

# EVALUATION OF JP3D FOR LOSSY AND LOSSLESS COMPRESSION OF HYPERSPPECTRAL IMAGERY

Jing Zhang<sup>\*†</sup>, James E. Fowler<sup>†</sup>, Nicolas H. Younan<sup>†</sup>, and Guizhong Liu<sup>\*</sup>

<sup>\*</sup>School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China

<sup>†</sup>Department of Electrical and Computer Engineering, Geosystems Research Institute,  
Mississippi State University, USA

## ABSTRACT

*The performance of the recent JPEG2000 Part 10 standard, known as JP3D, is evaluated for the lossy and lossless compression of hyperspectral imagery. Experimental results using a Karhunen-Loève transform (KLT) for spectral decorrelation and a 2D wavelet transform for spatial decorrelation compare the performance of JP3D against 2D JPEG2000 as specified by Part 2 of the standard. JP3D is used with both the 2D arithmetic-coding contexts as specified in the JP3D standard as well as non-standard experimental 3D contexts. Results reveal that, while for lossless coding, JP3D very slightly surpasses the performance of JPEG2000 Part 2, for lossy coding, JP3D fails to match the rate-distortion performance of the 2D Part-2 coder.*

**Index Terms**— JP3D, JPEG2000, hyperspectral compression

## 1. INTRODUCTION

The JPEG2000 standard has been widely used for the compression of hyperspectral imagery for both lossy as well as lossless representation. Strictly speaking, as defined by Parts 1 and 2 of the standard [1, 2], JPEG2000 is a 2D image coder designed for the coding of 2D still images. That said, Part 2 provides indirect support for the coding of 3D volumetric imagery such as hyperspectral data in that 1D transforms can be applied across image components such that, when coupled with 2D spatial transforms, a 3D transform decomposition results. Additionally, post-compression rate-distortion (PCRD) optimization is applied across all codeblocks in all components to effectively allocate rate across the entire image volume. However, the fundamental embedded-block-coding process in Part-2 compliant coders takes place on 2D codeblocks defined within 2D image components as prescribed in the original 2D EBCOT coder [3].

In an effort to provide a true 3D implementation of JPEG2000 for volumetric imagery, Part 10 of the JPEG2000 standard [4–6] has recently been established. Part 10, or “JP3D” as it is commonly known, is, in essence, a straightforward extension of Parts 1 and 2 to realize coding for volumetric data—JP3D provides the same functionality for 3D datasets as Parts 1 and 2 do for 2D data. That is, in addition to permitting true 3D dyadic wavelet transforms, JP3D coding is based on embedded block coding using 3D codeblocks.

Thus far, the JP3D standard has been mainly used for the compression of 3D medical imagery (e.g., [7, 8]). However, given that

hyperspectral imagery can also be considered to take a volumetric form, it is natural to consider JP3D for hyperspectral compression. The objective of this paper is thus to evaluate JP3D for the compression of hyperspectral imagery. We first explore issues of 3D transforms and 3D codeblock coding as pertaining to JP3D before presenting experimental results for lossy and lossless compression of the popular AVIRIS radiance datasets. We find that, while JP3D very slightly surpasses the performance of JPEG2000 Part 2 for lossless coding, JP3D fails to provide any gains in rate-distortion performance over Part-2 coding for lossy compression.

## 2. JP3D TRANSFORM ISSUES

In any coding of hyperspectral imagery, the first step is typically the application of some 3D decorrelating transform to the volume data. Discrete wavelet transforms (DWTs) are widely used in this task. Suitable 3D DWTs include two types—the 3D dyadic DWT and the 3D packet DWT. The 3D dyadic DWT, a straightforward extension to three dimensions of the 2D dyadic transform widely used in still-image coding, consists of the recursive decomposition of the 3D baseband subband. On the other hand, the packet transform consists of a 1D spectral DWT followed by a 2D dyadic DWT applied spatially. While JPEG2000 Part 2 supports the packet-transform decomposition structure, the 3D dyadic transform is not supported. However, JP3D adds support for the dyadic transform; in addition, since Part 10 subsumes Part 2, a JP3D-compliant coder can use either the dyadic or packet transform structure. That said, it has been shown that the 3D packet wavelet transform generally outperforms the dyadic transform by a significant margin for hyperspectral imagery (e.g., [9]). Thus, the dyadic-transform capability of JP3D is not of further consideration here.

