SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I5P120

Volume 13 Issue 5, 238-257, May 2026

Original Article

3D-Computationally Efficient Zero Memory Set Partitioned Embedded Block Coding Algorithm for Onboard WMSNs

Neeraj Kumar[1], Kaneez Zainab[2]

[1]*Department of Information Technology, Maharishi University of Information Technology, Lucknow, Uttar Pradesh, India.*
[2]*Computer Science and Engineering, Maharishi University of Information Technology, Lucknow, Uttar Pradesh, India.*

[1]*Corresponding Author : neerajmaurya2509@gmail.com*

Received: 16 February 2026 Revised: 16 March 2026 Accepted: 19 April 2026 Published: 27 May 2026

Abstract *- The hyperspectral image is known for its rich spatial–spectral information. The ability to differentiate between the spectra of various substances is provided by the spectral bands, which are essential for analysing materials. The high-dimensional data volume of hyperspectral images, on the other hand, presents challenges for data storage. This hyperspectral image data needs to be processed and sent to the ground station from the onboard flight sensor. The huge amount of data creates a problem for the transmission channel, sensor performance, and sensor energy management. Thus, a hyperspectral image compression algorithm is required to solve the above-mentioned issues before the image data is transmitted to the ground station. The compression algorithm should have high coding efficiency, fast image data processing speed, low coding memory requirement, and embeddedness. In the past, researchers presented many transform-based hyperspectral image compression algorithms, but either they suffer from high coding memory demand or high coding complexity. Among them, the mathematical transform-based set-partitioned compression algorithm uses the set structure to achieve the compression of a hyperspectral image. 3D-Zero Memory Set Partitioned Embedded Block (3D-ZM-SPECK) is a compression algorithm that requires no coding memory and is at par with coding. However, it suffers from a slightly higher coding efficiency than 3D-Listless SPECK. The proposed compression algorithm, 3D-Computationally Efficient Zero Memory Set Partitioned Embedded Block (3D-CE-ZM-SPECK), uses the same partition rule as 3D-ZM-SPECK, but it lowers the coding complexity by using very little coding memory.*

Keywords *- Compression, Transform Coding, Low Complexity, On-Board Data Processing, Lossy Compression.*

1. Introduction

HyperSpectral (HS) images have a large number of spectral frames ranging from visible to infrared wavelengths, which provide precious information about the object under supervision in the 1D spectral and 2D spatial domains [1]. Each pixel in an HS image corresponds to hundreds of reflected electromagnetic radiation frequency frames [2]. HS image can identify the minor changes in terms of chemical property, temperature change, and water activity of the object in the scene [3-5]. As an emerging field in imaging science, the HS images are used in many applications such as Aerospace [6], Agriculture [7], Atmospheric Sciences [8], Biomedicine [9], Cereals Quality Detection [10], Climate Change Monitoring [11], Corrosion (Nuclear, Steel Structure) [12], Counterfeit Detection [13], Earth Observation [14], Food Quality Analysis [15], Forestry [16], Forensics Research [17], Geology [18], Land Cover Analysis [19], Landmine Detection [20], Medical [21], Mineral Extraction and Identification [22], Military Surveillance [23], Pharmaceutical [24], Plastic Waste Characterization [25], Raw Material Classification [26], Surveying [27], Underwater Hyperspectral Imaging [27], Weather Prediction [28], Identification of Sintered Uranium Oxide (UO2) Fuel Pellets [29] etc. Apart from these applications, remote sensing [30] is the fastest growing area as researchers are developing the computer based algorithms for change detection [31], compression [32], classification [33], denoising [34], dimensionality reduction [35], feature extraction [36], feature identification [37], feature mapping [38], fusion [39], inpainting [40], object (target) detection [41], object identification [42], unmixing [43] and segmentation [44]. The remote sensing HS images are acquired by the satellite-based HS image sensors [45, 46].

Due to the three-dimensional hypercube with massive pixel depth (12 bits or more), image compression with HS image is always challenging [47]. The main reason is from the large volume (more than 150 MB) and 3D data structure,

where a high degree of redundancy in both spectral and spatial domains can be found [48]. The redundancy in the spatial domain is due to the statistical dependencies among pixels present in the same frequency frame [49]. At the same time, the redundancy in the spectral domain exists due to the pixel present in the same spatial coordinate with multiple frequency frames [50]. The main objective of any HyperSpectral Image Compression Algorithm (HSICA) is to reduce the redundancy present in the HS image [51]. Thus, HS image compression is needed to save the sensor memory, the transmission bandwidth, device power, and to reduce sensor complexity [52, 53].

The compression algorithms for the HS images can be classified into two different groups, which are based on the HS image data loss or coding process [54]. Further, HSICAs can be sub-classified into lossy, lossless, and near lossless on the basis of the data loss during the compression of the HS image. There is no data loss for the lossless compression, and these types of HSICAs have a compression ratio equal to 2 to 3, while for near-lossless compression, there is very low HS image data loss that happens, and it has a slightly higher compression ratio than lossless compression-based HSICAs [55, 56]. The lossy compression-based HSICA has a high compression ratio, but this comes at the cost of HS image data loss [57, 58].

HSICAs can be sub-classified (reference to the coding process) into Predictive Coding (PC) [59], Vector Quantization (VQ) [60], Transform Coding (TC) [61], Compressive Sensing (CS) [62], Tensor Decomposition (TD) [63], and Learning-Based Compression (LC) [64].

PC based HSICA uses different types of predictors to achieve the compression. Through the predictor, it exploits the correlation of the HS image and calculates the predictive error through the entropy coding methods (Huffman coding, or arithmetic coding, etc.). It has low coding complexity with near-average coding efficiency. It works only for lossless compression, and if any transmission error occurs in the transmission process, the whole reconstructed HS image is prone to corruption [59].

The VQ-based HSICA uses the specific codebook (also known as spectral libraries) for each HS image compression. The same codebook is present at the encoder and decoder ends. The VQ-based HSICA has three stages known as code book generation procedure, encoding process (at encoder end), and decoding process (at decoder end). The input HS image is split into independent blocks (same size) for the codebook. In the codebook generation procedure, the individual code of the codebook gets training (assigned) for the specific block of the HS image. All blocks of input HS image are encoded through the code book and transmitted through the communication channel to the decoder end. The decoder decodes the received HS image data from the encoder end and decodes it through the code book. Through VQ-based HSICA, only lossless compression is performed [65, 66].

The CS based HSICA implemented the compression of the HS image in three steps. The encoder senses the HS image and converts it into a small 2D matrix through the pre-defined samples [62, 67].

The TD-based HSICA uses tensors to achieve the compression of the HS image. A tensor is an n-dimensional matrix, and it can be decomposed in a very easy way. The HS image is stored in the tensors, and associated HSICA is applied on the same to decompose into a low-dimensional tensor. This low-dimensional tensor is encoded and transmitted to the channel [63].

The LC-based HSICA uses a different type of neural network or machine learning processes to achieve the HS image compression. These algorithms have very high coding complexity (slow processing) and high coding efficiency. These compression algorithms have high coding memory demand due to the training of the learning algorithm or neural network [4, 68].

The TC-based HSICAs utilized the mathematical transform to translate the HS image from the time domain to the frequency domain [69]. There are many mathematical transforms used, but the use of a transform depends on the compression algorithm requirements. Fourier, wavelet, and curvelet transforms are mainly preferred for the same [70]. The forward mathematical transform converts the HS image into the frequency domain, and then the compression algorithm is applied to the transform HS image at the encoder end [71].

Later, the encoder transmits the compressed HS image data to the decoder end through the communication channel [72]. The decoder applied the compression algorithm and decompressed the received HS image data. After that, the inverse mathematical transform is applied, and the HS image is converted into the time domain format [73].

The following sections of this presented manuscript are organized as follows: In Section 2, the different types of mathematical transform HSICAs are outlined, outlining the research areas under consideration. Section 3 describes our proposed compression algorithm, 3D-Computationally Efficient Zero Memory Set Partitioned Embedded Block Coding Algorithm.

In Section 4, we present the results of our experiment findings and compare the proposed HSICA with the other state-of-the-art wavelet transform-based HSICAs on different performance metrics. Finally, we summarize the conclusions of our work in the last section of the manuscript.

2. Related Work

The bulk of coefficients are zero or close to zero valued because of the wavelet transform's superior energy clustering feature and hierarchical nature [74, 75]. The utilization of low-valued clusters of coefficients is essential for wavelet-based image compression methods to be effective. They are organized into block cubes (zero block cubes) or trees (zero tree) or block trees (zero block cube tree) [76, 77].

The wavelet transformed coefficients that correspond to the same orientation and spatial position are clustered in zero-tree compression methods to create the Spatial Orientation Tree (SOT) [89]. The SOT is referred to as a zerotree if it possesses zero coefficients with magnitudes greater than a predetermined threshold. 3D Set Partitioning in Hierarchical Trees (3D-SPIHT) [78] and 3D No List SPIHT (3D-NLS) [81] are the state-of-the-art compression algorithms that belong to this category.

The wavelet-transformed image is divided into contiguous block cubes by the zero block cube compression techniques, and each block cube is subjected to a significance test. A block cube with no significant coefficients with regard to a specified threshold is known as a zeroblock cube. 3D Set Partitioned Embedded Block (3D-SPECK) [79] and 3D Listless SPECK (3D-LSK) [80] are state-of-the-art compression algorithms that belong to this group.

