MDPI

Article

Recursive Least Squares for Near-Lossless Hyperspectral Data Compression

Tie Zheng [1,2], Yuqi Dai [1,2], Changbin Xue [1,*] and Li Zhou [1]

1   National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China;
    zhengtie17@mails.ucas.ac.cn (T.Z.); daiyuqi18@mails.ucas.ac.cn (Y.D.); zhouli@nssc.ac.cn (L.Z.)
2   Department of Computer Science and Technology, University of Chinese Academy of Sciences,
    Beijing 100049, China
*   Correspondence: xuechangbin@nssc.ac.cn

Abstract: The hyperspectral image compression scheme is a trade-off between the limited hardware resources of the on-board platform and the ever-growing resolution of the optical instruments. Predictive coding attracts researchers due to its low computational complexity and moderate memory requirements. We propose a near-lossless prediction-based compression scheme that removes spatial and spectral redundant information, thereby significantly reducing the size of hyperspectral images. This scheme predicts the target pixel's value via a linear combination of previous pixels. The weight matrix of the predictor is iteratively updated using a recursive least squares filter with a loop quantizer. The optimal number of bands for prediction was analyzed experimentally. The results indicate that the proposed scheme outperforms state-of-the-art compression methods in terms of the compression ratio and quality retrieval.

Keywords: near-lossless compression; recursive least squares; hyperspectral image; predictive coding

1. Introduction

Hyperspectral images from space-borne spectrometers play a crucial role in multifarious aspects, including geological exploration, environmental monitoring, and material identification [1]. Researchers continue to enhance the spectral and spatial resolutions of the instruments, and the size of a hyperspectral image is currently more than hundreds of megabytes (MBs) [2–4]. Nevertheless, such a wealth of information places excessive demands on the transmission and storage processes. Data compression has proven to be an effective way to alleviate this issue [5,6].

The compression techniques for hyperspectral images are divided into three categories: lossless, lossy, and near-lossless compression [7]. Lossless compression allows for reconstruction of the original image, ideally at the price of a limited compression ratio. Lossy compression approximates the original image while generally minimizing distortion in the l_2-norm. It tolerates a small amount of information distortion between the original image I and the reconstruction image $\hat{I}$, which allows for a high compression ratio. Near-lossless compression aims to achieve a higher compression ratio than lossless techniques by allowing pixel-level distortion. It strictly bounds the l_∞-norm by setting the peak absolute error (PAE) [8]. The user-specified parameter guarantees the max. absolute distortion so that $PAE \leq \Lambda$ defines a limited error range for reconstruction of individual pixels. As a result, employing a proper parameter Λ makes the compression process almost lossless. It is a well-known fact that the quality of images is affected by the inherent noise of the device [9,10]. When the maximum error introduced by the compression process is smaller than the background noise, the quality of the reconstructed image is almost similar to that obtained with lossless compression.

Details on near-lossless compression techniques are discussed in a later section. Most near-lossless compression techniques can be roughly classified into three categories:

prediction-based subsequent quantization coding, lossless coding based on pre-quantization, and two-stage near-lossless encoding [11].

Predictive-based coding, one of the most popular schemes, enables low-complexity, high-throughput solutions [12–14]. These schemes first compute the prediction value of the target pixel from the previous encoding. The difference between the predicted and the original pixel value is known as the prediction error. Subsequently, a near-lossless compression scheme is obtained by encoding the quantized prediction error. The two most typical prediction schemes are JPEG-LS [15] and CALIC [16], which are widely used to process two-dimensional images. Meanwhile, the CALIC algorithm, with its better compression performance, is extended to 3D-CALIC [17] and M-CALIC [18] based on the correlation of hyperspectral images. However, the CALIC-based extension scheme cannot effectively remove the redundancy of hyperspectral images and is not friendly to hardware implementation. The Consultative Committee for Space Data Systems (CCSDS) proposes the standard CCSDS-123, which is based on the signed least mean square (SLMS) filter [19,20]. This compression scheme has low complexity and excellent compression results. NL-CCSDS-123 [10] and CCSDS-123-AC [21] are two near-lossless extended versions that both rely on the predictor of CCSDS-123. The NL-CCSDS-123 scheme encodes the quantized residual pixels after using a range coder, whereas CCSDS-123-AC employs a lightweight context-based arithmetic encoder. However, since their predictor uses a simple function to update the weight coefficients, the prediction accuracy can be further improved.

The second type of near-lossless compression category is based on pre-quantizing the original pixels with a quantizer and then applying a lossless compression technique. It is widely known that such a model is suboptimal, and the compression results are not outstanding [22]. However, it is suitable for a scene with high-speed compression demands since it does not need to include a feedback loop. S.-C Tai et al. proposed the Pre-CCSDS-IDC [23] compression scheme in order to improve the compression rate without modifying the existing CCSDS-IDC hardware system [24]. It can import the pre-quantized images into CCSDS-IDC directly. In [25] a pre-quantization compression scheme is implemented based on ground-based CNN reconstruction. The spaceborne part can be considered a combination of the pre-quantizer and CCSDS-123 predictor. Eventually, the CNN is employed as a feature extractor on the ground to do a secondary reconstruction of the decoded image, which leads to a higher signal-to-noise ratio.

