

Review

A Comprehensive Review on Hyperspectral Image Lossless Compression Algorithms

Shumin Liu ^{1,*} , Fahad Saeed ¹ , Zhenghui Yang ² and Jie Chen ¹ 

¹ School of Marine Science and Technology, Northwest Polytechnical University, Xi'an 710072, China; fahadsaeed@sbbu.edu.pk (F.S.); jie.chen@nwpu.edu.cn (J.C.)

² Aerial Photogrammetry and Remote Sensing Group Co., Ltd., Xi'an 710100, China; yangzhenghui.1986@163.com

* Correspondence: liu_shumin@ieee.org

Highlights

What are the main findings?

- The review provides a focused and systematic analysis of lossless hyperspectral image compression, categorizing existing algorithms into transform-based, prediction-based, and deep learning-based methods.
- It uniquely emphasizes the second stage of the compression pipeline—scanning and encoding order optimization—an aspect often overlooked in previous reviews but crucial for improving compression efficiency.

What is the implication of the main findings?

- By distinguishing the principles and performance characteristics of different algorithm classes, the review offers a comprehensive framework that helps researchers and practitioners select suitable lossless compression schemes for diverse remote-sensing applications.
- The analysis highlights future research directions, including the integration of deep learning with reversible transforms and the exploration of adaptive scanning strategies to enhance compression ratio and computational efficiency.

Abstract

The rapid advancement of imaging sensors and optical filters has significantly increased the number of spectral bands captured in hyperspectral images, leading to a substantial rise in data volume. This creates major challenges for data transmission and storage, making hyperspectral image compression a crucial area of research. Compression techniques can be either lossy or lossless, each employing distinct strategies to maximize efficiency. To provide a more focused and comprehensive analysis, this review concentrates exclusively on lossless compression, which is categorized into transform, prediction, and deep learning-based methods. Each category is systematically examined, with particular emphasis on the underlying principles and the strategies adopted to enhance compression performance. In addition to the core algorithms, encoding and scanning orders are also discussed, which is an essential aspect that is often overlooked in other reviews. By integrating these aspects into a unified framework, this paper offers an up-to-date and in-depth overview of the methodologies, trends, and challenges in lossless hyperspectral image compression.

Keywords: hyperspectral images; lossless compression; transform methods; prediction methods; deep learning



Academic Editor: Pedro Melo-Pinto

Received: 2 November 2025

Revised: 29 November 2025

Accepted: 4 December 2025

Published: 8 December 2025

Citation: Liu, S.; Saeed, F.; Yang, Z.; Chen, J. A Comprehensive Review on Hyperspectral Image Lossless Compression Algorithms. *Remote Sens.* **2025**, *17*, 3966. <https://doi.org/10.3390/rs17243966>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With advancements in imaging technology, hyperspectral images can now capture data across thousands of spectral bands, ranging from 400 nm to 2500 nm, far beyond the visible light spectrum. This enables hyperspectral images to store significantly more information, including details undetectable in traditional RGB images, making them indispensable in various remote sensing applications, such as surveillance [1], mineralogy [2], industrial inspection [3], agricultural analysis [4], and environmental monitoring [5]. However, the enhanced information content in hyperspectral images comes at the cost of increased file sizes, necessitating effective compression algorithms to manage storage space and ease data transmission.

Hyperspectral image compression algorithms can be either lossy or lossless. As their names imply, lossy compression is an irreversible process that permanently discards a substantial portion of information, while lossless compression is fully reversible, allowing for perfect reconstruction of the original image, albeit at the cost of a lower compression ratio. Additionally, the design principles behind these two methods differ significantly: lossy compression achieves higher compression ratios primarily by eliminating data that is less perceivable by human vision, whereas lossless compression focuses on decorrelating and removing repetitive information. For example, JPEG2000 [6], which supports both types of compression, relies heavily on truncating less significant bits using the Embedded Block Coding with Optimized Truncation (EBCOT) scheme in its lossy mode; whereas in lossless mode, the compression mainly stems from encoding repetitive values such as sequences of “0”s. In terms of noise presented in the original data, lossy compression may unintentionally remove the unwanted sensor noise. However, this inevitably affects subtle spectral features or introduces artifacts, leading to unwanted information loss that degrades the image’s scientific value. In contrast, lossless compression preserves all original data, including both the valuable signal and the inherent noise, thereby guaranteeing perfect reconstruction.

Due to its reversibility, lossless compression is particularly valuable in high-precision applications where any data loss is unacceptable. This includes areas such as medical imaging [7] and materials analysis [8], where hyperspectral imaging facilitates noninvasive diagnosis by capturing the absorption, fluorescence, and scattering characteristics of the scanned samples. Additionally, lossless compression is also crucial for managing image datasets, where the original data must be retrievable after compression to maintain application performance. To provide a more focused and thorough analysis, this review will concentrate on reviewing lossless compression methods only.

While several reviews on hyperspectral image (HSI) compression [9–11] exist, they generally present broad overviews that cover both lossy and lossless techniques, with a stronger emphasis on the lossy side. However, since the underlying principles and objectives of lossy and lossless compression differ fundamentally, such surveys cannot provide an in-depth treatment of the lossless domain. More recent work, such as [10], narrows its focus to deep learning approaches, highlighting their rapid progress in the field. Yet, deep learning methods are inherently better aligned with lossy scenarios, leaving lossless methods relatively underexplored. This review addresses that gap by offering an up-to-date, focused analysis of lossless HSI compression algorithms. In addition, it contributes unique value by devoting particular attention to the often-overlooked second stage of the compression pipeline: determining optimal scanning and encoding orders, which play a pivotal role in efficiently compressing transformed data or prediction residuals.

Understanding the compression processes allows for straightforward inference of the corresponding decompression algorithms, which will not be covered in this review. It is important to note that certain algorithms, particularly prediction methods, rely heavily on

information from preceding bands or rows. Consequently, different prediction rules are applied to the first band and row, as well as to some boundary pixels. This review will not address these cases unless explicitly stated. By focusing on these core categories and their methodologies, we aim to provide a comprehensive framework for understanding the landscape of lossless compression in hyperspectral imaging. This review categorizes lossless compression algorithms for hyperspectral images into three primary classes: transform, prediction, and deep learning methods. While these approaches employ distinct principles for redundancy removal, they share a common final stage of entropy encoding, as illustrated in Figure 1. This stage, which is seldom discussed, determines the scanning and encoding order used to efficiently compress the transformed data or prediction residuals, and is described in detail in Section 2. Among the three classes, transform methods generally apply global, reversible transforms to decorrelate the entire image block, producing a compact set of coefficients for encoding, as in Section 3. In contrast, prediction methods operate locally and sequentially, estimating the current pixel value from its neighbors and encoding the resulting prediction residuals, discussed in Section 4. Meanwhile, deep learning methods represent an emerging approach that leverages deep networks to learn non-linear, data-driven mappings, described in Section 5. Lastly, an overarching quantitative comparisons is analyzed in Section 6.1 and conclusion remark is given in Section 7.

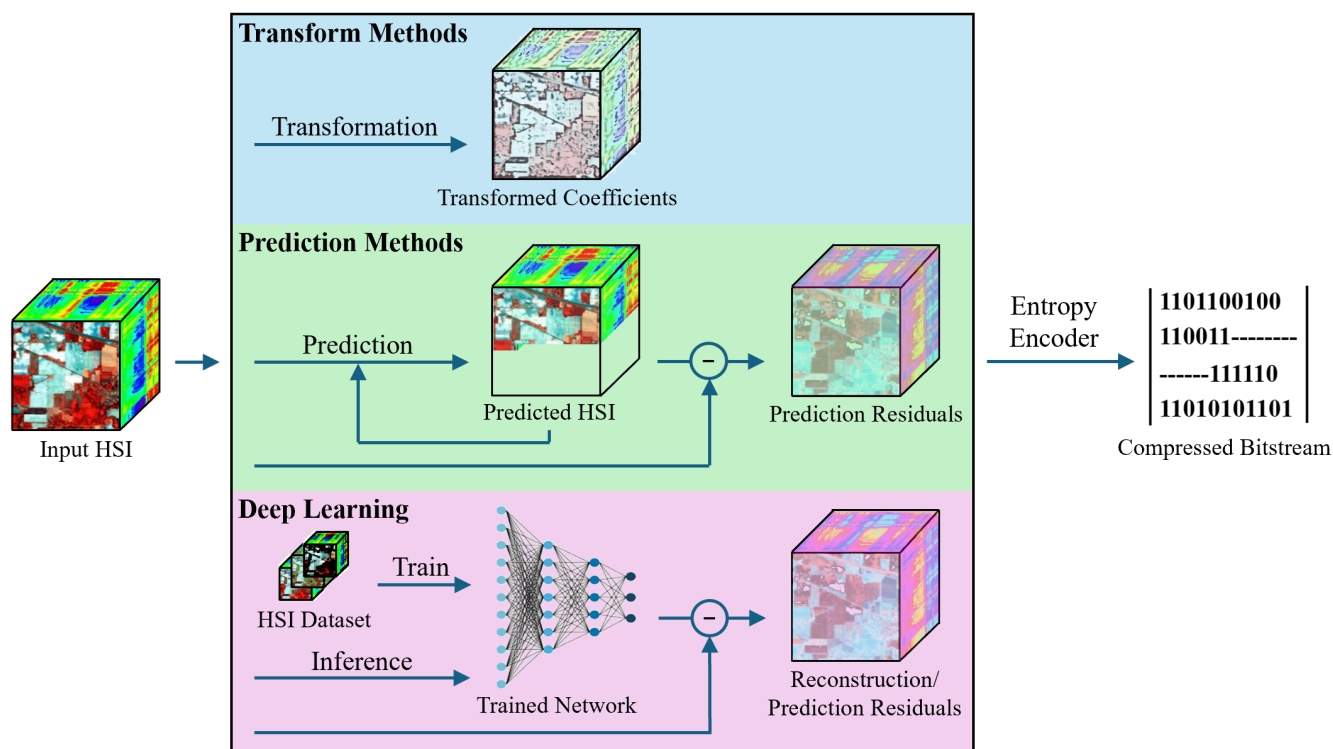


Figure 1. The general framework of hyperspectral image lossless compression.

1.1. Unique Characteristics of Hyperspectral Images

To effectively design and evaluate lossless compression algorithms for HSI, it is crucial to first understand the intrinsic properties that distinguish HSI from conventional RGB or grayscale images. These characteristics directly shape the priorities and challenges in compression algorithm design. This section elaborates on these properties and their implications for lossless compression.

- **High Dimensionality:** Unlike RGB images that contain only three channels, HSIs consist of dozens to thousands of spectral bands, with each pixel representing a full, high-resolution spectrum. This high dimensionality results in extremely large data

volumes, often reaching gigabytes per scene. Consequently, the goal of compression shifts from simple storage reduction to enabling practical data transmission and archiving. This demand necessitates highly efficient algorithms capable of exploiting all forms of data redundancy.

- **Strong Spectral Correlation:** A defining feature of HSIs is the strong correlation between adjacent spectral bands. Furthermore, the spectral correlation is often stronger than the spatial correlation within a single band. Therefore, algorithms that effectively leverage spectral correlation typically achieve superior compression performance compared with approaches that treat bands independently. This is a fundamental distinction from traditional 2D image compression, where the emphasis lies primarily on spatial redundancy.
- **Spatial-Spectral Heterogeneity:** Although spectral correlation is generally strong, its degree can vary significantly across different spatial regions and spectral ranges. For instance, homogeneous regions exhibit high correlation, whereas areas with sharp edges or material boundaries exhibit weaker correlation. This spatial-spectral heterogeneity complicates algorithm design, as a one-fits-all algorithm may be suboptimal. Effective methods must therefore adaptively balance the use of spatial and spectral context.
- **Sensor-Specific Noise:** Hyperspectral sensors are prone to various noise sources and artifacts, including thermal noise, shot noise and striping. In lossless compression, all information must be perfectly preserved. Because noise introduces randomness that is inherently uncorrelated with both spatial and spectral neighbors, it reduces the predictability of pixel values, which directly limits the compression ratios. Compression algorithms must be robust in the presence of noises without suffering substantial performance degradation.

1.2. Evaluation Metrics

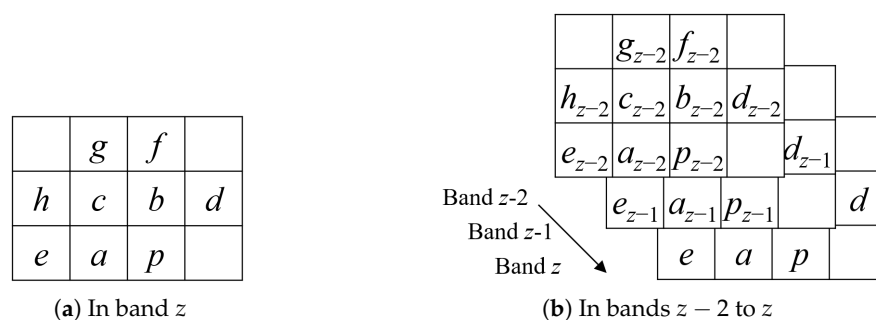
Unlike lossy compression, which requires evaluating the quality of the decompressed image using metrics such as peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [12], and spectral angle mapper (SAM) [13], lossless compression ensures that the decompressed image is identical to the original input. As a result, image quality is inherently preserved at the maximum level. Therefore, the evaluation of lossless compression focuses more on compression efficiency, typically measured by Compression Ratio (CR) or Bits Per Pixel Per Band (BPPPB). Specifically, CR is defined as the ratio between the size of the original image and the size of the compressed bitstream, while BPPPB represents the average number of bits required to encode one pixel in a single spectral band. Further practical considerations include memory cost and computational complexity, which can be analyzed using big O notation, or measured in bytes and floating-point operations per second (FLOPs).

1.3. Notations

The main notations used in this review is summarized in Table 1. Scalars are represented in non-bold italic font, vectors and matrixes are represented in bold non-italic font. Both $I_z(x, y)$ and p are defined as the current compressing pixel, which are used interchangeably, whichever is more suitable. The neighbouring pixels around p is defined as $a \sim h$ in band z and $a_{z-1} \sim h_{z-1}$ in band $z - 1$, whose relative positions are visualized in Figure 2a,b.

Table 1. Notations.

x	Horizontal position of the current compressing pixel
y	Vertical position of the current compressing pixel
z	Band index of the current compressing pixel
X	Height of image
Y	Width of image
Z	Number of bands of image
$\mathbf{I}$	The input image
$\mathbf{I}_z(x, y)$	Pixel value of image $\mathbf{I}$ at x^{th} column y^{th} row z^{th} band
$\hat{\mathbf{I}}_z(x, y)$	Predicted pixel value of image $\mathbf{I}$ at x^{th} column y^{th} row z^{th} band
$\bar{I}_z$	Mean of $\mathbf{I}_z$
p	Pixel value of $\mathbf{I}_z(x, y)$
$\hat{p}$	Predicted pixel value of $\mathbf{I}_z(x, y)$
p_{z-1}	Pixel value of $\mathbf{I}_{z-1}(x, y)$
$a \sim h$	Neighbours of $\mathbf{I}_z(x, y)$ in band z
$a_{z-1} \sim h_{z-1}$	Neighbours of $\mathbf{I}_{z-1}(x, y)$ in band $z - 1$
$\mathbf{S}_z$	A vector storing $a \sim h$, i.e., $[a, b, \dots, h]^T$
δ	Difference image between the current band and previous band
$\delta_z(a)$	Pixel value of difference image at band z position a

**Figure 2.** Position of neighbouring pixels around p .

2. Scanning and Encoding Patterns

2.1. Scanning Patterns

HSI is typically acquired in one of several modes, such as Band Interleaved by Line (BIL), Band Interleaved by Pixel (BIP), or Band Sequential (BSQ). For the purpose of compression, it is generally assumed that the entire data cube is available to the algorithm. While specialized compression methods tailored to specific acquisition modes do exist [14,15], they typically build upon a core algorithm that remains unchanged, with only marginal impacts on throughput and memory cost. The scanning patterns used during compression can be broadly categorized by their dimensional traversal strategy. Most patterns originate from 2D image processing and can be applied to HSI in two fundamental ways: as pure 2D spatial scans applied band-sequentially, or extended to true 3D patterns that mix spatial and spectral dimensions. Among them, 2D patterns apply scanning along the spatial dimensions first, processing each spectral band independently before moving to the next band; 3D patterns mix spatial and spectral dimensions during scanning, treating the data as a unified volume to leverage correlations across all dimensions simultaneously.

2.1.1. 2D Scanning Patterns

Raster Scan

This is the most widely used and fundamental scanning order in most algorithms. It processes or encodes the input image sequentially, from left to right, top to bottom, and from the first band to the last, as illustrated in Figure 3a.

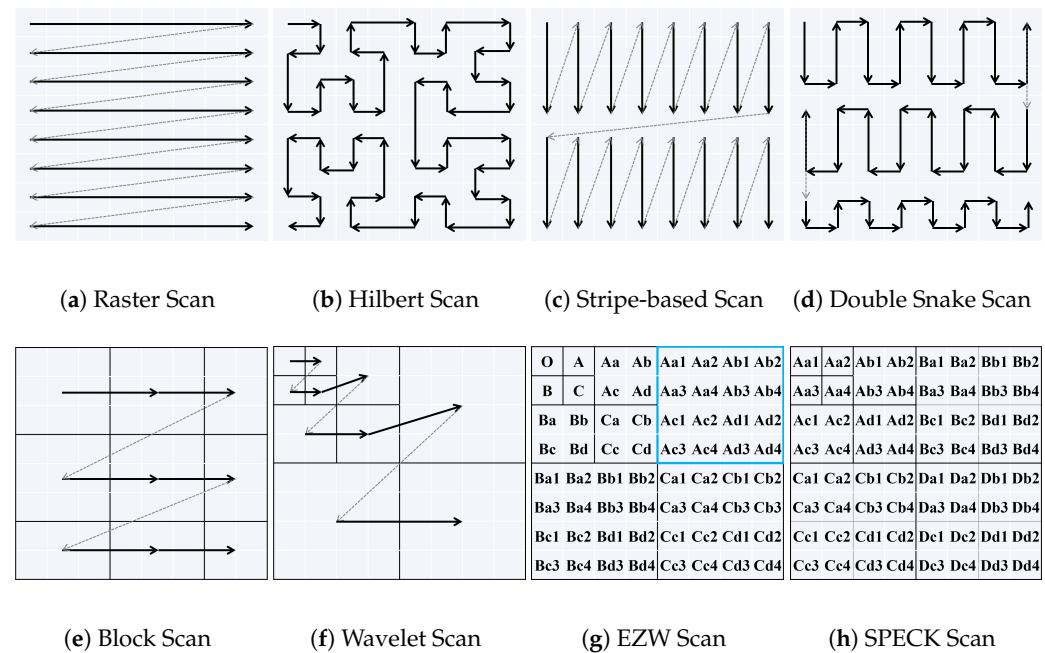


Figure 3. Demonstration of different scanning orders.

Hilbert Scan

Designed to improve the correlation between neighboring pixels during encoding, the Hilbert scan [16] is depicted in Figure 3b. However, despite its attempt to improve compression ratios, ref. [17] demonstrated that the simpler Raster scan is often more effective than the more complex Hilbert scan.

Stripe-Based Scan

This scanning order is more widely adopted than the Hilbert scan. It has been implemented in JPEG2000 [6] for processing monochromatic and RGB images, as shown in Figure 3c. This method is both simple and effective, fully leveraging the spatial correlation between pixels.

Double Snake Scan

Similar to the stripe-based scan, the Double Snake scan [18] is shown in Figure 3d. It further reduces the average distance between previously encoded pixels and the current pixel.

Block Scan

In this method, the input image is divided into non-overlapping blocks, with each block fully processed before moving on to the next in the raster order, as shown in Figure 3e. Within each block, any scanning pattern, such as Raster, Hilbert, stripe-based, or double snake can be applied.

2.1.2. 3D Scanning Patterns

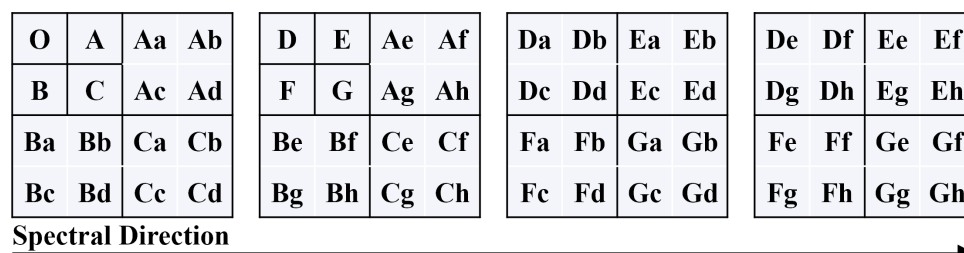
3D Extensions of 2D Patterns

Fundamental 2D patterns like raster and Hilbert can be extended to 3D through mathematical generalization such as 3D Hilbert curves [19]. However, these are less commonly used in HSI compression due to computational complexity and the superior performance of specialized 3D patterns like those used in wavelet-based methods.

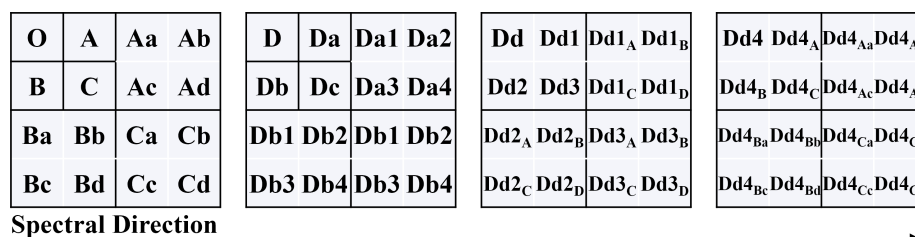
Wavelet Scan

Wavelet coefficients are typically encoded sub-band by sub-band, starting from the lowest frequency sub-band and progressing to the highest. An example of a 3-level Discrete Wavelet Transform (DWT) is illustrated in Figure 3f. Typical wavelet coding techniques include Embedded Zerotrees of Wavelet Transforms (EZW) [20,21], Set Partitioning in Hierarchical Trees (SPIHT) [22], and Set Partitioning Embedded bloCK (SPECK) [23]. These methods can be lossy by discarding less significant bits (bit position-wise) or less important bits (human perception-wise). In lossless compression, this discarding step is omitted. Furthermore, any scanning pattern can be applied within each sub-band.

