SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I8P123

Volume 12 Issue 8, 257-271, August 2025

Original Article

Block-Based Fractional Wavelet Filter For Low-Complexity Coding Algorithm of Hyperspectral Image With Memory Constrained Wireless Multimedia Sensor

Rajesh[1], Shrish Bajpai[2], Naimur Rahman Kidwai[3]

[1,2,3]*Electronics & Communication Engineering Department,*
Faculty of Engineering & Information Technology, Integral University, Lucknow, Uttar Pradesh, India.

[2]*Corresponding Author : shrishbajpai@gmail.com*

Received: 12 June 2025 Revised: 13 July 2025 Accepted: 14 August 2025 Published: 30 August 2025

Abstract - *Depending on the mammoth size of the hyperspectral image, the wavelet transform-based compression algorithm has achieved impressive performance in the compression of hyperspectral images. Set partition wavelet transform hyperspectral image compression algorithms have superior performance than other transform compression algorithms, such as embeddedness, low coding complexity and high coding efficiency. Fractional wavelet-based zero memory set partitioned embedded block (ZM-SPECK) reduces the demand for transform memory and coding memory with at par transform complexity and high coding efficiency. But comparing every coefficient/block/set for each frequency frame with the current threshold is time-consuming. The present algorithm deals with the complexity and memory of the transform image coding algorithms. The block-based fractional wavelet filter delivers exact transform results like other wavelet transforms, but demands the least transform memory with at par wavelet transform complexity. With the employment of the low complexity zero memory set partitioned embedded block (LC-ZM-SPECK), the coding complexity of the compression algorithm is further reduced. The simulation results show that the proposed compression algorithm reduces the overall complexity by ~ 25% to other state-of-the-art compression algorithms and reduces the transform memory by ~ 40%, making it a suitable choice for the resource-constrained hyperspectral image sensors.*

Keywords - *Hyperspectral image analysis, Coding algorithm, Wavelet transform, Fractional wavelet filter, Wireless sensor network.*

1. Introduction

HyperSpectral (HS) imaging is an advanced optical sensing technique that captures images in multiple narrow (~ 10 nm) and contiguous spectral wavelengths (400 nm to 2500 nm) [1, 2]. This technique gathers hundreds of images at densely packed spectral frequency bands (visible to short-wave infrared of the electromagnetic spectrum) for the same spatial location [3]. With its high spatial and spectral resolution capacity, an HS image can detect the object's chemical and physical properties under observation [4]. Every pixel present in the HS image consists of a spectral signature similar to a unique fingerprint [5]. This rich data is invaluable for multiple applications such as astronomy [6], biomedical [7], cultivation [8], defence [9], environmental monitoring [10], food quality analysis [11], health care (medical) [12], geology [13], remote sensing [14], urban planning [15], security [16] etc. Remote Sensing (RS) is fastest growing field of HS images in which researchers are involved in the development of the algorithm for band selection [17], change detection [18], classification [19], compression [20], denoising [21], dimensional reduction [22], feature extraction [23], and segmentation [24]. Despite the enormous potential it possesses, the implementation of HyperSpectral Imaging (HSI) technology is met with a number of obstacles [25]. Due to its data size (~150 MB), it poses big challenges to the HS image sensor performance, communication channel and cost [26].

An efficient compression algorithm is required to improve the sensor performance by reducing the coding complexity, trimming the sensor power consumption, saving the communication channel bandwidth, minimizing the browsing time and cutting down the overall processing cost [27]. The performance of the compression algorithm is measured by the coding efficiency, coding memory and coding complexity [28]. The Compression Ratio (CR), Peak Signal to Noise Ratio (PSNR) and SSIM (Structural Similarity Index) are the main key performance indicators for the coding efficiency, while the coding memory is defined as the memory required by the compression algorithm during the compression process [29]. The coding complexity is measured with the aid of the time taken by the compression algorithm to perform the encoding and decoding process (without the time required for the wavelet transform and inverse wavelet transform process) [30].

HS image compression algorithms can be split into different subgroups on the basis of loss of HS image data during transmission or on the basis of the working of the encoder of the compression algorithm [31, 32]. The compression algorithm is divided into 3 categories, namely lossy, lossless and near lossless on the basis of the HS image data loss (lost HS image data can not be recovered) [30]. While the compression ratio is low for lossless compression, for near-lossless loss less, the compression ratio is slightly higher. With the loss of some HS image data, lossy compression has the highest CR. It has been known that human eyes can not detect HS image degradation after 40 dB (PSNR) [2]. Compression algorithms are divided into the seven different sub group named as predictive coding [33], vector quantization [34], transform coding [35], tensor decomposition [36], compressive sensing [37], machine learning based compression algorithm [38] and hybrid compression algorithm [39], based on the process of coding (encoding and decoding) of compression algorithm.

The TC-based compression algorithms exploit the correlation (unwanted redundancy) of the HS image through the mathematical transform (fourier, wavelet, cosine, curvelet, shearlet) and convert the HS image from the time domain to the frequency domain [40]. Among all the transforms, the performance of the wavelet transform is best as it exhibits a simultaneous localization in the time and frequency domains [41].

The two-fold major contribution to the present manuscript is as follows

- The demand for the transform memory (wavelet) is only reduced to a few kilobytes.
- The complexity (coding) is reduced, saving transmission bandwidth and power consumption.

The present manuscript is organized as follows. Section 2 covers the background of the proposed compression algorithm, including the overview of the set-partitioned compression algorithms. Section 3 describes the materials and methods, including A description. Block-Based Fractional Wavelet Filter (BFrWF) and B. 2D-Low Complexity Zero Memory Set Partitioned Embedded Block (2D-LC-ZM-SPECK). Section 4 focuses on the proposed compression algorithm, detailing its procedure and analyzing its performance. Section 5 presents experimental validation of the compression algorithm’s performance on coding efficiency, memory, and complexity. Finally, the conclusions of the proposed compression algorithm are provided in Section 6.

2. Background

Compression of any HS image is achieved by removing the redundancy present within. There are two types of redundancy present in the HS image: spatial and spectral redundancy [42]. Spectral redundancy exists due to the similarity between the pixels present in the nearby continuous frequency frames of HS images, while spatial redundancy is present due to the correlation between the nearby pixels. It has been known that spectral redundancy has a higher weight than spatial redundancy in any HS image [43, 44].

The mathematical transform removes redundancy by eliminating the correlation present in the HS image [45]. The 3D dyadic wavelet transform (3D-DWT) is an optimum choice to reduce the present correlation in the HS image. Apart from this, it has the excellent property of energy clustering in time and space. The 3D wavelet transform is implemented as 1 D wavelet transform in all three dimensions of the HS image, one by one [28].

