Hyperspectral Image Compression Using Hybrid Transform With Embedded Zero-Tree Wavelet And Set Partitioning In Hierarchical Tree

Ranganathan Nagendran, Arumugam Vasuki

Abstract : Hyperspectral images are three-dimensional representations comprising spatial and spectral dimensions. Hence, these images contain a huge volume of data, and compression is required for their efficient transmission, storage, and processing. We propose a hyperspectral image compression technique using a hybrid transform. Specifically, the integer Karhunen–Loève transform with clustering and tiling is applied to the spectral dimension to decorrelate the corresponding data. Then, the 2D integer discrete wavelet transform is applied over the spatial dimension to decorrelate the spatial data. In addition, the decorrelated wavelet coefficients are applied in the embedded zero-tree wavelet transform. Alternatively, the data are processed by an algorithm containing set partitioning in hierarchical tree. The wavelet transform and partitioning on the hybrid transform over hyperspectral images retrieve high peak signal-to-noise ratio, low number of bits per pixel per band, and fast computation time compared to similar approaches, with the partitioning showing the best results.

Index Terms : Hyperspectral image; embedded zero-tree wavelet (EZW); set partitioning in hierarchical tree (SPIHT); integer Karhunen–Loève transform; integer discrete wavelet transform.

1. INTRODUCTION

Hyperspectral imaging has been widely used for large-scale remote sensing applications such as soil type studies, recognition and identification of the surface and constituents of the atmosphere, monitoring agricultural and forest status, environmental studies, and military surveillance. Hyperspectral images have hundreds of narrow and contiguous spectral bands and possess data regarding the complete reflectance spectrum of the observed region. For instance, the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) produces up to 224 spectral bands about 10 nm wide over the spectrum from 400 to 2500 nm[1]. Therefore, hyperspectral images have a huge volume of data by the generation of hundreds of bands, making necessary their compression for efficient transmission, storage, and processing. Image compression can be either lossless or lossy, where the latter retrieves an approximation of the original data after decompression, whereas the former exactly reproduces the original data. In recent years, several approaches have been proposed to compress hyperspectral images and can be classified into predictive coding, vector quantization, and transform-based coding. Roger & Cavenor[2] used adaptive differential pulse-code modulation for lossless compression of hyperspectral images and experimented with five linear spatial, spectral, and spatial–spectral predictors optimized using least squares. Then, the residuals were encoded using variable-length coding. Aiazzi et al.[3] introduced the fuzzy differential pulse-code modulation based on the fuzzy c-means algorithm, which groups the causal neighborhoods of each pixel, calculates the optimized coefficient predictor for each cluster, and finally calculates the weighted sum of all predictors. Wu & Memon [4] extended their 2D context-based adaptive lossless image coding to three dimensions (Wu & Memon,)[5] for application to hyperspectral images. This method is based on the dependence among fundamental neighboring pixels, and the predictor is switched between intra- and inter-bands. Vector quantization is a four-step approach that is used when a high compression ratio is required[6]. First, the image is divided into blocks. Then, a codebook is designed as the core for compression. Subsequently, the codebook is generated using the generalized Lloyd's and Linde–Buzo–Gray algorithms, where the latter retrieves a binary index. Finally, the closest code vector is considered, and the corresponding codewords and codebook are the output [6]. Ryan & Arnold [7] extended this approach to the mean-normalized vector quantization for lossless compression. Vector quantization operates on spectral blocks only, thus neglecting any spatial correlations. Merging vector quantization with transform-based coding has resulted in methods such as the embedded zero-tree wavelet (EZW), set partitioning in hierarchical tree (SPIHT), and set partitioning embedded block (da Silva, Sampson, & Ghanbari, Knipe, Li, & Han, Chao & Gray)[8][9][10]. Transform-based coding incorporates statistical properties such as energy compaction and decorrelation components and comprises techniques such as the discrete cosine transform, discrete Fourier transform, Karhunen–Loève transform (KLT), and discrete wavelet transform (DWT). For instance, the JPEG (Joint Photographic Experts Group) specification uses the 8 × 8 discrete cosine transform, and the more recent JPEG2000 uses the 2D DWT. The compression of hyperspectral images using JPEG2000 and the performance of different transform techniques, including the wavelet transform, discrete cosine transform, and KLT, were studied by Penna et al. [11]. Shapiro [12] proposed EZW coding using the wavelet transform to compress images, and the method can be extended to three dimensions. Likewise, Bilgin, Zweig, & Marcellin [13] proposed a 3D image compression algorithm based on the wavelet transform, and Kim, Xiong, & Pearlman [14] proposed the 3D SPIHT. We propose a hybrid image compression technique that uses one dimension for spectral representation and two dimensions for spatial representation. First, the KLT is applied to remove the correlations among contiguous highly correlated spectral bands using clustering and tiling. Then, the KLT bands are PLUS factored into matrices P (permutation), L (unit lower), U (upper triangular), and S (a special matrix) to obtain a reversible lossless image, conforming the integer KLT bands. The PLUS matrices are used as overhead information during decoding. Finally, the 2D integer DWT is applied for each integer KLT band to remove spatial redundancies[15]. After performing

the hybrid transform, either the EZW or SPIHT algorithm are applied to encode the transformed hyperspectral images aiming for performance evaluation. Figures 1 shows the block diagram of the proposed hybrid-transform-based image compression, and Figure 2 illustrates a pixel vector of a hyperspectral image. In the sequel, we detail the integer KLT and 2D integer DWT. Then, we introduce the EZW algorithm for hyperspectral images and detail the SPIHT algorithm. Experimental results verify the effectiveness of the proposed hybrid transform to compress hyperspectral images.

