

Original Article

Block-Based Fractional Wavelet Filter for Compression of Hyperspectral Images over Wireless Multimedia Sensor Network Platforms

Rajesh¹, Shrish Bajpai², Naimur Rahman Kidwai³

^{1,2,3}Electronics & Communication Engineering Department, Faculty of Engineering & Information Technology, Integral University, Lucknow, Uttar Pradesh, India.

² Corresponding Author : shrishbajpai@gmail.com

Received: 03 January 2025

Revised: 04 February 2025

Accepted: 03 March 2025

Published: 29 March 2025

Abstract - With the property of excellent energy compaction and the ability to analyse hyperspectral images in the space-frequency domain, the 3D Dyadic Wavelet Transform is broadly used for the compression of hyperspectral images. However, calculating the transform coefficient through the conventional wavelet transform is memory intensive, which makes the compression algorithm unsuitable for the resource constraint hyperspectral image sensors. In the present study, the transform coefficients are calculated through block-based fractional wavelet transform frame by frame of the hyperspectral image to reduce the transform memory. The transform coefficient is coded with the zero memory set partitioned embedded block. We evaluated the proposed compression algorithm through MATLAB simulations with six popular hyperspectral images. From the result, it has been observed that demand for transform memory and transform complexity is reduced significantly while the requirement of the coding memory is zero and coding complexity is at par with other listless compression algorithms. Thus, the proposed compression algorithm is an optimum choice for the resource constraint hyperspectral image sensors.

Keywords - Hyperspectral Image Compression, Low Memory Architecture, Lossy Compression, Hyperspectral Image Coding, Wavelet Transform, Low Complexity.

1. Introduction

HyperSpectral (HS) Imaging gathers spectral and spatial image data simultaneously in hundreds of narrow contiguous spectral bands from visible to infrared wavelengths [1]. HS image technology integrates both conventional imaging and spectroscopy, and spectral data are organized into hypercubes, having two spatial dimensions (2D spatial scene) and one spectral dimension (1D spectral profile) [2]. Applications of the HS images are ranging from the agriculture monitoring [3], astronomy [4], biomedical engineering [5], chemical imaging [6], corrosion detection (steel infrastructure) [7], cultural heritage [8], food quality analysis [9], forensics [10], land-cover mapping [11], medical [12], military (surveillance and reconnaissance) [13], pharmaceutical [14], remote sensing (earth monitoring), surgery (image-guided) [15], target detection [16], water pollution detection [17], weather prediction [18] etc. Hyperspectral image sensors (AVIRIS, HyMap, HYDICE, Hyperion, etc.) primarily capture electromagnetic radiation in the reflective range, which extends from the visible spectrum to the short-wave infrared region having hundreds of narrow contiguous spectral frequency frames with ten nanometres spacing [19]. Thus, a single pixel is represented as a stack of

array instead of a fixed value in grayscale images, which represents a piece of fine information across the wavelength range from 400 nm to 2500 nm [20, 21]. With such kind of detail information, the HS images are used in multiple applications such as feature extraction [22], feature selection [23], band selection [24], change detection [25], classification [26], compression, denoising [27], dimension reduction [28], segmentation [29], object recognition [30] etc. The hyperspectral image compression problem has recently received much attention in the remote sensing community [31]. Compression of the HS image is an essential step before the transmission of the HS images as it saves the storage memory [32], decreases image data browsing time [33], minimizes transmission bandwidth [34], shortens data transmission time and reduce HS image sensor complexity.

This study aims to develop a compression algorithm that has the demands of transforming memory in line with the coding memory. Many compression algorithms had been developed in the past, but they did not consider the transform memory in the development of the compression algorithm. Thus, it is necessary to reduce the demand for the transform memory. Previously, a fraction wavelet filter was applied



with the transform-based low memory compression algorithm, but it still has high complexity and transforms memory demand [35]. In this manuscript, we present a novel compression algorithm based on the block-based fractional wavelet transform and zero-memory set partitioned embedded block used for encoding and decoding the transform coefficients. The proposed compression algorithm works for both lossy compression and lossless compression.

The rest of this article is organized as follows: Section 2 presents a brief review of the different HSICAs based on data loss and coding process, while Section 3 introduces the background principles of the proposed compression algorithm, which includes a detailed description of block-based fractional wavelet filter, transform-based set partitioned hyperspectral image compression algorithm, zeroblock cube hyperspectral image compression algorithm and 2D-zero memory set partitioned embedded block. Section 4 presents the proposed compression algorithm with the associated pseudo code. We have validated the simulation results of our proposed algorithm in Section 5, demonstrating its effectiveness in the compression of HS images with a detailed analysis description. Finally, Section 6 summarizes the whole work's main findings and conclusions.

2. Literature Review

Many HyperSpectral Image Compression Algorithms (HSICAs) have been proposed for HS image data compression in the last two decades. There are two types of correlation (spatial and spectral) that exist in the HS image [36]. The spectral correlation exists between the multiple nearby frequency frames, while spatial correlation exists between the pixels in the same spectral frame [37]. It has been known that in HS images, spectral correlation is more significant than spatial correlation [38]. By removing the correlation, the redundancy in the HS image is also reduced [39].

The compression algorithms are divided based on two factors: image data loss and the coding process [39, 40]. In view of the data loss, the HSICA can be classified into three different subcategories names as lossy compression, lossless compression, and near-lossless compression. The Compression Ratio (CR) is the ratio between the size of the HS image and the size of the reconstructed HS image, and it is a unitless parameter. For the lossless HSICA, there is no loss of any HS image data during the compression process, but the CR is very low, while for the near-lossless HSICA, CR is a little bit higher, but there is a loss of some HS image data. This loss of HS image data is not so significant to the human eye [41].

Lossy compression had a very high CR, but it came at the cost of losing the HS image data. It has also been known that lossless and near-lossless compression has very high coding efficiency, while lossy compression has low coding

efficiency. In view of the complexity of the HSICA, lossy compression HSICA had low complexity, but lossy and lossless have high coding complexity due to the nature of computations during the compression process. In the view of the coding process, the HSICAs are divided into Predictive Coding (PC) [42], Vector Quantization (VQ) [43], Transform Coding (TC) [44], Compressive Sensing (CS) [45], Tensor Decomposition (TD) [46], Learning-Based Compression (LC) [47] and hybrid compression algorithms [48].

With the aid of the difference between the present pixel value and the previous pixel value, the predictive coding of HS images determines the future value of the pixel. Regarding the HS image, the predictive coding schemes are applied in the spatial domain with the help of different predictors. Entropy coding techniques, like arithmetic or Huffman coding, encode prediction errors [49]. These algorithms are data-dependable, and the HS image is only reconstructed when all HS image data is received by the decoder [42].

The dictionary-based HS image compression, known as vector quantization, divides the HS image into several distinct blocks based on their similarities while simultaneously attempting to find the best possible approximation and then generating a unique symbol or code for each of those blocks [50]. There are two primary aspects to it: the first is the creation of the codebook, and the second is the hunt for the code word. An efficient method that employs X code words to represent Y training vectors can accomplish the generation of codes or symbols for the codebook. This can be done through an optimal scheme [51].

The procedure for designing a codebook is the same as the construction process for X code words. The process of identifying code words that produce the least mean squares error with training vectors is referred to as "code word searching". During the decoding process, the HS image is created with the assistance of the received codebook [43].

The complexity of the coding is moved from the encoder end (onboard HS image sensor) to the decoder end (ground station) via the Compressive Sensing (CS) based HSICAs [31]. These methods are utilised in the real-time scenario, and in order for the compression process to work properly, they require a small amount of memory. After compressing the HS image data, the compression algorithm sends it to the ground station before proceeding to the next slice of the HS image.

This process repeats until both slices of the HS image have been processed. The most significant benefits of CS-based compression algorithms are their high compression performance, low encoder complexity, and required short coding memory. On the other hand, the most significant problems of these algorithms are their expensive decompression and complex decoding [45].

The Tensor Decomposition (TD) based HSICAs can be applied with either the TC, PC, or LC-based HSICAs. A tensor is a matrix with n dimensions that may be deconstructed straightforwardly. The HS image is saved in the tensor, and then a compression technique is run on it to break it down into a tensor with fewer dimensions. This tensor with a low number of dimensions is given a code and sent through the channel. These algorithms offer a shorter run time and a high coding efficiency, but they have drawbacks such as data dependence, a manual parameter update technique, and a complex coding process [46].

The deep learning and machine learning methods are used in the LC-based HSICA. These compression algorithms also use the neural network (convolutional, recurrent, autoencoder, feedforward, multilayer perceptron) [52]. It has high coding efficiency with moderate CR. These HSICA have very high coding complexity and coding memory requirements. It also requires the other hardware resources [47].

In order to achieve the desired level of HS picture compression, the algorithms for hybrid code compression use a combination of any two or three of the methodologies described above [20]. These particular kinds of algorithms have a high coding efficiency, but this comes at the expense of a higher level of complexity [53]. The most common types of combinations that are used to create the HSICA are neural networks with predictive coding and neural networks with transform coding. Both of these types of combinations involve neural networks [48].

Transform Coding (TC) is the approach that is used the most frequently for the lossy and lossless compression of HS images [54]. The mathematical transform is utilised by these types of compression algorithms in order to bring about the necessary transformation of the image into the frequency domain. The Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Dyadic Wavelet Transform (DWT), Discrete Wavelet Packet Transform (DWPT), Integer Wavelet Packet Transform (IWPT), and Karhunen-Loeve Transform (KLT) are the most popular mathematical transform use to compress the HS images [31, 35, 55].

