the sixteenth bitplane, we chose to use 2^{16}. Figures 7 and 8 show the per-band RMSE and PSNR performance of the KLT-TCQ with Lagrangian rate allocation algorithm for both HYDICE scenes. The KLT-Scalar algorithm performance was between the other two algorithms, although its performance was much closer to the KLT-TCQ than the 3D wavelet.

Looking at Figures 5 through 8, we see that the RMSE and PSNR performance for both algorithms is better on the high numbered spectral bands. The dynamic range of these bands is much less than that of the low numbered bands. In the desert scene bands 0-68 have a dynamic range on the order of [-25000, 25000]. The remaining bands have dynamic ranges of half this or less, with many bands having a range of one or two thousand counts (water-absorption bands have very small dynamic range). In these bands, the RMSE and PSNR figures will be very good, even at low bitrates due to the limited dynamic range. Comparing the two algorithms in these figures, we see that the KLT-TCQ algorithm performs better than 3D wavelet in terms of RMSE and PSNR for all spectral bands in both scenes except for band 64 in the desert scene. In this particular band the 3D wavelet results are ≈ 0.2 dB better, and the RMSE is reduced by ≈ 0.45.

Figures 9 and 10 are plots of the total RMSE and PSNR, computed jointly across all spectral bands, for the desert and forest scenes respectively. All three algorithms are included on these plots. One can clearly see that the performace of the KLT-based algorithms is improved over that of the 3D wavelet algorithm. The performance difference between the KLT-Scalar and KLT-TCQ algorithms is greatest at high rates (6.45 dB desert, 5.26 dB forest at 4.0 bpppb). At low rates the performance difference is small (0.06 dB desert, 0.17 dB forest at 0.25 bpppb). The performance gain between the KLT-TCQ and the 3D wavelet algorithms is 7.61 dB desert, 7.88 dB forest at 4.0 bpppb and 3.85 dB desert, 5.21 dB forest at 0.25 bpppb.

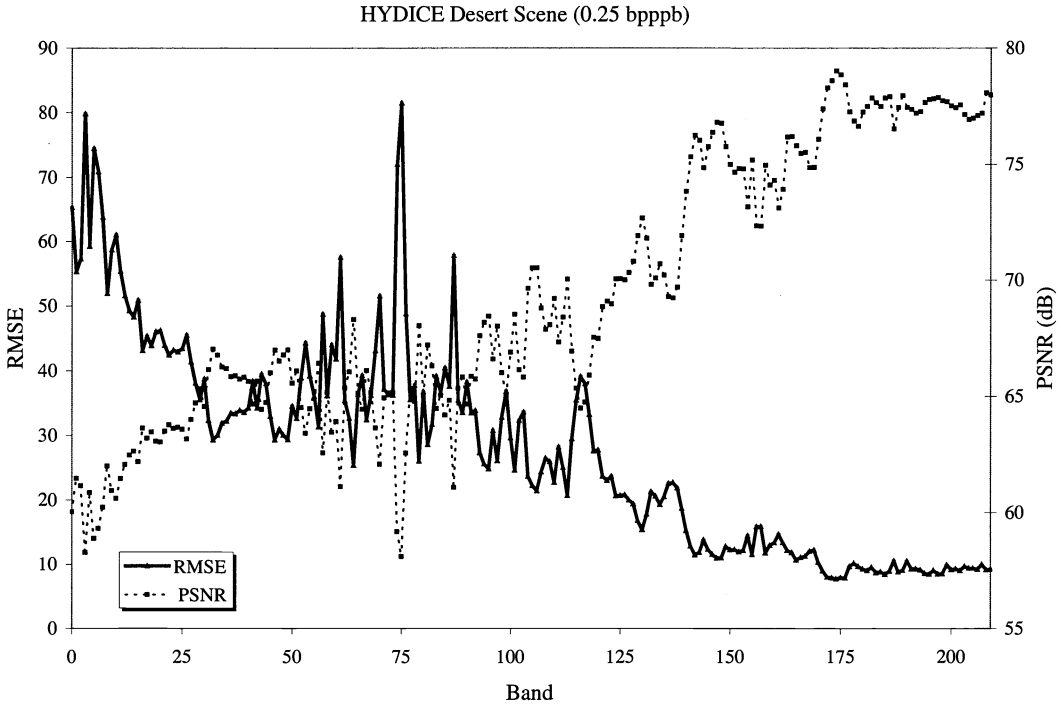

Figure 5. 3D Wavelet RMSE and PSNR Figures per Band (HYDICE Desert Scene at 0.25 bpppb)