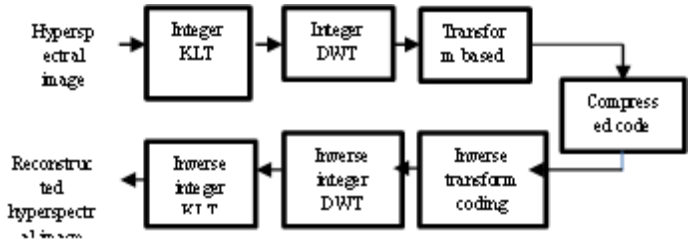


Figure 1. *Block diagram of proposed hybrid-transform-based hyperspectral image compression.*

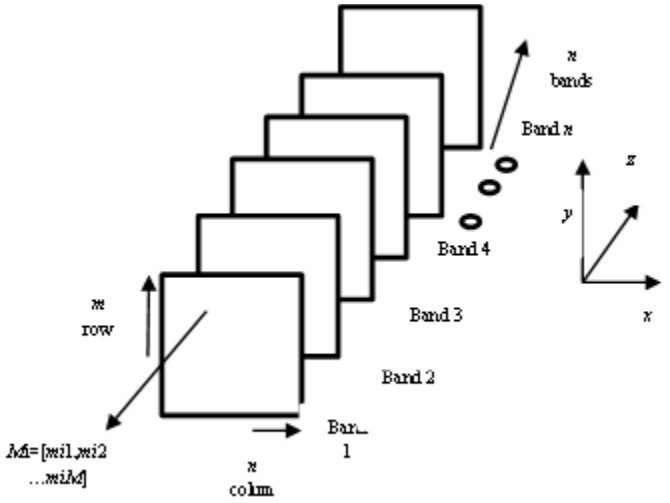


Figure 2. *Diagram of pixel vector in a hyperspectral image.*

2. KLT

2.1 Conventional KLT

The KLT algorithm provides lossy data compression for an optimal block-based transform in a statistical sense (Penna et al.)[16] and is similar to principal component analysis (Gerbrands)[17]. The conventional KLT provides high decorrelation and can be applied over the spectral dimension to remove redundancy of data in hyperspectral images. Let X denote a hyperspectral image with spatial dimension of size M × M and Z bands in the spectral dimension. Image X is arranged as a matrix with Z rows and M2 columns. A mean-centered function is computed by finding the difference between each band and X. Blanes & Serra-Sagristà [18] compute X in the KLT as

$$\mathbf{X}_{\text{encoded}} = \mathbf{KLT}_{\mathbf{\Sigma_X}}(\mathbf{X}) = \mathbf{A}^{\mathrm{T}}\mathbf{X} \quad (1)$$

where Xencoded is the hyperspectral image accompanied by the centered mean, $\mathbf{\Sigma_X} = \left({}^{1}/_{M^2}\,\&\right)\mathbf{X}\mathbf{X}^{\mathrm{T}}$ is the covariance matrix of X, A is the orthogonal matrix acquired from the singular value decomposition of $\mathbf{\Sigma_X} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^{\mathrm{T}}$, where Λ is diagonal matrix diag($\lambda_1, \lambda_2, \ldots, \lambda_z$) with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_z$.

Spectral decorrelation proceeds as follows, with input being hyperspectral image X as a matrix of dimension Z × M2:

The corresponding band mean is subtracted from every entry in X:

$$\mathbf{X}_{\text{adjust}} = [\, x_{z,m} - mean_z \,] \quad (2)$$

where $0 \leq z \leq Z-1; 0 \leq m \leq M^2-1$. The mean is also required for decoding.

Covariance matrix C of Xadjust is determined. Equation (3) shows the covariance between bands A and B, where $x_{\text{adjust}_{a,m}}$ and $x_{\text{adjust}_{b,m}}$ are individual values from bands A and B, respectively, whereas μ_a and μ_b are the mean values of bands A and B from Xadjust, respectively. Covariance matrix C is a Z × Z square matrix (Equation (4)) containing the variance of the bands along the diagonal. Band numbers are denoted with subscripts b in the matrix.

$$cov(A,B) = \frac{\Sigma_{m=0}^{M^2-1}\left(\, x_{\text{adjust}_{a,m}} - \mu_a\right)\left(\, x_{\text{adjust}_{b,m}} - \mu_b\right)}{M^2} \quad (3)$$

where $0 \leq m \leq M^2-1$.

$$\mathbf{C} = \begin{bmatrix} cov_{b0,b0} & cov_{b1,b0} & \cdots & cov_{bz-1,b0} \\ cov_{b0,b1} & cov_{b1,b1} & \ldots & cov_{bz-1,b1} \\ \vdots & \vdots & \ddots & \vdots \\ cov_{b0,bz-1} & cov_{b1,bz-1} & \ldots & cov_{bz-1,bz-1} \end{bmatrix} \quad (4)$$

The singular value decomposition is applied to C to obtain the eigenvectors as matrix A and the eigenvalues as vector v

$$\mathbf{A} = \begin{bmatrix} \alpha_{0,0} & \alpha_{0.1} & \ldots & \alpha_{0,z-1} \\ \alpha_{1,0} & \alpha_{1,1} & \ldots & \alpha_{1,z-1} \\ \cdots & \cdots & \ddots & \vdots \\ \alpha_{z-1,0} & \alpha_{z-1,1} & \cdots & \alpha_{z-1,z-1} \end{bmatrix} \qquad \boldsymbol{v} = \begin{bmatrix} \mathrm{v}_0 \\ \mathrm{v}_1 \\ \vdots \\ \mathrm{v}_{z-1} \end{bmatrix} \quad (5)$$

Every row in A is a principal component. Then, every eigenvector in A is sorted in descending order according to the eigenvalues in v, thus obtaining:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0.1} & \ldots & a_{0,z-1} \\ a_{1,0} & a_{1,1} & \ldots & a_{1,z-1} \\ \cdots & \cdots & \ddots & \vdots \\ a_{z-1,0} & a_{z-1,1} & \cdots & a_{z-1,z-1} \end{bmatrix} \quad (6)$$

with Z vectors of principal components. This matrix is called the KLT transform matrix.

