SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I3P102

Volume 13 Issue 3, 13-27, March 2026
© 2026 Seventh Sense Research Group®

Original Article

3D Modified No List Set Partitioning in Hierarchical Trees Coding for 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: 05 December 2025 Revised: 04 January 2026 Accepted: 03 February 2026 Published: 23 March 2026

Abstract - *Hyperspectral images have rich spatial-spectral information, which is used in various applications. But the large size of hyperspectral images poses a significant challenge to the image sensor performance. To properly handle the HS image data, an efficient compression algorithm is required to compress the HS image data. There are many types of compression algorithms that have been proposed in the past, but wavelet transform-based set-partitioned hyperspectral compression algorithms have superior coding performance. But, these compression algorithms do not provide the desired features of progressive transmission and spatial scalability at very low bit rate coding. Also, these compression algorithms use a linked list for tracking of the coefficients during the coding process, which becomes a bottleneck for fast implementation. The proposed compression algorithm uses markers instead of lists, which reduces the complexity significantly by ~10% to ~20% with reference to 3D-NLS. The significance reordering of the transform coefficients in the proposed algorithm ensures that highly significant coefficients are encoded first to exploit the energy compaction of the transform and increase the coding efficiency. Thus, the proposed compression algorithm is a choice for the low-resource HS image sensors.*

Keywords - *Hyperspectral Image, Compression, Zerotree, Set Partition Hyperspectral Image Compression Algorithm, Coding, Wireless Sensor Network, Energy Efficiency.*

1. Introduction

Hyperspectral (HS) images gather rich spatial-spectral information of the single scene in hundreds of narrow and continuous (from the visible spectrum to near-infrared spectrum) spectral bands (frequency frames) captured by onboard hyperspectral remote sensors and are more effective in distinguishing different land cover types than other multispectral images or RGB images [1, 2]. Therefore, HS image is used in several applications ranging from agriculture [3], biomedical engineering [4], environment (weather, droughts, sandstorms, wildfires etc) [5], geological exploration [6], meat quality analysis [7], mineral exploring [8], remote sensing [9], urban change detection (urban planning, land classification) [10] etc. Although HS image technology has several benefits in different applications, as mentioned above, researchers face numerous challenges. These challenges are addressed with the development of the computer based algorithms on change detection [11], classification [12, 13], compression [14], dimension reduction [15], feature extraction and selection [16], frequency band selection [17], denoising [18], object identification [19], target detection [20], unmixing [21] etc. The size of a single HS image is around 150 MB. To transmit this HS image data, a lot of transmission data bandwidth, a huge onboard memory requirement, and ample processing power are required for processing the HS image data [22]. Thus, compression of the HS image becomes a necessary preprocessing step before the transmission of the HS image from the transmitter of the onboard HS image sensor to the ground station [23]. Compression Ratio (CR) is a unitless parameter that is defined as a ratio between the number of bits in the original HS image and the number of bits used in the reconstructed HS image after the compression process [24]. Mathematically, it is defined as:

$$CR = \frac{Bits\ use\ in\ the\ original\ HS\ image}{Bits\ use\ to\ represent\ the\ reconstucted\ HS\ image} \tag{1}$$

The HSICAs are split into different groups on the basis of the HS image compression process or HS image data loss. Again, it is divided into three subgroups named lossless, near lossless, and lossy HS image compression. As the name suggests, in lossless compression, there is no loss of any HS image data during the compression process, and the associated CR to the compression algorithm is also very low (near two to four), while in near lossless compression, there is a slight loss in the HS image data, but it has slightly higher CR. The lossy compression, as the name suggests, has a loss of the HS image data, but it has high CR. In lossy compression, the HS image can be compressed according to the CR, while in lossless and near lossless compression, this cannot be performed [25-27].

On the basis of the coding process, the compression algorithms are divided into seven subgroups named as Prediction Coding (PC) based HSICA [28], Vector Quantization (VQ) based HSICA [29], Transform Coding (TC) based HSICA [30], Compression Sensing (CS) based HSICA [31], Tensor Decomposition (TD) HSICA [32], Machine Learning (ML) based HSICA [33], and hybrid compression algorithms [34].

Among the above-mentioned types of HSICA, on the basis of coding efficiency, ML-based HSICA has higher performance, while on the basis of coding efficiency, PC based HSICA is the least complex. TC-based HSICA works with lossy, lossless, and near-lossless compression according to the need [35].

Further, in the TC-based HSICA, a special type of subclass called Transform-based Set Partitioned Hyperspectral Image Compression Algorithms, which uses the set structures to define a large number of insignificant coefficients at the higher bit planes. These compression algorithms exploit the property (energy compaction) of the wavelet transform to achieve the compression of the HS image. These compression algorithms have an embeddedness property in which decoding of the compression algorithm can be performed with a lower bit rate [36-43].

The contributions for this manuscript are summarized as follows:

- The proposed compression algorithm uses a smaller number of markers, which reduces the demand for memory by 25% to the conventional 3D-NLS.
- The proposed compression algorithm has low coding complexity (encoding and decoding time) compared to the conventional 3D-NLS because of the use of a smaller number of markers, which reduces the multiple read/write operations.
- We also provide an empirical evaluation and experimental comparison of representative algorithms on three popular datasets, offering assistance with performance analysis and providing some discussions on future research.

The rest of this manuscript is organized as follows. Section 2 provides the preliminary details of this study, which include a detailed discussion of state-of-the-art HSICA 3D-SPIHT [37] and its listless version 3D-NLS [40]. The proposed compression algorithm is covered in Section 3, including its pseudo-code. In Section 4, experimental (simulation) results on four HS image datasets are provided to illustrate the effectiveness of the proposed compression algorithm and compare its performance with eight state-of-the-art HSICAs. Finally, in Section 5, we conclude this manuscript with a summary.

2. Related Work

The set partition-based HSICA is further split into three categories named as Zero block cube, zero block tree, and zero block cube tree [43]. The categorization is based upon the orientation of the insignificant coefficients, which are represented by the single bit [42]. In the zero block cube, the whole insignificant block cube for the current bit plane is represented by one bit, and the block cube is called the zero block cube [42]. 3D-SPECK [36], 3D-LSK [39] and 3D-ZM-SPECK [42]. In the zerotree, the whole tree is a zerotree if it is insignificant to the current bit plane, and a whole zerotree is represented by the one single digit ‘0’. 3D-SPIHT [37] and 3D-NLS [40] are state-of-the-art HSICA that belong to this family. The last type of compression algorithms uses the best property of the zeroblock cube and zerotree based HSICAs [41]. These HSICAs have high coding efficiency at very low bit rates as they can represent eight times more insignificant coefficients than the zeroblock cube-based HSICA. 3D-WBTC [38] and 3D-LMBTC [41] are the major HSICA that fall under this category.

A short overview of the different transform-based set partitioned HSICAs is covered in Table 1.