Among wavelet transform-based compression algorithms, mathematical transform-based set partition compression algorithms used the properties of the wavelet transform of energy clustering to achieve the compression [26]. It uses the set structure to represent the large number of insignificant coefficients at the high bit plane. Due to the set structure, this large number of insignificant coefficients in the set can be represented by a single bit ‘0’. Due to this property, these compression algorithms have a greater advantage than the other transform-based compression algorithms [30].

A short comparative analysis between the different compression algorithms is covered in Table 1.

3. Related Work

The present section will briefly overview the associated wavelet transform, “Block Based Fractional Wavelet Filter,” applied on the HS image frame by frame.

3.1. Block-Based Fractional Wavelet Filter

There are three types of approaches for calculating transform coefficients that minimize the necessity of transform memory [46]. Line-based DWT, strip-based DWT, and block-based DWT are the three approaches [47-49]. When using line-based DWT, lines from the image are read into the system buffer until vertical filtering is possible. When applied to wide blocks, the DWT based on stripes is analogous to the DWT based on lines. Block-based DWT is more applicable to the job that we are doing since it first divides the image into several different blocks and then transforms each of those blocks individually. They are not appropriate for altering images utilizing low- Cost sensor nodes or portable devices because of the memory and complexity limits those devices have.

Table 1. A short comparable review of multiple mathematical transform-based HSICAs for compression of HS images

Set Partition Type	MT-SP-HSICA	Year	Ref	List/ Listless	Coding Memory	Embeddedness Property
Zero Block Cube	3D-SPECK	2006	[51]	List (2)	Variable	Yes
	3D-SPEZBC	2007	[60]	List (2)	Variable	
	3D-LSK	2010	[52]	Listless	Fixed	
	3D-ST-SPECK	2015	[61]	List (2)	Variable	
	3D-ZM-SPECK	2022	[57]	Listless	Zero	
	3D-BCP-ZM-SPECK	2023	[32]	Listless	Fixed	
	3D-M-ZM-SPECK	2023	[28]	Listless	Zero	
	FrWF based 2D-ZMSPECK	2024	[58]	Listless	Zero	
	3D-LBCSPC	2024	[2]	Listless	Fixed	
	BFrWF based 2D-ZMSPECK	2025	[62]	Listless	Zero	
	CT-LSK	2025	[63]	Listless	Fixed	
	SFrWF based 2D-LC-ZMSPECK	2025	[64]	Listless	Fixed	
Zero Tree	3D-SPIHT	2004	[53]	List (3)	Variable	
	3D-FSPIHT	2012	[65]	List (3)	Variable	
	3D-NLS	2013	[54]	Listless	Fixed	
	3D-SDB-SPIHT	2017	[66]	List (3)	Variable	
	3D-LEZSPC	2023	[40]	Listless	Fixed	
	3D-BPEC	2023	[41]	Array (6)	Variable	
	3D-MELS	2023	[27]	Listless	Fixed	
	3D-LMZC	2024	[30]	Listless	Fixed	
	3D-SLS	2025	[67]	List (1)	Variable	
Zero Block Cube Tree	3D-WBTC	2019	[55]	List (3)	Variable	
	3D-LMBTC	2019	[56]	Listless	Fixed	
	3D-M-WBTC	2019	[59]	List (3)	Variable	
	3D-LCBTC	2022	[26]	List (2)	Fixed	
	3D-LBCTC	2022	[20]	Listless	Fixed	

The fractional wavelet filter (FrWF) [58] is one of the more recent developments that has helped reduce the amount of memory needed for the computation of forward DWT [47]. Although FrWF requires very little memory for its implementation, the amount of memory it needs to store data still varies depending on the size of the HS image frequency frame, making it unsuitable for transforming HS images on platforms with limited memory. Block-based FrWF [49] is a modified form of FrWF.

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

3.2. 2D-Low Complexity Zero Memory Set Partitioned Embedded Block (2D-LC-ZM-SPECK)

The 2D-Low Complexity Zero Memory Set Partitioned Embedded block is a low complexity version of 2D-ZM-SPECK, which uses the magnitude of the largest coefficient in each subband through search and stores it while computing the transform before the encoding or decoding process [50]. Thus, only a small piece of memory is required to store the highest coefficient for each sub-band according to the level of transformation. For the ‘L’ level of BFrWF transform for one frequency frame (spatial dimensions only), ‘N x (3L + 1)’ coefficients are required, with the necessity of coding memory of ‘8N x (3L + 1)’ bytes of memory. After this listing of the highest coefficients, the encoding process starts. When seen from a hierarchical perspective, the detailed sub-bands are grouped in three distinct orientations, which are the HLn, LHn and HHn, n = 1,2,3,…, L. The sub-bands are arranged in groups as S_θ^n and I^n For each frequency frame with each threshold (where n is represented as n = L, L-1,…, 3, 2,3,2,1).

4. Encoding Process

The proposed compression algorithm effectively exploits both spatial and spectral correlations inherent in HS images. Spatial correlation is utilized through the BFrWF, while spectral correlation between frames is leveraged by computing the difference between consecutive frames. To enhance efficiency, the algorithm organizes the spectral bands of the HS image into groups, each comprising eight consecutive frequency frames of the HS image. This grouping strategy facilitates the effective exploitation of inter-frequency frame dependencies, contributing to the overall compression performance.

The 2D BFrWF is applied to each frame of the HS image. Subsequently, the HS image cube is modified by grouping eight consecutive frames and processed according to the following sub-steps: The first frame is retained as is, while the second frame is computed as the difference between the second and first frames of the transformed HS image. This differential procedure is repeated for the remaining six frames, from the third to the eighth frame.

This process is iterated across all frames of the HS image to construct the modified HS image cube (MH). Morton mapping (linear indexing) is then employed to convert the MH into a 1D array, followed by the application of 2D-LC-ZM-SPECK to each frame of the HS image. The algorithm operates frame by frame, processing bit planes iteratively until the allocated bit budget is exhausted.

5. Simulation Result

The proposed compression algorithm performance is evaluated using the publicly available hyperspectral dataset ‘Yellowstone’ (named YSUS 0 HS Image is 'Image I', YSUS 0 HS Image is 'Image II', YSUS 3 HS Image is 'Image III', and YSUS 11 HS Image is 'Image IV') on the four different performance metric name as transform memory, transform complexity, Peak to Signal Noise Ratio, SSIM, Coding memory, and coding complexity [29, 68, 69].

For the simulation, the HS image under test is cropped to ‘128 x 128 x 128’. All simulations were carried out on a computing device with 20 GB, with the CPS processing speed of 1.6 GHz on Windows 11 OS using the MATLAB simulation software.