The 3D-Wavelet block tree coding (3D-WBTC) [82] has focused on integrating the nice properties of both the zerotree and zeroblock cube. 3D-WBTC divides the wavelet-transformed image into coefficient block cubes. Then block cube trees are formed with the roots in the topmost sub-band in a zerotree fashion [53, 82]. Nearly all of the methods discussed above mentioned data-dependent auxiliary lists to maintain track of the order of the transform coefficients that have been or are still being coded, such as lists of unimportant coefficients, insignificant sets, and significant coefficients [92]. These lists' items keep growing as bit rates rise, consuming a sizable amount of coding memory, making it difficult to use them on inexpensive sensor nodes with little coding memory [88]. The following section discusses the many methods used by the wavelet-based image compression algorithms to avoid lists.

Modern image compression methods with widespread adoption use fixed-size state tables instead of data-dependent variable-size lists in a variety of listless implementations. 3D-LSK [80] uses the four types of markers, while 3D-NLS [81] makes use of eight types of markers. Thus, 3D-NLS [81] had a memory requirement twice that of the 3D-LSK [80]. Due to the block cube tree structure, 3D-LMBTC [83] requires less coding memory than 3D-NLS [81] and 3D-LSK [80]. 3D-ZM-SPECK [84] is the particular case of the zero block cube compression algorithm, which does not require coding memory. This is because it uses linear indexing to track the significance of the sets or coefficients [85]. But, it has higher complexity than 3D-LSK [80] due to finding the significance of the sets or coefficients for each bit plane. This can be reduced by dividing the HS images into block cubes and applying the 3D-ZM-SPECK [84] in an independent way to each block cube. It reduces the complexity but also the coding efficiency. One more way to lower the complexity of 3D-ZM-SPECK [84] is through the use of a smaller number of recursive significance tests of sets.

Coding efficiency of the above-mentioned listless HSICAs is relatively less than that of the list-based HSICAs [86]. The curvelet transform is used to solve the problem of the low coding efficiency of the listless HSICAs. The curvelet transform is a special type of multiscale transform that can represent the edges and curves in the HS images more effectively than the dyadic wavelet transform. However, the curvelet transform increases the complexity of the HS image transform process, and the requirement for transform memory has also increased. Table 1 gives a short summary of the mathematical transform-based set partitioned HSICAs.

Table 1. Microscopic summary of different transform-based set partitioned HSICA

HSICA	Ref	Year	Partition Type	Coding Memory	Coding Efficiency	Data Processing
3D-SPIHT	[78]	2004	Block Cube	List (3)	High	Slow
3D-SPECK	[79]	2006	Tree	List (2)	High	Slow
3D-LSK	[80]	2010	Block Cube	Listless	Moderate	Fast
3D-F-SPIHT	[92]	2012	Tree	List (3)	High	Slow
3D-NLS	[81]	2013	Tree	Listless	Moderate	Fast
3D-SDB-SPIHT	[92]	2017	Tree	List (3)	High	Slow
3D-WBTC	[82]	2019	Block Cube Tree	List (3)	Moderate	Fast
3D-LMBTC	[83]	2019	Block Cube Tree	Listless	Moderate	Fast
3D-ZM-SPECK	[84]	2022	Block Cube	Listless	Moderate	Fast
3D-LCBTC	[85]	2022	Block Cube Tree	List (2) & Markers	Moderate	Medium
3D-LBCTC	[86]	2022	Block Cube Tree	Listless	Moderate	Fast

3D-BPEC	[87]	2023	Tree	Array (6)	Moderate	Medium
3D-LEZSPC	[48]	2023	Tree	Listless	High	Fast
3D-BCP-ZM-SPECK	[88]	2023	Block Cube	Listless	Moderate	Fast
3D-M-ZM-SPECK	[73]	2023	Block Cube	Listless	Moderate	Fast
FrWF-ZM-SPECK	[75]	2023	Block Cube	Listless	Moderate	Fast
3D-MELS	[89]	2023	Block Cube Tree	Listless	Moderate	Fast
3D-LBCSPC	[76]	2024	Block Cube	Listless	Moderate	Slow
3D-LMZC	[90]	2024	Tree	Listless	High	Fast
BFrWF-ZM-SPECK	[91]	2025	Block Cube	Listless	Moderate	Fast
SFrWF-ZM_SPECK	[92]	2025	Block Cube	Listless	Moderate	Fast
3D-SLS	[92]	2025	Block Cube Tree	Listless	Moderate	Fast
LBFrWF-LC-ZM-SPECK	[36]	2026	Block Cube	Listless	Moderate	Fast

3. 3D-Computationally Efficient Zero Memory Set Partitioned Embedded Block (3D-CE-ZM-SPECK)

The proposed HSICA is designed to reduce the coding complexity of the 3D-ZM-SPECK [84], which occurred due to the finding of a significant sub-band for each bit plane. The input HS image 'X' is transformed with the 'L' level 3D dyadic wavelet transform. For a one-level wavelet transform, 8 sub-bands are generated, of which one LLL sub-band (called the approximation sub-band), while the other seven sub-bands are considered as detail sub-bands.

For the next level of wavelet transform, the approximation sub-band is again partitioned into eight sub-bands. The 'L' level wavelet transform HS image has a total of (7L+1) sub-bands. Transform coefficients are arranged in a hierarchical manner in which detail sub-bands are organized in seven different orientations, which are denoted as $LLH_\alpha, LHL_\alpha, LHH_\alpha, HLL_\alpha\, HLH_\alpha, HHL_\alpha \,\&\, HHH_\alpha$; $\alpha = 1,2,\dots\dots, L$ For the encoding of the coefficients, the initial sub-bands are arranged as $S_\phi^\alpha \; and \; I^\alpha$. After that, the coefficients are encoded through the proposed HSICA.

3D-CE-ZM-SPECK computed the highest value of each sub-band in the transform HS image and stored it in the buffer memory. The total number of sub-bands present in the transform HS image is (7L+1). Thus, the size of the buffer memory is 8(7L+1), where each sub-band has allocated 8 bytes to store the maximum value of each sub-band present in the transform HS image. The buffer memory is denoted as $\mathcal{M}$. The ith element of the buffer memory is denoted by the $\mathcal{M}(i)$. The i=0 denotes the LLL sub-band, while the values for the other sub-bands are calculated as $i = [7(L-\alpha) + \phi]$. It depends on the pyramidal position and orientation of sub-bands. Through the use of buffer memory, the computational complexity of the proposed HSICA is reduced significantly, which reduces the sensor power consumption and requirement of the hardware resources.

Coding process of the proposed HSICA is initiated with the comparison of the values of the buffer memory of initial $\mathcal{M}(0)$ and $\max[\mathrm{M}(i); i = 1,2,3,\dots\dots\dots\dots, 7L+1]$ against the first (most significant) bit plane. Through this, the significance of the first 'S' and 'I' is identified as S_0^L & I^L. If 'S' set is significant to the current threshold, then it will be octa-partitioned into the eight new 'S' sets. If 'I' set is significant to the current threshold, then it will be partitioned into the seven new 'S' sets. S_1^{L-1} ; S_2^{L-1} ; S_3^{L-1} ; S_4^{L-1} ; S_5^{L-1} ; S_6^{L-1} & S_7^{L-1} and new I^{L-1} set.

Now $\mathcal{M}(1), \mathcal{M}(2), \mathcal{M}(3), \mathcal{M}(4), \mathcal{M}(5), \mathcal{M}(6) \& \mathcal{M}(7)$ and $\max[\mathrm{M}(i); i = 8,9,10,\dots\dots\dots\dots, 3L+1]$ are used to calculate the significance of the newly formed 'S' and 'I' set. This process is repeated for other bit planes until the last bit plane is executed or until the bit budget is available. Through this process (avoiding the comparison for sets or coefficients for each bit plane), the proposed HSICA lowers the computational complexity significantly.

The significance of the 'S' (S_ϕ^α) set and 'I' (I^αThe set is calculated against the current threshold, which is computed as in Equations 1 and 2

If 'S' set is significant against the current threshold, then it gets partitioned (octa partitioning) into the eight new 'S' sets, each having half the dimension of the previous 'S' set. The significant 'I' is partitioned into seven new 'S' sets with one new 'I' set through hexacontatetra band partitioning. The testing of the significance of new sets is performed with the help of Equations 1 and 2. However, Equation 3 is used to determine if the sets produced by octa partitioning or the significance of a coefficient are significant.

$$Z_n\left(S_\phi^\alpha\right) = \begin{cases} 0 & if \quad \mathrm{M(i)}_{i=7(L-\alpha)+\phi} < T \\ 1 & if \quad T < \mathrm{M(i)}_{i=7(L-\alpha)+\phi} < 2T \\ NULL & if \quad \mathrm{M(i)}_{i=7(L-\alpha)+\phi} < 2T \end{cases} \tag{1}$$

$$Z_n(I^\alpha) = \begin{cases} 0 & if \quad \mathrm{M(i)}_{i=7(L-\alpha)+\phi:7L} < T \\ 1 & if \quad T < \mathrm{M(i)}_{i=7(L-\alpha)+\phi:7L} < 2T \\ NULL & if \quad \mathrm{M(i)}_{i=7(L-\alpha)+\phi:7L} < 2T \end{cases} \tag{2}$$

$$Z_n(S) = \begin{cases} 1 & if \quad T \le \max_{i \in S}(|S(i)|) < 2T \\ 0 & if \quad \max_{i \in S}(|S(i)|) < T \\ NULL & if \quad \max_{i \in S}(|S(i)|) > 2T \end{cases} \tag{3}$$

3.1. Coding Complexity of the Proposed HSICA

In order to give a detailed analysis of the computational complexity between the 3D-ZM-SPECK [84] and the proposed HSICA, it is necessary to give a comparative analysis of the different computations performed at different stages of the HSICA. These stages with the associated computations are covered in Table 2.