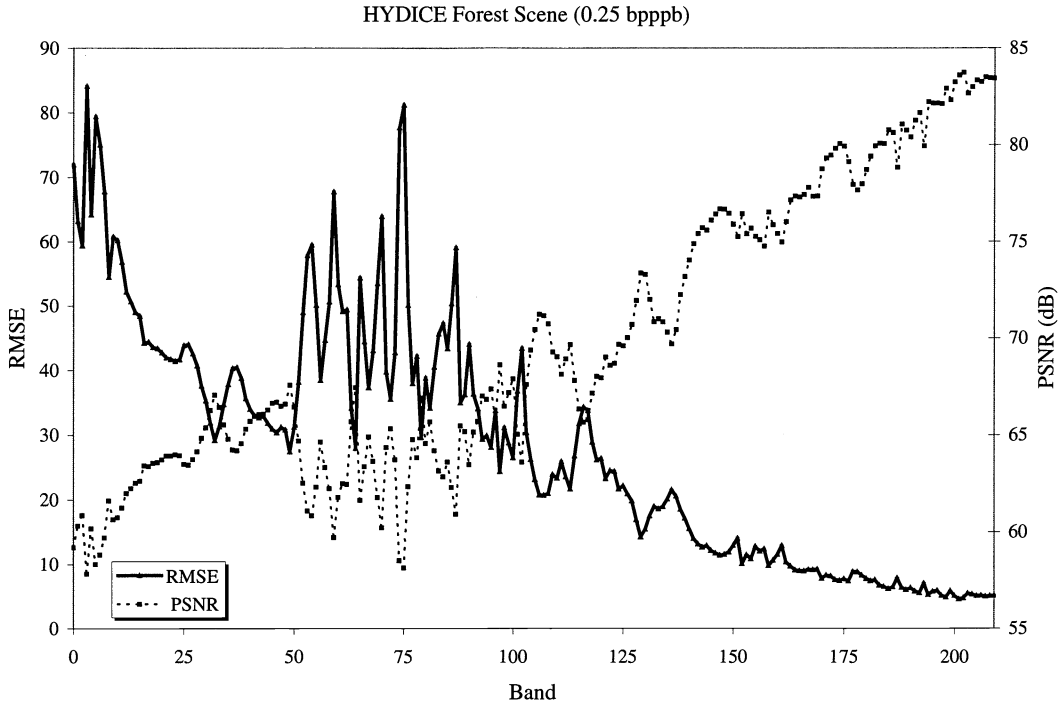

Figure 6. 3D Wavelet RMSE and PSNR Figures per Band (HYDICE Forest Scene at 0.25 bpppb)