The five-level BFrWF is applied to the spatial dimension of the HS image frame by frame. 1D wavelet transform is applied to the spectral dimension of the HS image. All compression algorithms, apart from FrWF-based 2D-ZM-SPECK [50], used 3D dyadic wavelet transform. The transform HS image coefficients are quantized to the nearest integer. The transformed HS image coefficients are arranged in a 1D array through linear indexing. This 1D array is encoded using the compression algorithms under test 3D-SPECK (CA 1) [51], 3D-LSK (CA 2) [52], 3D-SPIHT (CA 3) [53], 3D-NLS (CA 4) [54], 3D-WBTC (CA 5) [55], 3D-LMBTC (CA 6) [56], 3D-ZM-SPECK (CA 7) [57] and FrWF based 2D-ZM-SPECK (CA 8) [58].

5.1. Transform Memory and Complexity

From Table 2, it is clear that the requirement of transform memory is minimum for the BFrWF. It has been known that BFrWF requires only one input buffer for storing the part of the frequency frame and four other buffers to calculate the four final sub-bands [62]. Let the size of the HS image frequency frame be ‘N’ by ‘N’, and ‘b’ is the number of blocks of the HS image frequency frames. The dimension of the input buffer is $1 \times \frac{N}{b/2}$ while the rest of the buffer dimension is $1 \times \frac{N}{b}$.

It is clear from Table 3 that the requirement of the transform memory is minimum for the BFrWF, while for the other transforms. The 3D-DWT and 2D-DWT have very high transform memory requirements as they need to save the whole HS image (for 3D-DWT) or save the whole frequency frame (for 2D-DWT) [64, 70].

The fractional wavelet filter significantly reduces the need for transform memory, and further, it is reduced by the use of BFrWF to make the transform memory in line with the coding memory, which is near zero [58, 71].

The transform complexity is calculated by the time taken by the wavelet filter to calculate the transform coefficients. It has been clear from Table 3 that BFrWF has lower transform complexity than FrWF, but it has higher complexity than 2D-DWT and its 3D version. With the increase in size of the HS image, the complexity of the BFrWF decreases to the 2D-DWT.

The transform complexity of 3D-DWT is lower than that of the other 2D wavelet transform because 2D-DWT is applied to the 3D HS image one one frequency frame at a time. The complexity of BFrWF is lower due to the calculation of the coefficients in a different way.

5.2. Coding Efficiency

We evaluate our compression algorithm with the other state-of-the-art compression algorithm by calculating the average RD performance (PSNR and SSIM) on the different HS images [72, 73]. It is a quality measurement between the original and the compressed image.

The PSNR is widely used to measure the distortion effect after compression [2]. As a quantitative measure of the subjective quality, the SSIM demonstrates the quality of detailed information, such as the textures and edges of the reconstructed HS image [29, 74]

Table 2. Comparative Analysis between the different types of art wavelet transform with Block-Based Fractional Wavelet Transform on wavelet transform memory requirement

Type of wavelet Transform \ Dimension of HS Image Cube	128	256	512
3D-DWT	38.34 MB	306.72 MB	2453.76 MB
2D-DWT	174.592 KB	698.368 KB	2793.472 KB
FrWF	3.123 KB	6.246 KB	12.493 KB
Block-Based Fractional Wavelet Transform	1.5615 KB	3.123 KB	6.246 KB

Table 3. Average time requirement (for calculation of the transform complexity) of the different wavelet transforms for different HS image sizes

Type of wavelet Transform \ Dimension of HS Image Cube	256	512
3D-DWT	2.96 sec	18.47 sec
2D-DWT	5.65 sec	44.54 sec
FrWF	7.59 sec	47.24 sec
Block-Based Fractional Wavelet Transform (4)	6.01 sec	37.06 sec

Table 4. Structural Similarity (SSIM) index between different HSICAs at fifteen different bit rates for YSUS 0

Image Data Processing Rate	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm
	Image I								
0.00625	0.39	0.38	0.38	0.38	0.39	0.38	0.37	0.41	0.4
0.0125	0.44	0.43	0.43	0.43	0.47	0.43	0.43	0.44	0.44
0.025	0.56	0.55	0.55	0.53	0.57	0.53	0.53	0.61	0.61
0.0375	0.62	0.62	0.61	0.61	0.62	0.61	0.62	0.67	0.66
0.05	0.66	0.65	0.65	0.65	0.66	0.65	0.65	0.71	0.7
0.1	0.75	0.75	0.75	0.74	0.75	0.74	0.75	0.8	0.79
0.2	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.87	0.86
0.3	0.87	0.88	0.87	0.87	0.87	0.87	0.87	0.9	0.89
0.4	0.91	0.9	0.9	0.9	0.9	0.9	0.89	0.92	0.92
0.5	0.93	0.93	0.92	0.92	0.93	0.92	0.92	0.94	0.94
0.6	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.96	0.96
0.7	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.97	0.94
0.8	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.98	0.98
0.9	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.98	0.98
1	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.98	0.98

$$PSNR = 20\ \log_{10}\left[\frac{Max\ \{A(x,y,z)\}}{MSE}\right]$$

$$MSE = \frac{1}{(N\ \times N\ \times N)}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{z=1}^{N}[A(x,y,z) - B(x,y,z)]^2$$

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

The mean and variance of the original HS image A(x,y,z) are defined as μ_A and σ_A^2 while for the reconstructed HS image, it is defined as μ_B and σ_B^2. The covariance between these two HS images σ_{xy} and associated constant is defined as C_1 and C_2 [74].

From Table 4, it is clear that the SSIM of the proposed compression algorithm has higher numeric values than all other compression algorithms except for the FrWF-based 2D-ZM-SPECK [58]. The SSIM values of these two compression algorithms are nearly the same [73].

Table 5 (Appendix) shows that the proposed compression algorithm outperforms the other compression algorithms except for FrWF-based 2D-ZM-SPECK [58], which has nearly the same RD performance. It is due to the large number of insignificant coefficients that are identified and defined in the single bit at early bit plane passes.

5.3. Coding Memory

Coding memory is required by the HS image compression algorithm to keep the tracking record of significant/insignificant coefficients for the bit plane [63]. The proposed compression algorithm only required a fixed amount of memory, which is necessary to save the maximum magnitude of the set in the transform image [28]. The same memory data is shared at the decoder end. This is a very small quantity of memory [75]. Table 6 (Appendix) shows that only two compression algorithms have slightly lower coding memory requisition than the other compression algorithms.

5.4. Coding Complexity

The proposed compression algorithm reduces the coding complexity by storing the maximum magnitude of each set of transform HS image in a fixed memory. FrWF-based ZM-SPECK [40, 41] check the magnitude of each coefficient for every threshold, making it slow with increased complexity. Increasing complexity also increases the demand for power and processing time. The proposed compression algorithm, a modified version of ZM-SPECK, uses the same process but in a different fashion. Tables 7 and 8 (Appendix) show the encoding time and decoding time. The encoding time is the total time needed to encode the coefficients till the bit budget is available. The decoding time is the total time needed for decoding the bit stream received from the encoder end [2]. It has been known that decoding time is always less than the encoding time, as the decoder does not need to check the threshold each bit plane for every coefficient or set [27].

