SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I3P117

Volume 13 Issue 3, 230-247, March 2026

Original Article

3D-One List Set Partitioning in Hierarchical Trees 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]*Department of Computer Science and Engineering, Maharishi University of Information Technology, Lucknow, Uttar Pradesh, India.*

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

Received: 12 December 2025 Revised: 15 January 2026 Accepted: 24 February 2026 Published: 31 March 2026

Abstract *- Wavelet transform-based set-partitioned hyperspectral image compression algorithms have comparatively better coding efficiency, moderate coding complexity, embeddedness property, and scalability property. These compression algorithms use linked lists, arrays, or state tables to track the significance/insignificance of the partitioned sets/coefficients. The 3D-Set Partitioning in Hierarchical Trees (3D-SPIHT) employs three linked lists for the tracking of the significant/insignificant coefficients or zerotree nodes. These linked lists create problems associated with the coding complexity and exponential growth of coding memory. There are many listless compression algorithms for hyperspectral images that have been proposed in the past, but they suffer from low coding efficiency. The proposed compression algorithm 3D-One List- SPIHT (3D-OL-SPIHT) uses the same partition rule as 3D-SPIHT, but it has a single list for tracking of coefficients or zerotree nodes instead of three lists in 3D-SPIHT. From the experimental results, it is clear that the proposed compression algorithm reduces the coding complexity, coding memory demand, and increases the coding efficiency compared to 3D-SPIHT.*

Keywords *- Coding, Zerotree, Set partitioned hyperspectral image compression algorithm, Wavelet transform, Aerial image.*

1. Introduction

Onboard HyperSpectral (HS) image sensors collect the spectral and spatial information from 400 nm to 2500 nm in the 5-10 nm interval of the electromagnetic spectrum, having hundreds of grayscale images at different wavelengths [1, 2]. Owing with the valuable information, HS images are use in countless applications over last couple of decades such as astronomy [3], Biomedical Application [4], Biological Diagnosis [5], Cultivation (agriculture) [6], Civil (Terrain Detection, Land Cover Classification, Urban Planning) [7, 8], Corrosion Detection [9], Counterfeit Detection [10], Defence & Security [11], Environmental Research And Monitoring (oil spill monitoring etc) [12], Food Safety [13], Forestry [14], Forensics Research [15], Geography [16], Geology [17], Medical Surgery [18], Mineral Exploration (Mineralogy) [19], Pharma [20], Underwater Hyperspectral Imaging [21], Weather Prediction [22], etc. Beside the mention fields, remote sensing [23] is one of the fastest growing fields of research which uses hyperspectral images extensively for the different tasks (sub-applications) such as Change Detection [24], Compression [25], Classification [26], Denoising [27], Dimensionality Reduction [28], Feature Extraction [29], Feature Identification [30], Feature Mapping [31], Fusion [32], Inpainting [33], Object (Target) Detection [34], Object Identification [35], Unmixing [36] and Segmentation [37].

The HS images acquired by the remote sensing imaging system are often of a large size (~ 150 MB for a single HS image), thus requiring a lot of memory to store a single HS image [38]. Therefore, HS image compression is an essential step before the HS image is communicated through a communication channel. Through the compression of the HS image, the sensor performance has been improved, and the requirement of the communication channel bandwidth has also been reduced [39].

Each HS image has two different types of redundancies present, named as spectral redundancy and spatial redundancy [40]. The spatial redundancy exists due to the pixel statistical dependencies on the nearby pixels present in the same frame of the HS image, while spectral redundancy exists due to the single pixel present at the same spatial coordinate at nearby continuous frequency frames [29]. The ratio of original (uncompressed) HS image to reconstructed (compressed) HS image is known as the Compression Ratio (CR), which is a non-unit (Ratio) parameter [41] and numerically represented as in Equation 1.

$$Compression\ Ratio\ (CR) = \frac{Size\ of\ uncompress\ \ HS\ image}{Size\ of\ compressed\ HS\ image} \quad (1)$$

HyperSpectral Image Compression Algorithms (HSICAs) are divided into the sub-varieties on the basis of the two parameters, namely, the process of coding or HS image data loss. On the basis of data loss, the HSICAs can again be categorized by lossless HSICA, lossy HSICA, and near lossless HSICA. As the name suggests, for lossless HSICA, there is no loss of any image data, and it has a low value of CR, while for the near lossless HSICA, there is relatively (slightly) high CR, but it has a loss of some HS image data. For the lossy HSICA, as the name suggests, there is a loss of HS image data, but the HSICA has eminent CR. Lossy HSICA loses the HS image data, which is not significant for any human visualization. HSICAs are bisected into the nine distinct types based on the way the encoding/decoding process has been performed to achieve the compression. They are Transform Coding (TC) [42], Predicative Coding (PC) [43], Vector Quantization (VQ) [44], Compressive Sensing (CS) [45], Sparse Representation (SR) [46], Tensor Decomposition (TD) [47], Neural Network (NN)-based HS Image Compression [48], Machine Learning (ML)-based [49] and Hybrid Compression Algorithm [50]. For coding complexity, PC based HSICA is best, while for coding efficiency, NN-based HSICA and ML-based HSICA are best.

Among the above-mentioned types of HSICA, TC-based HSICAs can work for any type of compression (lossy or lossless) and have the best error tolerance mechanism [51, 52]. TC-based HSICA uses mathematical transforms such as cosine transform, Discrete Wavelet Transform (DWT), Dyadic Wavelet Transform (Dy WT), Curvelet Transform (CT), Shearlet Transform (ST) etc to translate the HS image into the frequency domain, and then HSICA is applied on the transformed image to obtain compression [53]. The choice of using a type of HSICA for any application depends upon the various factors such as memory usage, ease of deployment and update, inference on resource-constrained devices, power dissipation, and prediction latency. Besides these, for ML-based HSICAs, ease of training/finetuning and ease of distributed training are also important factors for choosing the HSICAs [54]. The main objective of the current study in this manuscript can be summarized as follows:

- To reduce the complexity, two different thresholds are used in the proposed compression algorithm. The threshold in the top LLL_L sub-band of the transform HS image is four to sixteen times the threshold of the rest of the transform HS image.
- It avoids the revalidation of the significant root (significant in the previous bit plane).

The structure of the present manuscript is organized as follows: In Section 2, the problem of HS image compression is revisited, and mathematical transform-based HS image compression algorithms are introduced. Section 3 explicitly explains the proposed 3D One List SPIHT (3D-OL-SPIHT).

Section 4 covers the experimental results with a detailed description of comparative results on the distinct performance parameters. Section 5 concludes the manuscript with key findings and remarks.

2. Preliminary

The TC-based HSICAs use a mathematical transform to convert the HS image into the frequency domain and pack the energy into a few coefficients (high-energy concentrated). The mathematical transform works as a spectral decorrelator and exploits both correlation (spectral and spatial). Among all the mathematical transforms, the wavelet transform provides better performance than other transforms. The based HSICAs process is divided into two sections: the Transform stage and Coding stage [52]. Mathematical Transform-based set partitioned HSICAs are a special type of compression algorithm that uses the set structures to define a large number of insignificant coefficients in the top bit plane through a single bit at a higher bit plane. Wavelet transform-based set partitioned HSICA has been divided into four different categories (basis of lists) based on the use of lists as List-based HSICA, Listless HSICA, Array-based HSICA, and list & marker-based HSICA. While on the basis of the set partition rule, it can be defined in three different ways, such as zero block, zerotree, or zero block tree [53].