A favored paradigm for lossy compression of hyperspectral imagery consists of spectral Karhunen-Loève transform (KLT) followed by a 2D spatial DWT (e.g., [10, 11]). It has been shown that KLT-based spectral decorrelation achieves rate-distortion efficiency substantially superior to that of the DWT-based spectral transform (e.g., [10]). Consequently, in the experimental results below, we focus on this KLT+DWT 3D transform to evaluate performance.

For lossy coding, the spectral KLT is implemented in its usual floating-point form and is thus an irreversible transform covered by Part 2 (and therefore Part 10) of the JPEG2000 standard. However, lossless compression requires a reversible KLT that maps integers to integers. Such a reversible integer approximation to the KLT was proposed in [12, 13] based on reversible matrix factorizations originating in [14]. The key to this integer KLT is that the KLT

This work was conducted while J. Zhang was visiting the Geosystems Research Institute at Mississippi State University as a joint Ph.D. candidate funded by China Scholarship Council under the State Scholarship Fund.

transform matrix  $A$  can be factored as

$$A = PLUS \quad (1)$$

where  $L$  and  $S$  are lower triangular matrices,  $U$  is an upper triangular matrix, and  $P$  is a reversible permutation matrix. Application of these four matrices can be achieved by lifting steps, while the inclusion of a rounding operation achieves a reversible integer-to-integer mapping.

In experimental results below, all approaches use the same 3D transform, namely, a spectral KLT followed by a 2D spatial DWT. For lossy compression, the spatial transform is the ubiquitous 9/7 DWT from JPEG2000 Part 1, while the spectral transform is the traditional floating-point KLT. For lossless compression, on the other hand, the reversible, integer-valued KLT [12, 13] is used spectrally, while the spatial transform is the reversible 5/3 from JPEG2000 Part 1. We note that the reversible KLT is not compliant with the JPEG2000 Part 2 multicomponent transform; however, it is a straightforward process to include this transform as a “preprocessing” step prior to encoding, and this is the approach we take below for lossless compression.

### 3. JP3D BLOCK CODING

JP3D follows the same block-coding paradigm as Part 1 of the JPEG2000 standard—bitplane coding is applied with arithmetic coding conditioned on a multitude of specialized contexts. JP3D simply casts these operations onto 3D codeblocks. As illustrated in Fig. 1, codeblocks in JPEG2000 Part 2 are 2D blocks which are processed in a stripe-based scan. In contrast, codeblocks in JP3D are 3D blocks with an alternated scan order as illustrated in Fig. 2. Specifically, in a 3D block, bitplanes are processed from the most significant bitplane to the least significant bitplane, while, within each bitplane, each “slice” is scanned in turn. In each slice, the scan order is the same as in a 2D block.

Although block coding takes place within 3D blocks, the JP3D standard specifies that context-based arithmetic coding follows the 2D context definitions of Part 1 of the standard despite the 3D nature of the codeblocks. There has been some work in devising truly 3D arithmetic-coding contexts for JP3D (e.g., [7]), and it has been observed that significant gains can arise in certain cases for 3D contexts that derive state information from all three directions (e.g., [15]). Yet, it is difficult to design a context model that is suitable across a variety of different types of data—for this reason, only 2D contexts were included in the JP3D standard [6]. However, should ongoing work produce such a generic model, it is anticipated that the JP3D standard will be extended to encompass such improved 3D context models [6].

In experimental results below, we use the JP3D Verification Model (JP3D VM) Version 1.1 as included in OpenJPEG Version 1.3 (see <http://www.openjpeg.org>). The JP3D VM includes the 2D arithmetic-coding contexts from Part 1 of the standard; these contexts are referred to as “2EB” below. In addition, the JP3D VM includes a set of experimental 3D context models (“3EB”) which are not currently compliant with the Part 10 standard.

