SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I1P115

Volume 13 Issue 1, 174-192, January 2026
© 2026 Seventh Sense Research Group®

Original Article

3D Shearlet Transform-based Block Cube Tree Coding for Resource Constraint Hyperspectral Image Sensors

Purushottam Lal Nagar[1], Shrish Bajpai[2]

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

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

Received: 08 November 2025 Revised: 10 December 2025 Accepted: 09 January 2026 Published: 14 January 2026

Abstract - *Compression algorithms are absolutely necessary for the effective storage and speedy transfer of remote imaging data. The present manuscript proposes a transform-based hyperspectral image compression algorithm that exploits both the inter- and intra-subband correlations among the transform coefficients. The compression algorithm is based on the Spatial Oriented Trees (SOTs), which are the basic unit in block cubes. In contrast to the hierarchical tree compression approach, which only uses a single coefficient for 3D set partitioning, the block cube data structure takes the form of a cube and has the coefficients m*m*m. The root node of each SOT is located in the LLL band, while child and descendant blocks are located in the high-frequency sub-band. The proposed compression algorithm exploits the best features of the zeroblock cube and zerotree compression algorithms.*

Keywords - *Transform Coding, Hyperspectral Image Compression, Shearlet transform, Transform coefficients, Coding efficiency.*

1. Introduction

The integration of two-dimensional plane imaging and spectroscopy is made possible by Hyperspectral (HS) imaging, which allows for the recording of spectral diagrams and signatures as well as the spatial distribution of objects within an area of interest [1, 2]. Spectroscopy can extract the individual spectrum of each pixel from the visible to near-infrared wavelength range (four hundred nanometres to twenty-five hundred nanometres, having a spectral spacing of two nanometres to ten nanometres) and divide it into a large number of continuous narrow spectral bands [3, 4]. This makes it easier to characterise pixels that have the same spectral signature [5]. Due to this vast information, HS images are used in many applications such as Aerospace [6], Agriculture [7], Archaeology [8], Biotechnology [9], Climate Monitoring [10], Document Verification [11], Environment Monitoring [12], Food Quality Measurement [13], Geology [14], Human Health Care [15], Infection Detection (Plant) [16], Military Reconnaissance [17], Mineralogy [18], Oceanography [19], Pharmaceuticals [20] etc. Besides the above-mentioned applications (Fields), remote sensing is a field that is expanding at the quickest rate [21]. The scientists/researchers are developing the computer-based algorithms for Change Detection [22], Classification [23], Compression [24], Denoising [25], Fusion [26], Feature Extraction [27], Feature Identification [28], Inpainting [29], Segmentation [30], and Unmixing [31] of HS images. HS images are often acquired by satellite-based HS image sensors, and after that, they are transmitted to an earth station via radio channels [32]. Because every HS image has both spatial and spectral dimensions, the datasets associated with HS images typically have a very high size [33]. The third dimension of an HS image is called "Wavelength," while the third dimension of video data corresponds to the "Time" [34].

Hyperspectral data are comparable to video data in this regard. Due to the fact that a memory capacity of this magnitude is required, it is essential to eliminate any unnecessary redundancy prior to the high-speed image's transmission to the ground station [35]. HS image has two types of redundancy (Spectral and Spatial) that need to be reduced or eliminated to achieve compression of any HS image [36]. Compression of the HS image is required before the transmission of the image data to save the onboard memory storage, reduce transmission bandwidth, lower coding complexity (processing speed), and save image sensor energy [37]. The HS image Compression Ratio (CR) is defined as a measurement of the relative reduction in the size of reconstructed HS images by the Hyper Spectral Image Compression Algorithm (HSICA) [38]. It is defined as in Equation 1.

$$CR = \frac{Bits\ per\ pixel\ in\ HS\ image\ before\ compression}{Bits\ per\ pixel\ in\ \ HS\ image\ after\ compression} \tag{1}$$

HSICAs can be split into the different categories either on the basis of data loss or coding process (compression process execution) [35]. HSICAs are split into three categories: lossless, near lossless, or lossy compression [36]. There is no data loss in lossless compression, while CR has a low numeric value that ranges from 1 to 3. In the same way, near-lossless compression has very minimal data loss, and CR has a slightly higher value than the lossless compression process. The lossy compression has a very high CR, while there is a significant loss of image data [39].

HSICAs are divided into six categories named as Transform Coding (TC) based HSICA [40], Predictive Coding (PC) based HSICA [41], Vector Quantization (VQ) based HSICA [42], Compressive Sensing (CS) based HSICA [43], Tensor Decomposition (TD) based HSICA [44], Learning-based Compression (LC) based HSICA [45], and Neural Network (NN) based HSICA [46].

The PC based HSICA uses a different predictor to determine the next (future) pixel value through prediction error. Prediction errors are encoded using several methods of entropy coding, such as Huffman coding and arithmetic coding, amongst others. The prediction-based methods are reliable with data, and these algorithms can only be used with lossless compression [41].

The VQ-based HSICA, also known as a dictionary-based method, uses the codebook that is present at the transmitter (encoder) and receiver (decoder) ends. The compression algorithm divides the HS image into small blocks at the encoder end and gives a unique code according to the codebook. The unique code had been used for the transmission. The decoder uses the received code word to reconstruct the HS image. If any error occurs during the transmission process, the whole reconstructed HS image can be distorted [42].

The performance of TD-based HSICAs is high, and this high performance is not dependent on the data type of the HS image. The tensor is an easily decomposable three-dimensional matrix. A 3D tensor representing the HS image is decomposed into lower-dimensional tensors via the technique. The channel carries an encoded version of these tensors in lower dimensions [44].

The three-stage compression is carried out by CS-based HSICAs. The encoder initially takes the HS image and, using a minimal number of samples, transforms the resulting 3D HS image into a compact 2D matrix. In the next step, this matrix is reduced in size by following the procedures outlined in the CS-based HSICA, and the data is then sent through the transmission channel. This method is repeated as many times as necessary until the entire HS image has been sent. The decoder is responsible for recreating the HS image using the entirety of the data that was received via the transmission channel [43].

The compression process is carried out with the help of Machine Learning (ML) and Deep Learning (DL) using LC-based HSICAs. In conjunction with the LC process, the PC application is carried out. Although a high compression ratio was attained, it came at the expense of increased complexity and utility [45].

Neural Networks (NN) such as autoencoders, Support Vector Machines (SVM), backpropagation networks, and Convolutional Neural Networks (CNN), have been used to minimize the dimensionality of HS images, convert HS images to compressed binary codes for retrieval, and to extract compact visual representations. Every NN has multiple layers of structures. Outstanding predictive capability may be found in the NN network that makes use of predictive coding. The coding gain improves as a result of a reduction in prediction error [46].

The TC-based HSICAs utilize a mathematical transform to transform the image data into a domain where the data is represented by the less correlated high-energy coefficients. Both lossy and lossless compression methods are supported by transform-based HS image compression techniques. When it comes to lossy compression, these compression algorithms work admirably even at low bit rates. The HS image's spatial and spectral correlation has been removed thanks to the mathematical transform. Low-frequency coefficients are compressed to hold the full power of the HS image [40].

The remaining manuscript is organized as follows: Section 2 gives theoretical background details of the proposed compression algorithm, while Section 3 sheds light on the motivation for the development of the compression algorithm, with the major Contribution in the development of the compression algorithm. Section 4 proposed the 3D Shearlet Transform-based Block Cube Tree Coding (3D-STBCTC). Result and discussion, including the detailed comparative analysis with other state-of-the-art compression algorithms, were explained in Section 5. Conclusion and future scope are covered in the last part of the manuscript.

2. Theoretical Background

The transform-based HSICA uses a mathematical transform (Fourier, cosine, dyadic wavelet, curvelet, fractional wavelet filter, etc) to convert the time domain HS image to the frequency domain HS image [34, 47]. They use the properties of the transform to identify the significant coefficients. Among all transform-based HSICAs, set partitioned (also known as bit plane coding) HSICAs are a special type of compression algorithms that use the set structure of the transform image to define the significant/insignificant coefficients [35, 48]. The insignificant coefficients are either arranged in the zero block cube or zero tree or zero block cube tree fashion. It has been known that at the top bit plane, there are a lot of insignificant coefficients, and these compression algorithms use a single ‘0’ to define a lot of insignificant coefficients [35]. Tang et al. proposed 3D-Set Partitioned Embedded Block (3D-SPECK), which 3D zero block cube structure to define the insignificant coefficients [49]. In the same way, 3D-Set Partitioning In Hierarchical Trees (3D-SPIHT) uses the zerotree structure to define the insignificant coefficients [50]. There are many modified versions of the 3D-SPIHT, and 3D-SPECK has been proposed in the past. 3D-Wavelet Block Tree Coding (3D-WBTC) uses the features of 3D-SPECK and 3D-SPIHT and shows the high coding gain at the low bit rates [51]. These HSICA uses a linked list for tracking the significance/insignificance of the sets or coefficients. These compression algorithms have higher coding gain, but they have high coding complexity and huge coding memory demand. These issues are addressed by the listless HSICAs.

Instead of the linked lists, these compression algorithms use the marker and state table to define the significance. The demand for memory for listless HSICA is fixed and only depends on the size of the HS image [34]. The complexity is also reduced significantly because of a very small number of read/write operations happening in the listless compression process. 3D-Listless SPECK (3D-LSK) [52], 3D-No List SPIHT (3D-NLS) [53] and 3D-Low Memory Block Tree Coding (3D-LMBTC) [54] are the popular listless HSICAs. The Low Complexity Bock Tee Coding (3D-LCBTC) uses the two small lists and markers to define the state of the sets or coefficients [48]. Zero Memory Set Partitioned Embedded Block (3D-ZM-SPECK) employs linear indexing for identification of the sets and does not require any coding memory [55]. But it has higher coding complexity than 3D-LSK. The issue of coding complexity had been solved by using parallel processing.