In order to accomplish the compression, these mathematical transformations reduce the spectral and spatial correlations. The TC-based HSICAs provide the highest possible performance even at the lowest possible bit rates. A significant rise in the total number of mathematical computations makes the TC compression technique more difficult to understand, which is the most significant drawback of this particular subcategory of compression algorithms [56-58]. The 3D-Set Partitioning Embedded Block (3D-SPECK) [59], 3D-Set Partitioning In Hierarchical Trees (3D-SPIHT) [60], 3D-Wavelet Block Tree Coding (3D-WBTC) [61], 3D-Listless SPECK (3D-LSK) [62], 3D-

No List SPIHT (3D-NLS) [63], 3D-Low memory block tree coding (3D-LMBTC) [64], 3D-Low complexity block tree coding (3D-LCBTC) [65], 3D-zero memory-set partitioned embedded bloCK (3D-ZM-SPECK) [66], 3D-Listless Embedded Zerotree Set Partitioning Coding (3D-LEZSPC) [67] and Fractional wavelet filter based ZM-SPECK [35] are the popular state of art TC based HSICAs.

In terms of complexity, PC-based HSICA has the lowest complexity (processing and very simple hardware implementation). In contrast, LC-based HSICA has very high complexity due to the implementation (learning process and complex hardware implementation). While on the view of the coding efficiency, LC-based HSICA has high coding efficiency. Some TC-based HSICAs have embeddedness properties, meaning that HS images can be reconstructed at a lower bit rate than the encoding bit rate. TC-based HSICAs work for lossy and lossless compression per the application requirement.

3. Background Principle

This section presents a comprehensive introduction to low-memory wavelet transform, a foundation for developing low-memory HS image sensors. The mathematical transform converts the time domain HS image to the frequency domain HS image [61]. Many mathematical transforms are used to compress the HS images, but the choice of transform depends only on the application demand [64-67].

A special type of TC-based HSICA is called Transform Based Set Partitioned Hyperspectral Image Compression Algorithms, which use the set structure to determine the significance against the current threshold. Among them, zeroblock cube HSICAs divide and transform the HS image into the contiguous block cubes and perform the significance test on the individual block cubes against the current threshold. 2D-Zero Memory Set Partitioned Embedded Block (2D-ZM-SPECK) is a special type of compression technique that does not require any coding memory and has low computational complexity.

3.1. Block-Based Fractional Wavelet Filter (BFrWF)

The 3D dyadic wavelet transforms (3D-DWT) had excellent energy clustering properties. However, it also requires a lot of memory for calculating the transform coefficients, which is impossible for low-resource HS image sensors [65]. For the calculation of the transform coefficients of the HS image through 3D-DWT, first, 1D-DWT is applied to all rows of the frequency frame, resulting in sub-bands L and H.

After that, the second 1D-DWT is applied to all columns of the frequency frame, resulting in four sub-bands: LL, LH, HL and HH. This process is repeated to the rest of the frequency frames to calculate the transform coefficients. Lastly, 1D-DWT is applied to the spectral dimension of the

HS image for each pixel array, resulting in eight sub-bands: LLL, LLH, LHL, LHH, HLL, HLH, HHL and HHH. Through this approach, the whole HS image needs to be saved on the HS image sensor memory. The transform memory requirement grows linearly with the HS image size. It has been known that only a few MB of memory is available with the HS image sensors to calculate the transform coefficients [64]. So, this approach is not suitable for low-resource HS image sensors. It has also been known that HS images are a collection of 2D frequency frames or slices taken for a single scene [35]. However, this is accomplished at the expense of coding gain due to the fact that 2D-DWT does not investigate the spectral correlation that exists between the frames of the HS image [35].

Apart from the above-mentioned approaches, there are three types of approaches for calculating the transform coefficients, which minimize the demand for transform memory. The line-based DWT [68], strip-based DWT [69] and block based DWT [70] are the three approaches. When using line-based DWT, lines from the image are read into the system's buffer until it is possible to apply vertical filtering [68]. When applied to wide blocks, the DWT based on stripes is analogous to the DWT based on lines. Block-based DWT is more applicable to the job that we are doing since it first divides the image into several different blocks and then transforms each of those blocks individually [70]. They are not appropriate for altering images utilising low-cost sensor nodes or portable devices because of the memory and complexity limits those devices have [71].

The Fractional Wavelet Filter (FrWF) is one of the more recent developments that has helped reduce the amount of memory needed for the computation of forward DWT [72]. Despite the fact that FrWF requires very little memory for its implementation, the amount of memory that it needs to store data still varies depending on the size of the HS image frequency frame, which makes it unsuitable for transforming HS images on platforms with limited memory [73].

Block-Based FrWF (BFrWF) is a modified form of FrWF [70]. Compared to FrWF, BFrWF has fewer complexity and memory requirements. When using BFrWF, a frequency frame is initially broken up into blocks, and then the FrWF algorithm is utilised on each individual block in turn. The BFrWF makes use of five buffers: an input buffer I for storing one image line from the vertical filter area selected from the frequency frame block, four temporary buffers for storing and updating the sub-band coefficients created, and an output buffer F for outputting the sub-band coefficients. In order to illustrate the concept that underpins BFrWF, it starts by supposing that a frequency frame of an HS image is divided up into a certain number (call it 'b') of blocks. For the sake of keeping things simple, we will simply look at one level of wavelet decomposition here.

3.2. Transform Based Set Partitioned Hyperspectral Image Compression Algorithm (TSP-HSICA)

Many wavelet-based HS image compression algorithms that have shown to be rather effective have been proposed over the course of the years. They can be broadly divided into three groups: zero tree-based HSICAs, zeroblock cube-based HSICAs and zeroblock cube tree HSICAs [66]. The zero tree-based HSICA does its task by forming a Spatial Orientation Tree (SOT) by grouping the wavelet coefficients corresponding to the same location and orientation into a single entity [65]. An HSICA based on a zero tree is then an SOT that does not have a significant coefficient in relation to the given threshold [].

In order for zeroblock cube HSICAs to function, the modified image must first be segmented into continuous blocks. Then, the significance test must be carried out on each individual block cube [61, 74]. Therefore, a block with no significant coefficients concerning the provided threshold is considered a candidate for a zero block-based HSICA. The zeroblock cube tree partitions a transformed HS image into coefficient block cubes, and then block cube trees are formed with the roots in the topmost sub-band in a zero tree fashion.

3.3. Zeroblock Cube-Based Hyperspectral Image Compression Algorithm

The zeroblock cube-based HSICA is a special type of TSP-HSICA, which partitions the HS image into two sets named the 'S' set and the 'I' set. The dimension of the 'S' set depends on the level of wavelet transform, and the initial 'S' set lies on the left corner of the HS image while the rest of the HS image is regarded as the 'I' set. These sets checked for significance against the current threshold. Suppose the 'S' set is partitioned into the eight 'S' sets, test for the significance.

The 'I' set found significance, then it is partitioned into the seven new 'S' sets, and a new 'I' set will be generated. This process repeats with the bit budget available. Further, these can be divided into two parts: zeroblock cube HSICA and listless zeroblock cube HSICA.

The list zeroblock cube HSICA uses lists to track the significance of the sets or coefficients, while for the same task, the listless zeroblock cube-based HSICA uses different types of markers or state tables. 3D-SPECK [59], 3D-ZM-SPECK [66], 3D-M-ZM-SPECK, 3D-BCP-ZM-SPECK and 3D-LSK [62] belong to zeroblock cube based HSICA group.

3.4. 2D-Zero Memory Set Partitioned Embedded Block (2D-ZM-SPECK)

The 2D-ZM-SPECK is a unique implementation of 2D-SPECK, which does not depend on any type of coding memory for the lists state tables or markers. It is performed with the help of Morton mapping or linear indexing to determine the length of the sets. 2D-ZM-SPECK does not use

any lists such as in 2D-SPECK [75] or state markers as in 2D-LSK [76]. Through the use of Morton mapping or indexing property of wavelet transform, it does not require List of Significant Pixels (LSP), and it avoids LSP by merging the sorting pass and refinement pass for the current threshold.

In this way, it does not require any static coding memory (fix size coding memory), but for some low-level calculations (logical, algebraic or arithmetic), it requires dynamic coding memory. It is a bit plane coding algorithm that encodes the high-priority bit plane first and then the other according to their weight until the bit budget is available.

The ‘L’ level transform HS image frequency frame is partitioned into ‘S’ set and ‘I’ set according to the level of transform. If ‘S’ set (S_{τ}^{σ}) is significant to the current threshold than the ‘S’ set is partitioned into the four new ‘S’ sub sets ($S_{\tau}^{\sigma/4}$; $S_{\tau+\sigma/4}^{\sigma/4}$; $S_{\tau+\sigma/2}^{\sigma/4}$ and $S_{\tau+3\sigma/4}^{\sigma/4}$) and significance testing is done accordingly against the current threshold.

For the ‘I’ set (I_{τ}) testing for the significance, if found significance, then three new ‘S’ sets (S_{τ}^{τ} ; $S_{2\tau}^{\tau}$ and $S_{3\tau}^{\tau}$) and a new ‘I’ set ($I_{4\tau}$) formed.

The new ‘S’ set maintains the same dimensions as its predecessors in terms of its size. The same approach is carried out until a significant coefficient corresponding to the present threshold can be identified. The testing of the ‘S’ set is performed with the significance function $\psi(\cdot)$, which is calculated through Equation (1) while σ is the set length.

$$\Psi_n(S) = \begin{cases} 1 & ; \text{if } 2^n \leq \max|S| \leq 2^{n+1} \\ 0 & ; \text{if } \max|S| < 2^n \\ Null & ; \max|S| \geq 2^{n+1} \end{cases} \quad (1)$$

σ is calculated as Eq 2 for the frequency frame of dimension ‘N’ by ‘N’, which is calculated as in Eq 2

$$\sigma = \left\lfloor \frac{N * N}{4^L} \right\rfloor \quad (2)$$

If the bit budget is depleted between the bit planes, then the coding gain of the 2D-ZM-SPECK algorithm will be reduced.

4. Process of Encoding

The block-based fractional wavelet transform transforms the HS image frame by frame. To calculate the discrete wavelet converted coefficients of an image, this study makes use of a Block-based fractional wavelet filter, also known as a BFrWF.

