Article

Remote Sensing Image Compression Based on Direction Lifting-Based Block Transform with Content-Driven Quadtree Coding Adaptively

Cuiping Shi [1,2,*], Liguo Wang [2], Junping Zhang [3], Fengjuan Miao [1] and Peng He [1]

[1] College of Communication and Electronic Engineering, Qiqihar University, Qiqihar 161000, China; miaofengjuan@163.com (F.M.); hphgy@163.com (P.H.)

[2] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China; wangliguo@hrbeu.edu.cn

[3] College of Electronic and Communication Engineering, Harbin Institute of Technology, Harbin 150001, China; zhangjp@hit.edu.cn

[*] Correspondence: scp1980@126.com; Tel.: +86-0452-2742607

check for updates

Abstract: Due to the limitations of storage and transmission in remote sensing scenarios, lossy compression techniques have been commonly considered for remote sensing images. Inspired by the latest development in image coding techniques, we present in this paper a new compression framework, which combines the directional adaptive lifting partitioned block transform (DAL-PBT) with content-driven quadtree codec with optimized truncation (CQOT). First, the DAL-PBT model is designed; it calculates the optimal prediction directions of each image block and performs the weighted directional adaptive interpolation during the process of directional lifting. Secondly, the CQOT method is proposed, which provides different scanning orders among and within blocks based on image content, and encodes those blocks with a quadtree codec with optimized truncation. The two phases are closely related: the former is devoted to image representation for preserving more directional information of remote sensing images, and the latter leverages adaptive scanning on the transformed image blocks to further improve coding efficiency. The proposed method supports various progressive transmission modes. Experimental results show that the proposed method outperforms not only the mainstream compression methods, such as JPEG2000 and CCSDS, but also, in terms of some evaluation indexes, some state-of-the-art compression methods presented recently.

Keywords: compression; remote sensing images; directional lifting; quadtree coding

1. Introduction

Along with the rapid development of remote sensing technology, it is becoming easier to acquire high spatial resolution remote sensing images from various satellites and sensors, which is undoubtedly beneficial to the application of remote sensing images [1,2]. On the other hand, massive data has brought a great burden to the storage and transmission of remote sensing images. Therefore, compression technology is indispensable in remote sensing image processing. In general, traditional compression methods can be used for the compression of remote sensing images, such as Embedded Zerotree Wavelet (EZW) [3], Set Partitioning in Hierarchical Tree (SPIHT) [4], Set Partitioned Embedded Block Coder (SPECK) [5], and Joint Photographic Experts Group 2000 (JPEG2000) [6], or some improved versions of them [7–10]. However, the remote sensing image has its own unique characteristics, which usually include complicated spatial information, clear details, and well-defined geographical objects, and even some small targets [11]. This means that some traditional methods that rarely consider high-frequency information and mainly focus on retaining more low-frequency information

are not very applicable. In order to efficiently compress the remote sensing image, its characteristics must be taken into account in the compression process.

In this decade, some compression methods specifically designed for remote sensing images have been proposed. The Consultative Committee for Space Data Systems (CCSDS) published an Image Data Compression (-CCSDS 122.0-B-1) standard [12], which specifically targets space-borne equipment and focuses more on the complexity of the algorithm and less on its flexibility. In [13], it makes some improvement on the CCSDS 122.0-B-1 standards, which supports several ways of interactive decoding. In [14], the CCSDS published the CCSDS 122.0-B-2, which adds some modifications to support the recommended standard for spectral pre-processing transform for multispectral and hyperspectral image compression. In [15] the authors extend the JPEG2000 for the compression of remote sensing images. These compression methods are based on Discrete Wavelet Transform (DWT). In recent years, DWT-based and embedded bit-plane coding techniques have been very popular [16]. However, due to DWT's property, the energy of images will be spread across subbands if the edges and textures are not horizontal or vertical [16]. For remote sensing images, after DWT, the energy of the high frequency subbands is usually very large for its characteristics related to complicated terrain information. Therefore, a more effective image representation method is desirable for the compression of remote sensing images. In the last few years, several specialized transform methods that aim to improve the treatment of geometric image structures have been proposed, such as curvelets [17], coutourlets [18], directionlets [19], and shearlets [20]. Instead of choosing an a priori basis or frame to represent an image, some adaptive wavelet basis, such as [21,22], have been proposed. In order to employ the direction transform methods for image compression, the authors of [23,24] integrate the direction prediction methods into the wavelet lifting framework, and preserve more directions of images. The authors of [25] present a compressive sensing method exploiting the interscale and intrascale dependencies by directional lifting wavelet transform. The authors of [26] provides an adaptive lifting wavelet transform, which adapts the lifting direction to local image blocks and can preserve more edge information of images. The authors of [27] provide an analysis of the characteristics of airborne LIDAR data and some efficient representation methods for remote sensing images. In [28], a compression technique that considers the properties of predictive coding and discrete wavelet coding is designed. The authors of [29] propose a compression method for remote sensing images based on both saliency detection and directional wavelet for remote sensing images. In recent years, there have also been many studies on multicomponent image compression. In September 2017, CCSDS published a recommended standard (CCSDS 122.1-B-1) [30]. This recommended standard is an extension to CCSDS 122.0-B-2 [14], which defines a data compression algorithm that can be applied to three-dimensional images, such as multispectral and hyperspectral images. The authors of [31] propose a rate control method for the onboard predictive compression. In [32], the authors propose a new compression method based on compressed sensing and universal scalar quantization for the low-complexity compression of multispectral images. In [33], the authors propose a sparse representation-based compression method for hyperspectral images, which tends to represent hyperspectral data with a learned dictionary via sparse coding. In [34], the authors propose a compression method for hyperspectral images, which constructs some superpixels to describe hyperspectral data. Most of these image compression methods focus only on the representation of image directional information, and combine it with existing coding methods. However, image representation and coding are two closely related stages of compression. Considering the two stages jointly will be advantageous for improving the compression performance of remote sensing images.

Embedded coding is one of the most popular coding techniques, which can truncate the bitstream at any position and reconstruct the image [35]. Embedded block coding with optimized truncation (EBCOT) is an effective coding method, which is the fundamental part of the JPEG2000 image compression standard. It divides each transformed subband into many independent code-blocks, encodes these code-blocks, and performs optimal truncation based on the specified bit rate, respectively.

However, the EBCOT method only considers the correlation within a subband, and there are a lot of truncated points that need to be stored during the process of running the algorithm [9,36].

Besides EBCOT, tree-based coding methods have been widely used. Some of the embedded coding methods are based on zerotree, such as [3–5,37–39]. The zerotree-based method exploits the property of self-similarity across scales in wavelet transformed images, which construct some zerotrees by spatial similarity of the wavelet coefficients at different scales. As a result, a great deal of non-significant coefficients can be represented by a root of the zero tree. Recently, some binary, tree-based coding methods were proposed for the compression of remote sensing images. The authors of [40] present a binary tree codec adaptively (BTCA) for on-board image compression. As a novel and robust coding way, the BTCA method has proven to be very effective. In [41], a human vision-based adaptive scanning (HAS) for the compression of remote sensing images was proposed, which combines the human visual model with the BTCA, and can provide a good visual quality for remote sensing images. Additionally, the quadtree-based coding methods are also presented for the compression of remote sensing image coding. The authors of [24] propose a compression method that is based on quadtree coding for Synthetic Aperture Radar (SAR) images and obtains a good coding performance.

In 2017, the state-of-the-art compression method based on quadtree set partitioning, known as quadtree coding with adaptive scanning and optimized truncation (QCAO), was proposed [42]. The main concept of this method is that the entire wavelet image is divided into several blocks and encoded individually. In the process of coding, the significant nodes and their neighbors are encoded before other nodes, and then the optimized truncation is followed to improve the coding efficiency. Although the QCAO is an effective coding method, similar to EBCOT, it mainly utilizes the clustering characteristics within a subband, but seldom considers the relations among subbands.

In this paper, a new compression method for remote sensing images is proposed. In order to improve the compression efficiency, the image representation phase and coding phase are jointly considered. In addition, both the characteristics of subbands and the scanning order among subbands and blocks are utilized in the process of coding. The overall framework of the proposed compression method is shown in Figure 1.
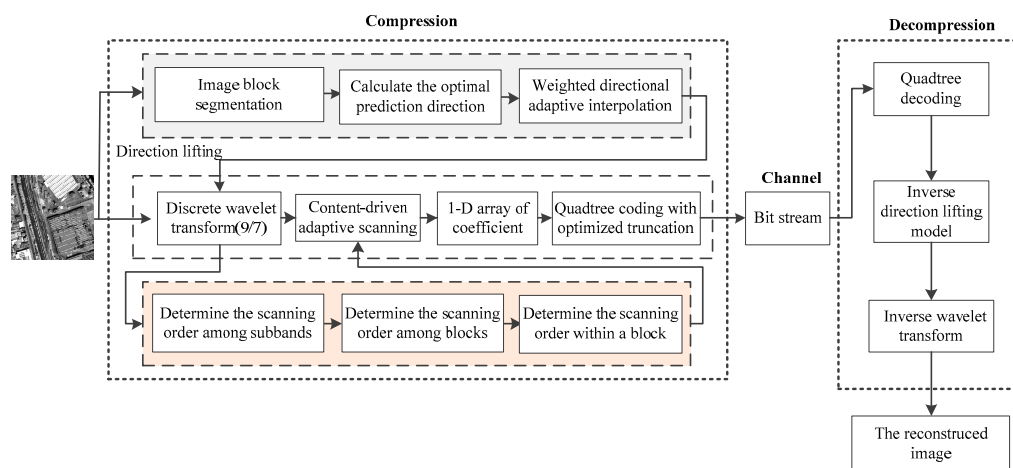

Figure 1. The overall framework of the proposed compression method.

The main contribution of the proposed method consists of two parts:

One part is the directional adaptive lifting partitioned block transform (DAL-PBT), which can provide the optimal prediction direction for each image block and preserve more orientation properties of remote sensing images by directional interpolation. It should be noted that the size of the image block here is consistent with that of the following coding phase.

Another part is the content-driven quadtree codec with optimized truncation (CQOT), which establishes the quadtree for each image block by analyzing the scanning order among blocks

and the scanning method of each block based on the image content and the characteristics of subbands, respectively. Then, the coding and optimized truncation is followed for each quadtree in order to provide a good overall coding performance.

The remainder of the paper is organized as follows. In Section 2, the proposed DAL-PBT model is described in detail. In Section 3, we first describe the content-driven adaptive scanning method, before the CQOT method is introduced. Section 4 gives some quality evaluation indicators used in this paper. In Section 5, some numerical experiments are performed and prove the high effectiveness of the proposed compression method. Finally, the conclusions are provided in Section 6.

2. The Directional Adaptive Lifting Partitioned Block Transform (DAL-PBT)

The two-dimensional (2-D) DWT is the most important image representation technique of the last decade [43]. Conventionally, the 2-D DWT is carried out as a separable transform by cascading two 1-D transforms in the vertical and horizontal directions, which only use the neighboring elements in the horizontal or vertical direction. However, most remote sensing images contain a lot of directional information, such as edges, contours of terrain, and complex texture of objects, which ensure that the conventional 2-D DWT fails to provide a good representation for them. How to provide an efficient image representation is one of the keys to improving the coding performance of remote sensing images. In this paper, a new DAL-PBT model is proposed. It first divides an image into several blocks and then calculates their optimal lifting directions. Then, a directional interpolation filter is utilized during the lifting process to improve the orientation property of the interpolated remote sensing images. The DAL-PBT model is described in detail as follows.

