SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I6P106

Volume 12 Issue 6, 65-89, June 2025
© 2025 Seventh Sense Research Group®

Original Article

Segmented Fractional Wavelet Filter Based Low Memory Hyperspectral Image Coding Algorithm for Wireless Multimedia Sensor Networks

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: 03 April 2025 Revised: 05 May 2025 Accepted: 06 June 2025 Published: 27 June 2025

Abstract - *Recently, the rapid development of transform-based coding has improved the performance of hyperspectral image sensors. Each hyperspectral image has a size of more than a hundred MB, which needs to be compressed before the transmission to save the data transmission bandwidth, reduce the sensor complexity, increase the sensor performance, and lower the energy consumption. This work proposed a novel lossy hyperspectral image compression algorithm, which has high coding efficiency, decreases coding complexity and reduces overall memory (coding and transform) demand. The Segmented Fractional Wavelet Filter (SFrWF) is a low-memory mathematical transform approach to compute the transform coefficient of the hyperspectral image. The Low Complexity Zero Memory Set Partitioned Embedded bloCK (LC-ZM-SPECK) is employed to code coefficients of the transform hyperspectral image, which is applied on a frame-by-frame basis. The simulation result shows that the proposed compression algorithm is faster than other state-of-the-art compression algorithms, making it an appropriate choice for low hardware resource hyperspectral image sensors.*

Keywords - *Sensor performance, Transform memory, Transform coding, Hyperspectral image compression, Set partitioned algorithm.*

1. Introduction

A HyperSpectral (HS) image includes a hundred to thousands of frequency frames with rich spectral and spatial information for a single scene [1-2]. Because the HS image contains hundreds of pieces of reflection information at various frequencies for each pixel [3], it is well suited for a wide variety of practical applications such as agriculture production (crops variability) [4], archaeological analyses [5], biomedicine [6], cultural heritage [7], disaster prediction & assessment [8], food quality assessment [9], forestry (vegetation coverage) [10], forensic research [11], map revision (land mapping) [12], maritime resources management [13], medical diagnosis [14], meteorological forecasting [15], military target detection [16], mining (identification of minerals) [17], oil & gas exploration [18], pharmaceutics [19], urban planning [20], verification of documents [21], water quality analysis [22] etc. Besides the mentioned applications, remote sensing (earth observation) is one of the growing applications of HS images where the researchers develop the computer-based algorithm associated with the HS image classification [23], HS image compression [24], HS image change detection [25], HS image feature extraction [26], HS image object detection [27], HS image segmentation [28], target identification [29], denoising [30] etc.

Unlike color images that record R, G and B channels per pixel, hyperspectral images record hundreds of channels per pixel, representing pixel's spectra [31]. This would imply that HS images demand storage space that is two orders of magnitude greater than required to store colour or multispectral images of a comparable size [32]. Thus, an efficient algorithm for capturing, storing, transmitting, and analyzing HS images is required [33].

The large redundant information in HS image data must be removed to achieve a satisfactory HS image compression performance. A very high spectral correlation exists between the nearby (adjacent) frequency frames, while the spatial correlation presents between the neighbourhood pixels in the same frequency frame [34, 35]. These correlations are exploited by the HyperSpectral Image Compression Algorithm (HSICA) to reduce the redundancy in HS images [36]. The main objective of any compression algorithm is to reduce the image data for the reconstruction of the HS image after the compression process and improve the image sensor's performance by reducing the processing time and power computation [37, 38]. The low complexity of the compression algorithm requires minimum data processing time and power computation [39].

In the subsequent sections, Section 2 presents a literature review by other researchers on the different compression algorithms for HS images. At the same time, related work is covered in the next section of the manuscript. The proposed HSICA is covered in detail with the associated pseudocode and flow figure (section 4). The experimental result and comparative study with state of art HSICA are covered in Section 5. Finally, Section 6 entails the concluding remarks of this work.

2. Literature Review

In the past two decades, many Hyperspectral Image Compression Algorithm (HSICA) have been proposed. They are broadly classified by the way the compression algorithm operates (predictive coding, vector quantization-based coding, transform coding, compressive sensing, tensor decomposition, leaning-based coding or hybrid coding algorithms) or based on data loss (lossless, lossy and near-lossless) [36].

The lossy, lossless or near-lossless compression algorithm is split based on the compression ratio or by the loss of the data [40]. The Compression Ratio (CR) is the original HS image's size compared to the reconstructed HS image [41]. In the lossless HS image compression, there is no image data loss, and the reconstructed HS image is similar to the original HS image. The value of CR is very low and almost near one to three [42]. The near-lossless HS image compression has caused the loss of some HS image data, but the reconstruction of the HS image is almost similar to the original HS image. Such HS image degradations are nearly invisible to human observers. These compression algorithms have a slightly higher CR than lossless HS image compression algorithms [43]. Lossy HS image compression algorithms have a loss of significant HS image data, but it has high CR [44]. The CR of these compression algorithms is related to the bit rates [45].

The Predictive Coding (PC) [49], Vector Quantization (VQ) based coding [47], Transform Coding (TC) [48], Compressive Sensing (CS) based coding [49], Tensor Decomposition (TD) based coding [50], Leaning based Coding (LC) [51] or Hybrid Coding (HC) algorithms [52] are classified based on the coding process of the compression [53].

In predictive coding, HSICA calculates the prediction error through the predictor. This predictor exploits the spectral correlation between the continuous frequency frames [57]. These HSICAs are data-dependable and perform only lossless HS image compression. Variable Length Coding (VLC) and Lookup Tables (LUT) are the major predictive coding-based HSICAs [46].

The VQ-based HSICA uses a coding book to achieve compression of the HS image. This codebook is present at the decoder and encoder end. The codebook differs in the compression of every HS image. HS image is partitioned into small blocks, and a specific code or symbol is given to the block throughout the training process. The HS image is reconstructed with the help of the codebook [47].

Compression of HS image through the CS-based coding algorithm is achieved in the three steps. The encoder of HSICA sensed the HS image and changed the 3D HS image into a small 2D data matrix. Again, the 2D data matrix is covered in a small data matrix. This small data matrix is transmitted to the communication channel. This process is repeated till the entire HS image gets transmitted. The decoder reconstructs the HS image when all the HS image data is received from the encoder end [49].

The TD-based HSICA uses a tensor (3D matrix) to decompose the HS image into small dimensions. These tensors of smaller dimensions are encoded, and their transmission is carried out through the channel [50].

The deep learning and machine learning methods are used in the LC-based HSICA. These compression algorithms also use the neural network (convolutional, recurrent, autoencoder, feedforward, multilayer perceptron) [54]. It has high coding efficiency with moderate CR. These HSICA have very high coding complexity and coding memory requirements. It also requires the other hardware resources [51].

The hybrid coding compression algorithms combine two or three techniques to achieve compression. Neural networks with predictive coding and neural networks with transform coding are the major combinations frequently used to design the HSICA [52].

The Transform Coding (TC) based HSICA uses mathematical transform (Wavelet, Curvelet, Fourier, Cosine, Shearlet) to convert the HS image to the frequency domain [55-59]. The energy is packed into a few coefficients through this transforming process (works as a decorrelator). These compression algorithms can work for lossless and lossy compression and have high CR [39]. The mathematical transform converts the correlation frequency frames into the energy-concentrated uncorrelated transform coefficients. Although high computed transform is complex, the associated coding algorithm is highly efficient and low complex. For the compression of HS images, many types of mathematical transform are used in a different working fashion, which depends upon the requirement of the application [40]. The 3D dyadic wavelet transform has good energy clustering in the space and time domain [60, 61]. Many other derived mathematical transforms, such as the lapped transform [62], are also used to achieve the compression of HS images. 3D-Set Partitioning Embedded Block (3D-SPECK) [63], 3D-Listless SPECK (3D-LSK) [64], 3D-Set Partitioning In Hierarchical Trees (3D-SPIHT) [65], 3D-No List SPIHT (3D-NLS) [66], 3D-Wavelet Based Tree Coding (3D-WBTC) [67], 3D-Low Memory Block Tree Coding Algorithm (3D-LMBTC) [68], 3D-Zero Memory Set Partitioning Embedded Block (3D-ZM-SPECK) [69], Fractional Wavelet Filter based Zero Memory Set Partitioning Embedded Block [70], 3D-Listless Embedded Zerotree Set Partitioning Coding (3D-LEZSPC) [56], 3D-Block Partitioning Embedded Coding (3D-BPEC) [40] are the famous transform based HSICAs.

Recently, a fractional wavelet filter was used to exploit the spatial decorrelation of HS images, while spectral decorrelation was removed by the prediction methodology [70-72]. It has been observed that with the use of a 2D fractional wavelet filter with 2D (2D-ZM-SPECK) [45], the coding efficiency of the compression algorithm increases with low coding memory and coding complexity. However, this compression algorithm has moderate transform memory, which is reduced by the use of SFrWF, and the complexity of the compression algorithm is reduced by storing the most significant coefficient of each subband for every frequency frame in the buffer memory [73, 74]. Through this buffer memory, the complexity of the proposed HSICA is reduced significantly compared to that of FrWF-based HSICA [75].

3. Related Work

The associated wavelet filter is covered in the first phase of this section, followed by linear indexing and the last in-depth description of the 2D-LC-ZM-SPECK [76].

3.1. Segmented Fractional Wavelet Filter (SFrWF)

The transformation of 3D images (hyperspectral image, magnetic resonance imaging, etc.) is a complex process that generates many terraform image data. It also has high complexity, depending on the image data [39, 76]. Because the 3D-DWT is applied directly to the complete HS image, the traditional technique of calculating the wavelet coefficients of HS images requires a massive memory [77]. The requirement of memory escalates rapidly with the high dimension HS image or level of wavelet transform. [67]. However, onboard HS image sensors suffer from low hardware resources [56]. Many other ways of wavelet transformation have been proposed to lower the coding complexity. The lifting approach to calculating the wavelet transform has been proposed, which is less complex and generates almost the same type of transform coefficients. Dyadic wavelet transform and integral wavelet transform had slightly higher complexity, but they had better energy compaction of the transform coefficients [68]. It has been known that an HS image is a combination of 2D frequency frames that are taken for a single scene [78]. Thus, they have a very high spectral correlation. 2D discrete wavelet transform (2D-DWT) has lower transform memory requirement and complexity, but they fail to remove the spectral correlation, reducing the coding efficiency [70, 79]. The same strategy has also been implemented for medical images [80]. However, using a predictor with 2D-DWT, the compression algorithm has a higher coding efficiency.