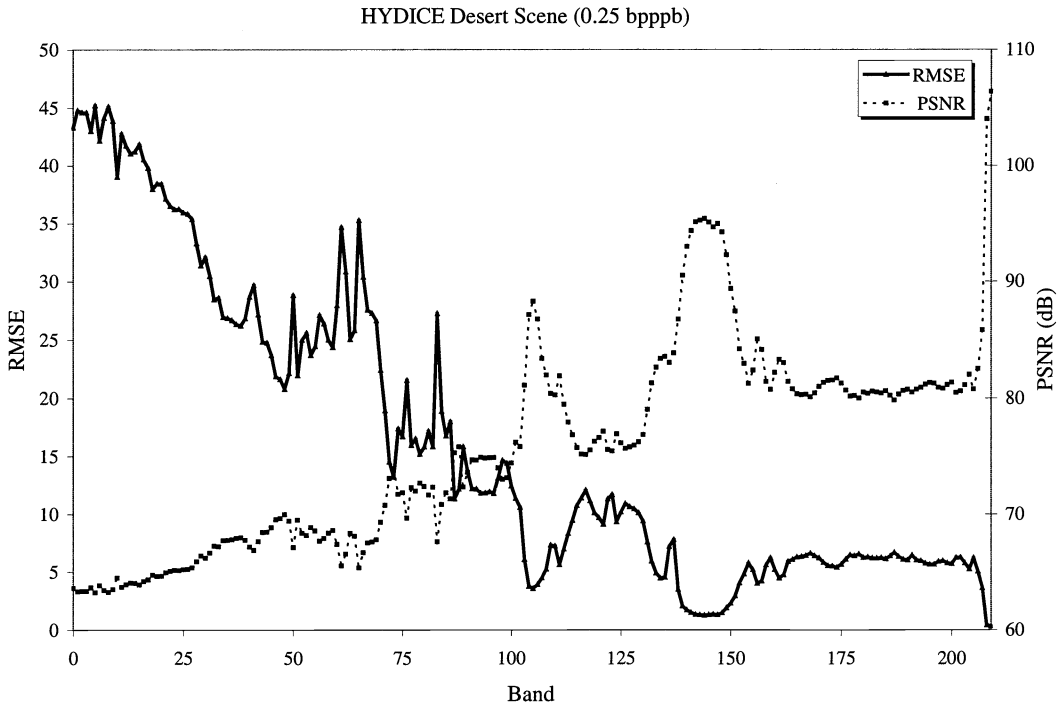

Figure 7. KLT-TCQ RMSE and PSNR Figures per Band (HYDICE Desert Scene at 0.25 bpppb)

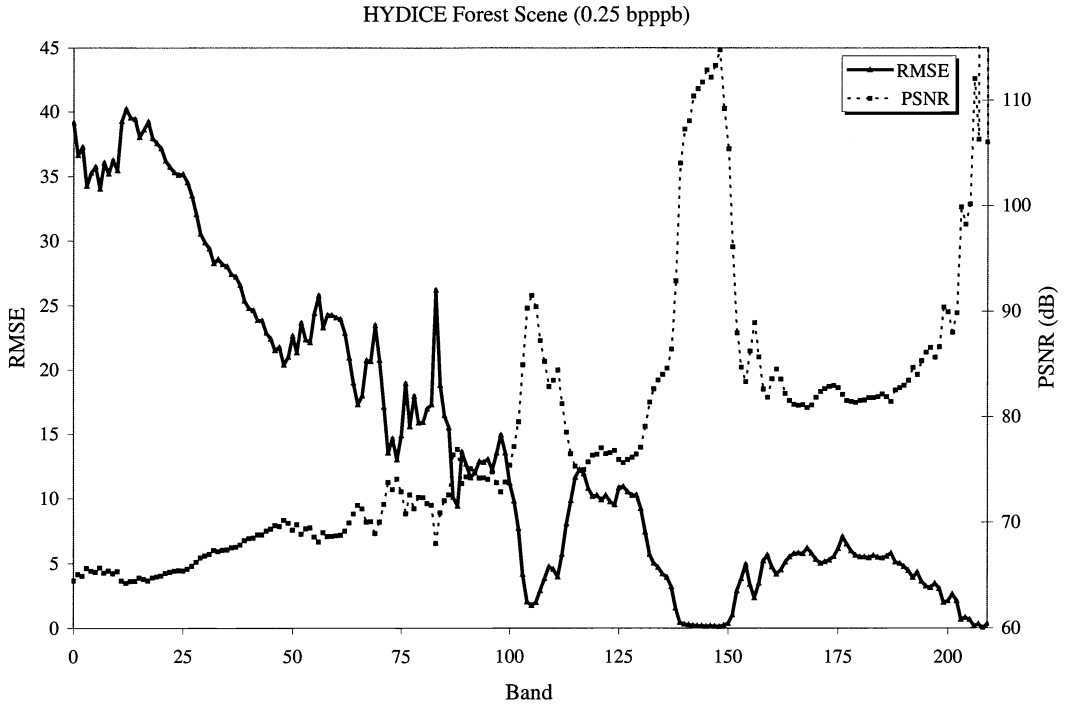

Figure 8. KLT-TCQ RMSE and PSNR Figures per Band (HYDICE Forest Scene at 0.25 bpppb)