The whole transformed HS image is divided into small block cubes, and ZM-SPECK is applied to each block in a parallel processing way. Through this, the complexity of the compression algorithm is reduced. 3D-Modified Zero Memory Set Partitioned Embedded Block (3D-M-ZM-SPECK) also solves the issue of complexity, but it requires some coding memory [35]. 3D-Block Cube-based Parallel Modified Zero Memory Set Partitioned Embedded Block (3D-BCP-ZM-SPECK) reduces the complexity of 3D-ZM-SPECK. Chandra and Bajpai proposed 3D-Block Partitioning Embedded Coding (3D-BPEC), which is based on an array instead of lists and markers [56]. Recently, Bajpai and Kidwai proposed a Fractional wavelet filter-based HSICA, which reduces the transform memory, but it had a very poor error resilience capability [57]. Spatial-orientation Tree Wavelet (STW) uses a hybrid mathematical transform to achieve the compression of the HS image [58].

Table 1 gives the comparative analysis (performance metric, advantage, and limitation) between different mathematical transform-based HSICAs. Algorithm Performance Metric (IPM) used in the HSICA mentioned in Table 1 are Peak Signal to Noise Ratio (IPM I), Structural Similarity Index (APM II), feature-similarity index (APM III), Bjontegaard Delta PSNR (APM IV), Coding Memory (APM V), and Coding Complexity (APM VI) [38].

2.1. Shearlet Transform

Despite their widespread use, classical wavelet approximations are for one-dimensional piecewise continuous functions, but they have poor performance in the representation of the edges present in the different objects present in the HS image. But, now multidimensional mathematical transforms such as curvelet, ripplet, shearlet, etc, have efficient representation of multivariate functions with spatially distributed discontinuities [59]. The shearlet transform was introduced by Negi and Labate as a multidimensional image representation tool. It has been able to show bivariate functions sparsely. It has been known that classical wavelet has optimal approximations with only one direction having a piecewise continuous function [60, 62].

3. Motivation and Contribution

In the past, many compression algorithms were proposed for lossy and lossless compression, but they suffer from low coding efficiency, high coding complexity, or high coding memory. The list-based HSICAs have high coding efficiency, but they suffer from exponentially rising complexity and coding memory requirements with the increasing bit rate. To reduce the coding memory and coding complexity, list-based HSICAs are proposed. 3D-LSK, 3D-NLS, and 3D-LCBTC reduce the complexity but still have high fixed-size coding memory. 3D-LMBTC required a small piece of coding memory with a cost of 0.5 dB to 1 dB loss of coding efficiency. Further, 3D-ZM-SPECK needs no coding memory due to the use of linear indexing, but has a higher coding complexity than 3D-LSK. 3D-BCP-ZM-SPECK and 3D-M-ZM-SPECK further reduce coding complexity at the cost of coding efficiency. The coding gain can increase with the use of another mathematical transform. Curvelet transform and shearlet transform are the two other mathematical transformations that can show good results for image compression. The curvelet transform needs fewer coefficients to represent the non-singularities (edge, curve, etc). 3D-Listless Embedded Zerotree Set Partitioning Coding (3D-LEZSPC) uses curvelet transform and gives superior performance than other wavelet transform-based HSICA. But the shearlet transform can also work for the geometrical features of the HS images. With the block cube tree coding and shearlet transform, the coding efficiency increased with reference to the other state-of-the-art HSICA.

3.1. Major Contribution of this Manuscript

The major contributions to the manuscript are listed below.

Table 1. Mathematical transform-based hyperspectral compression algorithms

HSICA	APM	Advantage	Limitation
3D-SPIHT [50]	I	Uses the block tree structure to obtain compression	Complexity increases exponentially with an increase in bit rate
3D-Trap Coding [69]	I	Low implementation complexity on the hardware	Poor coding efficiency at the high bit rates
3D-SPECK [49]	I	Uses the block cube structure to obtain compression	Not fit for the low-bit-rate HS image compression
3D SPEZBC [70]	I	It had a better rate distortion performance than other HSICAs	Low coding memory requirement compared to other list-based HSICA
Anisotropic SPIHT [71]	I	Coding Efficiency increased due to the classical isotropic decomposition	Complexity is very high compared to the listless HSICA
3-D Wavelet Fractal [72]	I	Wavelet-Fractal coding increases coding efficiency	Two different transform mechanisms increase the load on the sensor
3D-LSK [52]	I	Coding complexity reduces significantly for any bit rate	Coding memory is very high for the small-sized HS image
3D-NLS [53]	I, II	Demand for coding memory is reduced significantly compared to 3D-SPIHT	Coding memory is higher than 3D-LSK
3D-WBTC [51]	I, V, VI	Optimized performance at the low bit rates	Coding complexity is very high for high bit rates
3D-LMBTC [54]	I, V, VI	Low memory lossy compression algorithm	Coding complexity is higher than 3D-LSK and 3D-NLS
3D-STW [58]	I	High coding efficiency is achieved with the use of multiple transforms	The requirement of the coding memory is huge during the transformation process
3D-ZM-SPECK [55]	I, II, IV. V, VI	Zero coding memory is required for compression	Ineffective increase in computation complexity compared to 3D-LSK
3D-LCBTC [48]	I, II, III, IV, V, VI	Reduce coding complexity than 3D-LMBTC	Very high coding memory requirement compared to 3D-LMBTC
3D-LEZSPC [59]	I, II, III, IV, V, VI	High coding efficiency is due to the use of the curvelet transform	Demand for transform memory is very high in the curvelet transform
3D-BPEC [56]	I, VI	Complexity reduces significantly from 3D-WBTC	Complexity increases with an increase in bit rate
FrWF-Based ZM-SPECK [57]	I, II, V, VI	Spectral correlation exploited by the prediction	Very poor error resilience capability during the transmission process
3D-BCP-ZM-PECK [61]	I, II, III, IV, V, VI	Reduce the complexity of the 3D-ZM-SPECK	Effect on the coding efficiency of the 3D-ZM-SPECK
3D-M-ZM-PECK [35]	I, II, III, IV, V, VI	Reduce the complexity of the 3D-ZM-SPECK	Small coding memory is also required with reference to 3D-ZM-SPECK
3D-LBCSPC [34]	I, II, III, IV, V, VI	Reduce the complexity of the zeoblock compression algorithm	High complexity with the use of spatial trees
3D-SLS [66]	I, II, III, IV, V, VI	Reduce the number of lists based on 3D-SPIHT	Still, coding memory is high with an increase in the bit rate

- The proposed HSICA uses the Shearlet Transform to achieve the high coding gain.
- It uses the Zero Block Cube Tree structures to achieve a high compression ratio at a low data rate.

4. 3D Shearlet Transform Block Cube Tree Coding (3D-STBCTC)

The 3D Shearlet Transform Block Cube Tree Coding compression algorithm combines the good features of both zero-tree-based HSICA and zero block cube-based HSICA. It partitions the transformed HS image into coefficient block cubes and then constructs trees of block cubes with roots in the topmost sub-band in a tree fashion. In a block cube tree, significant block cubes are found using the tree partitioning concept of 3D-SPIHT [50], whereas significant coefficients within each block cube are found using the octa-tree partitioning of 3D-SPECK [49]. A significant block cube tree is recursively partitioned until significant coefficients are found, till a single coefficient level or bit budget is available. Like 3D-SPIHT and 3D-SPECK, 3D-STBCTC is also a bit plane compression scheme.

4.1. Compression Algorithm

Let us consider an HS image data of 'M x M x M' coefficients after taking the 3D shearlet transform, which transforms the image into a pyramidal sub-band structure. The transformed HS image is characterized by an indexed set of transform coefficients as $C_{\alpha,\ \beta,\ \gamma}$ with row index 'α' and column index 'β' of frame 'γ'. The coefficients are grouped in block cubes of size 'δ x δ x δ' coefficients, and then block cube trees are formed with roots in the topmost (LLL) sub-band situated at the left-hand side of the transform HS image. This block cube tree structure has three distinct advantages over the tree structure used in 3D-SPIHT.

1. It merges a large number of 3D-SPIHT clustered zero trees, which may appear in the early passes, to produce zero trees with more coefficients.
   - Zero-block can be used to describe a collection of 'δ x δ x δ' insignificant coefficients for that bit plane. This will enable more effective and less complex frame-to-frame exploitation of intra-subband correlations.
2. Because of its block cube-based design, encoding time and memory requirements are both dramatically decreased at low bit rates compared to pixel (coefficient) based approaches.

Within the 3D sub-band hierarchy, each block cube (with the exception of those in the highest resolution band) has eight progeny cubes occupying the corresponding spatial location in the next higher frequency sub-band. Two specific cases exist: firstly, the lowest frequency band (LLL) is structured as groups of '2×2×2' cubes. Within each group, the top-left cube is a leaf node with no descendants, while the other seven cubes each have eight progeny in higher-frequency sub-bands. Secondly, all cubes in the highest resolution band are leaf nodes with no progeny.

The associated notations and nomenclatures mentioned in Table 2 are used in the 3D-STBCTC.

Many of the Spatial Orientation Trees (SOTs) generated by 3D-SPIHT are integrated into one single spatial orientation block cube tree in 3D-STBCTC through the process of producing a block cube tree. The set of all descendant block cubes is referred to as a type 'A' block cube tree. On the other hand, the set of grand descendant block cubes is referred to as a type 'B' block cube tree. This is calculated by subtracting the set of descendant block cubes from the set of offspring block cubes. In particular, for a block cube size of '2x2x2', eight SOTs of 3D-SPIHT are combined into a single 3D-STBCTC SoT. The significant information about the wavelet coefficients is stored in the three ordered lists:

- List of Insignificant Block Cubes (LIBC)
- List of Insignificant Block Cube Sets (LIBCS)
- List of Significant Pixels (LSP)

At the first step of the proposed algorithm, all block cubes in the LLL band are added to LIBC, and others with their descendants are added to LIBCS as type 'A' entries. LSP starts as an empty list. In the same way as 3D-SPIHT and 3D-WBTC, the proposed compression algorithm 3D-STBCTC is also bit plane coding, having two sub-stages named as sorting pass and refinement pass

The 3D-STBCTC algorithm employs a top-down, bit-plane encoding strategy, commencing with the most significant bit plane. The initial step in the sorting pass involves traversing the List of Insignificant Block Cubes (LIBC). Each block cube within the LIBC is evaluated for its significance against the prevailing threshold. This test generates a single binary output. '1' is emitted if the block becomes significant, prompting its promotion to the LIS (List of Insignificant Sets) for further partitioning. Conversely, a '0' indicates the block remains insignificant at the current threshold; it is retained in the LIBC, and no additional bits are allocated to it for that bit plane.