The third near-lossless compression category builds on a combination of lossy and lossless compression. First, the reconstructed images are obtained by lossy compression. Then, the differences between these and the original image are quantized and encoded. X. Wu proposes an approximate lossless image compression scheme that combines wavelets and CALIC [26]. It uses CALIC to compress the residual image between the wavelet approximation and the original image. C.-W Chen employed CCSDS-IDC for the lossy phase, followed by bit-plane encoding (BPE) coding. However, the scheme does not obtain the optimal lossy bit rate. J.Beerten combines JPEG2000, as a lossy layer, with a near-lossless layer consisting of BPE and arithmetic coding [27]. This scheme uses computationally expensive iterative methods to determine the optimal lossy bitrate and obtain a competitive coding performance.

Due to the characteristics of the prediction-based compression scheme, it gradually replaces the transform-based algorithm in the on-board compression platform [28]. In this paper, we focus on the prediction-based quantization technique, which aims to enhance the performance of the near-lossless compression scheme by fully exploiting the spatial–spectral redundancy of hyperspectral images. The proposed method combines recursive least squares (RLS) with an in-loop quantizer and, subsequently, encodes the quantization residuals using an entropy encoder. Ultimately, a competitive compression ratio is produced while guaranteeing the quality of the reconstructed image.

The rest of the paper is organized as follows: Section 2 provides a detailed description of the near-lossless feedback loop's compression framework with the RLS prediction.

Section 3 presents the experimental dataset and comparative results of this method with an analysis and discussion. Finally, Section 4 concludes this paper.

2. Compression Scheme

In this work, a loop predictor is used for near-accurate prediction of target pixels by removing the correlation of hyperspectral images. The input image data is passed through the loop predictor point by point. The differences between the predicted and original values are quantized. Subsequently, the entropy encoder encodes the quantized residual image and outputs the compressed codestream. The same algorithm with the same parameters must be used at the decoding stage during the reconstruction process after inverse entropy coding. Figure 1 presents a pictorial representation of the compression scheme, which explains the essential steps. The five significant steps of the proposed scheme, including spatial predictor, spectral predictor, controlled quantization, sample representative, and entropy coding, are discussed in the following subsections.
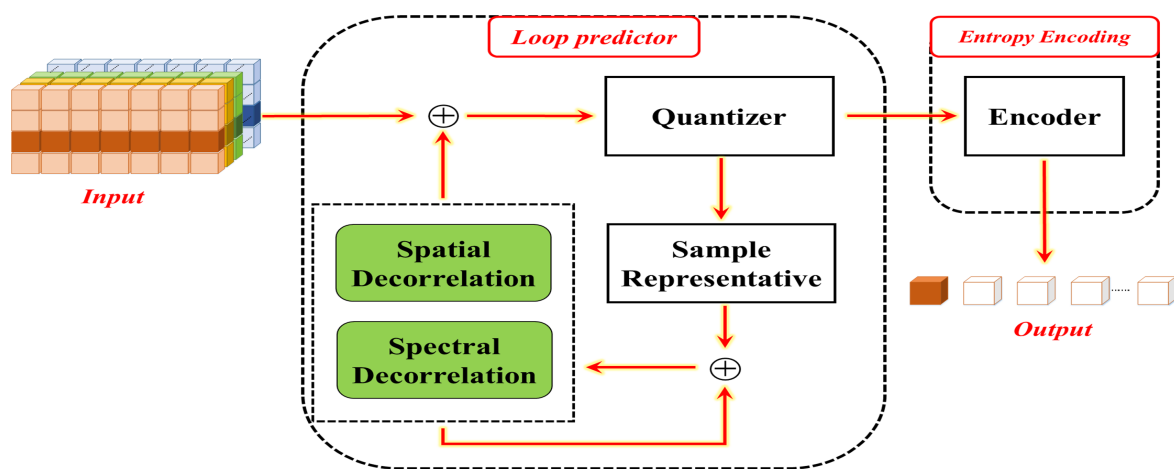

**Figure 1.** Compressor schematic.

2.1. Loop Predictor

Hyperspectral images can be regarded as a three-dimensional data. Let $s_z(x, y)$ denote the value of the original pixel in the zth band at the xth row with the yth column. We also use $s_z(n)$ to represent the nth pixel in band z, where the notation n means the pixel is derived from $n = NX * y + x$ in the spatial domain. NX, NY, and NZ provide the image's width, height, and number of bands.

The spatial prediction estimate is computed using a causal, linear predictor for each pixel. To remove spatial correlation from the hyperspectral image, the preliminary estimate $\hat{S}_z(n)$ of the target pixel $S_z(n)$ is generated by averaging the pixels in the context window of the same band. The context window of the target pixel can be explained using Figure 2. The parameter r denotes the local context window's radius, and the target pixel points are represented in orange. The blue and green parts show the neighborhood windows with radii of one and three, respectively.

Then, the spatial prediction error $d_z(n)$ for the nth pixel point in the zth band is defined as Equation (1).

$$d_z(n) = s_z(n) - \hat{S}_z(n). \tag{1}$$

Note that in order to ensure the feasibility of the decompressor, the preliminary prediction error $d_z(n)$ of the first spectrum is directly encoded with the entropy encoder that follows. The others are encoded after removing the inter-spectral redundancy.
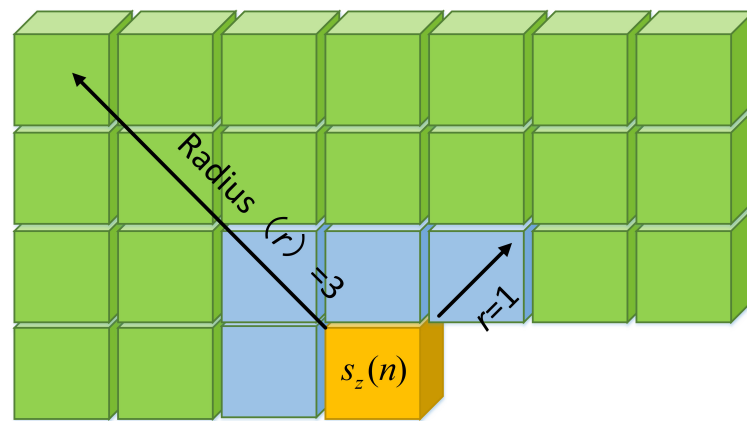
Figure 2. The context window of current pixel.