The Part-1 compliant 2EB contexts divide the 8 coefficients neighboring the current coefficient into three classes (horizontal, vertical, diagonal). This generates 9 significance contexts, 5 sign-coding contexts, and 3 magnitude-refinement contexts. On the other hand, the 3EB contexts consider the information from 26 neighboring coefficients not only in the same spectral band but

also in adjacent spectral bands. 3EB divides the 27 coefficients into seven classes (horizontal, vertical, diagonal, spectral, spectral+horizontal, spectral+vertical, spectral+diagonal), generating 15 significance contexts, 6 sign-coding contexts, and 3 magnitude-refinement contexts.

### 4. EXPERIMENTAL RESULTS

We now experimentally evaluate the performance of JP3D against the usual use of JPEG2000 for hyperspectral imagery, i.e., a Part-2 implementation with a spectral KLT (e.g., [10]) which we denote here as “JP2K P2.” For JP3D, we consider both the 2EB and 3EB context models (denoted “JP3D+2EB” and “JP3D+3EB,” respectively).

We evaluate performance using the the 1997 AVIRIS radiance images; these images have 224 bands with an average spectral separation between bands of 10 nm and a spatial size of  $614 \times 512$ . We use scene 1 for each image and crop the datasets spatially from the upper-left corner to a size of  $512 \times 512$ . Performance at lossless compression is evaluated in the form of bitrate in bits per pixel per band (bpppb) calculated as the size of the compressed file divided by the number of pixels. The performance of lossy compression is evaluated in the form of a signal to noise ratio (SNR) at a fixed bitrate; we measure SNR as the log ratio of signal variance to mean squared error. Additionally, we measure the average spectral angle mapper (SAM), the angle between original and reconstructed hyperspectral pixel vectors as averaged over the dataset.

Table 1 reports results for lossless compression. The spatial DWT in all cases is a 5-level decomposition. We see that the three coders have similar performance, with JP3D+2B being slightly better than the other two. Thus, we conclude that the use of 3D block coding can improve lossless-coding results over the traditional Part-2 2D coder, but the improvement is very slight.

Tables 2 and 3 report results for lossy compression at a fixed bitrate of 1.0 bpppb. In all cases, a 4-level DWT is used spatially. Unlike the lossless case, lossy compression performance is soundly in favor of the 2D Part-2 coder, both in terms of SNR as well as SAM. Even amongst the JP3D coders, the 2D context (2EB) outperforms its 3D counterpart (3EB).

### 5. CONCLUSIONS

In this paper, we evaluated for hyperspectral imagery the recent JP3D standard which forms Part 10 of the JPEG2000 body of standards. We investigated both lossy and lossless compression, considering issues surrounding the 3D transform and 3D embedded block coding for use with JP3D in both cases. We employed the widely-used transform structure of a 1D KLT applied spectrally followed by 2D spatial DWT, with reversible, integer-valued transforms employed for lossless compression. For embedded block coding, we explored the use of the 2D arithmetic-coding contexts from JPEG2000 Part 1 that are required for use in JP3D; we also considered an experimental 3D context model included with the current JP3D VM but outside of JP3D-standard compliance. Experimental results revealed that, for the popular AVIRIS radiance images, JP3D provides lossless coding performance very similar to that of JPEG2000 Part 2, with the standard 2D context models providing a very slight coding gain. However, for lossy compression, JP3D was unable to match the rate-distortion performance of JPEG2000 Part 2 with either the 2D or 3D context models.

Whereas JPEG2000 in its Part-1 and Part-2 incarnation is essentially a 2D image coder, JP3D is designed to provide true 3D coding. However, JP3D is clearly oriented toward data that is isotropic in all directions. That is, JP3D was created with medical imagery specifically in mind; however, 3D medical datasets are typically isotropic in that all directions represent spatial dimensions. Hyperspectral imagery, with one dimension being a spectrum, is fundamentally different—apparently, JP3D is less effective for such anisotropic datasets, at least in its current form. In may be possible that arithmetic-coding context models specifically designed for the anisotropic nature of hyperspectral imagery could improve JP3D performance on such data; however, to the best of our knowledge, no such context models currently exist.