2.1. The Structure of Adaptive Directional Lifting Scheme

Without loss of generality, we present a general formulation of the 2-D DWT implemented with lifting. The authors of [44] point out that the 2-D wavelet transforms can be realized by one pair of the lifting steps, i.e., one prediction step followed by one update step. The typical lifting process consists of four stages: split, prediction, update, and normalize [45]. The 1-D directional lifting wavelet transform and inverse transform are shown in Figure 2a,b.
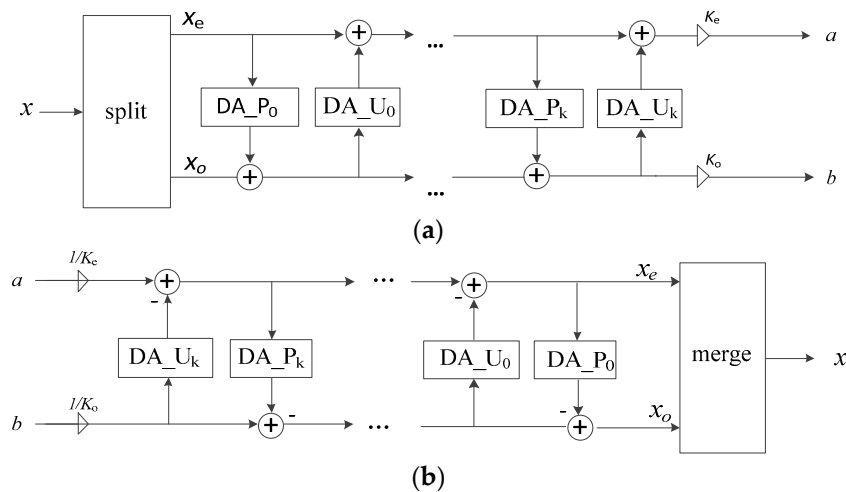

Figure 2. Generic 1-D directional lifting wavelet transform. (**a**) Forward decomposition. (**b**) Reverse synthesis.

We consider a 2-D image $x(m,n)_{m,n \in Z}$. First, all samples of the image are split into two parts: the even sample subset x_e and the odd sample subset x_o.

$$\begin{cases} x_e[m,n] = x[2m,n] \\ x_o[m,n] = x[2m+1,n] \end{cases} \tag{1}$$

In the predict stage, the odd elements are predicted by the neighboring even elements with a prediction direction obtained via a discriminant criteria. Suppose the direction adaptive prediction operator is DA_P; then, the predict process can be represented as

$$d[m,n] = x_o[m,n] + DA\_P_e[m,n] \tag{2}$$

In the update stage, the even elements are updated by the neighboring prediction error with the same direction of the predict stage. Suppose the direction adaptive update operator is DA_U, the updated process can be described as

$$c[m,n] = x_e[m,n] + DA\_U_d[m,n] \tag{3}$$

Here, the directional prediction operator DA_P is

$$DA\_P_e[m,n] = \sum_i p_i x_e[m + sign(i-1)\tan\theta_v, n+i] \tag{4}$$

The directional update operator DA_U is

$$DA\_U_d[m,n] = \sum_j u_j d[m + sign(j)\tan\theta_v, n+j] \tag{5}$$

The p_i and u_j are the coefficients of high pass filter and low pass filter, respectively. θ_v represents the direction of prediction and update.

Finally, the outputs of the lifting are weighted by the coefficients K_e and K_o, which normalize the scaling and wavelet functions, respectively.

For a remote sensing image, if the directional lifting is carried out along the edge or texture direction of the image, then the energy of the high frequency subbands will decrease, which is advantageous for improving the coding performance. In this paper, the integer pixels and the fractional pixels close to the predicted pixel are considered. Obviously, the more reference lifting directions, the better representation of the image block. However, more side information is needed. On the other hand, if there are very few reference lifting directions, the image cannot be well represented. In this paper, 15 reference lifting directions are chosen for one-dimensional horizontal and vertical transformations, which are shown in Figure 3a,b, respectively. Here, the directional filter can be represented along the direction $d = (d_x, d_y)^T$, $d \in \mathbb{R}^2$. Therefore, the 15 reference directions can be recorded as $d_{-7} = (3, -1)^T$, $d_{-6} = (2, -1)^T$, $d_{-5} = (1, -1)^T$, $d_{-4} = (3/4, -1)^T$, $d_{-3} = (1/2, -1)^T$, $d_{-2} = (1, -3)^T$, $d_{-1} = (1/4, -1)^T$, $d_0 = (0, -1)^T$, $d_1 = (-1/4, -1)^T$, $d_2 = (-1, -3)^T$, $d_3 = (-1/2, -1)^T$, $d_4 = (-3/4, -1)^T$, $d_5 = (-1, -1)^T$, $d_6 = (-2, -1)^T$, $d_7 = (-3, -1)^T$.

2.2. Image Segmentation and the Calculation of Optimal Prediction Direction

In order to ensure that the direction of the lifting is consistent with the local direction of the image, image segmentation is carried out. In [46], a rate-distortion optimization segmentation method based on quadtree is adopted, which divides a natural image recursively into blocks with different sizes by quad-tree segmentation. However, the efficiency of this segmentation method is closely related to the content of the image. For the remote sensing image, it usually reflects complex landforms rich in details, but rarely large flat areas, so the adaptive segmentation method hardly shows its advantages. The reason for this is that for images with complex content, the adaptive segmentation result most likely shows all of the blocks with the smallest allowed size, which is nearly equivalent to equal size segmentation but with a higher computational cost. In addition, for rate-distortion optimization segmentation method based on quadtree, the "segmentation tree" at each bit rate is different, and all of them should be transmitted to the receiver for correct decoding. The more complex the image is, the more branches of the "segmentation tree" there are, which leads to more side information.

Therefore, the rate-distortion optimization segmentation method based on quadtree is not very suitable for remote sensing images.
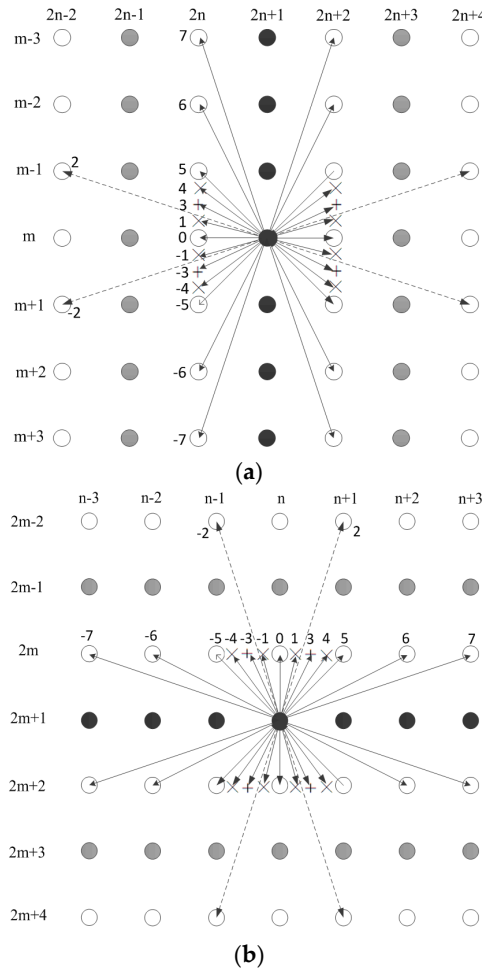

Figure 3. Set of reference directions of directional lifting-based wavelet transform. (**a**) Horizontal transform and (**b**) vertical transform.

Based on the above analysis, the block segmentation method with the same size is directly adopted for remote sensing images. In order to provide a better coding performance, the size of the partition block here is the same as that of the following coding phase.

For the directional lifting-based wavelet transform, the prediction error is closely related to the high frequency subbands. For an image block, the optimal prediction direction is the direction with minimal residual information in high frequency subbands.

Suppose the reference direction set is θ_{ref}, which includes 15 directions, $\{-7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7\}$, and the total number of blocks is N_a. For each block B_l, $l = 0, 1, \ldots, N_a - 1$, the optimal prediction direction θ_{opti}^* can be calculated as follows:

$$\theta_{opti,B_l}^* = \operatorname*{argmin}_{i \in \theta_{ref}} \sum_{m,n \in B_l} D\left\{ x(m,n) - DA\_P^i(x(m,n)) \right\} \tag{6}$$

Here, $D(\cdot)$ is the function of the image distortion, and $x(m,n)$ represents the value at the position (m,n) in the block B_l. In this paper, let $D(\cdot) = |\cdot|$. Equation (6) illustrates that the optimal prediction direction of an image block B_l is the direction with a minimal prediction error.

The process of calculating the optimal prediction direction of a block is shown in Figure 4. For an original image, the directional wavelet transform is carried out along all the reference directions $\theta_{ref,i}(i = 1, 2, \ldots, 15)$, respectively. Following this, for each image block B_l of these transformed images, we are looking for the "tile" with a minimum prediction error, and its corresponding prediction direction is the optimal prediction direction θ_{opti}^*. We repeat this process, until all the optimal prediction directions of all the image blocks are determined. For each block, with the optimal prediction direction θ_{opti}^*, the predicted process and the updated process of all its elements are carried out via Equations (4) and (5), respectively.
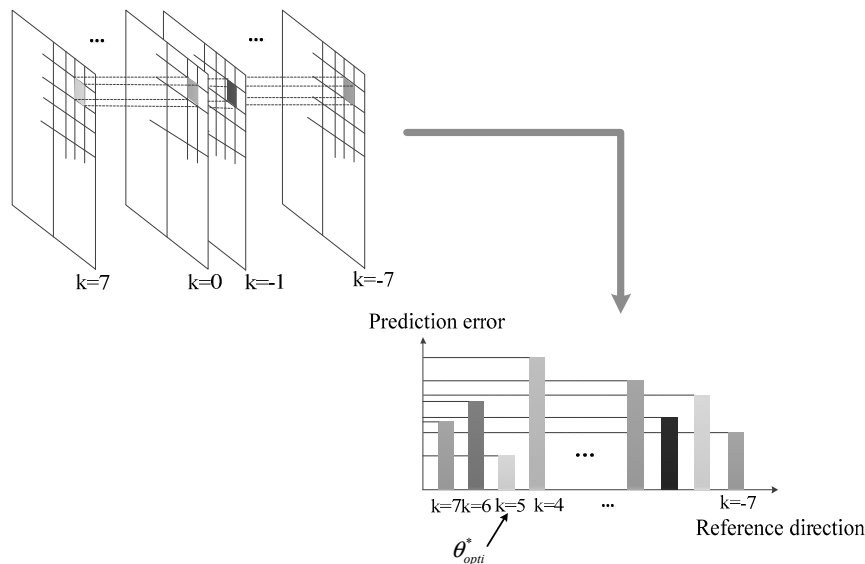

Figure 4. The process of calculating the optimal prediction direction of a block.