Still, 2D-DWT has a high transform memory demand, making it an unsuitable candidate for onboard sensors [70]. To solve the above-mentioned issues, three different approaches have been made: the line-based approach [81], the block-based approach [82], and the stripe-based approach [83]. In order to reduce the demand for memory for the computation of wavelet coefficients, the line-based DWT implements the lifting algorithm [82]. Applying the DWT to individual blocks, as opposed to the complete image, is yet another method for decreasing the required memory [82]. The line-based wavelet transform coefficients are saved in strip buffers in the modified version of the line-based discrete wavelet transform known as the strip-based discrete wavelet transform. This helps save memory [83]. The first two approaches have the exact transform memory requirement, but the last approach compresses high-resolution images and videos [81-84]. Recently, the DWT has been computed using the Fractional Wavelet Filter (FrWF), which is implementable on low-cost sensor nodes and requires far less memory than the approaches before it [71]. Even though the FrWF alleviates the memory constraints of low-cost portable devices and sensor nodes for HS images, the memory needs of the FrWF are significantly higher than the amount of memory often accessible on sensor nodes [85].

The SFrWF is an advanced method to transform the pixels of an HS image with a low transform memory requirement. It computes the wavelet coefficients differently [75]. Instead of taking frequency frames of the HS image into the buffer memory line by line, the image lines are partitioned into small multiple segments, and these segments will go one by one in the buffer for the calculation of wavelet transform [73]. The overlap-add method is used to combine the individual segments into one. There are nine different buffers (one input buffer, four buffers for filter coefficients, and four immediate buffers) used to calculate the wavelet coefficients in the SFrWF. Apart from these buffers, one more buffer is required to hold the convolution result [75]. The amount of memory that the SFrWF needs to operate is determined by the number of elements saved in the nine buffers and the size of each element [73]. Table 1 covers the transform memory requirement by the different transform with the coding memory of associated HSICA.

Table 1. Requirement of the memory required by the HS image coder when different types of wavelet transform and HSICAs are combined for encoding and decoding process

HS Image Coder			
Transform Stage	**Coding Stage**	**HS Image Resolution**	**Required Memory**
Conventional 3D Wavelet Transform	Any HSICA	Any Resolution	Conventional 3D Wavelet Transform
Conventional 2D Wavelet Transform	List Based HSICA	Low Resolution	Conventional 2D Wavelet Transform
		High Resolution	List Based HSICA
	3D-LSK	Any Resolution	According to the compression algorithm
	2D-ZM-SPECK	Any Resolution	Conventional 2D Wavelet Transform

3.2. Mathematical Transform Based Set Partition Hyperspectral Image Compression Algorithms (MT-SP-HSICA)

The MT-SP-HSICAs are a special compression algorithm that uses the set structure of transform HS image to achieve the compression. The mathematical transform exploits the correlation of the HS image and transforms the HS image into a few high-energy coefficients [36].

The MT-SP-HSICAs have multiple noticeable advantages over the other HSICAs, which are as follows.

- The MT-SP-HSICA starts from the most significant bit plane. Execution continues until the last bit plane is finished or the bit budget is fully consumed [34].
- The MT-SP-HSICA works at any bit rate. The reconstruction process generates the HS image at a lower bit rate post-compression. [56].
- The MT-SP-HSICA uses the set structure and partitions the transformed HS image into the partition rule (zeroblock cube, zero trees or zeroblock cube tree). With this, a single bit represents many insignificant coefficients [36].
- Since a single bit represents countless minor/inconsequential coefficients at low bit rates, these algorithms maintain strong compression performance [39].
- The MT-SP-HSICA has a lower coding complexity than other types of HSICA [40].

According to the set partition rule, the MT-SP-HSICA can be classified into three subcategories: zeroblock cube, zero trees and zeroblock cube tree [34]. The zeroblock cube method partitioned the transformed HS image into connecting block cubes. In contrast, the zero tree method groups transform coefficients corresponding to the analogous location and form the Spatial Orientation Tree (SOT) [68]. The zeroblock cube tree method uses the best features of the zeroblock cube and zero trees. It then divided the HS image into connecting block cubes, and after that, it created block cube trees in a zero tree pattern with the roots in the uppermost sub-band [34]. 3D-SPECK [63]. 3D-LSK [64] and 3D-ZM-SPECK [73] belong to the zeroblock coding, while 3D-SPIHT [65] and 3D-NLS [66] belong to the zero tree coding. 3D-WBTC [67], 3D-LMBTC [68], and 3D-LCBTC [34] belong to the zeroblock cube tree coding.

3.3. Linear Indexing

Linear indexing (Morton mapping or Z scan) converts a dimension transform matrix or three-dimensional transform matrix into a single-dimension array [70]. The HS image is converted into a 3D transform matrix using the wavelet transform, in which the sub-bands are grouped in a pyramidal shape. The sub-bands with a lower resolution can be found at the very top of the pyramid, while sub-bands with a higher resolution can be found closer to the base of the pyramid structure [39]. LLL sub-band is placed at the start of the 1D array [34].

3.4. 2D-Low Complexity Zero Memory Set Partitioned Embedded bloCK (2D-LC-ZM-SPECK)

The 2D-LC-ZM-SPECK is a low complexity version of the 2D-ZM-SPECK. It follows the same partition rule as 2D-SPECK [86] and 2D-LSK [87] and utilises the good features of linear coindexing of the transform coefficients. Through this, the proposed HSICA does not require lists, state tables, or markers to track the sets or coefficients. It has been noted that 2D-ZM-SPECK checks the significance of sets or coefficients for each bit plane. This increases the complexity of the compression algorithm. To reduce complexity, a maximum magnitude of each sub-band is saved in a buffer memory, which is used to test the significance of the sub-band.

Let us assume the HS image is transformed frame by frame through the ‘L’ level segmented fractional wavelet filter. An additional buffer memory is named M_{mem} of $[(3L+1) \times \gamma]$ elements, each having eight bytes to store the highest value of the sub-band. The first transform level generates four sub-bands, one LL sub-band (approximation subband), while the other three sub-bands are considered detail sub-bands. For the next transform level, the LL sub-band of the frequency frame is partitioned into four new sub-bands, and a new approximation sub-band is created. The detail sub-bands of the transform coefficients are grouped hierarchically in three different orientations, which are represented as LH_ω, HL_ω and HH_ω; $\omega = 1,2,3,\ldots\ldots, L$.

The initial sub-bands are arranged in the form of S_{ϕ}^{ω} & I^{ω}. The ith element of the γth frequency band is denoted as M_{mem} (i,γ) where M_{mem} (0,0) represents the LL sub-band of the first frequency frame while other sub-bands of the frequency frames are calculated as 'i = [3(L – ω) + ϕ]'. It depends on the pyramidal position and orientation of sub-bands. Through the use of buffer memory, the computational complexity of the proposed HSICA is reduced significantly, reducing the sensor power consumption and requirement of the hardware resources.
The encoding process of HSICA starts from the comparing values of first frequency frames of initial M_{mem} (0,0) and max [M_{mem} (i,0); i = 0,1,2,3,………,3L+1] against the top bit plane. The first 'S' and 'I' set is defined as S_0^L & I^L. If the 'S' set is significant to the ongoing threshold, it is quad-partitioned into four new 'S' sets. If the 'I' set is significant against the current threshold, then it will be partitioned into three new 'S' sets and a new 'I' set as $S_1^{L-1}, S_2^{L-1}, S_3^{L-1}$ & I^{L-1}. The buffer memory is now used to calculate the significance of the newly formed 'S' and 'I' sets. This process is repeated for the other transform HS image frequency frames and will continue when the last bit plane encoded or bit budget is available.
The significance of 'S' {S_{ϕ}^{ω}} set and I {I^{ω}} set is measured against the ongoing threshold is calculated with the Eq 1 & Eq 2. While the set produced by the significant 'S" set partitioning is calculated as in Eq 3.

$$\Lambda_n(S_{\phi}^{\omega}) = \begin{cases} 0 & if \quad \underset{i=3(L-\omega)+\phi}{\mathrm{M(i)}} < \eta \\ 1 & if \quad \eta < \underset{i=3(L-\omega)+\phi}{\mathrm{M(i)}} < 2\eta \\ NULL & if \quad \underset{i=3(L-\omega)+\phi}{\mathrm{M(i)}} < 2\eta \end{cases} \tag{1}$$

$$\Lambda_n(I^{\omega}) = \begin{cases} 0 & if \quad \underset{i=3(L-\omega)+\phi:3L}{\mathrm{M(i)}} < \eta \\ 1 & if \quad \eta < \underset{i=3(L-\alpha)+\phi:3L}{\mathrm{M(i)}} < 2\eta \\ NULL & if \quad \underset{i=3(L-\alpha)+\phi:3L}{\mathrm{M(i)}} < 2\eta \end{cases} \tag{2}$$

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

This same process is followed by the other frequency frames before the compression algorithms' encoding starts. By combining the sorting pass with the refinement pass, 2D-LC-ZM-SPECK does not employ the LSP.

A brief description of the different state-of-the-art MT-SP-HSICA for the compression of HS images is presented in Table 2.

4. Encoding Process of Proposed Algorithm

An unprocessed HS image (captured at the onboard sensor) is transferred to the frequency domain through a 2D SFrWF frame (L level) by frame. Through this process, the requirement of the transformed memory is reduced. A new Modified HS (MHS) image cube (the same size as the original HS image) is generated by gathering eight continuous frames. The first frequency frame is taken as it is, but the rest are taken differently. From the second frame the second frame of MHS differs between the second and first frames. This procedure is repeated for the remaining six frames of the slice of eight frames. For the generation, the rest of the frames in MHS are taken as a slice of eight continuous HS image frames and repeated until the last frame. The linear indexing scheme changes the MHS to the single-dimension array. Each frame of the MHS is encoded with the 2D-LC-ZM-SPECK. This process is repeated frame by frame (according to the bit plane) till the bit budget is obtainable. The associated pseudo code of proposed HSICA is covered in Table 3. The encoding procedure of the proposed HSICA is covered in Figure 1.