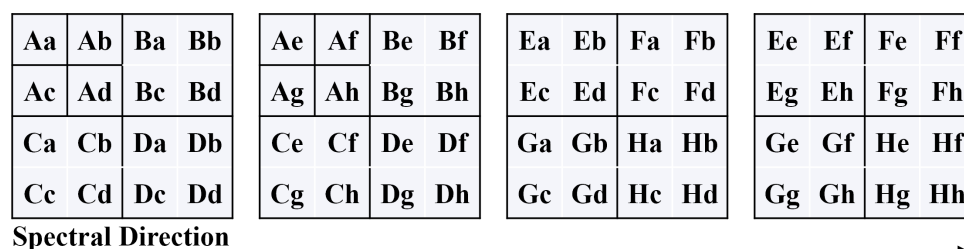
- **Embedded Zerotrees of Wavelet Transforms:** This method [20,21] is also named as Embedded Zerotree Wavelet (EZW). The scanning order of EZW is shown in Figure 3g, where the indexes represent the hierarchical structure. All coefficients, except those in the highest and lowest decomposition levels, have four direct descendants. For example, coefficients Aa, Ab, Ac, and Ad are the direct descendants of coefficient A, while Aa1, Aa2, Aa3, and Aa4 are the direct descendants of Aa. By inheritance, all coefficients within the blue box are descendants of A. If a coefficient and all its descendants (both direct and indirect) are insignificant, they are encoded into a single output, reducing the file size.
- **Set Partitioning in Hierarchical Trees:** SPIHT [22] is an improved version of EZW. One drawback of EZW is that it requires five outputs if a coefficient is significant but all its descendants are insignificant. SPIHT addresses this by separating the encoding of the coefficient from its descendants. As a result, the same example can be encoded with only two outputs instead of five. The extension of EZW and SPIHT from 2D to 3D can follow similar scanning orders, where the encoding pattern is applied separately to each bit-plane and spectral band. Alternatively, a more systematic extension can be implemented by increasing the number of direct descendants for each coefficient from four to eight (or from three to seven for low-frequency coefficients) [24], as shown in Figure 4a, which illustrates a 2-level 3D DWT. However, this structure is broad and shallow, lacking optimization of wavelet coefficients' inter-dependencies. An optimized hierarchical structure is proposed in [25], capping the number of direct descendants at four, as shown in Figure 4b. Besides modifying the scanning order, a variation of SPIHT, known as 3D-Wavelet Block Tree Coding (3D-WBTC) [26], classifies each block into three types and encodes them using different rules.
- **Set Partitioning Embedded Block:** The aforementioned scanning orders utilize the spatial and spectral consistency of wavelet coefficients, where coefficients derived from the same set of pixels tend to have similar magnitudes. In contrast, SPECK [23] leverages sub-band consistency, where coefficients within the same sub-band exhibit similar magnitudes. This leads to a different hierarchical structure, as depicted in Figure 3h. If all coefficients with identical initial indices, such as "Ba" to "Bh", are insignificant, they can be encoded as a single output. The 3D extension of SPECK is done by grouping wavelet coefficients into 3D cubes instead of 2D blocks, as illustrated in Figure 4c. It is noteworthy that the concept of 3D SPECK can be applied to non-wavelet coefficients as well, such as the k^2 -raster coding method in [27]. Besides, SPECK can be further enhanced by ZM-SPECK [28], which reduces the memory requirements while encoding.



(a) Symmetric 3D SPIHT



(b) Optimal 3D SPIHT



(c) 3D SPECK

Figure 4. Demonstration of different extension of EZW and SPIHT from 2D to 3D.

2.2. Encoding Methods

As illustrated in Figure 1, all lossless compression methods require an encoding step as the final stage. It has been established in the literature that both transformed coefficients [29] and prediction residuals [30] can be approximated by Laplace distributions. Consequently, the same entropy encoding techniques are applied to both transform and prediction methods. These encoders are categorized by their fundamental processing unit: pixel-based methods that encode complete integer values, and bit-based methods that operate on individual bitplanes. Pixel-based encodings typically adheres to the scanning orders introduced in Section 2.1. In contrast, bit-based encoding follows a hierarchical structure where the data is first organized into bitplanes, which are processed from the most significant to the least. For each bitplane, the encoder applies the spatial-spectral scanning patterns from Section 2.1 before proceeding to the next bitplane.

2.2.1. Pixel-Based Encoding

Straight Coding

Straight coding encodes pixel values in their binary form. It is often applied to the first pixel (or the first few rows) of a hyperspectral image compressed by predictive methods [31]. These pixels are not subject to prediction due to the unavailability of surrounding information.

Huffman Coding

When the distribution of the encoded pixels is known, Huffman Coding (HC) serves as an optimal prefix encoder, designed under the principle that more frequent data are

encoded with fewer bits. Hyperspectral image lossless compression algorithms that use HC include [31–34], and they can operate in either adaptive or hard-coded modes. In hard-coded mode, a hypothetical frequency distribution initializes the lookup Huffman codebook, which remains unchanged throughout encoding. In adaptive mode, the codebook is updated after encoding each piece of data. Due to the complexity of constructing a codebook, hard-coded HC is generally preferred [32].

Run Length Coding

Run-length coding (RLC) is particularly effective for data with consecutively repeating patterns. For example, the sequence “0000101010” can be encoded as 4 “0”s followed by 3 “10”s. To further enhance its effectiveness, RLC is often used in conjunction with other encoding methods. In JPEG, RLC is applied to compress high-frequency Discrete Cosine Transform (DCT) coefficients, which are predominantly “0”s, whereby the length of consecutive “0”s is further Huffman encoded. RLC has also been used in hyperspectral image lossless compression to encode high-frequency coefficients after 3D DCT [35].

Golomb Coding

There are various types of Golomb coding, including standard Golomb coding, Golomb-Rice coding, and exponential-Golomb coding. All of these methods are adaptive, adjusting a positive integer denominator $M \in \mathbb{Z}^+$.

- **Golomb Coding** It is an optimal prefix code when the input follows a Laplace distribution. It encodes a non-negative integer pixel value $p \in \mathbb{Z}^{0+}$ into a codeword consisting of two parts: a prefix and a suffix. The prefix contains the unary code of $\lfloor p/M \rfloor$, and the suffix represents the truncated binary form of $\text{mod}(p, M)$, with a bit length of $\lceil \log_2(M) \rceil$. Here, $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ represent the floor and ceiling operations respectively, $\text{mod}(p, M)$ is the remainder when p is divided by M . The most widely used implementation of M can be determined from the mean absolute error of previously encoded data. Other adaptations of M include geometrical distribution [36], correlation-based adaptation [37] and deep learning-based estimation [38].
- **Golomb-Rice Coding** This variation is a simplified variant of standard Golomb coding, where M is restricted to a power of 2, i.e., $M = 2^k$ for some non-negative integer $k \in \mathbb{Z}^{0+}$. The value of M is computed by rounding down the sample mean of previously encoded data to the nearest power of 2. This constraint significantly improves computational efficiency, making Golomb-Rice coding popular in many compression algorithms [39].
- **Exponential-Golomb coding** This encodes a pixel value p into a codeword composed of two parts: a prefix and a suffix. The prefix contains the unary code of $u = \lfloor \log_2(\frac{p+M}{M}) \rfloor$, and the suffix holds the truncated binary form of $\text{mod}(p + M, 2^{u+k})$, with a bit length of $u + k$. It is worth noting that exponential-Golomb coding is quite similar to the Huffman encoding used for DC coefficients in JPEG. A similar encoder named integer square root is described in [40], where the square root value of p is recorded in 4-bits binary form, and the residual is encoded as unary.

Lastly, since Golomb-based coding only encodes non-negative integers, a “mapping” process is required when dealing with inputs that can be negative, ensuring all values are non-negative before encoding, formulated as

$$p' = \begin{cases} 2p & p \geq 0 \\ -2p - 1 & \text{otherwise} \end{cases} \quad (1)$$

Context-Based Extensions

To further enhance the coding efficiency of pixel based encoding, pixels can be classified into groups based on their contexts. For instance, if the encoder identifies a pixel on an edge, the prediction residual is likely having a larger value than that in a flat region. In this case, “on an edge” is the contexts of this pixel. For each context, the Huffman table or parameter M from previously encoded pixels is calculated and used to encode the next pixel in the context. If there are n contexts, n corresponding Huffman tables or parameter M s are recorded. An algorithm making full use of the multiple context is Context-based Adaptive Lossless Image Codec (CALIC) [41], which will be detailed in Section 4.7.

2.2.2. Bit-Based Encoding

Arithmetic Coding

The basic concept of arithmetic coding (AC) [42] involves encoding information within a range between 0 and 1, based on the probabilities of different inputs, while tracking both the lower and upper bounds of this range. For example, encoding the binary sequence “101” with a constant probability of “1” equal to 0.8 initializes the range to $[0, 1)$. The sequence modifies the range sequentially to $[0.2, 1)$, $[0.2, 0.38)$, and $[0.236, 0.38)$, as illustrated in Figure 5. In fixed-point representation within $[0, 1)$, “01” corresponds to 0.25, which falls within $[0.236, 0.38)$ demonstrated by the yellow area. Thus, the binary sequence “101” is encoded as “01” in AC.

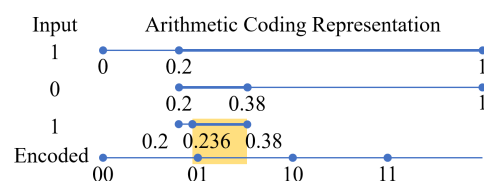


Figure 5. The arithmetic representation of binary sequence 101.

The explanations above imply that the encoder computes the lower and upper bounds with infinite precision, converting them to their final forms only at the end of the encoding process. However, rather than simulating infinite precision, most arithmetic coders operate at a fixed limit of precision, which they determine to be adequate for input sequences of limited length. For longer input sequences, renormalization is applied after encoding each input to prevent the finite precision from restricting the total number of encodable symbols. In practice, renormalization ensures that the interval $[l, u)$ does not shrink below the numerical resolution of the machine. In binary implementations, renormalization enlarges the interval by a factor of 2 whenever the range becomes smaller than half of the available precision. More concretely:

- If $l, u < 0.5$, then $l = l * 2$ and $u = u * 2$;
- If $l, u \geq 0.5$, then $l = l * 2 - 1$, $u = u * 2 - 1$;
- If $l \geq 0.25$ and $u < 0.75$, then $l = l * 2 - 0.5$, $u = u * 2 - 0.5$.

At each step, a binary digit is emitted when l and u lie entirely within the same half of the current range. If l and u straddle the midpoint, a special handling rule known is applied to delay the emission of bits until the interval can be safely normalized.

Range Coding

Range coding (RC) [43] closely resembles AC but initializes the range to $[0, R)$ instead of $[0, 1)$, with both u and l constrained to integers, where $R \in \mathbb{Z}^{0+}$ represents the maximum range. It is commonly believed that RC was developed to circumvent AC’s patent. Algorithms utilizing RC, such as [44–48], can be modified into AC with minimal efforts.

Asymmetric Numeral Systems

The concept of asymmetric numeral systems (ANSs) [49] is derived from symmetric numeral systems (SNSs), which include binary, decimal, and hexadecimal systems. In SNSs, an input pixel $p \in \mathbb{Z}^{0+}$ can be encoded from an existing number $n \in \mathbb{Z}^{0+}$ to $n' \in \mathbb{Z}^{0+}$ using the function $E_s(\cdot)$,

$$n' = E_s(n, p) = n \times b + p, \quad (2)$$

where $p \in [0, b-1]$ and $b \in \mathbb{Z}_{0+}$ is the base of the system. For example in a decimal system with $b = 10$, $n = 12$ and $p = 3$, then $n' = 12 \times 10 + 3 = 123$. Correspondingly, p can be decoded from n' by $D_s(\cdot)$ as

$$[n, p] = D_s(n') = [\lfloor n'/b \rfloor, \text{mod}(n', b)]. \quad (3)$$

The coding efficiency of SNSs is maximized when p is uniformly distributed in the range $[0, b-1]$. However, if p is not uniformly distributed, ANSs can be employed to improve the coding efficiency. The encoding function of ANSs is defined as follows:

$$n' = E_a(n, p, \omega, \psi) = N \times B + P, \quad (4)$$

where

$$\begin{aligned} N &= \lfloor n/\omega_p \rfloor, \\ B &= \psi_b, \\ P &= \psi_p + \text{mod}(n, \omega_p), \end{aligned} \quad (5)$$

$\omega \in \mathbb{Z}_{0+}^b$ and $\psi \in \mathbb{Z}_{0+}^{b+1}$ are the pre-defined tables containing the count of occurrence and the corresponding cumulative count of occurrence for p respectively. The terms ω_p and ψ_p represent the p^{th} coefficient in ω and ψ respectively. Assuming $b = 3$ and ω is $[5, 3, 4]$ ($\omega_0 = 5$, $\omega_1 = 3$, and $\omega_2 = 4$), then ψ would be $[0, 5, 8, 12]$.

For decoding, i.e., $[n, p] = D_a(n', \omega, \psi)$, it is straightforward to see the value of P can be calculated as follows,

$$P = \text{mod}(n', B), \quad (6)$$

and the input p can be decoded from P by

$$p = \arg \min_i (P - \psi_i, \forall P - \psi_i \geq 0). \quad (7)$$

Finally, n is decoded using,

$$n = \lfloor n'/B \rfloor \times \omega_p + P - \psi_p. \quad (8)$$

An ANS achieves a compression ratio comparable to that of AC while offering faster processing speeds. Additionally, it can be adaptive by varying ω and ψ . However, it is important to note that the decoded stream is in reverse order compared to the input, which may result in increased memory usage and make the ANS less convenient for applications in hyperspectral image lossless compression [50,51]. A summary of scanning and encoding orders reviewed in this section is given in Table 2.

Table 2. Summary of Scanning and Encoding Orders.

Method	Underlying Principle	Strengths	Limitations
Raster Scan	Sequentially processes pixels from left-to-right, top-to-bottom, band-by-band	Simple, widely used, low complexity	May not exploit local spatial correlation optimally
Hilbert Scan	Space-filling curve that improves locality of neighboring pixels	Enhances neighbor correlation, potentially better compression	Computationally more complex; sometimes less effective than Raster
Stripe-based Scan	Processes data in stripes, as used in JPEG2000	Balances complexity and efficiency, good for spatial correlation	Slightly more complex than Raster
Double Snake Scan	Alternating stripe scanning to minimize distance between successive pixels	Reduces prediction distance, improves correlation	More complex implementation
Block Scan	Divides image into blocks and applies scanning order within each block	Facilitates local processing, parallelizable	May introduce block boundary artifacts if not handled carefully
Run-Length Coding	Encodes consecutive repeating symbols as run-length pairs	Excellent for sparse/ highly repetitive data	Inefficient for highly variable data
Wavelet Coding (EZW, SPIHT, SPECK)	Encodes wavelet coefficients in hierarchical or sub-band order	Multi-resolution representation, progressive transmission possible, good compression ratio	Higher computational cost; needs careful bit-plane ordering for lossless mode
EZW	Encodes zerotrees of wavelet coefficients	Compact representation of insignificance	Requires multiple outputs for some cases
SPIHT	Improves EZW by separating parent/child significance	More efficient than EZW, fewer output symbols	3D extension increases complexity
SPECK	Groups coefficients by sub-band and encodes significance	Efficient sub-band exploitation, memory-efficient variants exist (ZM-SPECK)	May be less efficient when sub-band statistics vary significantly
Straight Coding	Directly encodes raw pixel values	Extremely simple, useful for initialization	No compression efficiency gain
Huffman Coding	Optimal prefix code based on symbol frequency	Simple, well-studied, fast decoding	Requires frequency table; adaptive mode incurs overhead
Golomb Coding	Prefix + suffix coding optimal for Laplacian distributions	Adaptive, near-optimal for prediction residuals	Requires parameter M tuning; mapping required for non-negative integers
Golomb-Rice Coding	Restricts M to powers of 2 for efficiency	Very fast, hardware-friendly, used in CCSDS-123	May be slightly less optimal than general Golomb coding
Exponential-Golomb Coding	Uses logarithmic prefix + binary suffix	Compact for small values, used in video coding	Slightly higher complexity than Rice
Context-based Extensions	Adjusts coding tables/ parameters per pixel context	Improves coding efficiency by modeling local statistics	Increases encoder complexity and memory usage
Arithmetic Coding	Encodes data into a fractional interval based on probabilities	Achieves near-entropy compression, highly efficient	Computationally expensive, requires renormalization
Range Coding	Integer-based version of AC	Similar compression to AC, avoids patent issues	Computationally expensive
Asymmetric Numeral Systems	Generalizes numeral systems for non-uniform distributions	Comparable ratio to AC, faster, vectorizable	Produces output in reverse order, higher memory usage

3. Transform Methods

Transform Methods refer to a class of lossless compression techniques that convert the original image data into a different domain through reversible mathematical operations. The primary goal of this transformation is to decorrelate the spatial and spectral information of HSI, thereby concentrating its energy into a smaller number of coefficients. These

transformed coefficients are typically more amenable to efficient entropy encoding, while ensuring perfect reconstructability through inverse transformation.

The most commonly used transformations for hyperspectral image lossless compression include the discrete cosine transform (DCT), Karhunen–Loeve transform (KLT), and discrete wavelet transform (DWT). Among these, DWT is the most widely adopted due to its superior performance and versatility. While DCT is more effective in lossy compression and can outperform DWT in such cases, it is less efficient for lossless compression [52]. KLT excels at spectral decorrelation but is computationally expensive and less effective for spatial decorrelation. In addition to reversible transformations, irreversible transformations with quantization and vector quantization can be applied to lossless compression as well; however, the quantization residuals must be recorded alongside the transformed coefficients. Finally, traditional 2D compression methods can be adapted for 3D hyperspectral images by making simple modifications to convert them into 2D representations.

3.1. Discrete Cosine Transform

Comparing among lossy compression methods, DCT is generally less favorable for lossless hyperspectral image compression, where it is typically applied either spectrally or in three dimensions. For instance, [52,53] utilize reversible one-dimensional DCT spectrally, while [54] first reduce brightness and contrast differences through luminance transformation, followed by applying three-dimensional DCT to the entire transformed image. However, it has been demonstrated in [53] that the discrete wavelet transform (DWT) with a 5/3 filter is more effective than DCT at capturing spectral correlation, suggesting that DCT may not be the optimal choice for lossless hyperspectral image compression.

3.2. Karhunen–Loeve Transform

In the context of lossless compression of discrete data, the terms KLT, principal component analysis (PCA), and principal components transform (PCT) [55] are often used interchangeably in the literature. Unlike DWT, which decorrelates images locally using a fixed set of filters, KLT generates global, data-dependent eigenvectors, leading to superior decorrelation and denser energy compaction. Besides KLT, a similar concept is Orthogonal Subspace Projection (OSP). By projecting the spectral data onto orthogonal subspaces, OSP effectively minimizes inconsistencies in reconstruction and enhances the independence among spectral channels [21].

With n spectral bands, KLT decomposes a spectrum into a weighted average of n spectra, necessitating the storage and transmission of n eigenvectors of length n , resulting in n^2 coefficients overhead. To minimize the compressed file size while ensuring reversibility, these coefficients are often stored as integer values, prompting research into integer-to-integer approximations of eigenvectors. However, constraining eigenvectors to integer values makes optimization NP-hard, which means that it is more practical to obtain only sub-optimal solutions [55–57].

KLT is particularly effective for decorrelating spectral information, based on the assumption that similar materials produce similar spectra. Notably, it has been shown that replacing DWT with KLT in the spectral direction for JPEG2000 can improve the compression ratio by approximately 5% [57]. Given that correlation in the spectral direction is generally much stronger than in the spatial directions [58], it is logical to focus KLT application in the spectral domain. For spatial two-dimensional transformations, either DWT [57,59,60] or binDCT [52] can be utilized. It is important to note that the sequences of transformations can also impact the compression ratio. In [60], one-dimensional KLT is performed before two-dimensional DWT; however, it has been found in [56,57] that

reversing this order (applying 2D DWT followed by 1D KLT) can yield a compression ratio improvement of around 10% [57].

3.3. Discrete Wavelet Transform

DWT is a widely used transformation in lossless compression and has been standardized in JPEG2000 for both lossy and lossless modes. While DWT is typically applied to one-dimensional and two-dimensional data, it can also be readily extended to three-dimensional images by adding an additional layer of decomposition along the spectral direction [61].

3.3.1. Wavelet Filters

Several adjustable parameters in DWT can significantly influence the overall compression ratio. Researchers in [62–64] conducted detailed analysis of the effects of various wavelet filters, including 5/3 [65], 2/6 [66], S+P transform version B (SPB) [67], S+P transform version C (SPC) [67], 9/7-M [68], (2, 4) [69], (6, 2) [69], 2/10 [70], 5/11-C [71], 5-11-A [72], 6/14 [73], 13/7-T [68], 13-T-C [73], and 9/7-F [74]. Different from the previous reviews which focused on forward wavelet transforms, this reviews display the coefficients for low pass and high pass filters to provide a different angle analysis. A direct observation from these filter coefficients is the number of vanishing moments. A filter with N vanishing moments annihilates polynomials of degree $N - 1$, thus a filter with a higher number of vanishing moments can represent signals more compactly.

The coefficients of the forward transforms for these filters are summarized in Table 3. Among them, SPB and SPC are based on finite impulse response (FIR) and infinite impulse response (IIR) filters, not all of which have linear phases. In contrast, the other 10 wavelets are based on linear-phase FIR filters. The properties of wavelet filters are further analyzed in [71], while their computational complexities, memory requirements, and performance in lossless image compression are detailed in [63].