In this method, the HS image frequency frame is first divided into blocks, and the FrWF algorithm is then utilised on each individual block in turn.

The modified HS image cube is created by taking a slice of eight continuous frequency frames of the transform HS image together, placing the first frame as it is while other frames are the difference between the current and the last frames.

This process repeats for all frequency frames. After that, each frequency frame of the HS image is divided into the number of blocks mentioned during the transformation of the frequency frames of the HS image.

The encoding and decoding time of every block of frequency frame is measured independently and then added together to calculate the coding complexity of the whole HS image. The same process is also followed to calculate the coding efficiency. In this way, the complexity of the coding process of the compression algorithm is reduced significantly.

With a few minor adjustments and an extra phase of significance testing sets and coefficients to find sets with refinement, the decoding process is similar to the encoding technique. Table 1 provides the pseudo-code for the proposed compression algorithm. Figure 1 shows the encoding process of the compression algorithm.

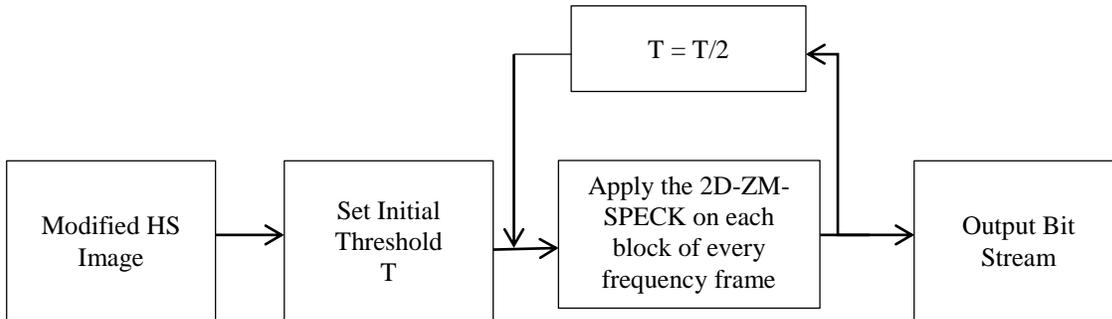


Fig. 1 Encoding process of the proposed HSICA

Table 1. Pseudo code of proposed HSICA

Input : Original HS image 'A' (α & β is defined as spatial dimension while spectral dimension as γ)		
Step 1	Transform of the HS Image HS image is transformed with the BFrWF and it is applied to each frequency frame of HS image. It has been performed one by one for each frequency frame. Transform HS image is defined as 'B'	
Step 2	Creation of the Modified HS Image Cube (MHSIC) for $\phi = 0: \gamma - 1$ { if($\text{rem}(\phi, 8) == 0$) { MHSIC(ϕ) = HS(ϕ) } else MHSIC(ϕ) = [HS(ϕ) - HS($\phi-1$)] } }	
Step 3	Initialization Step Generation of a 1D Linear array 'K' is created from the 3D MHSIC through Morton Mapping. To check the significance of the 'S' set and 'I' set against the current threshold is performed through ' Ψ ' function, which is defined in Equations (1) and (2) Set : Total number of coefficients in the linear array $\lambda = \text{length } K $ Set : Total number of bit plane in the linear array 'K' $n = \log_2[\max(K)]$ Set : Initial threshold of the linear array 'K' $\eta = 2^n$ Set : Initial starting index of the linear array 'K' $\tau = 0$ Set : Initial root set length σ_{root} Set : $\sigma = \sigma_{root}$	
Step 4	Sorting Pass while ($\gamma \leq 256$) { while ($\tau \leq \lambda$) { PSScan(S_τ^σ) if ($(\tau = \sigma) \ \&\& \ (\tau \geq \sigma_{root})$) { PIScan(I_τ) } } }	
Step 5	Quantization Step { $n = (n-1)$ }	
Description of the functions		
Function 1	PSScan() { Output $\Psi_n(S_\tau^\sigma)$ if ($\Psi_n(S_\tau^\sigma) = 0$) { $\tau = (\tau + \sigma)$ } else { if ($\sigma > 4$) { $\sigma = (\sigma/4)$ } } }	Function of 'S' set processing

	<pre> return else { PS(σ, η) } } } NSL(τ, σ) } </pre>	
Function 2	<pre> PIScan() { Output ($\Psi_n(I_\tau)$) { if ($\Psi_n(I_\tau) = 0$) { $\tau = \lambda$ } } } </pre>	Function of 'I' set processing
Function 3	<pre> PS() { for ($\epsilon = 0 : 3$) { Output ($\Psi_n(K(\epsilon + \tau))$) if { $\Psi_n(K(\epsilon + \tau)) = 1$ Output sign bit of $K(\epsilon + \tau)$ } else { $\Psi_n(K(\epsilon + \tau)) = Null$ Output of n^{th} MSB of $K(\epsilon + \tau)$ } } $\tau = (\tau + 4)$ } </pre>	Small 'S' set processing function
Function 4	<pre> NSL () { while bitand ($\tau, 3\sigma$) = 0 ; { $\sigma = 4\sigma$ } } </pre>	Function for the calculation of set length

5. Experimental Results and Discussion

In this section, we evaluate the performance of the proposed HSICA with the other state of art HSICAs on six HS images. The performance of the proposed HSICA has been done based on the transform side and coding side, which

is missing in previously stated TSP-HSICAs. The performance of the associated transform was analysed based on the demand for transform memory and transform complexity (calculation of wavelet transform coefficients). For the coding part of the HSICA, the performance of the

proposed HSICA is analysed based on coding complexity, coding memory, and coding efficiency.

5.1. Dataset

Six different HS images were taken to validate the performance of the different HSICAs with the proposed HSICA. Washington DC Mall (Hyperspectral Image I), Yellowstone Scene 0 (Hyperspectral Image II), Yellowstone Scene 3 (Hyperspectral Image III), Yellowstone Scene 18 (Hyperspectral Image IV), Pavia Centre (Hyperspectral Image V), and University (Hyperspectral Image VI) is taken for the simulation test on Matlab platform. Hyperspectral Image I is captured by the HYDICE HS image sensor (400 nm to 2500 nm with a 10 nm bandwidth having pixel depth 14) while Hyperspectral Image II, III and IV are captured by AVIRIS sensor (380 nm to 2500 nm with a 10 nm bandwidth having pixel depth 16). Hyperspectral Images V and VI are captured by ROSIS HS image sensor (430 nm to 860 nm with a 4 nm bandwidth having pixel depth 13). All HS images are cropped from the top left corner to the size of the cube of dimensions '256' and '512'. The HS images belonging to the Yellowstone dataset have featured forests (river, vegetation, etc), while the three HS images have city architectures (roads, buildings etc).

5.2. Performance Metrics

We evaluate the performance of state of art HSICs and proposed HSICA in Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) index, Compression Ratio (CR), Coding memory and coding complexity [75-78]. For the different types of wavelet transform, the block-based fractional wavelet filter performance was tested based on the transform memory requirement and computational complexity. All the simulation experiments were conducted on the i5 (11th generation) processor having RAM of 20 GB with Windows 11 operating system.

The CR is defined as mathematical, as in Equation (3).

$$CR = \frac{\text{Size of original HS image}}{\text{Size of the reconstructed HS image}} \quad (3)$$

PSNR is the ratio of the maximum possible power of a signal, which is the original image, to the power of the noise, which is based on the disparity between the original and processed images [77]. It is calculated through Equation (4).

$$PSNR = 20 \log_{10} \left[\frac{Max}{MSE} \right] \quad (4)$$

The highest possible pixel value of the HS image is denoted by the letter Max and the Mean Square Error (MSE) of the reconstructed HS image compared to the original HS image. It is calculated as Equation (5).

$$MSE = \frac{1}{N_{pix}} \sum_{x,y,z} [f(x,y,z) - g(x,y,z)]^2 \quad (5)$$

The total number of pixels that exist in the HS image is represented as N_{pix} , while the original HS image and reconstructed HS image after the compression process are represented as 'f(x,y,z)' and 'g(x,y,z)'.

5.3. Benchmark Hyperspectral Image Compression Algorithms

To evaluate the performance of the proposed compression algorithm, the proposed algorithm is compared with state-of-the-art HSICAs 3D-SPECK (CA-I) [59], 3D-SPIHT (CA-II) [60], 3D-WBTC (CA-III) [61], 3D-LSK (CA-IV) [62], 3D-NLS (CA-V) [63], 3D-LMBTC (CA-VI) [64], 3D-LCBTC (CA-VII) [65], 3D-ZM-SPECK (CA-VIII) [66], 3D-LEZSPC (CA-IX) [67] and Fractional wavelet filter based ZM-SPECK (CA-X) [35]. The CA-I, CA-II and CA-III are list-based HSICA while other uses state tables or markers for the tracking of the significance of the partitioned sets.

5.4. Transform Memory

The demand for memory during the transformation and calculation of the wavelet transform is known as transformed memory. BFrWF needs one input buffer to store the part of the frequency frame and four other buffers to calculate the four final sub-bands. Let the size of the HS image frequency frame be 'N' by 'N', and 'b' is the number of blocks of the HS image frequency frames. The dimension of the input buffer is $1 \times \frac{N}{b/2}$ while the rest of the buffer dimension is $1 \times \frac{N}{b}$.

It has been clear from Table 2 that the requirement of the transformed memory is minimal for the BFrWF while for the other transform. The 3D-DWT and 2D-DWT have very high transform memory requirements as they need to save the whole HS image (for 3D-DWT) or the whole frequency frame (for 2D-DWT). The fractional wavelet filter significantly reduces the need for transform memory. Further, it is reduced by using BFrWF to make the transform memory in line with the coding memory, which is near zero. From Table 2, it is also clear that when we increase the number of blocks in the frequency frame, the transform memory also reduces the factor of '2'.