Table 2. Brief review of the different MT-SP-HSICA for compression of HS images

Set Partition Type	MT-SP-HSICA	Year	Ref	List/Listless	Coding Memory	Embeddedness
Zero Block Cube	3D-SPECK	2006	[63]	List (2)	Variable	Yes
	3D-SPEZBC	2007	[91]	List (2)	Variable	

	3D-LSK	2010	[64]	Listless	Fixed	
	3D-ZM-SPECK	2022	[69]	Listless	Zero	
	3D-BCP-ZM-SPECK	2023	[37]	Listless	Fixed	
	3D-M-ZM-SPECK	2023	[36]	Listless	Zero	
	FrWF based ZMSPECK	2024	[70]	Listless	Zero	
	3D-LBCSPC	2024	[90]	Listless	Fixed	
	BFrWF based ZMSPECK	2025	[89]	Listless	Zero	
Zero Tree	3D-SPIHT	2004	[65]	List (3)	Variable	
	3D-NLS	2013	[66]	Listless	Fixed	
	3D-LEZSPC	2023	[56]	Listless	Fixed	
	3D-BPEC	2023	[92]	Array (6)	Variable	
	3D-MELS	2023	[40]	Listless	Fixed	
	3D-LMZC	2024	[88]	Listless	Fixed	
	3D-SLS	2025	[93]	List (1)	Variable	
Zero Block Cube Tree	3D-WBTC	2019	[67]	List (3)	Variable	
	3D-LMBTC	2019	[68]	Listless	Fixed	
	3D-M-WBTC	2019	[24]	List (3)	Variable	
	3D-LCBTC	2022	[34]	List (2)	Fixed	
	3D-LBCTC	2022	[39]	Listless	Fixed	

Table 3. Pseudocode for the encoder procedure of compression algorithm (proposed)

```
Input: HS image having the dimension of α, β (spatial) and γ (spectral). The HS image of transform frame by frame by the Segmented Fractional Wavelet Filter
Output: Output bit stream from the encoder of the HSICA
Generation of Modified HS (MHS) Image
    for δ = 0 : γ-1
    {
        if(rem(δ,8)) = = 0
        {
            MHS(δ) = HS(δ)
            else
                MHS(δ) = [HS(δ) – HS(δ-1)]
    }   }
Initialization
Convert the 3D-MHS to 1D array $'\Upsilon'$ through Morton mapping
The significance function to determine the current significance of the ‘S’ & ‘I’ sets is defined as ‘Λ’ and calculated through the Eq 1 & Eq 2
Set: Number of elements in the array $\lambda = length\ |\Upsilon|$
Set: Total bit planes in MHS image $n = \log_2[\max(\Upsilon)]$
Set:  Initial Threshold of the linear array $\eta = 2^n$
Set: Initial starting index $\tau = 0$
Set: Initial root set length $\sigma_{root}$
Set: $\sigma = \sigma_{root}$
Sorting Pass
    while (γ ≤ 128)
        {
            while (τ ≤ λ)
            {
                if {(τ = σ)  && (τ ≤ $\sigma_{root}$)}
                {
                        $PSS\ (S_\phi^\omega)$
                        else
                            $PSI(I^\omega)$
                    }
                }
            }

Quantization Pass
    {
        n = (n – 1)
    }

Associated Function Description
    PSS ()
    {
        Output {$\Lambda_n\ (S_\tau^\sigma)$}
        {
            $if\ \{\Lambda_n\ (S_\tau^\sigma) = 0\}$
            {
                $\tau = (\tau + \sigma)$
                else
                {
                    if (σ > 4)
                    {
                        σ = (σ/4)
                        else
                        {
                            PScan (τ, η)
                        }
                    }
                }
            NSL (τ, σ)
            }
        }
}
PSI()
{
    Output  $\{\Lambda_n\ (I^{\omega})\}$
    if  $[\{\Lambda_n\ (I^{\omega})\}\ = 0\ ]$
    {
        $\tau = \lambda$
}   }
PScan ()
{
    for ($\mu$ = 0 : 3)
    {   Output $[\Lambda_n\ \{\Upsilon\ (\mu + \tau\}]$
        if $[\Lambda_n\ \{\Upsilon\ (\mu + \tau\}]\ =\ 1$
        {
            Output: Sign bit
            else $[\Lambda_n\ \{\Upsilon\ (\mu + \tau\}]\ =\ Null$
            {
                Output: nth MSB of $\Upsilon\ (\mu + \tau\}$
                $\tau + \tau + \sigma_{root}$
            }
        }
    }
}

NSL ()
{
    while bitand ($\tau$, 3$\sigma$) =0 ;
        {
            $\sigma\ =\ 4\sigma$
.   }       }
```

5. Simulation Result & Analysis

To test the feasibility and validity of this HSICA, extensive experimental HS image data comparison and analysis are presented in this manuscript. Six HS image datasets were chosen for the experiments to compare the proposed HSICA with eight HSICAs and analyse associated impacts on coding efficiency, memory, and complexity.

5.1. Dataset

Six HS images are taken for the simulation test, of which four HS images belong to the Yellowstone dataset. The other two HS images are named Washington DC Mall and Urban. The characteristics of the HS images are summarized in Table 4.

5.2 *Performance Evaluation Metrics*

The performance of any mathematical transform-based HSICA is measured on the metrics of coding efficiency, coding memory, transform memory and coding complexity [34]. Image quality assessment metrics, including peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), are used to measure the coding efficiency of hyperspectral image compression algorithms.

The coding complexity of HSICA is defined as the time required for the encoding and decoding the transform coefficients. The coding memory is the requirement of memory by the HSICA to encode/decode the coefficients. It may be a list array or a state table [37].

The original HS image is presented as X(a,b,c) before the HS image compression process. In contrast, after the compression process, the reconstructed HS image is represented as Y(a,b,c).

The Rate-Distortion (RD) performance of any HS image compression algorithm can be evaluated based on the algorithm's coding efficiency. It is calculated as in Eq 4, while the numeric value of the MSE [94, 95] is calculated as in Eq 5.

The symmetrical similarity index, or SSIM, is a metric that compares two HS images (original and reconstructed) to determine how similar they are to one another [96, 97]. Mathematically, it is calculated as in Eq 6.

$$PSNR = 20\ \log_{10}\left[\frac{Max\{X(a,b,c)\}}{MSE}\right] \tag{4}$$

$$MSE = \frac{1}{N_{pix}}\sum_{x,y,z}[X(a,b,c) - Y(a,b,c)]^2 \tag{5}$$

$$\mathrm{SSIM}\ (\mathrm{X},\mathrm{Y}) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2+\mu_Y^2+c_1)(\sigma_X^2+\sigma_Y^2+c_2)} \tag{6}$$

5.3. Benchmark Hyperspectral Image Compression Algorithms

The proposed HSICA is compared with the state-of-the-art transform-based HSICA 3D-SPECK (CA 1) [63]. 3D-SPIHT (CA 2) [65], 3D-WBTC [67] (CA 3), 3D-LSK [64] (CA 4), 3D-NLS [66] (CA 5), 3D-LMBTC [68] (CA 6), 3D-ZM-SPECK [69] (CA 7) and Fraction wavelet filter based ZM-SPECK [70] (CA 8). The simulation result is calculated for different bit rates and presented here on the different performance metrics.

5.4. Implementation Detail

All HSICA are implemented on the computer having the same software and hardware platform. The operating system used for the analysis is Windows 11, which runs on a 20 GB RAM computing device.

During the execution, no other simulation was performed to measure the coding complexity. Matlab 2023A is used as a simulation software for the simulation of the HSICAs on the computing device.

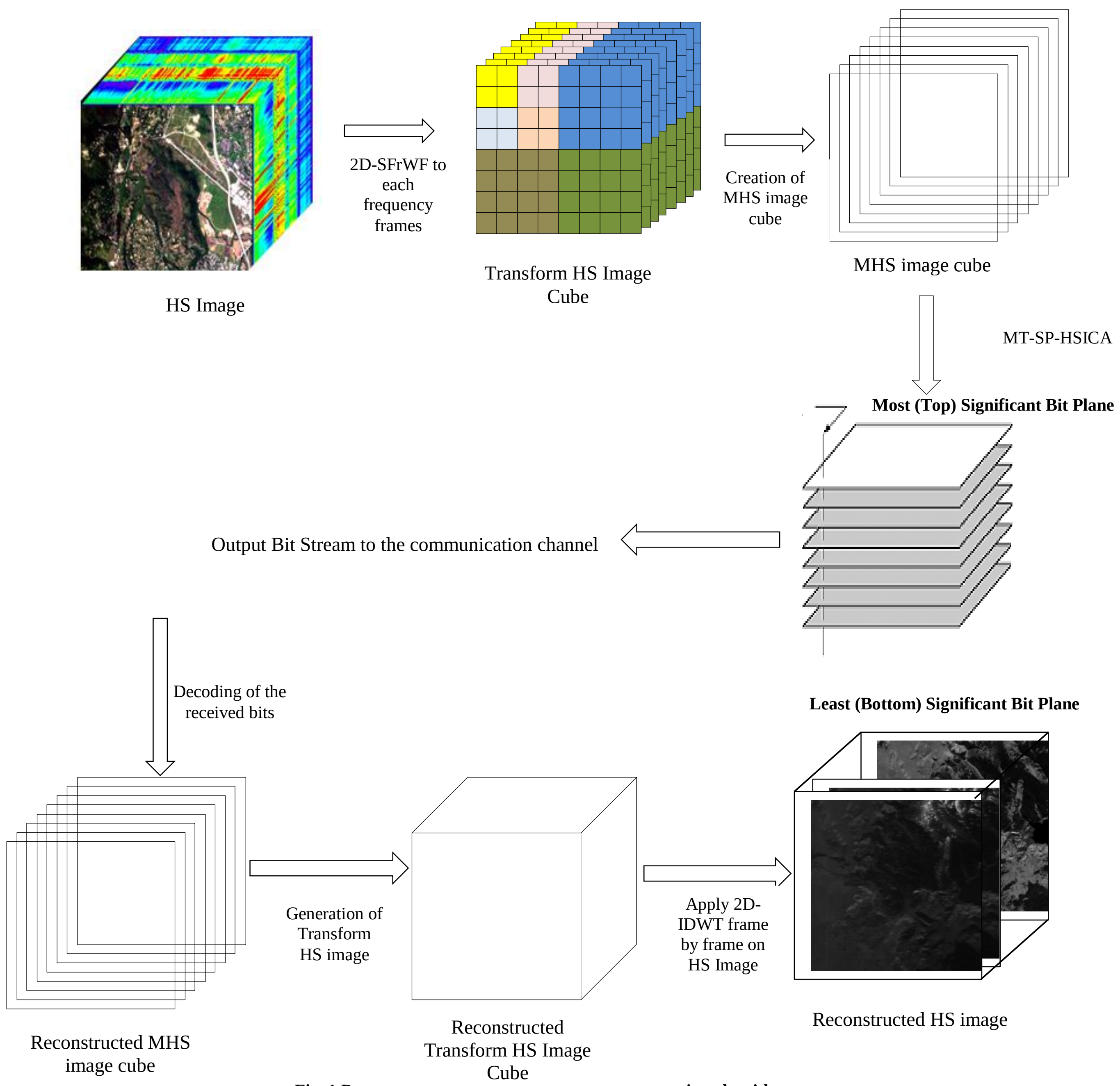
HS Image
2D-SFrWF to each frequency frames
Transform HS Image Cube
Creation of MHS image cube
MHS image cube
MT-SP-HSICA
Most (Top) Significant Bit Plane
Output Bit Stream to the communication channel
Least (Bottom) Significant Bit Plane
Decoding of the received bits
Reconstructed MHS image cube
Generation of Transform HS image
Reconstructed Transform HS Image Cube
Apply 2D-IDWT frame by frame on HS Image
Reconstructed HS image

Fig. 1 Processing pipeline for the proposed compression algorithm

Table 4. Short detail of the HS images used for analysis (simulation experiments)

HS Image	Dimension of the Image Under Test			Pixel Depth	HS Sensor
	Line	Pixel	Frame		
YSUS 0	512	680	224	14	AVIRIS
YSUS 3	512	680	224	16	AVIRIS
YSUS 10	512	680	224	16	AVIRIS
YSUS 11	512	680	224	16	AVIRIS
WDC Mall	1280	307	191	14	HYDICE
Urban	307	307	210	10	HYDICE

5.5. Coding Efficiency

Coding Efficiency is measured using PSNR and SSIM performance metrics. The proposed compression algorithm is zeroblock cube-based MT-SP-HSICA and has the same partition rules as 2D-LSK [87] and 2D-ZM-SPECK. It is clear from Table 5 that the proposed MT-SP-HSICA outperform all state-of-the-art MT-SP-HSICA and is almost the same as the FrWF-based compression algorithm. This is because they generate almost the same encoded coefficients. It has also been noted that the transform coefficient generated by the 2D-DWT, FrWF and SFrWF is similar. Thus, the coding efficiency is the same for the proposed HSICA and FrWF-based HSICA. For other MT-SP-HSICAs, the proposed algorithm outperforms from a range of two dB to three dB at low bit rates, while for high bit rates, coding efficiency differences lie from a range of four dB to five dB. The high coding gain exists because of the non-availability of the highest bit plane in 87.5% of frequency frames due to creating a modified HS image cube. This coding procedure benefit is accomplished by utilising the HS image's inherent spectral redundancy. When the bit budget between the bit planes is exhausted, the suggested compression technique's performance is poor, resulting in a modest drop in the coding gain. The only difference between the proposed HSICA and FrWF-based HSICA is the way of significance testing of the test. Table 6 compares SSIM [96] with the proposed compression algorithm and other MT-SP-HSICA.