## 6. REFERENCES

- [1] *Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System*, ISO/IEC 15444-1, 2000.
- [2] *Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions*, ISO/IEC 15444-2, 2004.
- [3] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.
- [4] *Information Technology—JPEG 2000 Image Coding System—Part 10: Extensions for Three-Dimensional Data*, ISO/IEC 15444-10, 2004.
- [5] P. Schelkens, A. Munteanu, A. Tzannes, and C. Brislawn, “JPEG2000 Part 10 – Volumetric data encoding,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Kos, Greece, May 2006, pp. 3874–3877.
- [6] T. Bruylants, A. Munteanu, A. Alecu, R. Deklerck, and P. Schelkens, “Volumetric image compression with JPEG2000,” *SPIE Newsroom*, June 2007.
- [7] T. Bruylants, A. Alecu, T. Kimpe, R. Deklerck, A. Munteanu, and P. Schelkens, “An optimized 3D context model for JPEG2000 Part 10,” in *Medical Imaging 2007: Image Processing*, J. P. W. Pluim and J. M. Reinhardt, Eds. San Diego, CA: Proc. SPIE 6512, February 2007, p. 65124K.
- [8] T. Kimpe, T. Bruylants, Y. Sneyders, R. Deklerck, and P. Schelkens, “Compression of medical volumetric datasets: Physical and psychovisual performance comparison of the emerging JP3D standard and JPEG2000,” in *Medical Imaging 2007: Image Processing*, J. P. W. Pluim and J. M. Reinhardt, Eds. San Diego, CA: Proc. SPIE 6512, February 2007, p. 65124L.
- [9] J. E. Fowler and J. T. Rucker, “3D wavelet-based compression of hyperspectral imagery,” in *Hyperspectral Data Exploitation: Theory and Applications*, C.-I. Chang, Ed. Hoboken, NJ: John Wiley & Sons, Inc., 2007, ch. 14, pp. 379–407.
- [10] Q. Du and J. E. Fowler, “Hyperspectral image compression using JPEG2000 and principal component analysis,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201–205, April 2007.
- [11] B. Penna, T. Tillo, E. Magli, and G. Olmo, “Transform coding techniques for lossy hyperspectral data compression,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1408–1421, May 2007.

Table 1: Lossless Compression Performance

Dataset	Bitrate (bpppb)		
	JP2K P2	JP3D+2EB	JP3D+3EB
Cuprite	4.92	4.89	4.96
Jasper Ridge	4.86	4.83	4.90
Lunar Lake	5.00	4.97	5.00
Moffett	4.96	4.93	5.00
Average	4.94	4.90	4.97

Table 2: SNR Performance of Lossy Compression at 1.0 bpppb

Dataset	SNR (dB)		
	JP2K P2	JP3D+2EB	JP3D+3EB
Cuprite	54.13	53.31	53.28
Jasper Ridge	50.33	49.69	49.67
Lunar Lake	55.21	54.62	54.59
Moffett	50.92	50.42	50.38

Table 3: SAM Performance of Lossy Compression at 1.0 bpppb

Dataset	SAM (degrees)		
	JP2K P2	JP3D+2EB	JP3D+3EB
Cuprite	0.08	0.09	0.09
Jasper Ridge	0.14	0.15	0.15
Lunar Lake	0.07	0.08	0.08
Moffett	0.12	0.13	0.13

- [12] P. Hao and Q. Shi, “Reversible integer KLT for progressive-to-lossless compression of multiple component images,” in *Proceedings of the International Conference on Image Processing*, vol. 1, Barcelona, Spain, September 2003, pp. 633–636.
- [13] L. Galli and S. Salzo, “Lossless hyperspectral compression using KLT,” in *Proceedings of the International Geoscience and Remote Sensing Symposium*, vol. 1, Anchorage, AK, September 2004, pp. 313–316.
- [14] P. Hao and Q. Shi, “Matrix factorization for reversible integer mapping,” *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2314–2324, October 2001.
- [15] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, “Wavelet coding of volumetric medical datasets,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 3, pp. 441–458, March 2003.