In the spectrum prediction process, the RLS filter input vector is formed by $\boldsymbol{d}_{z,k}(n) = [d_{z-k}(n), \cdots, d_{z-1}(n)]^T$, where k is the number of history bands used to predict the current pixel, i.e., the prediction length. The corresponding weight vector is $\boldsymbol{w}_{z,k}(n) = [w_{z-k}(n), \cdots, w_{z-1}(n)]^T$. Then, the RLS predictor is initialized as Equation (2).

$$\boldsymbol{k}_z(0) = 0, \boldsymbol{w}_z(0) = 0, \boldsymbol{p}_z(0) = 0. \tag{2}$$

The spectrum prediction residual $e_z(n)$ is calculated as Equation (3).

$$e_z(n) = d_z(n) - \boldsymbol{d}_{z,k}(n)\boldsymbol{w}_{z,k}^T(n-1). \tag{3}$$

The parameter's peak absolute error Λ determines the maximum allowable absolute difference between the original and reconstructed pixel values. Each predicted residual corresponds to a quantized residual $q_z(n)$ defined by Equation (4).

$$q_z(n) = sgn(e_z(n)) \times \left\lfloor \frac{\Lambda + |e_z(n)|}{2\Lambda + 1} \right\rfloor, \tag{4}$$

where the $sgn(x)$ is a function that extracts the sign value of x. For spectral images that need to be stored with absolute accuracy, a lossless compression mode can be used by setting $\Lambda = 0$. That is, $q_z(n) = e_z(n)$, thus ensuring that the decoding process can accurately reconstruct the original sample. However, when using a non-zero peak absolute error, $q_z(n)$ represents an approximation of the above prediction error, rather than the actual value.

Since the image introduces distortion affected by the quantizer, the compressed code stream cannot be directly employed to reconstruct the sample. In order to guarantee synchronization with the decompression stage, the reconstruction value $\widetilde{s}_z(n)$ should be calculated for each pixel, as shown in Equation (5). Then, $\widetilde{s}_z(n)$ is used in the prediction process for the next pixel point to be measured.

$$\widetilde{s}_z(n) = \hat{s}_z(n) + \boldsymbol{d}_{z,k}(n)\boldsymbol{w}_{z,k}(n-1) + \widetilde{e}_z(n). \tag{5}$$

where the center of the predicted residual reconstruction value $\widetilde{e}_z$ is calculated by Equation (6).

$$\widetilde{e}_z(n) = q_z(n)(2\Lambda + 1), \tag{6}$$

The gain of RLS is

$$\boldsymbol{k}_z^T(n) = \frac{\boldsymbol{p}_z(n-1)\boldsymbol{d}^T{}_z(n)}{1 + \boldsymbol{d}_z(n)\boldsymbol{p}(n-1)_z\boldsymbol{d}^T{}_z(n)}, \tag{7}$$

$$\boldsymbol{p}_z(n) = \boldsymbol{p}_z(n-1) - \boldsymbol{k}_z^T(n)\boldsymbol{d}_z(n)\boldsymbol{p}_z(n-1), \tag{8}$$

where $p(n)$ is an auxiliary vector required to reduce the computational burden.

Then, the weight vector $w_{z,k}(n)$ is updated by the recursive Equation 9.

$$w_{z,k}(n) = w_{z,k}(n-1) + k_z(n)\widetilde{e}_z(n). \tag{9}$$

Finally, the predictor is executed for each pixel in raster scan order until the last pixel is reached.

2.2. Entropy Encoding

Entropy coding techniques are used to encode the residual error after quantization in predictive compression. The adaptive arithmetic encoder is adopted in the encoding stage, whose compression ratio is near the theoretical entropy. The quantization residual is represented by 16 bits, meaning that the arithmetic code's codebook needs 65,536 symbols. Based on the probability distribution of the prediction residual, most symbols are not used. Therefore, an adaptive codebook is adopted. The initial codebook contains two symbols: 0 and ESC. When a new symbol needs to be encoded, the encoder will use the probability of ESC to encode it, and the 16-bit symbol will be appended to the code stream. The code book absorbs the new symbol after the symbol has been output to the stream.

3. Experimental Results

The proposed scheme, named near-lossless recursive least squares (NLRLS), has two parameters, namely, the radius of the context window r and the prediction length k. Since these two parameters critically impact the compression results, the selection of optimal values is explained in the first subsection. Furthermore, the compression performance of the proposal is compared with several state-of-the-art schemes, including compression results in the distortion metrics of the reconstructed image. This section highlights the corresponding results and analysis.