5.6. Transform Memory

Transform (wavelet) memory is the memory required by the mathematical transform for calculating (logical, algebraic and arithmetic) the transform coefficients and saving them in the different buffers associated with the mathematical transform. It is clear from Table 6 that the 3D wavelet transform requires much memory, while the calculation of the transform coefficient through the 2D wavelet transform requires less transform memory [98, 99]. It has also been known that 2D wavelet transform works only for the spatial dimension of the HS image and does not remove the associated correlation with the spectral dimension of the HS image [70]. However, the demand for transform memory is still high and needs to be reduced in line with the coding memory. Previously, a fractional wavelet filter was used to calculate the wavelet coefficient in the spatial dimension [70], but it still has high transform memory requirements that must be addressed. The segmented fractional wavelet transform is a special type of fractional wavelet transform that reduces the transform memory requirement. From Table 7, it is clear that SFrWF outperforms other wavelet transform types.

5.7. Coding Memory

HSICA requires memory for the significance / insignificance of the sets or coefficients against the current bit plane level. The MT-ST-HSICAs use either linked lists [67], state tables [34] or markers [37] for tracking sets or coefficients. 3D-SPECK [63], 3D-SPIHT [65], and 3D-WBTC [67] employ linked lists for tracking of set/coefficients, while 3D-NLS [70], 3D-LSK [64], and 3D-LMBTC [68] use state table and markers for the same task while 3D-ZM-SPECK [69] does not need any markers and linked lists for tracking of the coefficients or sets. It is known that list-based MT-SP-HSICA has a better coding memory requirement at low bit rates than the listless MT-SP-HSICA, but the demand for coding memory increases exponentially as bit rate.

The proposed compression algorithm has a fixed demand for memory (coding), which depends on the number of sub-bands present in one frequency frame. For the ‘L’ level of mathematical transform, the number of sub-bands is (3L+1), and each coefficient has ‘8’ bits. For each frequency frame, 18 bytes of extra coding memory is required for some other computations. Thus, the required coding memory for the 128 frequency frames is ‘[8(3L +1) + 18] x 128’. Table 8 shows that the proposed HSICA has a minimal requirement of the coding memory of 18.25 KB. It outperforms the other compression algorithms but has little more than those of Bajpai et al. [69] Bajpai and Kidwai [70].

5.8. Coding Complexity

The coding complexity of any compression algorithm is measured by the time required to encode and decode to convert the transform coefficient to the bit stream and vice versa [90]. Table 9 (encoding time) and Table 10 (decoding time) show that decoding always takes less time than encoding. It has been clear that the coding complexity has been reduced significantly. The amount of time necessary to convert the transform HS image to the encoded embedded bit stream is the encoding time. At the same time, the decoding procedure takes to reconstruct the HS picture from the data received in the bit stream. From the decoding time and encoding time, it has been noted that listless MT-SP-HSICA has a low coding time requirement while list-based MT-SP-HSICA time requirement increases rapidly for the higher bit rates. Regarding coding complexity, the proposed algorithm outperforms the FrWF-based compression algorithm [70] due to the use of buffer memory, and for another list-based, MT-SP-HSICA performs better. Table 11 presents the comparative analysis of the listless compression algorithms for the different HS image sizes Figure 2. Figure 2 displays the original HS image and the reconstructed HS image after compression for the Uncalibrated Yellowstone Scene 00 at a CR of 16.

6. Conclusion

HS images have great potential in many fields and are a powerful, non-destructive assessment technique. However, a huge amount of HS image data generated by the HS image sensors is hard to process [95]. Hence, to make HS images accessible to all, reducing HS image data is mandatory, and it should also work with low-resource HS image sensors. Many HSICAs were proposed in the past, but they are either complex in execution nature or have low coding efficiency. The present manuscript proposed a compression algorithm with low transform memory and computational complexity. It is due to the reduction in the multiple memory access, which makes the compression algorithm fast.

Acknowledgement

The authors thank Integral University for providing MCN "IU/R&D/2025-MCN0003634".

Declarations

Author Contributions

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

Table 5. Coding efficiency comparison between different MT-SP-HSICAs for six HS images

Bit Rate	**CA 1 [63]**	**CA 4 [64]**	**CA 2 [65]**	**CA 5 [66]**	**CA 3 [67]**	**CA 6 [68]**	**3CA 7 [69]**	**CA 8 [70]**	**Proposed Algorithm**
	YSUS 0 HS Image								
0.00625	26.4	26.9	26.3	26.3	26.4	26.9	26.9	30.2	30.2
0.0125	27.6	27.5	27.4	27.4	27.6	27.5	27.5	31.5	31.5
0.025	29.0	28.4	28.8	28.3	29.1	28.3	28.4	32.8	32.8
0.0375	30.0	29.9	29.8	29.8	29.9	30.0	30.0	34.2	34.2
0.05	30.9	30.5	30.7	30.6	30.8	30.5	30.5	34.7	34.7
0.1	32.8	32.8	32.7	32.7	32.8	32.8	32.8	37	37
0.2	35.1	34.6	35.1	34.5	35.2	34.6	34.6	39.8	39.8
0.3	36.9	36.7	36.8	36.7	36.9	36.6	36.6	41.5	41.5
0.4	38.4	37.6	38.2	37.8	38.3	37.5	37.5	43.3	43.3
0.5	39.5	39.5	39.4	39.3	39.5	39.1	39.1	45	45
0.6	40.8	40.6	40.7	40.6	40.8	40.5	40.5	45.7	45.7
0.7	41.9	41.3	41.8	41.8	41.9	41.2	41.2	47	47
0.8	43.0	42.3	42.9	42.5	43.0	42.1	42.1	48.1	48.1
0.9	44.0	43.8	43.8	43.5	44.0	43.0	43.0	49.4	49.4
1	44.9	44.9	44.7	44.7	44.8	44.8	44.8	49.9	49.9
	YSUS 3 HS Image								
0.00625	30.5	30.5	30.3	29.8	30.5	30.5	30.5	32.4	32.4
0.0125	31.4	31.3	31.3	31.2	31.4	31.3	31.3	33.9	33.9
0.025	32.9	32.9	32.7	32.6	32.9	32.9	33	34.1	34.1
0.0375	34	33.8	33.8	33.8	34	33.8	33.8	36.7	36.7
0.05	34.9	34.7	34.6	34.4	34.8	34.6	34.7	37.9	37.9
0.1	37.2	37.1	37	37	37.2	36.9	36.9	41	41
0.2	40.4	40.1	40.2	40.1	40.4	40.1	40.1	44.2	44.2
0.3	42.9	42.8	42.7	42.4	42.9	42.4	42.5	46.9	46.9
0.4	45	44.6	44.7	44.6	45	44.5	44.5	49.2	49.2
0.5	47	46.7	46.8	46.3	47	46.1	46.2	51.5	51.5
0.6	48.9	48.9	48.5	48.5	48.9	48.9	48.9	53.7	53.7
0.7	50.7	50.2	50.3	50.2	50.7	50	50	54.8	54.8
0.8	52.1	52.2	51.8	51.7	52	51.4	51.4	56.4	56.4
0.9	53.8	53.8	53.4	53.4	53.8	53.8	53.8	58.5	58.5
1	55.4	55.1	55	54.9	55.3	54.9	54.9	59.9	59.9
	YSUS 10 HS Image								
0.00625	26.5	26.2	26.2	26.5	26.5	26.1	26.1	28.9	28.9
0.0125	27.6	27.5	27.5	27.4	27.5	27.5	27.5	29.6	29.6
0.025	28.5	28.2	28.2	28.2	28.7	28.2	28.3	30.8	30.8
0.0375	29.4	29.3	29.3	29.1	29.3	29.4	29.4	31.6	31.6
0.05	30.0	29.8	29.8	29.8	30.0	29.8	29.8	32.5	32.5
0.1	31.5	31.4	31.4	31.3	31.5	31.4	31.5	34.1	34.1