3. 3D Modified No List SPIHT (3D-M-NLS)

The coding efficiency is the prime concern for any compression algorithm for the HS images. Many attempts have been made in the last decades to increase the coding efficiency, but the gain in coding efficiency comes at the cost of coding complexity or coding memory. An approach for low-complexity compression requires fewer mathematical and logical calculations, which can save both processing time and the amount of power that the sensor requires. The demand for coding memory can be reduced for high bit rates through the use of the listless approach, but it comes at the cost of coding efficiency.

In the proposed HSICA, rather than searching up the zerotree to fetch predictable insignificance, special state symbols (state table markers) with three bits per coefficient are set in a table of a defined size on particular lower nodes of insignificant zerotrees when the trees are built. This is done in order to avoid the hassle of searching up the zerotree. Whenever partitioning results in the formation of new zerotrees that are of no significance, these markers are updated.

The proposed HSICA uses the same set of structures and partition rule as 3D-SPIHT [37] and 3D-NLS [40]. It uses the six different markers to define the state of the coefficients. The size of the marker is three bits per coefficient. The detailed description of the markers is covered in Table 2.

Both the DES and GRANDES markers have meanings that are comparable to those of the 3D-SPIHT [37] set type A and type B. In 3D-SPIHT, type A and type B are associated with the zerotree root coefficient of a set, which is not actually in the set. On the other hand, these markers are directly related to a coefficient that is contained within the set.

The computation of the two maximum descendant magnitude matrices is carried out. The phrase ‘dmax (i, j, k)’ refers to the maximum magnitude of the descendant set that is rooted at node (i, j, k), while the phrase ‘gmax (i, j, k)’ refers to the maximum magnitude of the grand descendant set that is rooted at node (i, j, k).

During the process of encoding, these data are computed by scanning the first quarter of the transform coefficients matrix from the fine level to the coarse level using the following equations. This process is carried out in order to obtain the data.

$$\text{gmax}(i,j,k) = \max\{\text{dmax}(a,b,c) ; \text{dmax}(a,b,c+1) ; \text{dmax}(a,b+1,c) ; \text{dmax}(a,b+1,c+1) ; \text{dmax}(a+1,b,c) ; \text{dmax}(a+1,b,c+1) ; \text{dmax}(a+1,b+1,c) ; \text{dmax}(a+1,b+1,c+1)\} \tag{2}$$

$$\text{dmax}(i,j,k) = \max\{C(a,b,c) ; C(a,b,c+1) ; C(a,b+1,c) ; C(a,b+1,c+1) ; C(a+1,b,c) ; C(a+1,b,c+1) ; C(a+1,b+1,c) ; C(a+1,b+1,c+1); \text{gmax}(i,j,k)\} \tag{3}$$

Where nodes (a,b,c), (a,b,c+1), (a,b+1,c), (a,b+1,c+1), (a+1,b,c), (a+1,b,c+1), (a+1,b+1,c) and (a+1,b+1,c+1) are the offspring nodes rooted at node (i, j, k), due to the fact that these nodes do not have any grand descendants, the value zero is substituted for the gmax (i, j, k) function when the node (i, j, k) is located within the highest two levels.

The most significant non-zero bit plane is discovered by scanning the baseband and a tiny piece of the dmax waveform, and then transmitting the results. In order to begin the process of initializing the state table, each baseband coefficient is assigned the symbol ‘SIC’, and each full-size spatial tree is assigned the symbol ‘DES’. During the initialization process, just a small number of the coefficients are indicated.

Starting from the bitplane with the highest significance and working its way down to the bitplane with the least significance, the encoding process continues until a bit budget is reached. It is possible to compute the maximum descendant magnitude matrix, dmax (i, j, k), and the maximum grand descendant magnitude matrix, gmax (i, j, k), that are required during set partitioning by using the bitwise ‘OR’ function rather than the max function. Using bitwise ‘AND’ operations, it is possible to perform the significance analysis of coefficients while the encoding passes are being performed.

Using input rather than output, and changing the bits and signs of coefficients using bitwise 'OR' rather than verifying them with bitwise 'AND', are two examples of the slight changes made compared to the encoder. The decoder follows the same general approach as the encoder, with a few minor modifications.

It is important to note that the coefficient values and the maximum descendant magnitude values are analysed in the order of the scan during each run. Indeed, there is no multiplication involved, and the real addition is only required for indexing the coefficient coordinates in our algorithm. In other words, there is no multiplication involved.

The associated pseudo-code of the proposed HSICA is covered in Table 3.

Table 1. Summary of different Set Partitioned HSICAs (mathematical transform-based) with associated comparative analysis

HSICA	Ref	Year	Partition Type	List	CM	Embeddedness	Arithmetic Coding
3D-SPECK	[36]	2006	Zero Block Cube	List (2)	Variable	Yes	No
3D-SPEZBC	[44]	2007		List (2)	Variable		
3D-LSK	[39]	2010		Listless	Fixed		
3D-ST-SPECK	[45]	2015		List (2)	Variable		
3D-ZM-SPECK	[42]	2022		Listless	Zero		
3D-BCP-ZM-SPECK	[46]	2023		Listless	Fixed		
3D-M-ZM-SPECK	[46]	2023		Listless	Zero		
FrWF based ZMSPECK	[47]	2023		Listless	Zero		
3D-LBCSPC	[35]	2024		Listless	Fixed		
BFrWF based ZMSPECK	[47]	2025		Listless	Zero		
SFrWF based ZMSPECK	[47]	2025		Listless	Zero		
3D-CT-LSK	[48]	2025		Listless	Fixed		
3D-SPIHT	[37]	2004	Zero Tree	List (3)	Variable	Yes	No
3D-F-SPIHT	[49]	2012		List (3)	Variable		

3D-NLS	[40]	2013		Listless	Fixed		
3D-SDB-SPIHT	[50]	2017		List (3)	Variable		
3D-LEZSPC	[48]	2023		Listless	Fixed		
3D-BPEC	[46]	2023		Array (6)	Variable		
3D-MELS	[46]	2023		Listless	Fixed		
3D-LMZC	[46]	2024		Listless	Fixed		
3D-SLS	[46]	2025		Listless	Fixed		
3D-WBTC	[38]	2019	Zero Block Cube Tree	List (3)	Variable	Yes	No
3D-LMBTC	[41]	2019		Listless	Fixed		
3D-M-WBTC	[14]	2019		List (3)	Variable		
3D-LCBTC	[43]	2022		List (2)	Fixed		
3D-LBCTC	[46]	2022		Listless	Fixed		

Table 2. Description of the markers in the proposed compression algorithm

Marker	Description
SIC	The coefficient is not tested or insignificant for the current bit plane
SNC	The coefficient is newly significant to the current bit plane.
SSC	The coefficient is significant to the previous bit plane, and refinement is necessary for the current bit plane.
SCC	Similar to SIP, however, it is done while partitioning in the sorting pass. This means that the new coefficient set will be assessed for significance immediately within the same sorting pass, whereas SIP coefficients will be skipped.
DES	The coefficient is the offspring node, also known as the direct child, in a set that takes into account all of the nodes that are descendants of its parent.
GRANDES	In a set that includes all of the grand descendant nodes of its grandparent coefficient, the coefficient is the grandchild node. However, the children of the grandparents are not included in this set.