Compared with the adaptive segmentation method, the proposed direction lifting-based wavelet transform does not need to transmit the "segmentation tree" at all bit rates, and it only needs to transmit a small number of prediction directions as side information. Therefore, the side information of the proposed method is low.

2.3. Directional Adaptive Interpolation

For the directional lifting-based wavelet transform, the lifting direction often needs the values at fractional positions, i.e., the tangent of the lifting direction $\tan \theta$ is not always an integer. In this situation, it is necessary to interpolate those samples at fractional locations. The *Sinc* interpolation is a very popular interpolation method. However, it only uses the coefficients in the horizontal or vertical direction, which may blur the direction characteristics of the image. In this paper, a directional interpolation method that interpolates the sub-pixels along the local texture directions by using the adjacent integer pixels is adopted. As an example, for the horizontal transform, the process of direction interpolation is shown in Figure 5.

In Figure 5, for different sub-pixel positions, some different integer pixels are used to interpolate the sub-pixel, and the interpolation direction is adapted to the signal properties for interpolation [47]. For example, in order to interpolate a quarter-position coefficient, not only the integer coefficients $\{c_{-2}, c_{-1}, c_0, c_1\}$ are used but also the coefficients $\{c_{-3}, c_2\}$ along the predicted direction. The directional interpolation filter should be constructed by these coefficients $\{c_{-3}, c_{-2}, c_{-1}, c_0, c_1, c_2\}$ on the integer position and make a prediction to the sub-pixel position. The final coefficients of the directional interpolation filter are determined by three kinds of filters: bilinear filter, Telenor 4-tap filter, and 2-tap filter. The directional interpolation process is shown in Figure 6. The coefficients of these filters are listed in Table 1.
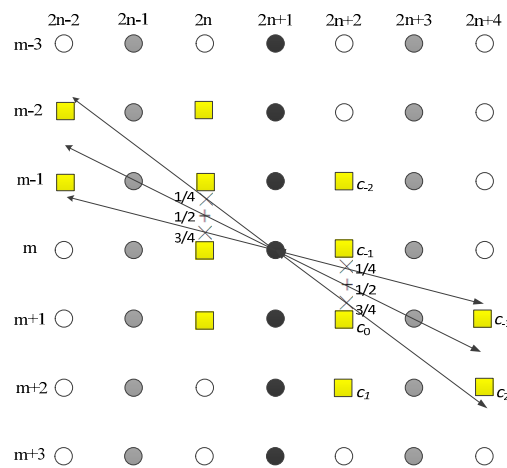
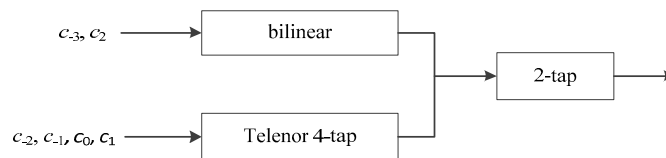**Figure 5.** The process of directional interpolation.

Figure 6. The process of directional interpolation.

Table 1. The applied filters of the directional interpolation process.

Filter	Position	The Weights of the Filter
Bilinear filter	1/4	3/4; 1/4
	1/2	1/2; 1/2
	3/4	1/4; 3/4
Telenor 4-tap filter	1/4	$-1/16; 13/16; 5/16; -1/16$
	1/2	$-1/8; 5/8; 5/8; -1/8$
	3/4	$-1/16; 5/16; 13/16; -1/16$
2-tap filter	N/A	$-1/4; 5/4$

Figure 6 shows that $\{c_{-3}, c_2\}$ are the inputs of the bilinear filter, $\{c_{-2}, c_{-1}, c_0, c_1\}$ are the inputs of the Telenor 4-tap filter, and both the outputs of the bilinear filter and the Telenor 4-tap filter are similar to the inputs of the 2-tap filter. As a result, the outputs of the 2-tap filter are the weighted coefficients of the directional interpolation filter.

2.4. Boundary Handing and Semidirection Displacement

The lifting operations of different image blocks are performed with their own optimal prediction directions. However, the boundary effect is a problem. In this paper, for the lifting of block boundaries, the adjacent pixels of other blocks are used, unless the boundary pixels are at the edge of the image. As a result, the error can be restrained to some extent and can avoid the boundary effect.

The processing of the semi-direction is another problem that might need attention. Assuming that a row transform is carried out firstly, the size of the image block is changed, which makes the block direction no longer accurate. To solve this problem, some studies try to reduce the lifting direction of a block by half, before performing another one-dimensional transform with the half angle, such as in [16]. However, the half-angle may not correspond exactly to the optimal prediction direction previously calculated, which may lead to direction deviation. In addition, the deviation will be accumulated with the increment of the wavelet decomposition levels. We adopt the directional interpolation method

introduced in Section 2.3 for solving this problem. The rectangular block is interpolated into a square block first, and then, another one-dimensional transform is performed using the previously calculated optimal direction.

3. A Content-Driven Quadtree Coding Method for Remote Sensing Images

Tree-based coding methods have been widely used in recent years. In 2017, a state-of-the-art coding method based on scanning for remote sensing images, known as quadtree coding with adaptive scanning order and optimized truncation (QCAO), was proposed [42]. The main idea of QCAO is that the transformed image is divided into several blocks, which are encoded by quadtree coding method, respectively. Then, the optimized truncation is performed for each codestream of a block in order to improve the coding performance.

Although the QCAO method is somehow related to the content of the image, it only considers the scanning process within a block. However, the contents of different images are often very different, especially for remote sensing images. In order to improve the coding performance, the scanning order among subbands should be different according to the image content. In addition, for a given subband, the scanning order and scanning methods among and within blocks should be determined while taking the characteristics of the subband into account. Therefore, a content-driven quadtree codec with optimized truncation (CQOT) is introduced, which provides an adaptive scanning method based on the image content and the characteristics of subbands, and then encodes more significant coefficients as much as possible at the same bit rate. The proposed CQOT algorithm is described in detail as follows.

3.1. Content-Driven Subband Scan and Block Scan

For a remote sensing image A with size $M \times N$, after the transform based on the DAL-PBT model, the transformed image can be recorded as B. The scanning process of B can be seen as a bijection f from a closed interval $[1, 2, \ldots, M \times N]$ to the set of ordered pairs $\{(i, j) : 1 \leq i \leq M, 1 \leq j \leq N\}$, in which the second set represents the locations of the image. After scanning, the 1-D coefficient sequence can be recorded as $[B_{f(1)}, B_{f(2)}, \ldots, B_{f(MN)}]$. Different scanning processes are actually bijective functions f with different definitions. If the function f is defined, then the image compression process becomes a process of compressing a 1-D coefficient sequence $\widetilde{B} = [B_{f(1)}, B_{f(2)}, \ldots, B_{f(MN)}]$.

For the image compression at the same bit rate, different scanning orders can lead to different coding efficiencies, which makes the scanning order very important. Generally, a 2-D transformed image can be converted into a 1-D coefficient sequence using some traditional scanning strategies, such as zigzag scan, morton scan, or raster scan. However, these scanning methods do not take the image content and the characteristics of the wavelet subband into consideration, which will have an impact on the coding performance. Moreover, remote sensing images usually contain some important details, such as terrain contours, complex object textures, or even small targets, which also need to be preserved as much as possible during the encoding process. Therefore, how to design an effective adaptive scanning method is also an important issue for the compression of remote sensing images.

In this paper, an adaptive scanning method is presented. This scanning method can provide different scanning orders among subbands based on the image content, and different scanning orders and scanning methods among and within a block according to the characteristics of subbands. The adaptive scanning process based on the image content and the characteristics of subbands (ASCC) is shown in *function ASCC* (Algorithm 1).

Algorithm 1. *Function $S = ASCC(X, J, b)$.*

Input: The transformed image X with the DAL-PBT model, the decomposition level J, and the size of block b.

- Calculate the energy of each subband, and represent it as $E_{\lambda,\theta}$.

  l-scale ($m = 1, 2, \ldots, J$), θ-direction ($d = 1, 2, 3, 4$). '1' represents the low frequency subband, '2' represents the horizontal direction, '3' represents the diagonal direction, and '4' represents the vertical direction, respectively.

  $$E_{l,\theta} = \frac{1}{RC} \sum_{i,j}^{r,c} X(l,\theta)(i,j)^2$$

  r and c are the number of row and column of the current subband $X(l,\theta)$, respectively; $X(l,\theta)(i,j)$ represents the value of the current subband at location (i,j).

- Determine the scanning order *among subbands* according to the descending order of $E_{l,\theta}$.
- Determine the scanning order *among blocks* in a subband based on the characteristics of this subband.

  - If these blocks are in the lowest frequency subband or horizontal subband, the "horizontal z-scan" is adopted.
  - If these blocks are in the vertical subband, the "vertical z-scan" is performed.
  - If these blocks are in the diagonal subband, then the scanning order among blocks depends on the energy of horizontal subband and vertical subband of this level.

    ①   If $E_{l,2} \geq E_{l,4}$, the "horizontal z-scan" is adopted.
    ②   If $E_{l,2} < E_{l,4}$, the "vertical z-scan" is adopted.

- Determine the scanning method *within a block* according to the characteristics of this subband.

  - **For** each block $X\_Block(i)$
  - If it is in the lowest frequency subband or a horizontal subband, the "horizontal z-scan" is adopted.
  - If it is in a vertical subband, the "vertical z-scan" is performed.
  - If it is in a diagonal subband, then the scanning method depends on the horizontal subband and vertical subband of this level.

    ①   If $E_{l,2} \geq E_{l,4}$, the "horizontal z-scan" is performed to this block.
    ②   If $E_{l,2} < E_{l,4}$, the "vertical z-scan" is performed to this subband.

    End

Output: The generated 1-D coefficient sequence S by scanning the 2-D transformed image X.