0.2	33.4	33.2	33.2	33.2	33.5	33.1	33.2	36.7	36.7
0.3	34.9	34.7	34.7	34.5	34.9	34.5	34.5	38.8	38.8
0.4	36.1	35.9	35.9	36.0	36.1	35.9	35.9	40.3	40.3
0.5	37.3	36.8	36.8	37.0	37.3	36.7	36.7	41.7	41.7
0.6	38.5	37.9	37.9	38.0	38.5	37.8	37.9	43.6	43.6
0.7	39.5	39.5	39.5	39.4	39.5	39.3	39.3	44.3	44.3
0.8	40.5	40.3	40.3	40.3	40.5	40.3	40.3	45.0	45.0
0.9	41.6	41.1	41.1	41.3	41.6	40.9	40.9	45.8	45.8
1	42.6	42.1	42.1	42.3	42.6	41.7	41.7	46.8	46.8
	YSUS 11 HS Image								
0.00625	30.4	30.4	30.2	30.2	30.4	30.3	30.4	31.9	31.9
0.0125	31.7	31.6	31.6	31.5	31.7	31.5	31.5	33.1	33.1
0.025	33	33	32.9	32.7	33	32.9	33	34.4	34.4
0.0375	34.1	33.9	33.9	33.8	34.1	33.8	33.8	35.8	35.8
0.05	34.9	34.4	34.6	34.5	34.8	34.4	34.4	36.7	36.7
0.1	36.9	36.6	36.7	36.7	36.9	36.5	36.6	39.1	39.1
0.2	38.9	38.8	38.8	38.8	38.9	38.8	38.8	42.2	42.2
0.3	40.5	40.1	40.4	40.2	40.5	40	40	44	44
0.4	41.9	41.7	41.8	41.5	41.9	41.3	41.4	46.0	46.0
0.5	43.2	43.1	43.1	43.0	43.2	43	43.0	47.7	47.7
0.6	44.5	43.9	44.3	44.3	44.4	43.8	43.8	48.6	48.6
0.7	45.6	45.1	45.4	45.1	45.7	44.8	44.8	50.1	50.1
0.8	46.7	46.7	46.6	46.5	46.7	46.1	46.1	51.5	51.5
0.9	47.8	47.8	47.7	47.7	47.8	47.8	47.8	52.1	52.1
1	48.8	48.5	48.6	48.5	48.8	48.3	48.3	53.1	53.1
	WDC Mall HS Image								
0.00625	32.1	32.2	33	32.1	32.1	32.9	32.1	35.5	35.5
0.0125	33.3	33.3	33.1	33.1	33.3	33.2	33.3	36.4	36.4
0.025	34.6	34.5	34.4	34.4	34.5	34.4	34.4	37.2	37.2
0.0375	35.6	35.5	35.3	35.1	35.5	35.2	35.5	38.4	38.4
0.05	36.5	36.5	36.3	36.3	36.5	36.5	36.5	39.7	39.7
0.1	38.5	38.4	38.3	38.1	38.5	38.3	38.3	40.9	40.9
0.2	41.5	41.5	41.3	41.3	41.5	41.2	41.4	44.2	44.2
0.3	43.5	43.6	43.3	43.3	43.5	43.5	43.6	46.5	46.5
0.4	45.3	45.1	45.1	45.1	45.3	44.6	45.2	48.6	48.6
0.5	46.8	46.8	46.6	46.4	46.8	46.1	46.7	49.4	49.4
0.6	48.5	48.4	48.2	48.2	48.4	48.4	48.4	51.1	51.1
0.7	49.8	49.7	49.5	49.5	49.7	49.2	49.7	52.7	52.7
0.8	51.1	51.1	50.8	50.8	51.1	50.3	51	54.4	54.4
0.9	52.2	52.2	52.1	52.1	52.2	51.7	52.1	55.7	55.7
1	53.5	53.5	53.3	53.3	53.5	53.5	53.5	57.2	57.2
	Urban HS Image								
0.00625	52.7	52.7	53	52.7	52.7	52.9	52.9	56.2	56.2
0.0125	53.3	53.3	53.2	53.2	53.3	53.2	53.2	56.9	56.9
0.025	54.3	54.3	54.2	54.1	54.3	54.3	54.3	57.4	57.4
0.0375	55.1	55.1	55.0	55	55.1	55	55.1	58.2	58.2
0.05	55.7	55.6	55.6	55.5	55.6	55.5	55.5	58.9	58.9
0.1	57	57	56.9	56.9	57.1	57.1	57.1	60.4	60.4
0.2	59	58.8	58.8	58.7	59	58.7	58.7	61.7	61.7
0.3	60.4	60.4	60.3	60.3	60.5	60.5	60.6	62.8	62.8

0.4	61.8	61.6	61.7	61.7	61.9	61.5	61.6	64.4	64.4
0.5	63	62.9	62.8	62.6	63	62.6	62.8	65.6	65.6
0.6	64.2	64.1	64	63.9	64.2	64	64	67.9	67.9
0.7	65.4	65.4	65.3	65.3	65.5	65.3	65.3	68.4	68.4
0.8	66.3	66.3	66.2	66.2	66.5	65.9	66.2	69.7	69.7
0.9	67.4	67.2	67.3	67.3	67.5	66.7	67.1	70.9	70.9
1	68.4	68.3	68.2	68.1	68.6	67.7	68.3	72.1	72.1

Table 6. SSIM calculation for compression algorithms for YSUS 0

Bit Rate	**CA 1 [63]**	**CA 4 [64]**	**CA 2 [65]**	**CA 5 [66]**	**CA 3 [67]**	**CA 6 [68]**	**3CA 7 [69]**	**CA 8 [70]**	**Proposed Algorithm**
	YSUS 0 HS Image								
0.00625	0.39	0.38	0.38	0.38	0.39	0.38	0.37	0.41	0.41
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	0.67
0.05	0.66	0.65	0.65	0.65	0.66	0.65	0.65	0.71	0.71
0.1	0.75	0.75	0.75	0.74	0.75	0.74	0.75	0.80	0.80
0.2	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.87	0.87
0.3	0.87	0.88	0.87	0.87	0.87	0.87	0.87	0.90	0.90
0.4	0.91	0.90	0.90	0.90	0.90	0.90	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	0.97
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

Table 7. Requirement of the transform memory

Transform memory requirement for the HS image under test.		
Type of Wavelet Transform	Reference	Transform Memory
3D DWT	[37]	38.34 MB
2D DWT	[70]	348.99 KB
2D FrWF	[70]	3.123 KB
2D BFrWF (4)	[87]	1.5615
2D SFrWF (2)	Applied in this compression algorithm	1.2 KB
2D SFrWF (4)		0.624 KB
2D SFrWF (8)		0.336 KB

Table 8. Requirement of coding memory by compression algorithms for HS images

Bit Rate	CA 1 [63]	CA 4 [64]	CA 2 [65]	CA 5 [66]	CA 3 [67]	CA 6 [68]	3CA 7 [69]	CA 8 [70]	Proposed Algorithm
	YSUS 0 HS Image								
0.00625	18	512	15	1024	8	12	0	0	18.25
0.0125	31	512	36	1024	23	12	0	0	18.25
0.025	62	512	71	1024	53	12	0	0	18.25
0.0375	109	512	111	1024	100	12	0	0	18.25
0.05	109	512	117	1024	108	12	0	0	18.25
0.1	237	512	249	1024	208	12	0	0	18.25
0.2	494	512	513	1024	489	12	0	0	18.25
0.3	597	512	615	1024	578	12	0	0	18.25
0.4	843	512	896	1024	844	12	0	0	18.25
0.5	1224	512	1222	1024	1184	12	0	0	18.25
0.6	1329	512	1354	1024	1230	12	0	0	18.25
0.7	1456	512	1461	1024	1356	12	0	0	18.25
0.8	1653	512	1680	1024	1554	12	0	0	18.25
0.9	1892	512	1906	1024	1876	12	0	0	18.25
1	2010	512	2072	1024	2022	12	0	0	18.25
	YSUS 3 HS Image								
0.00625	16	512	17	1024	8	12	0	0	18.25
0.0125	26	512	28	1024	20	12	0	0	18.25
0.025	71	512	73	1024	63	12	0	0	18.25
0.0375	78	512	81	1024	72	12	0	0	18.25
0.05	117	512	124	1024	109	12	0	0	18.25
0.1	205	512	204	1024	186	12	0	0	18.25
0.2	397	512	414	1024	378	12	0	0	18.25
0.3	678	512	682	1024	621	12	0	0	18.25
0.4	726	512	753	1024	708	12	0	0	18.25
0.5	947	512	947	1024	894	12	0	0	18.25
0.6	1119	512	1155	1024	1091	12	0	0	18.25
0.7	1234	512	1243	1024	1121	12	0	0	18.25
0.8	1343	512	1324	1024	1282	12	0	0	18.25
0.9	1439	512	1473	1024	1400	12	0	0	18.25
1	1498	512	1517	1024	1440	12	0	0	18.25
	YSUS 10 HS Image								
0.00625	19	512	15	1024	11	12	0	0	18.25
0.0125	33	512	37	1024	34	12	0	0	18.25
0.025	65	512	71	1024	64	12	0	0	18.25

0.0375	112	512	116	1024	101	12	0	0	18.25
0.05	163	512	123	1024	113	12	0	0	18.25
0.1	276	512	289	1024	282	12	0	0	18.25
0.2	448	512	489	1024	446	12	0	0	18.25
0.3	822	512	833	1024	826	12	0	0	18.25
0.4	920	512	881	1024	891	12	0	0	18.25
0.5	1038	512	1055	1024	1037	12	0	0	18.25
0.6	1352	512	1389	1024	1355	12	0	0	18.25
0.7	1634	512	1561	1024	1592	12	0	0	18.25
0.8	1681	512	1652	1024	1634	12	0	0	18.25
0.9	1741	512	1710	1024	1712	12	0	0	18.25
1	1825	512	1813	1024	1824	12	0	0	18.25
	YSUS 11 HS Image								
0.00625	15	512	16	1024	11	12	0	0	18.25
0.0125	26	512	29	1024	27	12	0	0	18.25
0.025	69	512	74	1024	71	12	0	0	18.25
0.0375	79	512	87	1024	80	12	0	0	18.25
0.05	112	512	121	1024	112	12	0	0	18.25
0.1	196	512	198	1024	196	12	0	0	18.25
0.2	467	512	486	1024	469	12	0	0	18.25
0.3	634	512	669	1024	633	12	0	0	18.25
0.4	993	512	1013	1024	998	12	0	0	18.25
0.5	1074	512	1102	1024	1076	12	0	0	18.25
0.6	1112	512	1102	1024	1112	12	0	0	18.25
0.7	1409	512	1431	1024	1407	12	0	0	18.25
0.8	1564	512	1560	1024	1547	12	0	0	18.25
0.9	1689	512	1678	1024	1640	12	0	0	18.25
1	1789	512	1821	1024	1790	12	0	0	18.25
	WDC Mall HS Image								
0.00625	26	512	28	1024	18	12	0	0	18.25
0.0125	33	512	38	1024	35	12	0	0	18.25
0.025	55	512	54	1024	55	12	0	0	18.25
0.0375	96	512	103	1024	99	12	0	0	18.25
0.05	136	512	145	1024	138	12	0	0	18.25
0.1	244	512	263	1024	250	12	0	0	18.25
0.2	416	512	438	1024	416	12	0	0	18.25
0.3	701	512	629	1024	704	12	0	0	18.25
0.4	734	512	724	1024	733	12	0	0	18.25
0.5	1049	512	1061	1024	1049	12	0	0	18.25

0.6	1191	512	1223	1024	1195	12	0	0	18.25
0.7	1277	512	1302	1024	1282	12	0	0	18.25
0.8	1408	512	1415	1024	1404	12	0	0	18.25
0.9	1703	512	1726	1024	1705	12	0	0	18.25
1	1802	512	1827	1024	1725	12	0	0	18.25
	Urban HS Image								
0.00625	18	512	22	1024	19	12	0	0	18.25
0.0125	28	512	31	1024	28	12	0	0	18.25
0.025	70	512	79	1024	73	12	0	0	18.25
0.0375	95	512	105	1024	97	12	0	0	18.25
0.05	109	512	113	1024	110	12	0	0	18.25
0.1	293	512	299	1024	294	12	0	0	18.25
0.2	478	512	530	1024	484	12	0	0	18.25
0.3	841	512	864	1024	843	12	0	0	18.25
0.4	841	512	867	1024	843	12	0	0	18.25
0.5	1077	512	1110	1024	1077	12	0	0	18.25
0.6	1419	512	1445	1024	1425	12	0	0	18.25
0.7	1492	512	1499	1024	1494	12	0	0	18.25
0.8	1563	512	1586	1024	1564	12	0	0	18.25
0.9	1590	512	1715	1024	1590	12	0	0	18.25
1	1890	512	1906	1024	1889	12	0	0	18.25