Table 3. Pseudo code for the proposed compression algorithm, 3D-M-NLS, for the hyperspectral images

```
Input     5 level wavelet transform is applied to the HS image
          3D transform HS image is converted to the 1D array '$C_i$' through the Morton mapping
          Coordinates of the
Output    Embedded Bit Stream from the encoder end to the decoder
          Initialization Pass
              Set: Number of bit planes
              Set: Threshold (Initial) T = $2^n$
              for (i,j,k)  ∈ Root
              {
                  State (i,j,k) = SIC
                  for (a,b,c) ∈ O (i,j,k)
                  {
                      State (a,b,c) = DES
                  }
              }
          Bit Plane Pass
              Insignificant Coefficient Pass
              {
                  for ((i,j,k) ∈ U)
                  {
                      if{State (i,j,k) == SIC}
                      {   Output((bit == C (i,j,k) && s)
                          if (bit)
                          {
                              Output sign {C(i,j,k)}
                              State (i,j,k) = SNC
                          }
                      }
                  }
```

```
}
Insignificant Set Pass
{
    for (i,j,k) ∈ Root
    {
        for (a,b,c) ∈  D(i,j,k)
        {
            if (State(a,b,c) = = DES)
            {
                Output (bit = (dmax(i,j,k) = SCC))
                if (bit)
                {
                    State(a,b,c) = State(a,b,c+1) = SCC
                    State(a+1,b,c) = State(a,b+1,c+1) = SCC
                    State(a+1,b+1,c) = State(a+1,b+1,c+1) = SCC
                    for (p,q,r)  ∈ O (a,b,c)
                    {
                        State(p,q,r) = GRANDES
                    }
                }
                    elseif(state (a,b,c)) = = GRANDES
                    {
                        Output(bit = (gmax (i,j,k) && s))
                        if(bit)
                        {
                            State(a,b,c) = State(a,b,c+2) = DES
                            State(a+2,b,c) = State(a,b,+2c+2) = DES
                            State(a+2,b+2,c) = State(a+2,b+2,c+2) = DES
                        }
                        elseif(state(a,b,) = = SCC)
                            {
                                Output(bit = (C(i,j,k) && s)
                                if (bit)
                                {
                                    Output sign (C(i,j,k))
                                    State(a,b,c) = SNC
                                }
                                    else
                                    {
                                           State(a,b,c) = SIC
                                    }
}
Refinement Pass
{
    for (i,j,k) ∈ U
    {
        if (State(i,j,k) = = SSC)
        {
            Output (C(i,j,k) && s)
        }
            elseif (State(i,j,k) = = SNC)
            {
                State (i,j,k) = SSC
            }
    }
}
```

4. Results and Discussion

The performance of any compression algorithm is measured on the three performance parameters named coding efficiency, coding memory, and coding complexity.

4.1. Evaluation Metrics

To evaluate the performance of the proposed HSICA, six different performance metrics are used, named as Peak Signal to Noise Ratio (PSNR), structural similarity index measure (SSIM), Feature-Similarity Index Measure (FSIM), Bjontegaard metric calculation (BD-PSNR) for coding efficiency measurement, while coding complexity is measured in the computation execution time of encoding and decoding process [51-53]. The coding memory is calculated as the memory required by the compression algorithm for the encoding or decoding process [54, 55]. The original HS image is denoted by A(x,y,z) while the reconstructed HS image after the compression process is defined as B(x,y,z) [46]. The coefficient location is denoted by 'x', 'y ', and 'z'. PSNR is a unitless parameter defined in decibels (dB) [56]. It is used to quantify the reconstruction quality of HS images affected by lossy compression of the HSICA [57-59].

$$PSNR \ = \ [20\log(MAX_A) - 10\log(MSE)] \tag{4}$$

MAX_A is the maximum value of the input HS image, while MSE is represented as the mean square error, which is defined as in the mathematical expression below.

$$MSE = \frac{1}{(N \ \times N \ \times N)} \sum_z \sum_y \sum_x [A(x,y,z) \ - B(x,y,z)]^2 \tag{5}$$

SSIM [58, 60] is a perceptual metric that quantifies image quality degradation caused by the HS image data loss during the compression process in lossy HSICAs. Mathematically, it is defined as the following mathematical expression [52]

$$SSIM\ (A,B) \ = \ \left[\frac{(2\mu_A\mu_B + \ C_1)(2\sigma_{AB} + \ C_2)}{(\mu_A^2 + \ \mu_B^2 + \ C_1)(\sigma_A^2 + \ \sigma_B^2 + \ C_2)}\right] \tag{6}$$

The relationship between the different features of the original HS image and the reconstructed HS image is measured by the Feature-Similarity (FSIM) index [43].

4.2. HS Image Data Sets

The four different HS images available in the public domain are used for the analysis. Washington DC Mall, Cuprite, Urban, and Jasper Ridge have the features of natural resources (trees, river, mountain, desert, etc) and man-made structures (roads, buildings, etc). Each HS image is cropped from the left corner to the size of an HS image cube, and a level wavelet transform is applied to the same. The transformed HS image is converted into a 1D array through the linear indexing [35].

4.3. Benchmark Hyperspectral Image Compression Algorithms

For the comparative analysis, eight benchmark transform-based set partitioned HSICAs have been used to compare the performance of the proposed HSICA. These compression algorithms are 3D-SPECK (CA-I) [36], 3D-SPIHT (CA-II) [37], 3D-WBTC (CA-III) [38], 3D-LSK (CA-IV) [39], 3D-NLS (CA-V) [40], 3D-LMBTC (CA-VI) [41], 3D-ZM-SPECK (CA-VII) [42], and 3D-LCBTC (CA-VIII) [43]. The proposed HSICA 3D-M-NLS is defined as CA-IX in the result analysis. The first three HSICAs are list-based HSICAs, while the rest of the HSICAs, except 3D-LCBTC, are listless (state table marker) HSICAs. 3D-LCBTC uses two short lists with state table markers. These compression algorithms are executed on the same hardware platform (20 GB RAM) and software environment (Windows 11 operating system and Matlab 2023A simulation software).

4.4. Coding Efficiency

The coding efficiency of any HSICA is calculated by the performance metrics named as PSNR, SSIM, and FSIM [42,61].