The first two computational steps (C_0 and C_1) are repeated only once at the initialization of the HSICA, while the last three steps (C_2, C_3, and C_4) are repeated for each bit plane. The total number of computations ($\mathbb{C}$) for the significance testing of sets and coefficients for the 3D-ZM-SPECK ($\mathbb{C}_{3D-ZM-SPECK}$) and proposed HSICA ($\mathbb{C}_{3D-CE-ZM-SPECK}$) is covered in Equations 3, 4, and 5.

Equations 4 and 5 gives all computations at the worst case scenario (assuming all sets are significant in each coding bit-planes).

For the best-case scenario, in which only S_0^L The set is significant in the last bit-plane and set I^L remains insignificant throughout the coding, the number of comparisons needed in the ZM-SPECK and the proposed algorithms are given by Equations 7 and 8, respectively.

Pseudo code associated with the proposed HSICA is covered in Table 3. Figure 1 shows the important stages of the compression process of the proposed HSICA.

Table 2. Detailed analysis of the computational complexity at different stages of the encoding process of 3D-ZM-SPECK and 3D-CE- ZM-SPECK

	Computational Steps	**HSICA**	
		3D-ZM-SPECK [84]	**3D-CE-ZM-SPECK**
C_0	To calculate the significant coefficient in each sub-band	0	7L + 1
C_1	To calculate the initial threshold of the transform HS image	N^3	N^3
C_2	For significant testing of S_0^L & I^L	N^3	7L + 1
C_3	For significant testing of new sets after hexacontatetra band partitioning of the significant I^α	$\sum_{\alpha=2}^{L}\left[N^3 - \left(\frac{N}{2^\alpha}\right)^3\right]$	$7 + \sum_{\alpha=2}^{L} 7(\alpha-1)$
C_4	For significant testing of the last seven sets $[S_1^1\,;\, S_2^1\,;\, S_3^1\,;\, S_4^1\,;\, S_5^1\,;\, S_6^1\,;\, S_7^1]$	$7\left(\frac{N}{2}\right)^3$	7

$$\mathbb{C} = C_0 + C_1 + n[C_2 + C_3 + C_4] \tag{4}$$

$$\mathbb{C}_{3D-ZM-SPECK} = N^3 + n[N^3 + \textstyle\sum_{\alpha=2}^{L}\left[N^3 - \left(\frac{N}{2^\alpha}\right)^3\right] + 7\left(\frac{N}{2}\right)^3] \tag{5}$$

$$\mathbb{C}_{3D-CE-ZM-SPECK} = (7L+1) + N^3 + n[(7L+1) + 7 + \sum_{\alpha=2}^{L} 7(\alpha-1)+7] \quad (6)$$

$$\mathbb{C}_{nz} = N^3 + n_\alpha(N^3) \quad (7)$$

$$\mathbb{C}_{nL} = N^3 + (7L+1) + n_\alpha(7L+1) \quad (8)$$

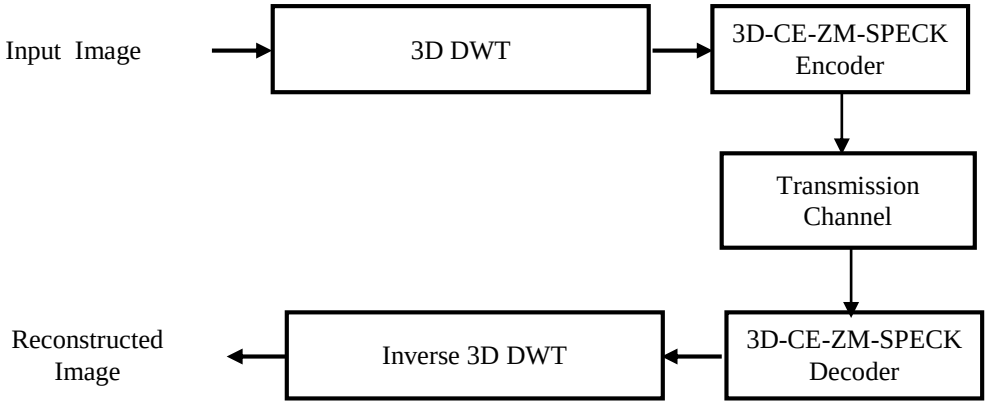

Fig. 1 Flow process of the proposed compression algorithm for hyperspectral image

Table 3. Pseudo code of the Encoder of the proposed HSICA

Algorithm: Encoding process of 3D-CE-ZM-SPECK
Input: The input HS image 'X' is transformed with 'L' wavelet level, and the transformed HS image is converted to a 1D array 'Γ' through linear indexing
Output: Encoded Embedded Bit Stream

INITIALIZATION

Set: Number of transform coefficients in linear array $\lambda = length$ [Γ]
Set : Root set length = η_r
Set : Small set length = η_0
Set: Number of bit planes in transform HS image $n = \log_2[\max(\Gamma)]$
Set: Initial threshold $T = 2^n$
Set: Starting Index β = 0
Set: Initial set length η = η_r

SORTING PASS

```
while (β < λ)
{
    if {(β = η) && (β ≥ ηr)}
        {
        Process S(S_φ^α)
        else
            Process I(I^α)
        }
}
```

QUANTIZATION STEP

```
{
    n = (n -1)
}
```

FUNCTIONS DESCRIPTION

Process S ()

```
{
    Output Z_n(S_φ^α)
    {
        if (Z_n(S_φ^α) = 0)
        {
            then
            {
                β = (β + η)
            }
            else
            {
                if (η > η0)
                    η = (η/8)
                    else
                        {
                         Pscan(β,Ω)
                        }
            }
        }
        EvalSL(β,η)
    }
}
```

Process I ()

```
{
    Output Z_n(I^α)
    if (Z_n(I^α)) = 0
        {
```

```
            then
                β = λ
        }
}

Pscan()
{
    for (j = 0 : 7)
        do
        {
            Output Z_n{Γ(β + j)}
            if [Z_n{Γ(β + j)} = 1]
            {
                then
                    Output: sign bit
                else [Z_n{Γ(β + j)} =
                NULL]
                        then
                            Output: n^th MSB of
                            Γ(β + j)
                            β = (β + η)
                    }
            }
}

EvalSL()
{
    while bitand (β,7η) = 0
        {
            η = 8*η
        }
}
```

4. Results and Discussion

This section presents a simulation assessment of the performance of the proposed HSICA with other state-of-the-art HSICAs on the basis of different performance metrics for coding complexity, coding memory, and coding efficiency. An extensive amount of simulation experiments has been performed on the HS images acquired by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS), HYperspectral Digital Image Collection Experiment (HYDICE), NASA Earth Observing (EO-1), and Reflective Optics System Imaging Spectrometer (ROSIS) HS image sensors [73, 75, 85]. In this section, the obtained results are shown and discussed.

4.1. Experimental Hyperspectral Image Dataset

In the present study, four publicly available HS image datasets are used, which are the Washington DC Mall HS image (acquired by HYDICE), the Botswana HS image (acquired by NASA EO-1), the Pavia Center HS image (acquired by ROSIS), and the Jasper Ridge HS image (acquired by AVIRIS).

Washington DC Mall HS image (Dataset I): It has 191 spectral frequency frames with a spatial dimension of 1280 by 307. It has a pixel depth of 14, a spectral resolution of 10 nm, and a spatial resolution of 3 mt to 4 mt.

Botswana HS image (Dataset II): It has 242 spectral frequency frames with a spatial dimension of 1476 by 256. It has a pixel depth of 16, a spectral resolution of 10 nm, and a spatial resolution of 30 mt.

Pavia Center HS image (Dataset III): It has 102 spectral frequency frames with a spatial dimension of 1096 by 1096. It has a pixel depth of 13, a spectral resolution of 5 nm, and a spatial resolution of 1.3 mt.

Jasper Ridge HS image (Dataset IV): It has 224 spectral frequency frames with a spatial dimension of 100 by. It has a pixel depth of 13, a spectral resolution of 10 nm, and a spatial resolution of 4 mt to 20 mt.

The HS images are cropped to the top left corner to the size of the cube of '128 x 128 x 128,' and zero padding is done if required.

4.2. Experimental Image Quality Assessment

To evaluate the performance of the proposed HSICA compared to the state-of-the-art HSICA. The complexity of any HSICA is measured by the total number of different computations (logical, arithmetic, algebraic) performed by the compression algorithm. Highly complex compression algorithms have a lot of complex computations, and performing these computations requires time [89]. Thus, the algorithm's running time is directly proportional to the complexity of the algorithm. In a compression algorithm, the algorithm running time is the sum of the time required to perform the encoding (at the encoder end) and the decoding (at the decoder end) time. The encoder encodes the transform coefficients through the encoding process of the HSICA and generates the output bit stream. The decoder decodes the output bit stream received from the encoder. All the hyperspectral image compression algorithms are executed on the same hardware platform (11th Generation i5 processor, 2.4 GHz, 20 GB RAM) with the use of Matlab simulation software.