3. 3D One List SPIHT (3D-OL-SPIHT)

The 3D-OL-SPIHT is a modified form of 3D-SPIHT [55] in which coding complexity is reduced, and there is an increase in coding efficiency by doing two small modifications. The first modification is related to the use of two different thresholds, while the second modification is related to the significance testing of the tree roots for each bit plane. It has been observed that the maximum threshold in the LLLL sub-band is four to sixteen times the maximum value of the other sub-bands of the transform HS image. The LLL sub-band represents the average value of all transform coefficients. Thus, the bit difference between the maximum value of the coefficient present in the LLLL sub-band and the rest of the transform HS image is 2 bits to 4 bits.

As in the other state-of-the-art HSICAs such as 3D-SPIHT [55], 3D-NLS [62], 3D-MELS, 3D-SLS [75], 3D-LEZSPC [72], a single threshold throughout the encoding/decoding process, the 3D-OL-SPIHT uses two different thresholds to save the coding memory, reduce coding complexity, and processing power. The size of the output bit stream also reduces for lossless compression, while for lossy compression, the compression ratio and coding efficiency are increased. When the root is present between the orientation level of R0 to R2 becomes significant, its associated offspring O(i,j,k) is added to the 'LRS' list. This significant root is tested for every bit plane in 3D-SLS [75], but for 3D-OL-SPIHT, it will not be tested because it is already present in the list; the significance testing of the offspring of the roots present at the orientation level RL-1 is tested. Through this, there is a reduction in the coding complexity. 3D-OL-SPIHT has two passes called the Initialization pass and the Coding Pass. The coding pass has three more sub passes called the sorting pass, quantization pass, and refinement pass. In the initialization pass, two thresholds had been initialized. After that, proposed HSICA performed the coding process for the LLLL sub-band in the mini coding Pass mentioned in the pseudo code in Table 1. The rest of the algorithm follows the same rule of 3D-SLS.

The decoding process of 3D-OL-SPIHT is symmetric and follows the same operation as the encoding process, using input instead of output. The decoder does not require performing the significance testing for the coefficient or SOTs. Thus, the decoding process is less complex and requires less run time than the encoder. The decoder of 3D-OL-SPIHT has to perform mid-tread dequantization for those coefficients that are not fully encoded. The detailed code of the proposed algorithm 3D-OL-SPIHT is covered in Table 1.

Table 1. Details of the proposed HSICA

Input	HS image cube of ‘3’ dimension ‘M.’
Preprocessing	The ‘L’ wavelet transform is applied to the input image of dimension ‘M.’
	The transform HS image is converted to the 1D array ‘ϒ’ from the transform 3D HS image through linear indexing.
	The top-most LLL sub-ban is defined as $LLL_{L.}$
	The current threshold of any bit plane is defined as ‘T.’
Initialization Pass	Top bit plane $n = \log_2\lfloor \max(\Upsilon) \rfloor$
	LLL sub-band of the HS image Y_{LLL}
	HS image other than LLL sub band $Y_{other} = [(\Upsilon) - (\Upsilon_{LLL})]$
	Threshold I : $T_1 = 2^n$
	Threshold II : $T_2 = [\log_2\{\lvert\Upsilon_{other}\rvert\}]$
	$T = T_1$
	Add all the coefficients in the LLL sub-band to the LRS.
Coding Pass (mini)	while ($T \geq T_2$)
	{
	for each coefficient (i,j,k) ∈ LLL_L do
	{
	if {σ(i,j,k) = (0 ‖ 1)} do
	{
	Code_Pix (i,j,k)
	}
	}
	for each coefficient (i,j,k) ∈ LLL_L do
	{
	if {σ(i,j,k) = 2} do
	{
	Ref_Pix(i,j,k)
	}
	}
	T = (T/2)
	}
Coding Pass	Set : $T = T_1$
	{
	for each coefficient (i,j,k) ∈ LLL_L^P do
	{
	for each coefficient (i,j,k) ∈ LLL_L do
	{
	if {σ(i,j,k) = (0 ‖ 1)} do
	{
	Code_Pix (i,j,k)
.	}
	}
	for each coefficient (i,j,k) ∈ LLL_L do
	{

```
            if {σ(i,j,k) = 2} do
            {
                Ref_Pix(i,j,k)
            }
        }
    }
    for each coefficient (i,j,k)  ∈  LRS do
    {
        if {σ(i,j,k) = (0 || 1)} do
        {
            Code_Pix (i,j,k)
            if {δ((i,j,k) = 1 && δ((i,j,k) ∈ R_{M-1}  do
            {
                Compute O(i,j,k)
                {
                    if { O(σ(i,j,k)) = (0 || 1} do
                    {
                        Code_Pix (i,j,k)
                    }
                }
            }
        }
    }
    for each coefficient (i,j,k)  ∈  LRS do
    {
        if {σ(i,j,k) = 2} do
        {
            Ref_Pix(i,j,k)
        }
        if {δ((i,j,k) = 1 && δ (i,j,k)) ∈ R_{M-1}  do
        {
            Compute O(i,j,k)
            {
                if {O(i,j,k) = 2} do
                {
                    Ref_Pix(i,j,k)
                }
            }
        }
    }
    for each coefficient (i,j,k)  ∈  LRS do
    {
        if {δ(i,j,k) = 0} do
        {
            if {SOT(i,j,k) is SG} do
            {
                Transmit '1' to the bit stream.
                δ(i,j,k) = 1
                Add O(i,j,k) to LRS
                for each coefficient O(i,j,k)  ∈  LRS do
                {
                    Code_Pix (i,j,k)
                }
                else
                {
                    Transmit '0' to the bit stream.
```

```
                                }
                            }
                        }
                    }
                    T = (T/2)
                }

Function 1      Code_Pix (i,j,k)
                {
                    if {σ(i,j,k) = 0} do
                    {
                        if {|C(i,j,k)|  ≥  T} do
                        {
                            Transmit '1' to the bit stream.
                            σ(i,j,k) = 1
                            if {C(i,j,k)  ≥  0} do
                            {
                                C(i,j,k)  = {C(i,j,k)  - T}
                                elseif {C(i,j,k)  <  0} do
                                {
                                    C(i,j,k)  = {C(i,j,k)  + T}
                                }
                            }
                            elseif {|C(i,j,k)|  <  0} do
                            {
                                Transmit '0' to the bit stream.
                                elseif {σ(i,j,k) = 1} do
                                {
                                    σ(i,j,k) = 2
                                }
                            }
                        }
                    }
                }

Function 2      Ref_Pix(i,j,k)
                {
                    if {|C(i,j,k)|  ≥  T} do
                    {
                        Transmit '1' to the bit stream.
                        if {C(i,j,k)  ≥  0} do
                            {
                                C(i,j,k)  = {C(i,j,k)  - T}
                                elseif {C(i,j,k)  <  0} do
                                {
                                    C(i,j,k)  = {C(i,j,k)  + T}
                                    elseif {|C(i,j,k)|  <  T} do
                                    {
                                        Transmit '0' to the bit stream.
                                    }
                                }
                            }
                    }
                }
```

4. Empirical Evaluation Discussion

This section presents a simulation assessment of the performance of the proposed HSICA with the other state-of-the-art HSICAs on the basis of different performance metrics for coding complexity, coding memory, and coding efficiency on four different HS images, which have artificial structures (building, road, etc.) and natural scenes (forest, water body, etc.). The HS image is transformed with a level wavelet transform and converted into a 1D array from the 3D transform HS image through linear indexing. The transform coefficients are quantized to the nearest integer. The input (original) HS image is denoted as A (i,j,k) while the reconstructed HS image after the compression process is denoted as B(i,j,k).