So, information about the 'm x m x m' insignificant coefficients for the current threshold is sent by a single '0' bit, whereas 3D-SPIHT generates 'α×α×α', '0' bits.

The coding procedure for a threshold pass is as follows:

1. Block Cube Testing: Each block cube in the dominant list is tested for significance.
   a. Insignificant: Encoded with a single '0' bit.
   b. Significant: Encoded with a '1' bit and partitioned into eight sub-cubes via octree partitioning.
2. Recursive Partitioning: Step 1 is repeated recursively on significant sub-cubes until the individual coefficient level is reached.
3. Coefficient-Level Processing: For a cube of individual coefficients:
   (a) Each coefficient is tested. A '0' denotes an insignificant coefficient, which is added to the List of Insignificant Block Cubes (LIBC). A '1' denotes a significant coefficient, and its sign bit is encoded; the coefficient is then moved to the List of Significant Pixels (LSP).
   (b) The processed block cube is deleted from the LIBC.
4. Set Processing: The algorithm then processes sets of block cubes in the List of Insignificant Block Cube Sets (LIBCS), partitioning significant sets into smaller subsets.

A significant type 'A' set with a root block cube $B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}$ is partitioned into a type 'B' set $L_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}$ with eight offspring's block cubes $O_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}$. Following the partitioning of a significant block cube, the resulting offspring are immediately assessed for their significance. The type 'B' set is appended to the List of Insignificant Block Cube Sets (LIBCS) for subsequent processing. When a type 'B' set is found to be significant, it is subdivided into eight type 'A' sets, which are also added to the end of the LIBCS. This iterative process continues for a given threshold, with all newly generated insignificant sets being queued at the end of the LIBCS until every set has been examined. Upon completing this sorting pass, a refinement pass is executed. This pass encodes the next bit of precision for coefficients already in the List of Significant Pixels (LSP) from previous thresholds, excluding those newly added in the current sorting pass. The entire cycle sorting followed by refinement is then repeated with the threshold halved until the target bitrate is attained. The proposed algorithm demonstrates superior rate distortion performance compared to 3D-SPIHT, particularly at low bitrates. The architecture of the proposed system (HSICA) is depicted in Figure 1, with the corresponding pseudo-code detailed in Table 3.

4.2. Coding Complexity of Proposed Compression Algorithm

The coding complexity of any HSICA depends upon the number of mathematical, arithmetical, and logical competitions and how complex these operations are. It has been known that list-based HSICA has high coding complexity, which is due to accessing the coefficients from the different lists. At low bit rates, these list-based HSICAs have comparable complexity, but with the increased bit rates, the coding complexity increases exponentially. There are three ways to reduce the coding complexity, which are the use of markers, the use of state tables, or the use of the 1D array system. But when these ways are used, there is a significant reduction in the coding efficiency. Thus, for the high coding efficiency, the list-based compression algorithm performs best. With the use of advanced mathematical transforms, the coding efficiency also increased. The coding complexity of 3D-STBCTC is slightly higher than that of 3D-SPECK, 3D-SPIHT, and 3D-WBTC.

5. Result and Discussion

The performance of the proposed compression algorithm was evaluated on four HS images and benchmarked against state-of-the-art transform-based HSICAs. The detailed descriptions of the simulation test images are summarized in Table 4.

The comparative analysis of the proposed compression algorithm was conducted on the basis of coding efficiency, coding memory, and coding complexity. The Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Feature-Similarity (FSIM) index are used to define the coding efficiency of any HS image [73, 74]. PSNR is mentioned as Image Quality Metric I (IQM I), SSIM is mentioned as Image Quality Metric II (IQM II), and FSIM is mentioned as Image Quality Metric III (IQM III) in the manuscript.

The original HS image is transformed with the 3D shearlet transform. The transform 3D HS image is mapped to a 1D array with the help of Morton Mapping (Linear Indexing) [63]. The transform coefficients of the 1D array have been encoded with the compression algorithm. After the compression of the HS image, it is converted to the 3D matrix [66-68]. The inverse transform is applied to the 3D matrix, and the coefficients are quantized to the nearest integers. In order to obtain an accurate assessment of the proposed 3D-STBCTC in relation to the other HSICAs, the compression algorithms have been programmed on the numerical computing software Matlab, which is running on a computer terminal running Windows 11 and containing 20 GB of storage space.

The 3D-STBCTC is compared with 3D-SPECK (Compression Algorithm I) [49], 3D-SPIHT (Compression Algorithm II) [50], 3D-WBTC (Compression Algorithm III) [51], 3D-LSK (Compression Algorithm IV) [52], 3D-NLS (Compression Algorithm V) [53], 3D-LMBTC (Compression Algorithm VI) [54], 3D-LCBTC (Compression Algorithm VII) [48], 3D-ZM-SPECK (Compression Algorithm VIII) [55] and 3D-LEZSPC (Compression Algorithm IX) [59].

5.1. Coding Efficiency

To quantify the Rate-Distortion (RD) performance of the compression algorithm, the Peak Signal-to-Noise Ratio (PSNR) is employed. The original and reconstructed Hyperspectral (HS) images are defined as A (x,y,z) and B (x,y,z). Given the Total Number Of Coefficients (pixel), Npix, the Mean Squared Error (MSE) is computed as shown in Equation 3. The PSNR is subsequently calculated based on this MSE value.

Table 2. Notations used in the proposed 3D-STBCTC

•	$B^{\alpha*\alpha*\alpha}_{(\eta,\sigma,\rho)}$	A block cube with dimensions of ‘m’ and transform coefficients is arranged in a 1D array form. $\{c_{\alpha,\beta,\gamma}\ :\ \eta \le \alpha\ \le\ (\eta+\delta)\ ,\ \sigma \le \beta\ \le\ (\sigma+\delta)\ ,\ \rho \le \gamma\ \le\ (\rho+\delta)\}$ Where (η,σ,ρ) are the coordinates of the top left coefficient of the block cube. The coordinate of the block cube's top-left corner is used to identify each individual block cube.
•	$Q_{(\eta,\sigma,\rho)}$	Set of all octa-block cubes of a parent block cube $B^{\delta*\delta*\delta}_{\eta,\sigma,\rho}$ It is defined as

		$Q_{(\eta,\sigma,\rho)}$ =	{ $B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta,\sigma,\rho}$ $B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta,\sigma,\rho+\frac{\delta}{2}}$
			$B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta,\sigma+\frac{\delta}{2},\rho}$ $B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta,\sigma+\frac{\delta}{2},\rho+\frac{\delta}{2}}$
			$B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta+\frac{\delta}{2},\sigma,\rho}$ $B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta+\frac{\delta}{2},\sigma,\rho+\frac{\delta}{2}}$
			$B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta+\frac{\delta}{2},\sigma+\frac{\delta}{2},\rho}$ $B^{\frac{\delta}{2}*\frac{\delta}{2}*\frac{\delta}{2}}_{\eta+\frac{\delta}{2},\sigma+\frac{\delta}{2},\rho+\frac{\delta}{2}}$ }
•	H	Block cubes in the LLL band (lowest resolution)	
•	$O_{(\eta,\sigma,\rho)}$	Set of all offspring block cubes of the root block cube. It is defined as	
		$O_{(\eta,\sigma,\rho)}$ =	{ $B^{\delta*\delta*\delta}_{2\eta,2\sigma,2\rho}$ $B^{\delta*\delta*\delta}_{2\eta,2\sigma,2\rho+\delta}$
			$B^{\delta*\delta*\delta}_{2\eta,2\sigma+\delta,2\rho}$ $B^{\delta*\delta*\delta}_{2\eta,2\sigma+\delta,2\rho+\delta}$
			$B^{\delta*\delta*\delta}_{2\eta+\delta,2\sigma,2\rho}$ $B^{\delta*\delta*\delta}_{2\eta+\delta,2\sigma,2\rho+\delta}$
			$B^{\delta*\delta*\delta}_{2\eta,+\delta,2\sigma+\delta,2r}$ $B^{\delta*\delta*\delta}_{2\eta+\delta,2\sigma+\delta,2\rho+\delta}$ }
•	$D_{(\eta,\sigma,\rho)}$	Set of all descendant blocks of root block cube $B^{\delta*\delta*\delta}_{\eta,\sigma,\rho}$	
•	$L_{(\eta,\sigma,\rho)}$	Set of all descendants except immediate offspring of the root block cube $B^{\delta*\delta*\delta}_{\eta,\sigma,\rho}$	
•	S_n (.)	significant function in the nth most significant bit plane applied to set T, defined as	
		$S_n(T) = \begin{cases} 1 \;;\; if \max(\{\lvert c_{\eta,\sigma,\rho}\rvert\} \geq 2^n) \bigvee C_{i\eta,\sigma,\rho} \in T \\ 0 \;;\; otherwise \end{cases}$ Where set T may either be an individual block cube of α*α*α wavelet coefficients or a block cube tree. For a given n, if $S_\alpha\{T\}$ = 1, then set T is said to be significant; otherwise, it is insignificant.	