KLT transform matrix A is multiplied with Xadjust to obtain the encoded hyperspectral image:

$$\mathbf{X}_{\text{encoded}} = \sum_{z=0}^{Z-1}\left(a_{i,z}\right)^{\mathrm{T}} x_{\text{adjust}_{z,m}} \quad (7)$$

where $0 \leq i \leq Z-1;\ 0 \leq m \leq M^2-1$.

For lossy compression, number P < Z of principal components in KLT transform matrix A corresponding to the largest eigenvalues in v results in a submatrix with dimension P × Z of A, being equivalent to principal component analysis. On the other hand, when all the principal components are preserved (i.e., A has dimension Z × Z), the transform coding corresponds to the KLT

[19]. Square matrix A is nonsingular (i.e., invertible) and is required along with the mean for decoding, because the latter is considered as overhead information. Equations (8) and (9) describe the KLT decoding for spectral components and the mean restoration, respectively.

$$\mathbf{X} = \sum_{z=0}^{Z-1} a_{i,z}\, x_{\text{encoded}_{z,m}} \quad (8)$$

where $0 \le i \le Z-1;\ 0 \le m \le M^2-1$.

$$\mathbf{X}_{\text{decoded}} = [X_{z,m} + mean_z] \quad (9)$$

where $0 \le z \le Z-1;\ 0 \le m \le M^2-1$.

2.2 Integer KLT

An approximation of the KLT has been proposed by Hao & Shi[20] using matrix factorization of matrix $\mathbf{A}^{\mathrm{T}}$, which was derived in (Hao & Shi)[21], representing encoded image Xencoded as an integer matrix, conforming the integer KLT, which differs from the conventional KLT in the following steps:

(1) The band mean is rounded for step 1.

The KLT transform matrix or eigenvector matrix, $\mathbf{A}^{\mathrm{T}}$, is factored into matrices, P, L, U, and S in step 3.

The PLUS matrices are applied to Xadjust using reversible lifting operations.

2.3 Eigenvector matrix factorization

The matrix factorization method presented by Hao & Shi [21,22] is also considered as an integer-to-integer transform, because it uses integer mapping to generate an integer output. The integer KLT is implemented based on matrix factorization applied to eigenvector matrix AT, which is obtained as the product of elementary reversible matrices (ERMs), whose diagonal elements are equal to ±1. An ERM that has unit diagonal elements is called unit triangular ERM. A triangular ERM [22] is similar to the LU (lower–upper) factorization (Turing)[23]. The factorization is applied to eigenvector matrix AT, which is nonsingular and composed of four Z × Z matrices, namely, reversible permutation matrix P, lower triangular ERMs L and S, and upper triangular ERM U, thus conforming a PLUS factorization.

2.4 Computational complexity reduction

In the KLT with clustering and tiling illustrated in Figure 3, each 1D spectral vector is converted into a 2D matrix, whose partial covariance matrix can be directly determined. Then, the covariance matrices are accumulated for eigen-decomposition and data projection. This way, each row of the matrix contains a group of W bands, where the Z bands are clustered into H groups or segments. When H = 1, the KLT with clustering and tiling degrades to the conventional KLT, whereas when H = Z, no grouping of bands occurs. The performance of KLT with clustering and tiling depends on the selection of H, as it affects the additional information that can be extracted with respect to the conventional KLT [24].

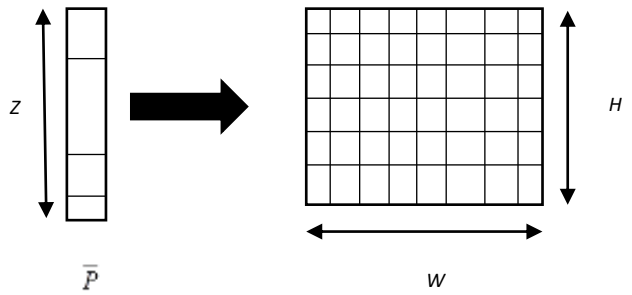


Figure 3. *Construction of 2D matrix by clustering and tiling 1D spectral vector.*

Considering the high redundancy between spectral bands, band clustering is used for band selection and feature characterization [25][26]. From this 2D matrix-based representation, the corresponding covariance matrix helps to extract not only the global structure but also local structures, and hence this approach is superior in performance compared to the conventional KLT (Pang et al., Yu, Bi, & Ye,)[27][28]. In fact, a better discrimination capability with less computational cost and memory requirements is achieved depending on the value of H, as it determines the spectral vector clustering and tiling in a process also known as folded principal component analysis (Zabalza et al.,)[24].

For a spectral vector $\boldsymbol{P}_n = [P_{n1}\ \ P_{n2}\ \ \ldots\ \ P_{nZ}]^T$, let $\mathbf{A}_n$ be the transformed matrix of size H × W (HW = Z). Let

$$\mathbf{A}_n = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \vdots \\ a_{nH} \end{bmatrix}_{H\,X\,W}$$

$$a_{nh} = \left[\boldsymbol{P}_n\left(1+W(h-1)\right)\ \boldsymbol{P}_n\left(2+W\,(h-1)\right)\right)\ \ldots\ \boldsymbol{P}_n\left(W+W\,(h-1)\right)\Big] \quad (10)$$

where $h \in [1, H]$. The covariance matrix of $\mathbf{A}_n$ is calculated as

$$\mathbf{C}_n = \mathbf{A}_n^{\mathrm{T}}\,\mathbf{A}_n\,, \mathbf{C}_n \in \mathbb{R}^{W\times W} \quad (11)$$