4.1. Performance Metrics

Performance of the proposed coder with other coders is evaluated on the basis of three parameters, which are named as coding efficiency, coding memory, and coding complexity [76-80]. The distinct performance metrics are used for the comparative analysis between the HSICAs. For coding efficiency, Peak Signal-To-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Feature Similarity Index (FSIM), and Bjøntegaard Metric Calculation (BD-PSNR) are used as performance metrics, while for the coding complexity, the total time required by the HSICAs for the calculation of encoding and decoding of the coefficients [73-81]. For the coding memory calculation, the memory required to complete the coding process is known as coding memory, and it is calculated in KiloBytes (KB) [75-82]. Numerically, PSNR is represented as in Equation 2, while linked Mean Squared Error (MSE) is specified in Equation 3 [83].

$$PSNR\ (dB) = 20\ \log_{10}\left[\frac{\max\{A(i,j,k)\}}{MSE}\right] \tag{2}$$

$$MSE = \frac{1}{(M \times M \times M)} \sum_k \sum_j \sum_i [A(i,j,k) - B(i,j,k)]^2 \tag{3}$$

SSIM [76] is Represented Mathematically as Equation 4

$$\mathrm{SSIM}\ (A, B) = \frac{(2\mu_A \mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)} \tag{4}$$

Input HS image is defined as ‘A’ while its mean, average, and variance are defined as $'\mu_A'$ & $'\sigma_A^2'$. Reconstructed HS image is defined as ‘B’ while its mean, average, and variance are defined as $'\mu_B'$ & $'\sigma_B^2'$. C_1 and C_2 are the correction factors and σ_{AB} is the cross-covariance between the input HS image and the reconstructed HS image.

4.2. Hyperspectral Image Dataset

Four publicly available HS images are used, whose details are covered in Table 2.

Table 2. Details of the HS images dataset

HS Image Notation in Simulation Test	HS Image	Sensor Detail	Spatial Measurement	Spectral Measurement	Spectral Resolution (nm)	Spatial Resolution (mt)
Dataset 1	Washington DC Mall	HYDICE	1280 & 307	191	10	3 to 4
Dataset 2	Botswana	Nasa EO-1	1476 & 256	242	10	30
Dataset 3	Pavia Centre	ROSIS	1096 & 1096	102	5	1.3
Dataset 4	Jasper Ridge	AVRIS	100 & 100	224	10	4 to 20

4.3. Experiment Result Analysis

Nine image compression algorithms have been used for the analysis named as 3D-SPECKas Coder I [56], 3D-SPIHT as Coder II [55], 3D-WBTC as Coder III [64], 3D-LSK as Coder IV [61], 3D-NLS as Coder V [62], 3D-LMBTC Coder VI [66], 3D-ZM-SPECK as Coder VII [68], 3D-M-ZM-SPECK as Coder VIII [74], 3D-SLS as Coder IX [75] and compare the associated result with the proposed compression algorithm named 3D-OL-SPIHT as Coder X.

4.3.1. Coding Efficiency

It is clear from Table 3 that 3D-OL-SPIHT has the best performance with the other HSICAs. It is due to the use of two different thresholds. It has been noticed from Table 3 that variation between the PNSR of the proposed 3D-OL-SPIHT and 3D-SPECK [56] exists in the range of 0.1 dB to 0.57 dB for HS Image I, 0.05 dB to 0.43 dB for HS Image II, 0.22 dB to 0.6 dB for HS Image III, and 0.23 dB to 1.04 dB for HS Image IV. In the same way, variation between the PNSR of proposed 3D-OL-SPIHT and 3D-SPIHT [55] exists in the range of 0.31 dB to 0.7 dB for HS Image I, 0.2 dB to 0.51 dB for HS Image II, 0.43 dB to 0.65 dB for HS Image III, and 0.4 dB to 0.69 dB for HS Image IV. The variation of 3D-OL-SPIHT and 3D-WBTC [64] exists in the range of 0.1 dB to 0.52 dB for HS Image I, 0.06 dB to 0.4 dB for HS Image II, 0.27 dB to 0.65 dB for HS Image III, and -0.16 dB to 0.64 dB for HS Image IV. The variation of 3D-OL-SPIHT and 3D-NLS (listless HSICA) [62] exists in the range of 0.5 dB to 1.45 dB for HS Image I, 0.28 dB to 0.89 dB for HS Image II, 0.5 dB to 0.81 dB for HS Image III, and 0.38 dB to 0.77 dB for HS Image IV. The variation of 3D-OL-SPIHT and 3D-SLS [75] exists in the range of 0.03 dB to 0.12 dB for HS Image I, 0.05 dB to 0.21 dB for HS Image II, 0.13 dB to 0.2 dB for HS Image III, and 0.06 dB to 0.23 dB for HS Image IV. From Table 4, it is clear that the proposed HSICA has a greater number of refinement coefficients (already significant coefficients) for almost all bit rates to the 3D-SPIHT and 3D-NLS, which gives a clear reason for the high coding efficiency. In the same way, in most cases, the count of newly significant coefficients is higher with reference to the 3D-SPIHT and 3D-NLS. Table 5 sheds light on the Bjontegaard Metric Calculation (BD-PSNR) of the proposed HSICA with the other transform-based HSICA. Table 6 gives the comparative analysis of SSIM [77], and Table 7 gives the comparative analysis of FSIM [78].

4.3.2. Coding Complexity

Coding complexity of any HSICA is calculated by the time required for encoding or decoding the transform coefficients [84-86]. If any HSICA is complex, then the calculation of the coefficient takes a lot of time, including the associated arithmetical/logical/algebraic calculations [80]. It has been clear from Table 8 and Table 9 that the encoding time is greater than the decoding time. This is due to two reasons

- The decoder does not have to perform the significance testing.
- Decoder does not need to build the SOTs during the decoding process.

From the encoding and decoding time, we come to know that the 3D-OL-SPIHT has the least encoding/decoding time with reference to the other list-based HSICAs. It is due to the use of only one list (or removing other lists), which reduces the processing time in writing/reading memory operations. Moreover, with reference to the 3D-SLS [75], speed improvement is the result of eliminating the need to recompute the offspring for every significant root. This root lies from the resolution level of R_0 to R_{L-2} twice in each biplane.

4.3.3. Coding Memory

The coding memory requirement of the proposed HSICA is measured as the memory required by the HSICA for the encoding or decoding of the coefficients [87, 88].