Table 3. Pseudo code for encoder of 3D shearlet transform block cube tree coding

```
1.  Initialization
        Output $n = \lfloor \log_2 [max_{(i,j,k)} \{|C_{(\eta,\sigma,\rho)}|\}] \rfloor$
        LSP = ϕ
        Add all $B^{\alpha*\alpha*\alpha}_{\eta,\sigma,\rho} \in H_{(\eta,\sigma,\rho)}$ To LIBC, and those with descendants also to LIBCS, as type 'A' entries.
2.  Sorting Pass
 /* LIBC testing */
            For each entry $B^{\alpha*\alpha*\alpha}_{\eta,\sigma,\rho}$ in LIBC
            do:
                •   Output $S_n[B^{\alpha*\alpha*\alpha}_{\eta,\sigma,\rho}]$
                        ▪  if $S_n[B^{\alpha*\alpha*\alpha}_{\eta,\sigma,\rho}] = 1$ then,
                           if (α ≠ 1 call fn_bcp()
                           else
                           add (η, σ, ρ) to LSP, and output the sign of $c_{(\eta,\sigma,\rho)}$
                           Remove entry $B^{\alpha*\alpha*\alpha}_{\eta,\sigma,\rho}$ from the LIBC
        /* Testing of LIBCS  list*/

            For each entry $B^{\alpha*\alpha*\alpha}_{\eta,\sigma,\rho}$ in LIBC
            do:
                    ➢  If the entry is of type 'A '; then

                •       Output $S_n [D_{(\eta,\sigma,\rho)}]$
                            if $S_n [D_{(\eta,\sigma,\rho)}] = 1$       then,
                    For each $B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha} \in O_{(\eta,\sigma,\rho)}$
                    do
                    Output $S_n\left[B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}\right]$
                    if $S_n\left[B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}\right] = 1$ ; then call fn_bcp()
                    else add $B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}$ to the end of LIBC
                    if $L_{(\eta,\sigma,\rho)} \neq \phi$ , then
                    Change the type of $B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}$ from type 'A' to type 'B.'
               ➢  if the entry is of type 'B';
                  then
             •      Output $S_n\left[L_{(\eta,\sigma,\rho)}\right]$
             •      if Output $S_n\left[L_{(\eta,\sigma,\rho)}\right] = 1$
                  then
                  add each $\left[B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}\right] \in O_{(\eta,\sigma,\rho)}$ to the end of the LIBCS

3.  Refinement Pass
    For every entry $(\alpha, \beta, \gamma)$ in the List of Significant Pixels (LSP), except those coefficients included in
    the last sorting pass (i.e., with the current bit plane)
    Output the 'n' most significant bit of $\left|c_{(\alpha,\beta,\gamma)}\right|$

4.  Quantization Step
    The counter n is decremented by one, and the algorithm returns to Step 2. This iterative process
    continues until the termination condition n = 0 is met or till the bit budget is exhausted.

fn_bcp()
    { /* Function for block cube partitioning */
    For a block cube $B_{\eta,\sigma,\rho}^{\alpha*\alpha*\alpha}$
    Do.
        ○  For each $B_{\eta,\sigma,\rho}^{v*v*v} \in O_{(\eta,\sigma,\rho)}$
        Do :
                ❖  if $(v \neq 1)$
                   Output $S_n\left[B_{\eta,\sigma,\rho}^{v*v*v}\right]$
                   if $S_n\left[B_{\eta,\sigma,\rho}^{v*v*v}\right] = 1$ then
                   call fn_bcp()
                   else
                   add $B_{\eta,\sigma,\rho}^{v*v*v}$ to the LIBC
                ❖  else
                   Output $S_n\left[c_{(\eta,\sigma,\rho)}\right]$
                   if
                   Output $S_n\left[c_{(\eta,\sigma,\rho)}\right] = 1$ then
                   add (η, σ, ρ) to LSP and output the sign of the coefficient
                   else
                   add $B_{\eta,\sigma,\rho}^{v*v*v}$ to LIBC as a single pixel block cube
    }
```

Table 4. Details of HS images used for the simulation test

HS Image	HS Sensor	HS Image Dimensions	Pixel Depth
Washington DC Mall (HS Image I)	HYDICE	1280 × 307 × 191	14
Yellowstone Scene 0 (HS Image II)	AVIRIS	512 × 680 × 224	16
Yellowstone Scene 3 (HS Image III)	AVIRIS	512 × 680 × 224	16
Yellowstone Scene 18(HS Image IV)	AVIRIS	512 × 680 × 224	16

$$PSNR = 10\ \log_{10}\left(\frac{MAX_A\ *\ MAX_A}{MSE}\right) \qquad (2)$$

$$MSE = \frac{1}{N_{pix}} \sum_z \sum_y \sum_z [A(x,y,z) - \ B(x,y,z)]^2 \qquad (3)$$

The 3D-STBCTC uses the same partition rule as 3D-WBTC, 3D-LMBTC, and 3D-LCBTC (zeroblock cube tree set partitioned). Table 5 gives the comparative analysis of the PSNR. SSIM and FSIM. It is clear from Table 5 that 3D-STBCTC outperforms for almost every bit rate compared to other HSICA. The variation of the PSNR between 3D-STBCTC and 3D-SPECK is 0.69 dB to 1.09 dB for the HS image I, 0.45 dB to 0.98 dB for the HS image II, 0.22 dB to 0.57 dB for the HS image III, and -0.25 dB to 0.89 dB for the HS image IV. The variation of the PSNR between proposed 3D-STBCTC and 3D-SPIHT is 0.69 dB to 1.12 dB for the HS image I, 0.46 dB to 1.34 dB for the HS image II, 0.55 dB to 0.88 dB for the HS image III, and -0.13 dB to 1.04 dB for the HS image IV. In the same way, the variation of the PSNR between proposed 3D-STBCTC and 3D-WBTC is 0.71 dB to 1.14 dB for the HS image I, 0.48 dB to 1 dB for the HS image II, 0.24 dB to 0.67 dB for the HS image III, and -0.23 dB to 0.9 dB for the HS image IV. The variation of the PSNR between proposed 3D-STBCTC and 3D-LMBTC is 0.83 dB to 1.74 dB for the HS image I, 0.56 dB to 1.34 dB for the HS image II, 0.16 dB to 1.22 dB for the HS image III, and -0.13 dB to 1.17 dB for the HS image IV. The variation between the PNSR of the proposed 3D-STBCTC and 3D-LCBTC is 0.63 dB to 1.08 dB for the HS image I, 0.49 dB to 1.22 dB for the HS image II, -0.03 dB to 0.73 dB for the HS image III, and -0.41 dB to 0.8 dB. The proposed 3D-STBCTC has a high PSNR due to the high number of significant coefficients compared to other compression algorithms for that bit rate. It has been known that a significant coefficient has more weight than a refinement coefficient.

The SSIM is a metric for measuring picture quality that determines how similar two HS images (original and reconstructed) are to one [74, 75]. It has been clear that the proposed compression algorithm has a slight gain (1% to 3%) over other compression algorithms. It is calculated as in Equation 4.

$$\mathrm{SSIM}\ (\mathrm{A},\mathrm{B}) = \frac{(2\mu_A\mu_B +\ \mathrm{C}_1)(2\sigma_{AB} +\ \mathrm{C}_2)}{(\mu_A^2 + \mu_B^2 +\ \mathrm{C}_1\ )(\sigma_A^2 + \sigma_B^2 +\ \mathrm{C}_2)} \qquad (4)$$

The mean average of the original and reconstructed HS image is defined. μ_A & μ_B While the variance of the HS images is defined as σ_A^2 & σ_B^2. σ_{AB} It is defined as the covariance between the HS images [48, 76]. The original HS image and the reconstructed HS image are compared using the Feature-Similarity index metric, which maps the features and measures the degree to which they are similar [38, 77]. The comparative analysis of the FSIM metric has been compared with different compression algorithms, as covered in Table 5. The Bjontegaard metric calculation, also referred to as BD-PSNR, is computed for each of the four HS images that are being evaluated and covered in Table 6. It has been clear from Table 6 that the proposed compression algorithm outperforms the other compression algorithms for all four HS images, as it has been clear with Table 6 (higher numeric value of the PSNR for many bit rates).

5.2. Coding Memory

It has been known that 3D-SPECK, 3D-SPIHT, 3D-WBTC, and 3D-STBCTC use linked lists to track the significance of the coefficients or partitioned sets. 3D-STBCTC uses three different lists for the same. While the rest of the compression algorithms are listless, they do not use any linked lists. These listless compression algorithms use state tables or markers for tracking the significance of the coefficients or partitioned sets. The coding memory required by the listless compression algorithm is fixed for all bit rates and only depends on the size of the HS image. Thus, at the very low bit rates, the demand for coding memory is very high. It is obvious that the memory requirement of 3D-SPECK and 3D-SPIHT is significantly higher in the early passes than it is in the later passes. This is due to the fact that at the lowest threshold level of later passes, more sets will become relevant, and set partitioning will result in more entries in linked lists. From Table 7, it is clear that the proposed compression algorithm 3D-STBCTC has almost the same coding memory requirement as 3D-SPECK, 3D-SPIHT, and 3D-WBTC but has a lower coding memory demand than the listless compression algorithm 3D-NLS at almost every bit rate.

5.3. Coding Complexity

Coding complexity of any compression algorithm is measured as the time required by the compression algorithm for the encoding and decoding process [68, 78]. It has been known that the decoding process requires less execution time than the encoding process [79]. This is because the encoding process requires one more step for the identification of the set (size) and testing of the test for every bit plane. From Table 8, it is clear that listless compression algorithms have a very minimum coding time requirement, while the list-based compression algorithms have a high coding time requirement [55, 80]. This is due to the multiple read/write/memory access operations, which require a lot of time. The Encoding Time (ET) and Decoding Time (DT) are measured in seconds in Table 7, while Total Time (TT) is the sum of time required by the algorithm to complete the encoding and decoding process for the given bit rate [38, 81-86].

The visual representation of the two HS images (Yellowstone Scene 0 and Washington DC MALL) for four different frequency frames (before compression and after compression process) has been shown in Figures 2 and 3.