6. Conclusion

This study presents a low-memory, low-complexity lossy hyperspectral image compression algorithm designed for resource-constrained HS image sensors. The proposed compression algorithm minimizes multiple memory accesses, thereby enhancing its computational speed. By leveraging the inherent correlation within HS images, the algorithm achieves high coding efficiency. The Block-based Fractional Wavelet Transform reduces transform memory requirements, while the 2D-LC-ZM-SPECK [50] eliminates coding memory to reduce coding complexity, resulting in a faster compression process compared to other transform-based compression algorithms. Although the proposed compression algorithm operates in a listless manner, it requires slightly more computational time than existing listless compression algorithms. This is attributed to its frame-by-frame search for significant coefficients, which introduces a marginal increase in algorithmic complexity.

Acknowledgements

The manuscript bears communication number IU/R&D/2025-MCN0003808 from Integral University, Lucknow, India.

References

[1] B. Krishna Mohan, and Alok Porwal, “Hyperspectral Image Processing and Analysis,” *Current Science*, vol. 108, no. 5, pp. 833-841, 2015. [Google Scholar] [Publisher Link]

[2] Huiwen Yu et al., “Hyperspectral Imaging Techniques for Lyophilization: Advances in Data-Driven Modeling Strategies and Applications,”*Advanced Science*, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Nafiseh Ghasemi et al, “Onboard Processing of Hyperspectral Imagery: Deep Learning Advancements, Methodologies, Challenges, and Emerging Trends,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 4780-4790, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Vijay Joshi, and J. Sheeba Rani, “An On-Board Satellite Multispectral and Hyperspectral Compressor (MHyC): An Efficient Architecture of a Simple Lossless Algorithm,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 72, no. 5, pp. 2167-2177, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Vijay Joshi, and J. Sheeba Rani, “A Simple Lossless Algorithm for On-Board Satellite Hyperspectral Data Compression,” *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Claire Guilloteau et al., “Hyperspectral and Multispectral Image Fusion under Spectrally Varying Spatial Blurs – Application to High Dimensional Infrared Astronomical Imaging,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1362-1374, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sravan Sikhakolli et al., “Effect of Filtering Techniques in Biomedical Hyperspectral Microscopic Images,” *2024 4th International Conference on Intelligent Technologies (CONIT)*, Bangalore, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Manoj Kaushik, Rama Rao Nidamanuri, and B. Aparna, “Hyperspectral Discrimination of Vegetable Crops Grown under Organic and Conventional Cultivation Practices: A Machine Learning Approach,” *Scientific Reports*, vol. 15, pp. 1-12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Vusal I. Pasha, and Dalila B. Megherbi, “A Deep Learning Approach for Hyperspectral Image Classification with Additive Noise for Remote Sensing and Airborne Surveillance,” *2022 IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Chemnitz, Germany, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[10] Roozbeh Rajabi et al., “Hyperspectral Imaging in Environmental Monitoring and Analysis,” *Frontiers in Environmental Science*, vol. 11, pp. 1-2, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[11] Dharmendrakumar Patel et al., “Non-Destructive Hyperspectral Imaging Technology to Assess the Quality and Safety of Food: A Review,” *Food Production, Processing and Nutrition*, vol. 6, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[12] Jyoti Seth, R. Kavitha, and Dhyan Chandra Yadav, “Automating Optimal Healthcare Delivery through Hyper Spectral Imaging in Sustainable Medical Environments,” *2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC)*, Debre Tabor, Ethiopia, pp. 1-9, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[13] Sima Peyghambari, and Yun Zhang, “Hyperspectral Remote Sensing in Lithological Mapping, Mineral Exploration, and Environmental Geology: An Updated Review,” *Journal of Applied Remote Sensing*, vol. 15, no. 3, pp. 1-25, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[14] K. Deepthi, Aditya K. Shastry, and E. Naresh, “A Novel Deep Unsupervised Approach for Super-Resolution of Remote Sensing Hyperspectral Image using Gompertz-Function Convergence War Accelerometric-Optimization Generative Adversarial Network (GF-CWAO-GAN),” *Scientific Reports*, vol. 14, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[15] A. Nisha, and A. Anitha, “Current Advances in Hyperspectral Remote Sensing in Urban Planning,” *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, Kannur, India, pp. 94-98, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[16] Helge Bürsing, and Wolfgang Gross, “Hyperspectral Imaging: Future Applications in Security Systems,” *Advanced Optical Technologies*, vol. 6, no. 2, pp. 61-66, 2017. [CrossRef] [Google Scholar] [Publisher Link]
[17] Mohammed Abdulmajeed Moharram, and Divya Meena Sundaram, “MultiGO: An Unsupervised Approach Based on Multi-Objective Growth Optimizer for Hyperspectral Image Band Selection,” *Remote Sensing Applications: Society and Environment*, vol. 37, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[18] Fenlong Jiang et al., “Adaptive Center-Focused Hybrid Attention Network for Change Detection in Hyperspectral Images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[19] Ajay Kaul, and Sneha Raina, “Support Vector Machine Versus Convolutional Neural Network for Hyperspectral Image Classification: A Systematic Review,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 15, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[20] Nan Zhao et al., “The Paradigm Shift in Hyperspectral Image Compression: A Neural Video Representation Methodology,” *Remote Sensing*, vol. 17, no. 4, pp. 1-20, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[21] Urvashi Mesariya et al., “Denoising Lunar Hyperspectral Images Using CNN with Skip Connections,” *2024 IEEE Space, Aerospace and Defence Conference (SPACE)*, Bangalore, India, pp. 398-401, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[22] Radhesyam Vaddi et al., “Strategies for Dimensionality Reduction in Hyperspectral Remote Sensing: A Comprehensive Overview,” *The Egyptian Journal of Remote Sensing and Space Sciences*, vol. 27, no. 1, pp. 82-92, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[23] 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]
[24] Reaya Grewal, Singara Singh Kasana, and Geeta Kasana, “Hyperspectral Image Segmentation: A Comprehensive Survey,” *Multimedia Tools and Applications*, vol. 82, pp. 20819-20872, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[25] 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]
[26] Shrish Bajpai, “Low Complexity Block Tree Coding for Hyperspectral Image Sensors,” *Multimedia Tools and Applications*, vol. 81, pp. 33205-33232, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[27] Kasper Skog et al., “Lossless Hyperspectral Image Compression in Comet Interceptor and Hera Missions with Restricted Bandwith,” *Remote Sensing*, vol.17, no. 5, pp. 1-18, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[28] 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]
[29] 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]
[30] Qizhi Fang et al., “ARM-Net: A Tri-Phase Integrated Network for Hyperspectral Image Compression,” *Sensors*, vol. 25, no. 6, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[31] Yaman Dua, Vinod Kumar, and Ravi Shankar Singh, "Comprehensive Review of Hyperspectral Image Compression Algorithms," *Optical Engineering*, vol. 59, no. 9, pp. 1-39, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] 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]