List other list-based HSICAs; 3D-OL-SPIHT coding memory is variable. It has been clear from Table 10 that 3D-OL-SPIHT has significantly lower coding memory requirement than 3D-SPIHT and 3D-WBTC, but it has slightly higher coding memory requirement than 3D-SLS. For 3D-SPIHT, the total memory required for the encoding of the HS image, the coding memory at the full bit rate is considered. At this rate, the number of entries for the LSP and LIP lists is equal to twice the total coefficients (M x M x M) while the total number of entries in LIS is equal to the ‘M x M x M/8’. Thus, the total coding memory is defined as in Equation 5.

$$MEM_{3D-SPIHT} = \left\{3\alpha(2N^3) + 3\beta\left(\frac{N^3}{8}\right)\right\} \quad (5)$$

Where α is the number of bits required to store the one address of the coefficients present in LSP and LIP, while β is the number of bits required to store each coefficient present in the LIS, the proposed HSICA 3D-OL-SPIHT uses fixed size memory where the total number of entries in the ‘LRS’ is ‘M x M x M/8’ and its size is 4 bits per coefficients and 2 bits per root. The coding memory of the proposed HSICA is defined as in Equation 6 (full HS image reconstruction)

$$MEM_{3D-OL-SPIHT} = \left\{3\beta\left(\frac{N^3}{8}\right) + 4N^3 + \frac{N^3}{4}\right\} \quad (6)$$

Table 3. Summary of coding efficiency in PSNR for compression algorithms

Bit Rate	Coder I [56]	Coder II [55]	Coder III [64]	Coder IV [61]	Coder V [62]	Coder VI [66]	Coder VII [68]	Coder VIII [74]	Coder IX [75]	Coder X (Proposed)
	Dataset 1									
0.025	34.56	34.43	34.54	34.47	34.43	34.35	34.44	34.11	34.89	34.92
0.05	36.52	36.27	36.47	36.51	36.28	36.52	36.48	36.34	36.77	36.84
0.1	38.53	38.28	38.50	38.35	38.12	38.29	38.33	38.04	38.62	38.69
0.2	41.54	41.34	41.52	41.49	41.27	41.19	41.42	41.17	41.76	41.82
0.25	42.97	42.89	43.08	43.12	41.92	42.02	42.17	42.87	43.26	43.37
0.5	46.81	46.6	46.81	46.76	46.41	46.09	46.73	46.21	46.82	46.91
0.75	50.54	50.41	50.59	50.61	50.33	50.37	50.51	50.11	51.01	51.11
1	53.52	53.32	53.51	53.49	53.33	53.46	53.47	53.12	53.84	53.92

2	66.09	66.02	66.19	65.97	65.91	65.84	66.01	65.22	66.41	66.53
	Dataset 2									
0.025	27.53	27.41	27.52	27.32	27.37	27.31	27.32	26.98	27.84	27.92
0.05	30.22	29.99	30.21	30.20	29.97	30.19	30.20	29.94	30.35	30.42
0.1	32.57	32.48	32.57	32.48	32.18	32.04	32.07	31.89	32.74	32.84
0.2	34.78	34.63	34.75	34.66	34.51	34.64	34.66	34.21	34.96	35.01
0.25	35.62	35.5	35.62	35.63	35.5	35.55	35.56	35.25	35.87	35.92
0.5	39.09	39.03	39.18	38.76	38.63	38.46	38.47	38.02	39.31	39.52
0.75	42.09	41.95	42.08	41.48	41.92	41.28	41.30	40.97	42.28	42.38
1	44.19	44.05	44.18	44.20	44.05	43.89	43.91	43.55	44.39	44.48
2	51.48	51.33	51.47	51.22	51.25	50.85	50.86	50.66	51.43	51.53
	Dataset 3									
0.025	28.83	28.68	28.82	28.68	28.62	28.64	28.67	28.44	29.08	29.21
0.05	30.32	30.06	30.30	30.29	30.06	30.28	30.29	30.01	30.45	30.59
0.1	32.43	32.17	32.38	32.23	32.01	32.25	32.29	31.98	32.51	32.65
0.2	34.84	34.62	34.83	34.61	34.51	34.56	34.58	34.38	35.01	35.14
0.25	35.70	35.49	35.69	35.49	35.31	35.42	35.44	35.08	35.92	36.05
0.5	39.06	39.03	39.01	38.97	38.85	38.77	38.78	38.64	39.48	39.66
0.75	42.24	42.04	42.24	42.01	41.97	41.83	41.84	41.69	42.49	42.65
1	45.16	44.93	45.16	45.14	44.82	44.50	44.50	44.34	45.38	45.58
2	55.46	55.31	55.46	55.34	55.24	54.89	54.91	54.31	55.57	55.74
	Dataset 4									
0.025	29.84	29.68	29.82	29.73	29.58	29.71	29.74	29.51	30.08	30.17
0.05	32.27	31.97	32.25	32.26	31.88	32.30	32.22	31.97	32.41	32.62
0.1	35.08	35.11	35.07	35.29	35.04	35.06	35.03	34.84	35.48	35.71
0.2	39.35	39.13	39.60	39.40	39.01	39.11	39.41	39.02	39.55	39.78
0.25	41.11	41.24	42.01	41.74	41.21	41.45	41.28	40.89	41.68	41.85
0.5	45.98	46.33	46.78	46.26	46.24	45.91	46.14	45.87	46.89	47.02

0.75	50.94	51.01	51.34	51.06	51.31	51.41	51.39	50.76	51.46	51.69
1	54.77	54.72	55.13	54.86	54.62	54.78	54.92	54.55	55.11	55.17
2	61.02	60.85	61.04	60.96	60.84	60.90	60.89	60.77	61.09	61.25

Table 4. Image quality of 3D-SPIHT, 3D-NLS, and the proposed coder for two HS images

Bit Rate	**Hyperspectral Image: Washington DC Mall**								
	3D-SPIHT [55]			**3D-NLS [62]**			**3D-OL-SPIHT**		
	HS Image Quality (PSNR)	**Newly Significant Coefficients of the last bit plane**	**Refinement Coefficients**	**HS Image Quality (PSNR)**	**Newly Significant Coefficients of the last bit plane**	**Refinement Coefficients**	**HS Image Quality (PSNR)**	**Newly Significant Coefficients of the last bit plane**	**Refinement Coefficients**
0.025	34.43	10541	25353	34.43	10402	25459	34.92	10311	26029
0.05	36.27	20617	69016	36.28	20691	70002	36.84	21097	71109
0.1	38.28	59345	119603	38.12	57214	118256	38.69	59018	128591
0.3	43.3	151341	356404	43.3	152051	356018	43.75	153094	369088
0.5	46.6	342421	522362	46.41	342098	522419	46.91	354119	531002
0.7	49.53	544101	612576	49.5	544094	611812	49.87	551594	625847
1	53.32	639863	876652	53.33	640120	876694	53.95	649912	884183
	Hyperspectral Image: Jasper Ridge								
0.025	29.68	19512	25986	29.58	19422	25512	30.17	19844	26109
0.05	31.97	29512	69220	31.88	27051	68529	32.62	27159	69451
0.1	35.11	59406	119540	35.04	57256	115269	35.71	57789	117005
0.3	41.89	235813	278102	41.74	233584	277169	42.12	238419	279111
0.5	46.33	413563	415800	46.24	399779	411204	47.02	398411	420059
0.7	50.06	495537	584115	49.94	482911	562135	50.98	479957	569488
1	54.72	679566	769397	54.62	662581	754284	55.17	661125	769895

Table 5. Summary of BD-PSNR calculation for different HSICA

HS Image	**Coder I [56]**	**Coder II [55]**	**Coder III [64]**	**Coder IV [61]**	**Coder V [62]**	**Coder VI [66]**	**Coder VII [68]**	**Coder VIII [74]**	**Coder IX [75]**
Dataset 1	0.3209	0.4975	0.3101	0.3558	0.6913	0.662	0.5087	0.7291	0.0805
Dataset 2	0.2698	0.4043	0.2694	0.4291	0.5229	0.6176	0.6014	0.9177	0.0914
Dataset 3	0.3462	0.544	0.3651	0.4936	0.6517	0.6405	0.6216	0.8926	0.1501
Dataset 4	0.5535	0.5928	0.2574	0.4014	0.6357	0.5262	0.4872	0.7992	0.1774