Table 3. Coefficients of Forward Transforms for Different Wavelet Filters.

Names	Low Pass Filter Coefficients	High Pass Filter Coefficients
S	$\frac{1}{2}[1, 1]$	$[-1, 1]$
5/3 [65]	$\frac{1}{8}[-1, 2, 6, 2, -1]$	$\frac{1}{2}[-1, 2, -1]$
2/6 [66]	$\frac{1}{2}[1, 1]$	$\frac{1}{8}[1, 1, -8, 8, -1, -1]$
SPB [67]	$\frac{1}{2}[1, 1]$	$\frac{1}{16}[2, 2, -15, 17, -7, 1]$
SPC [67]	$\frac{1}{2}[1, 1]$	$\frac{1}{32}[-1, -1, 5, 5, -28, 36, -20, 4]$
9/7-M [68]	$\frac{1}{64}[1, 0, -8, 16, 46, 16, -8, 0, 1]$	$\frac{1}{16}[1, 0, -9, 16, -9, 0, 1]$
(2, 4) [69]	$\frac{1}{128}[3, -6, -16, 38, 90, 38, -16, -6, 3]$	$\frac{1}{2}[-1, 2, -1]$
(6, 2) [69]	$\frac{1}{1024}[3, 0, -22, 0, -175, 256, 724, 256, -175, 0, -22, 0, 3]$	$\frac{1}{256}[3, 0, -25, 0, -150, 256, -150, 0, -25, 0, 3]$
2/10 [70]	$\frac{1}{2}[1, 1]$	$\frac{1}{128}[-3, -3, 22, 22, -128, 128, -22, -22, 3, 3]$
5/11-C [71]	$\frac{1}{8}[-1, 2, 6, 2, -1]$	$\frac{1}{128}[-1, 2, 7, 0, -70, 124, -70, 0, 7, 2, -1]$
5/11-A [72]	$\frac{1}{8}[-1, 2, 6, 2, -1]$	$\frac{1}{256}[-1, 2, 7, 0, -134, 252, -134, 0, 7, 2, -1]$
6/14 [73]	$\frac{1}{16}[-1, 1, 8, 8, 1, -1]$	$\frac{1}{256}[1, -1, -14, -2, 47, 49, -244, 244, -49, -47, 2, 14, 1, -1]$
13/7-T [68]	$\frac{1}{512}[-1, 0, 18, -16, -63, 144, 348, 144, -63, -16, 18, 0, -1]$	$\frac{1}{16}[1, 0, -9, 16, -9, 0, 1]$
13/7-C [73]	$\frac{1}{256}[-1, 0, 14, -16, -31, 80, 164, 80, -31, -16, 14, 0, -1]$	$\frac{1}{16}[1, 0, -9, 16, -9, 0, 1]$
9/7-F [74]	$2^{-13}[-5075, 3200, -1958, 3200, -5075]$	$2^{-26}[4, 977, 763, -3, 138, 688, -3, 225, 2631, 60, 831, 488, -32, 252, 631, -3, 138, 688, 4, 977, 763]$

Notably, [63] found that no single transform consistently outperforms others across all types of images. For natural (smooth) images, the 5/11-C, 5/11-A, and 13/7-T filters demonstrate relatively better performance; in contrast, the 5/3 filter excels for images with more high-frequency content. Furthermore, the strengths of different wavelet filters appear to be generalizable, as evidenced by [63], which indicates that the performance of various filters tends to be consistent across most lossless coders. In addition to filter selection,

the number of decomposition levels also impacts the compression ratio. This parameter can be fixed at 3 [75] or 5 [76], or made user-definable as seen in JPEG2000.

3.3.2. Wavelet Packet Transform

In ordinary DWT, decomposition occurs only on the lowest-frequency sub-band. In contrast, the Wavelet Packet Transform (WPT) [77–79] decomposes every sub-band. As the distribution of coefficients within a sub-band tends to be consistent, WPT can decorrelate wavelet coefficients more effectively, making it particularly suitable for wavelet encoders. Experimental results indicate that WPT can improve the compression ratio by 3.20% and 3.57% for SPECK and SPIHT, respectively [78,79]. Notably, a three-dimensional WPT can be employed to process hyperspectral images as a whole [77], or it can focus solely on decorrelating spectral information, while the remaining spatial information is decorrelated using ordinary DWT [80].

3.3.3. Multiwavelet Transform

To address the limitations of DWT, which cannot simultaneously exhibit properties of compact support, orthogonality, symmetry, vanishing moments, and short support [81], the MultiWavelet Transform (MWT) [82,83] was proposed. MWT can be viewed as an iterative application of DWT using two or more pairs of filters, thus increasing the design flexibility of wavelet filters. This is expected to enhance image compression performance compared to ordinary DWT. For example, [82] reported an improvement in compression ratio by 10% over ordinary DWT using 9/7 and 5/3 filters.

3.3.4. Regression Wavelet Transform

The Regression Wavelet Transform (RWT) [84] represents another variation of DWT aimed at further reducing redundancy in high-frequency information. In RWT, low-frequency coefficients are encoded normally, while high-frequency coefficients are predicted from the corresponding low-frequency coefficients using a regression model, as these coefficients are significantly correlated. The process is conducted level by level until the coefficients with the highest frequency are predicted, with the regression model and prediction residuals recorded. Various models and variants, including the maximum model, restricted model, fast variant, and exogenous variant, were compared in [85,86], revealing that the fast-restricted RWT performed best. Additionally, the number of decomposition levels impacts performance: RWT with five levels outperforms those with one or eight levels. Experimental results indicate that RWT outperforms the Haar and 5/3 wavelet transforms by approximately 0.5 bit per pixel per channel (bpppc) [84].

3.3.5. Dyadic Wavelet Transform

Dyadic wavelet transform is a specialized form of the general wavelet transform in which scales and translations are restricted to dyadic (power-of-two) values. This approach produces the minimum number of coefficients required to represent an image. Although it can be applied to hyperspectral image lossless compression [87], it is generally more suitable for low bit-rate scenarios and tends to be less effective for strictly lossless compression.

3.4. JPEG2000-Based Methods

JPEG2000-based method is a special subset of discrete wavelet transform (DWT) methods. JPEG2000 [6] is a renowned compression algorithm designed for color images, notable for its high compression ratio at the cost of high computational complexity. It offers both lossless and lossy compression modes. In lossless mode, input images undergo preprocessing that includes tiling, DC shifting, and color transformation. Tiling divides images into non-overlapping rectangular blocks, DC shifting centers pixel values around

zero, and color transformation reversibly converts images from RGB to the modified YCbCr color space. Each tile's three channels of YCbCr are separately processed using 2D DWT with 5/3 filters. The transformed coefficients are then encoded through embedded block coding with Embedded Block Coding with Optimized Truncation (EBCOT), followed by context-based arithmetic coding to produce the compressed bitstreams.

While JPEG2000 can compress hyperspectral images with up to 16,385 bands, it was initially designed for monochromatic or RGB images. Efforts have been made to enhance its compatibility with hyperspectral data through spectral (inter-band) decorrelation and 3D extensions. JPEG2000 Part I [6] allows decorrelation only for RGB images, treating each hyperspectral band as an individual monochromatic image. JPEG2000 Part II [88] addresses this limitation by permitting arbitrary spectral decorrelation methods, including wavelet transforms [89]. Additionally, the JPEG group introduced "JPEG2000 Part 10—Volumetric Data Encoding" [90], which incorporates systematic modifications for 3D support in tiling, DWT, EBCOT, and arithmetic coding contexts. This extension allows user-defined tile sizes and wavelet decomposition levels across all three dimensions. Furthermore, for EBCOT and arithmetic coding, the number of neighboring bits used for context determination increases from 8 to 26. These adaptations enable JPEG 2000 to achieve up to 6.8% higher efficiency than CCSDS for noiseless images, but it is less effective for noisy images due to the transform-based nature [91]. Besides, the scanning order can also impact compression efficiency: [92] suggests scanning in a band-by-band order within a bitplane, as opposed to the traditional bitplane-by-bitplane approach, yielding an average improvement of 0.04 bpppc without any additional cost.

In addition to the official JPEG2000 extensions, other modifications have been proposed in the context of hyperspectral data. For instance, [75] explored various filters, including Haar, 2/6, CDF 9/7, 9/3, and 5/11-C—at different decomposition levels, concluding that a 5-level DWT with the CDF 9/7 filter offers the most effective compression setup.

3.5. Simple Modification from 2D Compression Methods

3.5.1. Spectral Decorrelation

Compared with hyperspectral images, the compression of monochromatic and RGB images has been extensively studied, resulting in numerous established lossless compression standards such as PNG [93] and JPEG2000 [6] and JPEG-LS [94]. Although these standards were originally designed for two-dimensional images, they can be adapted for hyperspectral data. A common and straightforward approach is to introduce a spectral decorrelation pre-processing step.

The predominant strategy applies one-dimensional transformations in the spectral domain, followed by JPEG2000 encoding. Typical spectral decorrelation techniques include DWT [53,75], RWT [86], KLT [95], and DCT [53]. Experimental studies [53] consistently show that DWT outperforms DCT for spectral decorrelation, with CDF 9/7 wavelet and five decomposition levels yielding particularly strong results [75,76].

In addition, JPEG-LS [94] compression can achieve more than a 100% improvement in compression ratio when the input is first spectrally decorrelated using a three-level DWT with the 9/7 filter [76]. Another method for extending JPEG-LS to hyperspectral data involves enhancing the MED predictor by incorporating information from preceding spectral bands [96,97].

3.5.2. 3D to 2D Image Conversion

It is noteworthy that other transforms specific to hyperspectral images do exist [98] proposes a method that converts a 3D image into a 2D representation, which transforms each band of a hyperspectral image into a 1D stripe and stacking these stripes into a 2D

image, as illustrated in Figure 6. This conversion effectively changes a 3D hyperspectral image of size $\mathbb{Z}^{H \times W \times L}$ into a 2D image of size $\mathbb{Z}^{HW \times L}$, thereby allowing the direct application of standard 2D image compression algorithms. The study in [98] shows that this conversion can yield a 10% to 20% improvement in compression ratios when applied to various algorithms, such as CALIC, JPEG-LS, and JPEG2000, compared to applying these algorithms independently to each band.

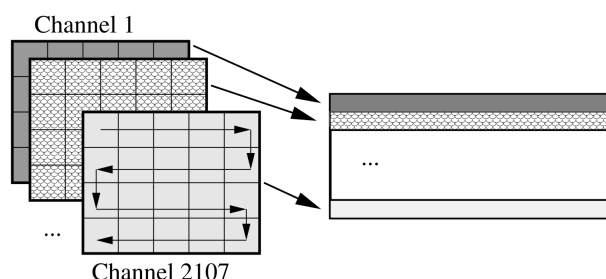


Figure 6. 3D -to-2D conversion presented in [98].

3.6. Irreversible Transforms with Residual Encoding

With irreversible transforms, the image is first decomposed using an approximation model, followed by the calculation of approximation residuals. This process requires encoding and storing three sets of information: the transformed coefficients, the approximation model, and the residuals. Viewed from another perspective, irreversible transform methods can be likened to cascading lossy compression techniques combined with residual encoding.

Both irreversible KLT and DCT could effectively concentrate most energy into a small number of coefficients. In the case of KLT, these filters are spectral-oriented eigenvectors with large eigenvalues, while for DCT, they represent spatially oriented decomposition filters concentrated in the low-frequency sub-bands. Following these transformations, filters with high energy content can be quantized, while those with low energy content can be omitted to reduce the size of the compressed file. Two representative algorithms of this type include [35,99]. In [99], lossy KLT is used to create a lossy compressed image, which is then transformed using a 3D Discrete Wavelet Transform (DWT), and the DWT coefficients are encoded. Conversely, ref. [35] applies a 3D DCT and subsequently encodes the transform residuals, calculated by subtracting the original image from the image reconstructed using inverse 3D DCT of the quantized coefficients. Experiments conducted in [35] explored various parameters and determined that a quantization interval of 64, along with partitioning the spectral data into groups of 20 bands for the spectral DCT, typically yields the best compression ratio. The results indicate that residual DCT can outperform Vector Quantization (VQ) by approximately 0.5 bpppc.

3.7. Vector Quantization

Vector quantization (VQ) [100] is a special subset of irreversible transform methods with residual encoding. It begins by partitioning the input image into non-overlapping blocks, each containing n pixels, which are then reshaped into n -dimensional vectors. VQ quantizes these vectors by dividing the vector space into several lower-dimensional subspaces, each represented by an index. These subspaces function as reference vectors, and the relationship between the vector space and reference vector indexes is stored in a codebook. Consequently, the n pixels in a block can be represented by a single index. However, as implied by the term “quantization,” not all information is captured by the index, necessitating the recording of quantization residuals for lossless compression. Therefore, VQ must document three types of information: the codebook, indexes, and quantization residuals.

The codebook generation method is primarily established through the Generalized Lloyd Algorithm (GLA) [101], a clustering algorithm that iteratively optimizes the centroid and index of each cluster until convergence. Ideally, all image vectors should be used as the training set, but this can be time-consuming. Research [102] indicates that a training set of 1000 randomly selected vectors is statistically sufficient, significantly reducing codebook generation complexity.

3.7.1. VQ Parameters

Several parameters influence VQ performance, including block shape, vector length, and the number of reference vectors. Blocks can be purely spectral [102], spatial [103], or three-dimensional. As the number of spectral bands increase, the spectral correlation becomes more pronounced, making spectral blocks increasingly popular. The vector length also significantly impacts compression ratio: longer vectors reduce address entropy but increase residual entropy, while shorter vectors do the opposite. A balance found in [102] sets the vector length to 16. Although [100] suggests a variable vector length based on the differing entropy of pixel values across bands, it simply uses an exhaustive search without a systematic method for optimization. An optimal reference vector number of 32 was found [102]. Similarly, this optimal number may not be suitable to all vectors and thus optimal bit allocation of reference vectors is proposed in [104]. This optimization is done with marginal analysis [105], which ensures a local optimal of bit allocation can be reached.

3.7.2. VQ Techniques

Besides the three main factors introduced above, several techniques can further affect compression ratios, such as forcing the DC component of all vectors to 0 by subtracting the mean [106]; normalizing vectors by dividing them with the mean [102] or the Euclidean norm [107] for the ease of quantization; post-processing the quantization residuals by recording the differences between current and previous bands instead [108]. Notably, normalization with mean yielded the most significant improvement in compression ratio [102]. Lastly, to mitigate codebook overhead, ref. [104] proposed using pre-computed VQ that does not need to be transmitted to the decoder, particularly effective for compressed images belonging to the same category, such as sounder data, which exhibit similar features across images.

3.8. Modification Add-Ons: Clustering

Clustering is a straightforward approach for grouping similar pixels, thereby increasing redundancy within each group. For instance, ref. [109] proposed improving KLT by dividing the spectral bands into distinct clusters, allowing KLT to be performed independently on each. This clustering strategy can marginally improve the compression ratio while significantly accelerating calculations. In a related direction, ref. [110] classifies spectral bands into high- and low-information categories and apply different compression strategies to them, based on an analysis of wavelet coefficients using Histogram of Oriented Gradients and K-means clustering. Alternatively, ref. [111] refines the RWT by clustering pixels based on their spectral information and applying different regression rules for each cluster, achieving an enhancement of 0.1 bpppc in compression. Beyond transformation methods, clustering can also be utilized in prediction methods, where pixels are spatially clustered using superpixel [112] or k-means [113] techniques. A summary of transform-based hyperspectral image compression methods reviewed in this section is given in Table 4.

Table 4. Summary of Transform-based Hyperspectral Image Compression Methods.

Method	Underlying Principle	Strengths	Limitations
KLT + DCT [52]	Performs 1D KLT along spectral dimension followed by 2D DCT in spatial domain	Compact energy representation; enables lossy-to-lossless compression	DCT is suboptimal for strictly lossless compression; eigenvector computation overhead from KLT
KLT + DWT [60]	Applies 1D spectral KLT followed by 2D DWT in spatial domain	Strong spectral decorrelation combined with multi-resolution spatial representation; typically better than pure DWT	High computational complexity for eigen-decomposition and multilevel wavelet transform
DWT [25]	Applies 3D DWT with improved SPECK	Highly effective for spatial and spectral decorrelation	Filter choice and decomposition levels should be tuned per dataset
Wavelet Packet Transform [80]	Decomposes not only the low-frequency sub-band but all sub-bands into further wavelet packets, allowing finer frequency partitioning	Provides more complete decorrelation; improves compression ratio for SPIHT/SPECK	Computationally expensive due to additional decompositions
Multiwavelet Transform [81]	Employs multiple pairs of scaling and wavelet functions simultaneously	Better compression than scalar DWT; flexible design space	Complex filter construction and implementation
Regression Wavelet Transform [85]	Predicts high-frequency wavelet coefficients using regression models based on low-frequency coefficients at each decomposition level	Significantly reduces redundancy in high-frequency sub-bands; improved coding efficiency over ordinary DWT	Requires storage of regression parameters and prediction residuals; model choice impacts performance
Dyadic Wavelet Transform [87]	Restricts scale and translation parameters to dyadic (power-of-two) values, minimizing number of coefficients required to represent signal	Extremely compact representation; low computational complexity; efficient at low bit-rates	Less effective for strictly lossless compression; limited adaptability
JPEG2000 (Part II: Extensions) [88]	Extends JPEG2000 to 3D images by allowing arbitrary spectral decorrelation	Internationally standardized; supports up to 16,385 spectral bands; progressive bitplane coding; excellent compression efficiency for noiseless data	High computational complexity; transform-based approach is sensitive to noise and can propagate errors
DWT + JPEG-LS [76]	Applies 1D spectral DWT followed by 2D JPEG-LS in spatial domain	Simple to implement; computationally efficient; improved coding efficiency over ordinary JPEG-LS	Compression performance lower than full 3D transform approaches; sensitive to filter and decomposition level selection
RWT + JPEG2000 [86]	Applies 1D spectral RWT followed by 2D JPEG2000 in spatial domain	Competitive compression results with reduced complexity compared to KLT-based approaches; benefits from regression decorrelation	Less efficient than KLT+DWT combination; residual coding overhead still present
3D-to-2D + JPEG2000 [98]	Rearranges hyperspectral cube into 2D strip image before standard JPEG2000 coding	Allows direct application of mature 2D coders; improved coding efficiency over per-band compression	Ignores intrinsic 3D correlation structure; performance depends on band ordering; less effective for highly nonlinear spectral correlations
DCT + Residual Encoding [35]	Applies lossy 3D DCT to obtain compact representation, then encodes residuals for perfect reconstruction	High energy compaction; tunable trade-off between compression ratio and reconstruction fidelity; outperforms vector quantization in some settings	Requires transmitting DCT coefficients and residuals; careful quantization design critical to avoid file-size overhead
Vector Quantization [100]	Divides data into vectors, quantizes them by mapping to nearest codebook entries, and stores indexes plus residuals	Very effective for highly correlated data; parameters can be tuned for target performance	Codebook generation can be computationally expensive; overhead for transmitting codebook; sensitive to training set quality

4. Prediction Methods

Unlike transform methods which transform a block of pixels globally, prediction methods focus on compressing individual pixels and encoding the prediction residuals. This type of methods is also called differential pulse code modulation (DPCM). Upon encoding

a pixel, local information is inserted. This information can be extracted from the current band or previous bands, forming spatial or spectral predictors. Under prediction methods, a large number of predictors are proposed in the literature, which can be categorized as spatial, differential, linear, hybrid and Look Up Table (LUT) predictors. Furthermore, two popular standards including CCSDS and CALIC, one add-on modification named band reordering are introduced.

4.1. Lookup Table

Although the term LUT [44] is used, this method does not require storing a LUT as a codebook, making it fundamentally different from VQ. Instead, the LUT is generated and updated dynamically during encoding, allowing it to be reproduced during decoding without the need for any side information. The principle behind the LUT is that if two pixels in band $z - 1$ have the same value, they are likely to have the same value in band z . To predict $\hat{\mathbf{I}}_z(x, y)$, we search in band $z - 1$ for a pixel at position $(x', y', z' - 1)$ with the same value as $\mathbf{I}_{z-1}(x, y)$, i.e., $\mathbf{I}_{z-1}(x, y) = \mathbf{I}_{z-1}(x', y')$, then set $\hat{\mathbf{I}}_z(x, y)$ to $\mathbf{I}_z(x', y')$. If no such pixel is found in band $z - 1$, the pixel is considered an outlier, and $\hat{\mathbf{I}}_z(x, y)$ is set to $\mathbf{I}_{z-1}(x, y)$. If two or more pixels in band $z - 1$ share the same value as $\mathbf{I}_{z-1}(x, y)$, the pixel closest to (x, y) is selected. However, the calculation of distances between every two pixels is resource intensive. To simplify this, when multiple pixels in band $z - 1$ match $\mathbf{I}_{z-1}(x, y)$, the most recently encoded pixel is used to predict $\hat{\mathbf{I}}_z(x, y)$.

This principle is practically implemented as follows, which allows the LUT to be generated and updated during encoding, avoiding the need for exhaustive search. For each band, a new LUT is initialized. While encoding band z , whenever a new value $\mathbf{I}_{z-1}(x', y')$ in band $z - 1$ is encountered, it is recorded as the index of the LUT, and $\mathbf{I}_z(x', y')$ is recorded as the corresponding value. If $\mathbf{I}_{z-1}(x', y')$ already exists as an index, its corresponding value is updated to $\mathbf{I}_z(x', y')$. Consequently, when encoding $\hat{\mathbf{I}}_z(x, y)$, if index $\mathbf{I}_{z-1}(x, y)$ exists in the LUT, $\hat{\mathbf{I}}_z(x, y)$ is predicted as the corresponding value; otherwise, it is predicted as $\mathbf{I}_{z-1}(x, y)$.