The platform for testing is a personal computer powered by a single Intel Core i7-7700K central processing unit (CPU) at 4.2GHz with 16GB random access memory (RAM). We adopted the standard hyperspectral image test data recommended by the International Consultative Committee for Space Data Systems, which includes Atmospheric Infrared Sounder (AIRS) and Airborne Visible/Infrared Imaging Spectrometer Calibrated and Uncalibrated (AC and AU) [29]. Each of them has a bit depth of 16 bit-per-pixel (BPP). Table 1 details the dataset used in the tests, including sensor abbreviations, scene names, and dimensions.

**Table 1.** The sensor names and their main features.

Sensor	Scene	Number of Scence	Rows	Columns	Bands	Formation
AC	Yellowstone	5	677	512	224	Signed 16 bit
AU	Yellowstone	3	680	512	224	Unsigned 16 bit
AIRS	Gran	8	90	135	1501	Unsigned 16 bit

3.1. Parameter Settings

The accuracy of the RLS predictor has a strong correlation with the prediction length k of the input vector $d_{z,k}(n)$. Additionally, to preserve the causality of the pixels in the context window, the local mean predictor is used to estimate $\hat{S}_z(n)$. Therefore, the radius r of the context window is another important parameter that directly affects the compression results. In order to evaluate the effects of the k and r parameters on near-lossless compression performances, average bit rates of the proposed scheme at different peak absolute errors are shared in Figure 3 for the AC datasets.

It can be seen from Figure 3a–d that the proposed compression schemes exhibit similar radio-compressed bit-rate characteristics for different peak absolute errors. The lowest bit rates are observed in the case of low r–high k parameter pairs. If the radius is treated as

a constant, we can see that the deceleration of the compressed bit rate gradually slows down as the prediction length increases. Moreover, the RLS predictor update requires the computation of $\boldsymbol{P}(n)$ as well as $\boldsymbol{d}_{z,k}(n)$. The algorithm has an $O(k^2)$ computational complexity, meaning that the computing resource consumption increases quadratically with k. Considering the dual effects of the actual compression results and the computational complexity, the initial radio r and prediction length k parameters are selected as 2 and 12, respectively.
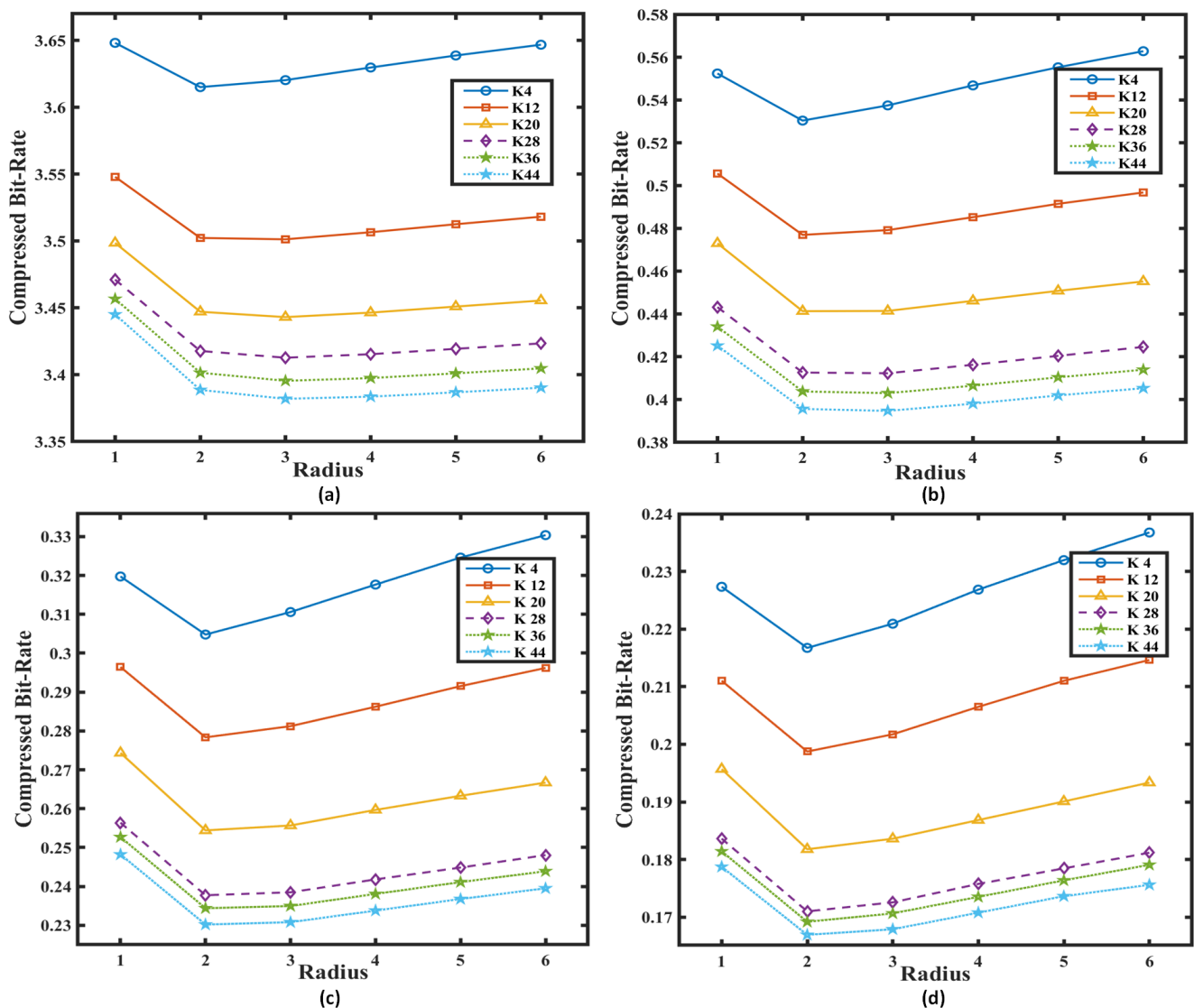
Figure 3. Average compressed bit-rate for different peak absolute errors: (**a**) encoded at $\Lambda = 0$, (**b**) encoded at $\Lambda = 10$, (**c**) encoded at $\Lambda = 20$, (**d**) encoded at $\Lambda = 30$.