Coding memory is the memory required by the compression algorithm to store the output bits generated by the encoder and decoder. For the list-based hyperspectral image compression algorithms, 3D-SPECK (Coder I) [79], 3D-SPIHT (Coder II) [78], 3D-WBTC (Coder III) [82]. The lists are used to track the significance/insignificance of

coefficients or sets. For state marker (listless) based hyperspectral image compression algorithms 3D-LSK (Coder IV) [80], 3D-NLS (Coder V) [81], 3D-LMBTC (Coder VI) [83], 3D-ZM-SPECK (Coder VII) [84], 3D-M-ZM-SPECK (Coder VIII) [73], they required a fixed-size coding memory depending upon the size of the HS image.

The coding efficiency of the HS image compression is performed by the multiple performance metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity (SSIM) index, Feature SIMilarity (FSIM) index, and Compression Ratio (CR) [92-94]. The CR is a unitless parameter that is the ratio between the size of the original HS image and the size of the reconstructed HS image [92]. Mathematically, it is defined as in the Equation. 9

Bit Rate (BR) is defined as the ratio of the bits required to represent a pixel in the original HS image to the CR [92]. Mathematically, it is defined as in Equation 10.

The Peak Signal-to-Noise Ratio (PSNR) [93] is a unitless engineering term that refers to the ratio of an HS image signal's highest achievable strength to the power of distortion that affects the representation's HS image quality. The original HS image and the reconstructed image are denoted as M(x,y,z) and N(x,y,z). PSNR can be mathematically obtained as Equation 11, and MSE is obtained as Equation 12.

SSIM is another coding efficiency parameter that is used to define the similarities between the original HS image and the reconstructed HS image [94-96]. SSIM has been shown to correlate well with the quality as perceived by humans for different types of distortions. Mathematically, it is defined as in [97].

4.3. Discussion

The five-level wavelet transform is applied to the original HS image. The transform HS image coefficients are quantized to the nearest integers. The 3D transform HS image is converted to the linear array through the Morton mapping [88, 98]. The HSICA is applied to this linear array to achieve the compression of the HS image.

$$CR = \frac{Size\ of\ Original\ HS\ image}{Size\ of\ Reconstructed\ HS\ image\ through\ compressed\ bit\ stream} \tag{9}$$

$$BR = \frac{Bits\ per\ pixel\ \ in\ original\ HS\ image}{CR} \tag{10}$$

$$PSNR(M,N) = 20\ \log_{10}\left[\frac{M_{max}}{MSE}\right] \tag{11}$$

$$MSE = \frac{1}{N_{pix}} \sum_x \sum_y \sum_z [M(x,y,z) - N(x,y,z)]^2 \tag{12}$$

4.3.1. Coding Complexity

The coding complexity of any HSICA is observed from the time taken by the HSICA form encoding and decoding of the coefficients [94]. It has been known that list-based HSICA had more execution time because of several computation executions, such as read operation, write operation, memory access operation, etc. [99]. The listless HSICA uses different types of markers to define the status of the coefficient or set. The encoding time is always greater than the decoding time, as the decoder is skipping the block cubes without testing. From Table 4 (encoding time) and Table 5 (decoding time), it is clear that the proposed HSICA has a lower computation time than all other HSICAs except 3D-LSK. This is because 3D-LSK uses the markers, but proposed HSICA uses the linear indexing to identify the sets. The proposed compression algorithm complexity is independent of the dimension of the HS image.

4.3.2. Coding Memory

When implementing HS image compression algorithms on the onboard HS image sensor, one of the most important factors that needs to be taken into consideration is the amount of coding memory available. The coding memory is the memory required by the compression algorithm to store the current status (significant/insignificant) of the coefficients or sets. The main difference between the proposed compression algorithm and 3D-ZM-SPECK lies in the significance check of the sets or coefficients. The proposed compression algorithm requires a little memory to store the maximum threshold of all sub-bands of the transform HS image. For the 'L' level wavelet transform (7L+1 sub-bands), the memory required to store the coefficients is equal to 8(7L + 1). For the low-level calculation (logical, arithmetical, etc.), very little dynamic memory is required. From Table 6, it is clear that 3D-ZM-SPECK outperforms, but the proposed compression algorithm requires a little memory, which is less than 10 KB (memory present in low-resource sensors). It is also clear that listless compression algorithms have fixed memory, while list-based compression algorithms' memory requirement changes multi-fold with the increase in the bit rate. It is also observed that listless compression algorithms perform better for the high bit rates, while list-based

compression algorithms have better performance at the low bit rates (bpppb less than 0.5).

4.3.3. Coding Efficiency

The coding efficiency (also known as subjective quality assessment) of any compression algorithm is determined by the PSNR, SSIM, and FSIM. Table 7 exhibits the comparative analysis of the PSNR with the state-of-the-art HSICAs and the proposed compression algorithm for the nine different bit rates. It has been observed that the proposed compression algorithm is almost identical (in reference to coding efficiency) to the listless compression algorithms, but it has a bit lower performance than the list-based HSICA 3D-SPECK, 3D-SPIHT, and 3D-WBTC. Table 8 gives the comparative analysis on the Bjontegaard metric calculation (also as BD-PSNR gain) with the other HSICAs. Bjontegaard metric presents a comparative overview of the PSNR with the associated bit rate. Table 9 delivers the insight analysis of the SSIM, while Table 10 covers the FSIM.

The visual representation of the Washington DC MALL HS image (before compression & after compression process) for the four sub bands (frame 30, 60, 90, and 120) is shown in Figure 2.

Table 4. Summary of encoding time for compression algorithms

Bit Rate	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]	Coder IX [Proposed]
	Dataset I								
0.025	3.09	1.44	1.71	0.43	0.51	1.33	0.83	0.49	0.54
0.05	6.23	2.71	2.78	0.55	0.65	1.90	1.11	0.61	0.63
0.1	25.10	7.50	6.50	0.80	0.91	3.90	1.78	0.88	0.84
0.2	57.90	25.80	24.80	1.10	1.21	5.10	2.81	1.17	1.15
0.25	104.58	31.19	29.92	1.71	1.79	10.61	5.05	1.77	1.76
0.5	414.80	140.10	211.20	2.50	2.64	11.30	7.41	2.61	2.59
0.75	950.77	370.29	713.02	3.57	3.88	17.22	10.47	3.71	3.69
1	1497.50	575.00	804.00	4.41	4.57	21.12	13.21	4.49	4.48
2	3822.03	1426.61	4409.82	11.55	14.58	40.09	23.39	12.91	12.31
	Dataset II								
0.025	5.19	1.08	2.05	0.51	1.99	2.33	1.55	0.91	0.84
0.05	8.01	2.19	3.68	0.94	9.43	3.08	2.08	1.74	1.49
0.1	42.11	5.47	7.26	2.01	12.38	4.79	3.49	3.12	2.84
0.2	68.31	17.14	20.14	4.27	14.01	8.64	5.26	4.98	4.78
0.25	91.77	23.38	67.74	7.24	17.56	10.44	9.09	8.17	8.02
0.5	402.48	100.15	179.99	9.95	18.22	19.91	11.76	10.92	10.74
0.75	1066.62	581.78	682.40	12.17	20.19	29.03	17.10	16.68	15.29
1	1320.53	641.34	875.01	14.96	22.96	35.17	19.04	18.07	17.08
2	5495.06	2771.88	4096.40	26.44	41.94	75.21	38.87	34.14	32.17
	Dataset III								
0.025	5.34	1.96	2.15	0.57	3.34	2.42	1.10	1.03	0.91
0.05	10.17	8.34	3.31	2.03	7.72	4.62	3.69	3.12	2.47
0.1	24.02	17.65	6.54	3.65	10.32	7.78	4.34	4.03	3.94
0.2	69.81	22.20	14.58	5.15	11.26	9.16	6.02	5.87	5.55

0.25	91.62	38.56	37.30	7.70	14.92	12.26	8.16	7.91	7.84
0.5	371.82	187.18	198.26	9.14	17.71	17.57	11.38	10.57	10.08
0.75	955.11	440.87	597.21	11.01	21.48	24.43	14.32	12.86	12.14
1	1553.24	715.10	1011.40	13.69	24.92	27.96	18.64	16.28	15.84
2	4770.03	2338.89	3882.91	27.52	39.89	55.15	36.57	32.14	31.04
	Dataset IV								
0.025	3.01	1.34	1.66	0.49	0.53	1.38	0.86	0.51	0.59
0.05	6.01	2.55	2.36	0.56	0.67	2.06	1.14	0.63	0.74
0.1	21.10	7.60	6.40	0.90	1.09	3.00	1.77	1.03	1.21
0.2	54.20	20.60	17.70	1.20	1.34	5.20	2.84	1.27	1.47
0.25	97.10	37.92	21.35	1.84	1.99	8.72	4.73	1.91	2.08
0.5	315.30	101.60	182.40	2.60	2.74	10.90	6.24	2.72	2.91
0.75	705.45	267.31	530.54	3.30	3.77	17.79	10.63	3.51	3.71
1	757.30	425.40	942.80	5.10	5.34	22.70	15.84	5.24	5.33
2	3255.30	1213.10	2380.40	9.55	14.20	59.26	56.42	13.67	14.05