[33] Diego Valsesia, Tiziano Bianchi, and Enrico Magli, "Onboard Deep Lossless and Near-Lossless Predictive Coding of Hyperspectral Images With Line-Based Attention," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] 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]

[35] 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]

[36] 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]

[37] Chengfu Huo, Rong Zhang, and Dong Yin, "Compression Technique for Compressed Sensing Hyperspectral Images," *International Journal of Remote Sensing*, vol. 33, no. 5, pp. 1586-1604, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[38] Azam Karami, Soosan Beheshti, and Mehran Yazdi, "Hyperspectral Image Compression using 3D Discrete Cosine Transform and Support Vector Machine Learning," *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, Montreal, QC, Canada, pp. 809-812, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[39] Bart Beusen et al., "On-Board Hyperspectral Image Compression Using Vector-Quantized Auto Encoders," *IEEE International Geoscience and Remote Sensing Symposium*, Athens, Greece, pp. 1703-1707, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[40] 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]

[41] 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]

[42] 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]

[43] 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]

[44] Rui Dusselaar, and Manoranjan Paul, "Hyperspectral Image Compression Approaches: Opportunities, Challenges, and Future Directions: Discussion," *Journal of the Optical Society of America A*, vol. 34, no. 12, pp. 2170-2180, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[45] 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]

[46] Mohd Tausif et al., "SMFrWF: Segmented Modified Fractional Wavelet Filter: Fast Low-Memory Discrete Wavelet Transform (DWT)," *IEEE Access*, vol. 7, pp. 84448-84467, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[47] Mohd Tausif et al., "Memory-Efficient Architecture for Frwf-Based DWT of High-Resolution Images for IoMT Applications," *Multimedia Tools and Applications*, vol. 80, pp. 11177-11199, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[48] Mohd Tausif et al., "SFrWF: Segmented Fractional Wavelet Filter Based Dwt for Low Memory Image Coders," *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics*, Mathura, India, pp. 593-597, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[49] Mohd Tausif, Ekram Khan, and Mohd Hasan, "BFrWF: Block-Based FrWF for Coding of High-Resolution Images with Memory-Complexity Constrained –Devices," *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Gorakhpur, India, pp. 1-5, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[50] 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]

[51] Xiaoli Tang, and William A. Pearlman, *Three-Dimensional Wavelet-Based Compression of Hyperspectral Images*, 1st ed., Hyperspectral Data Compression, Springer, Boston, MA, pp. 273-308, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[52] 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]

[53] 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, pp. 3283-3286, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[54] V.K. Sudha, and R. Sudhakar, “3D Listless Embedded Block Coding Algorithm for Compression of Volumetric Medical Images,” *Journal of Scientific and Industrial Research*, vol. 72, pp. 735-748, 2013. [Google Scholar]

[55] 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]

[56] 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]

[57] 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, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[58] 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]

[59] 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. [CrossRef] [Google Scholar] [Publisher Link]

[60] Ying Hou, and Guizhong Liu, “3D Set Partitioned Embedded Zero Block Coding Algorithm for Hyperspectral Image Compression,” *Proceedings Volume 6790, MIPPR 2007: Remote Sensing and GIS Data Processing and Applications; and Innovative Multispectral Technology and Applications*, Wuhan, China, vol. 6790, pp. 1339-1345, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[61] A. Karami, “Lossy Compression of Hyperspectral Images using Shearlet Transform and 3D SPECK,” *Proceedings Image and Signal Processing for Remote Sensing XXI*, vol. 9643, pp. 525-530, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[62] Yuanyuan Guo, Yanwen Chong, and Shaoming Pan, “Hyperspectral Image Compression via Cross-Channel Contrastive Learning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[63] K. Subhash Babu et al., “Hyperspectral Image Compression Algorithms-A Review,” *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014*, pp. 127-138, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[64] Raúl Guerra et al., “A New Algorithm for the On-Board Compression of Hyperspectral Images,” *Remote Sensing*, vol. 10, no. 3, pp. 1-41, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[65] Zhong Cuixiang, and Huaung Minghe, “An Effective Improvement on 3D SPIHT,” *2012 International Conference on Image Analysis and Signal Processing*, Huangzhou, China, pp. 1-4, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[66] Xiaoying Song et al., “Three-Dimensional Separate Descendant-Based SPIHT Algorithm for Fast Compression of High-Resolution Medical Image Sequences,” *IET Image Processing*, vol. 11, no. 1, pp. 80-87, 2017. [CrossRef] [Google Scholar] [Publisher Link]

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

[68] Mohd Tausif, and Ekram Khan, “Image Coding of Natural and Light Field Images: A Tutorial,” *IETE Journal of Education*, vol. 66, no. 1, pp. 25-34, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[69] Rajat Kumar Arya, Pratik Chattopadhyay, and Rajeev Srivastava, “Hyperspectral Image Classification Using Gated Adaptable Convolutional-Based Kolmogorov–Arnold Network,” *IEEE Geoscience and Remote Sensing Letters*, vol. 22, pp. 1-5, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[70] Divya Sharma, Jitendra Bahadur Maurya, and Yogendra Kumar Prajapati, “Effect of Noise on Constellation Diagram of 100 GBPS DP-QPSK Systems under Influence of Different Digital Filters,” *2015 International Conference on Microwave and Photonics (ICMAP)*, Dhanbad, India, pp. 1-2, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[71] 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]

[72] 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]

[73] 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]

[74] Dr. R. Nagendran et al., “Neural Reinforcement-Oriented Hyperspectral Image Compression: Adaptive Approaches for Enhanced Quality,” *Chemometrics and Intelligent Laboratory Systems*, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[75] Amal Altamimi, and Belgacem Ben Youssef, “Hardware Acceleration of Division-Free Quadrature-Based Square Rooting Approach for Near-Lossless Compression of Hyperspectral Images,” *Sensors*, vol. 25, no. 4, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[76] Amal Altamimi, and Belgacem Ben Youssef, “Leveraging Seed Generation for Efficient Hardware Acceleration of Lossless Compression of Remotely Sensed Hyperspectral Images,” *Electronics*, vol. 13, no. 11, pp. 1-22, 2024. [CrossRef] [Google Scholar] [Publisher Link]

Appendix

Table 5. Numeric values of PSNR for the proposed compression algorithm with the eight other algorithms on four different HS image datasets