From Table 4, the difference of PSNR between the proposed HSICA 3D-M-NLS and 3D-SPIHT is 0.42 dB to 1.06 dB for HS image I, 0.34 dB to 1.44 dB for HS image II, 0.39 dB to 0.74 dB for HS image III, and 0.3 dB to 0.66 dB for HS image IV. In the same way, the difference between the proposed HSICA 3D-M-NLS and 3D-NLS is 0.41 dB to 1.69 dB for HS image I, 0.36 dB to 1.5 dB for HS image II, 0.41 dB to 0.83 dB for HS image III, and 0.39 dB to 0.76 dB for HS image IV. The coding efficiency of the proposed HSICA is higher than that of 3D-SPIHT and 3D-NLS because the proposed HSICA generates a higher number of newly significant coefficients than 3D-SPIHT and 3D-NLS, as we know that the weight of a newly significant bit is higher than the old refinement bit.

Table 5 provides the information about the comparative analysis of the SSIM values, which are nearly similar values for every bit rate, while the FSIM comparative analysis of two HS images is also covered in Table 5. Table 6 shows a quantitative comparison of different HSICA with the proposed HSICA for ten-bit rates for BD-PSNR.

4.5. Coding Memory

Since the proposed HSICA uses markers for the tracking of the partition coefficients or sets, it has constant coding memory, which does not depend on the bit rates. It has been observed from Table 7 that the proposed compression algorithm required a fixed-size memory of 768 KB. On the basis of coding memory, proposed HSICA outperforms 3D-SPIHT [37] and 3D-NLS [40] while it has higher coding memory demand with reference to the 3D-LSK [39], 3D-ZM-SPECK [42], 3D-LCBTC [43], and 3D-LMBTC [41].

4.6. Coding Complexity

The complexity of the compression algorithm is determined by the number of computations (logical, algebraic, and arithmetic) performed during the compression process [42,62]. The other way to determine the complexity of the compression algorithm is coding (encoding and decoding) time required by the HSICA to calculate the coefficients (encoding coefficients and decoding coefficients) [44]. It has been known that listless HSICAs require very little coding time while list-based HSICAs require a lot of coding time [48, 63]. This is due to the multiple read/write operations performed by the list-based HSICAs. From Table 8 (encoding time and decoding time), it is clear that the proposed HSICA outperformed 3D-SPIHT and 3D-NLS. This is due to the reordering of the coefficients and the effective use of the markers. Also, the proposed HSICA required a smaller number of markers than 3D-NLS, which reduces the computational time for the proposed HSICA. 3D-LSK, 3D-ZM-SPECK, and 3D-LCBTC have less coding time requirement due to the use of a smaller number of markers, as 3D-ZM-SPECK does not require any markers, while 3D-LSK and 3D-LCBTC use fewer markers than the proposed HSICA. The visual representation of the HS images (before compression & after compression process) for the four sub bands (frame 29, 59, 89, 119) for our Urban HS image and Washington DC MALL HS image is shown in Figures 1 and 2.

5. Conclusion

In the present manuscript, a highly efficient, low-memory, and fast processing HSICA has been proposed. With the comparative analysis with the 3D-SPIHT and 3D-NLS, which follow the same partition rule, 3D-M-NLS has the best performance. The high coding efficiency is achieved with the significant reordering of the output encoding coefficient generation, in which more significant information is sent earlier to get better energy compaction of transform coefficients. The coding memory is reduced due to the use of fewer markers and a listless version, while the coding complexity is also reduced significantly compared to 3D-SPIHT and 3D-NLS. Thus, proposed HSICA 3D-M-NLS can be an optimum choice for the low-resource HS image sensors. Further, coding efficiency can be improved by the use of different types of advanced mathematical transforms, such as curvelet transform, shearlet transform, etc. The coding memory demand can also be reduced by using a smaller number of markers.

Acknowledgement

The authors want to express their gratitude to Integral University, Lucknow, for providing manuscript number IU/R&D/2026-MCN0004276 for the present research work.

References

[1] Suvita Rani Sharma, Birmohan Singh, and Manpreet Kaur, “A Hybrid Encryption Model for the Hyperspectral Images: Application to Hyperspectral Medical Images,” *Multimedia Tools and Applications*, vol. 83, pp. 11717-11743, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] 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]

[3] 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]

[4] Yueyun Weng, and Cheng Le, “Hyperspectral Imaging Flow Cytometry with Spatial-Temporal Encoding Enables High-throughput Single-Cell Analysis for Biomedical Applications,” *Device*, vol. 2, no. 2, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] 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]

[6] V. Sangeetha, and L. Agilandeeswari, “Artificial Intelligence Enabled Spectral-spatial Feature Extraction Techniques for Land Use and Land Cover Classification using Hyperspectral Images – An Inclusive Review,” *The Egyptian Journal of Remote Sensing and Space Sciences*, vol. 28, no. 3, pp. 455-467, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Logesh Dhanapal, and Chyngyz Erkinbaev, “Portable Hyperspectral Imaging Coupled with Multivariate Analysis for Real-Time Prediction of Plant-based Meat Analogues Quality,” *Journal of Food Composition and Analysis*, vol. 126, pp. 1-27, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] 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]

[9] 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]

[10] Xiaotong Ma et al., “Urban Feature Extraction within a Complex Urban Area with an Improved 3D-CNN Using Airborne Hyperspectral Data,” *Remote Sensing*, vol. 15, no. 4, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Sam Navin Mohan Rajan et al., "Fuzzy Swin Transformer for Land Use / Land Cover Change Detection using LISS-III Satellite Data," *Earth Science Informatics*, vol. 17, pp. 1745-1764, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] 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]

[13] 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]

[14] 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]

[15] Mohammed Abdulmajeed Moharram, and Divya Meena Sundaram, "Dimensionality Reduction Strategies for Land Use Land Cover Classification based on Airborne Hyperspectral Imagery: A Survey," *Environmental Science and Pollution Research*, vol. 30, no. 3, pp. 5580-5602, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Tong Qiao et al., "Effective Denoising and Classification of Hyperspectral Images using Curvelet Transform and Singular Spectrum Analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 119-133, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yulei Wang et al., "Fusion of Various Band Selection Methods for Hyperspectral Imagery," *Remote Sensing*, vol. 11, no. 18, pp. 1-19, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] 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]

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

[20] 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]

[21] 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]

[22] Subhashish Nabajja, and Mahendra Kanojia, "Choledochal Cancer Region Detection in Hyperspectral Images using U-Net based Models," *International Journal of Hybrid Intelligent Systems*, vol. 21, no. 2, pp. 96-114, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] 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]

[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] 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]

[26] 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]

[27] Kaijie Shi et al., "A Spectral Difference Preservation Network based on Mamba Pyramid for Hyperspectral Image Compression," *Pattern Recognition*, vol. 175, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[28] Agnieszka C. Miguel et al., *Predictive Coding of Hyperspectral Images*, Hyperspectral Data Compression, Springer, Boston, MA, pp. 197-231, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[29] 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]

[30] 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]