Table 9. Encoding time consumed by different HSICAs for the compression of HS images

Bit Rate	**CA 1 [63]**	**CA 4 [64]**	**CA 2 [65]**	**CA 5 [66]**	**CA 3 [67]**	**CA 6 [68]**	**3CA 7 [69]**	**CA 8 [70]**	**Proposed Algorithm**
	YSUS 0 HS Image								
0.00625	1.1	0.2	1.3	0.3	1.1	0.3	0.6	0.6	0.5
0.0125	1.4	0.4	1.1	0.6	1.3	1.2	0.7	0.7	0.6
0.025	2.8	0.5	1.5	0.7	1.7	1.5	0.9	0.8	0.8
0.0375	4.3	0.6	2.2	0.8	2.2	1.7	1	1.1	1
0.05	7.3	0.7	3.4	0.9	3.5	2	1.2	1.3	1.2
0.1	18.5	0.9	6.7	1	6.2	3.1	1.8	2.1	1.9
0.2	89.7	1.1	31	1.4	17.6	4.9	2.8	3.7	3.5
0.3	195	1.5	67	1.8	60.7	7.1	3.9	5.4	5.3
0.4	249	1.8	96	2.1	118	8.6	4.9	7	6.7
0.5	340	2.1	118	2.4	173	10	5.8	8.4	8.1

0.6	692	2.6	257	2.9	376	13	7.1	9.9	9.7
0.7	961	2.8	463	3.2	657	14.2	8.1	11.6	11.2
0.8	1167	3.0	491	3.5	759	15.8	9.2	13.4	12.9
0.9	1304	3.3	513	4.1	855	16.8	9.9	15.3	14.8
1	1441	3.8	560	4.9	878	17.4	10.9	20.6	20.2
	YSUS 3 HS Image								
0.00625	0.9	0.2	1.0	0.4	0.5	0.4	0.4	0.9	0.9
0.0125	1.6	0.3	1.1	0.6	1.3	1.1	0.8	1.7	1.6
0.025	2.9	0.4	1.6	0.7	1.8	1.4	0.9	2.1	2
0.0375	5.0	0.5	2.2	0.8	2.4	1.7	1.3	2.2	2.1
0.05	6.7	0.6	4.3	0.9	2.7	2.1	1.4	2.4	2.2
0.1	18.2	0.8	9.9	1.4	7.1	3.2	1.9	3.3	3.1
0.2	57.3	1.2	24.3	1.9	25.0	5.5	3.0	5.1	4.8
0.3	97.1	1.4	39.9	2.1	47.9	6.7	3.9	6.8	6.7
0.4	206	1.8	104	2.4	122	9.3	5.1	8.4	8.1
0.5	303	2.2	131	2.9	177	10.5	6.1	10.1	9.7
0.6	349	2.5	141	3.2	190	11.5	7.2	11.8	11.4
0.7	706	2.7	338	3.5	528	14.3	8.6	13.5	13.1
0.8	835	3.0	408	3.8	601	15.7	9.5	15.2	14.9
0.9	981	3.5	414	4.5	633	16.8	10.7	16.9	16.5
1	1373	3.8	632	5.1	977	19.9	11.9	18.3	17.8
	YSUS 10 HS Image								
0.00625	1.8	0.2	0.6	0.4	0.8	0.9	0.4	0.6	0.5
0.0125	2.5	0.3	1.1	0.5	1.4	1.2	0.7	0.9	0.8
0.025	5.1	0.4	1.6	0.6	1.7	1.5	0.9	1.1	1
0.0375	5.6	0.5	2.2	0.7	2.1	1.7	1.1	1.3	1.1
0.05	8.5	0.6	3.3	0.8	3.5	2.1	1.2	1.4	1.3
0.1	18.8	0.7	8.1	0.9	5.9	3.2	1.7	2.2	2
0.2	83.5	1.1	28.9	1.3	23.6	5.3	3.7	3.7	3.4
0.3	110	1.3	50.3	1.7	32.5	6.6	3.9	5.3	4.9
0.4	326	1.7	181	2.0	199	9.2	5.3	7.2	7.1
0.5	472	2.0	241	2.3	294	11.5	7.2	10.8	10.5
0.6	588	2.2	264	2.5	363	11.3	8.4	11.4	11.2
0.7	678	2.5	281	2.7	398	12.5	7.8	12.6	12.4
0.8	1151	2.9	531	3.2	772	15.8	10.0	15.2	15
0.9	1745	3.1	883	3.4	1223	17.1	12.0	17.6	17.4
1	2558	3.3	1034	3.9	1517	18.3	13.7	18.5	18.3
	YSUS 11 HS Image								
0.00625	1	0.2	0.7	0.3	0.6	0.9	0.5	0.5	0.4
0.0125	1.6	0.3	1.0	0.4	2.3	1.8	0.9	0.6	0.5

0.025	3.0	0.4	1.5	0.5	3.1	1.7	0.9	0.8	0.7
0.0375	5.5	0.5	2.1	0.6	6.0	2.1	1.2	1.1	1
0.05	7.4	0.6	2.9	0.7	15.4	2.5	1.3	1.3	1.1
0.1	19.7	0.7	7.8	0.9	17.8	3.6	2.1	2.1	1.9
0.2	81.4	1.1	21.3	1.3	29.1	6.7	3.2	4.4	4.1
0.3	142	1.4	67.0	1.7	70.6	7.7	4.1	6.7	6.5
0.4	227	1.8	77.7	2.0	99.8	9.3	4.9	7.8	7.6
0.5	509	2.1	227	2.4	280	12.2	7.2	9.3	9.1
0.6	755	2.3	405	2.8	495	14.2	7.6	11.1	10.8
0.7	981	2.5	424	3.0	578	14.5	8.5	12.2	12
0.8	1018	2.8	449	3.2	646	14.7	9.1	16.5	16.2
0.9	1329	3.3	447	3.5	738	17.9	10.2	18.2	18
1	2073	3.6	873	4.1	1291	19.9	11.4	24.3	24.1
	WDC Mall HS Image								
0.00625	0.6	0.3	0.9	0.4	0.7	0.9	0.5	0.7	0.6
0.0125	1.7	0.4	0.9	0.4	1.2	1.0	0.7	0.9	0.8
0.025	3.0	0.4	1.5	0.5	1.5	1.3	0.9	1.5	1.3
0.0375	5.1	0.5	2.2	0.6	2.3	1.6	1.2	1.9	1.7
0.05	7.6	0.5	3.1	0.7	3.1	1.8	1.4	2.1	2
0.1	15.6	1.6	6.8	1.8	6.9	3.5	2.3	3.8	3.7
0.2	49.5	3.0	19.7	3.2	19.1	5.6	3.5	6.5	6.2
0.3	85.7	4.1	48.9	4.4	26.4	7.8	4.6	8.8	8.6
0.4	312	5.6	102	5.9	192	10.7	6.1	11.1	12.8
0.5	416	7.1	158	8.2	254	11.7	7.1	13.7	13.4
0.6	887	8.8	207	9.1	300	13.2	8.2	15.2	14.9
0.7	905	9.5	212	10.7	372	16.9	9.4	17.7	17.5
0.8	1125	11.1	542	11.3	789	18.3	10.6	18.8	18.6
0.9	1543	12.7	774	13	1067	19.5	12.2	20.9	20.6
1	1703	14.4	780	14.5	1185	20.9	13	24.7	24.5
	Urban HS Image								
0.00625	0.3	0.3	0.9	0.3	0.9	0.9	0.4	0.7	0.6
0.0125	1.5	0.4	1.1	0.4	1.2	1.1	0.6	0.9	0.8
0.025	3.1	0.4	1.4	0.5	1.7	1.3	0.8	1.1	1
0.0375	5.3	0.5	2.1	0.6	2.2	1.6	1	1.7	1.6
0.05	6.2	0.6	2.7	0.7	2.8	1.9	1.1	2.4	2.2
0.1	25	0.8	7.5	0.9	6.5	3.9	1.8	3.1	2.9
0.2	57.9	1.1	25.8	1.2	24.8	5.1	2.8	4.9	4.7
0.3	92.1	1.5	37.5	1.7	32	7.7	3.7	7.1	6.8
0.4	270	2	118	2.1	196	9.7	5.7	9.9	9.5

0.5	415	2.5	140	2.6	211	11.3	7.4	12.8	12.6
0.6	576	2.9	166	3	248	13.3	8	14.7	14.5
0.7	888	3.2	406	3.4	625	18.1	9.7	17.1	16.8
0.8	1131	3.8	474	4	710	20.0	9.9	18.3	18
0.9	1335	4	556	4.1	746	20.6	12.5	20.4	20.2
1	1498	4.4	575	4.6	804	21.1	13.2	24.5	24.2

Table 10. Decoding time consumed by different HSICAs for the compression of HS images

Bit Rate	**CA 1 [63]**	**CA 4 [64]**	**CA 2 [65]**	**CA 5 [66]**	**CA 3 [67]**	**CA 6 [68]**	**3CA 7 [69]**	**CA 8 [70]**	**Proposed Algorithm**
	YSUS 0 HS Image								
0.00625	0.7	0.1	0.9	0.21	0.7	0.3	0.36	0.46	0.3
0.0125	0.7	0.3	0.3	0.4	0.4	0.5	0.7	0.5	0.4
0.025	1.5	0.4	0.7	0.8	0.7	0.7	0.8	0.7	0.6
0.0375	2.8	0.4	1.2	0.6	0.9	0.8	1.0	1	0.9
0.05	5.1	0.5	2.2	0.6	2.3	1.0	1.1	1.3	1.1
0.1	12.9	0.7	5.3	0.8	4.6	1.8	1.5	1.8	1.7
0.2	61.4	1.0	23.9	1.2	15.4	3.4	2.7	3.2	3
0.3	136	1.1	52.4	0.6	57.9	4.7	3.5	5	4.7
0.4	204	1.7	87.9	1.9	108	6.2	4.3	6.5	6.1
0.5	306	1.9	98.9	2.2	148	7.8	5.2	8	7.7
0.6	600	2.2	217	2.5	341	9.2	6.7	9.3	8.9
0.7	906	2.4	456	2.7	618	10.4	7.7	11.1	10.7
0.8	1005	2.5	486	3.0	727	13.8	8.5	12.5	12.2
0.9	1249	3.1	501	3.3	830	13.3	9.4	14.5	14.3
1	1389	3.6	512	4.0	832	14.5	10.6	19	18.8
	YSUS 3 HS Image								
0.00625	0.8	0.5	0.7	0.3	0.2	0.3	0.3	0.5	0.4
0.0125	0.9	0.3	0.8	0.5	0.4	0.5	0.6	0.7	0.6
0.025	1.6	0.4	1.2	0.6	0.6	0.7	0.8	0.9	0.8
0.0375	3.5	0.4	1.5	0.7	1.2	0.9	1.1	1.1	1
0.05	4.6	0.5	2.7	0.8	1.6	1.1	1.2	1.4	1.2
0.1	14.5	0.7	6.6	1.1	9.5	1.9	1.5	2.3	2.1
0.2	49.4	1	22.3	1.5	17.6	3.5	2.6	4.1	3.9
0.3	81.9	1.4	31.6	1.8	40.1	5.1	3.5	5.8	5.5
0.4	190	1.7	90.4	2.2	110	6.7	4.8	7.6	7.4
0.5	284	2	118	2.5	156	8.1	5.9	9.4	9.2