Image Data Processing Rate	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm
	Image I									**Image II**								
0.00625	26.4	26.9	26.3	26.3	26.4	26.9	26.9	30.2	30.1	26.5	26.2	26.2	26.5	26.5	26.1	26.1	28.9	28.8
0.0125	27.6	27.5	27.4	27.4	27.6	27.5	27.5	31.5	31.4	27.6	27.5	27.5	27.4	27.5	27.5	27.5	29.6	29.6
0.025	29	28.4	28.8	28.3	29.1	28.3	28.4	32.8	32.7	28.5	28.2	28.2	28.2	28.7	28.2	28.3	30.8	30.7
0.0375	30	29.9	29.8	29.8	29.9	30	30	34.2	34.2	29.4	29.3	29.3	29.1	29.3	29.4	29.4	31.6	31.6
0.05	30.9	30.5	30.7	30.6	30.8	30.5	30.5	34.7	34.7	30.0	29.8	29.8	29.8	30	29.8	29.8	32.5	32.5
0.1	32.8	32.8	32.7	32.7	32.8	32.8	32.8	37	37	31.5	31.4	31.4	31.3	31.5	31.4	31.5	34.1	34
0.2	35.1	34.6	35.1	34.5	35.2	34.6	34.6	39.8	39.7	33.4	33.2	33.2	33.2	33.5	33.1	33.2	36.7	36.7
0.3	36.9	36.7	36.8	36.7	36.9	36.6	36.6	41.5	41.5	34.9	34.7	34.7	34.5	34.9	34.5	34.5	38.8	38.8
0.4	38.4	37.6	38.2	37.8	38.3	37.5	37.5	43.3	43.3	36.1	35.9	35.9	36	36.1	35.9	35.9	40.3	40.3
0.5	39.5	39.5	39.4	39.3	39.5	39.1	39.1	45	45	37.3	36.8	36.8	37	37.3	36.7	36.7	41.7	41.6
0.6	40.8	40.6	40.7	40.6	40.8	40.5	40.5	45.7	45.7	38.5	37.9	37.9	38	38.5	37.8	37.9	43.6	43.6
0.7	41.9	41.3	41.8	41.8	41.9	41.2	41.2	47	46.9	39.5	39.5	39.5	39.4	39.5	39.3	39.3	44.3	44.3
0.8	43	42.3	42.9	42.5	43.0	42.1	42.1	48.1	48	40.5	40.3	40.3	40.3	40.5	40.3	40.3	45	44.9
0.9	44	43.8	43.8	43.5	44.0	43.0	43	49.4	49.4	41.6	41.1	41.1	41.3	41.6	40.9	40.9	45.8	45.8
1	44.9	44.9	44.7	44.7	44.8	44.8	44.8	49.9	49.8	42.6	42.1	42.1	42.3	42.6	41.7	41.7	46.8	46.7
	Image III									**Image IV**								
0.00625	30.5	30.5	30.3	29.8	30.5	30.5	30.5	32.4	32.3	30.4	30.4	30.2	30.2	30.4	30.3	30.4	31.9	31.9
0.0125	31.4	31.3	31.3	31.2	31.4	31.3	31.3	33.9	33.8	31.7	31.6	31.6	31.5	31.7	31.5	31.5	33.1	33
0.025	32.9	32.9	32.7	32.6	32.9	32.9	33	34.1	34	33	33	32.9	32.7	33	32.9	33	34.4	34.3
0.0375	34.0	33.8	33.8	33.8	34	33.8	33.8	36.7	36.6	34.1	33.9	33.9	33.8	34.1	33.8	33.8	35.8	35.7
0.05	34.9	34.7	34.6	34.4	34.8	34.6	34.7	37.9	37.9	34.9	34.4	34.6	34.5	34.8	34.4	34.4	36.7	36.7
0.1	37.2	37.1	37	37	37.2	36.9	36.9	41	41	36.9	36.6	36.7	36.7	36.9	36.5	36.6	39.1	39
0.2	40.4	40.1	40.2	40.1	40.4	40.1	40.1	44.2	44.1	38.9	38.8	38.8	38.8	38.9	38.8	38.8	42.2	42.1
0.3	42.9	42.8	42.7	42.4	42.9	42.4	42.5	46.9	46.8	40.5	40.1	40.4	40.2	40.5	40	40	44	44
0.4	45	44.6	44.7	44.6	45	44.5	44.5	49.2	49.2	41.9	41.7	41.8	41.5	41.9	41.3	41.4	46	45.9

0.5	47	46.7	46.8	46.3	47	46.1	46.2	51.5	51.5	43.2	43.1	43.1	43	43.2	43	43.0	47.7	47.7
0.6	48.9	48.9	48.5	48.5	48.9	48.9	48.9	53.7	53.6	44.5	43.9	44.3	44.3	44.4	43.8	43.8	48.6	48.6
0.7	50.7	50.2	50.3	50.2	50.7	50	50	54.8	54.8	45.6	45.1	45.4	45.1	45.7	44.8	44.8	50.1	50.1
0.8	52.1	52.2	51.8	51.7	52	51.4	51.4	56.4	56.3	46.7	46.7	46.6	46.5	46.7	46.1	46.1	51.5	51.5
0.9	53.8	53.8	53.4	53.4	53.8	53.8	53.8	58.5	58.5	47.8	47.8	47.7	47.7	47.8	47.8	47.8	52.1	52.1
1	55.4	55.1	55.0	54.9	55.3	54.9	54.9	59.9	59.8	48.8	48.5	48.6	48.5	48.8	48.3	48.3	53.1	53.1

Table 6. Coding memory requirement of nine compression algorithms for four different HS image datasets

Image Data Processing Rate	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm
	Image I									**Image II**								
0.00625	18	512	15	1024	8	12	0	0	0.018	19	512	15	1024	11	12	0	0	0.018
0.0125	31	512	36	1024	23	12	0	0	0.018	33	512	37	1024	34	12	0	0	0.018
0.025	62	512	71	1024	53	12	0	0	0.018	65	512	71	1024	64	12	0	0	0.018
0.0375	109	512	111	1024	100	12	0	0	0.018	112	512	116	1024	101	12	0	0	0.018
0.05	109	512	117	1024	108	12	0	0	0.018	163	512	123	1024	113	12	0	0	0.018
0.1	237	512	249	1024	208	12	0	0	0.018	276	512	289	1024	282	12	0	0	0.018
0.2	494	512	513	1024	489	12	0	0	0.018	448	512	489	1024	446	12	0	0	0.018
0.3	597	512	615	1024	578	12	0	0	0.018	822	512	833	1024	826	12	0	0	0.018
0.4	843	512	896	1024	844	12	0	0	0.018	920	512	881	1024	891	12	0	0	0.018
0.5	1224	512	1222	1024	1184	12	0	0	0.018	1038	512	1055	1024	1037	12	0	0	0.018
0.6	1329	512	1354	1024	1230	12	0	0	0.018	1352	512	1389	1024	1355	12	0	0	0.018