6. Conclusions and Future Work

In the present manuscript, it is proposed that a compression algorithm be used for leveraging inter- and intra-subband correlations in transformed HS images. The compression algorithm makes use of intra-subband correlations in the form of zeroblock cubes and inter-subband correlations in the form of block cube trees to achieve optimal compression. According to the findings, the suggested coding method consistently performs better than both the tree-based 3D-SPIHT algorithm and the block cube-based 3D-SPECK algorithm for all of the most common test images. When bit rates are lowered, performance improvements become more pronounced. Also, due to the fact that it is a block cube-based encoder, in comparison to 3D-SPIHT, it needs a significantly smaller amount of memory space to store the lists. In addition, the speed of the proposed encoder is significantly higher than that of the 3D-SPIHT encoder. Further, implementation of the sherelet and curvelet transform with the listless compression algorithm decreases the coding complexity, and it also decreases coding memory.

Acknowledgements

The manuscript bears communication number IU/R&D/2025-MCN0003909 from Integral University, Lucknow, India.

References

[1] Rajat Kumar Arya, Subhojit Paul, and Rajeev Srivastava, “An Efficient Hyperspectral Image Classification Method using Retentive Network,” *Advances in Space Research*, vol. 75, no. 2, pp. 1701-1718, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Pallavi Ranjan et al., “Revolutionizing Hyperspectral Image Classification for Limited Labeled Data: Unifying Autoencoder-Enhanced GANs with Convolutional Neural Networks and Zero-Shot Learning,” *Earth Science Informatics*, vol. 18, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Daniel Cozzolino, “Conventional Near-Infrared Spectroscopy and Hyperspectral Imaging: Similarities, Differences, Advantages, and Limitations,” *Molecules*, vol. 30, no. 12, pp. 1-13, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Isao Noda, “Two-Dimensional Correlation Spectroscopy (2D-COS) Analysis of Evolving Hyperspectral Images,” *Applied Spectroscopy*, vol. 79, no. 1, pp. 77-86, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Matúš Pivovar et al., “Spectral Signatures Discrimination of Norway Spruce Trees Under Experimentally Induced Drought and Acute Thermal Stress using Hyperspectral Imaging,” *Forest Ecology and Management*, vol. 586, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

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

[7] Harsha Chandra, and Rama Rao Nidamanuri, “Object-based Spectral Library for Knowledge-Transfer-Based Crop Detection in Drone-Based Hyperspectral Imagery,” *Precision Agriculture*, vol. 26, pp. 1-22, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Claudia Sciuto et al., “What Lies Beyond Sight? Applications of Ultraportable Hyperspectral Imaging (VIS-NIR) for Archaeological Fieldwork,” *Journal of Field Archaeology*, vol. 47, no. 8, pp. 522-535, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Nishir Mehta et al., “Dark-field Hyperspectral Imaging for Label Free Detection of Nano-Bio-Materials,” *Wiley Nanomedicine and Nanobiotechnology*, vol. 13, no. 1, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Chaitanya B. Pande, and Kanak N. Moharir, *Application of Hyperspectral Remote Sensing Role in Precision Farming and Sustainable Agriculture Under Climate Change: A Review*, Climate Change Impacts on Natural Resources, Ecosystems and Agricultural Systems, Springer, Cham, pp. 503-520, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Kartheek Garapati, Sri Satya Maram, and V.M. Manikandan, *Hyperspectral Imaging in Forensic Science: An Overview, Application Areas and Recent Advancements*, Computational Intelligence Based Hyperspectral Image Analysis and Applications, Springer, Cham, pp. 145-177, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] M. Rast et al., “Copernicus Hyperspectral Imaging Mission for the Environment (Chime),” *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, Brussels, Belgium, pp. 108-111, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Aswathi Soni et al., “Hyperspectral Imaging and Machine Learning in Food Microbiology: Developments and Challenges in Detection of Bacterial, Fungal, and Viral Contaminants,” *Comprehensive Reviews in Food Science and Food Safety*, vo. 21, no. 4, pp. 3717-3745, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Roberto De La Ros et al., “Automated Multi-Scale and Multivariate Geological Logging from Drill-Core Hyperspectral Data,” *Remote Sensing*, vol. 14, no. 11, pp. 1-23, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Jonghee Yoon, “Hyperspectral Imaging for Clinical Applications,” *BioChip Journal*, vol. 16, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Canh Nguyen et al., “Early Detection of Plant Viral Disease Using Hyperspectral Imaging and Deep Learning,” *Sensors*, vol. 21, no. 3, pp. 1-23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Václav Křivánek, Jean Motsch, and Yves Bergeon, “Hyperspectral Data Acquisition for Military Camouflage in Vegetation – Preliminary Results,” *2023 Communication and Information Technologies (KIT)*, Vysoke Tatry, Slovakia, pp. 1-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Claudio Sandoval-Muñoz et al., “Enhanced Elemental and Mineralogical Imaging of Cu-mineralized Rocks by Coupling μ-LIBS and HIS,” *Journal of Analytical Atomic Spectrometry*, vol. 37, no. 10, pp. 1981-1993, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] J.L. Garrett et al., “Hyperspectral Image Processing Pipelines on Multiple Platforms for Coordinated Oceanographic Observation,” *2021 11th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Amsterdam, Netherlands, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Lukasz Pieszczek, and Michal Daszykowski, “Integrating Hyperspectrograms with Class Modeling Techniques for the Construction of an Effective Expert System: Quality Control of Pharmaceutical Tablets based on Near-Infrared Hyperspectral Imaging,” *Journal of Pharmaceutical and Biomedical Analysis*, vol. 256, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Tie Wang et al., “Evaluation of Scale Effects on UAV-Based Hyperspectral Imaging for Remote Sensing of Vegetation,” *Remote Sensing*, vol. 17, no. 6, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Wenhui Hou et al., “A Cross-Scene Few-Shot Learning Based on Intra–Inter Domain Contrastive Alignment for Hyperspectral Image Change Detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1-14, 2025. [CrossRef] [Google Scholar] [Publisher Link]

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

[24] 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]

[25] Yuanyang Bu et al., “Transductive Gradient Injection for Improved Hyperspectral Image Denoising,” *Engineering Applications of Artificial Intelligence*, vol. 143, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[26] Dioline Sara et al., “Hyperspectral and Multispectral Image Fusion Techniques for High Resolution Applications: A Review,” *Earth Science Informatics*, vol. 14, pp. 1685-1705, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Pallavi Ranjan, and Ashish Girdhar, “Deep Siamese Network with Handcrafted Feature Extraction for Hyperspectral Image Classification,” *Multimedia Tools and Applications*, vol. 83, pp. 2501-2526, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ganji Tejasree, and Loganathan Agilandeeswari, “An Extensive Review of Hyperspectral Image Classification and Prediction: Techniques and Challenges,” *Multimedia Tools and Applications*, vol. 83, pp. 80941-81038, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] Tian-Heng Zhang et al., “Full-Mode-Augmentation Tensor-Train Rank Minimization for Hyperspectral Image Inpainting,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[30] 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]

[31] 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]

[32] 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]

[33] 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]

[34] Shumin Liu et al., “A Comprehensive Review on Hyperspectral Image Lossless Compression Algorithms,” *Remote Sesning*, vol. 17, no. 24, pp. 1-48, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[35] Anuja Bhargava et al., “Hyperspectral Imaging and its Applications: A Review,” *Heliyon*, vol. 10, no. 12, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[36] 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]

[37] 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]

[38] 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]

[39] S. Sanjith, and R. Ganesanm, “A Review on Hyperspectral Image Compression,” *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari, India, pp. 1159-1163, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[40] 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]

[41] 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]

[42] 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]

[43] Sheng Cang, and Achuan Wang, "Research on Hyperspectral Image Reconstruction Based on GISMT Compressed Sensing and Interspectral Prediction," *International Journal of Optics*, vol. 2020, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[44] 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]

[45] İrem Ülkü, and Behçet Uğur Töreyin, "Sparse Representations for Online-Learning-based Hyperspectral Image Compression," *Applied Optics*, vol. 54, no. 29, pp. 8625-8631, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[46] 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]

[47] Zainab N. Abdulhameed Al-Rawi, Haraa R. Hatem, and Israa H. Ali, "Image Compression Using Contourlet Transform," *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, Fallujah, Iraq, pp. 254-258, 2018. [CrossRef] [Google Scholar] [Publisher Link]

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

[49] 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]

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

[51] 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]

[52] 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]

[53] V.K. Sudha, and R. Sudhakar, "3D Listless Embedded Block Coding Algorithm for Compression of Volumetric Medical Images," *Journal of Scientific and Industrial Research*, vol. 72, no. 12, pp. 735-748, 2013. [Google Scholar] [Publisher Link]

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

[55] 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]

[56] Ming-Fang Cheng et al., "Modern Trends and Recent Applications of Hyperspectral Imaging: A Review," *Technologies*, vol. 13, no. 5, pp. 1-48, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[57] Amal Altamimi, and Belgacem Ben Youssef, "Lossless and Near-Lossless Compression Algorithms for Remotely Sensed Hyperspectral Images," *Entropy*, vol. 26, no. 4, pp. 1-35, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[58] 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]

[59] 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]

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

[61] Bing Lu et al., "Recent Advances of Hyperspectral Imaging Technology and Applications in Agriculture," *Remote Sensing*, vol. 12, no. 16, pp. 1-44, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[62] P. Anjaneya, and G.K Rajini, "A Comprehensive Framework of Combined Lossless Image Compression Algorithm for Light-Field Hyperspectral Images," *2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, Hamburg, Germany, pp. 322-327, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[63] Fahad Saeed et al., "Hyperspectral Image Compression with Deep Learning: A Review," *Signal Processing*, vol. 239, 2026. [CrossRef] [Google Scholar] [Publisher Link]

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

[65] Muhammad Azhar Iqbal, Muhammad Younus Javed, and Usman Qayyum, “Curvelet-Based Image Compression with SPIHT,” *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, Gwangju, Korea (South), pp. 961-965. [CrossRef] [Google Scholar] [Publisher Link]

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

[67] 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]

[68] 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]