Table 5. Summary of decoding time compression algorithms

Bit Rate	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]	Coder IX [Proposed]
	Dataset I								
0.025	0.79	0.99	0.75	0.38	0.42	0.64	0.79	0.64	0.51
0.05	1.09	2.58	1.36	0.47	0.57	1.00	1.01	0.86	0.59
0.1	17.40	6.10	5.02	0.70	0.79	2.30	1.71	1.15	0.74
0.2	48.80	24.80	22.50	1.07	1.07	3.30	2.71	2.12	1.02
0.25	88.32	28.85	17.44	1.64	1.67	6.98	4.78	4.18	1.67
0.5	339.10	135.40	191.70	2.20	2.31	7.70	6.82	5.87	2.41
0.75	899.76	312.97	612.75	2.94	3.01	13.84	9.01	9.01	3.55
1	1289.70	504.02	774.08	3.70	4.24	15.50	12.31	10.97	4.24
2	3480.44	1196.24	4021.39	10.08	14.02	31.28	21.89	19.43	12.03
	Dataset II								
0.025	4.03	0.87	1.43	0.44	1.14	1.34	1.43	0.79	0.71
0.05	7.21	1.67	2.49	0.81	8.66	2.01	1.74	1.61	1.08
0.1	29.82	4.09	5.05	1.59	10.48	3.54	3.08	2.94	2.71
0.2	49.95	15.09	15.70	3.92	12.57	6.61	4.51	4.02	3.96
0.25	69.76	21.09	52.75	6.74	16.19	8.45	8.24	7.94	7.57
0.5	347.89	89.45	167.04	9.12	17.04	16.58	10.19	10.08	9.75
0.75	948.92	505.29	598.09	11.77	19.53	27.55	15.22	15.74	15.02
1	1227.43	600.26	804.18	13.98	20.84	32.83	17.79	17.12	16.88

2	4872.53	2525.11	3785.90	24.84	39.73	70.88	34.37	32.97	31.77
	Dataset III								
0.025	3.06	0.77	1.12	0.49	2.89	2.04	1.01	0.92	0.8
0.05	4.49	6.04	1.85	1.87	5.01	4.31	3.31	2.89	2.31
0.1	13.33	15.02	5.06	3.24	9.14	7.32	3.97	3.84	3.65
0.2	53.56	20.33	12.55	4.58	10.08	8.47	4.84	4.44	4.64
0.25	60.84	33.43	31.45	7.02	14.17	11.87	7.21	7.01	7.06
0.5	315.94	180.19	186.22	8.48	16.91	16.79	10.21	9.97	9.25
0.75	860.41	412.32	546.32	10.26	20.48	22.89	13.86	12.47	12.01
1	1445.08	684.96	976.09	12.99	23.42	23.81	18.01	15.84	15.24
2	4357.62	2151.76	3423.52	25.74	37.19	50.21	34.92	31.03	30.52
	Dataset IV								
0.025	1.89	1.22	1.19	0.38	0.47	0.65	0.83	0.44	0.41
0.05	3.50	1.98	2.04	0.47	0.58	1.02	1.12	0.52	0.5
0.1	15.30	7.41	4.57	0.75	0.97	1.90	1.73	0.94	0.88
0.2	37.03	17.60	15.30	1.10	1.22	3.70	2.70	1.15	1.12
0.25	72.40	31.06	19.45	1.14	1.54	6.20	4.11	1.47	1.31
0.5	290.80	98.50	178.40	2.50	2.59	9.30	6.06	2.57	2.54
0.75	615.41	232.00	418.91	3.05	3.41	19.56	9.51	3.33	3.11
1	726.90	421.30	923.10	4.30	5.02	21.80	12.31	4.91	4.71
2	3062.80	1132.20	2068.90	6.69	13.54	47.20	45.18	12.65	12.47

Table 6. Summary of coding memory (in KB) for compression algorithms

Bit Rate	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]	Coder IX [Proposed]
	Dataset I								
0.025	54.87	54.42	55.21	512	1024	12	0	32	0.28
0.05	136.10	145.30	137.90	512	1024	12	0	32	0.28
0.1	243.80	263.30	250.10	512	1024	12	0	32	0.28
0.2	416.30	438.00	416.00	512	1024	12	0	32	0.28
0.25	586.67	605.78	630.49	512	1024	12	0	32	0.28
0.5	1048.80	1060.50	1049.00	512	1024	12	0	32	0.28
0.75	1287.31	1333.19	1329.77	512	1024	12	0	32	0.28
1	1802.50	1826.70	1724.60	512	1024	12	0	32	0.28
2	2865.17	2897.00	3052.09	512	1024	12	0	32	0.28
	Dataset II								
0.025	60.22	63.41	60.69	512	1024	12	0	32	0.28

0.05	114.86	120.09	115.57	512	1024	12	0	32	0.28
0.1	245.34	247.27	246.04	512	1024	12	0	32	0.28
0.2	463.49	495.61	473.86	512	1024	12	0	32	0.28
0.25	628.84	651.89	630.49	512	1024	12	0	32	0.28
0.5	1148.6	1164.8	1149.5	512	1024	12	0	32	0.28
0.75	1305.8	1299.3	1306.0	512	1024	12	0	32	0.28
1	2056.0	2090.4	2057.4	512	1024	12	0	32	0.28
2	3051.0	3081.8	3052.1	512	1024	12	0	32	0.28
	Dataset III								
0.025	54.84	61.36	55.03	512	1024	12	0	32	0.28
0.05	136.40	150.95	138.50	512	1024	12	0	32	0.28
0.1	246.14	260.59	251.71	512	1024	12	0	32	0.28
0.2	418.18	462.06	418.28	512	1024	12	0	32	0.28
0.25	602.43	630.18	610.53	512	1024	12	0	32	0.28
0.5	1042.3	1086.6	1041.9	512	1024	12	0	32	0.28
0.75	1453.5	1488.0	1453.2	512	1024	12	0	32	0.28
1	1937.9	1919.7	1936.8	512	1024	12	0	32	0.28
2	2713.4	2665.7	2714.3	512	1024	12	0	32	0.28
	Dataset IV								
0.025	55.23	55.52	55.61	512	1024	12	0	32	0.28
0.05	143.02	145.90	143.30	512	1024	12	0	32	0.28
0.1	241.40	245.90	245.80	512	1024	12	0	32	0.28
0.2	440.00	445.70	443.70	512	1024	12	0	32	0.28
0.25	480.11	462.30	489.73	512	1024	12	0	32	0.28
0.5	821.60	808.90	827.90	512	1024	12	0	32	0.28
0.75	1150.17	1152.97	1155.23	512	1024	12	0	32	0.28
1	1492.71	1503.80	1532.60	512	1024	12	0	32	0.28
2	2592.52	2626.30	2503.10	512	1024	12	0	32	0.28

Table 7. Summary of coding efficiency (PSNR) of the compression algorithms

Bit Rate	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]	Coder IX [Proposed]
	Dataset I								
0.025	34.56	34.43	34.54	34.47	34.43	34.35	34.44	34.11	34.21
0.05	36.52	36.27	36.47	36.51	36.28	36.52	36.48	36.34	36.44

0.1	38.53	38.28	38.50	38.35	38.12	38.29	38.33	38.04	38.31
0.2	41.54	41.34	41.52	41.49	41.27	41.19	41.42	41.17	41.44
0.25	42.97	42.89	43.08	43.12	41.92	42.02	42.17	42.87	43.08
0.5	46.81	46.60	46.81	46.76	46.41	46.09	46.73	46.21	46.67
0.75	50.54	50.41	50.59	50.61	50.33	50.37	50.51	50.11	50.49
1	53.52	53.32	53.51	53.49	53.33	53.46	53.47	53.12	53.45
2	66.09	66.02	66.19	65.97	65.91	65.84	66.01	65.22	65.84
	Dataset II								
0.025	27.53	27.41	27.52	27.32	27.37	27.31	27.32	26.98	27.25
0.05	30.22	29.99	30.21	30.20	29.97	30.19	30.20	29.94	30.11
0.1	32.57	32.48	32.57	32.48	32.18	32.04	32.07	31.89	32.38
0.2	34.78	34.63	34.75	34.66	34.51	34.64	34.66	34.21	34.58
0.25	35.62	35.50	35.62	35.63	35.50	35.55	35.56	35.25	35.61
0.5	39.09	39.03	39.18	38.76	38.63	38.46	38.47	38.02	38.69
0.75	42.09	41.95	42.08	41.48	41.92	41.28	41.30	40.97	41.35
1	44.19	44.05	44.18	44.20	44.05	43.89	43.91	43.55	44.02
2	51.48	51.33	51.47	51.22	51.25	50.85	50.86	50.66	51.05
	Dataset III								
0.025	28.83	28.68	28.82	28.68	28.62	28.64	28.67	28.44	28.67
0.05	30.32	30.06	30.30	30.29	30.06	30.28	30.29	30.01	30.21
0.1	32.43	32.17	32.38	32.23	32.01	32.25	32.29	31.98	32.21
0.2	34.84	34.62	34.83	34.61	34.51	34.56	34.58	34.38	34.61
0.25	35.70	35.49	35.69	35.49	35.31	35.42	35.44	35.08	35.54
0.5	39.06	39.03	39.01	38.97	38.85	38.77	38.78	38.64	38.92
0.75	42.24	42.04	42.24	42.01	41.97	41.83	41.84	41.69	42.12
1	45.16	44.93	45.16	45.14	44.82	44.50	44.50	44.34	45.02
2	55.46	55.31	55.46	55.34	55.24	54.89	54.91	54.31	55.21
	Dataset IV								
0.025	29.84	29.68	29.82	29.73	29.58	29.71	29.74	29.51	29.67
0.05	32.27	31.97	32.25	32.26	31.88	32.30	32.22	31.97	32.08
0.1	35.08	35.11	35.07	35.29	35.04	35.06	35.03	34.84	34.94
0.2	39.35	39.13	39.60	39.40	39.01	39.11	39.41	39.02	39.44
0.25	41.11	41.24	42.01	41.74	41.21	41.45	41.28	40.89	41.36
0.5	45.98	46.33	46.78	46.26	46.24	45.91	46.14	45.87	46.09
0.75	50.94	51.01	51.34	51.06	51.31	51.41	51.39	50.76	50.87
1	54.77	54.72	55.13	54.86	54.62	54.78	54.92	54.55	54.65
2	61.02	60.85	61.04	60.96	60.84	60.90	60.89	60.77	60.84