5.5. Transform Complexity

The complexity of any type of wavelet transform is measured by the time required to calculate wavelet coefficients. There are many ways to calculate the wavelet transform coefficients. Table 3 shows that FrWF has the highest time requirement while BFrWF (with four blocks) has almost a similar time requirement as 2D-DWT. As the number of blocks increases, the transform complexity increases, as shown in Table 3. The transform complexity of FrWF is higher than the 2D-DWT because FrWF calculates the wavelet coefficients in different ways and calculates the HS image frequency frame.

5.6. Coding Complexity

The coding complexity of any HSICA is calculated as the time consumed by the compression algorithm for the encoding and decoding process. The amount of time needed for the encoding process is significantly longer than the amount of time needed for the decoding procedure. A rise in the bit rate will result in an increase not just in the amount of time required for encoding but also for decoding [66]. It has been clear from Table 4 (Appendix) it has been clear that the proposed HSICA is less complex than its counterpart, FrWF-

based HSICA, but has high complexity compared to the other listless HSICA except for 3D-LEZSPC [67]. Also, it has a lower complexity than the other list based HSICA. Due to the 2D nature of the coding, its complexity has been increased but is around 2% to 5% less than the FrWF-based HSICA. The HSICA divide frequency frames into 4 blocks, and 2D-ZM-SPECK is applied independently to all blocks. Then, the computation time is calculated and summed up for the calculations.

Table 2. Transform memory requirement of different types of wavelet transform for different HS image size

Type of wavelet Transform \ Dimension of HS Image Cube	128	256	512
3D-Dyadic Wavelet Transform	38.34 MB	306.72 MB	2453.76 MB
2D- Dyadic Wavelet Transform	174.592 KB	698.368 KB	2793.472 KB
Fractional Wavelet Transform	3.123 KB	6.246 KB	12.493 KB
Block-Based Fractional Wavelet Transform (4)	1.5615 KB	3.123 KB	6.246 KB
Block-Based Fractional Wavelet Transform (8)	0.78075 KB	1.5615 KB	3.123 KB
Block-Based Fractional Wavelet Transform (16)	-	0.78075 KB	1.5615 KB

Table 3. Transform Complexity of the different wavelet transforms for different HS image sizes

Type of wavelet Transform \ Dimension of HS Image Cube	256	512
3D-Dyadic Wavelet Transform	2.96 sec	18.47 sec
2D- Dyadic Wavelet Transform	5.65 sec	44.54 sec
Fractional Wavelet Transform	7.59 sec	47.24 sec
Block-Based Fractional Wavelet Transform (4)	6.01 sec	37.06 sec
Block-Based Fractional Wavelet Transform (8)	6.67 sec	39.48 sec
Block-Based Fractional Wavelet Transform (16)	7.48 sec	45.02 sec

5.7. Coding Memory

It has been known that onboard HS image sensors have limited memory. It is clear from Table 5 (Appendix) that the proposed HSICA does not require any coding memory. It is due to the nature that the linear indexing property of the wavelet transform is used to avoid the use of associated lists and state table markers. Other HSICAs, except CA-VIII [66] and CA-X [65], have the requirement for coding memory, and coding memory grows rapidly with the increase in the bit rates for the list-based HSICA. For the listless HSICA, the coding memory is fixed and depends only on the size of the HS image under test. Due to the low (almost zero) coding memory, the proposed HSICA can be a candidate for the resource constraint HS image sensors.

5.8. Coding Efficiency

Table 6 (Appendix) presents the comparative analysis of the proposed HSICA with state-of-the-art HSICA. It has been observed that the proposed HSICA outperform all other HSICA except CA-X [35]. The proposed HSICA has higher

coding efficiency than other HSICA from 2 dB to 4 dB at different bit rates. When using 3D-SPECK [59], the coding gain of the PSNR is enhanced from 6% to 10 % across the board despite the varying bit rates. The large coding gain can be attributed to the fact that the topmost bit planes are unavailable in the majority of the frames of the modified HS image cube. This coding benefit is accomplished by the utilisation of the HS image's inherent spectral redundancy [64, 79]. When the bit budget between the bit planes is exhausted, the suggested compression technique's performance is poor, resulting in a modest drop in the coding gain. It was noted that the range of coding efficiency (PSNR) between 25 dB and 30 dB is adequate for optimal image quality [41]. Table 7 gives the comparative analysis between the 3D-ZM-SPECK [66], FrWF-based HSICA [35] and proposed HSICA for two different HS image sizes at bit rates of 0.1 based on coding efficiency, coding memory and coding complexity. The low-complex architecture of the compression algorithms plays a greater role in processing the image data [66, 79-81].

Table 7. Comparative analysis of the different zero memory HSICA at bit rate 0.1

HS Image	256			512		
	Coding Efficiency					
	CA-VIII [66]	CA-X [35]	Proposed HSICA	CA-VIII [66]	CA-X [35]	Proposed HSICA
Hyperspectral Image II	35.46	37.42	37.02	41.7	44.06	43.68
Hyperspectral Image III	36.37	40.34	39.95	40.13	43.54	43.03
Hyperspectral Image IV	36.2	39.47	39.01	44.29	47.28	47.14
	Coding Memory					
Hyperspectral Image II	0	0	0	0	0	0
Hyperspectral Image III	0	0	0	0	0	0
Hyperspectral Image IV	0	0	0	0	0	0
	Coding Complexity					
Hyperspectral Image II	36.23	64.4	63.35	124.85	201.51	195.36
Hyperspectral Image III	57.96	105.78	100.13	117.47	199.17	191.25
Hyperspectral Image IV	35.4	81.1	77.01	117.76	192.74	188.09

6. Conclusion

The manuscript presents a novel compression algorithm that reduces the HS image sensors' working memory requirement and complexity (transform and coding). It is also observed from the simulation results that the BFrWF has lower transform memory and transform complexity than FrWF. Incorporating parallel processing of blocks is something that will be looked into more in the future because it has the potential to further reduce the complexity of BFrWF.

Acknowledgement

The authors thank Integral University for providing MCN "IU/R&D/2025-MCN0003386".

Author Contributions

Rajesh, and Shrish Bajpai developed the algorithm, simulated the algorithms, and prepared the manuscript, while Naimur Rahman Kidwai edited the manuscript.