[69] Jing Zhang et al., “Improvements to 3D-Tarp Coding for the Compression of Hyperspectral Imagery,” *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, Boston, MA, USA, pp. II-982-II-985, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[70] Ying Hou, and Guizhong Liu, “3D Set Partitioned Embedded Zero Block Coding Algorithm for Hyperspectral Image Compression,” *Proceedings MIPPR 2007: Remote Sensing and GIS Data Processing and Applications; and Innovative Multispectral Technology and Applications*, Wuhan, China, vol. 679056, 2007. [CrossRef] [Google Scholar] [Publisher Link]

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

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

[73] 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]

[74] 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]

[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] Md. Toukir Ahmed et al., “A Comprehensive Review of Deep Learning-based Hyperspectral Image Reconstruction for Agri-food Quality Appraisal,” *Artificial Intelligence Review*, vol. 58, pp. 1-28, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[77] 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]

[78] 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]

[79] R. David Dunphy et al., “Hyperspectral Imaging Suggests Potential for Rapid Quantification of Fission Products in Spent Nuclear Fuel,” *Scientific Reports*, vol. 15, pp. 1-14, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[80] Chuan Fu, Bo Du, and Xinjian Huang, “Hyperspectral Image Compression based on Multiple Priors,” *Journal of the Franklin Institute*, vol. 361, no. 14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[81] Maitreyi Joglekar, and Ashwini M. Deshpande, “A Comprehensive Review of Hyperspectral Image Denoising Techniques in Remote Sensing,” *International Journal of Remote Sensing*, vol. 46, no. 16, pp. 5961-5995, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[82] 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]

[83] 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]

[84] Hridoy Biswas et al., “Wavelet-based Compression Method for Scale-Preserving in VNIR and SWIR Hyperspectral Data,” *Journal of Medical Imaging*, vol. 12, no. 4, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[85] 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]

[86] Vijay Joshi, and J. Sheeba Rani, "A Band Interleaved by Pixel (BIP) Architecture of a Simple Lossless Algorithm (SLA) for On-Board Satellite Hyperspectral Data Compression," *2025 23rd IEEE Interregional NEWCAS Conference (NEWCAS)*, Paris, France, pp. 490-494, 2025. [CrossRef] [Google Scholar] [Publisher Link]

Appendix

Table 5. Coding efficiency of the different HSICAs for seven different bit rates for four HS images

Bit Rate	Performance Metrics	Compression Algorithm I [49]	Compression Algorithm II [50]	Compression Algorithm III [51]	Compression Algorithm IV [52]	Compression Algorithm V [53]	Compression Algorithm VI [54]	Compression Algorithm VII [48]	Compression Algorithm VIII [55]	Compression Algorithm IX [59]	3D-STBCTC
	HS Image I										
0.001	IQM I	26.28	26.28	26.25	26.14	25.90	26.26	26.41	26.32	26.41	27.09
	IQM II	0.08	0.08	0.08	0.07	0.06	0.08	0.07	0.08	0.08	0.09
	IQM III	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.38	0.4
0.005	IQM I	28.95	28.95	28.93	28.71	28.71	28.70	28.66	28.73	28.97	29.74
	IQM II	0.20	0.20	0.19	0.19	0.18	0.19	0.19	0.19	0.19	0.21
	IQM III	0.43	0.43	0.43	0.43	0.44	0.44	0.44	0.44	0.44	0.44
0.01	IQM I	30.08	30.08	30.04	29.99	29.83	29.98	30.01	29.99	30.21	30.81
	IQM II	0.23	0.23	0.23	0.24	0.22	0.23	0.24	0.24	0.23	0.26
	IQM III	0.55	0.55	0.55	0.56	0.55	0.55	0.55	0.55	0.56	0.58
0.05	IQM I	34.23	34.23	34.21	34.04	33.81	33.99	34.29	34.06	34.42	34.92
	IQM II	0.37	0.37	0.37	0.38	0.37	0.38	0.37	0.38	0.38	0.42
	IQM III	0.69	0.68	0.69	0.69	0.68	0.68	0.68	0.68	0.69	0.7
0.1	IQM I	37.22	37.22	37.20	36.96	37	36.83	37.34	36.87	37.34	38.02
	IQM II	0.42	0.42	0.42	0.44	0.42	0.43	0.43	0.44	0.43	0.44
	IQM III	0.74	0.74	0.74	0.75	0.75	0.75	0.75	0.75	0.75	0.76
0.25	IQM I	42.17	42.17	42.16	41.62	41.69	41.34	42.28	41.37	42.38	43.08
	IQM II	0.5	0.5	0.5	0.52	0.52	0.52	0.52	0.52	0.52	0.53
	IQM III	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.81	0.83
0.5	IQM I	48.02	47.99	47.97	47.01	47.79	47.51	48.11	47.55	48.21	49.11
	IQM II	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.6
	IQM III	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.91	0.92
	HS Image II										
0.001	IQM I	27.11	26.75	27.09	26.83	26.61	26.75	26.87	26.82	26.87	28.09
	IQM II	0.27	0.27	0.27	0.27	0.26	0.27	0.27	0.26	0.27	0.28
	IQM III	0.59	0.58	0.59	0.59	0.58	0.58	0.59	0.58	0.59	0.6
0.005	IQM I	29.45	29.31	29.43	29.27	29.25	29.24	29.41	29.25	29.54	29.97
	IQM II	0.37	0.37	0.37	0.37	0.36	0.37	0.37	0.36	0.37	0.39
	IQM III	0.69	0.68	0.69	0.69	0.68	0.69	0.69	0.69	0.7	0.71
0.01	IQM I	30.28	30.19	30.27	30.27	30.15	30.31	30.53	30.33	30.48	31.08
	IQM II	0.42	0.42	0.42	0.42	0.41	0.42	0.42	0.42	0.42	0.43
	IQM III	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.71	0.73
0.05	IQM I	33.76	33.61	33.73	33.56	33.59	33.51	33.69	33.54	33.93	34.21
	IQM II	0.61	0.60	0.61	0.61	0.60	0.61	0.61	0.61	0.61	0.63
	IQM III	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.72	0.73
0.1	IQM I	35.57	35.44	35.56	35.49	35.41	33.45	35.55	35.46	35.69	36.32
	IQM II	0.67	0.66	0.67	0.67	0.66	0.67	0.67	0.67	0.67	0.69
	IQM III	0.73	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74	0.74
0.25	IQM I	39.30	39.19	39.29	39.26	39.17	39.22	39.37	39.23	39.41	39.94

	IQM II	0.76	0.76	0.76	0.77	0.76	0.76	0.76	0.77	0.76	0.77
	IQM III	0.77	0.77	0.77	0.77	0.77	0.78	0.77	0.77	0.78	0.78
0.5	IQM I	43.62	43.65	43.51	43.57	43.26	43.55	43.62	43.58	43.74	44.11
	IQM II	0.82	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.84
	IQM III	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.82
		HS Image III									
0.001	IQM I	27.82	27.49	27.8	27.78	27.28	27.88	28.07	27.92	27.77	28.04
	IQM II	0.16	0.17	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.18
	IQM III	0.54	0.54	0.54	0.54	0.53	0.53	0.53	0.53	0.54	0.55
0.005	IQM I	30.24	30.09	30.22	30.03	30.03	30.01	30.44	30.02	30.28	30.81
	IQM II	0.29	0.29	0.29	0.29	0.28	0.29	0.28	0.29	0.29	0.31
	IQM III	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.67
0.01	IQM I	31.27	31.14	31.25	31.17	31.1	31.13	31.42	31.14	31.38	31.83
	IQM II	0.34	0.34	0.34	0.33	0.33	0.33	0.33	0.33	0.33	0.35
	IQM III	0.69	0.69	0.70	0.69	0.69	0.69	0.69	0.69	0.69	0.7
0.05	IQM I	34.57	34.39	34.55	34.58	34.27	34.44	34.67	34.51	34.63	35.05
	IQM II	0.48	0.48	0.48	0.49	0.48	0.48	0.48	0.48	0.48	0.5
	IQM III	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.72
0.1	IQM I	36.63	36.49	36.64	36.42	36.49	36.35	36.74	36.37	36.71	37.08
	IQM II	0.54	0.54	0.54	0.55	0.54	0.54	0.53	0.54	0.54	0.56
	IQM III	0.74	0.74	0.74	0.74	0.75	0.74	0.74	0.74	0.75	0.76
0.25	IQM I	40.83	40.63	40.84	40.46	40.59	40.29	40.81	40.31	40.85	41.51
	IQM II	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.67
	IQM III	0.77	0.77	0.78	0.77	0.77	0.77	0.77	0.77	0.77	0.79
0.5	IQM I	45.88	45.66	45.87	45.39	45.57	45.13	45.58	45.15	45.92	46.31
	IQM II	0.72	0.72	0.72	0.73	0.72	0.72	0.73	0.72	0.72	0.73
	IQM III	0.81	0.82	0.81	0.81	0.82	0.80	0.80	0.80	0.82	0.83
		HS Image IV									
0.001	IQM I	28.11	27.94	28.06	28.08	27.88	28.07	28.14	28.16	28.15	28.81
	IQM II	0.21	0.21	0.21	0.21	0.20	0.21	0.21	0.20	0.21	0.23
	IQM III	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.4
0.005	IQM I	30.44	30.32	30.43	30.27	30.03	30.26	30.22	30.28	30.52	31.02
	IQM II	0.32	0.31	0.32	0.32	0.31	0.31	0.32	0.31	0.31	0.33
	IQM III	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.59
0.01	IQM I	31.41	31.29	31.39	31.32	31.1	31.29	31.57	31.43	31.52	31.16
	IQM II	0.37	0.37	0.36	0.36	0.37	0.36	0.36	0.36	0.37	0.39
	IQM III	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.69
0.05	IQM I	34.46	34.3	34.45	34.41	34.27	34.25	34.62	34.28	34.53	35.11
	IQM II	0.53	0.53	0.53	0.54	0.53	0.53	0.53	0.53	0.53	0.55
	IQM III	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.75
0.1	IQM I	36.43	36.29	36.43	36.25	36.49	36.19	36.51	36.2	36.55	37.14
	IQM II	0.60	0.59	0.60	0.60	0.60	0.60	0.60	0.60	0.59	0.61
	IQM III	0.74	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.76
0.25	IQM I	40.08	39.93	40.07	39.92	40.59	39.8	40.19	39.84	40.17	40.97
	IQM II	0.71	0.71	0.71	0.71	0.71	0.71	0.72	0.71	0.71	0.73
	IQM III	0.78	0.78	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.8
0.5	IQM I	44.51	44.47	44.5	44.31	44.46	44.22	44.63	44.22	44.62	45.19
	IQM II	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.81
	IQM III	0.81	0.82	0.81	0.81	0.81	0.81	0.81	0.81	0.82	0.83