4.1.1. Locally Averaged Interband Scaling

Various modifications have been made to this LUT method. For example, instead of predicting $\hat{\mathbf{I}}_z(x, y)$ as $\mathbf{I}_{z-1}(x, y)$ for outliers, ref. [114] proposed a method called Locally Averaged Interband Scaling (LAIS), which calculates the average scaling factor between bands z and $z - 1$ around the position (x, y) . LAIS then predicts $\hat{\mathbf{I}}_z(x, y)$ as the product of $\mathbf{I}_{z-1}(x, y)$ and the scaling factor, thereby preserving the local interband scaling. However, ref. [115] later found that LAIS does not always improve the prediction of outliers. If LAIS is closer to 1, the conventional outlier prediction ($\hat{\mathbf{I}}_z(x, y) = \mathbf{I}_{z-1}(x, y)$) should be used instead.

Moreover, instead of storing a single pixel value in the LUT, ref. [114] proposed storing up to N values for each index, creating N prediction candidates. The candidate whose value is closest to the LAIS-predicted value is selected as the final prediction [116]. Simultaneously, refs. [117,118] introduced the idea of generating M LUTs for the previous M bands, resulting in $N \times M$ prediction candidates per index. Similarly, the candidate closest to the LAIS prediction is chosen as the final prediction. According to [117], the benefits of increasing N and M diminish quickly as they grow larger, with optimal values of $N = 20$ and $M = 4$ balancing computational cost and compression efficiency. Further research by [115] introduced three additional LAIS variations, demonstrating that the “Quotient of Local Sums” outperforms standard LAIS in most scenarios.

4.1.2. LUT with Outliers

With the pixel bit depth increases from 8 to 16 bits for modern hyperspectral images, the LUT becomes sparser, causing more pixels to be classified as outliers. Since the prediction accuracy for outliers is generally lower than for those stored in the LUT, this sparsity significantly reduces overall prediction accuracy. To address this issue, ref. [119] proposed quantizing $I_{z-1}(x, y)$ before searching or inputting values into the LUT. They recommended using a uniform quantization factor of 10 for applications requiring lower computational complexity or optimizing the quantization factor through exhaustive search for scenarios demanding higher compression ratios.

4.1.3. Prediction Residuals of LUT

While the LUT-based method can accurately predict $\hat{I}_z(x, y)$, the resulting prediction residuals are not well-suited for entropy encoding. As shown in Figure 7, the distribution of prediction residuals for both LUT and LAIS-LUT fluctuates at an irregular frequency, whereas FL (Fast Lossless, a linear prediction method) produces residuals with a smoother distribution. Most entropy encoders are less effective at encoding residuals with such spiky distributions [120], which still remains a challenge.

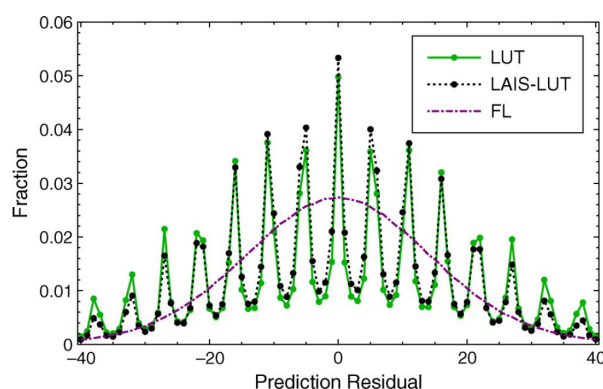


Figure 7. The distribution of prediction residual for LUT-based and linear prediction methods [120].

4.2. Spatial Predictors

Spatial predictors were initially designed for monochromatic and RGB images but can also be applied to hyperspectral images, either directly or with modifications. These predictors rely solely on the spatial correlation between pixels, classifying each pixel into different groups based on the values of its neighboring pixels and then applying specific prediction rules. Two prominent spatial predictors are the Median Edge Detector (MED) [121] and the Gradient Adjusted Predictor (GAP) [41].

4.2.1. Median Edge Detector

MED [121] is the core of the Low Complexity Lossless Compression for Images (LOCO-I) algorithm, a well-known 2D image compressor recognized for its low complexity. LOCO-I has been adopted by the international JPEG-LS standard [122]. MED utilizes five pixels surrounding (x, y) to predict $\hat{p}$, as shown in Figure 8. The underlying idea of MED is straightforward: if a vertical line exists to the left of $\hat{p}$, it assumes that $\hat{p}$ is also on the same vertical line and predicts $\hat{p} = a$; similarly, if a horizontal line exists above $\hat{p}$, then $\hat{p} = b$. If neither condition is met, $\hat{p}$ is assumed to lie in a flat region, and $a + b - c$ is used to predict $\hat{p}$. Since LOCO-I focuses on low complexity, MED employs a simple decision rule for distinguishing edge pixels from flat-region pixels: $\hat{p}$ is predicted as being in a flat region if c lies between a and b , and the predictor uses $a + b - c$, based on the assumption that $p - \frac{1}{2}(a + b)$ equals $\frac{1}{2}(a + b) - c$. Otherwise, the pixel is predicted to be

on a vertical edge if $\text{abs}(c - a) > \text{abs}(c - b)$, resulting in $\hat{p} = b$, or on a horizontal edge if $\text{abs}(c - a) < \text{abs}(c - b)$, resulting in $\hat{p} = a$. These prediction rules are summarized as

	c	a	d
e	b	$\hat{p}$	

Figure 8. A causal template for MED.

$$\hat{p} = \begin{cases} a + b - c & \text{if } a \leq c \leq b \text{ or } b \leq c \leq a \\ b & \text{elseif } |c - a| \geq |c - b| \\ a & \text{elseif } |c - a| < |c - b| \end{cases}, \quad (9)$$

and simplified to (10). It is worth noting that pixels d and e are not involved in the prediction phase but are used later in the context-based entropy encoding stage.

$$\hat{I}_{x,y} = \begin{cases} \min(a, b) & \text{if } c > \max(a, b) \\ \max(a, b) & \text{elseif } c < \min(a, b) \\ a + b - c & \text{otherwise} \end{cases}. \quad (10)$$

To make MED more appropriate to HSIs, a new fusion rule [96] is introduced in on top of (10) that analyzes local consistency, resulting in a 24% improvement in compression ratio. If both the absolute differences $|a - a_{z-1}|$ and $|b - b_{z-1}|$ are less than a threshold T , then $\hat{p}$ is predicted as $p_{z-1} + a - a_{z-1}$,

$$\hat{p} = \begin{cases} p_{z-1} + a - a_{z-1} & \text{if } |a - a_{z-1}| < T \text{ And } |b - b_{z-1}| < T \\ \min(a, b) & \text{elseif } c > \max(a, b) \\ \max(a, b) & \text{elseif } c < \min(a, b) \\ a + b - c & \text{otherwise} \end{cases}. \quad (11)$$

Alternatively, refs. [37,123] categorize the neighbors of p into four spatial groups (north, west, north-west, and north-east directions) and one spectral group. This categorization yields four types of neighborhood local sums and nine types of local differences, which are used to select the prediction mode and subsequently predict p . This method is referred to as simple lossless algorithm (SLA) by the author [124]. The neighborhood pixels can also be rearranged in band-interleaved-by-line [14] or band interleaved by pixel [15] orders to enhance its compatibility.

4.2.2. Gradient Adjusted Predictor

The Gradient Adjusted Predictor (GAP) forms the basis of the CALIC algorithm [41,125]. Like MED, GAP categorizes pixels based on the direction and magnitude of surrounding edges and applies different prediction rules accordingly. However, GAP provides a finer classification, distinguishing between weak, normal, and sharp horizontal and vertical edges, as well as smooth regions. GAP proceeds in two main steps. First, the horizontal gradient δ_x and vertical gradient δ_y are estimated by

$$\delta_x = |a - e| + |b - c| + |d - b|, \quad (12)$$

$$\delta_y = |a - c| + |b - f| + |c - g|. \quad (13)$$

Next, different prediction rules are applied based on the relative magnitudes of δ_x and δ_y . If δ_x exceeds δ_y by more than a threshold T_1 , $\hat{p}$ is predicted to lie on a strong vertical edge, with $\hat{p} = a$. Conversely, if δ_x is smaller than δ_y by T_1 , a strong horizontal edge is assumed, and $\hat{p} = b$. If the difference between δ_x and δ_y is less than T_3 , the pixel is classified as being in a flat region, with $\hat{p} = \frac{a+b}{2} + \frac{d-c}{4}$. In other cases, $\hat{p}$ is predicted as the weighted

average of a , b , and $\frac{a+b}{2} + \frac{d-c}{4}$, depending on the direction and magnitude of the edges. The full set of prediction rules is detailed in (14), where the thresholds $T_1 > T_2 > T_3 > 0$ are typically set to 80, 32, and 8, respectively.

$$\hat{p} = \begin{cases} a & \text{if } \delta_x - \delta_y > T_1 \\ b & \text{elseif } \delta_x - \delta_y < -T_1 \\ \frac{1}{2} \left(\frac{a+b}{2} + \frac{d-c}{4} \right) + \frac{1}{2}a & \text{elseif } \delta_x - \delta_y > T_2 \\ \frac{1}{2} \left(\frac{a+b}{2} + \frac{d-c}{4} \right) + \frac{1}{2}b & \text{elseif } \delta_x - \delta_y < -T_2 \\ \frac{3}{4} \left(\frac{a+b}{2} + \frac{d-c}{4} \right) + \frac{1}{4}a & \text{elseif } \delta_x - \delta_y > T_3 \\ \frac{3}{4} \left(\frac{a+b}{2} + \frac{d-c}{4} \right) + \frac{1}{4}b & \text{elseif } \delta_x - \delta_y < -T_3 \\ \frac{a+b}{2} + \frac{d-c}{4} & \text{otherwise} \end{cases} \quad (14)$$

4.3. Differential Predictor

Differential Predictor (DP), or differential encoding [91], might be the most fundamental and straightforward prediction methods. It assumes that pixel values between neighboring pixels are correlated, and thus predicts the current pixel value based on its neighbors. To minimize encoding complexity and memory cost, the neighbor selected for monochromatic or RGB image compression is typically the last encoded pixel, i.e., $\hat{p} = a$ or $\hat{p} = b$. Besides calculating the difference, a similar concept can be achieved by employing a bitwise exclusive or (XOR) operation [126]. While the exact origin of DP is unclear, it has been widely applied in computationally efficient methods such as DC component prediction in the JPEG international standard [127], which exploits spatial correlation. For hyperspectral images, which exhibit strong spectral correlation, DP is usually used as a spectral predictor, i.e., $\hat{p} = p_{z-1}$. In [128], the predictor is further refined by dividing each spectral band into non-overlapping blocks, which are then predicted independently. In this case, $\hat{p}$ is assumed to be linearly related to p_{z-1} ,

$$\hat{p} = \beta \cdot p_{z-1}, \quad (15)$$

with a constant scaling factor β for all pixels in a block.

4.3.1. Scaling Factor of DP

Since β cannot be generated by the decoder, it must be transmitted and stored with the prediction residual. Although smaller block sizes yield more accurate values of β , they also increase overhead. Trial experiments in [128] determined a block size of 16×16 to be optimal.

To reduce the storage overhead, β can be predicted during encoding using surrounding information. This is achieved by assuming that β at position (x, y) is similar to the values at $(x-1, y)$, $(x, y-1)$, and other neighboring positions. Thus, β is computed by solving a linear regression problem,

$$\min_{\beta} \left\| \beta \begin{bmatrix} a_{z-1} \\ b_{z-1} \\ c_{z-1} \\ \vdots \end{bmatrix} - \begin{bmatrix} a \\ b \\ c \\ \vdots \end{bmatrix} \right\|_2^2. \quad (16)$$

To further enhance accuracy, a bias term α is introduced [129], reformulating (15) to

$$\hat{p} = \beta \cdot p_{z-1} + \alpha, \quad (17)$$

and the regression problem becomes,

$$\min_{\beta, \alpha} \left\| \beta \begin{bmatrix} a_{z-1} \\ b_{z-1} \\ c_{z-1} \\ \dots \end{bmatrix} + \alpha - \begin{bmatrix} a \\ b \\ c \\ \dots \end{bmatrix} \right\|_2^2 = \min_{\alpha, \beta} \|\beta \mathbf{B} + \alpha - \mathbf{A}\|_2^2. \quad (18)$$

which is solvable by

$$\beta = \frac{N \sum_{n=1}^N \mathbf{A}_n \mathbf{B}_n - \sum_{n=1}^N \mathbf{A}_n \sum_{n=1}^N \mathbf{B}_n}{N \sum_{n=1}^N \mathbf{B}_n^2 - \left(\sum_{n=1}^N \mathbf{B}_n \right)^2}, \quad (19)$$

$$\alpha = \frac{1}{N} \sum_{n=1}^N \mathbf{A}_n - \beta \frac{1}{N} \sum_{n=1}^N \mathbf{B}_n. \quad (20)$$

where N is the number of neighboring pixels.

4.3.2. Higher Order DP

Other ways to improve prediction accuracy is by either increasing the context window [130], or prediction order. In [31,131]. A third-order predictor is proposed, formulated as

$$\hat{p} = p_{z-1} + \beta_1(p_{z-1} - p_{z-2}) + \beta_2(p_{z-3} - p_{z-4}) + \beta_3(p_{z-3} - p_{z-4}), \quad (21)$$

where β_1 , β_2 and β_3 are derived by solving the Wiener-Hopf equation,

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \sigma_{01} \\ \sigma_{02} \\ \sigma_{03} \end{bmatrix}. \quad (22)$$

Here, σ_{ij} represents a quantity similar to the covariance of neighboring pixels around (x, y) between band i and band j (where σ_{02} represents the covariance between band z and band $z - 2$).

In addition, spatial information can be integrated to DP by incorporating additional terms into the prediction rules. For instance, instead of simply predicting $\hat{p}$ as p_{z-1} , information from neighboring pixels can be included to enhance prediction accuracy [132] as

$$\hat{p} = p_{z-1} + \frac{1}{2}(a - a_{z-1}) + \frac{1}{2}(b - b_{z-1}). \quad (23)$$

4.4. Linear Predictor

Linear prediction, also known as Fast Lossless (FL) [133], encompasses a group of methods that predict the current pixel $\hat{p}$ based on a weighted average of pixels at the same (x, y) position across the previous P spectral bands. The prediction order is typically fixed to one, as inter-band correlations are generally linear. Experimental results indicate that adding second-order terms to the prediction model does not justify the added complexity and overhead [134,135]. Linear prediction is formulated as

$$\hat{p} = \mathbf{W}^T \mathbf{V}, \quad (24)$$

where

$$\mathbf{V} = \begin{bmatrix} 1 \\ p_{z-1} \\ p_{z-2} \\ \dots \\ p_{z-P} \end{bmatrix}, \quad (25)$$

and $\mathbf{W}$ is the weight vector, having the same dimension as $\mathbf{V}$. The first element in $\mathbf{V}$ represents a DC shift, which can be omitted.

Other than predictions based on p_{z-1} to p_{z-P} , pixels from the current band can also be incorporated into the vector $\mathbf{V}$. For example, [136,137] allows the inclusion of a , b , and c in the prediction, i.e.,

$$\mathbf{V} = \begin{bmatrix} a \\ b \\ c \\ p_{z-1} \\ \dots \\ p_{z-P} \end{bmatrix}. \quad (26)$$

4.4.1. Weight Vector of LP

There are multiple ways to compute $\mathbf{W}$. One approach, as described in [47,138], involves classifying all spectral data into clusters. This clustering can be performed using k-means or soft k-means [139], by minimizing the standard deviation of pixel values within each cluster, or by minimizing the variance of KLT coefficients for all pixels in a cluster [134]. Within each cluster, $\mathbf{W}$ is optimized by minimizing the squared prediction error for each band. However, this method requires both the cluster index and all $\mathbf{W}$ values to be stored, leading to significant overhead. To address this, [140] proposes dividing each band into 16×16 non-overlapping blocks (or 8×8 blocks [141]) and assigning a single $\mathbf{W}$ to all pixels within each block, theoretically reducing the overhead by a factor of $\frac{255}{256}$.

To eliminate the overhead associated with storing the coefficients of $\mathbf{W}$, several adaptive methods have been proposed that predict $\mathbf{W}$ using previously encoded information. For example, ref. [46] suggests encoding and decoding a small number of representative spectral data points from the image, then optimizing $\mathbf{W}$ based on these data points. Alternatively, ref. [142] proposes predicting $\mathbf{W}$ using spectral data from neighboring pixels, specifically those located to the left, top-left, top, and top-right of the current pixel. However, this method limits the number of coefficients in $\mathbf{W}$ to four, meaning P is capped at three. Including more spectral data would allow the optimization of $\mathbf{W}$ with additional coefficients, but this significantly increases computational complexity. To address this issue, an update scheme for $\mathbf{W}$ is employed, where the next $\mathbf{W}$ is updated from the current $\mathbf{W}$ by minimizing the square error recursively, which is also named least mean squares (LMS) [143]. It can also be solved with recursive least squares (RLS) [144] at the cost of high complexity. Similar to other update schemes, $\mathbf{W}$ is first initialized and then updated to compensate for prediction errors, with further details discussed in Section 4.6, based on the international CCSDS standard. Furthermore, the forgetting factor in RLS plays a crucial role in prediction. In the standard RLS, this factor is typically considered fixed. Ref. [145] takes an innovative step by proposing a variable forgetting factor, which significantly improves prediction accuracy.

4.4.2. Length of Weight Vector

In addition to the estimation methods for $\mathbf{W}$, the choice of prediction length P can significantly impact the compression ratio. The value of P can be fixed at 2 [136], 3 [137],

4 [146], or include all previous bands [147]. In some cases, P is selected from values between 10 and 200 in steps of 10, through an exhaustive search to achieve the best compression ratio [48]. Moreover, the selection of bands used to predict the current band can also be optimized [148] proposes calculating the spatial correlation between the current band and all previous bands, then choosing the best P bands for prediction. If the correlations between the current band and all previous bands exceed a threshold, the pixel values are encoded directly (i.e., $\hat{p}$ is set to 0) instead of encoding the prediction residual.

4.5. Hybrid Methods

4.5.1. Pre-Processing with Spectral Decorrelation

Spatial and spectral information can be decorrelated by pre-processing hyperspectral images. For spatial predictors, spectral decompositions can be performed prior to prediction. For instance, refs. [149,150] utilize a 5-level 2D discrete wavelet transform (DWT) with a 5/3 filter to decompose each spectral band, followed by DP and AC, achieving a 10% improvement in compression ratio over CALIC. Additionally, ref. [97] proposed predicting the difference between $\hat{p}$ and p_{z-1} . Specifically, ref. [97] applies MED to the difference image δ_z , which represents the difference in pixel values between the current and previous bands at position a : $\delta_z(a) = a - a_{z-1}$. This modification enhances the compression ratio by 81%.

$$\hat{p} = \begin{cases} p_{z-1} + \min(\delta_z(a), \delta_z(b)) & \text{if } \delta_z(c) > \max(\delta_z(a), \delta_z(b)) \\ p_{z-1} + \max(\delta_z(a), \delta_z(b)) & \text{elseif } \delta_z(c) < \min(\delta_z(a), \delta_z(b)) \\ p_{z-1} + \delta_z(a) + \delta_z(b) - \delta_z(c) & \text{otherwise} \end{cases} \quad (27)$$

4.5.2. Pre-Processing with Spatial Decorrelation

Similarly, spatial decorrelation can be applied before using spectral predictors. The CCSDS [39] can be viewed as a modified linear predictor that operates on a difference image δ , where each pixel is calculated by subtracting the current pixel value p from the average of four surrounding pixels, given by

$$\delta_z(p) = p - \frac{1}{4}(a + b + c + d). \quad (28)$$

Afterward, the prediction shifts from $\hat{p}$ to $\hat{\delta}_z(p)$. The detailed process for calculating $\hat{\delta}_z(p)$ is presented in Section 4.6. Once $\hat{\delta}_z(p)$ is determined, the predicted pixel value $\hat{p}$ can be obtained as follows,

$$\hat{p} = \hat{\delta}_z(p) + \frac{1}{4}(a + b + c + d). \quad (29)$$

Besides (28), the local-mean-removed pixel values can be formulated as $\delta_z(p) = p - \frac{1}{3}(a + b + c)$ in [151]. Alternatively, $\bar{p}$ can be derived by averaging τ previously encoded pixels, as described in [112].

4.5.3. Cascading Different Predictors

Besides pre-processings, [97] utilizes MED predictor on the difference image δ to generate a residual image, which is further predicted spatially using $\hat{p}_{residual} = \frac{1}{2}(a_{residual} + b_{residual})$. The residuals of residual image are entropy encoded to produce the output bitstream [131] takes a step further by implementing a hybrid method in the first stage, using both a median predictor and a linear predictor selected based on the band index, with predicted pixel values refined through a modified LUT method. The residuals of refined predictions are subsequently encoded.

It is noteworthy the number of cascaded operations for prediction are not limited to two, as demonstrated by [152], which is a modified version of [131]. Ref. [152] performs

hybrid prediction identical to [131], refining the results with LUT methods that incorporate multiple candidates and prediction bands, followed by another refinement with a modified LUT method. Experimental results indicate that [152] could outperform [131] by 1.5%. Similarly, [113] employs a three-level cascaded predictor, consisting of a similar neighborhood mean predictor to eliminate spatial redundancy, a low-order RLS predictor to address spectral redundancy, and a high-order LMS predictor to obtain the final predicted value. However, due to their complexity and ineffectiveness, multi-stage algorithms are not preferred in the literature.

4.5.4. Cascading Bias Cancellation to Predictors

Another technique for processing residual images is bias cancellation (BC). After prediction, the residuals undergo BC first, followed by the encoding of the compensated residual. Assuming the residuals are Laplacian distributed, shifting the mean of the residuals to zero reduces their overall magnitude, which ultimately shortens the bitstream generated by the entropy encoder.