Then, the overall covariance matrix for the resulting hypercube is determined as the accumulation of the partial covariance matrices:

$$\mathbf{C}_{\mathrm{KLTCT}} = \frac{1}{S}\sum_{n=1}^{S}\mathbf{C}_n = \frac{1}{S}\sum_{n=1}^{S}\mathbf{A}_n^{\mathrm{T}}\,\mathbf{A}_n \quad (12)$$

where S represents the spatial size of the hypercube. Then, the conventional KLT can be applied for eigen-decomposition with the reduced size, W × W, of C_{KLTCT}, whereas the covariance matrix from the conventional KLT has size Z × Z, thus representing a notably reduced computational cost. Regarding data projection, the equations in the conventional KLT should be adjusted to manage with the input in the clustered vector, i.e., matrix $A_n \in \mathbb{R}^{H\times W}$ rather than spectral vector Pn, For covariance matrix $\mathbf{C}_{\mathrm{KLTCT}}$, let q' be the number of selected eigenvalues and $\Lambda \in \mathbb{R}^{W\times q'}$ the matrix formed using the eigenvectors for data projection. For each row vector in $\mathbf{A}_n$, q' components are extracted during multiplication with Λ, which results in $q = Hq'$ features for the whole image. The data acquired from projection, $\mathbf{Z}_n \in \mathbb{R}^{H\times q'}$, is determined by multiplying two smaller matrices:

$$\mathbf{Z}_n = \mathbf{A}_n\,\boldsymbol{\Lambda},\ \mathbf{A}_n \in \mathbb{R}^{H\times W}, \boldsymbol{\Lambda} \in \mathbb{R}^{W\times q'}$$

which also considerably reduces the computational cost (Zabalza et al., 2014).

3. Integer DWT

The lifting scheme was proposed by Daubechies & Sweldens [29] to simplify the implementation of the discrete wavelet transform by considering stages of split, prediction, and update: Split (lazy wavelet transform): Input discrete signal x[n] is divided into odd and even samples: s0[n] = x[2n] and d0[n] = x[2n + 1]. Each group contains one half of the samples from the input signal. This process is known as lazy wavelet transform. Prediction: The even and odd signal samples are interleaved. If the signal presents local correlations, some of the nearest neighbors can be used to predict samples. In the integer DWT, even samples are used to predict the odd ones. Then, the difference between the prediction and actual odd sample is calculated. Prediction allows

to considerably compress data. Update: At this stage, the global properties of the original signal are preserved, as signals at the highest-level sub-band have a similar average as the original signal. Each even sample is substituted by the average.

Prediction and update can be expressed as

$$s_j[n] = s_{j-1}[n] - \sum_k u_j[k] d_j[n-k] \quad (13)$$

$$d_j[n] = d_1[n] \sum_k p_j[k] s_1[nk] \quad (14)$$

where $s_j[n]$ and $d_j[n]$ are the outputs of the j-th pair of prediction and update, $p_j[k]$ and $u_j[k]$ are parameters of prediction and update, respectively, which are computed by factorizing the resultant matrix obtained through the integer KLT. Figure 4 illustrates the lifting scheme with m pairs of predictions and updates.

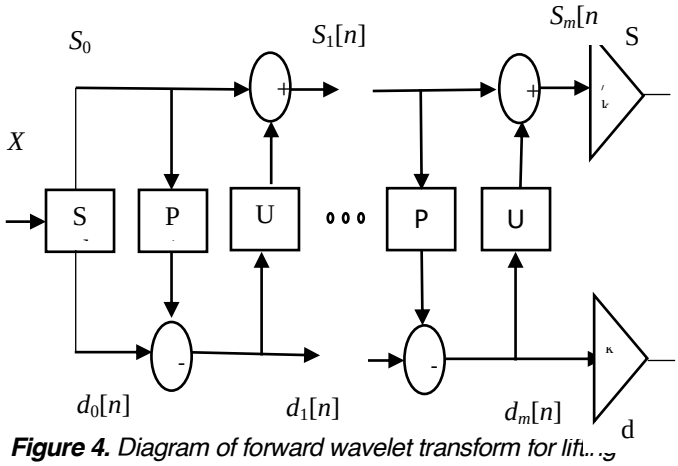


Figure 4. *Diagram of forward wavelet transform for lifting scheme. (Pi, i-th prediction step; Ui, i-th update step)*

For hyperspectral images, we implement the 2D integer DWT optimized by the lifting scheme instead of the 3D DWT in the spatial dimension. The lifting scheme reduces the computational complexity and retrieves the output as a reversible integer wavelet transform (Jensen & la Cour-Harbo)[30]. We decomposed the 3D hyperspectral image per band using the 2D dyadic wavelet. Figure 5 illustrates the second-level decomposition on each band. The integer DWT has two important properties which are essential for image compression, namely, energy packing and self-similarity.

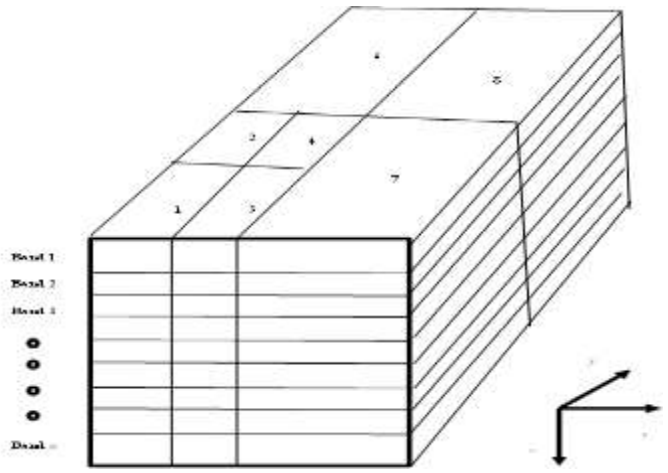


Figure 5. *Three-dimensional integer wavelet transform stored in a data cube.*

We evaluated the transform-based coding illustrated in Figure 1 using two techniques, namely, EZW and SPIHT. Therefore, we obtained the hybrid transform with EZW and the hybrid transform with SPIHT, as detailed below.