Table 8. Summary of Bjøntegaard Delta PSNR gain for compression algorithms

HS Image	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]
Dataset I	- 0.1185	0.0581	- 0.1293	- 0.0836	0.2519	0.2226	0.0693	0.2897
Dataset II	- 0.2547	- 0.1202	- 0.2551	- 0.0954	- 0.0016	0.0931	0.0769	0.3933
Dataset III	- 0.1722	0.0257	- 0.1532	- 0.0248	0.1333	0.1221	0.1033	0.3743
Dataset IV	- 0.0366	0.0026	- 0.3328	- 0.1888	0.0455	-0.064	- 0.103	0.209

Table 9. Summary of Structural Similarity (SSIM) index for compression algorithms

Bit Rate	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]	Coder IX [Proposed]
	Dataset I								
0.025	0.385	0.386	0.384	0.385	0.386	0.386	0.384	0.377	0.377
0.050	0.480	0.479	0.480	0.480	0.479	0.480	0.480	0.471	0.472
0.100	0.587	0.587	0.585	0.584	0.587	0.589	0.590	0.581	0.583
0.200	0.677	0.675	0.678	0.677	0.678	0.677	0.677	0.664	0.667
0.250	0.701	0.702	0.702	0.699	0.703	0.702	0.702	0.697	0.699
0.500	0.790	0.788	0.787	0.789	0.789	0.788	0.787	0.774	0.775
0.750	0.847	0.846	0.849	0.846	0.846	0.846	0.846	0.845	0.848
1.000	0.886	0.886	0.886	0.888	0.887	0.886	0.886	0.884	0.887
2.000	0.914	0.912	0.915	0.914	0.912	0.915	0.915	0.911	0.91
	Dataset IV								
0.025	0.297	0.288	0.299	0.299	0.285	0.300	0.301	0.299	0.301
0.05	0.346	0.338	0.345	0.346	0.341	0.349	0.349	0.248	0.344
0.1	0.437	0.431	0.437	0.436	0.430	0.437	0.437	0.437	0.439
0.2	0.518	0.513	0.518	0.518	0.514	0.518	0.518	0.517	0.52
0.25	0.545	0.543	0.545	0.552	0.545	0.553	0.553	0.550	0.552
0.5	0.603	0.601	0.603	0.608	0.606	0.611	0.611	0.604	0.606
0.75	0.630	0.629	0.630	0.636	0.631	0.636	0.636	0.633	0.63
1	0.645	0.645	0.645	0.648	0.646	0.647	0.647	0.644	0.646
2	0.666	0.666	0.666	0.666	0.666	0.668	0.668	0.663	0.665

Table 10. Summary of Feature-Similarity (FSIM) Index for compression algorithms

Bit Rate	Coder I [79]	Coder II [78]	Coder III [82]	Coder IV [80]	Coder V [81]	Coder VI [83]	Coder VII [84]	Coder VIII [85]	Coder IX [Proposed]
	Dataset I								
0.025	0.626	0.626	0.660	0.624	0.626	0.616	0.616	0.614	0.611
0.05	0.641	0.642	0.712	0.639	0.643	0.637	0.638	0.633	0.637
0.1	0.710	0.695	0.787	0.712	0.712	0.716	0.716	0.711	0.712

0.2	0.774	0.759	0.830	0.780	0.779	0.784	0.784	0.777	0.774
0.25	0.784	0.776	0.843	0.783	0.790	0.795	0.795	0.791	0.79
0.5	0.851	0.844	0.909	0.855	0.856	0.863	0.863	0.849	0.843
0.75	0.891	0.888	0.947	0.910	0.889	0.903	0.903	0.896	0.899
1	0.919	0.917	0.970	0.920	0.918	0.918	0.918	0.912	0.917
2	0.978	0.978	0.996	0.980	0.980	0.981	0.981	0.978	0.974
	Dataset IV								
0.025	0.286	0.286	0.282	0.290	0.288	0.295	0.303	0.288	0.289
0.05	0.293	0.288	0.289	0.294	0.286	0.302	0.302	0.294	0.293
0.1	0.321	0.319	0.320	0.321	0.320	0.338	0.338	0.337	0.339
0.2	0.443	0.434	0.443	0.452	0.439	0.488	0.488	0.482	0.485
0.25	0.518	0.522	0.518	0.530	0.529	0.537	0.537	0.531	0.529
0.5	0.694	0.692	0.694	0.699	0.696	0.750	0.750	0.745	0.746
0.75	0.807	0.812	0.808	0.817	0.813	0.835	0.834	0.827	0.829
1	0.871	0.867	0.872	0.869	0.867	0.873	0.873	0.866	0.869
2	0.979	0.978	0.979	0.979	0.978	0.979	0.979	0.965	0.967

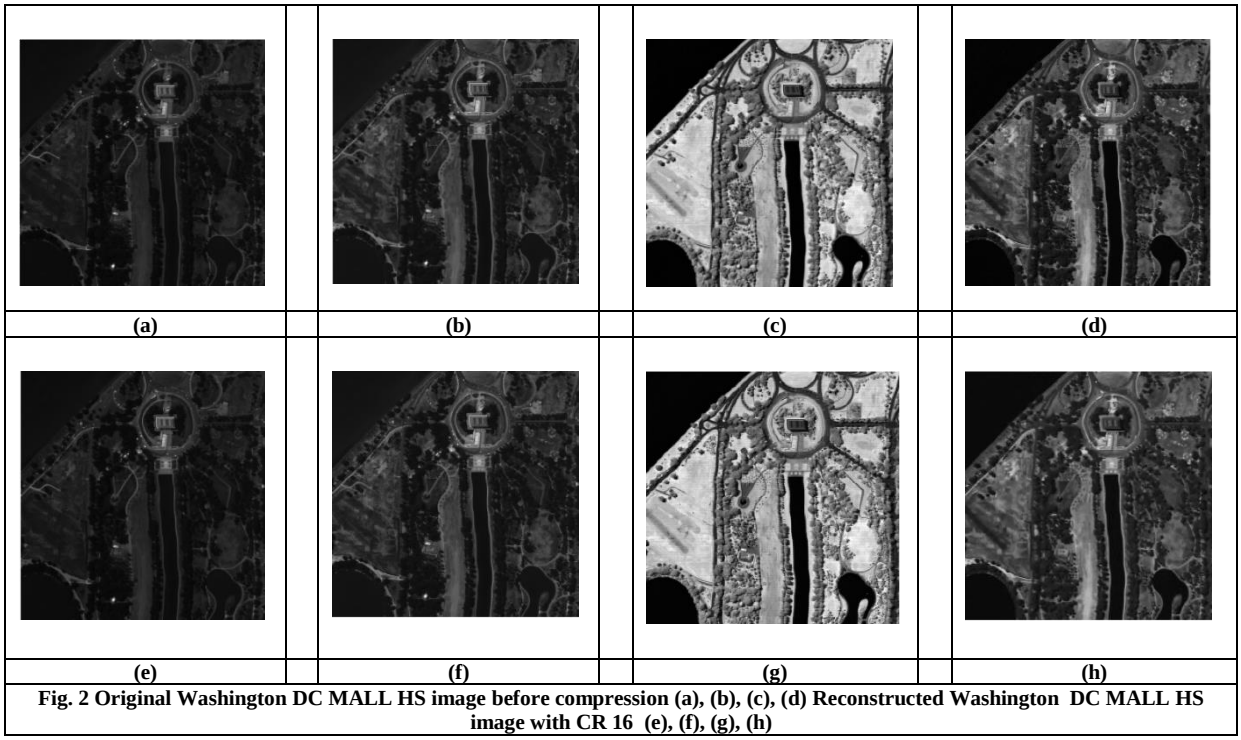

Fig. 2 Original Washington DC MALL HS image before compression (a), (b), (c), (d) Reconstructed Washington DC MALL HS image with CR 16 (e), (f), (g), (h)

5. Conclusion

In the present study, a low-complexity version of the 3D-CE-ZM-SPECK is proposed. It stores the maximum numeric values of each partitioned sub-band in the buffer coding memory, which has a very small size. Through this, the complexity of the 3D-ZM-SPECK is reduced significantly. The coding memory of the proposed compression algorithm is independent of the size of the HS image, the HS image pixel depth, and the bit rates. It depends only on the level of the wavelet transform. Although the present compression algorithm is focused on the complexity of the algorithm, it can be extended to color images, video, and multi-temporal hyperspectral images.