There are various forms of BC, with two typical examples adopted by CALIC [41] and LOCO-I [121]. Both methods generate the compensated pixel value by adding a bias $\epsilon(n_c)$ to the predicted pixel value $\hat{p}$, i.e.,

$$\tilde{p} = \hat{p} + \epsilon(n_c). \quad (30)$$

Here, ϵ is a vector whose size corresponds to the number of contexts defined by the encoder, $\epsilon(n_c)$ refers to the n_c^{th} bias in ϵ , and n_c is the index of the context to which the current pixel p belongs.

Among these, CALIC tracks the prediction error between the predicted pixel value $\hat{p}$ and the actual pixel value p , setting the bias to the mean error, i.e.,

$$\epsilon(n_c) = \text{mean}(p - \hat{p}, \forall p \in \text{context } n_c). \quad (31)$$

In contrast, LOCO-I utilizes the residual between the compensated pixel value $\tilde{p}$ and the actual pixel value. Whenever the residual after bias compensation averages 0.5 or more (or -0.5 or less), the bias is incremented (or decremented) by 0.5,

$$\epsilon(n_c) = \lfloor \text{mean}(p - \tilde{p}, \forall p \in \text{context } n_c) \times 2 \rfloor / 2. \quad (32)$$

4.5.5. Band Adaptive Selection

Various studies indicate that no single predictor can accurately estimate $\hat{p}$ across all frequency bands. As shown in [153], Figure 9 compares the entropies of spatial and spectral prediction residuals and unprocessed pixel values for each band in the Jasper Scene 3 image. While the spectral predictor yields the lowest average entropy, the spatial predictor or raw pixels perform better in many bands. Consequently, using different predictors for different bands has been proposed to improve compression.

The simplest predictor assignment relies on fixed rules. For instance, refs. [34,125,154] apply spatial predictors to the first band and spectral predictors to the rest. Similarly, refs. [136,153] note that images in the same dataset have similar spectral responses, enabling a single rule for all images. For AVIRIS, spatial predictors are used for bands 1–4, 107–114, 154–166, and 221–224, with spectral predictors for the others. A more accurate but costly method encodes each band with both predictors, selects the one with the least mean absolute error, and stores its choice as 1-bit side information per band.

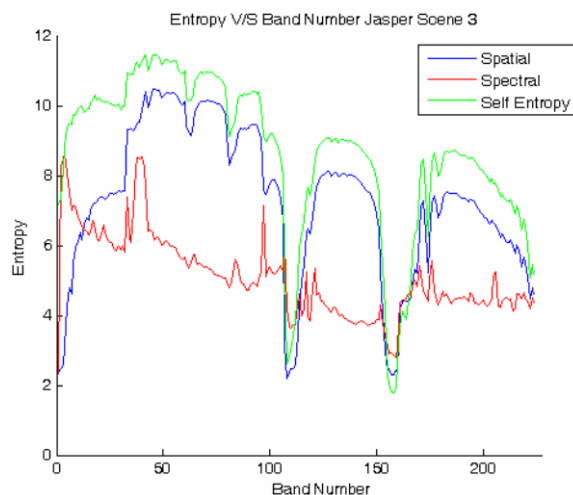


Figure 9. Entropy plots for different predictors in different bands [153].

In addition to this simple approach, adaptive spectral band grouping can also be employed [125] uses the spectral correlation factor $C_z(z_1, z_2)$ to estimate the correlation between bands z_1 and z_2 , as

$$C_z(z_1, z_2) = \frac{\sum_{x=1}^X \sum_{y=1}^Y (\mathbf{I}_{z_1}(x, y) - \bar{I}_{z_1})(\mathbf{I}_{z_2}(x, y) - \bar{I}_{z_2})}{\sqrt{\sum_{x=1}^X \sum_{y=1}^Y (\mathbf{I}_{z_1}(x, y) - \bar{I}_{z_1})^2 \sum_{x=1}^X \sum_{y=1}^Y (\mathbf{I}_{z_2}(x, y) - \bar{I}_{z_2})^2}}, \quad (33)$$

alongside the spatial correlation factor $C_{xy}(z)$,

$$C_{xy}(z) = \frac{\sum_{x=1}^{X-1} \sum_{y=1}^{Y-1} (\mathbf{I}_z(x, y) - \bar{I}_z)(\mathbf{I}_z(x+1, y+1) - \bar{I}_z)}{\sum_{x=1}^{X-1} \sum_{y=1}^{Y-1} (\mathbf{I}_z(x, y) - \bar{I}_z)^2}. \quad (34)$$

When $C_z(z, z_2)$ (typically $z_2 = z - 1$) exceeds $C_{xy}(z)$, a spectral predictor is used; otherwise, a spatial one is applied. To simplify, [155] computes only C_z and selects the spectral predictor if it exceeds 0.9. This band-level choice is stored as binary side information. Alternatively, [32,33] avoid computing C_z by predicting the first row or column with both predictors, encoding the residuals, and choosing the predictor with the better compression ratio for the rest of the band.

4.5.6. Block Adaptive Selection

This method divides each band into non-overlapping blocks [146] or clusters [18], followed by applying neighbor-driven rules based on pixel position. Ref. [156] improves it by computing spectral correlations to enable more precise predictor selection. Predictor choices can be sent as side information, inferred from an arbitrary pixel's error, or approximated using neighboring blocks. Finer granularity can be achieved by assigning predictors per pixel based on pixel-wise correlation C_p with neighbors [129],

$$C_p(z_1, z_2) = \frac{|\mathbf{S}| \sum_{i=1}^{|\mathbf{S}|} \mathbf{s}_{z_1}(s) \mathbf{s}_{z_2}(s) - \sum_{s=1}^{|\mathbf{S}|} \mathbf{s}_{z_1}(s) \sum_{s=1}^{|\mathbf{S}|} \mathbf{s}_{z_2}(s)}{\sqrt{\left(|\mathbf{S}| \sum_{i=1}^{|\mathbf{S}|} \mathbf{s}_{z_1}(s)^2 - \left(\sum_{i=1}^{|\mathbf{S}|} \mathbf{s}_{z_1}(s) \right)^2 \right) \times \left(|\mathbf{S}| \sum_{i=1}^{|\mathbf{S}|} \mathbf{s}_{z_2}(s)^2 - \left(\sum_{i=1}^{|\mathbf{S}|} \mathbf{s}_{z_2}(s) \right)^2 \right)}}. \quad (35)$$

Other variations of correlation coefficients do exist, including the example in [157], where the variance of $\mathbf{S}_{z_2}$ is used in the denominator.

$$C_p(z_1, z_2) = \frac{|S| \sum_{i=1}^{|S|} \mathbf{S}_{z_1}(s) \mathbf{S}_{z_2}(s) - \sum_{s=1}^{|S|} \mathbf{S}_{z_1}(s) \sum_{s=1}^{|S|} \mathbf{S}_{z_2}(s)}{|S| \sum_{i=1}^{|S|} \mathbf{S}_{z_2}(s)^2 - \left(\sum_{i=1}^{|S|} \mathbf{S}_{z_2}(s) \right)^2}. \quad (36)$$

When the correlation exceeds a threshold T , a spectral predictor is applied; otherwise, spatial prediction is used. In [129], the threshold T is set at 0.5. The equation in (35) also has variations, such as using 4 or 21 neighboring pixels to compute C_p in [157,158], respectively, instead of the 8 pixels used in (35). Instead of assigning predictors rigidly, the assignment can be performed more flexibly [138,159] suggest applying both spatial and spectral predictors for all bands, then computing a weighted average prediction value, i.e., $\hat{p} = C_p \hat{p}_{\text{spatial}} + (1 - C_p) \hat{p}_{\text{spectral}}$.

4.5.7. Kalman Filtering

Kalman filtering can be applied to predict p [160]. It is well-known that Kalman filtering requires three inputs: predicted value ρ , measured value ω and Kalman coefficient K . The final predicted pixel value $\hat{p}$ is calculated as

$$\hat{p} = \rho + K(\omega - \rho). \quad (37)$$

In the context of image compression, Kalman Filtering is fundamentally predicting $\hat{p}$ using two different methods and output the weighted average of ρ and ω , where the weighting is controlled by K . Here, ρ is calculated using DP as in (18), while ω is obtained via the standard 3D-CALIC algorithm. The Kalman coefficient K is according

$$P^- = \beta^2 P_{z-1}^+ + \varepsilon_{z-1}^\rho, \quad (38)$$

$$K = \frac{P^-}{P^- + \varepsilon_{z-1}^\omega}, \quad (39)$$

$$P^+ = (1 - K)P^-. \quad (40)$$

The first equation computes the *a priori* estimate P^- from the prediction error covariance. The second equation then calculates the Kalman coefficient K , which is subsequently used in the third equation to update the *a priori* estimate to an *a posteriori* estimate P^+ . In these equations, β is taken from (18), while ε_{z-1}^ρ and ε_{z-1}^ω can be seen as the errors of ρ and ω respectively,

$$\varepsilon_{z-1}^\rho = p_{z-1} - \beta p_{z-2}, \quad (41)$$

$$\varepsilon_{z-1}^\omega = \omega_{z-1} - p_{z-1}, \quad (42)$$

since the actual value of the predicting pixel $p = p_z$ is not available, p_{z-1} is used instead. Overall, Kalman filtering combined with 3D-CALIC as the measurement input yields a 5% improvement in compression ratio over the baseline 3D-CALIC.

4.6. CCSDS

The CCSDS [39] algorithm is built upon LP. It adopts local-mean-removed pixel $\delta_z(p)$ for prediction, as formulated in (28). It assumes that $\delta_z(p)$ is closely related to the values in the preceding P bands, denoted as $\delta_{z-1}(p)$, $\delta_{z-2}(p)$, ..., $\delta_{z-P}(p)$. Therefore, the predicted

value $\hat{\delta}_z(p)$ is considered a weighted average of the previous P number of $\delta(p)$ values, formulated as

$$\hat{\delta}_z(p) = \mathbf{W}^T \mathbf{V}, \quad (43)$$

where

$$\mathbf{V}_{x,y,z} = \begin{bmatrix} \delta_{z-1}(p) \\ \delta_{z-2}(p) \\ \dots \\ \delta_{z-P}(p) \end{bmatrix}, \quad (44)$$

and $\mathbf{W}$ is the weight vector with the same dimension as $\mathbf{V}$. The elements of $\mathbf{W}$ are continually updated after each pixel prediction to capture the statistical properties of the image. Similar to other feedback algorithms, the initialization and updating of $\mathbf{W}$ are critical for achieving accurate predictions.

In the CCSDS, $\mathbf{W}$ is initialized as

$$\mathbf{W} = \begin{bmatrix} \frac{7}{8} \\ \frac{7}{64} \\ \dots \\ \frac{7}{8^P} \end{bmatrix}, \quad (45)$$

which the sum is not necessarily equal to 1. This does not introduce significant errors because the mean of δ is assumed to be zero, and the sum of $\mathbf{W}$ approaches 1 over multiple iterations. The initialization strategy assigns lower weights to pixels (or channels) further from the current one.

On the other side, $\mathbf{W}$ is updated according to

$$\mathbf{W} \rightarrow \mathbf{W} + \text{sgn}(\delta_z(p) - \hat{\delta}_z(p)) 2^{-\rho} \mathbf{V}, \quad (46)$$

where $\text{sgn}(\cdot)$ denotes the sign function, controls the direction of update, while $2^{-\rho}$ and $\mathbf{V}$ control the magnitude of update. The logic behind the sign function is that if $\delta_z(p) > \hat{\delta}_z(p)$, the predicted value $\hat{\delta}_z(p)$ should increase, which is achieved by aligning the signs of $(\delta_z(p) - \hat{\delta}_z(p))$ with those in $\mathbf{V}$. Conversely, if $\delta_z(p) < \hat{\delta}_z(p)$, the signs should be opposite, resulting in a negative product and decreasing $\hat{\delta}_z(p)$. The magnitude of the update is proportional to $\mathbf{V}$ because both $\hat{\delta}$ and $\mathbf{V}$ have a mean of zero. A larger coefficient in $\mathbf{V}$ suggests that more significant information exists in that channel, so the weight vector $\mathbf{W}$ should adapt more quickly. Finally, the second term, $2^{-\rho}$, controls the learning rate of $\mathbf{W}$. As with most learning algorithms, the learning rate should be fast at the beginning to achieve rapid convergence, then slow down for fine-tuning. The parameter ρ governs this learning rate by

$$\rho = v_{\min} + \left(\frac{x - X}{t_{\text{inc}}} \right), \quad (47)$$

where $v_{\min}$ and t_{inc} are user-defined parameters, and X represents the image width.

With both $\mathbf{W}$ and $\mathbf{V}$ defined, the predicted difference $\hat{\delta}_z(p)$ can be calculated using (43). Once $\hat{\delta}_z(p)$ is determined, the CCSDS-123.0-B standard can losslessly encode the prediction residual $\delta_z(p) - \hat{\delta}_z(p)$. The standard achieves a high compression ratio with low complexity [161], and has been demonstrated through a verified Python 3.7 implementation [162].

4.7. CALIC

4.7.1. 2D CALIC

The Context-based Adaptive Lossless Image Codec (CALIC) [41] is a two-dimensional compression scheme comprising prediction, bias cancellation, and coding modules.

The prediction module employs GAP as the predictor. After performing GAP, CALIC classifies the current pixel into a context. This context is determined by the values of $[a, b, c, d, e, f, 2a - f, 2b - e]$ and an error energy estimator calculated by $\delta_x + \delta_y + 2|p_{z-1} - \hat{p}_{z-1}|$. The former 8 coefficients are quantized to 2 levels by threshold p_{z-1} , whereas the error energy estimator is quantized to 4 levels at thresholds 15, 42 and 85. These quantization creates $4 \times 2^8 = 1024$ contexts in total. Among them, certain combinations are dependent on the others which reduces the number of valid contexts to 576. Within each context, the residuals undergo bias cancellation and sign flapping: if the mean of all encoded residuals is negative, the sign of the encoding residual is reversed, otherwise the sign stays unchanged. The 576 contexts are further quantized to 32 contexts. Standard AC is performed for this final residual based on the 32 contexts.

In order to make 2D CALIC applicable to 3D hyperspectral images, [163,164] proposes to modify the encoding scheme. Three different coding strategies are used under different conditional entropies of the final residual between the current and previous band. If the conditional entropy is 0, it means that the current and the previous bands are identical, thus no further information is needed. On the other side, if the conditional entropy approaches 1, the current band can be considered as independent of the previous band, and thus the ordinary adaptive arithmetic is applied. Otherwise, biplane-wise distributed source coding [165] is applied, making full use of the redundant information between adjacent bands.

4.7.2. 3D-CALIC

The 3D Context-based Adaptive Lossless Image Codec (3D-CALIC) [129] employs two types of predictors: intraband and interband predictors. When the current and previous bands exhibit weak correlation, GAP is applied as CALIC; otherwise, DP as in (18) is utilized.

However, DP may not perform well when p lies on a strong edge, which can be either horizontal or vertical. These edges are predicted using different formulas as

$$\hat{p}_h = a + \beta(p_{z-1} - a_{z-1}), \quad (48)$$

$$\hat{p}_v = b + \beta(p_{z-1} - b_{z-1}). \quad (49)$$

with $\hat{p}_h$ and $\hat{p}_v$ representing the candidates for predicted values assuming horizontal and vertical edges, respectively. It is important to note that the DC shift α is omitted in Equations (48) and (49). The predicted candidates can be combined using a soft approach by

$$\hat{p} = \frac{|p_{z-1} - a_{z-1}|\hat{p}_h + |p_{z-1} - b_{z-1}|\hat{p}_v}{|p_{z-1} - a_{z-1}| + |p_{z-1} - b_{z-1}|}. \quad (50)$$

Or perform a hard combination as

$$\hat{p} = \begin{cases} \hat{p}_h & \text{if } |p_{z-1} - a_{z-1}| - |p_{z-1} - b_{z-1}| > T \\ \hat{p}_v & \text{if } |p_{z-1} - b_{z-1}| - |p_{z-1} - a_{z-1}| > T \\ \frac{1}{2}(\hat{p}_h + \hat{p}_v) & \text{otherwise} \end{cases}. \quad (51)$$

4.7.3. Multiband CALIC

Multiband CALIC (M-CALIC) [166] is an improved version of 3D-CALIC. In 3D-CALIC, a prediction rule selection module is utilized, where the interband predictor is applied to predict a pixel only if the current and previous bands are locally correlated. However, it has been shown in [166] that the interband predictor is generally more effective

for hyperspectral images. Consequently, M-CALIC employs only the interband predictor, eliminating the need for a prediction rule selection module.

Unlike 3D-CALIC, which uses only the previous band as the reference band, ref. [166] demonstrates that the current band is strongly correlated with several previous bands. Assuming a linear relationship, the pixel in the reference band $\tilde{p}$ is formulated as

$$\tilde{p} = k_1 + k_2 p_{z-1} + k_3 p_{z-2}, \quad (52)$$

which is a regression problem solvable by

$$\min_{k_1, k_2, k_3} |k_1 + k_2 \mathbf{I}_{z-1} + k_3 \mathbf{I}_{z-2} - \mathbf{I}_z|_2^2. \quad (53)$$

For simplicity, k_1 , k_2 , and k_3 are set to 8, 0.65, and 0.35 respectively for all images based on trial experiments. Since the distribution of images varies, this reference pixel $\tilde{p}$ is refined through bias compensation with the average error of the previous band:

$$\tilde{p}' = \tilde{p} - (8 + 0.65 \bar{I}_{z-2} + 0.35 \bar{I}_{z-3} - \bar{I}_{z-1}). \quad (54)$$

Finally, DP is applied to compute $\hat{p}$:

$$\hat{p} = \beta \cdot \tilde{p}' + \alpha. \quad (55)$$

4.8. Modification Add-Ons: Band Reordering

Compression algorithms with spectral predictors assume that the current band is closely correlated with the previous band. However, it has been observed that the most correlated band may not necessarily be the preceding one. By reordering the prediction sequence of bands according to their degree of correlation, the prediction residual and, consequently, the compression ratio can be enhanced. In fact, it has been demonstrated in [166] that an improvement in compression ratio of 18% can be achieved with optimal band reordering. A figure in [134] clearly illustrates this concept: after calculating the correlation (using mutual information in this case) between all bands permutatively, as shown in Figure 10a, an optimal prediction dependency emerges in Figure 10b. Here, the circles and edges indicate band indexes and correlations respectively. Band 5 is encoded first, followed by Band 4 and Band 2, while both Bands 1 and 3 use Band 2 as the reference band. The degree of correlation can be approximated using mutual information or the correlation coefficient introduced in (33).

The most basic reordering method involves selecting the best reference band from all previously encoded bands [18,154]. Alternatively, the current band can be set as the reference, encoding the most correlated band subsequently [167]. If a group of images is generated by the same type of hyperspectral camera, the band ordering can be calculated in advance [159,168]. More advanced methods convert reordering into a minimum spanning tree (MST) problem in weighted graphs, where the bands are vertices and the correlations between bands are edges. This graph can be directed or undirected. For directed graphs, this means that each band can serve as the reference band only once, which can be solved using Edmonds' algorithm [169]. Conversely, for undirected graphs [155,170–172], this limitation is removed, and the optimal connections can be found using Prim's algorithm [173], which offers a less complex approach for optimization calculations [155]. Given that constructing the MST for all spectral bands is complex, adaptive band grouping can be employed, followed by creating the MST within each group to reduce complexity [174]. Other modifications to band reordering include applying different types of reordering methods for different images [175], and utilizing reordering aided by normalization [31]. A

summary of prediction-based hyperspectral image compression methods reviewed in this section is given in Table 5.

Table 5. Summary of Prediction-based Hyperspectral Image Compression Methods.

Method	Underlying Principle	Strengths	Limitations
Lookup Table [44]	Dynamically builds a lookup table during predicting, mapping pixel values in band $z - 1$ to their corresponding values in band z	Requires no side information; simple to implement; efficient when strong spectral correlation exists	Suffers from sparsity as bit depth increases, more outliers; residuals have irregular distribution, reduced coding efficiency
Lookup Table with LAIS [118]	Enhances LUT by computing local scaling factors between bands; multiple candidates per index and multi-band prediction ($N \times M$ LUTs)	Improves prediction accuracy for outliers and captures subtle scaling effects between bands	Gains diminish as N and M increase; little improvement when LAIS approaches 1; adds computational overhead
Median Edge Detector [121]	Classifies pixels into flat or edge regions based on causal neighborhood and applies simple piecewise predictors	Low complexity; well-suited for natural images	Considers only local context; struggles with highly textured or noisy hyperspectral data
Simple Lossless Algorithm [124]	Groups neighbors into spatial and spectral categories and predicts using local sums and differences	Exploits both spatial and spectral correlation simultaneously	Neighborhood grouping must be carefully designed; higher complexity compared to MED
Gradient Adjusted Predictor [125]	Classifies pixel into edge or flat categories, and applies weighted prediction rules	High prediction accuracy; robust to edges and local variations	More computationally expensive than MED; requires careful threshold tuning for classification
Differential Predictor [91]	Predicts current pixel value from the corresponding pixel in the previous spectral band	Simple and efficient; low computational cost	Limited modeling capacity; poor performance for weak spectral correlation
Higher-Order Differential Predictor [31]	Uses multiple previous bands and spatial neighbors to improve prediction accuracy	Captures long-range spectral dependencies; improved compression over simple DP	Increased computational cost and memory usage; requires parameter estimation and storage
Linear Predictor [145]	Predicts pixel as a weighted linear combination of previous bands and spatial neighbors, with weights adaptively updated	Accurately models linear inter-band correlation; supports adaptive learning; yields high compression performance	Weight estimation and update add computational complexity; sensitive to initialization and learning rate
Band-Adaptive Selection [174]	Selects MED for spatially correlated bands and DP for spectrally correlated bands based on inter-band statistics	Computationally efficient; adaptively chooses best predictor per band	Not suitable for scenes with rapid spatial variability or low band-to-band correlation
Block-Adaptive Selection [18]	Divides image into blocks and selects predictor type per block using correlation	Improves local adaptability; boosts compression efficiency on heterogeneous scenes	Requires correlation computation per block; incurs side information overhead
Three-Level Cascaded Predictor [113]	Applies a multi-stage prediction: local-mean removal $\rightarrow$ DP $\rightarrow$ LP refinement.	Significantly reduces residual energy; exploits spatial and spectral redundancy hierarchically.	Multi-stage approach adds computational cost and memory usage
CCSDS-123 [161]	Standardized predictor for spaceborne applications; uses local-mean-removed pixel with LP	High compression efficiency; low complexity	Suboptimal for highly nonlinear or nonstationary spectra; requires careful learning rate tuning
M-CALIC [166]	Extends CALIC to hyperspectral data by combining inter-band predictors	State-of-the-art prediction-based compression; excellent performance on diverse image statistics	Computationally expensive; memory-intensive; complex to implement in real-time systems
Band Reordering + 3D-CALIC [168]	Reorders bands to maximize spectral similarity before applying 3D-CALIC prediction	Ensures stronger spectral correlation and improves residual compressibility	Requires preprocessing step for band ordering; increases latency and overall complexity
Kalman Filtering + 3D-CALIC [160]	Fuses DP with 3D-CALIC using Kalman gain for better prediction	Achieves additional compression gain over 3D-CALIC; theoretically optimal fusion	High computational cost; requires per-pixel covariance estimation and storage

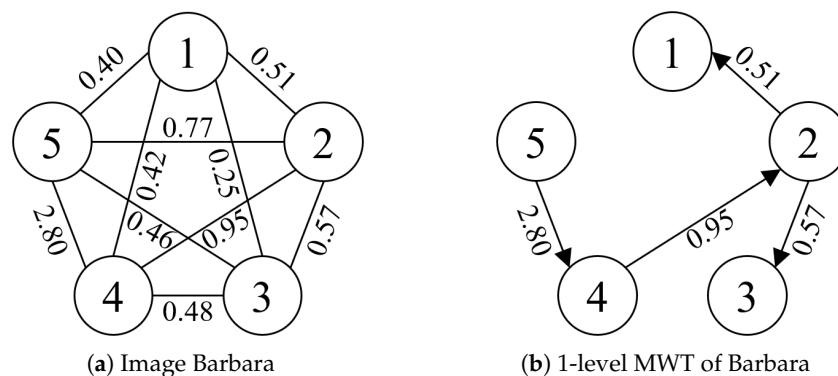


Figure 10. Demonstration of 1-level MWT for image Barbara.