4. EZW

Shapiro introduced the EZW algorithm that exploits the advantages of the wavelet transform[12]. The algorithm represents a progressive and embedded image coding method based on the zero-tree data structure, and its most remarkable characteristic is the existence of wavelet coefficients in different sub-bands that represent the same spatial location in an image. By energy packing, the wavelet coefficients in the higher-level sub-bands are processed before those in the lower-level sub-bands using the Morton scan order. The quadtree decomposition of the EZW algorithm has two passes, namely, dominant and subordinate passes. The dominant pass keeps track of the search for the significant coefficients, which have four possible labels: significant positive, significant negative, zero-tree root, and isolated zero. The subordinate pass quantizes each significant coefficient that is found during the dominant pass. If the absolute value of a root coefficient is larger than a threshold, it is labeled as either significant positive or significant negative. Hence, some of the descendants of the coefficients are significant. An isolated zero is a root coefficient that is insignificant but has some significant descendants. If the coefficient is a zero-tree root, the root coefficient itself and its descendants are all insignificant, and thus these descendants need not be encoded at the current iteration (Cheng & Dill)[15]. The transform-based coding illustrated in Figure 1 can be achieved by using either EZW alone or the proposed hybrid transform with EZW.

5. SPIHT

The SPIHT algorithm proposed by Said & Pearlman [31] generalizes the EZW algorithm and uses tree partitioning, which tends to maintain insignificant coefficients together in large subsets. The trees are further partitioned into four types of sets, which are termed the set of coordinates of the coefficients:

O(i,j): set of coordinates of the offspring of wavelet coefficients at location (i,j).

D(i,j): set of all descendants of the coefficients at location (i,j).

G: set of root nodes.

L(i,j): set of coordinates of all the descendants of the coefficient at location (i,j), except for immediate offspring at this location.

SPIHT can be used as an alternate transform-based coding mechanism, and as with EZW, we also established a hybrid transform with SPIHT for processing the hyperspectral images.

6. Experimental Results

We considered test hyperspectral images acquired using AVIRIS for evaluation of the proposed compression method over the regions of Indian Pines, Salinas, Botswana, Urban, and KSC (Hyperspectral remote sensing scenes; Sun: Hyperspectral datasets)[32][33]. We cropped the images to 128 × 128 × 128 from the top-left corner. The evaluation was implemented on MATLAB (Mathworks, Inc., Natick, MA, USA) and executed on a computer with Intel(R) Core(TM) i3-3217U @ 1.80 GHz

processor and 6 GB RAM. We applied different wavelet filters to the images, including Haar wavelets, Daubechies wavelets (db1 and db4), Symlets (sym6), and biorthogonal wavelets (Bior2.4 and Bior4.4). We compared the results using these filters and the EZW and SPIHT algorithms, the proposed hybrid transform with EZW, and the proposed hybrid transform with SPIHT. Tables 1 to 4 list the compression results using the different wavelet filters in terms of bits per pixel per band (bpppb) and peak signal-to-noise ratio (PSNR, expressed in decibels) when applying the EZW algorithm, the SPIHT algorithm, the hybrid transform with EZW algorithm, and the hybrid transform with SPIHT algorithm, respectively. Table 5 lists the computation time in seconds for the compared techniques when applied on hyperspectral images.

TABLE 1. *COMPRESSION RESULTS FROM EZW ALGORITHM USING VARIOUS WAVELET FILTERS.*

	Compression Performance									
Hyperspectral images	Indian Pines		Salinas		Botswana		Urban		KSC	
Wavelet filter	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR(dB)	bpppb	PSNR (dB)	bpppb
Haar	57.71	12.80	49.25	10.08	52.57	11.85	51.14	7.77	45.68	6.71
db1	57.71	12.80	49.25	10.08	52.57	11.85	51.14	7.77	45.68	6.71
db4	58.39	13.05	53.34	10.64	55.66	11.93	50.07	7.69	49.80	7.06
Sym6	58.31	12.99	53.15	10.64	54.72	11.90	49.39	7.72	50.18	7.30
Bior2.4	59.24	12.79	54.33	10.06	56.30	11.54	51.86	7.31	42.26	6.69
Bior4.4	58.33	12.98	53.82	10.49	53.85	11.72	50.70	7.51	48.33	6.99

TABLE 2. *COMPRESSION RESULTS FROM SPIHT ALGORITHM USING VARIOUS WAVELET FILTERS.*

Compression Performance										
Hyperspectral images	Indian Pines		Salinas		Botswana		Urban		KSC	
Wavelet filter	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb
Haar	57.71	9.13	49.25	7.34	52.57	8.50	51.14	5.66	45.68	4.60
db1	57.71	9.13	49.25	7.34	52.57	8.50	51.14	5.66	45.68	4.60
db4	58.39	9.23	53.34	7.48	55.66	8.40	50.07	5.51	49.80	4.85
Sym6	58.31	9.17	53.15	7.50	54.72	8.39	49.39	5.51	50.18	5.15
Bior2.4	59.24	9.05	54.33	7.24	56.30	8.16	51.86	5.28	42.26	4.52
Bior4.4	58.33	9.18	53.82	7.36	53.85	8.29	50.70	5.42	48.33	4.83