Table 6. Summary of coding efficiency in SSIM for compression algorithms

Bit Rate	Coder I [56]	Coder II [55]	Coder III [64]	Coder IV [61]	Coder V [62]	Coder VI [66]	Coder VII [68]	Coder VIII [74]	Coder IX [75]	Coder X (Proposed)
	Dataset 1									
0.025	0.385	0.386	0.384	0.385	0.386	0.386	0.384	0.377	0.39	0.395
0.05	0.480	0.479	0.480	0.480	0.479	0.480	0.480	0.471	0.475	0.48
0.1	0.587	0.587	0.585	0.584	0.587	0.589	0.590	0.581	0.589	0.592
0.2	0.677	0.675	0.678	0.677	0.678	0.677	0.677	0.664	0.681	0.688
0.25	0.701	0.702	0.702	0.699	0.703	0.702	0.702	0.697	0.709	0.712
0.5	0.790	0.788	0.787	0.789	0.789	0.788	0.787	0.774	0.798	0.808
0.75	0.847	0.846	0.849	0.846	0.846	0.846	0.846	0.845	0.85	0.858
1	0.886	0.886	0.886	0.888	0.887	0.886	0.886	0.884	0.89	0.894
2	0.914	0.912	0.915	0.914	0.912	0.915	0.915	0.911	0.918	0.924
	Dataset 4									
0.025	0.297	0.288	0.299	0.299	0.285	0.300	0.301	0.299	0.302	0.311
0.05	0.346	0.338	0.345	0.346	0.341	0.349	0.349	0.248	0.339	0.351
0.1	0.437	0.431	0.437	0.436	0.430	0.437	0.437	0.437	0.432	0.442
0.2	0.518	0.513	0.518	0.518	0.514	0.518	0.518	0.517	0.511	0.525
0.25	0.545	0.543	0.545	0.552	0.545	0.553	0.553	0.550	0.547	0.558
0.5	0.603	0.601	0.603	0.608	0.606	0.611	0.611	0.604	0.601	0.612
0.75	0.63	0.629	0.63	0.636	0.631	0.636	0.636	0.633	0.632	0.641
1	0.645	0.645	0.645	0.648	0.646	0.647	0.647	0.644	0.649	0.658
2	0.666	0.666	0.666	0.666	0.666	0.668	0.668	0.663	0.666	0.667

Table 7. Summary of coding efficiency in FSIM for compression algorithms

Bit Rate	Coder I [56]	Coder II [55]	Coder III [64]	Coder IV [61]	Coder V [62]	Coder VI [66]	Coder VII [68]	Coder VIII [74]	Coder IX [75]	Coder X (Proposed)
	Dataset 1									
0.025	0.626	0.626	0.660	0.624	0.626	0.616	0.616	0.614	0.63	0.632
0.05	0.641	0.642	0.712	0.639	0.643	0.637	0.638	0.633	0.648	0.651
0.1	0.71	0.695	0.787	0.712	0.712	0.716	0.716	0.711	0.709	0.711
0.2	0.774	0.759	0.83	0.78	0.779	0.784	0.784	0.777	0.789	0.805
0.25	0.784	0.776	0.843	0.783	0.790	0.795	0.795	0.791	0.81	0.825
0.5	0.851	0.844	0.909	0.855	0.856	0.863	0.863	0.849	0.864	0.871
0.75	0.891	0.888	0.947	0.910	0.889	0.903	0.903	0.896	0.901	0.899

1	0.919	0.917	0.970	0.920	0.918	0.918	0.918	0.912	0.931	0.933
2	0.978	0.978	0.996	0.980	0.980	0.981	0.981	0.978	0.988	0.991
	Dataset 4									
0.025	0.286	0.286	0.282	0.290	0.288	0.295	0.303	0.288	0.29	0.293
0.05	0.293	0.288	0.289	0.294	0.286	0.302	0.302	0.294	0.294	0.299
0.1	0.321	0.319	0.320	0.321	0.320	0.338	0.338	0.337	0.329	0.334
0.2	0.443	0.434	0.443	0.452	0.439	0.488	0.488	0.482	0.441	0.444
0.25	0.518	0.522	0.518	0.530	0.529	0.537	0.537	0.531	0.526	0.531
0.5	0.694	0.692	0.694	0.699	0.696	0.750	0.750	0.745	0.698	0.701
0.75	0.807	0.812	0.808	0.817	0.813	0.835	0.834	0.827	0.828	0.832
1	0.871	0.867	0.872	0.869	0.867	0.873	0.873	0.866	0.889	0.889
2	0.979	0.978	0.979	0.979	0.978	0.979	0.979	0.965	0.985	0.984

Table 8. Summary of encoding time for compression algorithms

Bit Rate	Coder I [56]	Coder II [55]	Coder III [64]	Coder IV [61]	Coder V [62]	Coder VI [66]	Coder VII [68]	Coder VIII [74]	Coder IX [75]	Coder X (Proposed)
	Dataset 1									
0.025	3.09	1.44	1.71	0.43	0.51	1.33	0.83	0.49	1.59	1.01
0.05	6.23	2.71	2.78	0.55	0.65	1.9	1.11	0.61	2.91	1.97
0.1	25.1	7.5	6.50	0.8	0.91	3.9	1.78	0.88	8.11	5.69
0.2	57.9	25.8	24.80	1.1	1.21	5.1	2.81	1.17	28.09	21.38
0.25	104.58	31.19	29.92	1.71	1.79	10.61	5.05	1.77	33.24	29.08
0.5	414.8	140.1	211.2	2.5	2.64	11.3	7.41	2.61	149	109.3
0.75	950.77	370.29	713.02	3.57	3.88	17.22	10.47	3.71	381.35	198.2
1	1497.5	575	804	4.41	4.57	21.12	13.21	4.49	594.2	487.1
2	3822.03	1426.61	4409.82	11.55	14.58	40.09	23.39	12.91	1484.21	1297.8
	Dataset 2									
0.025	5.19	1.08	2.05	0.51	1.99	2.33	1.55	0.91	1.24	0.85
0.05	8.01	2.19	3.68	0.94	9.43	3.08	2.08	1.74	2.59	2.01
0.1	42.11	5.47	7.26	2.01	12.38	4.79	3.49	3.12	7.02	5.19
0.2	68.31	17.14	20.14	4.27	14.01	8.64	5.26	4.98	19.22	16.59
0.25	91.77	23.38	67.74	7.24	17.56	10.44	9.09	8.17	28.95	22.31
0.5	402.48	100.15	179.99	9.95	18.22	19.91	11.76	10.92	109.71	89.68
0.75	1066.62	581.78	682.40	12.17	20.19	29.03	17.10	16.68	602.37	518.9
1	1320.53	641.34	875.01	14.96	22.96	35.17	19.04	18.07	688.21	549.1
2	5495.06	2771.88	4096.40	26.44	41.94	75.21	38.87	34.14	2908.63	2789.2