Table 6. BD-PSNR gain of 3D-STBCTC with the other HSICAs

HS Image	Compression Algorithm I [49]	Compression Algorithm II [50]	Compression Algorithm III [51]	Compression Algorithm IV [52]	Compression Algorithm V [53]	Compression Algorithm VI [54]	Compression Algorithm VII [48]	Compression Algorithm VIII [55]	Compression Algorithm IX [59]
HS Image I	0.7943	0.7963	0.8205	1.0663	1.1025	1.0812	0.8102	1.0433	0.6659
HS Image II	0.6575	0.7961	0.6805	0.7788	0.8659	1.0937	0.6458	0.7814	0.5560
HS Image III	0.5005	0.6781	0.5126	0.6649	0.7511	0.7352	0.3902	0.7085	0.4477
HS Image IV	0.4850	0.6179	0.4998	0.6050	0.6351	0.6687	0.4346	0.6152	0.3958

Table 7. Analysis compares the memory efficiency of the 3D-STBCTC algorithm with other HSICAs, with a specific focus on coding memory consumption (Measured in KB)

Bit Rate	Compression Algorithm I [49]	Compression Algorithm II [50]	Compression Algorithm III [51]	Compression Algorithm IV [52]	Compression Algorithm V [53]	Compression Algorithm VI [54]	Compression Algorithm VII [48]	Compression Algorithm VIII [55]	Compression Algorithm IX [59]	3D-STBCTC
	HS Image I									
0.001	26.67	37.33	28.08	4096	8192	96	2318	0	2304	30.05
0.005	102.3	99.21	89.33	4096	8192	96	2318	0	2304	86.19
0.01	232.2	222.7	202.4	4096	8192	96	2318	0	2304	198.1
0.05	1084	1041	991.7	4096	8192	96	2318	0	2304	921.5
0.1	1846	1931	1756	4096	8192	96	2318	0	2304	1698
0.25	4571	4463	4289	4096	8192	96	2318	0	2304	4019
0.5	8644	8555	8514	4096	8192	96	2318	0	2304	8411
	HS Image II									
0.001	22.58	21.51	22.69	4096	8192	96	2318	0	2304	19.87
0.005	91.12	98.91	91.29	4096	8192	96	2318	0	2304	90.25
0.01	265.9	267.8	266.4	4096	8192	96	2318	0	2304	264.9
0.05	982.4	1036	985.4	4096	8192	96	2318	0	2304	987.2
0.1	2219	2326	2229	4096	8192	96	2318	0	2304	2177
0.25	5450	5611	5464	4096	8192	96	2318	0	2304	5514
0.5	10005	9981	9832	4096	8192	96	2318	0	2304	9609
	HS Image III									
0.001	25.28	24.94	25.06	4096	8192	96	2318	0	2304	24.71
0.005	101.2	105.8	101.5	4096	8192	96	2318	0	2304	99.25
0.01	205.1	218.9	208.6	4096	8192	96	2318	0	2304	212.1
0.05	1108	1149	1136	4096	8192	96	2318	0	2304	1109
0.1	1855	1808	1854	4096	8192	96	2318	0	2304	1819
0.25	4401	4449	4412	4096	8192	96	2318	0	2304	4408
0.5	7918	7805	7935	4096	8192	96	2318	0	2304	8002
	HS Image IV									
0.001	24.67	22.41	24.55	4096	8192	96	2318	0	2304	23.28
0.005	100.8	105.5	101.1	4096	8192	96	2318	0	2304	102.2
0.01	210.9	229.9	214.4	4096	8192	96	2318	0	2304	211.8
0.05	1088	1212	1106	4096	8192	96	2318	0	2304	1084
0.1	1970	2083	1980	4096	8192	96	2318	0	2304	1971
0.25	4867	5047	4878	4096	8192	96	2318	0	2304	4759
0.5	9078	8488	9093	4096	8192	96	2318	0	2304	8948

Table 8. A comparative analysis of the coding complexity for various set-partitioning algorithms used in hyperspectral image compression, evaluated at different bit rates

Bit Rate	Computation Time	Compression Algorithm I [49]	Compression Algorithm II [50]	Compression Algorithm III [51]	Compression Algorithm IV [52]	Compression Algorithm V [53]	Compression Algorithm VI [54]	Compression Algorithm VII [48]	Compression Algorithm VIII [55]	Compression Algorithm IX [59]	3D-STBCTC
	HS Image I										
0.001	ET	3.99	4.06	5.94	2.67	14.18	5.91	3.17	3.24	20.09	5.67
	DT	1.78	2.92	1.59	2.08	12.79	2.48	2.21	3.02	3.57	1.51
	TT	5.77	6.98	7.53	4.75	26.97	8.39	5.38	6.26	23.66	7.18
0.005	ET	9.85	9.73	8.2	2.78	61.33	8.35	3.35	4.83	82.29	7.91
	DT	5.18	5.25	2.41	2.43	48.29	3.86	2.68	4.65	14.87	2.32
	TT	15.13	14.98	10.61	5.21	109.62	12.21	6.03	9.48	97.16	10.23
0.01	ET	20.45	29.93	10.99	3.25	73.64	9.26	4.41	5.97	93.55	10.28
	DT	10.78	14.31	4.51	2.68	57.16	4.04	3.08	5.61	21.59	3.94
	TT	31.23	44.24	15.5	5.93	130.8	13.3	7.49	11.58	115.14	14.22
0.05	ET	222.2	303.4	94.36	5	90.57	19.45	5.49	12.18	102.89	90.25
	DT	172.7	236.2	84.75	4.02	69.23	12.01	4.34	11.79	51.47	67.22
	TT	394.9	539.6	179.11	9.02	159.8	31.46	9.83	23.97	154.38	157.47
0.1	ET	1163	1297	762.6	7.31	102.5	34.74	7.94	19.55	117.8	711.2
	DT	1081	1078	762.11	6.24	77.57	21.79	6.71	18.36	59.6	657.7
	TT	2244	2375	1524.71	13.55	180.07	56.53	14.65	37.91	177.4	1368.9
0.25	ET	6234	6871	4703	13.35	120.8	68.15	14.02	40.25	131.8	4641
	DT	6012	6305	4358	11.68	90.45	50.91	12.02	37.86	67.2	4129
	TT	12246	13176	9061	25.03	211.25	119.06	26.04	78.11	199	8770
0.5	ET	17995	18742	19551	24.12	151.3	122.5	26.03	74.87	160.8	18925
	DT	17597	18534	15400	22.65	100.5	96.84	25.07	69.02	89.7	14950
	TT	35592	37276	34951	46.77	251.8	219.34	51.10	143.89	250.5	33875
	HS Image II										
0.001	ET	3.42	4.33	5.94	2.35	15.97	5.73	2.47	2.94	17.89	4.57
	DT	1.87	1.52	1.46	1.4	12.18	2.18	1.61	2.79	9.78	1.36
	TT	5.29	5.85	7.4	3.75	28.15	7.91	4.08	5.73	27.67	5.93
0.005	ET	9.84	5.85	8.5	2.71	75.93	7.36	3.87	6.44	84.67	8.11
	DT	5.4	2.45	2.77	2.49	66.24	3.21	3.01	6.05	54.94	2.54
	TT	15.24	8.3	11.27	5.2	142.17	10.57	6.88	12.49	139.61	10.65
0.01	ET	22.53	9.41	10.83	2.88	90.43	16.99	4.29	10.28	99.47	10.21
	DT	10.01	4.92	3.86	2.71	81.48	6.23	3.27	10.04	66.38	3.38
	TT	32.54	14.33	14.69	5.59	171.9	23.22	7.56	20.32	165.85	13.59
0.05	ET	250.3	134.4	131.5	4.14	106.55	27.4	5.02	16.02	121.8	128.2
	DT	207.2	127.8	130.1	3.38	94.49	14.94	3.94	11.35	79.7	120.1
	TT	457.5	262.2	261.6	7.52	201.04	42.34	8.96	27.37	201.5	248.3
0.1	ET	966.7	570.8	632.6	6.04	125.87	36.27	7.21	18.42	143.8	600.2
	DT	887.6	717.5	614.3	5.98	106.8	23.01	6.64	17.81	80.3	581.7
	TT	1854.3	1288.3	1246.9	12.02	232.67	59.28	13.85	36.23	224.1	1181.9
0.25	ET	4973	3032	4100	10.24	134.4	96.34	12.21	56.67	155.9	4084
	DT	4796	3029	4040	6.74	113.86	58.62	7.18	47.06	87.5	4011
	TT	9769	6161	8240	16.98	248.26	154.96	19.39	103.73	243.4	9095
0.	ET	12007	10112	12975	17.25	154.41	177.73	18.95	67.74	180.9	12874