3.2. Compression Performance Analysis

3.2.1. Compression Results

A total of 16 scene data from three types of hyperspectral image datasets (AC, AU, AIRS) were used as test data sources. The results of CCSDS-123-AC, NLCCSDS-123, and M-CALIC are listed here to compare with the proposed near-lossless compression scheme. Table 2 shows the average compression results for multiple compression schemes at five different peak absolute errors.

Table 2 reports compression results in terms of the bit rate. The second column reports the first-order entropy on average (in bit-per-pixel) for each scene type. It represents the

entropy of individual pixels, regardless of any correlation among the pixels. In addition, each subsequent column shows the average compression results for a compression scheme at different peak absolute errors Λ. The best compression results for each test are indicated in bold. The bracketed text indicates the coding gain of our method relative to other techniques, and a positive difference means our scheme is better.

Table 2. Comparison of different near-lossless schemes for several different peak absolute errors.

Sensor Abbreviation	Entropy	Λ Values	CCSDS-123-AC	NL-CCSDS-123	M-CALIC	NLRLS
AC	9.77	$\Lambda = 0$	3.66 (0.14)	3.73 (0.21)	4.03 (0.51)	**3.52**
		$\Lambda = 1$	2.45 (0.38)	2.54 (0.47)	2.87 (0.80)	**2.07**
		$\Lambda = 10$	0.58 (0.10)	0.94 (0.46)	0.88 (0.40)	**0.48**
		$\Lambda = 20$	0.35 (0.07)	0.72 (0.44)	0.53 (0.25)	**0.28**
		$\Lambda = 30$	0.26 (0.06)	0.63 (0.43)	0.40 (0.20)	**0.20**
AU	12.13	$\Lambda = 0$	5.87 (0.14)	5.95 (0.22)	6.13 (0.40)	**5.73**
		$\Lambda = 1$	4.80 (0.66)	4.89 (0.75)	5.05 (0.91)	**4.14**
		$\Lambda = 10$	1.96 (0.38)	2.24 (0.66)	2.22 (0.64)	**1.58**
		$\Lambda = 20$	1.14 (0.22)	1.46 (0.54)	1.40 (0.48)	**0.92**
		$\Lambda = 30$	0.83 (0.17)	1.18 (0.52)	1.05 (0.39)	**0.66**
AIRS	11.39	$\Lambda = 0$	4.25 (0.41)	4.31 (0.47)	4.38 (0.54)	**3.84**
		$\Lambda = 1$	3.08 (0.77)	3.13 (0.82)	3.21 (0.90)	**2.31**
		$\Lambda = 10$	0.61 (0.28)	0.98 (0.65)	0.70 (0.37)	**0.33**
		$\Lambda = 20$	0.29 (0.15)	0.66 (0.52)	0.36 (0.22)	**0.14**
		$\Lambda = 30$	0.20 (0.10)	0.57 (0.47)	0.26 (0.16)	**0.10**

It can be seen that the compression bit rate of each technique decreases rapidly as a function of Λ. The reported results indicate that our scheme outperforms the other known schemes for all sensors. For lossless coding ($\Lambda = 0$), our scheme beats the other schemes by a slight margin. For near-lossless encoding ($\Lambda > 0$), our scheme provides outstanding compression results; i.e., it clearly yields the lowest ratio of all the compared methods. Compared to CCSDS-123-AC, which has the best-known compression results, on average, the proposal provides benefits ranging from 0.11 to 0.6 bpp, depending on the allowed absolute error of the peak.

The previous subsection stated that the computational complexity of NLRLS is $O(k^2)$. The computational complexity of the SLMS filter used by CCSDS-123-AC is $O(k)$, where k is the number of bands. Although NLRLS has higher complexity than CCSDS-123-AC, it produces better convergence and more accurate predictions. Therefore, the compression results of the NLRLS scheme are better.

Graphically, the relative gain of our scheme over CCSDS-123-AC is represented in Figure 4 for different datasets. Each color represents the comparison results using different datasets, and each cluster indicates a different peak absolute error. Therefore, the higher percentage means that our scheme is superior. The results indicate that the proposed scheme provides an average optimization from 5.29% to 31.19% in bit-per-pixel change for different peak absolute errors. The AIRS images, especially, outperform by almost 50% with $\Lambda \geq 10$.