5. Deep Learning

Traditional hyperspectral image compression algorithms rely on handcrafted transforms and explicit statistical models, such as linear transforms, predictive coding, and entropy coding. Their design is analytically derived, assuming specific signal statistics such as Gaussianity or local correlation. In contrast, deep learning-based methods replace these manually designed components with data-driven neural models that learn transforms or predictors directly from data through optimization. These approaches can capture complex nonlinear correlations across spatial and spectral domains, offering greater representational flexibility. However, since neural networks act as high-accuracy regressors, they generate estimated values rather than exact ones, making them unsuitable for direct application in lossless compression. As a result, they are typically employed either as irreversible transforms with residual encoding (see Section 3.6) or as pixel-by-pixel prediction methods.

5.1. Deep Learning as Irreversible Transforms

The most popular irreversible transform in deep learning is the auto-encoder structure [176–178]. This structure integrally reduces the dimensionality of the input data while reconstructing the original data with minimal errors. For such structures, both the bitstream outputted by the encoder and the reconstruction residuals need to be stored in order to reconstruct the original image losslessly. To reduce these residuals, ref. [179] employed a stacked autoencoder. Alternatively, deep learning can be used more simply as a downsampling and upsampling tool, serving a similar function but with greater interpretability than autoencoders, as demonstrated in [180]. Recent research has seen a growing reliance on transformer-based models to capture long-range dependencies in data, which is a key limitation of convolutional autoencoders. For instance, ref. [181] proposed a hybrid convolution–transformer autoencoder that uses a 1D CNN for local spectral features and transformer layers with rotary embeddings for global dependencies, achieving competitive compression fidelity. Similarly, ref. [182] designed a CNN-transformer mixture architecture for compressed sensing, significantly improving reconstruction quality. In a different approach, ref. [183] introduces a fully transformer-based autoencoder that surpasses convolutional models with reduced computational requirements.

Beyond basic autoencoders, learned compression for HSI has evolved significantly. Generative neural networks have been proposed [184] to learn the underlying probability distribution of HSI data, offering a compact representation. To enhance the preservation of critical information, methods now incorporate semantic guidance. For instance, ref. [185] uses classification maps to maintain the spectral fidelity, while [186] employs contrastive learning to prevent the collapse of discriminative spectral features during compression. Architecturally, specialized models like [187] explicitly target spatio-spectral redundancies using channel attention and 3D convolutions. At the frontier of generative modeling,

diffusion-based frameworks such as [188] decouple efficient residual compression from high-fidelity, iterative reconstruction, achieving superior spectral and perceptual quality.

Additionally, ref. [189] subsamples the input image into six subimages. The main diagonal subimage is compressed using a traditional method, while the other five subimages are compressed by P2Net using the prior probability distribution obtained from the main diagonal subimage. Building on this idea, ref. [190] further incorporates pixel probability modules to achieve more accurate predictions.

5.2. Deep Learning as Prediction Methods

For prediction methods leveraging deep learning, the Recursive Neural Network (RNN) is particularly well-suited for predicting sequential data. For example, ref. [191] proposes a C-DPCM-RNN network to enhance the generalization ability and prediction accuracy of the model. Meanwhile, ref. [192] incorporates the receptance weighted key value (RWKV) for both line and spectral predictors, facilitating fast and accurate predictions [193] further integrates an attention module into the RNN. By focusing on important features, attention mechanisms help in achieving more efficient and visually pleasing compressed images. Moreover, ref. [194] applies a simple transformer encoding unit to predict the current pixel value from 20 neighbors. Unlike pixel-by-pixel prediction, ref. [195] further advances the approach by predicting entire rows, using inputs from preceding rows in the same band as well as corresponding rows from previous bands.

Beyond directly predicting pixel values, ref. [196] advances this approach by predicting the weight $\mathbf{W}$ of a linear predictor described in Section 4.4, leveraging the longer-term dependencies of $\mathbf{W}$. It inputs the predicted $\mathbf{W}$ into a RNN, which outputs the adjusted $\mathbf{W}$. Conversely, ref. [197] predicts $\mathbf{W}$ directly using a convolutional neural network by inputting pixel values surrounding the current pixel. Moreover, the prediction with deep learning can also be performed in the wavelet domain, as demonstrated in [198]. A summary of deep learning hyperspectral image compression methods reviewed in this section is given in Table 6.

5.3. Challenges in Strictly Lossless Compression

Despite the strong representational power, deep learning models face significant hurdles in strictly lossless compression scenarios, which limit their practicality, especially for onboard deployment.

- **Entropy Modeling and Residual Encoding:** The prediction or reconstruction residuals generated by deep networks often deviate significantly from the smooth, Laplacian-like distributions that traditional entropy coders are optimized for. These residuals can be sparse, multi-modal, or exhibit irregular patterns. Consequently, the bit savings from the compact latent representation can be offset by the cost of encoding the residuals, reducing the overall performance.
- **Computational and Memory Overhead:** The computational and memory footprint of deep models is typically orders of magnitude higher than that of traditional algorithms like CCSDS-123. The requirements for powerful GPUs/CPU and large RAM buffers are incompatible with the low-power, radiation-hardened hardware used in spaceborne or aerial platforms, making real-time, onboard inference infeasible for most current architectures.
- **Limited Generalizability:** A model trained on one type of HSI often generalizes poorly to data from different sources or with different characteristics, due to changes in the underlying data distribution. Furthermore, deep networks may overfit to the specific statistical properties of their training set, learning to exploit redundancies that do not generalize across scenes. As a result, when encountering a new type of scene,

prediction accuracy declines, yielding larger residuals and ultimately reducing the compression ratio.

Table 6. Summary of Deep Learning Hyperspectral Image Compression Methods.

Method	Underlying Principle	Strengths	Limitations
Autoencoder [177]	Learns a nonlinear mapping to compress input data into a low-dimensional latent representation, then reconstructs it. Residuals must be encoded for lossless reconstruction.	Powerful nonlinear dimensionality reduction; effective feature extraction; flexible architectures.	Requires storage of residuals; training can be computationally expensive.
Stacked Autoencoder [179]	Uses multiple autoencoders in sequence to iteratively reduce residual error.	Reduced residuals compared to a single autoencoder	Increased network depth leads to higher computational cost and memory consumption.
Transformer-based Autoencoder [181]	Uses self-attention mechanisms to capture long-range dependencies in data.	Better modeling of global correlations; competitive compression fidelity.	High computational complexity; requires large training data.
Generative Models [188]	Learns underlying probability distribution of HSI data using generative networks.	Compact representation; superior spectral and perceptual quality.	Training instability; complex architecture; slow sampling.
Transform with Subimages [190]	Subsamples input to multiple subimages, compresses one subimage conventionally, and predicts others using learned probability distribution.	Effectively reduces redundancy between subimages; improves coding efficiency by exploiting spectral-spatial priors.	More complex architecture and longer training time.
Pixel-wise Prediction [192]	Use neural network for pixel-by-pixel prediction.	High prediction accuracy; good generalization ability.	Sequential nature may limit inference speed; computationally heavy for large scenes.
Row-wise Prediction [193]	Predicts entire rows based on previous rows and bands, instead of pixel-by-pixel.	Significantly accelerates prediction; reduces context-switching overhead.	May lose fine local adaptivity compared to pixel-wise prediction.
LP with Weight Prediction [197]	Predicts linear predictor weights W using neural network to gather both spatial and spectral dependencies.	Adapts well to varying spectral statistics; reduces residual entropy.	Training requires large datasets; may overfit if spectral variability is low.
Prediction in Wavelet Domain [198]	Performs prediction in the wavelet-transformed domain, leveraging multiscale features.	Improves coding efficiency by separating frequency components; reduces spatial redundancy.	Additional computational overhead for wavelet transforms.

6. Algorithm Performance and Comparative Assessment

6.1. Quantitative Comparisons

The quantitative comparison of representative hyperspectral image lossless compression algorithms is presented in Table 7, expressed in bits per pixel per band (BPPPB) across five standard datasets: Botswana (BO), Kennedy Space Center (KS), Indian Pines (IP), Pavia University (PU), and Salinas (SA) [199]. To ensure a reproducible and fair comparison, all experiments were conducted under a consistent environment. The experiments were conducted on Windows 10 with an AMD 7900X CPU and an NVIDIA GeForce RTX 4090 GPU. Traditional methods were implemented in MATLAB 2019a, while deep learning models were built using PyTorch. For methods requiring training, we employed a leave-one-image-out cross-validation scheme, where each test image was trained on the other four available images. The data was partitioned into patches of size $64 \times 64 \times 48$. All networks were trained for a maximum of 160 epochs with a batch size of 32, using the Adam optimizer with an initial learning rate of 5×10^{-4} and He initialization for the weights.

Among the transform-based methods (Methods 1–11), an interesting observation is that converting 3D hyperspectral data into 2D followed by JPEG2000 compression [98] achieves the best performance. This is likely because the spectral correlation is generally much stronger than the spatial one; hence, methods capable of fully exploiting spectral

redundancy alone can achieve high compression efficiency. Among the three DWT-based schemes (Methods 1–3), SPECK [25] outperforms others, as coefficients within the same wavelet band exhibit similar magnitudes, forming more zerotrees. Between DWT [75] and KLT [95], the decorrelation capabilities are comparable, and varying the wavelet basis for DWT yields only marginal compression differences on these datasets. Both DCT+residual [35] and VQ [100] must store the lossy coefficients along with residuals, leading to inferior compression efficiency compared with standard transform-based SPECK.

Table 7. Summary of Compression Performances.

Class	Group	Method	BO	KS	IP	PU	SA	AVG
Transform	3D DWT	EZW [20]	7.74	5.70	7.80	7.58	6.00	6.96 ²³
		SPIHT [22]	7.35	6.21	7.59	7.10	6.08	6.86 ²²
		SPECK [25]	7.07	5.40	7.23	6.97	5.39	<u>6.41</u> ⁰⁸
	Spectral Decorrelation	JP2K [88]	8.52	5.47	7.83	9.70	6.46	7.60 ²⁸
		JP2K + DCT [53]	8.46	5.27	8.18	8.58	6.55	7.41 ²⁶
		JP2K + KLT [95]	7.32	4.10	7.86	7.80	6.16	6.65 ¹⁸
		JP2K+DWT5/11-C [75]	7.71	4.58	7.70	7.25	5.96	6.64 ¹⁶
		JP2K + DWT 5/3 [75]	7.76	4.44	7.57	7.56	5.94	6.65 ¹⁸
		JP2K + 3Dto2D [98]	7.40	3.88	7.43	7.26	5.69	<u>6.33</u> ⁰⁶
	Irreversible Transform	DCT + Residual [35]	7.35	5.44	6.96	7.95	5.75	6.69 ²⁰
		VQ [100]	7.50	5.47	7.00	7.49	5.75	6.64 ¹⁶
Prediction	LUT	LUT [44]	7.81	3.74	7.36	7.67	5.39	<u>6.40</u> ⁰⁷
		LUT + LAIS [118]	7.44	4.86	7.40	7.35	6.06	6.62 ¹⁴
		LUT + Distance [118]	7.64	3.88	6.94	7.42	5.20	6.22 ⁰⁵
	Spatial	MED [121]	8.84	5.97	8.34	9.79	7.34	8.06 ³⁰
		Simple Lossless [124]	8.04	4.10	8.12	7.68	6.98	6.98 ²⁴
		GAP (CALIC) [125]	8.68	5.50	7.84	9.35	6.64	7.60 ²⁸
	Differential	DP [91]	7.24	6.40	7.17	6.95	5.31	6.61 ¹³
		Higher Orders DP [31]	7.17	5.80	7.07	6.70	5.32	<u>6.41</u> ⁰⁸
	Linear	Linear [145]	6.65	4.88	6.54	6.96	5.09	6.02 ⁰³
		CCSDS-123 [161]	6.46	4.38	6.48	6.56	4.74	5.72 ⁰²
	CALIC	3D-CALIC [168]	7.41	4.24	7.39	7.16	6.78	6.60 ¹¹
		M-CALIC [166]	7.24	4.21	7.11	6.85	5.46	6.17 ⁰⁴
	Hybrid	Band Adaptive [174]	7.41	4.24	7.39	7.16	6.78	6.60 ¹¹
		Block Adaptive [18]	8.50	5.13	8.29	7.29	8.00	7.44 ²⁷
		Kalman Filtering [160]	7.06	5.98	7.12	6.66	6.31	6.63 ¹⁵
Deep Learning	Transform	AE [177]	7.65	6.75	7.71	7.94	6.79	7.37 ²⁵
	Prediction	Pixel [192]	7.09	5.81	6.78	6.66	5.77	<u>6.42</u> ¹⁰
		Row [193]	7.43	6.22	7.09	7.12	6.09	6.79 ²¹
		LP [197]	6.38	4.36	6.38	6.40	4.73	5.65 ⁰¹

The overall performance rank is denoted by superscripts of the average BPPPB. The top five and next five ranked average BPPPB are emphasized in **bold** and underline.

Prediction-based methods (Methods 12–26) generally outperform transform-based ones. The international standard CCSDS-123 [161] achieves the best results by effectively utilizing spectral correlation through adaptive weights $\mathbf{W}$ and spatial correlation via local mean removal. LUT-based schemes (Methods 12–14) show limited benefit from LAIS matching for outliers [118], while distance-based matching provides modest improvement. Compared with spatial predictors such as MED [121], simple lossless (an improved MED variant) [124], and GAP [125], spectral predictors like DP [91] and linear regression [145] perform better, with the linear model showing particularly strong results. CALIC [125], 3D-CALIC [168], and M-CALIC [166] represent successive refinements of the same principle, yielding progressively improved compression performance. Although adaptive schemes (Methods 24–26) attempt to merge spatial and spectral prediction, their fusion mechanisms are not fully systematic, resulting in limited gains.

For deep learning-based methods (Methods 27–30), the LP network [197] achieves the lowest average BPPPB, demonstrating that adaptive linear prediction integrated with neural modeling can approach near-optimal lossless compression. Autoencoder-based [177], pixel-wise [192], and row-wise [193] predictors also deliver promising results. Although their compression ratios are slightly inferior to those of finely tuned statistical models, these methods offer greater flexibility by learning nonlinear spatial-spectral dependencies directly from data.

Overall, deep learning-based prediction [197] exhibits the greatest potential for future improvement and is particularly suited for applications with stringent storage constraints, as training can be customized for specific datasets. In contrast, CCSDS-123 [161] remains an excellent choice for resource-limited environments due to its low complexity and competitive performance. Finally, for scenarios requiring compatibility with standard decompression tools, JPEG2000 with 3D-to-2D conversion provides a practical and effective solution.

6.2. Comparative Analysis of Algorithm Categories

Based on the comprehensive review of methodologies and quantitative results presented in Table 7, this subsection provides an analysis of the three main categories of HSI lossless compression algorithms, highlighting their inherent characteristics, performance, and application scenarios.

- Transform methods decorrelate HSI by projecting it into a different domain. The main strength lies in exploiting global redundancies while enabling useful features such as progressive transmission and multi-resolution analysis, which are advantageous for data browsing and scalable streaming. However, the computational cost and large memory footprint make them less suitable for real-time, on-board compression. Overall, for use cases requiring broad compatibility and convenient decoding, transform-based approaches, particularly JPEG2000 with 3D-to-2D conversion, offer a practical and effective solution.
- Prediction methods estimate each pixel from its neighbors and encode the resulting prediction residuals. They typically achieve high compression efficiency with low complexity, due to the ability to capture local spatial and spectral correlations with minimal memory usage. The main drawbacks are limited parallelization due to sequential processing and a vulnerability to error propagation. Despite these constraints, prediction-based approaches, especially the CCSDS-123 standard, remain highly suitable for resource-constrained platforms such as satellite on-board systems, where low computational demand and strong overall performance are essential.
- Deep learning methods replace manually designed transforms and predictors with data-driven deep networks. Their key advantage is the capacity to learn complex

nonlinear spatial-spectral dependencies, enabling flexible and potentially superior compression. The primary challenges include high training costs, limited interpretability, and the lack of standardized frameworks. Nevertheless, deep learning-based prediction currently shows the greatest promise for future advances, particularly in scenarios with severe storage constraints where a model can be tailored to a specific hyperspectral dataset to achieve near-optimal performance in controlled, ground-based processing environments.

7. Conclusions

Hyperspectral image lossless compression has garnered significant attention, leading to extensive research across various aspects of compression algorithms, including entropy encoders, scanning orders, primary compression techniques, and other modification add-ons. This review offers a detailed analysis of the principal compression algorithms, systematically categorizing them into three distinct groups. The other areas are briefly introduced, creating a comprehensive understanding of the entire compression process from beginning to end. For each category, we present various methods along with modification techniques highlighted in the literature, establishing a solid foundation for future advancements. Through this review, we aim to illuminate areas that require further exploration, ultimately guiding future research endeavors toward overcoming existing gaps and enhancing the efficacy of hyperspectral image compression.

Despite these advances, several critical challenges remain unresolved:

- The trade-offs between compression efficiency and computational complexity, which limit practical deployment in real-time or resource-constrained scenarios.
- The lack of standardized and universal datasets that are consistently tested across different methods, hindering fair comparisons and reproducibility.
- Robust performance across diverse scenes, varying noise conditions, and heterogeneous acquisition scenarios continues to be a major challenge.
- The limited adaptability of existing entropy coders to the highly correlated and complex distributions inherent in hyperspectral data.
- The prevalence of traditional methods that are largely based on permutations or incremental modifications of existing techniques, restricting significant innovation.
- The insufficient exploration and development of learning-based methods that strictly adhere to lossless requirements, leaving their potential untapped.

Future research directions should focus on building large-scale, standardized benchmark datasets that can support fair evaluation across methods, while also exploring entropy modeling techniques tailored to the unique statistical structures of hyperspectral data. The integration of physics-driven priors and domain knowledge into machine learning frameworks could enhance robustness and interpretability, especially under noise and varying acquisition conditions. Moreover, hybrid approaches that bridge traditional algorithmic strategies with learning-based paradigms may unlock new levels of compression efficiency while preserving strict lossless guarantees.

Author Contributions: Writing—original draft preparation, S.L. and F.S.; writing—review and editing, Z.Y. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation of China No. 62201470, and Fundamental Research Funds for the Central Universities, No. G2023KY05108.

Data Availability Statement: The data supporting this research was completely sourced from public repositories. Detailed references are provided in the manuscript..