It has been demonstrated that from the ASCC algorithm, the adaptive scanning process is determined by the image content and the characteristics of subbands together. To explain this algorithm clearly, we give an example of the ASCC algorithm. The remote sensing image "Europa3" is chosen as a test image. Suppose the decomposition level is 3, after the wavelet transform with DAL-PBT model, the scanning order among the subbands is determined according to the descending order of energy of these subbands, i.e., LL_3, LH_3, HH_3, HL_3, LH_2, HH_2, HL_2, LH_1, HL_1, HH_1. Following this, the scanning order among blocks in each subband is determined based on the characteristics of subbands. Take the subbands LH_1, HL_1, and HH_1, for example: the scanning order among blocks in LH_1 is conducted with the "vertical z-scan" method, and that in HL_1 follows the "horizontal z-scan" method. Because the energy of LH_1 is larger than that of HL_1, the scanning order among blocks in HH_1 follows the "vertical z-scan" method. Finally, for a given block, its scanning method depends on the characteristics of the subband in which the block is located. The adaptive scanning process of the test image is shown in Figure 7. In Figure 7, take a 4×4 block, for example: the scanning result and the corresponding quadtree is provided. The numbers outside the circles represent the establishing order

for the quadtree. The numbers in the circles are the coefficient values. Moreover, the green circles and the brown circles refer to the significant coefficients at different scanning thresholds of the quadtree, which will be discussed in detail in Section 3.2.
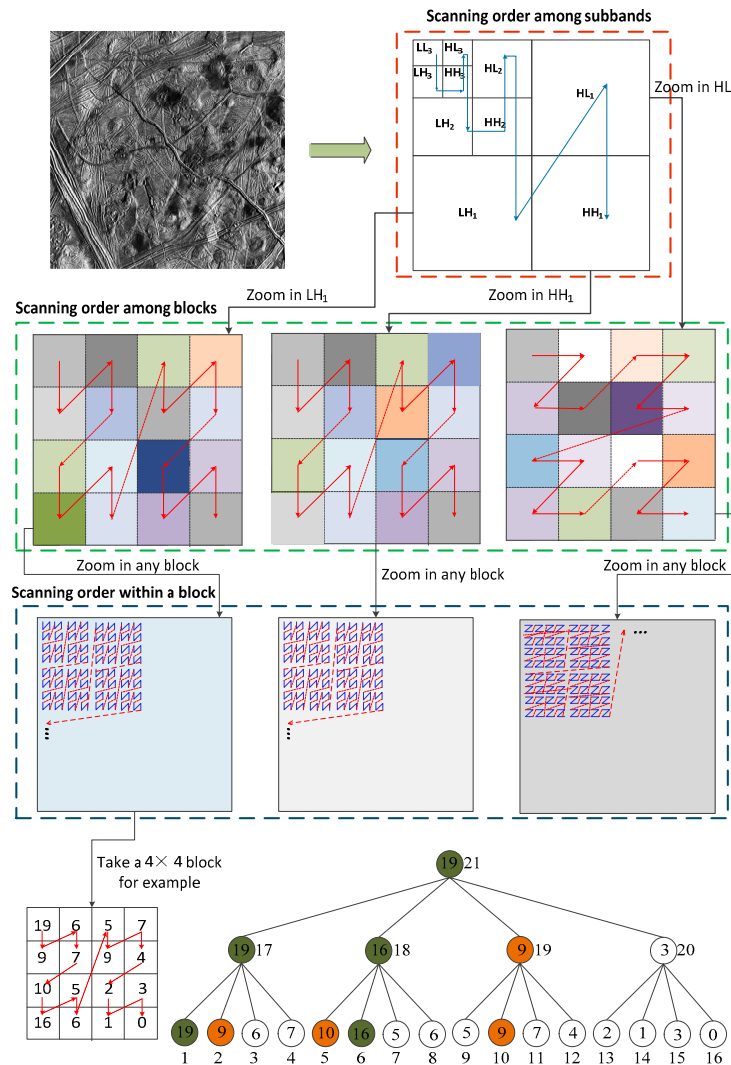

Figure 7. The adaptive scanning process of the test image.

3.2. Content-Driven Quadtree Codec with Optimized Truncation (CQOT)

The tree-based coding method has been widely used in recent years. Inspired by the most recent development in the quadtree coding technique [42], we propose a CQOT method that combines the quadtree codec with the content-based adaptive scanning method, introduced in the last section. In this framework, for a transformed image with a DAL-PBT model, the ASCC method is adopted first to provide the whole scanning order of the image. Secondly, for each block, a 1-D coefficient sequence is obtained after scanning, and a quadtree is established based on the sequence by comparing the four adjacent coefficients in turn. Then, each quadtree of a block is encoded individually, and the optimized truncation is performed by setting the truncation points during the coding process. The CQOT algorithm is described as Algorithm 2.

Algorithm 2. *Function* $[code, P] = CQOT(X, J, b, T_k)$.

Input: The transformed image X with the DAL-PBT model, and the wavelet decomposition level J; the size of each block b, T_k is the current threshold.

- For each quadtree Γ of a block, suppose that the total number of coefficients is L_0, the height of the quadtree is $H = log(L_0)/\log 4$, and the total number of nodes of the quadtree is $L = \sum_{i-0}^{H} 4^i$.
- $S = ASCC(X, J, b)$.
- Establish the quadtree Γ for each block based on the 1-D coefficient sequence obtained by the scanning order S, respectively.
- *Initialization*: $code = \{\}$.
- Encode each quadtree Γ.

$d = H$;
$e = L$;
$b = e - 4^d + 1$;
$P = \{\}$
while $(b > 1)$
{
  $j = b$;
    while $(j < e)$
    {
      if $\max|\Gamma(j : j + 3)| \geq 2T_k$,then
      if $|\Gamma(j)| < 2T_k$, then
      $code = code \cup QC\{\Gamma, j, T_k\}$;
      end if
      if $|\Gamma(j + 1)| < 2T_k$, then
      $code = code \cup QC\{\Gamma, j + 1, T_k\}$;
      end if
      if $|\Gamma(j + 2)| < 2T_k$, then
      $code = code \cup QC\{\Gamma, j + 2, T_k\}$;
      end if
      if $|\Gamma(j + 3)| < 2T_k$, then
      $code = code \cup QC\{\Gamma, j + 3, T_k\}$;
      end if
      end if
      $j = j + 4$;
    Calculate the series of optimized truncation points $\left\{ P_{j_k} \right\}$;
    $P = P \cup \left\{ P_{j_k} \right\}$;
    }
  $d = d - 1$;
  $e = b - 1$;
  $b = e - 4^d + 1$;
}
Output: The codestream *code* of the transformed image at the given threshold T_k, and the truncation points set *p*.
}

For a transformed image, the neighbors of the significant coefficients are usually important, because they often reflect some contour or edge information, which is very beneficial for preserving the details of remote sensing images. The CQOT algorithm scans the neighbors of the previous significant coefficients before other regions are scanned. A simple example is shown in Figure 8. In Figure 8, the green nodes are the significant coefficients at the current threshold, and the brown nodes are the previous significant coefficients. During the process of scanning, the neighbors of those brown nodes are encoded in priority, and then the encoding of green nodes follows.
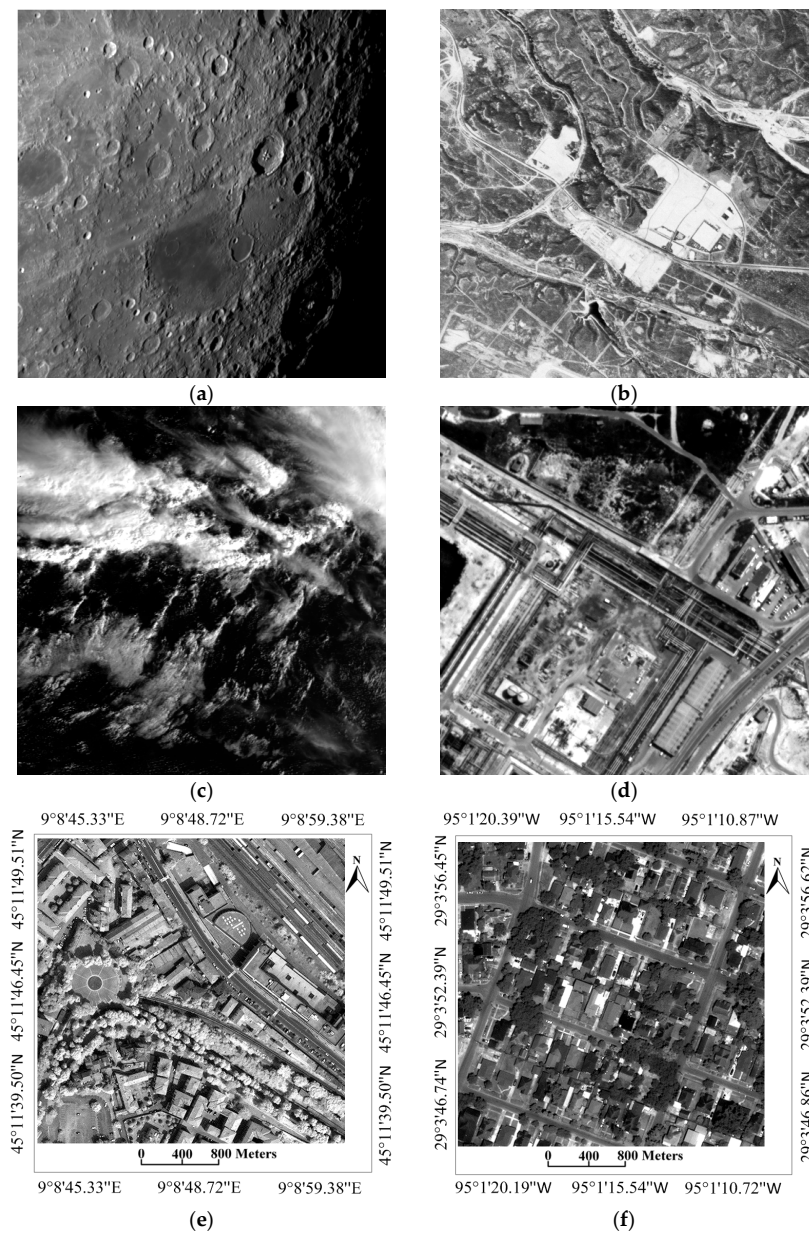
Figure 8. The test remote sensing image set. (**a**) Lunar; (**b**) Miramar NAS; (**c**) ocean_2kb1; (**d**) pleiades_portdebouc_pan; (**e**) Pavia; and (**f**) Houston.

To apply a rate-distortion optimization, the question of how to obtain the valid truncation points is an important issue. Similar to EBCOT, a Post-Compression-Rate-Distortion (PCRD) algorithm is adopted to select the candidate truncation points for each block so that the minimum distortion can be obtained at a given bit rate.

For the *j*th candidate truncation point of a block, suppose the bit rate is R_j and corresponding distortion is D_j, then

$$P_j = \frac{D_{j-1} - D_j}{R_j - R_{j-1}} \tag{7}$$

represents the distortion-rate of the block.

If $P_{j+1} > P_j$, the jth candidate truncation point is not selected; then, the distortion-rate of the block is updated with

$$P_{j+1} = \frac{D_{j-1} - D_{j+1}}{R_{j+1} - R_{j-1}} \tag{8}$$

Following this, the P_{j+1} is compared with P_{j-1}. If $P_{j+1} > P_{j-1}$, the $(j-1)$th truncation point should also not be selected, and we update the distortion-rate of the block with

$$P_{j+1} = \frac{D_{j-2} - D_{j+1}}{R_{j+1} - R_{j-2}} \tag{9}$$

The process is carried out until $P_{j+1} < P_{j_k}$, then

$$P_{j+1} = \frac{D_{j_k-1} - D_{j+1}}{R_{j+1} - R_{j_k-1}} \tag{10}$$

Finally, the strictly decreasing $\{P_{j_k}\}$ is selected as the series of valid truncation points.

The QC algorithm adopted in the CQOT algorithm is described as follows. Algorithm 3 uses a stack to avoid a recursive function.

Algorithm 3. *Function* $code = QC(\Gamma, j, T_k)$.

Input: Γ represents a quadtree, and j is the index of a node of the quadtree. T_k represents the threshold.