References

[1] Manoj K. Mishra et al., "Retrieval of Atmospheric Parameters and Data-processing Algorithms for AVIRIS-NG Indian Campaign Data," *Current Science*, vol. 116, no. 7, pp. 1089-1100, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] J. Zabalza et al., "Hyperspectral Imaging based Characterization and Identification of Sintered UO2 Fuel Pellets," *2023 IEEE Nuclear Science Symposium, Medical Imaging Conference and International Symposium on Room-Temperature Semiconductor Detectors (NSS MIC RTSD)*, Vancouver, BC, Canada, pp. 1-1, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Chiara Cevoli et al., "Storage of Wafer Cookies: Assessment by Destructive Techniques, and Non-destructive Spectral Detection Method," *Journal of Food Engineering*, vol. 336, pp. 1-33, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Saurabh Kumar et al., "Onboard Hyperspectral Image Compression Using Compressed Sensing and Deep Learning," *Proceedings of the 2018 European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 30-42, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shaohui Mei et al., "Hyperspectral Image Spatial Super-Resolution via 3D Full Convolutional Neural Network," *Remote Sensing*, vol. 9, no. 11, pp. 1-22, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Assiya Sarinova et al., "Development of Compression Algorithms for Hyperspectral Aerospace Images Based on Discrete Orthogonal Transformations," *Eastern-European Journal of Enterprise Technologies*, vol. 1, no. 2, pp. 22-30, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Prabira Kumar Sethy et al., "Hyperspectral Imagery Applications for Precision Agriculture - A Systemic Survey," *Multimedia Tools and Applications*, vol. 81, pp. 3005-3038, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Sudhanshu Shekhar Jha et al., "Target Detection in Hyperspectral Imagery Using Atmospheric-Spectral Modeling and Deep Learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Liu Lixin et al., "Recent Advances of Hyperspectral Imaging Application in Biomedicine," *Chinese Journal of Lasers*, vol. 45, no. 2, pp. 1-10, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Dong An et al., "Advances in Infrared Spectroscopy and Hyperspectral Imaging Combined with Artificial Intelligence for the Detection of Cereals Quality," *Critical Reviews in Food Science and Nutrition*, vol. 63, no. 29, pp. 9766-9796, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] 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]

[12] Pengfei Ma, Liang Fan, and Genda Chen, "Hyperspectral Reflectance for Determination of Steel Rebar Corrosion and Cl− Concentration," *Construction and Building Materials*, vol. 368, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Arvind Mukundan et al., "Automatic Counterfeit Currency Detection Using a Novel Snapshot Hyperspectral Imaging Algorithm," *Sensors*, vol. 23, no. 4, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Subhadyouti Bose, Mili Ghosh Nee Lala, and Akhouri Pramod Krishna, "Photometric Correction of Images of Visible and Near-Infrared Bands from Chandrayaan-1 Hyper-Spectral Imager (HySI)," *Earth, Moon, and Planets*, vol. 126, pp. 1-33, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Dhritiman Saha, and Annamalai Manickavasagan, "Machine Learning Techniques for Analysis of Hyperspectral Images to Determine Quality of Food Products: A Review," *Current Research in Food Science*, vol. 4, pp. 28-44, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Antonio Plaza et al., "Recent Advances in Techniques for Hyperspectral Image Processing," *Remote Sensing of Environment*, vol. 113, pp. S110-S122, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kristiane de Cássia Mariotti, Rafael Scorsatto Ortiz, and Marco Flôres Ferrão, "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]

[18] Maitreya Mohan Sahoo et al., "Modelling Spectral Unmixing of Geological Mixtures: An Experimental Study Using Rock Samples," *Remote Sensing*, vol. 15, no. 13, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] 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, pp. 5580-5602, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ihab Makki et al., "A Survey of Landmine Detection using Hyperspectral Imaging," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 124, pp. 40-53, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Suvita Rani Sharma, Birmohan Singh, and Manpreet Kaur, "A Hybrid Encryption Model for the Hyperspectral Images: Application to Hyperspectral Medical Images," *Multimedia Tools and Applications*, vol. 83, pp. 11717-11743, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Alejandro Ehrenfeld et al., "HIDSAG: Hyperspectral Image Database for Supervised Analysis in Geometallurgy," *Scientific Data*, vol. 10, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Michal Shimoni, Rob Haelterman, and Christiaan Perneel, "Hyperspectral 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]

[24] Xuesan Su et al., "A Review of Pharmaceutical Robot based on Hyperspectral Technology," *Journal of Intelligent & Robotic Systems*, vol. 105, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Owen Tamin et al., "A Review of Hyperspectral Imaging-based Plastic Waste Detection State-of-the-art," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 3, pp. 3407-3419, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Chao Xia et al., "Maize Seed Classification using Hyperspectral Image Coupled with Multi-linear Discriminant Analysis," *Infrared Physics & Technology*, vol. 103, pp. 1-8, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Matthew S. Mills et al., "Assessment of the Utility of Underwater Hyperspectral Imaging for Surveying and Monitoring Coral Reef Ecosystems," *Scientific Reports*, vol. 13, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[28] William J. Blackwell et al., "Improved All-weather Atmospheric Sounding using Hyperspectral Microwave Observations," *2010 IEEE International Geoscience and Remote Sensing Symposium*, Honolulu, HI, USA, pp. 734-737, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[29] Jaime Zabalza et al., "Hyperspectral Imaging Based Corrosion Detection in Nuclear Packages," *IEEE Sensors Journal*, vol. 23, no. 21, pp. 25607-25617, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] Joseph B. Boettcher, Qian Du, and James E. Fowler, "Hyperspectral Image Compression with the 3D Dual-tree Wavelet Transform," *2007 IEEE International Geoscience and Remote Sensing Symposium*, Barcelona, Spain, pp. 1033-1036, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[31] M. Sam Navin, and L. Agilandeeswari, "Multispectral and Hyperspectral Images based Land Use / land Cover Change Prediction Analysis: An Extensive Review," *Multimedia Tools and Applications*, vol. 79, pp. 29751-29774, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] R. Nagendran, and A. Vasuki, "Hyperspectral Image Compression using Hybrid Transform with Different Wavelet-based Transform Codinguki," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 18, no. 1, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[33] Aloke 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]

[34] Qiang Zhang et al., "Hyperspectral Image Denoising: From Model-Driven, Data-Driven, to Model-Data-Driven," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13143-13163, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[35] Madhumitha Ramamurthy et al., "RETRACTED: Auto Encoder based Dimensionality Reduction and Classification using Convolutional Neural Networks for Hyperspectral Images," *Microprocessors and Microsystems*, vol. 79, pp. 1-10, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] Purushottam Lal Nagar, and Shrish Bajpai, "Lifting-Based Block Fractional Wavelet Filter Compression of Hyperspectral Images over Wireless Multimedia Sensor Network Platforms," *ITEGAM-JETIA*, vol. 12, no. 58, pp. 284-295, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[37] Mahesh Kumar Tripathi, and H. Govil, "Evaluation of AVIRIS-NG Hyperspectral Images for Mineral Identification and Mapping," *Heliyon*, vol. 5, no. 11, no. 1-10, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[38] Jing Zhang et al., "Multi-Scale Feature Mapping Network for Hyperspectral Image Super-Resolution," *Remote Sensing*, vol. 13, no. 20, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[39] Gemine Vivone et al., "Multispectral and Hyperspectral Image Fusion in Remote Sensing: A Survey," *Information Fusion*, vol. 89, pp. 405-417, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Honghui Xu et al., "Nonlocal B-spline Representation of Tensor Decomposition for Hyperspectral Image Inpainting," *Signal Processing*, vol. 206, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[41] 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]

[42] Pangambam Sendash Singh, and Subbiah Karthikeyan, "Salient Object Detection in Hyperspectral Images using Deep Background Reconstruction based Anomaly Detection," *Remote Sensing Letters*, vol. 13, no, 2, pp. 184-195, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[43] 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]

[44] 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]

[45] Divya Sharma, Yogendra Kumar Prajapati, and Rajeev Tripathi, "Spectrally Efficient 1.55 Tb/s Nyquist-WDM Superchannel with Mixed Line Rate Approach using 27.75 Gbaud PM-QPSK and PM-16QAM," *Optical Engineering*, vol. 57, no. 7, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[46] Ying Hou, and Guizhong Liu, "Hyperspectral Image Lossy-to-lossless Compression using the 3D Embedded Zeroblock Coding Algorithm," *2008 International Workshop on Earth Observation and Remote Sensing Applications*, Beijing, China, pp. 1-6, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[47] 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]

[48] 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]

[49] 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]

[50] Palash Uddin, Al Mamun, and Ali Hossain, "PCA-based Feature Reduction for Hyperspectral Remote Sensing Image Classification," *IETE Technical Review*, vol. 38, no. 4, pp. 377-396, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[51] Peter Schelkens, "*Multi-dimensional Wavelet Coding-algorithms and Implementations*," Ph.D Dissertation, Vrije Universiteit Brussel, 2001. [Google Scholar] [Publisher Link]