0.6	321	2.4	128	2.9	176	9.4	6.9	11.1	10.8
0.7	684	2.5	309	3.2	577	11	8.3	13	12.8
0.8	822	2.8	373	3.4	553	12.5	9.3	14.7	14.5
0.9	860	3.2	400	3.8	589	13.9	10.3	16.2	16
1	1258	3.4	589	4	849	15.6	11.5	18.1	17.8
	YSUS 10 HS Image								
0.00625	0.8	0.1	0.3	0.3	0.2	0.5	0.3	0.4	0.3
0.0125	1.9	0.2	0.3	0.4	0.4	0.6	0.6	0.6	0.6
0.025	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	0.9
0.05	5.9	0.5	2.2	0.7	2.3	1.0	1.2	1.1	1
0.1	12.8	0.6	6.5	0.8	4.4	1.8	1.6	1.9	1.7
0.2	62.2	1.0	25.8	1.1	21.6	3.3	2.8	3.2	3
0.3	80.0	1.2	47.3	1.5	29.2	4.9	4.0	5.1	4.9
0.4	307	1.4	172	1.9	175	6.7	5.5	7	6.7
0.5	441	1.7	219	2.1	262	7.6	7.8	9.6	9.4
0.6	555	1.9	230	2.2	319	9.0	6.6	10.1	9.9
0.7	609	2.2	246	2.5	359	10.5	7.5	12	11.7
0.8	1080	2.5	496	2.8	717	11.8	9.2	14.3	14
0.9	1692	2.9	818	3.1	1130	13.3	10.0	16.5	16.3
1	2179	3.4	930	3.4	1384	14.7	11.5	17.2	17
	YSUS 11 HS Image								
0.00625	0.66	0.2	0.2	0.2	0.2	0.5	0.4	0.4	0.3
0.0125	0.93	0.3	0.3	0.4	0.8	0.6	0.7	0.5	0.4
0.025	1.63	0.4	0.6	0.4	0.9	0.8	0.8	0.7	0.6
0.0375	3.74	0.4	1.2	0.5	3.3	1.2	1.1	0.9	0.8
0.05	4.9	0.5	1.9	0.6	4.1	1.7	1.2	1.1	1
0.1	15.8	0.7	6.5	0.8	11.5	2.3	1.9	1.9	1.7
0.2	61.1	1	16.2	1.1	20.9	3.9	2.8	4.1	3.8
0.3	118	1.3	56.1	1.6	54.1	5.7	3.9	5.5	5.2
0.4	195	1.7	70.2	1.9	89.4	7.3	4.8	6.5	6.2
0.5	400	2	216	2.2	250	9.1	6.1	8	7.8
0.6	847	2.2	394	2.5	471	11.3	7.3	10.2	9.9
0.7	840	2.3	419	2.9	575	12.6	8.2	11.8	11.6
0.8	1059	2.6	436	3.1	621	13.4	8.7	15.6	15.4
0.9	1186	2.9	440	3.3	706	14.2	9.5	17.8	17.6
1	1901	3.4	867	3.9	1260	17.4	10.6	20.9	20.6
	WDC Mall HS Image								
0.00625	0.3	0.2	0.3	0.3	0.2	0.3	0.3	0.5	0.4
0.0125	0.6	0.3	0.8	0.3	0.4	0.4	0.6	0.7	0.6

0.025	0.8	0.4	1.0	0.4	0.8	0.6	0.8	1.2	1.1
0.0375	0.9	0.4	1.9	0.5	1.1	0.8	0.9	1.5	1.4
0.05	1.1	0.5	2.6	0.6	1.4	1	1	1.8	1.6
0.1	17.4	0.7	6.1	0.8	5.0	2.3	1.7	2.2	2
0.2	48.8	1.1	24.8	1.1	22.5	3.3	2.7	4.8	4.6
0.3	75.4	1.5	34.8	1.4	28.5	4.9	3.6	7.4	7.1
0.4	264	1.7	106	1.9	180	8.1	5.4	8.4	8
0.5	339	2.2	135	2.3	192	7.7	6.8	11.1	10.8
0.6	532	2.6	150	2.8	245	9.8	8	14.2	14
0.7	808	2.7	327	3	558	11.6	8.8	16.4	16.1
0.8	1058	3.1	449	3.7	675	13.4	9.4	17.4	17.2
0.9	1142	3.2	486	3.9	725	13.6	11.8	19.1	18.7
1	1290	3.7	504	4.2	774	15.5	12.3	22.9	22.6
	Urban HS Image								
0.00625	0.2	0.3	0.6	0.3	0.3	0.2	0.3	0.6	0.5
0.0125	1	0.35	0.7	0.4	0.4	0.4	0.7	0.8	0.7
0.025	1.7	0.4	1.1	0.4	0.6	0.6	0.8	0.9	0.8
0.0375	3.4	0.4	1.3	0.5	1.2	0.8	1.1	1.5	1.3
0.05	5.7	0.5	2.2	0.5	2.1	1	1.2	1.8	1.6
0.1	11.8	1.5	4.3	1.7	4.8	1.7	1.9	2.1	2
0.2	41.6	2.4	16.8	2.9	16.4	3.2	2.9	3.9	3.7
0.3	72.3	3.2	42.5	4.1	23.2	4.7	3.7	6.6	6.4
0.4	292	4.7	88	5.7	185	6	4.7	9.1	8.8
0.5	388	5.4	149	8	243	7.3	5.4	11.1	10.7
0.6	487	6.1	197	8.9	295	8.6	5.9	12.8	12.6
0.7	593	8.2	201	10.5	366	10	7.1	14.7	14.5
0.8	1067	8.8	507	11.1	772	11.4	8	15.8	15.6
0.9	1506	9.4	750	12.8	1052	12.6	8.8	16.9	16.7
1	1662	10	762	14.2	1173	13.9	9.5	22.2	21.9

Table 11. Requirement of coding memory for different transform-based compression algorithms at three HS image sizes (in KB)

3D-LSK	3D-NL	3D-LMBTC	3D-ZM-SPECK	3D-BP-ZM-SPECK	3D-M-ZM-SPECK	3D-LCBTC	3D-LEZSPC	3D-MELS	3D-LMZC	FrWF based ZM-SPECK	BFrWF based ZM-SPECK	3D-LBCSPC	Proposed Compression Algorithm
[64]	[66]	[68]	[69]	[37]	[36]	[34]	[56]	[40]	[88]	[70]	[89]	[90]	
128 x 128 x 128 (Size of HS image under test)													
512	1024	12	0	0	0	300.5	-	128	-	0	0	-	18.25
256 x 256 x 256													
4096	8192	96	0	0	0	2318	2304	1024	2176	0	0	4094	146
512 x 512 x 512													
32768	65568	768	0	0	0	18544	-	8192	17408	0	0	32752	1168

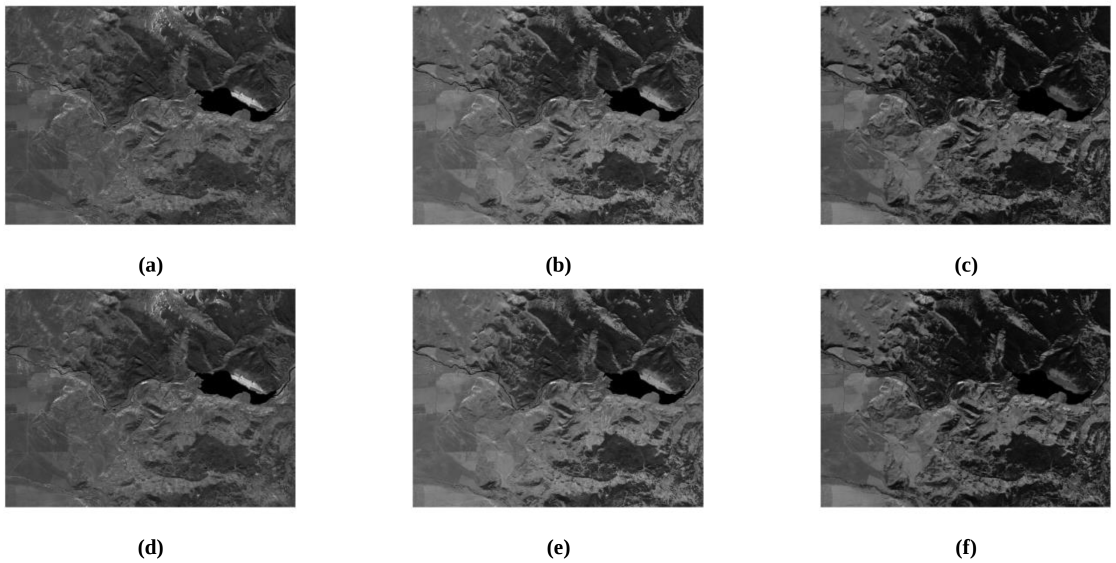

Fig. 2 Original uncalibrated yellowstone scene 00, (a) Band 48, (b) Band 98, (c) Band 148 reconstructed uncalibrated yellowstone scene 00 with CR=16, (d) Band 48, (e) Band 98, and (f) Band 148.