- $code = \{\}$
- *Stack S; StackInit(S); Push(S, j);*
- **while** $(!Empty(S))$

{

- $j = pop(S)$;
- if $|\Gamma(j)| \geq 2T_k$ and $4j + 1 < L$ then

Push(S, 4j + 1);
Push(S, 4j);
Push(S, 4j − 1);
Push(S, 4j − 2);
else if $j > 1$ and $j \bmod 4 = 1$ and $|\Gamma(j-1)| < T_k$ and $|\Gamma(j-2)| < T_k$ and $|\Gamma(j-3)| < T_k$ then
if $4j + 1 \leq L$ then
Push(S, 4j + 1);
Push(S, 4j);
Push(S, 4j − 1);
Push(S, 4j − 2);
else
$code = code \cup sign(\Gamma(j))$
end if
else if $|\Gamma(j)| \geq T_k$ then
$code = code \cup \{1\}$;
if $4j + 1 \leq L$ then
Push(S, 4j + 1);
Push(S, 4j);
Push(S, 4j − 1);
Push(S, 4j − 2);
else
$code = code \cup sign(\Gamma(j))$
end if
else
$code = code \cup \{0\}$;
end if

}

Output: The codestream *code* of the quadtree at the given threshold. T_k.

3.3. The Overhead of Bits

For the proposed compression method, the bit overhead, including the optimal prediction directions of those image blocks, the scanning order among the subbands and the scanning method of those diagonal subbands are also encoded and sent to the decoder with the bitstream.

Suppose the size of the image is $M \times N$ and the block size is $P \times Q$. Then, the number of blocks is MN/PQ, which means that the number of optimum prediction directions that need to be saved is MN/PQ. Since the number of reference prediction directions is 15, one byte can store two prediction directions (record a direction with 4 bit). That is, $MN/2PQ$ bytes are required to store those prediction directions. Moreover, suppose the number of wavelet decomposition levels is J. Then, only $3J + 1$ statuses are needed to represent the scanning order among the subbands. Additionally, J statuses are needed to represent the scanning methods of J diagonal subbands. If the wavelet decomposition is inferior or equal to five levels, the $(3J + 1) + J$ status can be stored with no more than 11 bytes (record each status with 4 bit). Usually, the maximum byte overhead of the side information that needs to be sent to the decoder is $MN/2PQ + 11$.

We take the image in Figure 8 as an example. The size of 'Europa3' is 512×512. According to [42], the proper size of the image block is 64×64. Then, it requires $(512 \times 512)/(2 \times 64 \times 64) = 32$ bytes to store those optimal prediction directions. In addition, we record the scanning order among the subbands according to LL_J, HL_i, HH_i, and LH_i ($i = J, J - 1, \ldots\ldots, 1$), and the scanning order of those diagonal subbands according to HH_i ($i = J, J - 1, \ldots\ldots, 1$). Based on the algorithm *ASCC*, the scanning order among the subbands is 1, 4, 2, 3, 7, 5, 6, 9, 8, and 10, and the scanning methods of the diagonal subbands are 11, 11, and 11. (The 11 represents the "vertical z-scan" method, and 12 represents the "horizontal z-scan" method). Therefore, they can be stored with 7 bytes. The bit rate of side information is $(32 + 7) \times 8/512 \times 512 = 0.00119$ bpp. It is worth mentioning that, if the size of the image is larger or the bit depth of the image is greater than 8 bit, the proportion of overhead bits is smaller. Moreover, if entropy encoding is used for these overhead bits, the overhead cost would be further reduced. Based on the analysis above, the overhead bits of the proposed compression method are extremely small, which makes it have almost no impact on the coding performance.

4. Quality Evaluation Index

In order to evaluate the performance of the proposed compression method comprehensively, in this paper, PSNR (peak signal-to-noise ratio, dB), MS-SSIM (multi-scale structural similarity method), and VIF (visual information fidelity) are chosen as the evaluation indexes, respectively.

4.1. PSNR

Suppose X is the original image, its size is $M \times N$ and its possible maximum value is L. Suppose Y is the reconstructed image, then the PSNR can be represented as

$$PSNR = 10\log_{10}\frac{L^2}{MSE} \tag{11}$$

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{ij} - y_{ij})^2}{M \times N} \tag{12}$$

4.2. MS-SSIM

The MS-SSIM incorporates the variations of viewing conditions and is more flexible than SSIM [43]. Thus, we adopt the MS-SSIM as an evaluation index in this paper.

$$MS\_SSIM(x,y) = [l_M(x,y)]^{\alpha_M} \cdot \prod_{j=1}^{M}\left[c_j(x,y)^{\beta_j}s_j(x,y)^{\gamma_j}\right] \tag{13}$$

Here, $l_M(x, y)$ represents the luminance comparison at scale M, and $c_j(x, y)$ and $s_j(x, y)$ represent the contrast and structure comparison at scale j, respectively. The exponents α_M, β_j, and γ_j are used to adjust the relative importance of different components.

4.3. VIF

Based on the theory in [48], the model of VIF can be represented as

$$Y(i, j) = \beta(i, j)X(i, j) + V(i, j) + W(i, j) \tag{14}$$

$$\chi(i, j) = X(i, j) + W(i, j) \tag{15}$$

Here, $X(i, j)$ is the original image, $Y(i, j)$ is the reconstructed image, $\chi(i, j)$ is the image captured by humans, and $W(i, j)$ is the noise image of the human vision system (HVS). $X(i, j)$ is decomposed into several subbands $(X_1(i, j), X_2(i, j), \ldots, X_M(i, j))$ by wavelet transformation. $\beta(i, j)$ is the value of the window used in the distortion channel estimation. $V(i, j)$ is a stationary additive, zero-mean Gaussian noise. If the variance of $W(i, j)$ is σ_W, $c = 0.01$, and

$$Z(i, j)^2 = \frac{1}{M}\sum_{k=1}^{M} X_k(i, j)^2 \tag{16}$$

Then

$$VIF = \frac{\log_2\left(\frac{\beta(i,j)^2 Z(i,j)^2 + \sigma_w + c}{\sigma_w + c}\right)}{\log_2\left(\frac{Z(i,j)^2 + c}{c}\right)} \tag{17}$$

5. Experiments and Discussion

In this section, some experiments are carried out to evaluate the performance of the proposed compression method. The proposed compression method is compared with other five scan-based methods, which are the mainstream or the state-of-the-art compression methods, in terms of different evaluation indexes at different bit rates.

5.1. Space-Borne Images from Different Sensors

To fully verify the performance of the proposed compression method, the test space-borne images come from different sensors. Some test images are from a CCSDS test image set [49], which includes a variety of space imaging instrument data, such as planetary, solar, stellar, earth observations, and radar. In this paper, "lunar", "ocean_2kb1", and "pleiades_portdebouc_pan" are chosen. Moreover, the test image "Miramar NAS" is derived from the USCSIPI Image Database [50]. In addition, two other remote sensing images are chosen. "Pavia" is derived from an image acquired by the Quickbird sensor over Pavia, Italy, which has a resolution of 0.6 m. "Houston" is derived from an image acquired by the World View-2 sensor over Houston, USA in 2013, which has a resolution of 0.5 m. We crop the left upper part of these images with a size of 512×512 for comparison under the same condition. The test image set is shown in Figure 8. The data source and bit depth of the test image set are listed in Table 2.

Table 2. List of the test image set.

Image	Bit Depth (bpp)	Source—Copyright
lunar	8	Galileo Image—NASA
Miramar NAS	8	USC-SIPI
ocean_2kb1	10	NOAA Polar Orbiter (AVHRR)—NOAA
Pavia	11	QuickBird (0.6 m)
Houston	11	WorldView-2 Sensor(0.5 m)
pleiades_portdebouc_pan	12	Simulated PLEIADES—CNES

5.2. Performance Comparison of the Proposed Compression Method with Other Scan-Based Methods

In the experiments, the 9/7-tap biorthogonal wavelet filter frame is used for the directional adaptive lifting. The size of the image block is set to 64×64. All the test remote sensing images are compressed by the proposed compression methods and the other five scan-based methods, respectively. The methods used for comparison are the mainstream or the state-of-the-art compression methods, including CCSDS 122.0-B-1, JPEG2000, BTCA, HAS, and QCAO. The compared results, in terms of PSNR, MS-SSIM, and VIF, are tabulated in Tables 3–5. The results are evaluated at eight bit rates, namely, 0.0313 bpp, 0.0625 bpp, 0.125 bpp, 0.25 bpp, 0.5 bpp, 1 bpp, 2 bpp, and 3 bpp.

Table 3. PSNRs (dB) of the proposed compression method and other scan-based compression methods.

Image	Algorithm	Bit Rate (bpp)							
		0.0313	0.0625	0.125	0.25	0.5	1	2	3
lunar	CCSDS	9.86	18.92	24.45	27.64	30.68	35.07	35.07	41.10
	JPEG2000	22.73	24.11	26.04	28.25	31.35	35.61	41.53	46.21
	BTCA	23.28	24.64	26.49	28.74	31.71	36.13	42.47	48.45
	HAS	23.32	24.70	26.51	28.77	31.76	36.08	42.41	48.40
	QCAO	23.32	24.93	26.50	28.87	31.79	36.15	42.46	48.51
	Proposed	**23.40**	**25.26**	**26.63**	**29.42**	**31.83**	**36.29**	**42.60**	**48.62**
Miramar NAS	CCSDS	12.30	16.83	21.62	24.21	26.71	29.81	34.74	39.41
	JPEG2000	19.67	20.81	22.54	24.50	26.80	29.97	35.14	40.03
	BTCA	20.17	21.28	22.87	24.78	27.12	30.33	35.31	40.53
	HAS	20.15	21.35	22.85	24.79	27.19	30.29	35.15	40.24
	QCAO	20.37	21.57	22.90	25.05	27.32	30.33	35.30	40.56
	Proposed	**20.61**	**21.76**	**23.00**	**25.27**	**27.44**	**30.54**	**35.37**	**40.66**
ocean_2kb1	CCSDS	14.33	17.63	29.69	33.72	36.90	40.86	47.01	52.41
	JPEG2000	28.08	29.91	32.17	34.51	37.54	41.84	48.40	53.80
	BTCA	28.81	30.55	32.72	35.00	37.97	42.19	48.59	54.53
	HAS	28.79	30.59	32.60	34.98	37.99	42.13	48.51	54.49
	QCAO	28.85	30.53	32.78	34.96	37.90	42.26	48.64	54.57
	Proposed	**28.97**	**30.84**	**32.95**	**35.15**	**38.11**	**42.35**	**48.69**	**54.59**
Pavia	CCSDS	19.30	25.14	34.26	37.10	39.39	42.13	46.79	51.77
	JPEG2000	32.51	34.05	35.75	37.64	39.79	42.65	**47.74**	**52.97**
	BTCA	32.96	34.21	35.94	37.78	39.93	42.71	47.49	52.61
	HAS	32.90	34.33	35.95	37.73	39.94	42.71	47.34	52.38
	QCAO	33.41	34.65	36.29	37.85	39.86	42.62	47.40	52.58
	Proposed	**33.65**	**34.80**	**36.44**	**38.09**	**40.02**	**42.83**	47.57	52.70
Houston	CCSDS	15.95	21.73	31.69	35.45	38.35	41.75	46.63	51.37
	JPEG2000	29.94	31.73	33.52	35.92	38.65	42.17	**47.28**	**52.28**
	BTCA	30.34	31.99	33.89	36.07	38.69	42.23	47.08	52.04
	HAS	30.33	31.92	33.87	36.04	38.69	42.24	46.99	51.90
	QCAO	31.02	32.51	34.03	36.17	38.75	42.33	46.89	51.90
	Proposed	**31.20**	**32.65**	**34.15**	**36.24**	**38.79**	**42.39**	46.82	51.79
pleiades_portdebouc	CCSDS	13.20	17.93	29.24	34.21	38.20	42.87	48.99	54.05
	JPEG2000	27.61	29.77	31.87	34.89	38.28	42.64	48.99	54.12
	BTCA	28.47	30.40	32.88	35.57	39.23	43.52	49.31	54.42
	HAS	28.44	30.41	32.71	35.55	39.18	43.48	49.24	54.32
	QCAO	28.61	30.45	33.35	35.73	39.22	43.52	49.35	54.48
	Proposed	**28.73**	**30.59**	**33.49**	**35.92**	**39.39**	**43.65**	**49.67**	**54.76**

Table 4. MS-SSIMs (dB) of the proposed compression method and other scan-based compression methods.