0.7	1456	512	1461	1024	1356	12	0	0	0.018	1634	512	1561	1024	1592	12	0	0	0.018
0.8	1653	512	1680	1024	1554	12	0	0	0.018	1681	512	1652	1024	1634	12	0	0	0.018
0.9	1892	512	1906	1024	1876	12	0	0	0.018	1741	512	1710	1024	1712	12	0	0	0.018
1	2010	512	2072	1024	2022	12	0	0	0.018	1825	512	1813	1024	1824	12	0	0	0.018
	Image III									**Image IV**								
0.00625	16	512	17	1024	8	12	0	0	0.018	15	512	16	1024	11	12	0	0	0.018
0.0125	26	512	28	1024	20	12	0	0	0.018	26	512	29	1024	27	12	0	0	0.018
0.025	71	512	73	1024	63	12	0	0	0.018	69	512	74	1024	71	12	0	0	0.018
0.0375	78	512	81	1024	72	12	0	0	0.018	79	512	87	1024	80	12	0	0	0.018
0.05	117	512	124	1024	109	12	0	0	0.018	112	512	121	1024	112	12	0	0	0.018
0.1	205	512	204	1024	186	12	0	0	0.018	196	512	198	1024	196	12	0	0	0.018
0.2	397	512	414	1024	378	12	0	0	0.018	467	512	486	1024	469	12	0	0	0.018
0.3	678	512	682	1024	621	12	0	0	0.018	634	512	669	1024	633	12	0	0	0.018
0.4	726	512	753	1024	708	12	0	0	0.018	993	512	1013	1024	998	12	0	0	0.018
0.5	947	512	947	1024	894	12	0	0	0.018	1074	512	1102	1024	1076	12	0	0	0.018
0.6	1119	512	1155	1024	1091	12	0	0	0.018	1112	512	1102	1024	1112	12	0	0	0.018
0.7	1234	512	1243	1024	1121	12	0	0	0.018	1409	512	1431	1024	1407	12	0	0	0.018
0.8	1343	512	1324	1024	1282	12	0	0	0.018	1564	512	1560	1024	1547	12	0	0	0.018
0.9	1439	512	1473	1024	1400	12	0	0	0.018	1689	512	1678	1024	1640	12	0	0	0.018
1	1498	512	1517	1024	1440	12	0	0	0.018	1789	512	1821	1024	1790	12	0	0	0.018

Table 7. Encoding time (sec) requirement for the complexity analysis of nine different compression algorithms for four different datasets

Image Data Processing Rate	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm
	Image I									**Image II**								
0.00625	1.1	0.2	1.3	0.3	1.1	0.3	0.6	0.6	0.5	1.8	0.2	0.6	0.4	0.8	0.9	0.4	0.6	0.5
0.0125	1.4	0.4	1.1	0.6	1.3	1.2	0.7	0.7	0.6	2.5	0.3	1.1	0.5	1.4	1.2	0.7	0.9	0.8
0.025	2.8	0.5	1.5	0.7	1.7	1.5	0.9	0.8	0.7	5.1	0.4	1.6	0.6	1.7	1.5	0.9	1.1	1
0.0375	4.3	0.6	2.2	0.8	2.2	1.7	1	1.1	0.9	5.6	0.5	2.2	0.7	2.1	1.7	1.1	1.3	1.1
0.05	7.3	0.7	3.4	0.9	3.5	2	1.2	1.3	1	8.5	0.6	3.3	0.8	3.5	2.1	1.2	1.4	1.2
0.1	18.5	0.9	6.7	1	6.2	3.1	1.8	2.1	1.6	18.8	0.7	8.1	0.9	5.9	3.2	1.7	2.2	1.8
0.2	89.7	1.1	31	1.4	17.6	4.9	2.8	3.7	2.4	83.5	1.1	28.9	1.3	23.6	5.3	3.7	3.7	3.6
0.3	195	1.5	67	1.8	60.7	7.1	3.9	5.4	4.1	110	1.3	50.3	1.7	32.5	6.6	3.9	5.3	4.2
0.4	249	1.8	96	2.1	118	8.6	4.9	7	5.3	326	1.7	181	2.0	199	9.2	5.3	7.2	5.5
0.5	340	2.1	118	2.4	173	10	5.8	8.4	7.2	472	2	241	2.3	294	11.5	7.2	10.8	8.9
0.6	692	2.6	257	2.9	376	13	7.1	9.9	7.9	588	2.2	264	2.5	363	11.3	8.4	11.4	9.5
0.7	961	2.8	463	3.2	657	14.2	8.1	11.6	8.9	678	2.5	281	2.7	398	12.5	7.8	12.6	10.6
0.8	1167	3.0	491	3.5	759	15.8	9.2	13.4	11.2	1151	2.9	531	3.2	772	15.8	10	15.2	12.2
0.9	1304	3.3	513	4.1	855	16.8	9.9	15.3	12.9	1745	3.1	883	3.4	1223	17.1	12	17.6	14.8
1	1441	3.8	560	4.9	878	17.4	10.9	20.6	16.6	2558	3.3	1034	3.9	1517	18.3	13.7	18.5	15.8
	Image III									**Image IV**								
0.00625	0.9	0.2	1.0	0.4	0.5	0.4	0.4	0.9	0.7	1.0	0.2	0.7	0.3	0.6	0.9	0.5	0.5	0.4
0.0125	1.6	0.3	1.1	0.6	1.3	1.1	0.8	1.7	0.9	1.6	0.3	1.0	0.4	2.3	1.8	0.9	0.6	0.5
0.025	2.9	0.4	1.6	0.7	1.8	1.4	0.9	2.1	1.2	3.0	0.4	1.5	0.5	3.1	1.7	0.9	0.8	0.6
0.0375	5.0	0.5	2.2	0.8	2.4	1.7	1.3	2.2	1.6	5.5	0.5	2.1	0.6	6.0	2.1	1.2	1.1	1
0.05	6.7	0.6	4.3	0.9	2.7	2.1	1.4	2.4	1.9	7.4	0.6	2.9	0.7	15.4	2.5	1.3	1.3	1.1
0.1	18.2	0.8	9.9	1.4	7.1	3.2	1.9	3.3	2.2	19.7	0.7	7.8	0.9	17.8	3.6	2.1	2.1	1.9
0.2	57.3	1.2	24.3	1.9	25.0	5.5	3	5.1	3.7	81.4	1.1	21.3	1.3	29.1	6.7	3.2	4.4	4.1
0.3	97.1	1.4	39.9	2.1	47.9	6.7	3.9	6.8	4.6	142	1.4	67.0	1.7	70.6	7.7	4.1	6.7	5.3

0.4	206	1.8	104	2.4	122	9.3	5.1	8.4	6.8	227	1.8	77.7	2.0	99.8	9.3	4.9	7.8	6.6
0.5	303	2.2	131	2.9	177	10.5	6.1	10.1	7.5	509	2.1	227	2.4	280	12.2	7.2	9.3	8.1
0.6	349	2.5	141	3.2	190	11.5	7.2	11.8	10.1	755	2.3	405	2.8	495	14.2	7.6	11.1	9.4
0.7	706	2.7	338	3.5	528	14.3	8.6	13.5	11.5	981	2.5	424	3.0	578	14.5	8.5	12.2	10.6
0.8	835	3.0	408	3.8	601	15.7	9.5	15.2	12.8	1018	2.8	449	3.2	646	14.7	9.1	16.5	11.3
0.9	981	3.5	414	4.5	633	16.8	10.7	16.9	14.1	1329	3.3	447	3.5	738	17.9	10.2	18.2	14.9
1	1373	3.8	632	5.1	977	19.9	11.9	18.3	16.3	2073	3.6	873	4.1	1291	19.9	11.4	24.3	16.3