	Dataset 3									
0.025	5.34	1.96	2.15	0.57	3.34	2.42	1.1	1.03	2.08	1.94
0.05	10.17	8.34	3.31	2.03	7.72	4.62	3.69	3.12	8.91	8.08
0.1	24.02	17.65	6.54	3.65	10.32	7.78	4.34	4.03	19.29	17.95
0.2	69.81	22.2	14.58	5.15	11.26	9.16	6.02	5.87	26.35	24.87
0.25	91.62	38.56	37.3	7.7	14.92	12.26	8.16	7.91	40.6	37.21
0.5	371.82	187.18	198.26	9.14	17.71	17.57	11.38	10.57	198.21	184.2
0.75	955.11	440.87	597.21	11.01	21.48	24.43	14.32	12.86	478.77	357.1
1	1553.24	715.1	1011.4	13.69	24.92	27.96	18.64	16.28	731.02	702.9
2	4770.03	2338.89	3882.91	27.52	39.89	55.15	36.57	32.14	2501.16	2384.2
	Dataset 4									
0.025	3.01	1.34	1.66	0.49	0.53	1.38	0.86	0.51	1.51	1.27
0.05	6.01	2.55	2.36	0.56	0.67	2.06	1.14	0.63	2.94	2.51
0.1	21.1	7.6	6.4	0.9	1.09	3	1.77	1.03	9.03	8.19
0.2	54.2	20.6	17.7	1.2	1.34	5.2	2.84	1.27	22.31	21.17
0.25	97.1	37.92	21.35	1.84	1.99	8.72	4.73	1.91	54.29	52.29
0.5	315.3	101.6	182.4	2.6	2.74	10.9	6.24	2.72	121.38	111.3
0.75	705.45	267.31	530.54	3.3	3.77	17.79	10.63	3.51	299.74	197.8
1	757.3	425.4	942.8	5.1	5.34	22.7	15.84	5.24	438.21	387.6
2	3255.3	1213.1	2380.4	9.55	14.2	59.26	56.42	13.67	1249.8	1201.2

Table 9. Summary of decoding time for compression algorithms

Bit Rate	**Coder I [56]**	**Coder II [55]**	**Coder III [64]**	**Coder IV [61]**	**Coder V [62]**	**Coder VI [66]**	**Coder VII [68]**	**Coder VIII [74]**	**Coder IX [75]**	**Coder X (Proposed)**
	Dataset 1									
0.025	0.79	0.99	0.75	0.38	0.42	0.64	0.79	0.64	1.28	0.92
0.05	1.09	2.58	1.36	0.47	0.57	1	1.01	0.86	2.66	1.75
0.1	17.4	6.1	5.02	0.7	0.79	2.3	1.71	1.15	7.09	5.28
0.2	48.8	24.8	22.5	1.07	1.07	3.3	2.71	2.12	26.38	18.59
0.25	88.32	28.85	17.44	1.64	1.67	6.98	4.78	4.18	30.94	27.08
0.5	339.1	135.40	191.7	2.2	2.31	7.7	6.82	5.87	140.28	104.27
0.75	899.76	312.97	612.75	2.94	3.01	13.84	9.01	9.01	341.97	184.1
1	1289.7	504.02	774.08	3.7	4.24	15.50	12.31	10.97	547.16	452.7
2	3480.44	1196.24	4021.39	10.08	14.02	31.28	21.89	19.43	1298.27	1224.3
	Dataset 2									
0.025	4.03	0.87	1.43	0.44	1.14	1.34	1.43	0.79	1.08	0.74
0.05	7.21	1.67	2.49	0.81	8.66	2.01	1.74	1.61	2.11	1.89

0.1	29.82	4.09	5.05	1.59	10.48	3.54	3.08	2.94	6.58	4.59
0.2	49.95	15.09	15.70	3.92	12.57	6.61	4.51	4.02	17.81	14.51
0.25	69.76	21.09	52.75	6.74	16.19	8.45	8.24	7.94	23.19	21.74
0.5	347.89	89.45	167.04	9.12	17.04	16.58	10.19	10.08	101.77	81.58
0.75	948.92	505.29	598.09	11.77	19.53	27.55	15.22	15.74	558.34	502.1
1	1227.43	600.26	804.18	13.98	20.84	32.83	17.79	17.12	641.89	537.2
2	4872.53	2525.11	3785.90	24.84	39.73	70.88	34.37	32.97	2729.31	2789.2
	Dataset 3									
0.025	3.06	0.77	1.12	0.49	2.89	2.04	1.01	0.92	1.79	1.71
0.05	4.49	6.04	1.85	1.87	5.01	4.31	3.31	2.89	7.28	6.94
0.1	13.33	15.02	5.06	3.24	9.14	7.32	3.97	3.84	17.38	16.84
0.2	53.56	20.33	12.55	4.58	10.08	8.47	4.84	4.44	23.24	22.67
0.25	60.84	33.43	31.45	7.02	14.17	11.87	7.21	7.01	37.08	34.36
0.5	315.94	180.19	186.22	8.48	16.91	16.79	10.21	9.97	188.54	174.2
0.75	860.41	412.32	546.32	10.26	20.48	22.89	13.86	12.47	447.31	341.9
1	1445.08	684.96	976.09	12.99	23.42	23.81	18.01	15.84	702.97	694.2
2	4357.62	2151.76	3423.52	25.74	37.19	50.21	34.92	31.03	2271.05	2220.7
	Dataset 4									
0.025	1.89	1.22	1.19	0.38	0.47	0.65	0.83	0.44	1.38	1.19
0.05	3.5	1.98	2.04	0.47	0.58	1.02	1.12	0.52	2.17	2.41
0.1	15.3	7.41	4.57	0.75	0.97	1.9	1.73	0.94	8.02	7.34
0.2	37.03	17.6	15.3	1.1	1.22	3.7	2.7	1.15	20.35	19.87
0.25	72.4	31.06	19.45	1.14	1.54	6.2	4.11	1.47	44.27	52.98
0.5	290.8	98.5	178.4	2.5	2.59	9.3	6.06	2.57	114.21	108.2
0.75	615.41	232	418.91	3.05	3.41	19.56	9.51	3.33	248.21	187.1
1	726.9	421.3	923.1	4.3	5.02	21.8	12.31	4.91	430.78	341.6
2	3062.8	1132.2	2068.9	6.69	13.54	47.2	45.18	12.65	1181.96	1184.2

Table 10. Summary of coding memory for compression algorithms

Bit Rate	**Coder I [56]**	**Coder II [55]**	**Coder III [64]**	**Coder IV [61]**	**Coder V [62]**	**Coder VI [66]**	**Coder VII [68]**	**Coder VIII [74]**	**Coder IX [75]**	**Coder X (Proposed)**
	Dataset 1									
0.025	54.87	54.42	55.21	512	1024	12	0	32	8.95	10.23
0.05	136.1	145.3	137.9	512	1024	12	0	32	23.78	31.29
0.1	243.8	263.3	250.1	512	1024	12	0	32	41.08	44.69
0.2	416.3	438	416	512	1024	12	0	32	68.33	74.61
0.25	586.67	605.78	630.49	512	1024	12	0	32	96.92	112.3