We note that complex image preprocessing can achieve better compression results, such as band reorder, clustering, and super-pixel methods [30,31]. However, this is beyond the scope of this paper and will not be discussed here.
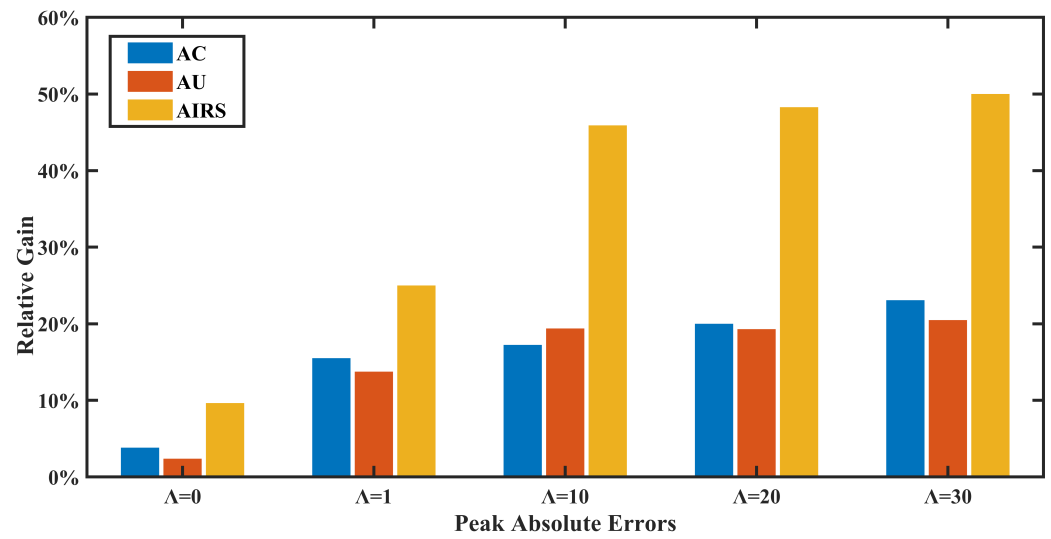
Figure 4. The relative gain of our scheme on CCSDS-123-AC at different peak absolute errors for AC, AU, and AIRS datasets.

3.2.2. Reconstruction Quality Analysis

In the near-lossless compression scheme, the peak absolute error Λ introduced by the compression process directly affects the quality of the reconstructed image. Figure 5 shows the original image of the AIRS Grand16 image and the reconstructed image at $\Lambda = \{0, 1, 10, 20, 30\}$. It can be noted that the visual performance of each image is extremely similar.
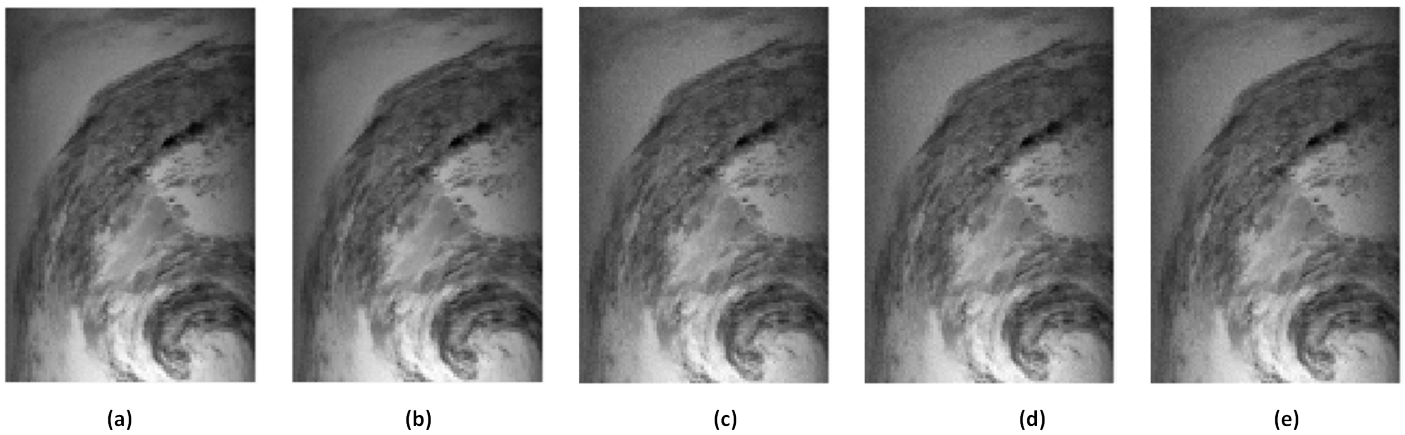
Figure 5. Visual comparison for the "AIRS Grand16 1256th band" image: (**a**) original, (**b**–**e**) reconstructed image at $\Lambda = \{1, 10, 20, 30\}$, respectively.

To objectively evaluate the quality of the reconstructed images, we compare the proposed scheme with the best-known near-lossless compression scheme CCSDS-123-AC, including two metrics, namely the peak signal-to-noise ratio and spectral angle mapper.

If D is the dynamic range (in bits) of the original image, the maximum pixel value is $2^D - 1$. The reconstructed image quality is evaluated in terms of the peak signal-to-noise ratio (PSNR), which is measured in dB and defined as:

$$PSNR\left(S, \widetilde{S}\right) = 10 \log_{10} \frac{\left(2^D - 1\right)^2}{MSE\left(S, \widetilde{S}\right)} (dB) , \qquad (10)$$

where $MSE\left(S, \widetilde{S}\right)$ is the mean square error between the original image S and the reconstructed image $\widetilde{S}$.

$$MSE\left(S, \widetilde{S}\right) = \frac{\sum_1^{NX} \sum_1^{NY} \sum_1^{NZ} \left(S_z(x, y) - \widetilde{S}_z(x, y)\right)^2}{NX \times NY \times NZ},$$ (11)

Spectral Angle Mapper (SAM) treats the spectrum of each image element as a high-dimensional vector and measures the similarity between two spectra by calculating the angle of the vector. The SAM of each pixel spectrum in the original and reconstructed image is denoted as

$$\alpha(x, y) = \cos^{-1} \left(\frac{\sum_Z^{NZ} S_z(x, y) \times \widetilde{S}_z(x, y)}{\sqrt{\sum_Z^{NZ} S_z^2(x, y)} \sqrt{\sum_Z^{NZ} \widetilde{S}_z^2(x, y)}} \right),$$ (12)

The average spectral angle is used to calculate the spectral variability of the reconstructed image, and, the smaller the angle, the less distortion.