Table 8. Decoding time (sec) requirement for the complexity analysis of nine different compression algorithms for four different datasets

Image Data Processing Rate	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm	CA 1 [51]	CA 2 [52]	CA 3 [53]	CA 4 [54]	CA 5 [55]	CA 6 [56]	CA 7 [57]	CA 8 [58]	Proposed Algorithm
	Image I									**Image II**								
0.00625	0.7	0.1	0.9	0.2	0.7	0.3	0.36	0.46	0.33	0.8	0.1	0.3	0.3	0.2	0.5	0.3	0.4	0.4
0.0125	0.7	0.3	0.3	0.4	0.4	0.5	0.7	0.5	0.4	1.9	0.2	0.3	0.4	0.4	0.6	0.6	0.6	0.6
0.025	1.5	0.4	0.7	0.8	0.7	0.7	0.8	0.7	0.6	2.3	0.3	0.7	0.5	0.7	0.7	0.8	0.8	0.7
0.0375	2.8	0.4	1.2	0.6	0.9	0.8	1	1	0.8	2.8	0.4	1.2	0.6	0.9	0.8	1	1	0.9
0.05	5.1	0.5	2.2	0.6	2.3	1	1.1	1.3	0.9	5.9	0.5	2.2	0.7	2.3	1	1.2	1.1	1
0.1	12.9	0.7	5.3	0.8	4.6	1.8	1.5	1.8	1.5	12.8	0.6	6.5	0.8	4.4	1.8	1.6	1.9	1.7
0.2	61.4	1.0	23.9	1.2	15.4	3.4	2.7	3.2	2.6	62.2	1	25.8	1.1	21.6	3.3	2.8	3.2	3.1
0.3	136	1.1	52.4	1.6	57.9	4.7	3.5	5	3.3	80.0	1.2	47.3	1.5	29.2	4.9	4	5.1	3.9
0.4	204	1.7	87.9	1.9	108	6.2	4.3	6.5	4.9	307	1.4	172	1.9	175	6.7	5.5	7	5.9
0.5	306	1.9	98.9	2.2	148	7.8	5.2	8	6.6	441	1.7	219	2.1	262	7.6	7.8	9.6	7.5
0.6	600	2.2	217	2.5	341	9.2	6.7	9.3	7.1	555	1.9	230	2.2	319	9.0	6.6	10.1	8.1
0.7	906	2.4	456	2.7	618	10.4	7.7	11.1	8.2	609	2.2	246	2.5	359	10.5	7.5	12	9.6
0.8	1005	2.5	486	3	727	13.8	8.5	12.5	10.4	1080	2.5	496	2.8	717	11.8	9.2	14.3	11.4
0.9	1249	3.1	501	3.3	830	13.3	9.4	14.5	12.5	1692	2.9	818	3.1	1130	13.3	10	16.5	12.9
1	1389	3.6	512	4	832	14.5	10.6	19	15.1	2179	3.4	930	3.4	1384	14.7	11.5	17.2	14.8
	Image III									**Image IV**								
0.00625	0.8	0.5	0.7	0.3	0.2	0.3	0.3	0.5	0.4	0.66	0.2	0.2	0.2	0.2	0.5	0.4	0.4	0.3

0.0125	0.9	0.3	0.8	0.5	0.4	0.5	0.6	0.7	0.7	0.93	0.3	0.3	0.4	0.8	0.6	0.7	0.5	0.4
0.025	1.6	0.4	1.2	0.6	0.6	0.7	0.8	0.9	0.8	1.63	0.4	0.6	0.4	0.9	0.8	0.8	0.7	0.6
0.0375	3.5	0.4	1.5	0.7	1.2	0.9	1.1	1.1	1	3.74	0.4	1.2	0.5	3.3	1.2	1.1	0.9	0.8
0.05	4.6	0.5	2.7	0.8	1.6	1.1	1.2	1.4	1.2	4.9	0.5	1.9	0.6	4.1	1.7	1.2	1.1	1
0.1	14.5	0.7	6.6	1.1	9.5	1.9	1.5	2.3	1.9	15.8	0.7	6.5	0.8	11.5	2.3	1.9	1.9	1.5
0.2	49.4	1	22.3	1.5	17.6	3.5	2.6	4.1	3.2	61.1	1.0	16.2	1.1	20.9	3.9	2.8	4.1	3.5
0.3	81.9	1.4	31.6	1.8	40.1	5.1	3.5	5.8	4.2	118	1.3	56.1	1.6	54.1	5.7	3.9	5.5	4.6
0.4	190	1.7	90.4	2.2	110	6.7	4.8	7.6	6.5	195	1.7	70.2	1.9	89.4	7.3	4.8	6.5	5.9
0.5	284	2.0	118	2.5	156	8.1	5.9	9.4	6.9	400	2.0	216	2.2	250	9.1	6.1	8	6.7
0.6	321	2.4	128	2.9	176	9.4	6.9	11.1	8.6	847	2.2	394	2.5	471	11.3	7.3	10.2	8.9
0.7	684	2.5	309	3.2	577	11.0	8.3	13	10.2	840	2.3	419	2.9	575	12.6	8.2	11.8	9.8
0.8	822	2.8	373	3.4	553	12.5	9.3	14.7	11.3	1059	2.6	436	3.1	621	13.4	8.7	15.6	10.6
0.9	860	3.2	400	3.8	589	13.9	10.3	16.2	14.1	1186	2.9	440	3.3	706	14.2	9.5	17.8	12.3
1	1258	3.4	589	4	849	15.6	11.5	18.1	15.7	1901	3.4	867	3.9	1260	17.4	10.6	20.9	14.6

Table 9. Requirement of coding memory for different transform-based compression algorithms at three HS image sizes (in KB)

3D-LSK	**3D-NLS**	**3D-LMBTC**	**3D-ZM-SPECK**	**3D-BP-ZM-SPECK**	**3D-M-ZM-SPECK**	**3D-LCBTC**	**3D-LEZ SPC**	**3D-MELS**	**3D-LMZC**	**FrWF based ZM-SPECK**	**Proposed Compression Algorithm**
[52]	[54]	[56]	[57]	[32]	[28]	[26]	[40]	[27]	[30]	[58]	
512	1024	12	0	0	0	300.5	-	128	-	0	18.25
4096	8192	96	0	0	0	2318	2304	1024	2176	0	146
32768	65568	768	0	0	0	18544	-	8192	17408	0	1168