5	DT	11898	9954	12299	14.7	125.56	120.33	15.34	60.13	99.5	12511
	TT	23905	20066	25274	31.95	279.97	298.06	34.29	127.87	280.4	25386
	HS Image III										
0.001	ET	4.08	4.03	5.85	2.07	15.97	5.68	2.76	3.19	17.79	5.77
	DT	1.74	1.39	1.32	1.89	8.43	4.1	2.11	3.02	6.54	1.29
	TT	5.82	5.42	7.17	3.96	24.4	9.78	4.87	6.21	24.33	7.06
0.005	ET	9.12	5.96	7.87	2.89	75.93	7.78	3.28	4.74	88.54	7.57
	DT	5.13	2.24	2.44	2.47	66.02	6.02	2.74	3.99	49.87	2.31
	TT	14.25	8.2	10.31	5.36	141.95	13.8	6.02	8.73	138.41	9.88
0.01	ET	20.18	9.7	11.64	3.34	90.43	8.55	4.01	7.52	101.5	11.37
	DT	12.51	5.18	5.14	2.69	84.96	7.06	3.02	6.33	79.5	4.74
	TT	32.69	14.88	16.78	6.03	175.39	15.61	7.03	13.85	181	16.08
0.05	ET	204.3	125.2	89.77	4.57	106.55	19.48	5.31	22.88	119.5	81.29
	DT	160.3	114.7	80.01	4.46	92.68	14.84	5.19	18.56	87.4	74.35
	TT	364.6	239.9	169.78	9.03	199.23	34.32	10.5	41.44	206.9	155.64
0.1	ET	1183	775.8	835.9	5.91	125.87	32.46	6.47	30.14	138.8	798.3
	DT	1074	760.5	827.8	5.59	104.98	21.49	6.37	27.82	100.5	766.1
	TT	2257	1536.3	1663.7	11.5	230.85	53.95	12.84	57.96	239.3	1564.4
0.25	ET	8499	5151	6309	10.41	134.14	70.4	11.91	43.49	149.5	6281
	DT	8387	5832	6233	9.27	115.94	48.95	10.34	39.95	108.2	6008
	TT	16886	10983	12858	19.68	250.08	119.35	22.25	83.44	257.7	12289
0.5	ET	29849	18383	23861	16.19	154.41	125.42	17.09	72.62	165.8	22589
	DT	26948	15672	23161	14.97	141.97	114.52	16.68	67.23	130.1	21598
	TT	56797	34055	47022	31.16	296.38	239.94	33.77	139.85	295.9	44187
	HS Image IV										
0.001	ET	4.56	5.6	7.23	2.39	6.03	5.74	2.89	2.82	7.85	6.94
	DT	2.41	1.64	1.73	2.02	5.27	2.1	2.24	2.74	4.34	1.59
	TT	6.97	7.24	8.96	4.41	11.3	7.84	5.13	5.56	12.19	8.53
0.005	ET	15.24	6.23	8.15	2.81	11.53	7.53	3.34	4.44	12.94	7.91
	DT	9.57	2.33	2.55	2.34	8.26	2.88	2.47	4.28	7.19	2.41
	TT	24.81	8.56	10.7	5.15	19.79	10.41	5.81	8.72	20.13	10.32
0.01	ET	21.67	10.2	12.64	3.18	18.44	8.93	3.98	5.64	21.18	11.59
	DT	12.68	5.23	6.11	2.89	14.44	3.91	3.23	5.41	12.78	5.87
	TT	34.35	15.43	18.75	6.07	32.88	12.84	7.21	11.05	33.96	17.46
0.05	ET	269.6	130.4	98.12	4.3	22.64	18.61	4.88	13.02	24.91	94.23
	DT	226.5	120.5	89.08	3.74	19.5	11.48	4.29	11.36	17.47	84.51
	TT	496.1	250.9	187.2	8.04	42.14	30.09	9.17	24.38	42.38	178.74
0.1	ET	1336	893.4	882.3	6.11	25.53	32.45	6.41	18.18	30.58	841.2
	DT	1241	829.1	866.3	5.96	21.07	21.02	6.57	17.22	20.17	801.1
	TT	2577	1722.5	1748.6	12.07	46.6	53.47	12.98	35.4	50.75	1642.3
0.25	ET	8435	5133	5501	10.35	34.5	69.66	11.38	36.3	38.54	5204
	DT	9067	4536	5494	6.62	29.65	48.91	7.08	33.79	24.25	5129
	TT	17502	9669	10995	16.97	64.15	118.57	18.46	70.09	62.79	10333
0.5	ET	27917	17945	18818	17.43	65.13	125.19	19.01	66.91	70.04	18429
	DT	25042	17677	18136	12.03	55.03	92.97	12.87	62.31	48.95	17922
	TT	52959	35622	36954	29.46	120.16	218.16	31.88	129.22	118.99	36351

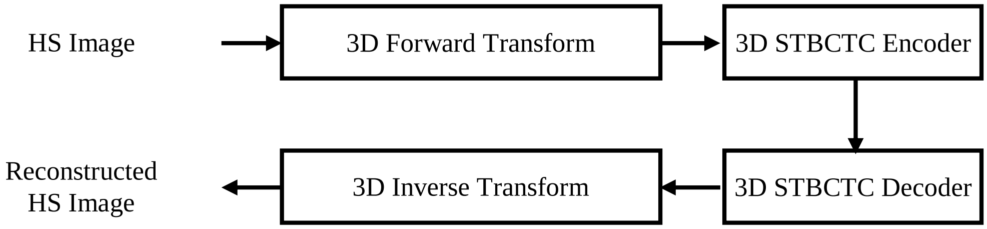

Fig. 1 A block diagram illustrating the operational flow of the proposed HSICA 3D-STBCTC

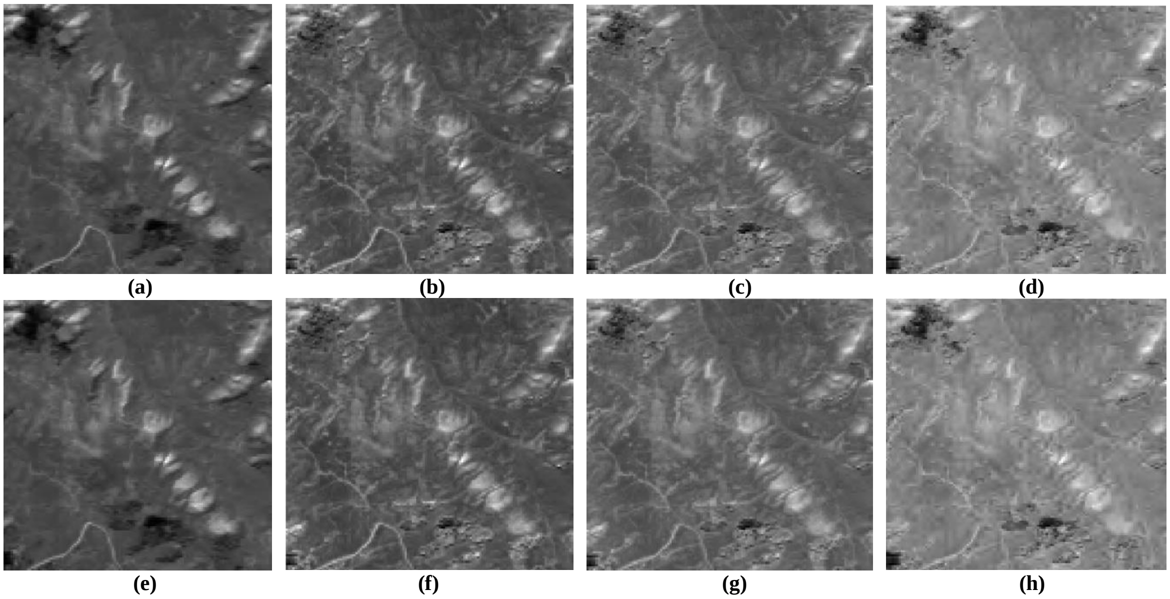

Fig. 2 Original HS image II of different frames (with frame number) (a) 25, (b) 50, (c) 75, (d) 100, reconstructed HS image II with CR 16 of different frames (with frame number), (e) 25, (f) 50, (g) 75, and (h) 100.

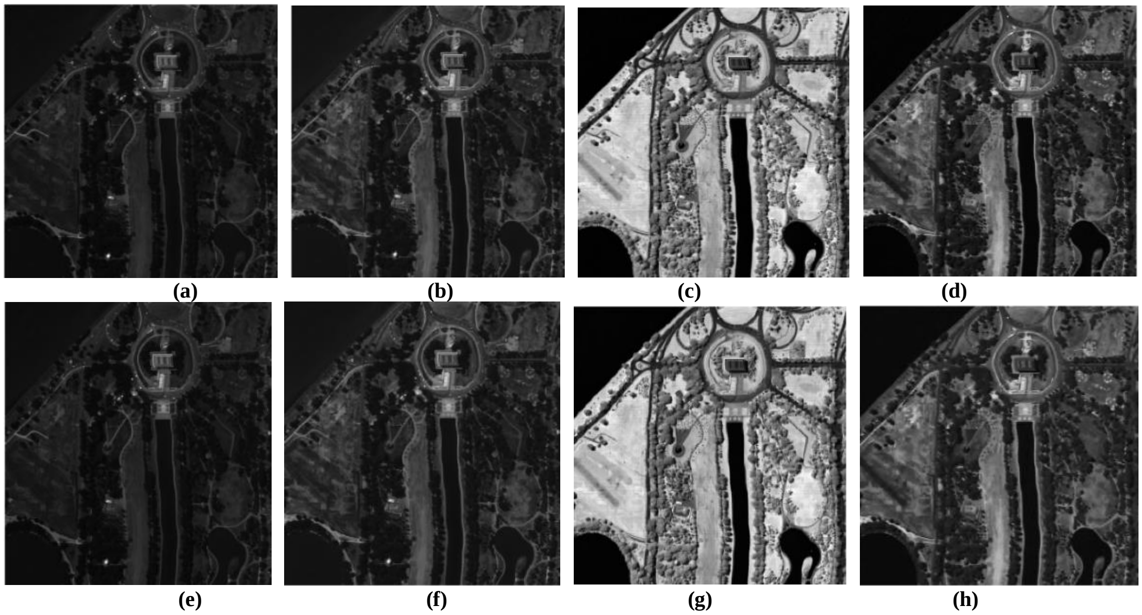

Fig. 3 Original HS image I of different frames (with frame number), (a) 30, (b) 60, (c) 90, (d) 150, reconstructed HS image I with CR 14 of different frames (with frame number), (e) 30, (f) 60, (g) 90, and (h) 150.