[31] K.S. Gunasheela, and H.S. Prasantha, "Compressive Sensing Approach to Satellite Hyperspectral Image Compression," *Proceedings of ICTIS 2018 Information and Communication Technology for Intelligent Systems*, vol. 1, pp. 495-503, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[32] 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]

[33] 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]

[34] Daniel Báscones, Carlos González, and Daniel Mozos, “Hyperspectral Image Compression using Vector Quantization, PCA and JPEG2000,” *Remote Sensing*, vol. 10, no. 6, pp. 1-13, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[35] 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]

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

[37] 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]

[38] 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]

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

[40] V.K. Sudha, and R. Sudhaka, “3D Listless Embedded Block Coding Algorithm for Compression of Volumetric Medical Images,” *Journal of Scientific & Industrial Research*, vol. 72, pp. 735-738, 2013. [Google Scholar]

[41] 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]

[42] 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]

[43] 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]

[44] 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. 6790, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[45] 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]

[46] 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]

[47] Rajesh, Shrish Bajpai, and Naimur Rahman Kidwai, “Segmented Fractional Wavelet Filter based Low Memory Hyperspectral Image Coding Algorithm for Wireless Multimedia Sensor Networks,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 12, no. 6, pp. 65-89, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[48] Vinod Kumar Tripathi, and Shrish Bajpai, “Contourlet Transform based Listless Set Partitioned Embedded Block Coding Algorithm for Wireless Multimedia Sensor Networks,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 12, no. 6, pp. 132-146, 2025. [CrossRef] [Google Scholar] [Publisher Link]

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

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

[51] A. Said, and W.A. Pearlman, “A New, Fast, and Efficient Image Codec based on Set Partitioning in Hierarchical Trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, 1996. [CrossRef] [Google Scholar] [Publisher Link]

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

[53] 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]

[54] Shumin Liu et al., “A Comprehensive Review on Hyperspectral Image Lossless Compression Algorithms,” *Remote Sensing*, vol. 17, no. 24, pp. 1-48, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[55] Divya Sharma et al., “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]

[56] 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]

[57] 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]

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

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

[60] 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]

[61] 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]

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

[63] 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]

Appendix

Table 4. Summary of PSNR for state-of-the-art HSICA with 3D-M-NLS for 4 HS images at different bit rates

Bit Rate	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA-VII	CA-VIII	CA-IX	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA-VII	CA-VIII	CA-IX
	Washington DC Mall									**Urban**								
0.1	38.53	38.28	38.50	38.35	38.12	38.29	38.33	38.31	39.01	57.04	56.94	57.12	57.04	56.92	57.08	57.09	57.14	57.33
0.2	41.54	41.34	41.52	41.49	41.27	41.19	41.42	41.59	41.94	58.95	58.8	58.99	58.76	58.65	58.7	58.72	59.02	59.37
0.3	43.51	43.3	43.49	43.55	43.3	43.48	43.57	43.58	43.84	60.43	60.29	60.49	60.43	60.29	60.54	60.55	60.53	60.89
0.4	45.26	45.11	45.25	45.09	45.09	44.59	45.24	45.28	45.59	61.77	61.67	61.86	61.64	61.65	61.46	61.57	61.91	62.11
0.5	46.81	46.6	46.81	46.76	46.41	46.09	46.73	46.83	47.11	62.95	62.79	63.01	62.89	62.64	62.56	62.82	62.99	63.19
0.6	48.45	48.24	48.43	48.42	48.21	48.38	48.39	48.49	48.83	64.16	64	64.21	64.05	63.94	64.02	64.02	64.28	64.51
0.7	49.76	49.53	49.74	49.73	49.50	49.17	49.69	49.78	50.05	65.37	65.27	65.45	65.35	65.27	65.32	65.32	65.47	65.89
0.8	51.12	50.84	51.09	51.07	50.76	50.28	50.97	51.17	51.42	66.33	66.21	66.47	66.29	66.17	65.92	66.22	66.53	66.75
0.9	52.24	52.06	52.22	52.24	52.06	51.67	52.12	52.26	52.51	67.36	67.25	67.48	67.23	67.25	66.73	67.14	67.59	67.85
1	53.52	53.32	53.51	53.49	53.33	53.46	53.47	53.59	53.74	68.4	68.23	68.59	68.31	68.12	67.69	68.29	68.93	68.91
	Cuprite									**Jasper Ridge**								
0.1	25.64	24.67	25.77	25.65	24.61	25.6	25.79	25.49	26.11	35.08	35.11	35.07	35.29	35.04	35.06	35.03	35.14	35.41
0.2	30.92	29.44	31.03	30.88	29.33	30.77	30.87	30.84	30.2	39.35	39.13	39.6	39.40	39.01	39.11	39.41	39.51	39.58
0.3	34.55	33.36	34.58	34.55	33.27	34.42	34.59	34.61	33.98	41.72	41.89	42.4	41.95	41.74	41.71	41.92	42.62	42.33
0.4	38.05	37.04	38.15	38.05	36.97	37.50	38.16	38.18	38.01	44.52	44.41	44.81	44.55	44.31	44.52	44.56	44.97	44.98
0.5	41.27	40.51	41.37	41.32	40.45	41.17	41.26	41.39	41.21	45.98	46.33	46.78	46.26	46.24	45.91	46.14	46.89	46.91
0.6	43.46	42.58	43.57	43.47	42.5	43.36	43.43	43.52	43.38	48.17	48.19	48.62	48.31	48.07	48.18	48.23	48.83	48.62
0.7	45.55	45	45.81	45.78	44.89	45.60	45.68	45.83	45.57	49.94	50.06	50.53	50.05	49.94	49.95	49.98	50.73	50.48
0.8	47.12	46.43	47.26	47.07	46.38	47.03	47.11	47.16	47.09	51.73	51.74	52.02	51.91	51.96	52.13	52.07	52.17	52.29
0.9	48.74	47.95	48.85	48.75	47.91	48.66	48.78	48.91	48.68	52.97	53.09	53.51	53.30	53.01	52.98	53.02	53.68	53.74
1	49.83	49.24	49.98	49.86	49.22	49.68	49.71	50.01	49.96	54.77	54.72	55.13	54.86	54.62	54.78	54.92	55.21	55.38

Table 5. Summary of SSIM & FSIM for state-of-the-art HSICA with 3D-M-NLS for 2 HS images at different bit rates