0.5	1048.8	1060.5	1049	512	1024	12	0	32	167.54	197.8
0.75	1287.31	1333.19	1329.77	512	1024	12	0	32	296.26	335.4
1	1802.5	1826.7	1724.6	512	1024	12	0	32	411.42	457.2
2	2865.17	2897	3052.09	512	1024	12	0	32	652.48	701.5
	Dataset 2									
0.025	60.22	63.41	60.69	512	1024	12	0	32	10.54	12.28
0.05	114.86	120.09	115.57	512	1024	12	0	32	20.39	23.68
0.1	245.34	247.27	246.04	512	1024	12	0	32	46.74	50.35
0.2	463.49	495.61	473.86	512	1024	12	0	32	105.7	119.3
0.25	628.84	651.89	630.49	512	1024	12	0	32	123.4	129
0.5	1148.6	1164.8	1149.5	512	1024	12	0	32	267.8	298.1
0.75	1305.8	1299.3	1306	512	1024	12	0	32	288.7	308.8
1	2056	2090.4	2057.4	512	1024	12	0	32	470.8	499.1
2	3051	3081.8	3052.1	512	1024	12	0	32	708.3	759.8
	Dataset 3									
0.025	54.84	61.36	55.03	512	1024	12	0	32	10.09	13.21
0.05	136.4	150.95	138.5	512	1024	12	0	32	24.83	29.91
0.1	246.14	260.59	251.71	512	1024	12	0	32	41.3	47.2
0.2	418.18	462.06	418.28	512	1024	12	0	32	79.07	83.87
0.25	602.43	630.18	610.53	512	1024	12	0	32	117.9	128.3
0.5	1042.3	1086.6	1041.9	512	1024	12	0	32	196.2	211.3
0.75	1453.5	1488	1453.2	512	1024	12	0	32	335.2	367.1
1	1937.9	1919.7	1936.8	512	1024	12	0	32	426.6	464.6
2	2713.4	2665.7	2714.3	512	1024	12	0	32	633.1	662.8
	Dataset 4									
0.025	55.23	55.52	55.61	512	1024	12	0	32	8.8	10.08
0.05	143.02	145.9	143.3	512	1024	12	0	32	26.11	29.11
0.1	241.4	245.9	245.8	512	1024	12	0	32	46.31	50.47
0.2	440	445.7	443.7	512	1024	12	0	32	87.22	92.51
0.25	480.11	462.3	489.73	512	1024	12	0	32	90.29	95.66
0.5	821.6	808.9	827.9	512	1024	12	0	32	168.5	181.2
0.75	1150.17	1152.97	1155.23	512	1024	12	0	32	271.3	287.3
1	1492.71	1503.8	1532.6	512	1024	12	0	32	346.5	361.3
2	2592.52	2626.3	2503.1	512	1024	12	0	32	643.7	662.9

5. Conclusion

The manuscript presents a compression algorithm that follows the same partition rule as 3D-SPIHT, 3D-NLS, and 3D-SLS. 3D-OL-SPIHT is a modified version of 3D-SLS, which has high coding efficiency and low coding complexity due to the change of the list structure and use of two different thresholds. Due to the above-mentioned gains, 3D-OL-SPIHT can be part of real-time hyperspectral image transmission systems. Further use of advanced wavelet transforms, such as curvelet shearlet transform, can increase the coding efficiency.

Reference

[1] Chiara Cevoli et al., "Data Fusion of FT-NIR Spectroscopy and Vis/NIR Hyperspectral Imaging to Predict Quality Parameters of Yellow Flesh "Jintao" Kiwifruit," *Biosystems Engineering*, vol. 237, pp. 157-169, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Songtao Ding et al., "Removal of Environmental Influences for Estimating Soil Texture Fractions based on ZY1 Satellite Hyperspectral Images," *CATENA*, vol. 236, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Qiwei Li et al., "Lenslet-Array-based Snapshot Hyperspectral Imaging Polarimeter using Opposite Spectral Modulation," *Optics and Lasers in Engineering*, vol. 174, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Vladislav Batshev et al., "Polarizer-Free AOTF-based SWIR Hyperspectral Imaging for Biomedical Applications," *Sensors*, vol. 20, no. 16, pp. 1-10, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Yating Zhang et al., "Applications of Hyperspectral Imaging in the Detection and Diagnosis of Solid Tumors," *Translational Cancer Research*, vol. 9, no. 2, pp. 1265-1277, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] 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]

[7] Yanyi Li et al., "Adoption of Machine Learning in Intelligent Terrain Classification of Hyperspectral Remote Sensing Images," *Computational Intelligence and Neuroscience*, vol. 2020, no. 1, pp. 1-13, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] 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]

[9] 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]

[10] 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]

[11] Likitha Gongalla, and S.V. Sudha, "A Deep Learning Framework for Classification of Hyperspectral Images," *Sentiment Analysis and Deep Learning: Proceedings of ICSADL*, Springer, Singapore, vol. 1432, pp. 173-181, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mary B. Stuart et al., "High-Resolution Hyperspectral Imaging using Low-Cost Components: Application within Environmental Monitoring Scenarios," *Sensors*, vol. 22, no. 12, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Hongbin Pua, Qingyi Wei, and Da-Wen Sun, "Recent Advances in Muscle Food Safety Evaluation: Hyperspectral Imaging Analyses and Applications," *Critical Reviews in Food Science and Nutrition*, vol. 63, no. 10, pp. 1297-1313, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Wenjun Luo et al., "High-Accuracy Filtering of Forest Scenes based on Full-Waveform LiDAR Data and Hyperspectral Images," *Remote Sensing*, vol. 15, no. 14, pp. 1-25, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Kristiane de Cássia Mariotti, Rafael Scorsatto Ortiz, and Marco Flôres Ferrão, "Hyperspectral Imaging in Forensic Science: An Overview of Major Application Areas," *Science and Justice*, vol. 63, no. 3, pp. 387-395, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Lei Feng et al., "Application of Visible/Infrared Spectroscopy and Hyperspectral Imaging with Machine Learning Techniques for Identifying Food Varieties and Geographical Origins," *Frontiers in Nutrition*, vol. 8, pp. 1-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Sarat Kurapati, and Arun PV, "Spatial-Spectral Attention for Geological Mapping of Hyperspectral Sensor on Board Chandrayaan-2 Mission," *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, Pasadena, CA, USA, pp. 4158-4161, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] 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]

[19] Amit Rotem et al., "Interpretation of Hyperspectral Shortwave Infrared Core Scanning Data using SEM-based Automated Mineralogy: A Machine Learning Approach," *Geosciences*, vol. 13, no. 7, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Pravin V. Dhole et al., "A Survey on Hyperspectral Sensing Techniques for Identification of Fake Pharmaceuticals Medicines," *GRENZE International Journal of Engineering and Technology*, vol. 9, no. 2, pp. 1152-1161, 2023. [Google Scholar] [Publisher Link]

[21] Matthew S. Mills et al., "Assessment of the Utility of Underwater Hyperspectral Imaging for Surveying and Monitoring Coral Reef Ecosystems," *Scientific Reports*, vol. 13, no. 1, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Qi Zhang, and Min Shao, "Impact of Hyperspectral Infrared Sounding Observation and Principal-Component-Score Assimilation on the Accuracy of High-Impact Weather Prediction," *Atmosphere*, vol. 14, no. 3, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Shrish Bajpai, Harsh Vikram Singh, and Naimur Rahman Kidwai, "Feature Extraction and Classification of Hyperspectral Images using Singular Spectrum Analysis and Multinomial Logistic Regression Classifiers," *2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, Aligarh, India, pp. 97-100, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[24] Seyd Teymoor Seydi, Reza Shah-Hosseini, and Mahdi Hasanlou, "New Framework for Hyperspectral Change Detection based on Multi-Level Spectral Unmixing," *Applied Geomatics*, vol. 13, no. 4, pp. 763-780, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Yaman Dua, Ravi Shankar Singh, and Vinod Kumar, "Compression of Multi-Temporal Hyperspectral Images based on RLS Filter," *The Visual Computer*, vol. 38, no. 1, pp. 65-75, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[26] 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]