TABLE 3. *COMPRESSION RESULTS FROM HYBRID TRANSFORM WITH EZW ALGORITHM FOR VARIOUS WAVELET FILTERS.*

Compression Performance										
Hyperspectral images	Indian Pines		Salinas		Botswana		Urban		KSC	
Wavelet filter	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb
Haar	60.29	12.57	57.77	8.43	56.81	10.79	50.52	6.63	42.00	4.99
db1	60.29	12.57	57.77	8.43	56.81	10.79	50.52	6.63	42.00	4.99
db4	60.94	12.81	57.48	9.04	58.38	11.06	51.22	7.76	45.6	7.03
Sym6	54.34	13.09	47.86	9.95	49.70	11.53	39.98	9.30	28.8	9.33
Bior2.4	60.60	12.84	56.44	8.90	57.67	11.11	51.02	7.51	48.86	6.22
Bior4.4	60.01	12.76	56.08	9.12	57.17	11.17	48.44	8.05	46.66	7.27

TABLE 4. *COMPRESSION RESULTS FROM HYBRID TRANSFORM WITH SPIHT ALGORITHM FOR VARIOUS WAVELET FILTERS.*

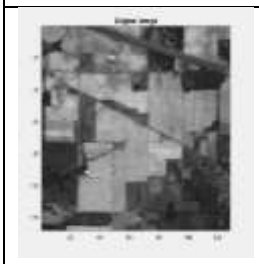
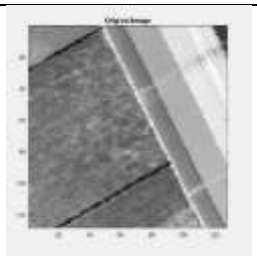
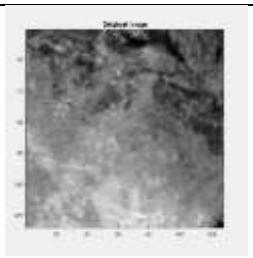
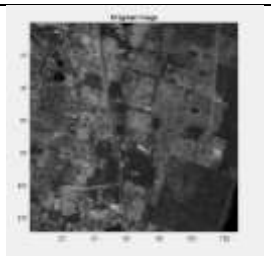
Compression Performance										
Hyperspectral Images	Indian Pines		Salinas		Botswana		Urban		KSC	
Wavelet filter	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR (dB)	bpppb	PSNR(dB)	bpppb	PSNR (dB)	bpppb
Haar	60.29	9.01	57.77	6.24	56.81	7.77	50.52	5.03	42.00	3.79
db1	60.29	9.01	57.77	6.24	56.81	7.77	50.52	5.03	42.00	3.79
db4	60.94	9.08	57.48	6.51	58.38	7.87	51.22	5.58	45.60	4.93
Sym6	54.34	9.31	47.86	7.10	49.70	8.18	39.98	6.60	28.80	6.63
Bior2.4	60.60	9.01	56.44	6.34	57.67	7.84	51.02	5.37	48.86	4.09
Bior4.4	60.01	9.07	56.08	6.54	57.17	7.92	48.44	5.82	46.66	4.97

TABLE 5. *COMPUTATION TIME IN SECONDS OF EVALUATED TECHNIQUES TO COMPRESS HYPERSPECTRAL IMAGES.*

	Indian Pines	Salinas	Botswana	Urban	KSC
EZW	661	440	562	285	254
SPIHT	183	195	185	98	142
Hybrid transform w/EZW	465	242	365	169	187
Hybrid transform w/SPIHT	143	139	118	90	102

Figure 6. *Original hyperspectral images from different regions.*

The PSNR for the all the tested images using the hybrid transforms are high, and the bits per pixel per band are better than the other comparison approaches. In Tables 1 and 2, the Bior2.4 wavelet filter retrieves the highest PSNR and lowest number of bits per pixel per band among the evaluated filters. Likewise, in Tables 3 and 4, both the Bior2.4 and Haar wavelet filters retrieve the best results. The proposed hybrid transform with both the EZW and SPIHT algorithms retrieves better results than the corresponding conventional algorithms for all the test hyperspectral images. In addition, the hyperspectral images rom Salinas, Urban, and KSC are more influenced by the evaluated algorithms than those from Indian Pines and Botswana. From Table 5, the computation time using the hybrid transform with EZW and SPIHT algorithms is lower than that of the corresponding conventional algorithms for all test images. These results suggest that the hybrid transform with SPIHT algorithm provides the best performance than the other approaches. Figure 6 shows original hyperspectral images from the evaluated regions, and Figures 7 to 10 show reconstructed images using the four evaluated approaches applied to these hyperspectral images using the Bior2.4 wavelet filter at the 128th band. All the reconstructed images are very similar to the original images. In fact, the reconstructed images retrieved a very low mean squared error.

Indian Pines	Salinas	Botswana	Urban	KSC
				

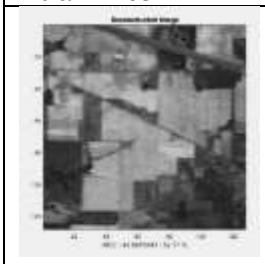
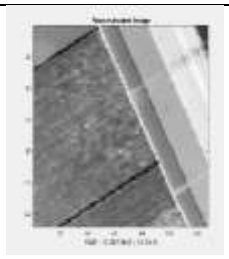
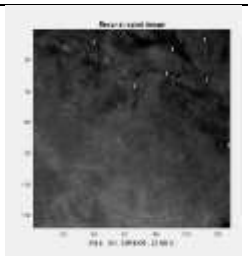
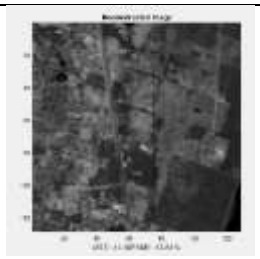
Indian Pines	Salinas	Botswana	Urban	KSC
				

Figure 7. *Reconstructed images using EZW compression with Bior2.4 wavelet filter at the 128th band.*