Bit Rate	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA -VII	CA -III	CA-IX	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA -VII	CA-VIII	CA-IX
	Structural Similarity Index																	
	Washington DC Mall									**Jasper Ridge**								
0.1	0.587	0.587	0.585	0.584	0.587	0.589	0.59	0.589	0.591	0.437	0.431	0.437	0.436	0.430	0.437	0.437	0.437	0.432
0.2	0.677	0.675	0.678	0.677	0.678	0.677	0.677	0.678	0.681	0.518	0.513	0.518	0.518	0.514	0.518	0.518	0.518	0.512
0.3	0.713	0.713	0.713	0.713	0.712	0.714	0.714	0.715	0.716	0.561	0.558	0.56	0.563	0.56	0.565	0.565	0.56	0.558
0.4	0.766	0.765	0.765	0.767	0.765	0.771	0.766	0.767	0.766	0.585	0.582	0.585	0.588	0.582	0.588	0.588	0.587	0.582
0.5	0.79	0.788	0.787	0.789	0.789	0.788	0.787	0.791	0.788	0.603	0.601	0.603	0.608	0.606	0.611	0.611	0.603	0.6
0.6	0.814	0.815	0.814	0.814	0.815	0.814	0.814	0.816	0.816	0.617	0.616	0.617	0.619	0.615	0.618	0.618	0.619	0.618

0.7	0.83	0.83	0.829	0.833	0.833	0.836	0.829	0.837	0.835	0.626	0.626	0.626	0.632	0.628	0.633	0.632	0.627	0.629
0.8	0.849	0.848	0.852	0.852	0.853	0.851	0.849	0.855	0.854	0.634	0.633	0.634	0.637	0.635	0.639	0.639	0.64	0.633
0.9	0.873	0.869	0.872	0.872	0.869	0.872	0.871	0.874	0.871	0.641	0.641	0.641	0.643	0.642	0.642	0.642	0.643	0.64
1	0.886	0.886	0.886	0.888	0.887	0.886	0.886	0.889	0.888	0.645	0.645	0.645	0.648	0.646	0.647	0.647	0.647	0.647
	Feature Similarity Index																	
	Washington DC Mall									**Jasper Ridge**								
0.1	0.71	0.695	0.709	0.712	0.712	0.716	0.716	0.708	0.695	0.321	0.319	0.32	0.321	0.32	0.338	0.338	0.334	0.32
0.2	0.774	0.759	0.77	0.78	0.779	0.784	0.784	0.771	0.759	0.443	0.434	0.443	0.452	0.439	0.488	0.488	0.479	0.434
0.3	0.803	0.804	0.802	0.801	0.79	0.805	0.805	0.801	0.804	0.554	0.548	0.548	0.572	0.558	0.599	0.599	0.587	0.548
0.4	0.831	0.825	0.827	0.847	0.826	0.852	0.852	0.829	0.825	0.65	0.627	0.65	0.659	0.628	0.665	0.658	0.651	0.627
0.5	0.851	0.844	0.847	0.855	0.856	0.863	0.863	0.848	0.844	0.694	0.692	0.694	0.699	0.696	0.742	0.75	0.744	0.692
0.6	0.865	0.865	0.865	0.872	0.865	0.865	0.865	0.874	0.865	0.715	0.708	0.715	0.732	0.712	0.749	0.759	0.756	0.708
0.7	0.88	0.88	0.88	0.899	0.886	0.898	0.898	0.884	0.88	0.801	0.794	0.799	0.805	0.804	0.787	0.787	0.792	0.794
0.8	0.898	0.893	0.896	0.912	0.901	0.91	0.91	0.911	0.893	0.82	0.82	0.819	0.819	0.822	0.84	0.84	0.824	0.82
0.9	0.914	0.912	0.914	0.913	0.912	0.917	0.917	0.917	0.912	0.833	0.827	0.833	0.839	0.834	0.849	0.849	0.847	0.827
1	0.919	0.917	0.919	0.920	0.918	0.918	0.918	0.921	0.917	0.871	0.867	0.872	0.869	0.867	0.873	0.873	0.874	0.867

Table 6. Summary of BD-PSNR for state-of-the-art HSICA with 3D-M-NLS for 4 HS images

HS Image	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA-VII	CA-VIII
Washington DC Mall	0.6561	06857	0.5756	0.6069	0.8174	0.7491	0.6293	0.6471
Cuprite	-0.0118	0.6983	-0.0114	0.0111	0.7653	0.143	-0.0203	0.0993
Urban	0.4363	0.5304	0.4093	0.5045	0.589	0.5533	0.5006	0.3781
Jasper Ridge	0.3101	0.4453	0.1473	0.2733	0.5317	0.3837	0.3103	0.0855

Table 7. Summary of coding memory for state-of-the-art HSICA with 3D-M-NLS for 4 HS images at different bit rates

Bit Rate	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA-VII	CA-VIII	CA-IX	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA -VII	CA-VIII	CA-IX
	Washington DC Mall									**Urban**								
0.1	243.8	263.3	250.1	512	1024	12	0	300.59	768	293.2	299.4	294.3	512	1024	12	0	300.59	768
0.2	416.3	438	416	512	1024	12	0	300.59	768	478.1	529.8	483.5	512	1024	12	0	300.59	768
0.3	701.1	628.6	704	512	1024	12	0	300.59	768	841.2	864	842.7	512	1024	12	0	300.59	768
0.4	733.8	723.6	733	512	1024	12	0	300.59	768	841.2	867	842.7	512	1024	12	0	300.59	768
0.5	1048.8	1060.5	1049	512	1024	12	0	300.59	768	1076.6	1110.3	1077	512	1024	12	0	300.59	768
0.6	1191.1	1222.6	1195.4	512	1024	12	0	300.59	768	1419.4	1444.9	1424.8	512	1024	12	0	300.59	768
0.7	1277.3	1302.1	1281.7	512	1024	12	0	300.59	768	1491.8	1499.2	1493.5	512	1024	12	0	300.59	768
0.8	1407.7	1415.3	1404.4	512	1024	12	0	300.59	768	1563.4	1586.4	1564.1	512	1024	12	0	300.59	768
0.9	1702.5	1725.5	1704.6	512	1024	12	0	300.59	768	1590.3	1714.6	1589.8	512	1024	12	0	300.59	768
1	1802.5	1826.7	1724.6	512	1024	12	0	300.59	768	1889.5	1906.4	1888.9	512	1024	12	0	300.59	768
	Cuprite									**Jasper Ridge**								
0.1	277.7	277.6	282.8	512	1024	12	0	300.59	768	241.4	245.9	245.8	512	1024	12	0	300.59	768

0.2	414.5	434.3	417.2	512	1024	12	0	300.59	768	440	445.7	443.7	512	1024	12	0	300.59	768
0.3	544.3	514.7	546.3	512	1024	12	0	300.59	768	541.3	555.3	549.3	512	1024	12	0	300.59	768
0.4	601.9	576.5	594.5	512	1024	12	0	300.59	768	729.6	759.5	741.6	512	1024	12	0	300.59	768
0.5	671.2	701.9	674.5	512	1024	12	0	300.59	768	821.6	808.9	827.9	512	1024	12	0	300.59	768
0.6	854	783.7	857.6	512	1024	12	0	300.59	768	1099.9	1123.2	1106.7	512	1024	12	0	300.59	768
0.7	947.5	865.5	971.7	512	1024	12	0	300.59	768	1123.8	1145.9	1131.2	512	1024	12	0	300.59	768
0.8	1065.3	964.6	1057.3	512	1024	12	0	300.59	768	1178.9	1192.7	1189.1	512	1024	12	0	300.59	768
0.9	1158.5	1182.4	1159.5	512	1024	12	0	300.59	768	1443.2	1468	1450.3	512	1024	12	0	300.59	768
1	1286.1	1308.2	1292.1	512	1024	12	0	300.59	768	1492.7	1503.8	1532.6	512	1024	12	0	300.59	768