References

- [1] B. Krishna Mohan, and Alok Porwal, "Hyperspectral Image Processing and Analysis," *Current Science*, vol. 108, no. 5, pp. 833-841, 2015. [[Google Scholar](#)] [[Publisher Link](#)]
- [2] D. Chutia et al., "Hyperspectral Remote Sensing Classifications: A Perspective Survey," *Transactions in GIS*, vol. 20, no. 4, pp. 463-490, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Chaitanya B. Pande, and Kanak N. Moharir, *Application of Hyperspectral Remote Sensing Role in Precision Farming and Sustainable Agriculture under Climate Change: A Review*, Climate Change Impacts on Natural Resources, Ecosystems and Agricultural Systems, Springer, Cham, pp. 503-520, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Jean-Baptiste Courbot et al., "Pairwise Markov Fields for Segmentation in Astronomical Hyperspectral Images," *Signal Processing*, vol. 163, pp. 41-48, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Yihui Zhou et al., "Machine Learning Empowered Coherent Raman Imaging and Analysis for Biomedical Applications," *Communications Engineering*, vol. 4, pp. 1-14, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Jessica Roberts et al., "A Short Update on the Advantages, Applications and Limitations of Hyperspectral and Chemical Imaging in Food Authentication," *Applied Sciences*, vol. 8, no. 4, pp. 1-7, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Dominik Thomas, and Max Gündel, "Hyperspectral Imaging Systems for Corrosion Detection from Remotely Operated Vehicles," *CE/Papers*, vol. 6, no. 5, pp. 934-938, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Lingxi Liu et al., "Neural Networks for Hyperspectral Imaging of Historical Paintings: A Practical Review," *Sensors*, vol. 23, no. 5, pp. 1-25, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Ndubisi A. Aviara et al., "Potential Application of Hyperspectral Imaging in Food Grain Quality Inspection, Evaluation and Control during Bulk Storage," *Journal of Agriculture and Food Research*, vol. 8, pp. 1-10, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Kristiane de Cássia Mariotti et al., "Hyperspectral Imaging in Forensic Science: An Overview of Major Application Areas," *Science & Justice*, vol. 63, no. 3, pp. 387-395, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] K. Karthik et al., "Study of Land Cover Classification from Hyperspectral Images Using Deep Learning Algorithm," *Computer Networks and Inventive Communication Technologies: Proceedings of Fifth ICCNCT 2022*, pp. 721-737, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Baowei Fei, "Hyperspectral Imaging In Medical Applications," *Data Handling in Science and Technology*, vol. 32, pp. 523-565, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [13] Michal Shimoni, Rob Haelterman, and Christiaan Perneel, "Hypersectral Imaging for Military and Security Applications: Combining Myriad Processing and Sensing Techniques," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 101-117, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Laureen Coic et al., "Essential Spectra to Improve Vibrational Imaging of Pharmaceutical Samples," *Microchemical Journal*, vol. 209, pp. 1-9, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Alankar Kotwal et al., "Hyperspectral Imaging in Neurosurgery: A Review of Systems, Computational Methods, and Clinical Applications," *Journal of Biomedical Optics*, vol. 30, no. 2, pp. 1-54, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Sneha, and Ajay Kaul, "Hyperspectral Imaging and Target Detection Algorithms: A Review," *Multimedia Tools and Applications*, vol. 81, pp. 44141-44206, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Kohzo Homma et al., "Water Pollution Monitoring Using a Hyperspectral Imaging Spectropolarimeter," *Multispectral and Hyperspectral Remote Sensing Instruments and Applications II*, vol. 5655, pp. 419-426, 2005. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Mihaela Antonina Calin, Adrian Cantemir Calin, and Doina Nicoleta Nicolae, "Application of Airborne and Spaceborne Hyperspectral Imaging Techniques for Atmospheric Research: Past, Present, and Future," *Applied Spectroscopy Reviews*, vol. 56, no. 4, pp. 289-323, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] V.R.S. Mani, "Comparison of Three Un-Mixing Based Hyperspectral Image Fusion Techniques," *Journal of the Geological Society of India*, vol. 91, pp. 541-546, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Yaman Dua et al., "Convolution Neural Network Based Lossy Compression of Hyperspectral Images," *Signal Processing: Image Communication*, vol. 95, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Yaman Dua, Ravi Shankar Singh, and Vinod Kumar, "Compression of Multi-Temporal Hyperspectral Images Based on RLS Filter," *The Visual Computer*, vol. 38, pp. 65-75, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Brajesh Kumar et al., "Feature Extraction for Hyperspectral Image Classification: A Review," *International Journal of Remote Sensing*, vol. 41, no. 16, pp. 6248-6287, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Gulsen Taskin, E. Fatih Yetkin, and Gustau Camps-Valls, "A Scalable Unsupervised Feature Selection with Orthogonal Graph Representation for Hyperspectral Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-13, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Mohammad Esmaceli et al., "Hyperspectral Image Band Selection Based on CNN Embedded GA (CNNeGA)," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1927-1950, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Chiman Kwan, "Methods and Challenges Using Multispectral and Hyperspectral Images for Practical Change Detection Applications," *Information*, vol. 10, no. 11, pp. 1-29, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [26] Alope Datta, Susmita Ghosh, and Ashish Ghosh, "Supervised Feature Extraction of Hyperspectral Images Using Partitioned Maximum Margin Criterion," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 1, pp. 82-86, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Xing Wei, Jiahua Xiao, and Yihong Gong, "Blind Hyperspectral Image Denoising with Degradation Information Learning," *Remote Sensing*, vol. 15, no. 2, pp. 1-23, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Mohammed Abdulmajeed Moharram, and Divya Meena Sundaram, "Dimensionality Reduction Strategies for Land Use Land Cover Classification Based on Airborne Hyperspectral Imagery: A Survey," *Environmental Science and Pollution Research*, vol. 30, no. 3, pp. 5580-5602, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Reaya Grewal, Singara Singh Kasana, and Geeta Kasana, "Hyperspectral Image Segmentation: A Comprehensive Survey," *Multimedia Tools and Applications*, vol. 82, pp. 20819-20872, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Silvia Valero, Philippe Salembier, and Jocelyn Chanussot, "Object Recognition in Hyperspectral Images Using Binary Partition Tree Representation," *Pattern Recognition Letters*, vol. 56, pp. 45-51, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [31] Yaman Dua, Vinod Kumar, and Ravi Shankar Singh, "Comprehensive Review of Hyperspectral Image Compression Algorithms," *Optical Engineering*, vol. 59, no. 9, pp. 1-39, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [32] Nor Rizuan Mat Noor, and Tanya Vladimirova, "Investigation Into Lossless Hyperspectral Image Compression for Satellite Remote Sensing," *International Journal of Remote Sensing*, vol. 34, no. 14, pp. 5072-5104, 2013. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [33] Jose Melián et al., "A Novel Data Reutilization Strategy for Real-Time Hyperspectral Image Compression," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [34] Antonis Tsigkanos et al., "High-Performance COTS FPGA SoC for Parallel Hyperspectral Image Compression with CCSDS-123.0-B-1," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2397-2409, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [35] Shrish Bajpai, and Naimur Rahman Kidwai, "Fractional Wavelet Filter Based Low Memory Coding for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 83, pp. 26281-26306, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [36] Rui Dusselaar, and Manoranjan Paul, "Hyperspectral Image Compression Approaches: Opportunities, Challenges, and Future Directions: Discussion," *Journal of the Optical Society of America A*, vol. 34, no. 12, pp. 2170-2180, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [37] Dristi Datta et al., "Comparative Analysis of Machine and Deep Learning Models for Soil Properties Prediction from Hyperspectral Visual Band," *Environments*, vol. 10, no. 5, pp. 1-18, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [38] Dristi Datta et al., "Soil Moisture, Organic Carbon, and Nitrogen Content Prediction with Hyperspectral Data Using Regression Models," *Sensors*, vol. 22, no. 20, pp. 1-20, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [39] Nisha Rani, Venkata Ravibabu Mandla, and Tejpal Singh, "Performance of Image Classification on Hyperspectral Imagery for Lithological Mapping," *Journal of the Geological Society of India*, vol. 88, pp. 440-448, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [40] Zhuang Zhao et al., "Redundant Compressed Single-Pixel Hyperspectral Imaging System," *Optics Communications*, vol. 546, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [41] Mohd Tausif, Ekram Khan, and Antonio Pinheiro, "Computationally Efficient Wavelet-Based Low Memory Image Coder for WMSNs/IoT," *Multidimensional Systems and Signal Processing*, vol. 34, pp. 657-680, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [42] Diego Valsesia, and Enrico Magli, "A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6341-6355, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [43] Shen-en Qian et al., "Fast Three-Dimensional Data Compression of Hyperspectral Imagery Using Vector Quantization with Spectral-Feature-Based Binary Coding," *Optical Engineering*, vol. 35, no. 11, pp. 3242-3249, 1996. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [44] R. Nagendran, and A. Vasuki, "Hyperspectral Image Compression Using Hybrid Transform with Different Wavelet-Based Transform Coding," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 18, no. 1, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [45] K.S. Gunasheela, and H.S. Prasantha, "Compressive Sensing Approach to Satellite Hyperspectral Image Compression," *Information and Communication Technology for Intelligent Systems*, pp. 495-503, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [46] Samiran Das, "Hyperspectral Image, Video Compression Using Sparse Tucker Tensor Decomposition," *IET Image Processing*, vol. 15, no. 4, pp. 964-973, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [47] Chubo Deng, Yi Cen, and Lifu Zhang, "Learning-Based Hyperspectral Imagery Compression through Generative Neural Networks," *Remote Sensing*, vol. 12, no. 21, pp. 1-19, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [48] Yuanyuan Guo et al., "Edge-Guided Hyperspectral Image Compression with Interactive Dual Attention," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-17, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [49] Agnieszka C. Miguel et al., *Predictive Coding of Hyperspectral Images*, Hyperspectral Data Compression, Springer, Boston, MA, pp. 197-231, 2006. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [50] Daniel Báscones, Carlos González, and Daniel Mozos, "Hyperspectral Image Compression Using Vector Quantization, PCA and JPEG2000," *Remote Sensing*, vol. 10, no. 6, pp. 1-13, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [51] Rui Li, Zhibin Pan, and Yang Wang, "The Linear Prediction Vector Quantization for Hyperspectral Image Compression," *Multimedia Tools and Applications*, vol. 78, pp. 11701-11718, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [52] Irem Ulku, and Behçet Uğur Töreyn, "Sparse Representations for Online-Learning-Based Hyperspectral Image Compression," *Applied optics*, vol. 54, no. 29, pp. 8625-8631, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [53] Azam Karami, Soosan Beheshti, and Mehran Yazdi, "Hyperspectral Image Compression Using 3D Discrete Cosine Transform and Support Vector Machine Learning," *11th International Conference on Information Science, Signal Processing and their Applications*, Montreal, QC, Canada, pp. 809-812, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [54] Francesco Rizzo, Giovanni Motta, and James A. Storer, *Hyperspectral Data Compression*, Springer US, pp. 1-417, 2006. [[Google Scholar](#)] [[Publisher Link](#)]
- [55] Amal Altamimi, and Belgacem Ben Youssef, "A Systematic Review of Hardware-Accelerated Compression of Remotely Sensed Hyperspectral Images," *Sensors*, vol. 22, no. 1, pp. 1-53, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [56] Basant K. Mohanty, and Pramod K. Meher, "Memory-Efficient Architecture for 3-D DWT Using Overlapped Grouping of Frames," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5605-5616, 2011. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [57] Jiaji Wu, Zhensen Wu, and Chengke Wu, "Lossy to Lossless Compressions of Hyperspectral Images Using Three-Dimensional Set Partitioning Algorithm," *Optical Engineering*, vol. 45, no. 2, 2006. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [58] Emmanuel Christophe, Corinne Mailhes, and Pierre Duhamel, "Hyperspectral Image Compression: Adapting SPIHT and EZW to Anisotropic 3-D Wavelet Coding," *IEEE Transactions on Image processing*, vol. 17, no. 12, pp. 2334-2346, 2008. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [59] Xiaoli Tang, and W.A. Pearlman, "Lossy-to-Lossless Block-Based Compression of Hyperspectral Volumetric Data," *International Conference on Image Processing*, Singapore, vol. 5, pp. 3283-3286, 2004. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [60] Xiaoli Tang, and William A. Pearlman, *Three-Dimensional Wavelet-based Compression of Hyperspectral Images*, Hyperspectral Data Compression, Springer, Boston, MA, pp. 273-308, 2006. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [61] Shrish Bajpai, Naimur Rahman Kidwai, and Harsh Vikram Singh, "3D Wavelet Block Tree Coding for Hyperspectral Images," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6C, pp. 64-68, 2019. [[Google Scholar](#)] [[Publisher Link](#)]
- [62] Ruzelita Ngadiran et al., "Efficient Implementation of 3D Listless Speck," *International Conference on Computer and Communication Engineering*, Kuala Lumpur, Malaysia, pp. 1-4, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [63] V.K. Sudha, and R. Sudhakar, "3D Listless Embedded Block Coding Algorithm for Compression of Volumetric Medical Images," *Journal of Scientific and Industrial Research*, vol. 72, pp. 735-748, 2013. [[Google Scholar](#)] [[Publisher Link](#)]
- [64] Shrish Bajpai et al., "Low Memory Block Tree Coding for Hyperspectral Images," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27193-27209, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [65] Shrish Bajpai, "Low Complexity Block Tree Coding for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 33205-33323, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [66] Shrish Bajpai et al., "A Low Complexity Hyperspectral Image Compression through 3D Set Partitioned Embedded Zero Block Coding," *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 841-872, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [67] Shrish Bajpai et al., "Curvelet Transform Based Compression Algorithm for Low Resource Hyperspectral Image Sensors," *Journal of Electrical and Computer Engineering*, vol. 2023, pp. 1-18, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [68] Mohd Tausif et al., "Memory-Efficient Architecture for FrWF-Based DWT of High-Resolution Images for IoMT Applications," *Multimedia Tools and Applications*, vol. 80, pp. 11177-11199, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [69] Mohd Tausif et al., "SFrWF: Segmented Fractional Wavelet Filter Based DWT for Low Memory Image Coders," *4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics*, Mathura, India, pp. 593-597, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [70] Mohd Tausif, Ekram Khan, and Mohd Hasan, "BFrWF: Block-Based FrWF for Coding of High-Resolution Images with Memory-Complexity Constrained-Devices," *5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering*, Gorakhpur, India, pp. 1-5, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [71] Mohd Tausif et al., "SMFrWF: Segmented Modified Fractional Wavelet Filter: Fast Low-Memory Discrete Wavelet Transform (DWT)," *IEEE Access*, vol. 7, pp. 84448-84467, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [72] Stephan Rein, and Martin Reisslein, "Performance Evaluation of the Fractional Wavelet Filter: A Low-Memory Image Wavelet Transform for Multimedia Sensor Networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 482-496, 2011. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [73] Mohd Tausif et al., "Lifting-Based Fractional Wavelet Filter: Energy-Efficient DWT Architecture for Low-Cost Wearable Sensors," *Advances in Multimedia*, vol. 2020, no. 1, pp. 1-13, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [74] Divya Sharma, Y.K. Prajapati, and R. Tripathi, "Success Journey of Coherent PM-QPSK Technique with its Variants: A Survey," *IETE Technical Review*, vol. 37, no. 1, pp. 36-55, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [75] W.A. Pearlman et al., "Efficient, Low-Complexity Image Coding with a Set-Partitioning Embedded Block Coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219-1235, 2004. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [76] Mrityunjaya V. Latte, Narasimha H. Ayachit, and D.K. Deshpande, "Reduced Memory Listless Speck Image Compression," *Digital Signal Processing*, vol. 16, no. 6, pp. 817-824, 2006. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [77] Divya Sharma, "Image Quality Assessment Metrics for Hyperspectral Image Compression Algorithms," *Second International Conference Computational and Characterization Techniques in Engineering & Sciences*, Lucknow, India, pp. 1-5, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [78] Nadia Zikiou, Mourad Lahdir, and David Helbert, "Support Vector Regression-Based 3D-Wavelet Texture Learning for Hyperspectral Image Compression," *The Visual Computer*, vol. 36, no. 7, pp. 1473-1490, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [79] Divya Sharma, Y.K. Prajapati, and Rajeev Tripathi, "0.55 Tb/s Heterogeneous Nyquist-WDM Superchannel Using Different Polarization Multiplexed Subcarriers," *Photonic Network Communications*, vol. 39, pp. 120-128, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [80] Samiran Das, and Sandip Ghosal, "Unmixing Aware Compression of Hyperspectral Image by Rank Aware Orthogonal Parallel Factorization Decomposition," *Journal of Applied Remote Sensing*, vol. 17, no. 4, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [81] Pallavi Ranjan et al., "Revolutionizing Hyperspectral Image Classification for Limited Labeled Data: Unifying Autoencoder-Enhanced GANs with Convolutional Neural Networks and Zero-Shot Learning," *Earth Science Informatics*, vol. 18, no. 2, pp. 1-26, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