Image	Algorithm	Bit Rate (bpp)							
		0.0313	**0.0625**	**0.125**	**0.25**	**0.5**	**1**	**2**	**3**
lunar	CCSDS	0.1630	0.6538	0.8215	0.9220	0.9672	0.9895	0.9895	0.9971
	JPEG2000	0.6702	0.7849	0.8716	0.9310	0.9649	0.9882	0.9968	0.9988
	BTCA	0.7211	0.8035	0.8779	0.9413	0.9720	0.9906	0.9979	0.9989
	HAS	0.7296	0.8082	0.8871	0.9474	0.9751	0.9919	0.9980	0.9991
	QCAO	0.7239	0.8241	0.8843	0.9445	0.9721	0.9908	0.9979	0.9994
	Proposed	**0.7374**	**0.8456**	**0.8902**	**0.9500**	**0.9789**	**0.9946**	**0.9985**	**0.9996**
Miramar NAS	CCSDS	0.2877	0.5683	0.8008	0.9148	0.9627	0.9848	0.9950	0.9980
	JPEG2000	0.6242	0.7395	0.8506	0.9222	0.9638	0.9850	0.9945	0.9980
	BTCA	0.6679	0.7726	0.8680	0.9250	0.9634	0.9857	0.9956	0.9982
	HAS	0.6761	0.7793	0.8601	0.9292	0.9679	0.9867	**0.9959**	**0.9986**
	QCAO	0.6945	0.7908	0.8721	0.9335	**0.9682**	0.9855	0.9956	0.9985
	Proposed	**0.6963**	**0.7917**	**0.8756**	**0.9379**	0.9681	**0.9871**	0.9956	**0.9986**
ocean_2kb1	CCSDS	0.1640	0.4233	0.8351	0.9319	0.9645	0.9849	0.9962	0.9986
	JPEG2000	0.7014	0.8015	0.8789	0.9265	0.9624	0.9833	0.9959	0.9986
	BTCA	0.7396	0.8121	0.8820	0.9355	0.9660	0.9863	0.9967	0.9987
	HAS	0.7267	0.8256	0.8869	0.9322	0.9647	0.9857	0.9966	0.9990
	QCAO	0.7425	0.8237	0.8917	0.9347	0.9659	0.9863	0.9964	0.9988
	Proposed	**0.7477**	**0.8337**	**0.8936**	**0.9361**	**0.9676**	**0.9878**	**0.9968**	**0.9991**
Pavia	CCSDS	0.1222	0.4259	0.7703	0.8912	0.9432	0.9731	0.9905	0.9969
	JPEG2000	0.5410	0.7008	0.8089	0.8899	0.9448	0.9742	0.9911	0.9970
	BTCA	0.5836	0.7046	0.8240	0.8939	0.9419	0.9731	0.9907	0.9970
	HAS	0.6080	0.7250	0.8185	0.8962	0.9484	**0.9759**	**0.9913**	**0.9971**
	QCAO	0.6427	0.7506	0.8440	0.9029	0.9456	0.9734	0.9903	0.9970
	Proposed	**0.6865**	**0.7625**	**0.8518**	**0.9057**	**0.9488**	0.9742	0.9906	**0.9971**
Houston	CCSDS	0.0993	0.4287	0.7617	0.8965	0.9506	0.9785	0.9925	0.9974
	JPEG2000	0.5608	0.7013	0.8173	0.8935	0.9489	0.9779	0.9922	0.9972
	BTCA	0.5937	0.7106	0.8054	0.8901	0.9497	0.9777	0.9924	0.9974
	HAS	0.5791	0.7021	0.8255	0.8957	**0.9512**	**0.9788**	**0.9928**	**0.9976**
	QCAO	0.6529	0.7608	0.8264	0.8953	0.9503	0.9774	0.9920	0.9973
	Proposed	**0.6626**	**0.7681**	**0.8271**	**0.8960**	0.9507	0.9769	0.9919	0.9973
pleiades_portdebouc	CCSDS	0.1084	0.2578	0.7476	0.9204	0.9726	0.9911	0.9972	0.9989
	JPEG2000	0.5768	0.7473	0.8572	0.9322	0.9725	0.9906	0.9970	0.9988
	BTCA	0.6469	0.7745	0.8755	0.9405	0.9765	0.9911	0.9973	0.9989
	HAS	0.6350	0.7889	0.8820	0.9413	0.9751	**0.9918**	0.9974	0.9990
	QCAO	0.6657	0.7891	0.8937	0.9419	0.9765	0.9912	0.9974	0.9990
	Proposed	**0.6737**	**0.7990**	**0.8946**	**0.9433**	**0.9775**	0.9917	**0.9977**	**0.9991**

Table 5. VIFs (dB) of the proposed compression method and other scan-based compression methods.

Image	Algorithm	Bit Rate (bpp)							
		0.0313	**0.0625**	**0.125**	**0.25**	**0.5**	**1**	**2**	**3**
lunar	CCSDS	0.0221	0.0552	0.1136	0.2553	0.4288	0.6622	0.6622	0.8668
	JPEG2000	0.0446	0.0892	0.1686	0.2830	0.4336	0.6777	0.8841	0.9552
	BTCA	0.0585	0.1026	0.1797	0.3074	0.4733	0.7149	0.9106	0.9749
	HAS	0.0550	0.1086	0.1856	0.3083	**0.4794**	**0.7235**	**0.9297**	**0.9817**
	QCAO	0.0571	0.1253	0.1836	0.3142	0.4703	0.7181	0.9101	0.9746
	Proposed	**0.0599**	**0.1521**	**0.1928**	**0.3405**	0.4792	0.7217	0.9086	0.9738
Miramar NAS	CCSDS	0.0206	0.0464	0.0977	0.2174	0.3610	0.5373	0.7494	0.8699
	JPEG2000	0.0355	0.0671	0.1326	0.2331	0.3696	0.5551	0.7426	0.8828
	BTCA	0.0452	0.0844	0.1468	0.2409	0.3669	0.5587	0.7808	0.9138
	HAS	0.0481	0.0834	0.1427	0.2474	0.3950	**0.5639**	**0.8098**	**0.9265**
	QCAO	0.0508	0.0893	0.1487	0.2511	0.3958	0.5573	0.7829	0.9146
	Proposed	**0.0527**	**0.0900**	**0.1527**	**0.2563**	**0.3976**	0.5583	0.7781	0.9125
ocean_2kb1	CCSDS	0.0205	0.0387	0.0887	0.2117	0.3473	0.5351	0.7693	0.8931
	JPEG2000	0.0403	0.0745	0.1339	0.2199	0.3387	0.5259	0.7960	0.9180
	BTCA	0.0506	0.0853	0.1465	0.2384	0.3654	0.5633	0.8011	0.9297
	HAS	0.0497	0.0904	0.1472	0.2348	0.3678	**0.5714**	**0.8293**	**0.9468**
	QCAO	0.0514	0.0860	0.1469	0.2387	0.3641	0.5623	0.7998	0.9281
	Proposed	**0.0525**	**0.1013**	**0.1488**	**0.2430**	**0.3694**	0.5576	0.7992	0.9294

Table 5. *Cont.*

Image	Algorithm	Bit Rate (bpp)							
		0.0313	0.0625	0.125	0.25	0.5	1	2	3
Pavia	CCSDS	0.0164	0.0355	0.0716	0.1622	0.2629	0.3957	0.5876	0.7606
	JPEG2000	0.0258	0.0514	0.0938	0.1667	0.2706	0.4106	0.6141	0.7771
	BTCA	0.0311	0.0559	0.1029	0.1708	0.2675	0.3985	0.5987	0.7849
	HAS	0.0340	0.0576	0.1010	0.1726	**0.2848**	**0.4363**	**0.6445**	**0.8081**
	QCAO	0.0489	0.0741	0.1259	0.1760	0.2769	0.4145	0.5949	0.7823
	Proposed	**0.0596**	**0.0805**	**0.1267**	**0.1777**	0.2760	0.4136	0.5921	0.7829
Houston	CCSDS	0.0152	0.0307	0.0633	0.1595	0.2707	0.4196	0.6161	0.7768
	JPEG2000	0.0272	0.0541	0.0949	0.1620	0.2763	0.4322	0.6218	0.7607
	BTCA	0.0326	0.0561	0.0960	0.1643	0.2749	0.4263	0.6305	0.7984
	HAS	0.0300	0.0568	0.1028	0.1693	**0.2847**	**0.4498**	**0.6638**	**0.8262**
	QCAO	0.0652	0.0774	0.1027	0.1685	0.2773	0.4208	0.6259	0.7946
	Proposed	**0.0697**	**0.0791**	**0.1039**	**0.1696**	0.2726	0.4176	0.6186	0.7906
pleiades_portdebouc	CCSDS	0.0103	0.0180	0.0462	0.1215	0.2176	0.3599	0.5682	0.7232
	JPEG2000	0.0220	0.0448	0.0764	0.1330	0.2242	0.3607	0.5467	0.7204
	BTCA	0.0301	0.0524	0.0899	0.1468	0.2384	0.3632	0.5496	0.7268
	HAS	0.0297	0.0552	0.0909	0.1475	0.2405	**0.3839**	0.5824	**0.7529**
	QCAO	0.0317	0.0534	0.1074	0.1479	0.2375	0.3652	0.5936	0.7105
	Proposed	**0.0361**	**0.0567**	**0.1092**	**0.1519**	**0.2437**	0.3776	**0.6072**	0.7299

PSNR is a quality index that evaluates an algorithm under the sense of a means squared error (MSE). In Table 3, the compared results are listed from the perspective of MSE. It can be seen that the coding performance of CCSDS is the worst. The reason for this is that the CCSDS is designed for onboard compression, which considers the complexity of the algorithm more than the coding performance. As a result, the low complexity of the algorithm is at the cost of the coding performance. The PSNR results of JPEG2000 are good, because some complicated models of JPEG2000, such as the context model and rate-distortion optimization, can help to provide a good coding performance. Compared with JPEG2000, BTCA can encode the neighbors of those significant coefficients in priority, which makes it more advantageous for the compression of remote sensing images, because more details can be preserved. Therefore, the PSNR results of BTCA are better than those of JPEG2000 when bit rate is not very high. The HAS algorithm is an improved version of BTCA that takes human vision into consideration, aiming to improve the visual quality of the reconstructed remote sensing images. It can be seen that from Table 3, the PSNR results of HAS are similar to those of BTCA. The QCAO method is the state-of-the-art compression method, which is an efficient image compression method based on quadtree, and it can provide quality, resolution, and position scalability. The PSNR results of QCAO are higher than those of the previously mentioned methods. The proposed compression method can provide the best coding performance in most cases. The reason is that the designed DAL-PBT model can provide a more efficient representation for remote sensing images, and the CQOT method can provide a reasonable scanning method among and within blocks based on image content, before establishing some quadtrees based on the scanning results. The effective image representation and appreciate scanning mode are all beneficial to the improvement of the coding performance. When the bit rate is very high, sometimes the JPEG2000 can provide a higher PSNR.