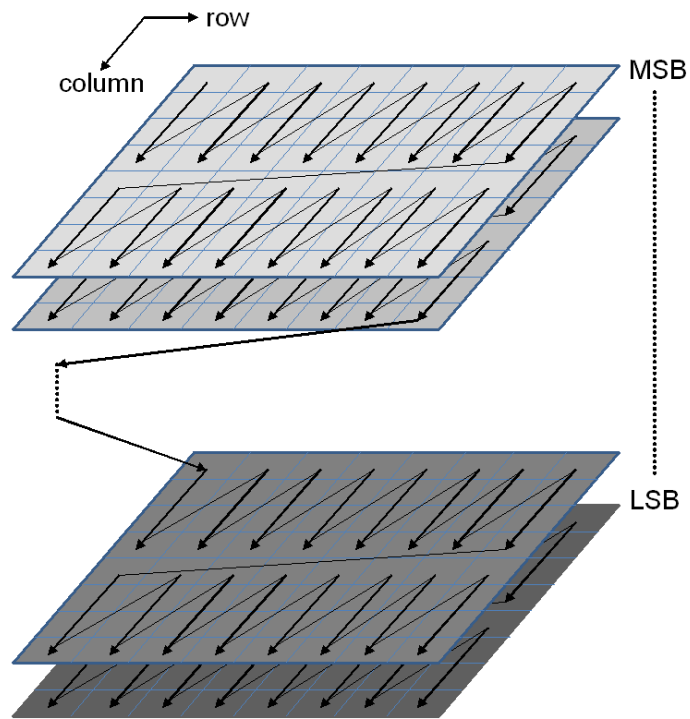


Figure 1: Embedded block coding of a 2D  $8 \times 8$  codeblock.

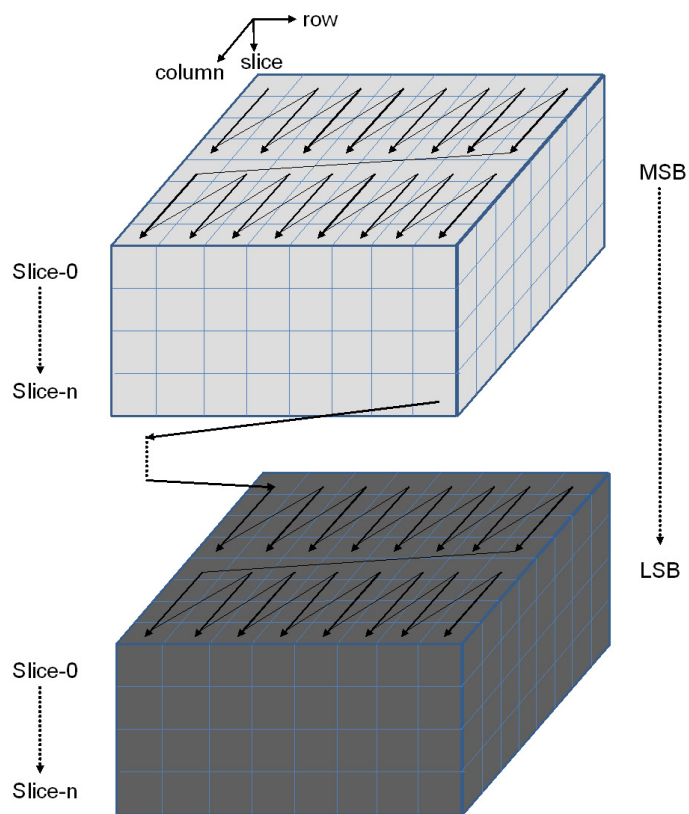


Figure 2: Embedded block coding of a 3D  $8 \times 8 \times 4$  codeblock consisting of four  $8 \times 8$  slices.