Appendix

Table 4. Coding complexity analysis (encoding and decoding time) of the different set partitioned HS Image compression algorithms for different bit rates

Bit Rate	Computation Time	CA-I [59]	CA-II [60]	CA-III [61]	CA-IV [62]	CA-V [63]	CA-VI [64]	CA-VII [65]	CA-VIII [66]	CA-IX [67]	CA-X [35]	Proposed CA
Hyperspectral Image I												
0.001	Enc Time	3.99	4.06	5.94	2.67	14.18	5.91	3.17	3.24	20.09	3.29	3.14
	Dec Time	1.78	2.92	1.59	2.08	12.79	2.48	2.21	3.02	3.57	3.11	3.97
	Total Time	5.77	6.98	7.53	4.75	26.97	8.39	5.38	6.26	23.66	6.4	6.11
0.005	Enc Time	9.85	9.73	8.2	2.78	61.33	8.35	3.35	4.83	82.29	4.91	4.7
	Dec Time	5.18	5.25	2.41	2.43	48.29	3.86	2.68	4.65	14.87	4.68	4.38
	Total Time	15.13	14.98	10.61	5.21	109.62	12.21	6.03	9.48	97.16	9.59	9.08
0.01	Enc Time	20.45	29.93	10.99	3.25	73.64	9.26	4.41	5.97	93.55	7.89	7.55
	Dec Time	10.78	14.31	4.51	2.68	57.16	4.04	3.08	5.61	21.59	7.29	7.05
	Total Time	31.23	44.24	15.5	5.93	130.8	13.3	7.49	11.58	115.14	15.18	14.6
0.05	Enc Time	222.2	303.4	94.36	5	90.57	19.45	5.49	12.18	102.89	20.27	19.76
	Dec Time	172.7	236.2	84.75	4.02	69.23	12.01	4.34	11.79	51.47	18.57	18.03
	Total Time	394.9	539.6	179.11	9.02	159.8	31.46	9.83	23.97	154.38	38.84	37.79
0.1	Enc Time	1163	1297	762.6	7.31	102.5	34.74	7.94	19.55	117.8	33.29	31.95
	Dec Time	1081	1078	762.11	6.24	77.57	21.79	6.71	18.36	59.6	31.84	30.09
	Total Time	2244	2375	1524.71	13.55	180.07	56.53	14.65	37.91	177.4	65.13	62.04
0.25	Enc Time	6234	6871	4358	13.35	120.8	68.15	14.02	40.25	131.8	77.51	74.89
	Dec Time	6012	6305	4703	11.68	90.45	50.91	12.02	37.86	67.2	69.84	68.08
	Total Time	12246	13176	9061	25.03	211.25	119.06	26.04	78.11	199	147.35	142.97
0.5	Enc Time	17995	18742	19551	24.12	151.3	122.5	26.03	74.87	160.8	112.81	108.59
	Dec Time	17597	18534	15400	22.65	100.5	96.84	25.07	69.02	89.7	107.94	98.87
	Total Time	35592	37276	34951	46.77	251.8	219.34	51.10	143.89	250.5	220.75	207.46
Hyperspectral Image II												
0.001	Enc Time	3.42	4.33	5.94	2.35	15.97	5.73	2.47	2.94	17.89	3.38	3.21
	Dec Time	1.87	1.52	1.46	1.4	12.18	2.18	1.61	2.79	9.78	3.14	3.04

	Total Time	5.29	5.85	7.4	3.75	28.15	7.91	4.08	5.73	27.67	6.52	6.25
0.005	Enc Time	9.84	5.85	8.5	2.71	75.93	7.36	3.87	6.44	84.67	9.02	8.12
	Dec Time	5.4	2.45	2.77	2.49	66.24	3.21	3.01	6.05	54.94	8.47	7.74
	Total Time	15.24	8.3	11.27	5.2	142.17	10.57	6.88	12.49	139.61	17.49	15.86
0.01	Enc Time	22.53	9.41	10.83	2.88	90.43	16.99	4.29	10.28	99.47	18.37	17.52
	Dec Time	10.01	4.92	3.86	2.71	81.48	6.23	3.27	10.04	66.38	17.89	17.01
	Total Time	32.54	14.33	14.69	5.59	171.9	23.22	7.56	20.32	165.85	36.26	34.53
0.05	Enc Time	250.3	134.4	131.5	4.14	106.55	27.4	5.02	16.02	121.8	28.81	28.02
	Dec Time	207.2	127.8	130.1	3.38	94.49	14.94	3.94	11.35	79.7	25.29	24.62
	Total Time	457.5	262.2	261.6	7.52	201.04	42.34	8.96	27.37	201.5	54.1	52.64
0.1	Enc Time	966.7	570.8	632.6	6.04	125.87	36.27	7.21	18.42	143.8	33.59	33.06
	Dec Time	887.6	717.5	614.3	5.98	106.8	23.01	6.64	17.81	80.3	30.81	30.29
	Total Time	1854.3	1288.3	1246.9	12.02	232.67	59.28	13.85	36.23	224.1	64.4	63.35
0.25	Enc Time	4973	3032	4100	10.24	134.4	96.34	12.21	56.67	155.9	87.29	82.29
	Dec Time	4796	3029	4040	6.74	113.86	58.62	7.18	47.06	87.5	81.08	78.68
	Total Time	9769	6161	8240	16.98	248.26	154.96	19.39	103.73	243.4	168.37	160.97
0.5	Enc Time	12007	10112	12975	17.25	154.41	177.73	18.95	67.74	180.9	109.05	101.94
	Dec Time	11898	9954	12299	14.7	125.56	120.33	15.34	60.13	99.5	101.75	95.39
	Total Time	23905	20066	25274	31.95	279.97	298.06	34.29	127.87	280.4	210.8	197.33
		Hyperspectral Image III										
0.001	Enc Time	4.08	4.03	5.85	2.07	15.97	5.68	2.76	3.19	17.79	5.54	5.02
	Dec Time	1.74	1.39	1.32	1.89	8.43	4.1	2.11	3.02	6.54	4.87	4.61
	Total Time	5.82	5.42	7.17	3.96	24.4	9.78	4.87	6.21	24.33	10.41	9.63
0.005	Enc Time	9.12	5.96	7.87	2.89	75.93	7.78	3.28	4.74	88.54	7.09	6.74
	Dec Time	5.13	2.24	2.44	2.47	66.02	6.02	2.74	3.99	49.87	6.78	6.09
	Total Time	14.25	8.2	10.31	5.36	141.95	13.8	6.02	8.73	138.41	13.87	12.83
0.01	Enc Time	20.18	9.7	11.64	3.34	90.43	8.55	4.01	7.52	101.5	12.87	12.05
	Dec Time	12.51	5.18	5.14	2.69	84.96	7.06	3.02	6.33	79.5	11.49	11.38