Table 4 compares the results in terms of the MS-SSIM. The MS-SSIM can provide a whole approximation to the perceived image quality from the view of structural similarity. From Table 4, we can see that the proposed compression method is still better than the other five compression methods in most cases, especially at low bit rates. The reason is that the DAL-PBT model can provide a good representation of the high frequency information of remote sensing images, and the CQOT method encodes the brothers of significant coefficients in priority during the coding process, which ensures that more important outlines and details of remote sensing images are preserved. Therefore, the structural similarity between the original image and the reconstructed image is improved. As the bit rate increases, some other compression methods, such as the HAS-based method, can sometimes provide a better result. Take the "Houston", for example; Table 4 shows

that the MS-SSIM results of the HAS-based method are 0.9512, 0.9788, 0.9928, and 0.9976 at 0.5 bpp, 1 bpp, 2 bpp, and 3 bpp, respectively. In comparison, the MS-SSIM results of the proposed method are 0.9507, 0.9769, 0.9919, and 0.9973, respectively. The slight performance improvements of the HAS-based method are caused by the visual weighing mask. Nevertheless, for most of the test images, the proposed compression method can still provide the best results at high bit rates.

Table 5 lists the comparison of the compression results in terms of VIF. The VIF is derived from a quantification of two mutual information quantities: the mutual information between the input and the output of the HVS channel for the reference image, and the mutual information between the input of the distortion channel and the output of the HVS channel for the reconstructed image. Therefore, VIF is a quality index that evaluates an algorithm from the perspective of human visual perception. Table 5 shows that the proposed compression method is superior to the other five compression methods in most cases. At a few high bit rates, the HAS method can provide a better visual quality for reconstructed images. The reason is that the HAS method is specially designed for image visualization applications. The retina-based visual weighting mask of the HAS aims to preserve more visually sensitive information of the image.

Table 6 shows the encoding times(s) of these compression methods. The results are the average encoding time for six test images at three bit rates. These programs are evaluated on a PC with Intel Core i7-3770 CPU @ 3.40 GHz and 32 GB memory.

Table 6. The encoding times(s) of different compression methods.

Algorithm	1 bpp	0.5 bpp	0.25 bpp
CCSDS	1.78	0.93	0.41
JPEG2000	2.12	1.19	0.53
BTCA	0.30	0.27	0.25
HAS	0.37	0.33	0.31
QCAO	0.28	0.25	0.23
Proposed	1.29	1.16	0.98

In Table 6, we can see that BTCA is faster than CCSDS with entropy coding. The encoding time of QCAO is slightly shorter than that of BTCA. The reason is that, compared with binary tree, the establishment and scanning of quadtree need fewer comparison times. For HAS, it takes some time to calculate the retina-based visual weighting mask, so its encoding time is longer than that of BTCA. The proposed method is an improved version of QCAO. Compared with QCAO, the proposed method takes a lot of time to calculate the optimum prediction directions of those blocks and directional interpolation. Therefore, the encoding time of the proposed method is longer than that of QCAO, and sometimes even longer than that of JPEG2000. Thus, the proposed method is not applicable to onboard compression, and it can be used for occasions with low real-time requirements and high compression efficiency.

In order to compare the visual quality of reconstructed images obtained by the different compression methods, some experiments are carried out. Here, we provide the visual quality of reconstructed images obtained by the proposed compression method and the mainstream JPEG2000. The comparison results of the image "Lunar" at 0.0625 bpp are shown in Figure 9. It can be seen that the reconstructed image with the proposed method is clearer than that with JPEG2000 in the red box. Figure 10 gets a similar result with the image "Ocean" at 0.125 bpp.

Based on the analysis above, we can conclude that, compared with other scan-based methods, the proposed compression method can provide a better coding performance. One reason is that the designed DAL-PBT model can provide a more efficient representation for remote sensing images. Another reason is that the CQOT method provides a reasonable scanning method among and within subbands/blocks based on image content, which ensures that the important information of the image

can be scanned in priority. The effective image representation and the adaptive scanning-based coding method are all beneficial to the improvement of the coding performance.
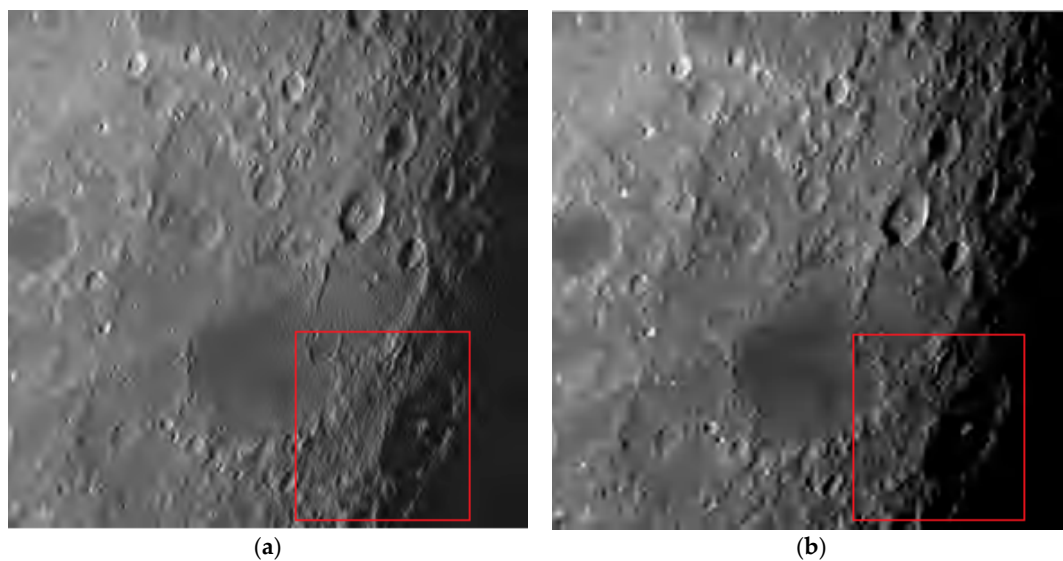

Figure 9. The comparison results of two compression algorithms with the image "Lunar" at 0.0625 bpp. (**a**) Reconstructed image with the proposed method and (**b**) reconstructed image with JPEG2000.
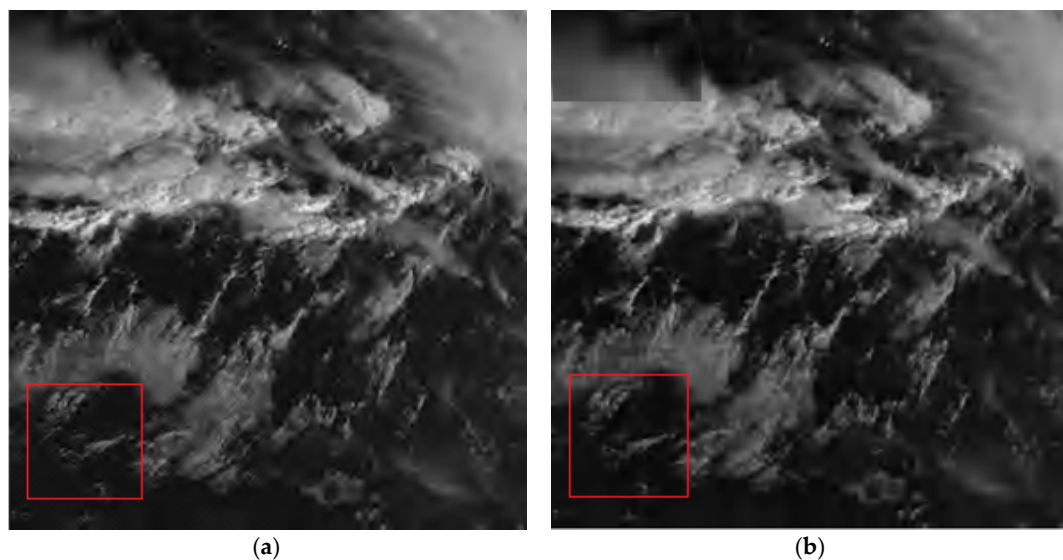

Figure 10. The comparison results of two compression algorithms with the image "Ocean" at 0.125 bpp. (**a**) Reconstructed image with the proposed method and (**b**) reconstructed image with JPEG2000.

6. Conclusions

In this paper, we present a new compression framework for remote sensing images. It combines the directional adaptive lifting partitioned block transform with content-driven quadtree codec with optimized truncation. The image representation phase and coding phase are closely related; the former aims to provide a good representation for more directional information of remote sensing images, and the latter focuses on the adaptive scanning mode of the transformed image blocks for further improving the coding efficiency. The experiments are carried out with the images from different sensors (Galileo Image, NOAA Polar Orbiter, QuickBird, WorldView-2, and Simulated PLEIADES) and covering different scenes (lunar, ocean, city, and suburbs). Experimental results show that the

proposed method outperforms the mainstream JPEG2000 up to 2.41 dB in PSNR. Compared with the latest QCAO method, up to 0.55 dB improvement in PSNR is reported. Moreover, some improvements in subjective quality are also observed. The proposed method also supports various progressive transmission modes, such as the quality progressive mode, resolution progressive mode, and position progressive mode. Since the block coding is independent, the proposed method can also support the region of interest (ROI)-based progressive transmission mode.

Multicomponent images, such as multispectral and hyperspectral images, are also very popular in remote sensing applications. Therefore, a natural idea is to apply the proposed method to the compression of multicomponent images. The proposed method employs block coding, which can be combined with parallel computing to improve compression efficiency. Actually, we are now doing this work for the compression of hyperspectral images. In addition, for the compression of hyperspectral images, another issue that should be considered is the bit allocation strategy among those transformed components. In our next work, we will research this.