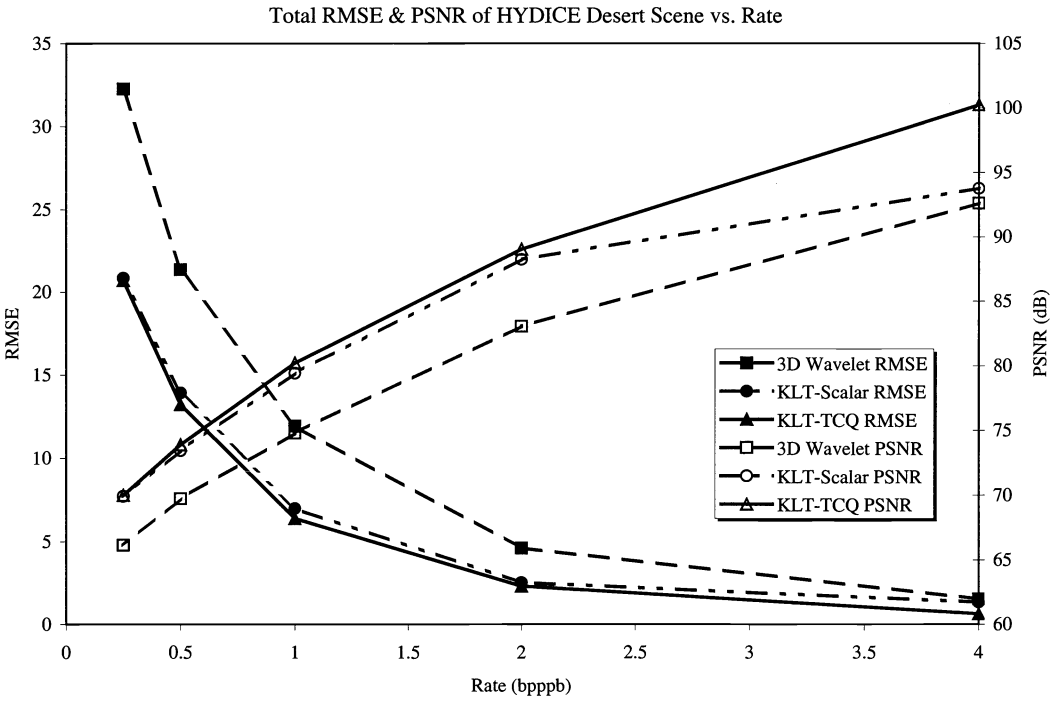

Figure 9. Total RMSE and PSNR vs. Rate (HYDICE Desert Scene, all algorithms)

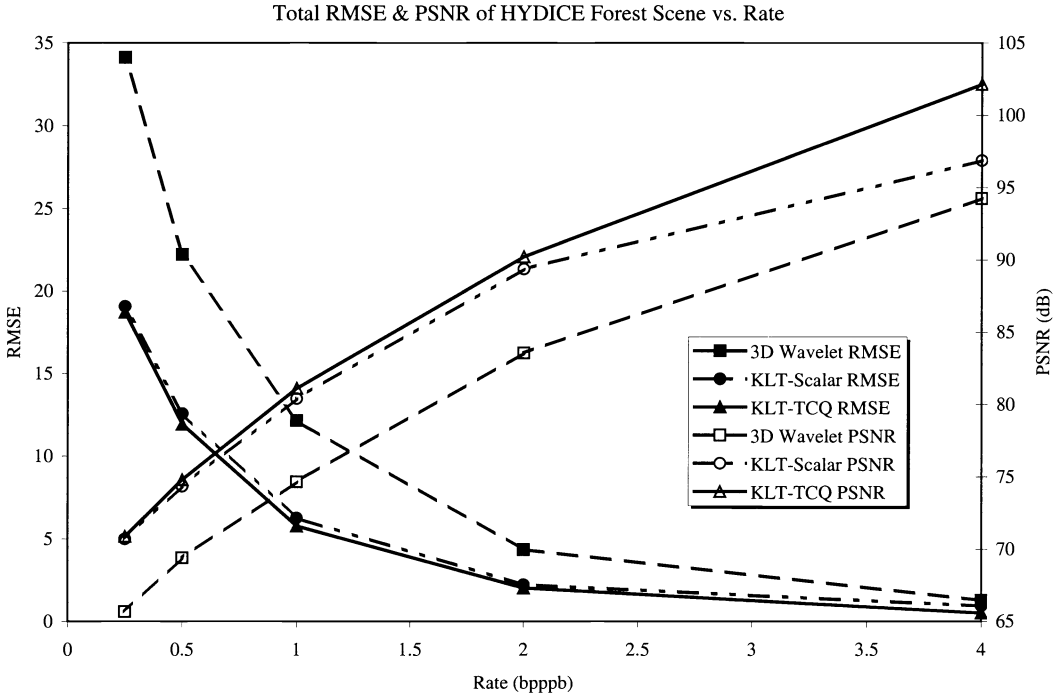

Figure 10. Total RMSE and PSNR vs. Rate (HYDICE Forest Scene, all algorithms)