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Yuen, P.W.; Richardson, M. An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition. *Imaging Sci. J.* **2010**, *58*, 241–253. [\[CrossRef\]](#)
2. Fox, N.; Parbhakar-Fox, A.; Moltzen, J.; Feig, S.; Goemann, K.; Huntington, J. Applications of hyperspectral mineralogy for geoenvironmental characterisation. *Miner. Eng.* **2017**, *107*, 63–77. [\[CrossRef\]](#)
3. Willoughby, C.T.; Folkman, M.A.; Figueroa, M.A. Application of hyperspectral-imaging spectrometer systems to industrial inspection. In Proceedings of the Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology. International Society for Optics and Photonics, Philadelphia, PA, USA, 19 January 1996; Volume 2599, pp. 264–272.
4. Park, B.; Lu, R. *Hyperspectral Imaging Technology in Food and Agriculture*; Springer: Berlin/Heidelberg, Germany, 2015.
5. Obermeier, W.A.; Lehnert, L.W.; Pohl, M.J.; Gianonni, S.M.; Silva, B.; Seibert, R.; Bendix, J. Grassland ecosystem services in a changing environment: The potential of hyperspectral monitoring. *Remote Sens. Environ.* **2019**, *232*, 111273. [\[CrossRef\]](#)
6. ISO/IEC 15444-1; Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System. International Organization for Standardization: Geneva, Switzerland, 2000.
7. Fei, B. Chapter 3.6—Hyperspectral imaging in medical applications. In *Hyperspectral Imaging*; Amigo, J.M., Ed.; Elsevier: Amsterdam, The Netherlands, Data Handling in Science and Technology; 2019; Volume 32, pp. 523–565.
8. Selci, S. The future of hyperspectral imaging. *J. Imaging* **2019**, *5*, 84. [\[CrossRef\]](#)
9. Dua, Y.; Kumar, V.; Singh, R.S. Comprehensive review of hyperspectral image compression algorithms. *Opt. Eng.* **2020**, *59*, 090902. [\[CrossRef\]](#)
10. Afrin, A.; Al Mamun, M. A comprehensive review of deep learning methods for hyperspectral image compression. In Proceedings of the International Conference on Advancement in Electrical and Electronic Engineering, Gazipur, Bangladesh, 25–27 April 2024; IEEE: New York, NY, USA, 2024; pp. 1–6.
11. Zhang, F.; Chen, C.; Wan, Y. A survey on hyperspectral remote sensing image compression. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 16–21 July 2023; IEEE: New York, NY, USA, 2023; pp. 7400–7403.
12. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Kruse, F.A.; Lefkoff, A.B.; Boardman, J.W.; Heidebrecht, K.B.; Shapiro, A.; Barloon, P.; Goetz, A.F. The spectral image processing system (SIPS)—interactive visualization and analysis of imaging spectrometer data. *Remote Sens. Environ.* **1993**, *44*, 145–163. [\[CrossRef\]](#)
14. Joshi, V.; Rani, J.S. A band interleaved by line (BIL) architecture of a simple lossless algorithm (SLA) for on-board satellite hyperspectral data compression. In Proceedings of the International Symposium on Circuits and Systems, London, UK, 25–28 May 2025; IEEE: New York, NY, USA, 2025; pp. 1–5.
15. Joshi, V.; Rani, J.S. A band interleaved by pixel (BIP) architecture of a simple lossless algorithm (SLA) for on-board satellite hyperspectral data compression. In Proceedings of the IEEE Interregional NEWCAS Conference, Paris, France, 22–25 June 2025; IEEE: New York, NY, USA, 2025; pp. 490–494.
16. Lempel, A.; Ziv, J. Compression of two-dimensional data. *IEEE Trans. Inf. Theory* **1986**, *32*, 2–8. [\[CrossRef\]](#)
17. Memon, N.; Neuhoff, D.; Shende, S. An analysis of some common scanning techniques for lossless image coding. In *Proceedings of the Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2 November–5 November 1997*; IEEE: New York, NY, USA, 1997; Volume 2, pp. 1446–1450.
18. Zhu, F.; Wang, H.; Yang, L.; Li, C.; Wang, S. Lossless Compression for Hyperspectral Images based on Adaptive Band Selection and Adaptive Predictor Selection. *Ksii Trans. Internet Inf. Syst.* **2020**, *14*, 3295–3311.
19. Jia, L.; Liang, B.; Li, M.; Liu, Y.; Chen, Y.; Ding, J. Efficient 3D Hilbert curve encoding and decoding algorithms. *Chin. J. Electron.* **2022**, *31*, 277–284. [\[CrossRef\]](#)
20. Shapiro, J. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Process.* **1993**, *41*, 3445–3462. [\[CrossRef\]](#)
21. Nagendran, R.; Ramadass, S.; Thilagavathi, K.; Ravuri, A. Lossless hyperspectral image compression by combining the spectral decorrelation techniques with transform coding methods. *Int. J. Remote Sens.* **2024**, *45*, 6226–6248. [\[CrossRef\]](#)
22. Kim, B.J.; Pearlman, W. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In Proceedings of the DCC '97. Data Compression Conference, Snowbird, UT, USA, 25–27 March 1997; pp. 251–260.
23. Pearlman, W.; Islam, A.; Nagaraj, N.; Said, A. Efficient, low-complexity image coding with a set-partitioning embedded block coder. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 1219–1235. [\[CrossRef\]](#)

24. Lin, G.; Liu, Z. 3D wavelet video codec and its rate control in ATM network. In Proceedings of the International Symposium on Circuits and Systems, Orlando, FL, USA, 30 May–2 June 1999; Volume 4, pp. 447–450.
25. Bajpai, S. Low Complexity Image Coding Technique for Hyperspectral Image Sensors. *Multimed. Tools Appl.* **2023**, *82*, 31233–31258. [\[CrossRef\]](#)
26. Bajpai, S.; Kidwai, N.R.; Singh, H.V.; Singh, A.K. Low memory block tree coding for hyperspectral images. *Multimed. Tools Appl.* **2019**, *78*, 27193–27209. [\[CrossRef\]](#)
27. Chow, K.; Tzamarias, D.E.O.; Hernández-Cabronero, M.; Blanes, I.; Serra-Sagristà, J. Analysis of Variable-Length Codes for Integer Encoding in Hyperspectral Data Compression with the k 2-Raster Compact Data Structure. *Remote Sens.* **2020**, *12*, 1983. [\[CrossRef\]](#)
28. Bajpai, S.; Kidwai, N.R.; Singh, H.V.; Singh, A.K. A low complexity hyperspectral image compression through 3D set partitioned embedded zero block coding. *Multimed. Tools Appl.* **2022**, *81*, 841–872. [\[CrossRef\]](#)
29. Liu, S.; Chen, J.; Ai, Y.; Rahardja, S. An Optimized Quantization Constraints Set for Image Restoration and its GPU Implementation. *IEEE Trans. Image Process.* **2020**, *29*, 6043–6053. [\[CrossRef\]](#)
30. Yu, R.; Ko, C.; Rahardja, S.; Lin, X. Bit-plane Golomb coding for sources with Laplacian distributions. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, China, 6–10 April 2003; Volume 4, p. IV–277.
31. Anand Swamy, A.; Mamatha, A.; Shylashree, N.; Nath, V. Lossless Compression of Hyperspectral Imagery by Assimilating Decorrelation and Pre-processing with Efficient Displaying Using Multiscale HDR Approach. *IETE J. Res.* **2022**, *69*, 6673–6684. [\[CrossRef\]](#)
32. Song, H.; Song, Z.; Deng, G.; Ma, Y.; Ma, P. Differential prediction-based lossless compression with very low-complexity for hyperspectral data. In Proceedings of the IEEE International Conference on Communication Technology, Jinan, China, 25–28 September 2011; pp. 323–327.
33. Mamatha, A.; Singh, V. Lossless hyperspectral image compression based on prediction. In Proceedings of the IEEE Recent Advances in Intelligent Computational Systems, hITrivandrum, India, 19–21 December 2013; pp. 193–198.
34. Mamatha, A.S.; Singh, V. Lossless hyperspectral image compression using intraband and interband predictors. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Delhi, India, 24–27 September 2014; pp. 332–337.
35. Baizert, P.; Pickering, M.; Ryan, M. Compression of hyperspectral data by spatial/spectral discrete cosine transform. In Proceedings of the Scanning the Present and Resolving the Future. Proceedings. IEEE International Geoscience and Remote Sensing Symposium, Sydney, NSW, Australia, 9–13 July 2001; Volume 4, pp. 1859–1861.
36. Ding, J.J.; Chen, H.H.; Wei, W.Y. Adaptive golomb code for joint geometrically distributed data and its application in image coding. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *23*, 661–670. [\[CrossRef\]](#)
37. Joshi, V.; Rani, J.S. An on-board satellite multispectral and hyperspectral compressor (MHyC): An efficient architecture of a simple lossless algorithm. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2025**, *72*, 2167–2177. [\[CrossRef\]](#)
38. Jiang, Z.; Pan, W.D.; Shen, H. Universal golomb—Rice coding parameter estimation using deep belief networks for hyperspectral image compression. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3830–3840. [\[CrossRef\]](#)
39. CCSDS 123.0.B-2; Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression. Consultative Committee for Space Data Systems: Hamburg, Germany, 2019.
40. Altamimi, A.; Ben Youssef, B. Lossless and Near-Lossless Compression Algorithms for Remotely Sensed Hyperspectral Images. *Entropy* **2024**, *26*, 316. [\[CrossRef\]](#)
41. Wu, X.; Memon, N. Context-based, adaptive, lossless image coding. *IEEE Trans. Commun.* **1997**, *45*, 437–444. [\[CrossRef\]](#)
42. Witten, I.H.; Neal, R.M.; Cleary, J.G. Arithmetic coding for data compression. *Commun. Acn* **1987**, *30*, 520–540. [\[CrossRef\]](#)
43. Martin, G.N.N. Range encoding: An algorithm for removing redundancy from a digitised message. In Proceedings of the Proc. Institution of Electronic and Radio Engineers International Conference on Video and Data Recording, Southampton, UK, 24–27 July 1979; p. 48.
44. Mielikainen, J. Lossless compression of hyperspectral images using lookup tables. *IEEE Signal Process. Lett.* **2006**, *13*, 157–160. [\[CrossRef\]](#)
45. Mielikainen, J.; Toivanen, P. Optimal granule ordering for lossless compression of ultraspectral sounder data. In Proceedings of the Satellite Data Compression, Communication, and Processing IV, San Diego, CA, USA, 5 September 2008; Volume 7084, p. 708404.
46. Mielikainen, J.; Toivanen, P. Lossless Compression of Ultraspectral Sounder Data Using Linear Prediction With Constant Coefficients. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 495–498. [\[CrossRef\]](#)
47. Mielikainen, J.; Toivanen, P. Clustered DPCM for the lossless compression of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 2943–2946. [\[CrossRef\]](#)
48. Mielikainen, J.; Huang, B. Lossless Compression of Hyperspectral Images Using Clustered Linear Prediction with Adaptive Prediction Length. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 1118–1121. [\[CrossRef\]](#)

49. Duda, J.; Tahboub, K.; Gadgil, N.J.; Delp, E.J. The use of asymmetric numeral systems as an accurate replacement for Huffman coding. In Proceedings of the Picture Coding Symposium, Cairns, QLD, Australia, 31 May–3 June 2015; pp. 65–69.
50. Alonso, T.; Sutter, G.; De Vergara, J.E.L. LOCO-ANS: An Optimization of JPEG-LS Using an Efficient and Low-Complexity Coder Based on ANS. *IEEE Access* **2021**, *9*, 106606–106626. [\[CrossRef\]](#)
51. Wang, D.; Zhang, Y. A lossless compression of remote sensing images based on ANS entropy coding algorithm. In Proceedings of the MIPPR Remote Sensing Image Processing, Geographic Information Systems, and Other Applications, Wuhan, China, 7 March 2024; SPIE: Bellingham, WA, USA, 2024; Volume 13088, pp. 17–24.
52. Wang, L.; Wu, J.; Jiao, L.; Shi, G. Lossy-to-Lossless Hyperspectral Image Compression Based on Multiplierless Reversible Integer TDLT/KLT. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 587–591. [\[CrossRef\]](#)
53. Penna, B.; Tillo, T.; Magli, E.; Olmo, G. Embedded lossy to lossless compression of hyperspectral images using JPEG 2000. In Proceedings of the Proceedings. IEEE International Geoscience and Remote Sensing Symposium, Seoul, Republic of Korea, 29 July 2005; Volume 1, p. 4.
54. Can, E.; Karaca, A.C.; Danışman, M.; Urhan, O.; Güllü, M.K. Compression of hyperspectral images using luminance transform and 3D-DCT. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; IEEE: New York, NY, USA, 2018; pp. 5073–5076.
55. Luo, X.; Guo, L.; Liu, Z. Lossless Compression of Hyperspectral Imagery Using Integer Principal Component Transform and 3-D Tarp Coder. In Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, China, 30 July–1 August 2007; Volume 1, pp. 553–558.
56. Galli, L.; Salzo, S. Lossless hyperspectral compression using KLT. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AK, USA, 20–24 September 2004; Volume 1, p. 316.
57. Luo, X.; Guo, L.; Liu, Z. Reversible Integer Principal Component Transform for Hyperspectral Imagery Lossless Compression. In Proceedings of the IEEE International Conference on Control and Automation, Guangzhou, China, 30 May–1 June 2007; pp. 2968–2972.
58. Tang, X.; Pearlman, W.A. Three-dimensional wavelet-based compression of hyperspectral images. In *Hyperspectral Data Compression*; Springer: Berlin/Heidelberg, Germany, 2006; Chapter 10, pp. 273–308.
59. Zhang, J.; Fowler, J.E.; Liu, G. Lossy-to-Lossless Compression of Hyperspectral Imagery Using Three-Dimensional TCE and an Integer KLT. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 814–818. [\[CrossRef\]](#)
60. Cheng, K.J.; Dill, J. Lossless to Lossy Dual-Tree BEZW Compression for Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 5765–5770. [\[CrossRef\]](#)
61. Jin, W.; Xiao-ling, Z.; Lan-sun, S.; Yan, C. Hyperspectral image lossless compression using wavelet transforms and trellis coded quantization. In Proceedings of the IEEE International Symposium on Communications and Information Technology, Beijing, China, 12–14 October 2005; Volume 2, pp. 1452–1455.
62. Calderbank, A.R.; Daubechies, I.; Sweldens, W.; Yeo, B.L. Lossless image compression using integer to integer wavelet transforms. In Proceedings of the International Conference on Image Processing, Santa Barbara, CA, USA, 26–29 October 1997; Volume 1, pp. 596–599.
63. Adams, M.; Kossentni, F. Reversible integer-to-integer wavelet transforms for image compression: Performance evaluation and analysis. *IEEE Trans. Image Process.* **2000**, *9*, 1010–1024. [\[CrossRef\]](#)
64. Zheng, J.; Fang, J.; Han, C. The selection of reversible integer-to-integer wavelet transforms for dem multi-scale representation and progressive compression. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, China, 3–11 July 2008; Volume 37.
65. Le Gall, D.; Tabatabai, A. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, New York, NY, USA, 11–14 April 1988; Volume 2, pp. 761–764.
66. Zandi, A.; Allen, J.D.; Schwartz, E.L.; Boliek, M. CREW: Compression with reversible embedded wavelets. In Proceedings of the Proceedings DCC Data Compression Conference, Snowbird, UT, USA, 28–30 March 1995; pp. 212–221.
67. Said, A.; Pearlman, W.A. An image multiresolution representation for lossless and lossy compression. *IEEE Trans. Image Process.* **1996**, *5*, 1303–1310. [\[CrossRef\]](#)
68. Strang, G.; Nguyen, T. *Wavelets and Filter Banks*; SIAM: Philadelphia, PA, USA, 1996.
69. Sheng, F.; Bilgin, A.; Sementilli, P.; Marcelling, M. Lossy and lossless image compression using reversible integer wavelet transforms. In Proceedings of the Proceedings International Conference on Image Processing, Chicago, IL, USA, 7 October 1998; Volume 3, pp. 876–880.
70. Gormish, M.J.; Schwartz, E.L.; Keith, A.F.; Boliek, M.P.; Zandi, A. Lossless and nearly lossless compression for high-quality images. In Proceedings of the Very High Resolution and Quality Imaging II, San Jose, CA, USA, 4 April 1997; Volume 3025, pp. 62–70.

71. Calderbank, A.R.; Daubechies, I.; Sweldens, W.; Yeo, B.L. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal.* **1998**, *5*, 332–369. [[CrossRef](#)]
72. Adams, M.D.; Kossentini, F. Low-complexity reversible integer-to-integer wavelet transforms for image coding. In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, 22–24 August 1999; pp. 177–180.
73. ISO/IEC JTC 1/SC 29/WG 1 N1015; Report on Core Experiment CodEff4: Performance Evaluation of Several Rerversible Integer-to-Integer Wavelet Transforms in the JPEG-2000 Verification Model (Version 2.1). ISO: Geneva, Switzerland, 1998.
74. Antonini, M.; Barlaud, M.; Mathieu, P.; Daubechies, I. Image coding using wavelet transform. *IEEE Trans. Image Process.* **1992**, *1*, 205–220. [[CrossRef](#)] [[PubMed](#)]
75. Töreyn, B.U.; Yilmaz, O.; Mert, Y.M. Evaluation of on-board integer wavelet transform based spectral decorrelation schemes for lossless compression of hyperspectral images. In Proceedings of the Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Lausanne, Switzerland, 24–27 June 2014; pp. 1–4.
76. Töreyn, B.U.; Yilmaz, O.; Mert, Y.M.; Türk, F. Lossless hyperspectral image compression using wavelet transform based spectral decorrelation. In Proceedings of the International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, 16–19 June 2015; pp. 251–254.
77. Hou, Y.; Liu, G. Hyperspectral Image Lossless Compression Using the 3D Set Partitioned Embedded Zero Block Coding Alogrithm. In Proceedings of the International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 2, pp. 955–958.
78. Hou, Y.; Liu, G. Hyperspectral image lossy-to-lossless compression using the 3D Embedded Zeroblock Coding alogrithm. In Proceedings of the International Workshop on Earth Observation and Remote Sensing Applications, Beijing, China, 30 June–2 July 2008; pp. 1–6.
79. Hou, Y.; Liu, G. Lossy-to-lossless compression of hyperspectral image using the improved AT-3D SPIHT algorithm. In Proceedings of the International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 2, pp. 963–966.
80. Wang, Y.; Rucker, J.T.; Fowler, J.E. Three-dimensional tarp coding for the compression of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2004**, *1*, 136–140. [[CrossRef](#)]
81. Moazami-Goudarzi, M.; Moradi, M.H.; Abbasabadi, S. High performance method for electrocardiogram compression using two dimensional multiwavelet transform. In Proceedings of the IEEE Workshop on Multimedia Signal Processing, Shanghai, China, 30 October–2 November 2005; pp. 1–5.
82. Huang, B.; Sriraja, Y.; Huang, H.L.; Goldberg, M. Lossless multiwavelet compression of ultraspectral sounder data. In Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing, Seoul, Republic of Korea, 29 July 2006; pp. 3541–3544.
83. Cheung, K.W.; Cheung, C.H.; Po, L.M. A novel multiwavelet-based integer transform for lossless image coding. In Proceedings of the Proceedings International Conference on Image Processing, Kobe, Japan, 24–28 October 1999; Volume 1, pp. 444–447.
84. Amrani, N.; Serra-Sagrìstà, J.; Laparra, V.; Marcellin, M.W.; Malo, J. Regression wavelet analysis for lossless coding of remote-sensing data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5616–5627. [[CrossRef](#)]
85. Álvarez-Cortés, S.; Amrani, N.; Serra-Sagrìstà, J. Low complexity regression wavelet analysis variants for hyperspectral data lossless compression. *Int. J. Remote Sens.* **2018**, *39*, 1971–2000. [[CrossRef](#)]
86. Álvarez-Cortés, S.; Amrani, N.; Hernández-Cabronero, M.; Serra-Sagrìstà, J. Progressive lossy-to-lossless coding of hyperspectral images through regression wavelet analysis. *Int. J. Remote Sens.* **2018**, *39*, 2001–2021. [[CrossRef](#)]
87. Bajpai, S. 3D-listless block cube set-partitioning coding for resource constraint hyperspectral image sensors. *Signal Image Video Process.* **2024**, *18*, 3163–3178. [[CrossRef](#)]
88. ISO/IEC 15444-2; Information Technology—JPEG 2000 Image Coding System—Part 2: Extensions. International Organization for Standardization: Geneva, Switzerland, 2000.
89. Rucker, J.T.; Fowler, J.E.; Younan, N.H. JPEG2000 coding strategies for hyperspectral data. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Seoul, Republic of Korea, 29 July 2005; Volume 1, p. 4.
90. Schelkens, P.; Munteanu, A.; Tzannes, A.; Brislawn, C. JPEG2000. Part 10. Volumetric data encoding. In Proceedings of the IEEE International Symposium on Circuits and Systems, Kos, Greece, 21–24 May 2006; pp. 4–3877.
91. Skog, K.; Kohout, T.; Kašpárek, T.; Penttilä, A.; Wolfmayr, M.; Praks, J. Lossless hyperspectral image compression in comet interceptor and hera missions with restricted bandwidth. *Remote Sens.* **2025**, *17*, 899. [[CrossRef](#)]
92. Zhang, J.; Fowler, J.E.; Younan, N.H.; Liu, G. Evaluation of JP3D for lossy and lossless compression of hyperspectral imagery. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Cape Town, South Africa, 12–17 July 2009; Volume 4, pp. 474–477.
93. ISO/IEC 15948; Information Technology—Computer Graphics and Image Processing—Portable Network Graphics (PNG): Functional specification. International Organization for Standardization: Geneva, Switzerland, 2004.