References

1. Zhang, L.; Lv, X.; Liang, X. Saliency Analysis via Hyperparameter Sparse Representation and Energy Distribution Optimization for Remote Sensing Images. *Remote Sens.* **2017**, *9*, 636. [CrossRef]
2. Shi, C.; Zhang, J.; Zhang, Y. Content-based onboard compression for remote sensing images. *Neurocomputing* **2016**, *191*, 330–340. [CrossRef]
3. Shapiro, J.M. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Process.* **1993**, *41*, 3445–3462. [CrossRef]
4. Said, A.; Pearlman, W.A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.* **1996**, *6*, 243–250. [CrossRef]
5. Pearlman, W.A.; Islam, A.; Nagaraj, N.A. Said, Efficient low complexity image coding with a set-partitioning embedded block coder. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 1219–1235. [CrossRef]
6. Christopoulos, C.; Skodras, A.; Ebrahimi, T. The JPEG2000 still image coding system: An overview. *IEEE Trans. Consumer Electron.* **2000**, *46*, 1103–1127. [CrossRef]
7. Wu, Z.; Bilgin, A.; Marcellin, M.W. Joint source/channel coding for image transmission with JPEG2000 over memoryless channels. *IEEE Trans. Image Process* **2005**, *14*, 1020–1032. [PubMed]
8. Suruliandi, A.; Raja, S.P. Empirical evaluation of EZW and other encoding techniques in the wavelet-based image compression domain. *Int. J. Wavelets Mult.* **2015**, *13*, 1550012. [CrossRef]
9. Zhang, Y.; Cao, H.; Jiang, H. Mutual information-based context template modeling for bitplane coding in remote sensing image compression. *J. Appl. Remote Sens.* **2016**, *10*, 025011. [CrossRef]
10. Hamdi, M.; Rhouma, R.; Belghith, S. A selective compression-encryption of images based on SPIHT coding and Chirikov Standard Map. *Signal Process.* **2017**, *131*, 514–526. [CrossRef]
11. Zhang, L.; Li, A.; Zhang, Z.; Yang, K. Global and local saliency analysis for the extraction of residential areas in high-spatial-resolution remote sensing image. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3750–3763. [CrossRef]

12. CCSDS122.0-B-1. Image Data Compression. November 2005. Available online: http://public.ccsds.org/publications/archive/122x0b1c3.pdf (accessed on 20 May 2018).
13. García-Vílchez, F.; Serra-Sagristà, J. Extending the CCSDS recommendation for image data compression for remote sensing scenarios. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 3431–3445.
14. CCSDS122.0-B-2. Image Data Compression. November 2017. Available online: https://public.ccsds.org/Pubs/122x0b2.pdf (accessed on 20 May 2018).
15. Kulkarni, P.; Bilgin, A.; Marcellin, M.W.; Dagher, J.C.; Kasner, J.H.; Flohr, T.J.; Rountree, J.C. Compression of earth science data with JPEG2000. In *Hyperspectral Data Compression*; Springer: Boston, MA, USA, 2006; pp. 347–378. ISBN 978-0-387-28579-5.
16. Li, B.; Yang, R.; Jiang, H.X. Remote-sensing image compression using two-dimensional oriented wavelet transform. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 236–250. [CrossRef]
17. Candes, E.J.; Donoho, D.L. New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities. *Commun. Pure. Appl. Math.* **2004**, *57*, 219–266. [CrossRef]
18. Do, M.N.; Vetterli, M. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Trans. Image Process.* **2005**, *14*, 2091–2106. [CrossRef] [PubMed]
19. Velisavljevic, V.; Beferull-Lozano, B.; Vetterli, M.; Dragotti, P.L. Directionlets: Anisotropic multidirectional representation with separable filtering. *IEEE Trans. Image Process.* **2006**, *15*, 1916–1933. [CrossRef] [PubMed]
20. Guo, K.; Labate, D. Optimally sparse multidimensional representation using shearlets. *SIAM J. Math. Anal.* **2007**, *39*, 298–318. [CrossRef]
21. Wang, C.; Lin, H.; Jiang, H. CANS: Towards Congestion-Adaptive and Small Stretch Emergency Navigation with Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2016**, *15*, 1077–1089. [CrossRef]
22. Shi, C.; Zhang, J.; Chen, H.; Zhang, Y. A Novel Hybrid Method for Remote Sensing Image Approximation Using the Tetrolet Transform. *IEEE J. Sel. Top. Appl. Earth Observ.* **2014**, *7*, 4949–4959. [CrossRef]
23. Hou, X.; Jiang, G.; Ji, R.; Shi, C. Directional lifting wavelet and universal trellis coded quantization based image coding algorithm and objective quality evaluation. In Proceedings of the IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 501–504.
24. Hou, X.; Yang, J.; Jiang, G.; Qian, X.M. Complex SAR image compression based on directional lifting wavelet transform with high clustering capability. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 527–538. [CrossRef]
25. Hou, X.; Zhang, L.; Gong, C. SAR image Bayesian compressive sensing exploiting the interscale and intrascale dependencies in directional lifting wavelet transform domain. *Neurocomputing* **2014**, *133*, 358–368. [CrossRef]
26. Zhang, L.; Qiu, B. Edge-preserving image compression using adaptive lifting wavelet transform. *Int. J. Electron.* **2015**, *102*, 1190–1203. [CrossRef]
27. Yan, W.; Shaker, A. El-Ashmawy N. Urban land cover classification using airborne LiDAR data: A review. *Remote Sens. Environ.* **2015**, *158*, 295–310. [CrossRef]
28. Hussain, A.J.; Al-Jumeily, D.; Radi, N. Hybrid Neural Network Predictive-Wavelet Image Compression System. *Neurocomputing* **2015**, *151*, 975–984. [CrossRef]
29. Zhang, L.; Chen, J.; Qiu, B. Region-of-Interest Coding Based on Saliency Detection and Directional Wavelet for Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 23–27. [CrossRef]
30. CCSDS122.1-B-1. Spectral Preprocessing Transform For Multispectral and Hyperspectral Image Compression. November 2017. Available online: https://public.ccsds.org/Pubs/122x1b1.pdf (accessed on 20 May 2018).
31. 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]
32. Valsesia, D.; Boufounos, P.T. Universal encoding of multispectral images. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20–25 March 2016; pp. 4453–4457.
33. Wang, H.; Celik, T. Sparse Representation-Based Hyperspectral Data Processing: Lossy Compression. *IEEE J. Sel. Topics Appl. Earth Observ.* **2017**, *99*, 1–10. [CrossRef]
34. Fu, W.; Li, S.; Fang, L.; Benediktsson, J. Adaptive Spectral–Spatial Compression of Hyperspectral Image with Sparse Representation. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 671–682. [CrossRef]
35. Van, L.P.; De Praeter, J.; Van Wallendael, G.; Van Leuven, S.; De Cock, J.; Van de Walle, R. Efficient Bit Rate Transcoding for High Efficiency Video Coding. *IEEE Trans. Multimed.* **2016**, *18*, 364–378.
36. Huang, K.K.; Liu, H.; Ren, C.X.; Yu, Y.F.; Lai, Z.R. Remote sensing image compression based on binary tree and optimized truncation. *Digit. Signal Process.* **2017**, *64*, 96–106. [CrossRef]

37. Barret, M.; Gutzwiller, J.L.; Hariti, M. Low-Complexity Hyperspectral Image Coding Using Exogenous Orthogonal Optimal Spectral Transform (OrthOST) and Degree-2 Zerotrees. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 1557–1566. [CrossRef]

38. Karami, A. Lossy compression of hyperspectral images using shearlet transform and 3D SPECK. In Proceedings of the Image and Signal Processing for Remote Sensing XXI, Toulouse, France, 15 October 2015; Volume 9643.

39. Song, X.; Huang, Q.; Chang, S.; He, J.; Wang, H. Three-dimensional separate descendant-based SPIHT algorithm for fast compression of high-resolution medical image sequences. *IET Image Process.* **2016**, *11*, 80–87. [CrossRef]

40. Huang, K.; Dai, D. A New On-Board Image Codec Based on Binary Tree with Adaptive Scanning Order in Scan-Based Mode. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3737–3750. [CrossRef]

41. Shi, C.; Zhang, J.; Zhang, Y. A Novel Vision-Based Adaptive Scanning for the Compression of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1336–1348. [CrossRef]

42. Liu, H.; Huang, K.; Ren, C.; Yu, Y.; Lai, Z. Quadtree Coding with Adaptive Scanning Order for Space-borne Image Compression. *Signal Process. Image Commun.* **2017**, *55*, 1–9. [CrossRef]

43. Antonini, M.; Barlaud, M.; Mathieu, P.; Daubechies, I. Image coding using wavelet transform. *IEEE Trans. Image Process.* **1992**, *1*, 205–220. [CrossRef] [PubMed]

44. Chang, C.L.; Girod, B. Direction-adaptive discrete wavelet transform for image compression. *IEEE Trans. Image Process.* **2007**, *16*, 1289–1302. [CrossRef] [PubMed]

45. Sweldens, W. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.* **1998**, *29*, 511–546. [CrossRef]

46. Ding, W.; Wu, F.; Wu, X.; Li, S.; Li, H. Adaptive directional lifting-based wavelet transform for image coding. *IEEE Trans. Image Process.* **2007**, *16*, 416–427. [CrossRef] [PubMed]

47. Liu, Y.; Ngan, K.N. Weighted adaptive lifting-based wavelet transform for image coding. *IEEE Trans. Image Process.* **2008**, *17*, 500–511. [CrossRef] [PubMed]

48. Sheikh, H.R.; Bovik, A.C. Image information and visual quality. *IEEE Trans. Image Process.* **2006**, *15*, 430–444. [CrossRef] [PubMed]

49. CCSDS Reference Test Image Set. April 2007. Available online: http://cwe.ccsds.org/sls/docs/sls-dc/ (accessed on 15 March 2018).

50. USC-SIPI Database. Available online: http://sipi.usc.edu/database (accessed on 15 March 2018).

检 索 报 告

一、检索要求

  1. 委 托 人：石翠萍 (Shi, CP；Shi, Cuiping)

  2. 委托单位：齐齐哈尔大学

  3. 检索目的：论文被收录情况

二、检索范围
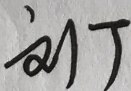
Science Citation Index Expanded (SCI-EXPANDED)	1975-present	网络版

三、检索结果

委托人提供的1篇论文被SCI-E数据库收录，详细情况见附件一。

特此证明！

东北师范大学科技查新咨询中心

教育部科技查新工作站（L24）

（盖 章）

检索报告人：刘丁

2018 年 9 月 3 日

附件一： SCI-E收录情况

地址: [Shi, Cuiping; Miao, Fengjuan; He, Peng] Qiqihar Univ, Coll Commun & Elect Engn, Qiqihar 161000, Peoples R China.

[Shi, Cuiping; Wang, Liguo] Harbin Engn Univ, Coll Informat & Commun Engn, Harbin 150001, Heilongjiang, Peoples R China.

[Zhang, Junping] Harbin Inst Technol, Coll Elect & Commun Engn, Harbin 150001, Heilongjiang, Peoples R China.

通讯作者地址: Shi, CP (通讯作者)，Qiqihar Univ, Coll Commun & Elect Engn, Qiqihar 161000, Peoples R China.

Shi, CP (通讯作者)，Harbin Engn Univ, Coll Informat & Commun Engn, Harbin 150001, Heilongjiang, Peoples R China.

电子邮件地址: scp1980@126.com; wangliguo@hrbeu.edu.cn; zhangjp@hit.edu.cn; miaofengjuan@163.com; hphgy@163.com

The End