5. CONCLUSIONS

In terms of RMSE and PSNR metrics, all three algorithms yielded excellent performance. It is not surprising that the KLT-based algorithms performed better than the 3D wavelet algorithm, the KLT is an image-specific decorrelation transform. It is somewhat surprising how little difference was seen in the performance between the KLT-TCQ and KLT-Scalar algorithms below 2.0 bpppb. If one were to choose between the two KLT-based algorithms on features and complexity merits, the KLT-Scalar algorithm would be preferable. Discounting the cost of generating and applying the KLT, the KLT-TCQ algorithm is still computationally expensive, due to its Lagrangian rate allocation and iterative rate control. If we similarly discount the KLT costs in the KLT-Scalar algorithm, its complexity is only slightly more than that of the 3D wavelet algorithm. Furthermore, the KLT-Scalar encoder *does* produce an embedded bitstream. Truncation of a single encoded file to a desired bitrate was used to generate all decoded output. While the KLT-TCQ bitstream can be progressively decoded, the benefits of TCQ are not realized unless all bitplanes are decoded within a subband block. The 3D wavelet algorithm does possess one feature that neither KLT-based algorithm has: lossy and lossless compression from a single embedded bitstream. If lossless compression is needed, it is the only algorithm to choose.

The results from this study suggest interesting future experiments. An exploitation comparison between KLT-TCQ and KLT-Scalar algorithms is needed to determine if the performance of both algorithms is similar for such tasks. The perceptual performance of TCQ is typically better than PSNR and RMSE metrics indicate. The efficacy of more general decorrelating transforms for classes of images needs to be determined. Given the 3.5 to 7 dB gap between the 3D wavelet algorithm and KLT-based algorithms, it may be possible to derive and catalogue decorrelation transforms for different image classes and maintain a performance advantage. This would alleviate the need to compute the KLT. Results seen during this study suggest that spectral band re-ordering and segmentation may improve compression performance or at least reduce complexity without performance loss. We plan to investigate a new algorithm based on the KLT for spectral decorrelation and the (5,3) integer wavelet for spatial decorrelation. Such a system would produce embedded bitstreams with lossy to near-lossless performance.

6. REFERENCES

1. Shen, S. S., and Kasner, J. H., "Effects of 3D wavelets and KLT-based JPEG-2000 hyperspectral compression on exploitation," To appear in Proceedings of Imaging Spectrometry VI, Vol. 4132.

2. Vaughn, V. D., and Wilkinson, T. S., "Systems considerations for multispectral image compression designs," *IEEE Signal Processing Magazine*, vol. 12, no. 1, pp. 19-31, Jan. 1995.

3. Said, A., and Pearlman, W. A., "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.

4. Taubman, D., "High performance scalable image compression with EBCOT," to appear in *IEEE Transactions on Image Processing*.

5. Zandi, A., Allen, J. D., Schwartz, E. L., and Boliek, M., "CREW: Compression with reversible embedded wavelets," *Proceedings of the Data Compression Conference*, Snowbird, UT, pp. 212-221, March 1995.

6. Boliek, M., Christopoulos, and C., Majani, E., "JPEG 2000 Part I final committee draft version 1.0," *ISO/IEC/JTC 1/SC29/WG1 N1646*, April 2000.

7. Basedow, R., Silverglate, P., Rappoport, W., Rockwell, R., Rosenburg, D., Shu, K., Whittlesey, R., and Zalewski, E. (1992) "The HYDICE instrument design," *Proceedings of the International Symposium on Spectral Sensing Research*, Vol. 1, pp. 430-445.

8. Christopoulos, C., "JPEG 2000 verification model 7.0 (technical description)," *ISO/IEC/JTC 1/SC29/WG1 N1684*, April 2000.

9. Flohr, T., "VM7.0 software," *ISO/IEC/JTC 1/SC29/WG1 N1685*, April 2000.

10. Flohr, T., "VM7.1 software," *ISO/IEC/JTC 1/SC29/WG1 N1691*, April 2000.

11. Flohr, T., "VM7.0 software," *ISO/IEC/JTC 1/SC29/WG1 N1699*, May 2000.

12. Ordentlich, E., Weinberger, M., and Seroussi, G., "A low-complexity modeling approach for embedded coding of wavelet coefficients," *Proceedings of IEEE Data Compression Conference*, pp. 408-417, 1998.

13. Sementilli, P. J., Bilgin, A., Kasner, J. H., and Marcellin, M. W., "Wavelet TCQ: submission to JPEG-2000," (invited paper), *Proceedings of SPIE, Applications of Digital Image Processing*, pp. 2-12, July 1998.

14. Bilgin, A., Kasner, J. H., and Marcellin, M. W., "Hyperspectral image compression using JPEG-2000," To appear in *Proceedings of ICASSP 2000*.

15. Kasner, J. H., Marcellin, M. W., and Hunt B. R., "Universal Trellis Coded Quantization," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1677-1687.

16. Shen, S. S., and Beard, B., "Effects of hyperspectral compression on nonliteral exploitation," proceedings of SPIE on imaging spectrometry, vol. 3439, 1998.