[52] 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]

[53] Shrish Bajpai, Harsh Vikram Singh, and Naimur Rahman Kidwai "3D Modified Wavelet Block Tree Coding for Hyperspectral Images," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 2, pp. 1001-1008, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[54] Yonghui Wang, J.T. Rucker, and J.E. Fowler, "Three-dimensional Tarp Coding for the Compression of Hyperspectral Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 2, pp. 136-140, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[55] Wei Pan, Yi Zou, and Lu Ao, "A Compression Algorithm of Hyperspectral Remote Sensing Image based on 3-D Wavelet Transform and Fractal," *2008 3rd International Conference on Intelligent System and Knowledge Engineering*, Xiamen, China, pp. 1237-1241, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[56] R. Suresh Kumar, and P. Manimegalai, "Near Lossless Image Compression using Parallel Fractal Texture Identification," *Biomedical Signal Processing and Control*, vol. 58, pp. 1-9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[57] Shaheer Mohamed et al., "FactoFormer: Factorized Hyperspectral Transformers with Self-Supervised Pretraining," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[58] Jiaji Wu, Zhensen Wu, and Chengke Wu, "Lossy to Lossless Compressions of Hyperspectral Images using Three-Dimensional Set Partitioning Algorithms," *Optical Engineering*, vol. 45, no. 2, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[59] 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]

[60] Yongjian Nian, Mi He, and Jianwei Wan, "Low-complexity Compression Algorithm for Hyperspectral Images based on Distributed Source Coding," *Mathematical Problems in Engineering*, vol. 2013, pp. 1-7, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[61] Kai-jen Cheng, and Jeffrey Dill, "Lossless to Lossy Dual-tree BEZW Compression for Hyperspectral Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 9, pp. 5765-5770, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[62] Yaman Dua, Vinod Kumar, and Ravi Shankar Singh, "Parallel Lossless HSI Compression based on RLS Filter," *Journal of Parallel and Distributed Computing*, vol. 150, pp. 60-68, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[63] 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]

[64] Zhuocheng Jiang, W. David Pan, and Hongda Shen, "Spatially and Spectrally Concatenated Neural Networks for Efficient Lossless Compression of Hyperspectral Imagery," *Journal of Imaging*, vol. 6, no. 6, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[65] Yaman Dua, Vinod Kumar, and Ravi Shankar Singh, "Comprehensive Review of Hyperspectral Image Compression Algorithms," *Optical Engineering*, vol. 59, no. 9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[66] Xianghai Wang et al., "Distributed Source Coding of Hyperspectral Images based on Three-dimensional Wavelet," *Journal of the Indian Society of Remote Sensing*, vol. 46, pp. 667-673, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[67] K. S. Gunasheela, and H. S. Prasantha, "Compressive Sensing Approach to Satellite Hyperspectral Image Compression," *Conference Proceedings Information and Communication Technology for Intelligent System*, pp. 495-303, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[68] 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]

[69] Qian Du, and James E. Fowler, "Hyperspectral Image Compression using JPEG2000 and Principal Component Analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201-205, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[70] Ying Hou, and Guizhong Liu, "3D Set Partitioned Embedded Zero Block Coding Algorithm for Hyperspectral Image Compression," *Proceedings Remote Sensing and GIS Data Processing and Applications; and Innovative Multispectral Technology and Applications*, Wuhan, China, vol. 6790, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[71] Ying Hou, and Guizhong Liu, "Lossy-to-Lossless Compression of Hyperspectral Image Using the Improved AT-3D SPIHT Algorithm," *2008 International Conference on Computer Science and Software Engineering*, Wuhan, China, pp. 963-966, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[72] Xiao Jiang et al., "Compression of the Multispectral Image by the Three-dimensional EBCOT Coding Algorithm," *Journal of Xidian University*, vol. 32, no. 4, pp. 549-554, 2005. [Google Scholar] [Publisher Link]

[73] Shrish Bajpai, "Low Complexity and Low Memory Compression Algorithm for Hyperspectral Image Sensors," *Wireless Personal Communications*, vol. 131, pp. 805-833, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[74] 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, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[75] 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]

[76] Shrish Bajpai, "3D-Listless Block Cube Set-partitioning Coding for Resource Constraint Hyperspectral Image Sensors," *Signal, Image and Video Processing*, vol. 18, pp. 3163-3178, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[77] R. Nagendran et al., "Lossless Hyperspectral Image Compression by Combining the Spectral Decorrelation Techniques with Transform Coding Methods," *International Journal of Remote Sensing*, vol. 45, no. 18, pp. 6226-6248, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[78] Xiaoli Tang, and W.A. Pearlman, "Lossy-to-lossless Block-based Compression of Hyperspectral Volumetric Data," *2004 International Conference on Image Processing, 2004. ICIP '04*, Singapore, vol. 5, pp. 3283-3286, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[79] 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]

[80] Ruzelita Ngadiran et al., "Efficient Implementation of 3D Listless SPECK," *International Conference on Computer and Communication Engineering (ICCCE'10)*, Kuala Lumpur, Malaysia, pp. 1-4, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[81] VK. Sudha, and R. Sudhakar, "3D Listless Embedded Block Coding Algorithm for Compression of Volumetric Medical Images," *Journal of Scientific & Industrial Research*, vol. 72, pp. 735-748, 2013. [Google Scholar]

[82] 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]

[83] Shrish Bajpai et al., "Low Memory Block Tree Coding for Hyperspectral Image," *Multimedia Tools and Applications*, vol. 78, pp. 27193-27209, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[84] Shrish Bajpai et al., "A Low Complexity Hyperspectral Image Compression through 3D Set Partitioned Embedded Zero Block Coding," *Multimedia Tools and Applications*, vol. 81, pp. 841-872, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[85] Shrish Bajpai, "Low Complexity Block Tree Coding for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 81, pp. 33205-33323, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[86] Harshit Chandra, and Shrish Bajpai, "Listless Block Cube Tree Coding for Low Resource Hyperspectral Image Compression Sensors," *2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, Aligarh, India, pp. 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[87] Harshit Chandra, and Shrish Bajpai, "3D-Block Partitioning Embedded Coding for Hyperspectral Image Sensors," *2023 International Conference on Power, Instrumentation, Energy and Control (PIECON)*, Aligarh, India, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[88] Shrish Bajpai, "Low Complexity Image Coding Technique for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 82, pp. 31233-31258, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[89] Harshit Chandra et al., "3D-Memory Efficient Listless Set Partitioning in Hierarchical Trees for Hyperspectral Image Sensors," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 6, pp. 11163-11187, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[90] Vinod Kumar Tripathi, and Shrish Bajpai, "Curvelet Transform Based Hyperspectral Image Compression with Listless Set Partitioned Compression Algorithm for Unmanned Aerial Vehicle Image Sensor," *SSRG International Journal of Electronics and Communication Engineering*, vol. 11, no. 12, pp. 71-82, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[91] Rajesh, Shrish Bajpai, and Naimur Rahman Kidwai, "Block-Based Fractional Wavelet Filter for Compression of Hyperspectral Images Over Wireless Multimedia Sensor Network Platforms," *SSRG International Journal of Electronics and Communication Engineering*, vol. 12, no. 3, pp. 21-42, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[92] Divya Sharma, "Image Quality Assessment Metrics for Hyperspectral Image Compression Algorithms," *2024 Second International Conference Computational and Characterization Techniques in Engineering & Sciences (IC3TES)*, Lucknow, India, pp. 1-5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[93] De Rosal Igantius Moses Setiadi, "PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography," *Multimedia Tools and Applications*, vol. 80, pp. 8423-8444, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[94] R. Nagendran et al., "Neural Reinforcement-oriented Hyperspectral Image Compression: Adaptive Approaches for Enhanced Quality," *Chemometrics and Intelligent Laboratory Systems*, vol. 266, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[95] Anna de Juan, and Rodrigo Rocha de Oliveira, "Hyperspectral Image and Chemometrics. A Step Beyond Classical Spectroscopic PAT Tools," *Analytical and Bioanalytical Chemistry*, vol. 418, pp. 23-34, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[96] Rajeev Kumar Sachan et al., "Assessment of Non-Uniform Channel-based Dual Metal Negative Capacitance Ge-Pocket Tunnel Field-Effect Transistor with Parametric Optimization for Low Power and High Frequency Applications," *Technical Physics Letters*, vol. 51, pp. 334-348, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[97] Nadia Zikiou, Mourad Lahdir, and David Helbert, "Support Vector Regression-based 3D-wavelet Texture Learning for Hyperspectral Image Compression," *The Visual Computer*, vol. 36, pp. 1473-1490, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[98] Fahad Saeed, Shumin Liu, and Jie Chen, "SpecResNet: Hyperspectral Image Compression via Hybrid Residual Learning and Spectral Calibration," *Remote Sensing*, vol. 18, no. 7, pp. 1-17, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[99] Guisong Wang et al., "CUINR: Combining Unmixing with Implicit Neural Representation for Enhanced Hyperspectral Image Compression," *The Visual Computer*, vol. 42, 2026. [CrossRef] [Google Scholar] [Publisher Link]