We experimentally obtained a series of reconstructed image data by varying the maximum allowable peak absolute error. Figures 6 and 7 show the PSNR and the average SAM variation with the bit rate, respectively.
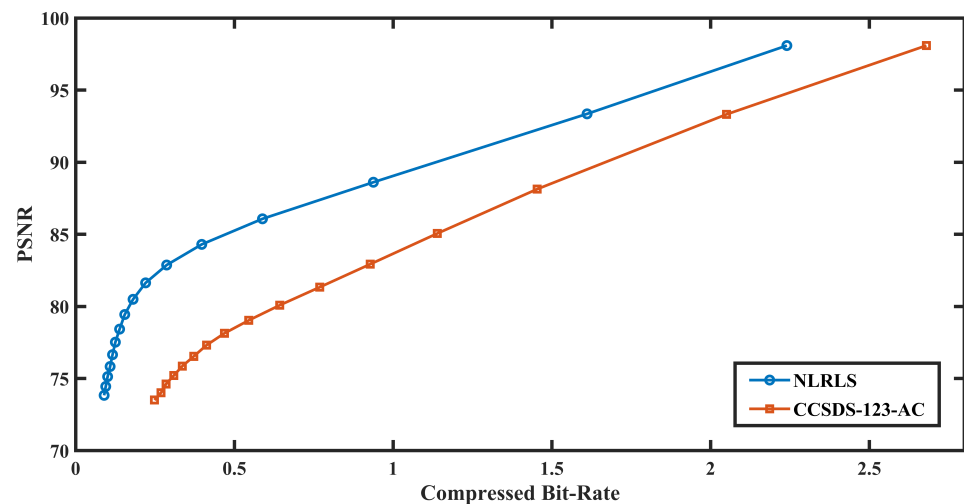

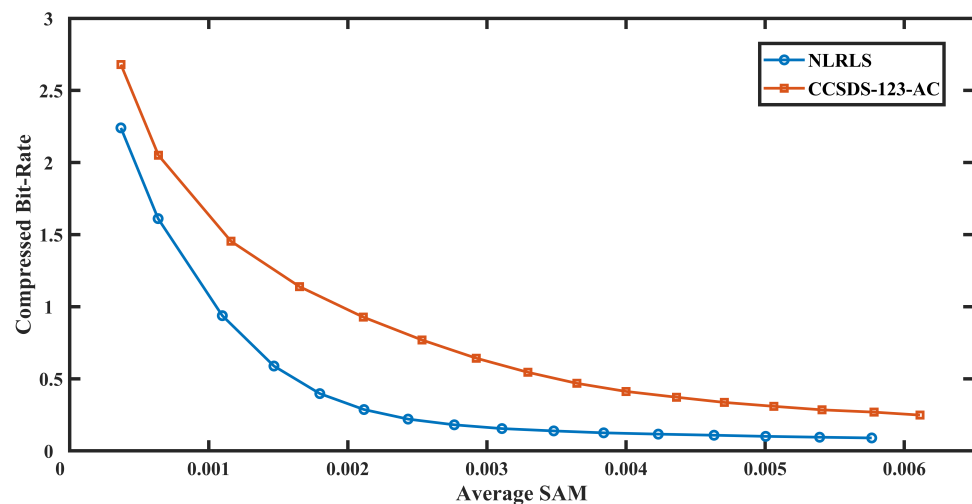**Figure 6.** The PSNR performance of the proposed method compared to CCSDS-123-AC.

Figure 7. The SAM performance of the proposed method compared to CCSDS-123-AC.

The experimentally obtained data points have been highlighted by marked dots, and linear interpolation is used between the markers. It can be seen that under the same compressed bit rate, the proposed scheme has a higher peak signal-to-noise ratio and a smaller average spectral angle. Further, for image quality, as measured by the PSNR and SAM, our approach is competitive compared to the best-known CCSDS-123-AC scheme.

4. Conclusions

In this paper, the prediction-based near-lossless compression technique is used to reduce the size of the hyperspectral image. The target pixel is predicted from the combination of previous pixels in the spatial and spectral bands. The coefficients are predicted using the weight matrix of the RLS filter with an in-loop quantizer. The experiments were performed on three types of CCSDS hyperspectral images' datasets, including three, five, and eight scenes, respectively. The optimal number of bands for the loop predictor was analyzed experimentally. The results indicate that the proposed scheme provides an average optimization from 5.29% to 31.19% in bit-per-pixel for different peak absolute errors and achieves a competitive reconstructed image quality compared to the state-of-the-art methods.

In the future, we plan to develop an automated model for parallel processing of near-lossless compression schemes and feasible hardware implementation solutions for space-based platforms. This research work will further pave the way for future developments in the field of deep space exploration.