Table 8. Summary of encoding time & decoding time for state-of-the-art HSICA with 3D-M-NLS for 4 HS images at different bit rates

Bit Rate	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA-VII	CA-VIII	CA-IX	CA-I	CA-II	CA-III	CA-IV	CA-V	CA-VI	CA-VII	CA-VIII	CA-IX
	Encoding Time																	
	Washington DC Mall									**Urban**								
0.1	25	7.5	6.5	0.8	0.91	3.9	1.78	0.76	0.84	15.6	6.8	6.9	1.6	1.81	3.5	2.25	1.53	1.71
0.2	57.9	25.8	24.8	1.1	1.21	5.1	2.81	1.04	1.11	49.5	19.7	19.1	3	3.19	5.6	3.5	2.96	3.02
0.3	92.1	37.5	32	1.5	1.65	7.7	3.68	1.41	1.53	85.7	48.9	26.4	4.1	4.37	7.8	4.57	4.02	4.31
0.4	269.7	117.9	195.5	2	2.12	9.7	5.69	1.93	2.01	312.4	102.1	191.8	5.6	5.89	10.7	6.09	5.55	5.73
0.5	414.8	140.1	211.2	2.5	2.64	11.3	7.41	2.44	2.52	416.2	158.1	253.6	7.1	8.24	11.7	7.06	7.07	8.09
0.6	576	166.4	247.9	2.9	3.02	13.3	7.99	2.82	2.94	886.9	206.7	300.3	8.8	9.07	13.2	8.15	8.64	8.84
0.7	887.5	405.7	625	3.2	3.37	18.1	9.66	3.13	3.21	905.1	211.9	371.7	9.5	10.71	16.9	9.4	9.39	10.54
0.8	1130.5	474.2	710.2	3.8	3.96	20	9.91	3.85	3.74	1125.2	541.8	788.7	11.1	11.34	18.3	10.6	11.11	10.91
0.9	1334.6	555.7	746	4	4.14	20.6	12.5	4.04	4.08	1542.7	774.1	1067.1	12.7	13.02	19.5	12.16	12.67	12.84
1	1497.5	575	804	4.41	4.57	21.1	13.2	4.38	4.43	1702.8	780.2	1184.5	14.4	14.53	20.9	12.96	14.39	14.09
	Cuprite									**Jasper Ridge**								
0.1	17.3	6.3	4.7	0.9	1.12	3.2	1.78	0.86	0.97	21.1	7.6	6.4	0.9	1.09	3	1.77	0.88	0.91
0.2	55.8	26	16.6	1.2	1.54	6.8	3.01	1.09	1.48	54.2	20.6	17.7	1.2	1.34	5.2	2.84	1.11	1.21
0.3	107.9	45.5	39.1	2	2.27	7.1	4.08	1.92	2.21	100.6	39.4	42.8	1.5	1.64	7.5	4.76	1.47	1.58
0.4	182.3	75.6	68.2	2.1	2.41	9.2	5.21	2.07	2.29	150.9	47.8	70.7	2.2	2.32	9.3	5.90	2.18	2.11
0.5	276.1	95.4	93.3	2.2	2.58	11.1	6.32	2.11	2.52	315.3	101.6	182.4	2.6	2.74	10.9	6.24	2.58	2.49
0.6	298.4	161.7	155.7	3.4	3.61	12.9	7.55	3.24	3.54	356	115.3	227.5	3.03	3.21	13.5	7.65	2.97	2.97
0.7	438.8	179.2	202.2	3.9	4.21	15	8.76	3.79	4.11	426.1	232.3	480.9	3.2	3.42	15.2	8.70	3.14	3.28
0.8	558.7	198.5	358.5	4.2	4.48	16.5	9.66	4.02	4.29	585.7	382.3	676.4	3.7	3.95	17.8	14.45	3.62	3.74
0.9	656.1	282.8	371	4.4	4.69	18.1	11.2	4.12	4.53	701.2	415	771.9	4	4.29	18.8	12.98	4.04	4.01
1	905.1	364	652.5	5	5.23	20.3	15.9	5.02	5.08	757.3	425.4	942.8	5.1	5.34	22.7	15.84	5.07	5.21
	Decoding Time																	
	Washington DC Mall									**Urban**								
0.1	17.4	6.1	5	0.7	0.79	2.3	1.71	0.64	0.71	11.8	4.3	4.80	1.5	1.67	1.7	1.9	1.48	1.51
0.2	48.8	24.8	22.5	1.07	1.07	3.3	2.71	1.01	0.99	41.6	16.8	16.4	2.4	2.94	3.2	2.9	2.36	2.74
0.3	75.4	34.8	28.5	1.45	1.43	4.9	3.59	1.33	1.38	72.3	42.5	23.2	3.2	4.09	4.7	3.65	3.14	3.91