	Total Time	32.69	14.88	16.78	6.03	175.39	15.61	7.03	13.85	181	24.36	23.43
0.05	Enc Time	204.3	125.2	89.77	4.57	106.55	19.48	5.31	22.88	119.5	31.65	30.25
	Dec Time	160.3	114.7	80.01	4.46	92.68	14.84	5.19	18.56	87.4	29.81	28.36
	Total Time	364.6	239.9	169.78	9.03	199.23	34.32	10.5	41.44	206.9	61.46	58.61
0.1	Enc Time	1183	775.8	835.9	5.91	125.87	32.46	6.47	30.14	138.8	54.79	51.39
	Dec Time	1074	760.5	827.8	5.59	104.98	21.49	6.37	27.82	100.5	50.99	48.74
	Total Time	2257	1536.3	1663.7	11.5	230.85	53.95	12.84	57.96	239.3	105.78	100.13
0.25	Enc Time	8499	5151	6309	10.41	134.14	70.4	11.91	43.49	149.5	89.32	81.29
	Dec Time	8387	5832	6233	9.27	115.94	48.95	10.34	39.95	108.2	83.67	76.32
	Total Time	16886	10983	12858	19.68	250.08	119.35	22.25	83.44	257.7	182.99	154.61
0.5	Enc Time	29849	18383	23861	16.19	154.41	125.42	17.09	72.62	165.8	121.21	117.62
	Dec Time	26948	15672	23161	14.97	141.97	114.52	16.68	67.23	130.1	118.37	111.95
	Total Time	56797	34055	47022	31.16	296.38	239.94	33.77	139.85	295.9	239.58	229.67
		Hyperspectral Image IV										
0.001	Enc Time	4.56	5.6	7.23	2.39	6.03	5.74	2.89	2.82	7.85	3.03	2.91
	Dec Time	2.41	1.64	1.73	2.02	5.27	2.1	2.24	2.74	4.34	2.81	2.7
	Total Time	6.97	7.24	8.96	4.41	11.3	7.84	5.13	5.56	12.19	5.84	5.61
0.005	Enc Time	15.24	6.23	8.15	2.81	11.53	7.53	3.34	4.44	12.94	4.91	4.77
	Dec Time	9.57	2.33	2.55	2.34	8.26	2.88	2.47	4.28	7.19	4.75	4.42
	Total Time	24.81	8.56	10.7	5.15	19.79	10.41	5.81	8.72	20.13	9.66	9.19
0.01	Enc Time	21.67	10.2	12.64	3.18	18.44	8.93	3.98	5.64	21.18	9.08	8.11
	Dec Time	12.68	5.23	6.11	2.89	14.44	3.91	3.23	5.41	12.78	8.48	7.81
	Total Time	34.35	15.43	18.75	6.07	32.88	12.84	7.21	11.05	33.96	17.56	15.92
0.05	Enc Time	269.6	130.4	98.12	4.3	22.64	18.61	4.88	13.02	24.91	21.35	20.18
	Dec Time	226.5	120.5	89.08	3.74	19.5	11.48	4.29	11.36	17.47	19.48	18.89
	Total Time	496.1	250.9	187.2	8.04	42.14	30.09	9.17	24.38	42.38	40.83	39.07
0.1	Enc Time	1336	893.4	882.3	6.11	25.53	32.45	6.41	18.18	30.58	41.81	40.39
	Dec Time	1241	829.1	866.3	5.96	21.07	21.02	6.57	17.22	20.17	37.29	36.62

	Total Time	2577	1722.5	1748.6	12.07	46.6	53.47	12.98	35.4	50.75	81.1	77.01
0.25	Enc Time	8435	5133	5501	10.35	34.5	69.66	11.38	36.3	38.54	67.28	64.23
	Dec Time	9067	4536	5494	6.62	29.65	48.91	7.08	33.79	24.25	61.3	59.19
	Total Time	17502	9669	10995	16.97	64.15	118.57	18.46	70.09	62.79	128.58	123.42
0.5	Enc Time	27917	17945	18818	17.43	65.13	125.19	19.01	66.91	70.04	111.92	105.32
	Dec Time	25042	17677	18136	12.03	55.03	92.97	12.87	62.31	48.95	109.21	103.09
	Total Time	52959	35622	36954	29.46	120.16	218.16	31.88	129.22	118.99	221.13	208.41
		Hyperspectral Image V										
0.001	Enc Time	1.47	1.4	2.04	0.95	3.17	2.17	1.87	1.40	5.07	3.02	2.95
	Dec Time	0.78	0.53	0.49	0.89	1.55	0.76	0.71	1.29	3.57	2.91	2.41
	Total Time	2.25	1.93	2.53	1.84	4.72	2.93	2.58	2.69	8.64	5.93	5.36
0.005	Enc Time	4.89	2.9	3.70	0.99	4.03	4.84	3.94	1.71	5.54	3.39	3.11
	Dec Time	3.36	1.15	1.13	0.96	1.89	1.67	1.57	1.58	3.94	3.01	2.87
	Total Time	8.25	4.05	4.83	1.95	5.92	6.51	5.51	3.29	9.48	6.4	5.98
0.01	Enc Time	11.30	4.5	6.56	1.52	4.80	5.33	4.34	2.20	6.27	4.06	3.84
	Dec Time	5.62	2.21	3.65	1.39	2.13	2.84	2.94	1.97	5.37	3.32	3.41
	Total Time	16.92	6.71	10.21	2.91	6.93	8.17	7.28	4.17	11.64	7.38	7.25
0.05	Enc Time	113.8	44.0	42.07	1.98	6.11	8.07	6.98	4.14	6.84	8.04	7.48
	Dec Time	100.9	40.69	39.48	2.09	2.31	4.58	4.03	3.67	5.84	7.33	7.02
	Total Time	214.7	84.69	81.55	4.07	8.42	12.65	11.01	7.81	12.68	15.37	14.5
0.1	Enc Time	622.6	270.5	254.4	3.05	7.58	12.94	9.11	6.75	8.89	11.84	11.6
	Dec Time	591.9	268.7	272.4	2.91	3.34	8.72	7.89	6.02	8.02	10.41	10.21
	Total Time	1214.5	539.2	526.8	5.96	10.92	21.66	17	12.77	16.91	22.25	21.81
0.25	Enc Time	3211	3145	1757	5.69	9.83	24.13	14.24	16.12	11.06	28.74	27.07
	Dec Time	2879	3004	1637	5.32	5.84	16.36	12.59	14.89	9.28	26.32	25.51
	Total Time	6090	6149	3394	11.01	15.67	40.49	26.83	31.01	20.34	55.06	52.58
0.5	Enc Time	20522	21256	22415	9.39	11.92	38.28	21.29	29.80	14.28	55.68	53.94
	Dec Time	17695	18111	18942	8.17	9.18	31.38	19.54	26.85	12.67	52.86	51.83

	Total Time	38217	39367	41357	17.56	21.1	69.66	40.83	56.65	26.95	108.54	105.77
		Hyperspectral Image VI										
0.001	Enc Time	1.33	1.38	2.38	0.87	3.95	2.02	1.79	1.32	4.11	2.22	2.01
	Dec Time	0.62	0.53	0.70	0.83	1.29	0.67	0.61	1.34	1.59	2.09	1.94
	Total Time	1.95	1.91	3.08	1.7	5.24	2.69	2.4	2.66	5.7	4.31	3.95
0.005	Enc Time	3.55	2.17	3.53	1.22	5.54	3.46	3.17	2.19	6.29	3.59	3.33
	Dec Time	1.86	0.84	0.97	1.10	2.10	1.23	1.12	2.42	2.68	3.32	3.09
	Total Time	5.41	3.01	4.5	2.32	7.64	4.69	4.29	4.61	8.97	6.91	6.41
0.01	Enc Time	7.09	2.97	4.84	1.43	5.64	4.13	3.69	2.77	6.78	4.48	4.21
	Dec Time	4.24	1.47	1.82	1.26	2.74	1.66	1.48	2.54	2.59	4.01	3.99
	Total Time	11.33	4.44	6.66	2.69	8.38	5.79	5.17	5.31	9.37	8.49	8.2
0.05	Enc Time	63.50	34.88	30.20	2.60	6.59	8.17	6.08	4.68	8.05	8.11	7.74
	Dec Time	49.05	32.07	25.15	2.43	3.39	4.86	4.41	4.43	3.01	7.47	7.05
	Total Time	112.55	66.95	55.35	5.03	9.98	13.03	10.49	9.11	11.06	15.58	14.79
0.1	Enc Time	437.6	295.7	222.1	3.30	8.24	13.61	7.79	7.18	9.28	13.68	13.02
	Dec Time	357.2	247.2	216.6	3.10	5.08	8.71	7.98	6.68	4.21	12.91	12.24
	Total Time	794.8	542.9	438.7	6.4	13.32	22.32	15.77	13.86	13.49	26.59	25.26
0.25	Enc Time	2241	2014	1983	5.22	14.10	25.99	12.91	13.85	16.21	22.35	21.29
	Dec Time	1946	1905	1751	5.27	12.34	19.54	11.29	12.77	9.58	20.68	20.09
	Total Time	4187	3919	3734	10.49	26.44	45.53	24.2	26.62	25.79	43.03	41.38
0.5	Enc Time	6956	7019	7145	10.64	20.99	50.41	17.83	25.96	21.78	48.91	46.09
	Dec Time	6625	6762	6829	9.01	15.07	36.01	29.58	25.11	11.81	46.29	44.58
	Total Time	13581	13781	13974	19.65	36.06	86.42	47.41	51.07	33.59	95.2	90.57