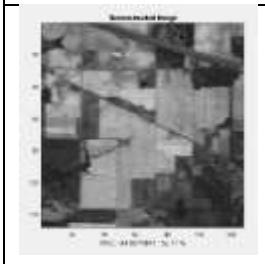
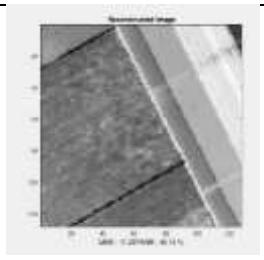
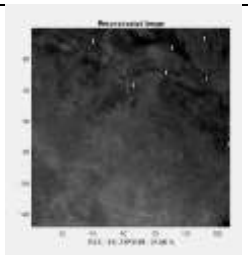
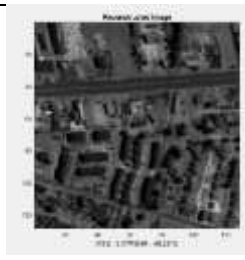
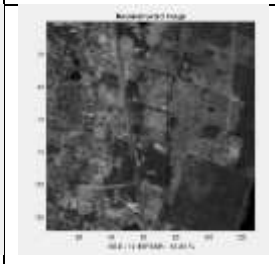
Indian Pines	Salinas	Botswana	Urban	KSC
				

Figure 8. *Reconstructed images using SPIHT compression with Bior2.4 wavelet filter at the 128th band.*

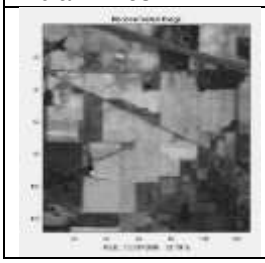
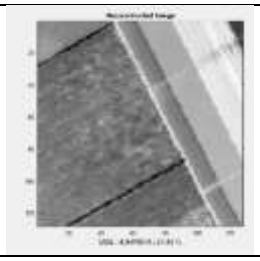
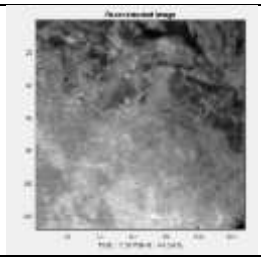
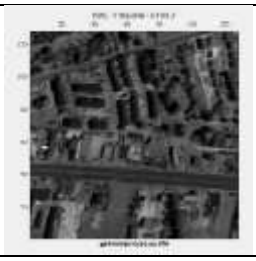
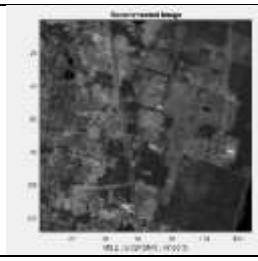
Indian Pines	Salinas	Botswana	Urban	KSC
				

Figure 9. *Reconstructed images using hybrid-transform compression with EZW and Bior2.4 wavelet filter at the 128th band.*

Indian Pines	Salinas	Botswana	Urban	KSC

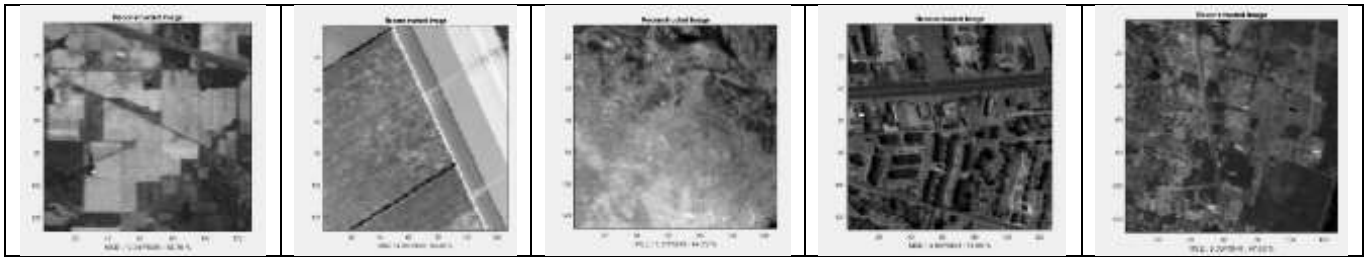


Figure 10. *Reconstructed images using hybrid-transform compression with SPIHT and Bior2.4 wavelet filter at the 128th band.*

7. CONCLUSION

We propose hybrid transforms for compressing hyperspectral images using either the EZW or SPIHT algorithm. We evaluated the compression performance on hyperspectral images, and the hybrid transform with SPIHT algorithm retrieved better performance results in terms of PSNR, bits per pixel per band, and computation time than the transform with the EZW algorithm and the conventional SPIHT and EZW algorithms. Moreover, the underlying KLT with clustering and tiling reduces the computational complexity of hyperspectral images. The obtained compression was between 3.79 and 12.99 bits per pixel per band for the hyperspectral images.

REFERENCES

[1] AVIRIS—Airborne Visible/Infrared Imaging Spectrometer (https://aviris.jpl.nasa.gov/).

[2] Roger, R.E., & Cavenor, M.C, "Lossless compression of AVIRIS images," IEEE Transactions on Image Processing, 5(5), 713–719, 1996, doi: 10.1109/83.495955

[3] Aiazzi, B, Alba, P., Alparone, L., & Baronti, S, "Lossless compression of multi/hyper-spectral imagery based on a 3-D fuzzy prediction," IEEE Transactions on Geoscience and Remote Sensing, 37(5), 2287–2294. 1999, doi: 10.1109/36.789625

[4] Wu, X., & Memon, N, "Context-based, adaptive, lossless image coding," IEEE Transactions on Communications, 45(4), 437–444. 1997, doi: 10.1109/26.585919

[5] Wu, X., & Memon, N, "Context-based lossless interband compression-extending CALIC," IEEE Transactions on Image Processing, 9(6), 994–1001, 2000,. doi: 10.1109/83.846242

[6] Sayood, K, "Introduction to data compression," Cambridge, MA: Morgan Kaufmann, p. 680, 2006.