[27] Behnood Rasti et al., "How Hyperspectral Image Unmixing and Denoising Can Boost Each Other," *Remote Sensing*, vol. 12, no. 11, pp. 1-20, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] Md. Palash Uddin, Md. Al Mamun, and Md. 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]

[29] 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]

[30] Qiyou Jiang et al., "Hyperspectral Imaging for Early Identification of Strawberry Leaves Diseases with Machine Learning and Spectral Fingerprint Features," *Infrared Physics and Technology*, vol. 118, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[31] Peng Dou, and Chao Zeng, "Hyperspectral Image Classification using Feature Relations Map Learning," *Remote Sensing*, vol. 12, no. 18, pp. 1-25, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] Dioline Sara et al., "Hyperspectral and Multispectral Image Fusion Techniques for High Resolution Applications: A Review," *Earth Science Informatics*, vol. 14, no. 4, pp. 1685-1705, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[33] Jianwei Zheng et al., "Manifold-based Nonlocal Second-Order Regularization for Hyperspectral Image Inpainting," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 224-236, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[34] Sneha, and Ajay Kaul, "Hyperspectral Imaging and Target Detection Algorithms: A Review," *Multimedia Tools and Applications*, vol. 81, no. 30, pp. 44141-44206, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[35] Mehrube Mehrubeoglu, Austin Van Sickle, and Jeffrey Turner, "Detection and Identification of Plastics using SWIR Hyperspectral Imaging," *Imaging Spectrometry XXIV: Applications, Sensors, and Processing*, vol. 11504, pp. 85-95, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] 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]

[37] Reaya Grewal, Singara Singh Kasana, and Geeta Kasana, "Hyperspectral Image Segmentation: A Comprehensive Survey," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 20819-20872, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[38] 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]

[39] 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]

[40] Sourish Sarkar, and Rajiv Ranjan Sahay, "A Non-Local Superpatch-based Algorithm Exploiting Low Rank Prior for Restoration of Hyperspectral Images," *IEEE Transactions on Image Processing*, vol. 30, pp. 6335-6348, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[41] 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]

[42] 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]

[43] 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]

[44] Rui Li, Zhibin Pan, and Yang Wang, "The Linear Prediction Vector Quantization for Hyperspectral Image Compression," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11701-11718, 2018. [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: Proceedings of ICTIS*, Springer, Singapore, vol. 106, pp. 495-503, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[46] Wei Fu et al., "Adaptive Spectral-Spatial Compression of Hyperspectral Image with Sparse Representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 671-682, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[47] 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]

[48] 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]

[49] Ben Sujitha et al., "Optimal Deep Learning based Image Compression Technique for Data Transmission on Industrial Internet of Things Applications," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, 2021. [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] V.K. Bairagi, A.M. Sapkal, and M.S. Gaikwad, "The Role of Transforms in Image Compression," *Journal of the Institution of Engineers (INDIA): Series B*, vol. 94, no. 2, pp. 135-140, 2013. [CrossRef] [Google Scholar] [Publisher Link]

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

[53] Shrish Bajpai, and Naimur Rahman Kidwai, "Fractional Wavelet Filter based Low Memory Coding for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 26281-26306, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[54] Manish Gupta, and Puneet Agrawal, "Compression of Deep Learning Models for Text: A Survey," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 16, no. 4, pp. 1-55, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[55] 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]

[56] Xiaoli Tang, and William 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]

[57] Ying Liu, and William A. Pearlman, "Region of Interest Access with Three-Dimensional SBHP Algorithm," *Visual Communications and Image Processing*, San Jose, California, United States, vol. 6077, 2006. [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 Algorithm," *Optical Engineering*, vol. 45, no. 2, pp. 484-453, 2006. [CrossRef] [Google Scholar] [Publisher Link]

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

[60] 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]

[61] 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]

[62] 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, no. 12, pp. 735-748, 2013. [Google Scholar] [Publisher Link]

[63] A. Karami, "Lossy Compression of Hyperspectral Images using Shearlet Transform and 3D SPECK," *Image and Signal Processing for Remote Sensing XXI*, Toulouse, France, vol. 9643, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[64] 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 (IJEECS)*, vol. 15, no. 2, pp. 1001-1008, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[65] 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]

[66] 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]

[67] Shrish Bajpai, Naimur Rahman Kidwai, and Vishal Singh Chandel, "Low Memory Wavelet-based Hyperspectral Image Coding using 2D Dyadic Wavelet Transform," *International Journal of Electrical Engineering and Technology (IJEET)*, vol. 11, no. 6, pp. 25-33, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[68] 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, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[69] Shrish Bajpai, "Low Complexity Block Tree Coding for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 33205-33232, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[70] 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]

[71] 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]

[72] Shrish Bajpai et al., "Curvelet Transform based Compression Algorithm for Low Resource Hyperspectral Image Sensors," *Journal of Electrical and Computer Engineering*, vol. 2023, no. 1, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[73] Shrish Bajpai, "Low Complexity Image Coding Technique for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, vol. 82, no. 20, pp. 31233-31258, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[74] Shrish Bajpai, "Low Complexity and Low Memory Compression Algorithm for Hyperspectral Image Sensors," *Wireless Personal Communications*, vol. 131, no. 2, pp. 805-833, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[75] Vinod Kumar Tripathi, and Shrish Bajpai, "3D Single List Set Partitioning in Hierarchical Trees for Onboard Hyperspectral Image Sensors," *International Journal of Electronics and Communication Engineering*, vol. 12, no. 4, pp. 265-281, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[76] 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, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[77] De Rosal Igantius Moses Setiadi, "PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423-8444, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[78] Umme Sara, Morium Akter, and Mohammad Shorif Uddin, "Image Quality Assessment Through FSIM, SSIM, MSE and PSNR-A Comparative Study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8-18, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[79] 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, no. 3, pp. 657-680, 2023. [CrossRef] [Google Scholar] [Publisher Link]

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

[81] 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]

[82] Qizhi Huang, Zhe Zhang, and Zhenzhong Chen, "End-to-End Optimized Lossless Hyperspectral Image Compression with Bit Partition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 64, pp. 1-13, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[83] Venkata Siva Kumar Tankani et al., "Lightweight Spectral-Spatial Attention and Tucker Decomposition for Label-Free Hyperspectral Anomaly Detection," *2025 5th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumkur, India, pp 1-7, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[84] 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, no. 8, pp. 334-348, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[85] 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, no. 2, pp. 120-128, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[86] Vinod Kumar et al., "Deep Learning for Hyperspectral Image Classification: A Survey," *Computer Science Review*, vol. 53, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[87] Swalpa Kumar Roy et al., "Spectral-Spatial Morphological Attention Transformer for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[88] Chanki Pandey et al., "Hyperspectral Imagery Applications for Precision Agriculture," *Multimedia Tools and Applications*, vol. 81, no. 2, pp. 3005-3038, 2021. [CrossRef] [Google Scholar] [Publisher Link]