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] 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]

[3] 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]

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

[5] Claudia Defrasne et al., “The Contribution of VNIR and SWIR Hyperspectral Imaging to Rock Art Studies: Example of the Otello Schematic Rock Art Site (Saint-Rémy-de-Provence, Bouches-du-Rhône, France),” *Archaeological and Anthropological Sciences*, vol. 15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Farideh Foroozandeh Shahraki et al., *Deep Learning for Hyperspectral Image Analysis, Part II: Applications to Remote Sensing and Biomedicine*, Hyperspectral Image Analysis, Springer, Cham, pp. 69-115, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Lingxi Liu et al., “Neural Networks for Hyperspectral Imaging of Historical Paintings: A Practical Review,” *Sensors*, vol. 23, no. 5, pp. 1-25, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Payal Bhadra, Avijit Balabantaray, and Ajit Kumar Pasayat, “MFEMANet: An Effective Disaster Image Classification Approach for Practical Risk Assessment,” *Machine Vision and Applications*, vol. 34, 2023. [CrossRef] [Google Scholar] [Publisher Link]

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

[10] L. Singh et al., “Hyperspectral Remote Sensing for Foliar Nutrient Detection in Forestry: A Near-Infrared Perspective,” *Remote Sensing Applications: Society and Environment*, vol. 25, 2022. [CrossRef] [Google Scholar] [Publisher Link]

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

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

[13] Aksel Alstad Mogstad, Geir Johnsen, and Martin Ludvigsen, “Shallow-Water Habitat Mapping Using Underwater Hyperspectral Imaging from an Unmanned Surface Vehicle: A Pilot Study,” *Remote Sensing*, vol. 11, no. 6, pp. 1-20, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Harshita Mangotra et al., “Hyperspectral Imaging for Early Diagnosis of Diseases: A Review,” *Expert Systems*, vol. 40, no. 8, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] 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]

[16] Jayasimha Chilakamarri, Rama Rao Nidamanuri, and Palani Murugan, "Multi-Scenario Target Detection Using Neural Networks on Hyperspectral Imagery," *2023 International Conference on Machine Intelligence for GeoAnalytics and Remote Sensing (MIGARS)*, Hyderabad, India, pp. 1-4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] C. Deepa, Amba Shetty, and A.V. Narasimhadhan, "Performance Evaluation of Dimensionality Reduction Techniques on Hyperspectral Data for Mineral Exploration," *Earth Science Informatics*, vol. 16, pp. 25-36, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Paul Linton et al., *The Application of Hyperspectral Core Imaging for Oil and Gas*, Geological Society, London, vol. 527, 2023. [CrossRef] [Google Scholar] [Publisher Link]

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

[20] 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*, Kannur, India, pp. 94-98, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Zainab Zaman, Saad Bin Ahmed, and Muhammad Imran Malik, "Analysis of Hyperspectral Data to Develop an Approach for Document Images," *Sensors*, vol. 23, no. 15, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Siyoon Kwon et al., "Unsupervised Classification of Riverbed Types for Bathymetry Mapping in Shallow Rivers Using UAV-Based Hyperspectral Imagery," *Remote Sensing*, vol. 15, no. 11, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] 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]

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

[25] Wenqian Dong et al., "Abundance Matrix Correlation Analysis Network Based on Hierarchical Multihead Self-Cross-Hybrid Attention for Hyperspectral Change Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Jaime Zabalza et al., "Singular Spectrum Analysis for Effective Feature Extraction in Hyperspectral Imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1886-1890, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] 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]

[28] Reaya Grewal, Singara Singh Kasana, and Geeta Kasana, "Hyperspectral Image Segmentation: A Comprehensive Survey," *Multimedia Tools and Applications*, vol. 82, pp. 20819-20872, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] 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]

[30] Orhan Torun et al., "Hyperspectral Image Denoising via Self-Modulating Convolutional Neural Networks," *Signal Processing*, vol. 214, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[31] Lingling Zhang et al., "Near-Infrared II Hyperspectral Imaging Improves the Accuracy of Pathological Sampling of Multiple Cancer Types," *Laboratory Investigation*, vol. 103, no. 10, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Vinod Kumar, Ravi Shankar Singh, and Yaman Dua, "Morphologically Dilated Convolutional Neural Network for Hyperspectral Image Classification," *Signal Processing: Image Communication*, vol. 101, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Yaman Dua, Ravi Shankar Singh, and Vinod Kumar, "Compression of Multi-Temporal Hyperspectral Images Based on RLS Filter," *The Visual Computer*, vol. 38, pp. 65-75, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[34] Shrish Bajpai, "Low Complexity Block Tree Coding for Hyperspectral Image Sensors," *Multimedia Tools and Applications*, Vol. 81, no. 23, pp. 33205-33323, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[35] Divya Sharma et al., "112 Gb/s Coherent NG-PON2 Downstream Transmission Using Advance Polarization Multiplexed Modulation Formats," *Optoelectronics and Advanced Materials-Rapid Communications*, vol. 14, no. 5-6, pp. 224-232, 2020. [Google Scholar] [Publisher Link]

[36] 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]

[37] 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]

[38] 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]

[39] 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]

[40] 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]

[41] sEmmanuel Christophe, Corinne Mailhes, and Pierre Duhamel, “Hyperspectral Image Compression: Adapting SPIHT and EZW to Anisotropic 3-D Wavelet Coding,” *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2334-2346, 2008. [CrossRef] [Google Scholar] [Publisher Link]

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

[43] 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]

[44] 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]

[45] K. Subhash Babu et al., “Hyperspectral Image Compression Algorithms-A Review,” *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, pp. 127-138, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[46] Diego Valsesia, and Enrico Magli, “Fast and Lightweight Rate Control for Onboard Predictive Coding of Hyperspectral Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 3, pp. 394-398, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[47] Daniel Báscones, Carlos González, and Daniel Mozos, “An FPGA Accelerator for Real-Time Lossy Compression of Hyperspectral Images,” *Remote Sensing*, vol. 12, no. 16, pp. 1-20, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[48] Hyun S. Lee, Nicolas H. Younan, and Roger L. King, “Hyperspectral Image Cube Compression Combining JPEG-2000 and Spectral Decorrelation,” *IEEE International Geoscience and Remote Sensing Symposium*, Toronto, ON, Canada, vol. 6, pp. 3317-3319, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[49] K.S. Gunasheela, and H.S. Prasantha, “Compressive Sensing Approach to Satellite Hyperspectral Image Compression,” *Information and Communication Technology for Intelligent Systems*, pp. 495-503, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[50] 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]

[51] 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*, Montreal, QC, Canada, pp. 809-812, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[52] 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]

[53] Francesco Rizzo, Giovanni Motta, and James A. Storer, *Hyperspectral Data Compression*, Springer US, pp. 1-417, 2006. [Google Scholar] [Publisher Link]

[54] Monika Kumari, and Ajay Kaul, “Deep Learning Techniques for Remote Sensing Image Scene Classification: A Comprehensive Review, Current Challenges, and Future Directions,” *Concurrency and Computation: Practice and Experience*, vol. 35, no. 22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[55] Diego Valsesia, and Enrico Magli, “High-throughput Onboard Hyperspectral Image Compression with Ground-Based CNN Reconstruction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 9544-9553, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[56] James E. Fowler, and Justin T. Rucker, *Three-Dimensional Wavelet-Based Compression of Hyperspectral Imagery*, Hyperspectral Data Exploitation: Theory and Applications, pp. 379-407, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[57] Nor Rizuan Mat Noor, and Tanya Vladimirova, “Investigation into Lossless Hyperspectral Image Compression for Satellite Remote Sensing,” *International Journal of Remote Sensing*, vol. 34, no. 14, pp. 5072-5104, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[58] 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]

[59] Vinayak K. Bairagi, Ashok M. Sapkal, and M.S. Gaikwad, "The Role of Transforms in Image Compression," *Journal of the Institution of Engineers (INDIA): Series B*, vol. 94, pp. 135-140, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[60] Ali Bilgin, George Zweig, and Michael W. Marcellin, "Three-Dimensional Image Compression with Integer Wavelet Transforms," *Applied Optics*, vol. 39, no. 11, pp. 1799-1814, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[61] Lei Wang et al., "Hyperspectral Image Compression based on Lapped Transform and Tucker Decomposition," *Signal Processing: Image Communication*, vol. 36, pp. 63-69, 2015. [CrossRef] [Google Scholar] [Publisher Link]

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

[63] 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]

[64] 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]

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

[66] 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]

[67] 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]

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

[69] 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]

[70] Stephan Rein, and Martin Reisslein, "Performance Evaluation of the Fractional Wavelet Filter: A Low-Memory Image Wavelet Transform for Multimedia Sensor Networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 482-496, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[71] 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]

[72] 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]

[73] Mohd Tausif et al., "Lifting-Based Fractional Wavelet Filter: Energy-Efficient DWT Architecture for Low-Cost Wearable Sensors," *Advances in Multimedia*, vol. 2020, no. 1, pp. 1-13, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[74] 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]

[75] 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]

[76] B.K.N. Srinivasarao, and Indrajit Chakrabarti, "High Performance VLSI Architecture for 3-D Discrete Wavelet Transform," *2016 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, pp. 1-4, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[77] Aoife Keane et al., "Hyperspectral Imaging Analysis of Corrosion Products on Metals in the UV Range," *Hyperspectral Imaging and Applications II*, vol. 12338, pp. 44-53, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[78] Divya Sharma, Y.K. Prajapati, and Rajeev Tripathi, "0.55 Tb/s Heterogeneous Nyquist-WDM Superchannel Using Different Polarization Multiplexed Subcarriers," *Photonic Network Communications*, vol. 39, pp. 120-128, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[79] Garima Jaiswal et al., "Integration of Hyperspectral Imaging and Autoencoders: Benefits, Applications, Hyperparameter Tunning and Challenges," *Computer Science Review*, vol. 50, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[80] Christos Chrysafis, and Antonio Ortega, "Line-Based, Reduced Memory, Wavelet Image Compression," *IEEE Transactions on Image processing*, vol. 9, no. 3, pp. 378-389, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[81] Yiliang Bao, and C-CJ Kuo, "Design of Wavelet-Based Image Codec in Memory-Constrained Environment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 642-650, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[82] Wai Chong Chia et al., "Low Memory Image Stitching and Compression for WMSN Using Strip-Based Processing," *International Journal of Sensor Networks*, vol. 11, no. 1, pp. 22-32, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[83] Li Wern Chew et al., "Low-Memory Video Compression Architecture Using Strip-Based Processing for Implementation in Wireless Multimedia Sensor Networks," *International Journal of Sensor Networks*, vol. 11, no. 1, pp. 33-47, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[84] Mohd Tausif et al., "Low Memory Architectures of Fractional Wavelet Filter for Low-Cost Visual Sensors and Wearable Devices," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 6863-6871, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[85] Mohd Tausif, Ekram Khan, and Antonio Pinheiro, "Computationally Efficient Wavelet-Based Low Memory Image Coder for WMSNs/IoT," *Multidimensional Systems and Signal Processing*, vol. 34, pp. 657-680, 2023. [CrossRef] [Google Scholar] [Publisher Link]

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

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

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

[89] 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]

[90] 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]

[91] W.A. Pearlman et al., "Efficient, Low-Complexity Image Coding with a Set-Partitioning Embedded Block Coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219-1235, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[92] Mrityunjaya V. Latte, Narasimha H. Ayachit, and D.K. Deshpande, "Reduced Memory Listless Speck Image Compression," *Digital Signal Processing*, vol. 16, no. 6, pp. 817-824, 2006. [CrossRef] [Google Scholar] [Publisher Link]

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

[94] 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]

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

[96] 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]

[97] Adrian S. Lewis, and G. Knowles, "Image Compression Using the 2-D Wavelet Transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 244-250, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[98] Shigeru Muraki, "Volume Data and Wavelet Transforms," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 50-56, 1993. [CrossRef] [Google Scholar] [Publisher Link]

[99] Jaime Zabalza et al., "Fast Implementation of Two-Dimensional Singular Spectrum Analysis for Effective Data Classification in Hyperspectral Imaging," *Journal of the Franklin Institute*, vol. 355, no. 4, pp. 1733-1751, 2018. [CrossRef] [Google Scholar] [Publisher Link]