Table 5. Coding Memory requirement (in KB) between the proposed HSICA with the other wavelet transform based set partitioned HSICA

Bit Rate	CA-I [59]	CA-II [60]	CA-III [61]	CA-IV [62]	CA-V [63]	CA-VI [64]	CA-VII [65]	CA-VIII [66]	CA-IX [67]	CA-X [35]	Proposed CA
Hyperspectral Image I											
0.001	26.67	37.33	28.08	4096	8192	96	2318	0	2304	0	0
0.005	102.3	99.21	89.33	4096	8192	96	2318	0	2304	0	0
0.01	232.2	222.7	202.4	4096	8192	96	2318	0	2304	0	0
0.05	1084	1041	991.7	4096	8192	96	2318	0	2304	0	0
0.1	1846	1931	1756	4096	8192	96	2318	0	2304	0	0
0.25	4571	4463	4289	4096	8192	96	2318	0	2304	0	0
0.5	8644	8555	8514	4096	8192	96	2318	0	2304	0	0
Hyperspectral Image II											
0.001	22.58	21.51	22.69	4096	8192	96	2318	0	2304	0	0
0.005	91.12	98.91	91.29	4096	8192	96	2318	0	2304	0	0
0.01	265.9	267.8	266.4	4096	8192	96	2318	0	2304	0	0
0.05	982.4	1036	985.4	4096	8192	96	2318	0	2304	0	0
0.1	2219	2326	2229	4096	8192	96	2318	0	2304	0	0
0.25	5450	5611	5464	4096	8192	96	2318	0	2304	0	0
0.5	10005	9981	9832	4096	8192	96	2318	0	2304	0	0
Hyperspectral Image III											
0.001	25.28	24.94	25.06	4096	8192	96	2318	0	2304	0	0
0.005	101.2	105.8	101.5	4096	8192	96	2318	0	2304	0	0
0.01	205.1	218.9	208.6	4096	8192	96	2318	0	2304	0	0
0.05	1108	1149	1136	4096	8192	96	2318	0	2304	0	0
0.1	1855	1808	1854	4096	8192	96	2318	0	2304	0	0
0.25	4401	4449	4412	4096	8192	96	2318	0	2304	0	0
0.5	7918	7805	7935	4096	8192	96	2318	0	2304	0	0

Hyperspectral Image IV											
0.001	24.67	22.41	24.55	4096	8192	96	2318	0	2304	0	0
0.005	100.8	105.5	101.1	4096	8192	96	2318	0	2304	0	0
0.01	210.9	229.9	214.4	4096	8192	96	2318	0	2304	0	0
0.05	1088	1212	1106	4096	8192	96	2318	0	2304	0	0
0.1	1970	2083	1980	4096	8192	96	2318	0	2304	0	0
0.25	4867	5047	4878	4096	8192	96	2318	0	2304	0	0
0.5	9078	8488	9093	4096	8192	96	2318	0	2304	0	0
Hyperspectral Image V											
0.001	22.27	24.22	22.89	4096	8192	96	2318	0	2304	0	0
0.005	102.7	102.1	102.8	4096	8192	96	2318	0	2304	0	0
0.01	210.0	231.4	213.8	4096	8192	96	2318	0	2304	0	0
0.05	1082	1138	1099	4096	8192	96	2318	0	2304	0	0
0.1	1866	1961	1864	4096	8192	96	2318	0	2304	0	0
0.25	4670	4778	4669	4096	8192	96	2318	0	2304	0	0
0.5	8880	8891	9002	4096	8192	96	2318	0	2304	0	0
Hyperspectral Image VI											
0.001	23.65	23.24	24.19	4096	8192	96	2318	0	2304	0	0
0.005	103.3	106.9	103.0	4096	8192	96	2318	0	2304	0	0
0.01	213.3	239.5	218.0	4096	8192	96	2318	0	2304	0	0
0.05	1128	1207	1156	4096	8192	96	2318	0	2304	0	0
0.1	1923	2038	1925	4096	8192	96	2318	0	2304	0	0
0.25	4906	4954	4916	4096	8192	96	2318	0	2304	0	0
0.5	8392	8425	8245	4096	8192	96	2318	0	2304	0	0

Table 6. Comparative Analysis of the Coding Efficiency (PSNR) between the proposed HSICA with the other wavelet transform based set partitioned HSICA

Bit Rate	CR	CA-I [59]	CA-II [60]	CA-III [61]	CA-IV [62]	CA-V [63]	CA-VI [64]	CA-VII [65]	CA-VIII [66]	CA-IX [67]	CA-X [35]	Proposed CA
Hyperspectral Image I												
0.001	14000	26.28	26.28	26.25	26.14	25.90	26.26	26.41	26.32	26.41	28.71	28.24
0.005	2800	28.95	28.95	28.93	28.71	28.71	28.70	28.66	28.73	28.97	31.24	31.01
0.01	1400	30.08	30.08	30.04	29.99	29.83	29.98	30.01	29.99	30.21	32.84	32.44
0.05	280	34.23	34.23	34.21	34.04	33.81	33.99	34.29	34.06	34.42	36.94	36.37
0.1	140	37.22	37.22	37.20	36.96	37	36.83	37.34	36.87	37.34	40.05	39.32
0.25	56	42.17	42.17	42.16	41.62	41.69	41.34	42.28	41.37	42.38	45.11	44.78
0.5	28	48.02	47.99	47.97	47.01	47.79	47.51	48.11	47.55	48.21	52.21	51.89
Hyperspectral Image II												
0.001	14000	27.11	26.75	27.09	26.83	26.61	26.75	26.87	26.82	26.87	30.24	29.57
0.005	2800	29.45	29.31	29.43	29.27	29.25	29.24	29.41	29.25	29.54	32.01	31.49
0.01	1400	30.28	30.19	30.27	30.27	30.15	30.31	30.53	30.33	30.48	32.94	32.59
0.05	280	33.76	33.61	33.73	33.56	33.59	33.51	33.69	33.54	33.93	35.32	34.99
0.1	140	35.57	35.44	35.56	35.49	35.41	33.45	35.55	35.46	35.69	37.42	37.02
0.25	56	39.30	39.19	39.29	39.26	39.17	39.22	39.37	39.23	39.41	42.76	42.22
0.5	28	43.62	43.65	43.51	43.57	43.26	43.55	43.62	43.58	43.74	46.22	45.76
Hyperspectral Image III												
0.001	14000	27.82	27.49	27.8	27.78	27.28	27.88	28.07	27.92	27.77	29.81	29.44
0.005	2800	30.24	30.09	30.22	30.03	30.03	30.01	30.44	30.02	30.28	32.47	32.04
0.01	1400	31.27	31.14	31.25	31.17	31.1	31.13	31.42	31.14	31.38	33.74	33.51
0.05	280	34.57	34.39	34.55	34.58	34.27	34.44	34.67	34.51	34.63	37.09	36.51
0.1	140	36.63	36.49	36.64	36.42	36.49	36.35	36.74	36.37	36.71	40.34	39.95
0.25	56	40.83	40.63	40.84	40.46	40.59	40.29	40.81	40.31	40.85	43.68	43.27
0.5	28	45.88	45.66	45.87	45.39	45.57	45.13	45.58	45.15	45.92	47.81	47.11

Hyperspectral Image IV												
0.001	14000	28.11	27.94	28.06	28.08	27.88	28.07	28.14	28.16	28.15	30.41	29.96
0.005	2800	30.44	30.32	30.43	30.27	30.03	30.26	30.22	30.28	30.52	33.05	32.49
0.01	1400	31.41	31.29	31.39	31.32	31.1	31.29	31.57	31.43	31.52	34.14	33.48
0.05	280	34.46	34.3	34.45	34.41	34.27	34.25	34.62	34.28	34.53	37.38	37.03
0.1	140	36.43	36.29	36.43	36.25	36.49	36.19	36.51	36.2	36.55	39.47	39.01
0.25	56	40.08	39.93	40.07	39.92	40.59	39.8	40.19	39.84	40.17	43.84	43.37
0.5	28	44.51	44.47	44.5	44.31	44.46	44.22	44.63	44.22	44.62	47.09	46.58
Hyperspectral Image V												
0.001	14000	27.13	26.80	27.10	27.03	26.80	27.06	27.21	27.08	27.31	30.02	29.54
0.005	2800	29.76	29.53	29.73	29.62	29.45	29.55	29.81	29.59	30.11	33.24	32.87
0.01	1400	31.03	30.85	31.00	30.93	30.81	30.87	31.01	30.89	31.95	34.84	34.37
0.05	280	35.23	34.92	35.20	35.11	34.79	35.03	35.34	35.10	36.08	38.59	38.05
0.1	140	37.73	37.49	37.72	37.55	37.43	37.48	37.98	37.49	38.89	41.75	41.12
0.25	56	42.43	42.35	42.43	42.14	41.98	41.92	42.48	41.93	43.02	47.28	46.91
0.5	28	48.51	48.45	48.44	48.48	48.19	47.92	48.29	47.93	50.07	53.51	53.08
Hyperspectral Image VI												
0.001	14000	27.43	27.07	27.38	27.30	27.02	27.40	27.41	27.42	28.08	30.65	30.08
0.005	2800	30.07	29.85	30.05	29.92	29.77	29.86	30.02	29.90	30.56	32.74	32.47
0.01	1400	31.47	31.27	31.45	31.35	31.24	31.31	31.44	31.33	32.28	34.01	33.41
0.05	280	35.78	35.44	35.75	35.57	35.32	35.56	35.84	35.59	36.06	39.05	38.68
0.1	140	38.46	38.20	38.45	38.22	38.07	38.11	38.44	38.14	39.34	42.89	42.33
0.25	56	43.64	43.63	43.64	43.51	43.24	43.29	43.71	43.31	44.12	47.18	46.73
0.5	28	49.68	49.67	49.71	49.69	49.37	49.70	49.78	49.71	50.79	53.51	52.94