Author Contributions: Conceptualization, T.Z.; methodology, T.Z. and Y.D.; investigation, T.Z. and C.X.; formal analysis, C.X., L.Z. and T.Z.; writing—original draft, T.Z.; writing—review and editing, Y.D. and C.X.; funding acquisition, C.X. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. 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]
2. Santos, L.; Gomez, A.; Sarmiento, R. Implementation of CCSDS standards for lossless multispectral and hyperspectral satellite image compression. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *56*, 1120–1138. [CrossRef]
3. Bascones, D.; Gonzalez, C.; Mozos, D. A Real-Time FPGA Implementation of the CCSDS 123.0-B-2 Standard. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5525113. [CrossRef]
4. Fjeldtvedt, J.; Orlandic, M.; Johansen, T.A. An Efficient Real-Time FPGA Implementation of the CCSDS-123 Compression Standard for Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3841–3852. [CrossRef]
5. 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]
6. Zheng, T.; Xue, C.; Song, J. Lossless compression of hyperspectral images using recursive least square lattice filter group. *Opt. Precis. Eng.* **2021**, *29*, 896. [CrossRef]
7. Altamimi, A.; Ben Youssef, B. A Systematic Review of Hardware-Accelerated Compression of Remotely Sensed Hyperspectral Images. *Sensors* **2022**, *22*, 263. [CrossRef]
8. Alvarez-Cortes, S.; Serra-Sagrista, J.; Bartrina-Rapesta, J.; Marcellin, M.W. Regression Wavelet Analysis for Near-Lossless Remote Sensing Data Compression. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 790–798. [CrossRef]
9. Roger, R.; Arnold, J. Reversible image compression bounded by noise. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 19–24. [CrossRef]
10. Blanes, I.; Magli, E.; Serra-Sagrista, J. A Tutorial on Image Compression for Optical Space Imaging Systems. *IEEE Geosci. Remote Sens. Mag.* **2014**, *2*, 8–26. [CrossRef]

11. Dua, Y.; Kumar, V.; Singh, R.S. Comprehensive review of hyperspectral image compression algorithms. *Opt. Eng.* **2020**, *59*, 090902. [CrossRef]

12. Valsesia, D.; Magli, E. Fast and Lightweight Rate Control for Onboard Predictive Coding of Hyperspectral Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 394–398. [CrossRef]

13. Valsesia, D.; Magli, E. A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6341–6355. [CrossRef]

14. Bartrina-Rapesta, J.; Marcellin, M.W.; Serra-Sagrista, J.; Hernandez-Cabronero, M. A Novel Rate-Control for Predictive Image Coding With Constant Quality. *IEEE Access* **2019**, *7*, 103918–103930. [CrossRef]

15. 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]

16. Wu, X.; Memon, N. Context-based, adaptive, lossless image coding. *IEEE Trans. Commun.* **1997**, *4*, 437–444. [CrossRef]

17. Wu, X.; Memon, N. Context-based lossless interband compression - Extending CALIC. *IEEE Trans. Image Process.* **2000**, *9*, 994–1001. [CrossRef]

18. 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]

19. Klimesh, M. Low-Complexity Lossless Compression of Hyperspectral Imagery via Adaptive Filtering. No. IPN-PR-42-163 2005, Volume 42–163, pp. 1–10. Available online: https://ipnpr.jpl.nasa.gov/progress_report/42-163/163H.pdf (accessed on 12 July 2022).

20. Consultative Committee for Space Data Systems. Lossless Multispectral & Hyperspectral Image Compression. 2012. Available online: https://public.ccsds.org/Pubs/123x0b1ec1s.pdf (accessed on 12 July 2022).

21. Bartrina-Rapesta, J.; Blanes, I.; Auli-Llinas, F.; Serra-Sagrista, J.; Sanchez, V.; Marcellin, M.W. A Lightweight Contextual Arithmetic Coder for On-Board Remote Sensing Data Compression. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4825–4835. [CrossRef]

22. Ansari, R.; Memon, N.D.; Ceran, E. Near-lossless image compression techniques. *J. Electron. Imaging* **1998**, *7*, 486–494.

23. Tai, S.C.; Kuo, T.M.; Ho, C.H.; Liao, T.W. A near-lossless compression method based on CCSDS for satellite images. In Proceedings of the 2012 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 4–6 June 2012; pp. 706–709.

24. Consultative Committee for Space Data Systems. Recommendation for Space Data System Standards; 2005. Available online: https://public.ccsds.org/Pubs/122x0b1c3s.pdf (accessed on 12 July 2022).

25. Valsesia, D.; Magli, E. High-Throughput Onboard Hyperspectral Image Compression with Ground-Based CNN Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9544–9553. [CrossRef]

26. Wu, X.; Bao, P. Near-lossless image compression by combining wavelets and CALIC. In Proceedings of the Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No. 97CB36136), Pacific Grove, CA, USA, 2–5 November 1997; Volume 2, pp. 1427–1431.

27. Beerten, J.; Blanes, I.; Serra-Sagrista, J. A Fully Embedded Two-Stage Coder for Hyperspectral Near-Lossless Compression. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1775–1779. [CrossRef]

28. Dua, Y.; Singh, R.S.; Kumar, V. Compression of multi-temporal hyperspectral images based on RLS filter. *Vis. Comput.* **2022**, *38*, 65–75. [CrossRef]

29. CCSDS. 123.0-B-Info, TestData. 2019. Available online: http://cwe.ccsds.org/sls/docs/SLS-DC/123.0-B-Info/TestData (accessed on 1 June 2019).

30. Ibn Afjal, M.; 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]

31. Karaca, A.C.; Gullu, M.K. Superpixel based recursive least-squares method for lossless compression of hyperspectral images. *Multidimens. Syst. Signal Process.* **2019**, *30*, 903–919. [CrossRef]