[7] Ryan, M.J., & Arnold, J.F, "The lossless compression of AVIRIS images by vector quantization," IEEE Transactions on Geoscience and Remote Sensing, 35(3), 546–550, 1997 doi: 10.1109/36.581964

[8] da Silva, E.A., Sampson, D.G., & Ghanbari, M, "A successive approximation vector quantizer for wavelet transform image coding," IEEE Transactions on Image Processing, 5(2), 299–310, 1996, doi: 10.1109/83.480765

[9] Knipe, J., Li, X., & Han, B, "An improved lattice vector quantization scheme for wavelet compression," IEEE Transactions on Signal Processing, 46(1), 239–243, 1998, doi: 10.1109/78.651227

[10] Chao, C.C., & Gray, R.M, "Image compression with a vector speck algorithm," Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, France. doi: 10.1109/ICASSP.2006.1660375

[11] Penna, B., Tillo, T., Magli, E., & Olmo, G, "Embedded lossy to lossless compression of hyperspectral images using JPEG 2000," Proceedings IEEE International Geoscience and Remote Sensing Symposium, Seoul, Korea, 2005,doi: 10.1109/IGARSS.2005.1526124

[12] Shapiro, J.M, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Transactions on Signal Processing, 41(12), 3445–3462, 1993,. doi: 10.1109/78.258085

[13] Bilgin, A., Zweig, G., & Marcellin, M. W, "Three-dimensional image compression with integer wavelet transform," Applied Optics, 39(11), 1799–1814, 2000, doi: 10.1364/AO.39.001799

[14] Kim, B.J., Xiong, Z., & Pearlman, W.A, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," IEEE Transactions on Circuits and Systems for Video Technology, 10(8), 1374–1387, 2000, doi: 10.1109/76.889025

[15] Cheng, K.J., & Dill, J.C, "An improved EZW hyperspectral image compression". Journal of Computer and Communications, 2(2), 31–36, 2014, doi: 10.4236/jcc.2014.22006

[16] Penna, B., Tillo, T., Magli, E., & Olmo, G, "Transform coding techniques for lossy hyperspectral data compression," IEEE Transactions on Geoscience and Remote Sensing, 45(5), 1408–1421, 2007, doi: 10.1109/TGRS.2007.894565

[17] Gerbrands, J.J, "On the relationships between SVD, KLT and PCA,". Pattern Recognition, 14(1–6), 375–381, 1981, doi: 10.1016/0031-3203(81)90082-0

[18] Blanes, I., & Serra-Sagristà, J. (2010). Cost and scalability improvements to the Karhunen–Loêve transform for remote-sensing image coding. IEEE Transactions on Geoscience and Remote Sensing, 48(7), 2854–2863, 2010, doi: 10.1109/TGRS.2010.2042063

[19] Du, Q., & Fowler, J.E, "Hyperspectral image compression using JPEG2000 and principal component analysis," IEEE Geoscience and Remote Sensing Letters, 4(2), 201–205. doi: 10.1109/LGRS.2006.888109

[20] Hao, P. & Shi, Q. (2003). Reversible integer KLT for progressive-to-lossless compression of multiple component images. Proceedings of the International Conference on Image Processing, Barcelona, Spain, pp. I-633–6, 2003, doi: 10.1109/ICIP.2003.1247041

[21] Hao, P., & Shi, Q, "Matrix factorizations for reversible integer mapping," IEEE Transactions on Signal Processing, 49(10), 2314–2324. doi: 10.1109/78.950787

[22] Galli, L., & Salzo, S, "Lossless hyperspectral compression using KLT," IEEE International Conference on Geoscience and Remote Sensing Symposium, Anchorage, AK, pp. 313–316, 2004, doi: 10.1109/IGARSS.2004.1369024

[23] Turing, A.M, "Rounding-off errors in matrix processes," The Quarterly Journal of Mechanics and Applied Mathematics, 1(1),

287–308, 1948, doi: 10.1093/qjmam/1.1.287
[24] Zabalza, J., Ren, J., Yang, M., Zhang, Y., Wang, J., Marshall, S., & Han, J, "Novel folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing," ISPRS Journal of Photogrammetry and Remote Sensing, 93, 112–122, 2014, doi: 10.1016/j.isprsjprs.2014.04.006
[25] Wang, S., & Chang, C.I, "Variable-number variable-band selection for feature characterization in hyperspectral signatures," IEEE Transactions on Geoscience and Remote Sensing, 45(9), 2979–2992, 2007, doi: 10.1109/TGRS.2007.901051
[26] Zhao, Y.Q., Zhang, L., & Kong, S.G, "Band-subset-based clustering and fusion for hyperspectral imagery classification," IEEE Transactions on Geoscience and Remote Sensing, 49(2), 747–756, 2011, doi: 10.1109/TGRS.2010.2059707
[27] Pang, E., Wang, S., Qu, M., Liu, R., Jia, C., & Yu, Z, "2D-SPP: a two-dimensional extension of sparsity preserving projections," Journal of Information and Computational Science, 9(13), 3683–3692, 2012..
[28] Yu, S., Bi, J., & Ye, J, "Probabilistic interpretations and extensions for a family of 2D PCA-style algorithms," Proceedings of the KDD Workshop on Data Mining Using Matrices and Tensors, Las Vegas, NV, 2008.
[29] Daubechies, I., & Sweldens, W, "Factoring wavelet transforms into lifting steps," Journal of Fourier Analysis and Applications, 4(3), 247–269, 1998. doi: 10.1007/BF02476026
[30] A. Jensen and A. La Cour-Harbo, "Ripples in Mathematics : The Discrete Wavelet Transform," New York: Springer, 2001.
[31] Said, A., & 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, 6(3), 243–250, 1996, doi: 10.1109/76.499834
[32] Hyperspectral remote sensing scenes (http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes)
[33] Sun, L. Hyperspectral datasets. Retrieved from http://lesun.weebly.com/hyperspectral-data-set.html.