94. ISO/IEC 14495-1; Information Technology—Lossless and Near-Lossless Compression of Continuous-Tone Still Images: Baseline and Extensions. International Organization for Standardization: Geneva, Switzerland, 1994.
95. McNeely, J.; Geiger, G. K-Means Based Spatial Aggregation for Hyperspectral Compression. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 26–28 March 2014; p. 416.
96. Hunt, S.; Rodriguez, L. Fast piecewise linear predictors for lossless compression of hyperspectral imagery. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AK, USA, 20–24 September 2004; Volume 1, p. 312.
97. Bai, L.; He, M.; Dai, Y. Lossless Compression of Hyperspectral Images Based on 3D Context Prediction. In Proceedings of the IEEE 3rd Conference on Industrial Electronics and Applications, Singapore, 3–5 June 2008; pp. 1845–1848.
98. Serra-Sagristà, J.; García-Vílchez, F.; Minguillón, J.; Megías, D.; Huang, B.; Ahuja, A. Wavelet lossless compression of ultraspectral sounder data. In Proceedings of the Proceedings. IEEE International Geoscience and Remote Sensing Symposium, Seoul, Republic of Korea, 29 July 2005; Volume 1, p. 4.
99. Mielikainen, J.; Kaarna, A. Improved back end for integer PCA and wavelet transforms for lossless compression of multispectral images. In Proceedings of the International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; Volume 2, pp. 257–260.
100. Motta, G.; Rizzo, F.; Storer, J. Partitioned vector quantization: Application to lossless compression of hyperspectral images. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Baltimore, MD, USA, 6–9 July 2003; Volume 3, p. 241.
101. Linde, Y.; Buzo, A.; Gray, R. An Algorithm for Vector Quantizer Design. *IEEE Trans. Commun.* **1980**, *28*, 84–95. [\[CrossRef\]](#)
102. Ryan, M.; Arnold, J. The lossless compression of AVIRIS images by vector quantization. *IEEE Trans. Geosci. Remote Sens.* **1997**, *35*, 546–550. [\[CrossRef\]](#)
103. Gupta, S.; Gersho, A. Feature predictive vector quantization of multispectral images. *IEEE Trans. Geosci. Remote Sens.* **1992**, *30*, 491–501. [\[CrossRef\]](#)
104. Huang, B.; Ahuja, A.; Huang, H.L. Fast precomputed VQ with optimal bit allocation for lossless compression of ultraspectral sounder data. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 29–31 March 2005; pp. 408–417.
105. Fox, B. Discrete optimization via marginal analysis. *Manag. Sci.* **1966**, *13*, 210–216. [\[CrossRef\]](#)
106. Baker, R.; Gray, R. Image compression using non-adaptive spatial vector quantization. In Proceedings of the Asilomar Conference on Circuits Systems and Computers, Pacific Grove, CA, USA, 7–10 November 1982; pp. 55–61.
107. Ramamurthi, B.; Gersho, A. Image vector quantization with a perceptually-based cell classifier. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, San Diego, CA, USA, 19–21 March 1984; Volume 9, pp. 698–701.
108. Mielikäinen, J.; Toivanen, P. Improved vector quantization for lossless compression of AVIRIS images. In Proceedings of the European Signal Processing Conference, Toulouse, France, 3–6 September 2002; pp. 1–3.
109. Blanes, I.; Serra-Sagristà, J. Clustered Reversible-KLT for Progressive Lossy-to-Lossless 3d Image Coding. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 16–18 March 2009; pp. 233–242.
110. Le, N.T.; Bui, C.V. Optimized multispectral satellite image compression using wavelet transforms. In *Proceedings of the Advances in Data Science and Optimization of Complex Systems: Proceedings of the International Conference on Applied Mathematics and Computer Science—ICAMCS 2024*; Springer Nature: Berlin/Heidelberg, Germany, 2025; Volume 2, p. 110.
111. Ahanonu, E.; Marcellin, M.; Bilgin, A. Clustering regression wavelet analysis for lossless compression of hyperspectral imagery. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019.
112. Karaca, A.C.; Güllü, M.K. Superpixel Based Recursive Least-squares Method for Lossless Compression of Hyperspectral Images. *Multidimens. Syst. Signal Process.* **2019**, *30*, 903–919. [\[CrossRef\]](#)
113. Zhu, F.; Hu, H. Lossless Compression for Hyperspectral Images Using Cascaded Prediction. In Proceedings of the International Conference on Communication, Image and Signal Processing, Chengdu, China, 17–19 November 2023; IEEE: New York, NY, USA, 2023; pp. 265–269.
114. Huang, B.; Sriraja, Y. Lossless compression of hyperspectral imagery via lookup tables with predictor selection. In Proceedings of the Image and Signal Processing for Remote Sensing XII. International Society for Optics and Photonics, Stockholm, Sweden, 29 September 2006; Volume 6365, p. 63650L.
115. Acevedo, D.; Ruedin, A. Lossless compression of hyperspectral images: Look-up tables with varying degrees of confidence. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010; pp. 1314–1317.
116. Gao, Z.C.; Zhang, X.L. Lossless compression of hyperspectral images using improved Locally Averaged Interband Scaling Lookup Tables. In Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, Guilin, China, 10–13 July 2011; pp. 91–96.

117. Aiazzi, B.; Baronti, S.; Alparone, L. Lossless Compression of Hyperspectral Imagery Via Lookup Tables and Classified Linear Spectral Prediction. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Boston, MA, USA, 7–11 July 2008; Volume 2, pp. 978–981.
118. Aiazzi, B.; Baronti, S.; Alparone, L. Lossless Compression of Hyperspectral Images Using Multiband Lookup Tables. *IEEE Signal Process. Lett.* **2009**, *16*, 481–484. [[CrossRef](#)]
119. Mielikainen, J.; Toivanen, P. Lossless Compression of Hyperspectral Images Using a Quantized Index to Lookup Tables. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 474–478. [[CrossRef](#)]
120. Kiely, A.B.; Klimesh, M.A. Exploiting Calibration-Induced Artifacts in Lossless Compression of Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 2672–2678. [[CrossRef](#)]
121. Weinberger, M.; Seroussi, G.; Sapiro, G. LOCO-I: A low complexity, context-based, lossless image compression algorithm. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 31 March–3 April 1996; pp. 140–149.
122. Weinberger, M.; Seroussi, G.; Sapiro, G. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. Image Process.* **2000**, *9*, 1309–1324. [[CrossRef](#)]
123. Joshi, V.; Rani, J.S. An efficient fpga implementation of a simple lossless algorithm (SLA) for on-board satellite hyperspectral data compression. In Proceedings of the International Symposium on Circuits and Systems, Singapore, 19–22 May 2024; IEEE: New York, NY, USA, 2024; pp. 1–5.
124. Joshi, V.; Rani, J.S. A simple lossless algorithm for on-board satellite hyperspectral data compression. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [[CrossRef](#)]
125. Li, C.; Guo, K. Lossless compression of hyperspectral images using interband gradient adjusted prediction. In Proceedings of the IEEE 4th International Conference on Software Engineering and Service Science, Beijing, China, 23–25 May 2013; pp. 724–727.
126. Altamimi, A.; Ben Youssef, B. Leveraging seed generation for efficient hardware acceleration of lossless compression of remotely sensed hyperspectral images. *Electronics* **2024**, *13*, 2164. [[CrossRef](#)]
127. ISO/IEC 10918-1; Information Technology—Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines. International Organization for Standardization: Geneva, Switzerland, 1994.
128. Slyz, M.; Zhang, L. A block-based inter-band lossless hyperspectral image compressor. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 29–31 March 2005; pp. 427–436.
129. Wu, X.; Memon, N. Context-based lossless interband compression-extending CALIC. *IEEE Trans. Image Process.* **2000**, *9*, 994–1001.
130. Karaca, A.C.; Gullu, M.K. Lossless compression of ultraspectral sounder data using recursive least squares. In Proceedings of the International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, 19–22 June 2017; pp. 109–112.
131. Lin, C.C.; Hwang, Y.T. An Efficient Lossless Compression Scheme for Hyperspectral Images Using Two-Stage Prediction. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 558–562. [[CrossRef](#)]
132. Roy, J.; Potvin, S.; Deschenes, J.D.; Genest, J. Lossless compression of hyperspectral data obtained from Fourier-transform infrared imaging spectrometers. In Proceedings of the Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Reykjavik, Iceland, 14–16 June 2010; pp. 1–4.
133. Klimesh, M. *Low-Complexity Lossless Compression of Hyperspectral Imagery via Adaptive Filtering*; NASA: Washington, DC, USA, 2005.
134. Fernandez i Ubiergo, G. Lossless region-based multispectral image compression. In Proceedings of the International Conference on Image Processing and Its Applications, Lyon, France, 9–11 January 1997; Volume 1, pp. 64–68.
135. Lee, J. Least squares approach for predictive coding of 3-D images. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 19 September 1996; Volume 2, pp. 875–878.
136. Chen, Y.; Shi, Z.; Li, D. Lossless Compression of Hyperspectral Image Based on 3DLMS Prediction. In Proceedings of the International Congress on Image and Signal Processing, Tianjin, China, 17–19 October 2009; pp. 1–6.
137. Klimesh, M. Low-complexity adaptive lossless compression of hyperspectral imagery. In Proceedings of the Satellite Data Compression, Communications, and Archiving II. International Society for Optics and Photonics, San Diego, CA, USA, 1 September 2006; Volume 6300, p. 63000N.
138. Aiazzi, B.; Alba, P.; Alparone, L.; Baronti, S. Lossless compression of multi/hyper-spectral imagery based on a 3-D fuzzy prediction. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 2287–2294. [[CrossRef](#)]
139. Aiazzi, B.; Alparone, L.; Baronti, S. Quality Issues for Compression of Hyperspectral Imagery Through Spectrally Adaptive DPCM. In *Satellite Data Compression*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 115–147.
140. Aiazzi, B.; Baronti, S.; Lastri, C.; Santurri, L.; Alparone, L. Low-complexity lossless/near-lossless compression of hyperspectral imagery through classified linear spectral prediction. In Proceedings of the Proceedings. IEEE International Geoscience and Remote Sensing Symposium, Seoul, Republic of Korea, 29 July 2005; Volume 1, pp. 132–135.
141. Aiazzi, B.; Alparone, L.; Baronti, S.; Lastri, C. Crisp and Fuzzy Adaptive Spectral Predictions for Lossless and Near-Lossless Compression of Hyperspectral Imagery. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 532–536. [[CrossRef](#)]

142. Rizzo, F.; Carpentieri, B.; Motta, G.; Storer, J. Low-complexity lossless compression of hyperspectral imagery via linear prediction. *IEEE Signal Process. Lett.* **2005**, *12*, 138–141. [\[CrossRef\]](#)
143. Shen, H.; David Pan, W. Predictive lossless compression of regions of interest in hyperspectral image via Maximum Correntropy Criterion based Least Mean Square learning. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016; pp. 2182–2186.
144. Gao, F.; Guo, S. Lossless compression of hyperspectral images using conventional recursive least-squares predictor with adaptive prediction bands. *J. Appl. Remote Sens.* **2016**, *10*, 015010. [\[CrossRef\]](#)
145. Li, C.; Zhu, F. Novel lossless compression method for hyperspectral images based on variable forgetting factor recursive least squares. *J. Inf. Process. Syst.* **2024**, *20*, 663–674.
146. Karaca, A.C.; Güllü, M.K. Lossless hyperspectral image compression using bimodal conventional recursive least-squares. *Remote Sens. Lett.* **2018**, *9*, 31–40. [\[CrossRef\]](#)
147. Song, J.; Zhou, L.; Deng, C.; An, J. Lossless compression of hyperspectral imagery using a fast adaptive-length-prediction RLS filter. *Remote Sens. Lett.* **2019**, *10*, 401–410. [\[CrossRef\]](#)
148. Huo, C.; Zhang, R.; Peng, T. Lossless Compression of Hyperspectral Images Based on Searching Optimal Multibands for Prediction. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 339–343.
149. Zhang, J.; Liu, G. A Novel Lossless Compression for Hyperspectral Images by Adaptive Classified Arithmetic Coding in Wavelet Domain. In Proceedings of the International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 2269–2272.
150. Zhang, J.; Liu, G. A Novel Lossless Compression for Hyperspectral Images by Context-Based Adaptive Classified Arithmetic Coding in Wavelet Domain. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 461–465. [\[CrossRef\]](#)
151. Shen, H.; Jiang, Z.; Pan, W.D. Efficient lossless compression of multitemporal hyperspectral image data. *J. Imaging* **2018**, *4*, 142. [\[CrossRef\]](#)
152. Li, C.; Guo, K. Lossless compression of hyperspectral images using three-stage prediction. In Proceedings of the IEEE International Conference on Software Engineering and Service Science, Beijing, China, 23–25 May 2013; pp. 1029–1032.
153. Jain, S.K.; Adjero, D.A. Edge-Based Prediction for Lossless Compression of Hyperspectral Images. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 27–29 March 2007; pp. 153–162.
154. Ni, G.; Fan, B.; Li, H. Onboard Lossless Compression of Hyperspectral Imagery Based on Hybrid Prediction. In Proceedings of the Asia-Pacific Conference on Information Processing, Shenzhen, China, 18–19 July 2009; Volume 2, pp. 164–167.
155. Zhang, J.; Liu, G. An Efficient Reordering Prediction-Based Lossless Compression Algorithm for Hyperspectral Images. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 283–287. [\[CrossRef\]](#)
156. Wang, H.; Babacan, S.D.; Sayood, K. Lossless Hyperspectral-Image Compression Using Context-Based Conditional Average. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4187–4193. [\[CrossRef\]](#)
157. Lin, C.C.; Hwang, Y.T. Lossless Compression of Hyperspectral Images Using Adaptive Prediction and Backward Search Schemes. *J. Inf. Sci. Eng.* **2011**, *27*, 419–435.
158. Chen, Y.-h.; Shi, Z.-l.; Ma, L. Lossless compression of hyperspectral image based on spatial-spectral hybrid prediction. In Proceedings of the International Conference on Signal Processing, Beijing, China, 26–29 October 2008; pp. 993–997.
159. Chai, Y.; Zhang, X.-l.; Shen, L.-s. Lossless compression of hyperspectral imagery through 2D/3D hybrid prediction. In Proceedings of the IEEE International Symposium on Communications and Information Technology, Beijing, China, 12–14 October 2005; Volume 2, pp. 1456–1459.
160. Magli, E. Multiband Lossless Compression of Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 1168–1178. [\[CrossRef\]](#)
161. Hernandez-Cabronero, M.; Kiely, A.B.; Klimesh, M.; Blanes, I.; Ligo, J.; Magli, E.; Serra-Sagrista, J. The CCSDS 123.0-B-2 “Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression” Standard: A comprehensive review. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 102–119. [\[CrossRef\]](#)
162. Vorhaug, D.; Boyle, S.; Orlandić, M. High-level CCSDS 123.0-B-2 hyperspectral image compressor verification model. In Proceedings of the Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing, Helsinki, Finland, 9–11 December 2024; IEEE: New York, NY, USA, 2024; pp. 1–5.
163. Nonnis, A.; Grangetto, M.; Magli, E.; Olmo, G.; Barni, M. Improved low-complexity intraband lossless compression of hyperspectral images by means of Slepian-Wolf coding. In Proceedings of the IEEE International Conference on Image Processing 2005, Genova, Italy, 14 September 2005; Volume 1, p. I-829.
164. Gong, Y.; Yan, X.; Wu, J. Hyperspectral image lossless compression using DSC and 2-D CALIC. In Proceedings of the International Conference on Computer and Communication Technologies in Agriculture Engineering, Chengdu, China, 12–13 June 2010; Volume 3, pp. 460–463.
165. Slepian, D.; Wolf, J. Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory* **1973**, *19*, 471–480. [\[CrossRef\]](#)

166. Magli, E.; Olmo, G.; Quacchio, E. Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC. *IEEE Geosci. Remote Sens. Lett.* **2004**, *1*, 21–25. [\[CrossRef\]](#)
167. Afjal, M.I.; Mamun, M.A.; Uddin, M.P. Weighted-Correlation based Band Reordering Heuristics for Lossless Compression of Remote Sensing Hyperspectral Sounder Data. In Proceedings of the International Conference on Advancement in Electrical and Electronic Engineering, Gazipur, Bangladesh, 22–24 November 2018; pp. 1–4.
168. Afjal, M.I.; Uddin, P.; Mamun, A.; Marjan, A. An efficient lossless compression technique for remote sensing images using segmentation based band reordering heuristics. *Int. J. Remote Sens.* **2021**, *42*, 756–781. [\[CrossRef\]](#)
169. Edmonds, J. Optimum branchings. *Math. Decis. Sci. Part* **1968**, *1*, 25. [\[CrossRef\]](#)
170. Tate, S. Band ordering in lossless compression of multispectral images. *IEEE Trans. Comput.* **1997**, *46*, 477–483. [\[CrossRef\]](#)
171. Kubasova, O.; Toivanen, P. Lossless compression methods for hyperspectral images. In Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004; Volume 2, pp. 803–806.
172. Toivanen, P.; Kubasova, O.; Mielikainen, J. Correlation-based band-ordering heuristic for lossless compression of hyperspectral sounder data. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 50–54. [\[CrossRef\]](#)
173. Prim, R.C. Shortest connection networks and some generalizations. *Bell Syst. Tech. J.* **1957**, *36*, 1389–1401. [\[CrossRef\]](#)
174. Gao, X.; Wang, L.; Li, T.; Xie, J. A Method of Reordering Lossless Compression of Hyperspectral Images. *Isprs Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *10*, 821–826. [\[CrossRef\]](#)
175. Afjal, M.I.; Al Mamun, M.; Uddin, M.P. Band reordering heuristics for lossless satellite image compression with 3D-CALIC and CCSDS. *J. Vis. Commun. Image Represent.* **2019**, *59*, 514–526. [\[CrossRef\]](#)
176. Wang, L.; Zhang, T.; Fu, Y.; Huang, H. Hyperreconnet: Joint coded aperture optimization and image reconstruction for compressive hyperspectral imaging. *IEEE Trans. Image Process.* **2018**, *28*, 2257–2270. [\[CrossRef\]](#)
177. Haut, J.M.; Gallardo, J.A.; Paoletti, M.E.; Cavallaro, G.; Plaza, J.; Plaza, A.; Riedel, M. Cloud deep networks for hyperspectral image analysis. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9832–9848. [\[CrossRef\]](#)
178. Choi, I.; Kim, M.; Gutierrez, D.; Jeon, D.; Nam, G. High-quality hyperspectral reconstruction using a spectral prior. *Acm Trans. Graph.* **2017**, *36*, 218. [\[CrossRef\]](#)
179. Afrin, A.; Haque, M.R.; Al Mamun, M. Enhancing hyperspectral image compression through stacked autoencoder approach. In Proceedings of the International Conference on Electrical Engineering and Information & Communication Technology, Dhaka, Bangladesh, 2–4 May 2024; IEEE: New York, NY, USA, 2024; pp. 1372–1377.
180. Li, J.; Liu, Z. Multispectral transforms using convolution neural networks for remote sensing multispectral image compression. *Remote Sens.* **2019**, *11*, 759. [\[CrossRef\]](#)
181. Sheikh, J.; Gross, W.; Michel, A.; Weinmann, M.; Kuester, J. Transformer-based lossy hyperspectral satellite data compression. In Proceedings of the Earth Resources and Environmental Remote Sensing/GIS Applications XVI, Madrid, Spain, 28 October 2025; SPIE: Bellingham, WA, USA, 2025; Volume 13671, pp. 217–226.
182. Zhang, L.; Zhang, L.; Song, C.; Zhang, P. Hyperspectral image compression sensing network with CNN–Transformer mixture architectures. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 5506305. [\[CrossRef\]](#)
183. Fuchs, M.H.P.; Rasti, B.; Demir, B. HyCoT: A Transformer-Based Autoencoder for Hyperspectral Image Compression. In Proceedings of the 2024 14th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Helsinki, Finland, 9–11 December 2024; IEEE: New York, NY, USA, 2024; pp. 1–5.
184. Deng, C.; Cen, Y.; Zhang, L. Learning-based hyperspectral imagery compression through generative neural networks. *Remote Sens.* **2020**, *12*, 3657. [\[CrossRef\]](#)
185. Guo, Y.; Li, W.; Peng, Q.; Tu, L. Spectral constrained generative adversarial network for hyperspectral compression. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 23372–23386. [\[CrossRef\]](#)
186. Guo, Y.; Chong, Y.; Pan, S. Hyperspectral image compression via cross-channel contrastive learning. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5513918. [\[CrossRef\]](#)
187. Byju, A.P.; Fuchs, M.H.P.; Walda, A.; Demir, B. Generative Adversarial Networks for Spatio-Spectral Compression of Hyperspectral Images. In Proceedings of the 2024 14th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Helsinki, Finland, 9–11 December 2024.
188. Liu, J.; Zhang, L.; Wang, J.; Qu, L. Bi-residual compression network with conditional diffusion model for hyperspectral image compression. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5521015.
189. Feng, X.; Gu, E.; Zhang, Y.; Li, A. Probability prediction network with checkerboard prior for lossless remote sensing image compression. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 17971–17982. [\[CrossRef\]](#)
190. Gu, E.; Zhang, Y.; Wang, X.; Jiang, X. Lossless compression framework using lossy prior for high-resolution remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 8590–8601. [\[CrossRef\]](#)
191. Luo, J.; Wu, J.; Zhao, S.; Wang, L.; Xu, T. Lossless compression for hyperspectral image using deep recurrent neural networks. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2619–2629. [\[CrossRef\]](#)

192. Valsesia, D.; Bianchi, T.; Magli, E. Hybrid recurrent-attentive neural network for onboard predictive hyperspectral image compression. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024; IEEE: New York, NY, USA, 2024; pp. 7898–7902.
193. Valsesia, D.; Bianchi, T.; Magli, E. Onboard deep lossless and near-lossless predictive coding of hyperspectral images with line-based attention. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5532714. [[CrossRef](#)]
194. Cao, X.; Li, J.; Li, Y.; Yan, J.; Zheng, Z. Transformer-based lossless compression of aurora spectral images: A spatial-temporal-spectral joint approach. In Proceedings of the International Conference on Signal Processing, Suzhou, China, 28–31 October 2024; IEEE: New York, NY, USA, 2024; pp. 418–423.
195. Valsesia, D.; Bianchi, T.; Magli, E. Onboard hyperspectral image compression with deep line-based predictive architectures. In Proceedings of the 9th International Workshop on On-Board Payload Data Compression, Gran Canaria, Spain, 2–4 October 2024; ESA: Paris, France, 2024.
196. Jiang, Z.; Pan, W.D.; Shen, H. LSTM Based Adaptive Filtering for Reduced Prediction Errors of Hyperspectral Images. In Proceedings of the IEEE International Conference on Wireless for Space and Extreme Environments, Huntsville, AL, USA, 11–13 December 2018; pp. 158–162.
197. Jiang, Z.; Pan, W.D.; Shen, H. Spatially and Spectrally Concatenated Neural Networks for Efficient Lossless Compression of Hyperspectral Imagery. *J. Imaging* **2020**, *6*, 38. [[CrossRef](#)]
198. Anuradha, D.; Sekhar, G.C.; Mishra, A.; Thapar, P.; Baker El-Ebiary, Y.A.; Syamala, M. Efficient compression for remote sensing: Multispectral transform and deep recurrent neural networks for lossless hyper-spectral image. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 531. [[CrossRef](#)]
199. Graña, M.; Veganzons, M.A.; Ayerdi, B. Hyperspectral Remote Sensing Scenes. Available online: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 3 December 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.