0.4	264.2	106.3	180.4	1.7	1.94	8.1	5.41	1.62	1.79	292.2	88.3	184.5	4.7	5.74	6	4.66	4.63	5.52
0.5	339.1	135.4	191.7	2.2	2.31	7.7	6.82	2.07	2.15	388.4	148.8	243.2	5.4	7.95	7.3	5.4	5.33	7.67
0.6	532.4	149.6	244.6	2.6	2.79	9.8	7.95	2.48	2.64	487.2	197.2	294.7	6.1	8.87	8.6	5.89	6.07	8.49
0.7	807.6	327.1	558	2.7	3.04	11.6	8.8	2.53	2.91	592.7	201.1	365.7	8.2	10.52	10	7.08	8.11	10.02
0.8	1058.1	448.9	675.3	3.1	3.67	13.4	9.38	3.09	3.38	1066.7	507.4	771.7	8.8	11.08	11.4	7.96	8.72	10.79
0.9	1142.3	486.2	725	3.2	3.93	13.6	11.8	3.33	3.74	1506.4	749.8	1052.3	9.4	12.84	12.6	8.76	9.33	12.41
1	1289.7	504	774	3.7	4.24	15.5	12.3	3.87	4.02	1661.8	761.9	1173.4	10	14.22	13.9	9.5	10.04	13.94
	Cuprite									**Jasper Ridge**								
0.1	13.4	5	3.1	0.7	0.94	2.2	1.7	0.63	0.91	15.3	7.41	4.57	0.75	0.97	1.9	1.73	0.71	0.89
0.2	46.7	22.1	14.6	1	1.32	4.8	2.92	0.91	1.21	37	17.6	15.3	1.1	1.22	3.7	2.7	1.04	1.18
0.3	93.7	40.2	35.4	1.8	2.04	5.5	3.98	1.69	1.97	84.7	37.1	40	1.4	1.51	6.1	4.73	1.41	1.47
0.4	162.5	70.1	65.8	1.9	2.31	7	5.07	1.81	2.11	128.8	45	69	2.1	2.17	8.7	5.14	2.04	2.04
0.5	236.1	88.3	91.5	2	2.45	8.5	6.01	1.92	2.31	290.8	98.5	178.4	2.5	2.59	9.3	6.06	2.47	2.34
0.6	281.2	160.9	149	2.9	3.38	10.2	7.17	2.79	3.19	330.9	101.8	217.1	2.8	2.98	10.5	7.23	2.74	2.68
0.7	435	175.8	196.8	3.1	4.03	11.9	8.28	2.97	3.89	386.4	232.2	432.1	3	3.17	11.3	8.41	2.97	2.94
0.8	525.9	195.3	316	3.8	4.24	13.1	9.21	2.61	4.08	487.8	382.2	608.4	3.5	3.72	12.9	9.86	3.46	3.49
0.9	599.2	273.5	366.9	4	4.47	15	10.3	3.83	4.22	667.4	402.7	673.5	3.6	3.97	17.5	11.73	3.55	3.75
1	884.4	346.6	596	4.5	5.07	15.9	14	4.38	4.89	726.9	421.3	923.1	4.3	5.02	21.8	12.31	4.24	4.82

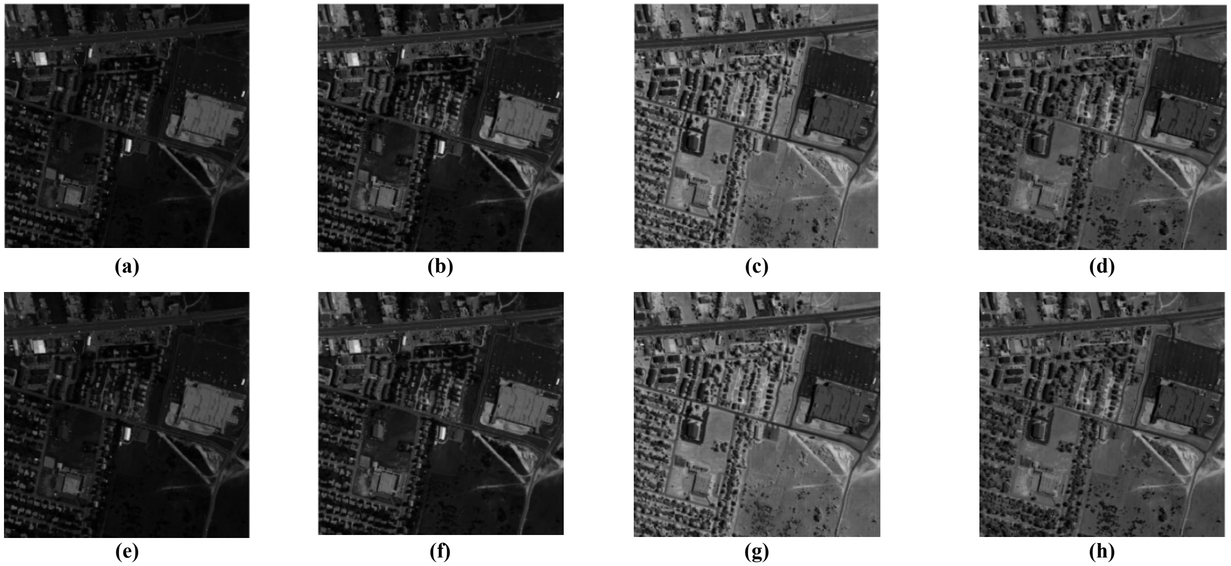

Fig. 1 Urban HS image before compression process (a) Frame 29, (b) Frame 59, (c) Frame 89, (d) Frame 119, urban HS image after compression process with CR=10 (e) Frame 29, (f) Frame 59, (g) Frame 89, and (h) Frame 119.

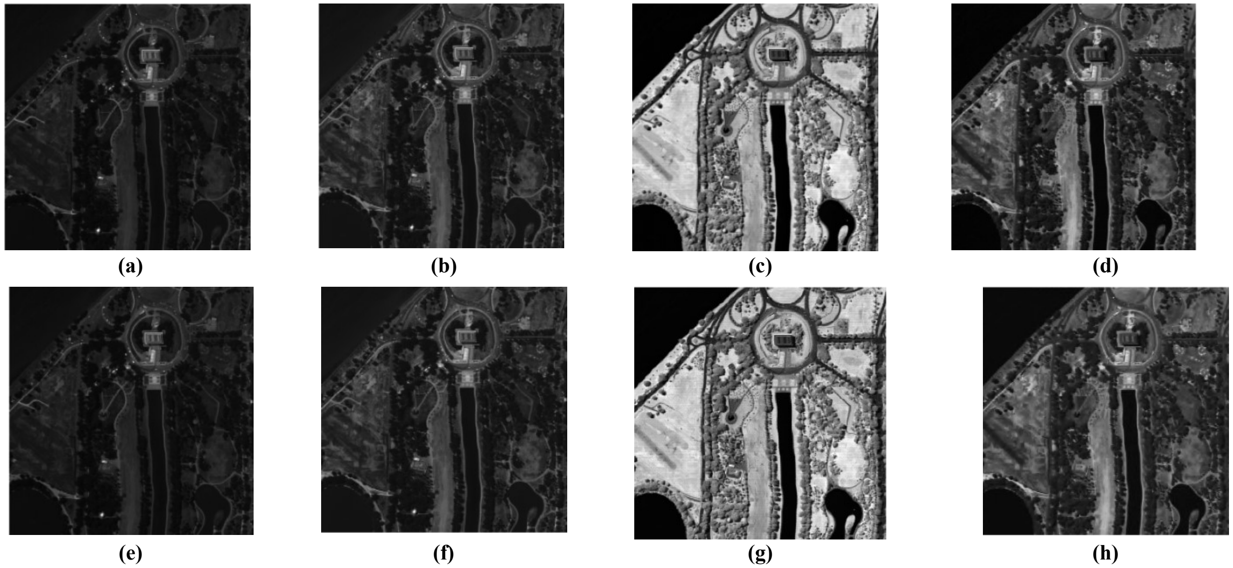

Fig. 2 Washington DC MALL HS image before compression process (a) Frame 29, (b) Frame 59, (c) Frame 89, (d) Frame 119, washington DC MALL HS image after compression process with CR=10 (e) Frame 29, (f) Frame 59, (g) Frame 89, and (h) Frame 119.