Signal Processing: Image Communication 55 (2017) 1–9

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Quadtree coding with adaptive scanning order for space-borne image compression ☆

Hui Liu[a], Ke-Kun Huang[a,*], Chuan-Xian Ren[b], Yu-Feng Yu[b], Zhao-Rong Lai[c]

[a] School of Mathematics, JiaYing University, Meizhou, Guangdong 514015, China
[b] Intelligent Data Center and Department of Mathematics, Sun Yat-Sen University, Guangzhou 510275, China
[c] Department of Mathematics, College of Information Science and Technology, JiNan University, Guangzhou 510632, China

ARTICLE INFO

Keywords:
Remote sensing image compression
Quadtree coding
Rate-distortion optimization
Adaptive scanning order

ABSTRACT

Space-borne equipments produce very big images while their capacities of storage, calculation and transmission are limited, so low-complexity image compression algorithms are necessary. In this paper, we develop an efficient image compression algorithm based on quadtree in wavelet domain for this mission. First, we propose an adaptive scanning order for quadtree, which traverses prior the neighbors of previous significant nodes from bottom to the top of quadtree, so that more significant coefficients are encoded at a specified bit rate. Second, we divide the entire wavelet image to several blocks and encode them individually. Because the distortion-rate usually decreases as the level of the quadtree increases with the adaptive scanning order, to control bit rate for each block, we set the points exactly after coding each level of the quadtree as the candidate truncation points. The proposed method can provide quality, position and resolution scalability, which is simple and fast without any entropy coding, so it is very suitable for space-borne equipments. Experimental results show that it attains better performance compared with some state-of-the-art algorithms.

1. Introduction

With the rapid development of sensor technology on space-borne equipments, high spatial resolution remote sensing images are becoming more easily acquired. Since the images will take up a great deal of storage space, and the hardware of space-borne equipments is limited, great efforts are made to seek low-complexity image compression algorithms. The main requirements to image compression techniques for a considered application are two folds. First, the distortion between original image and decoding image needs as small as possible at a certain specified compression ratio. Second, the algorithm's complexity needs as low as possible.

Because of the high energy compaction property of wavelet transform, most image compression algorithms are based on set partition. A class of successful methods exploit the property of self-similarity across scales in wavelet image, such as Embedded Zerotree Wavelet (EZW) [1], Set Partitioning in Hierarchical Tree (SPIHT) [2] and Set Partitioned Embedded Block (SPECK) [3]. These methods construct zero-tree through spatial similarity of the wavelet coefficients at different scales in each bit-plane so that a large number of non-significant coefficients are predicted successfully by a root of zero-tree.

Other than the zero-tree methods, some methods utilize the clustering characteristic within each subband. The Quadtree Coding (QC) is a successful method that takes the advantage, and has been reported in some literatures [4–6]. QC is based on hierarchical set partitioning. There is only one node at the top level of quadtree, which consists of all coefficients. If there is any significant coefficient in a node, then it splits into 2×2 descendent nodes that are two dimensional homogeneous blocks. We can perform the splitting recursively to quickly locate a significant coefficient. Given the splitting information, the same significance map can be reproduced by the decoder. Through the bit-plane coding, QC produces a lossless and embedded data stream, supports quality scalability, and permits region-of-interest coding. The excellent performance for this scheme is presented in [6].

In recent years, some compression schemes that are specifically designed for remote sensing images have been proposed. The Consultative Committee for Space Data Systems (CCSDS) published an Image Data Compression (CCSDS-IDC) [7], which specifically targets space-borne equipments, and focuses more on compression and less on options of handling and distributing compressed data. Some extensions

☆ This work is supported in part by National Science Foundation of China under Grants 61403164 and 61572536, in part by the Fundamental Research Funds for the Central Universities under Grant 16lgzd16, and in part by the Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (2013LYM_0085).

* Corresponding author.
E-mail addresses: imlhxm@163.com (H. Liu), kkcocoon@163.com (K.-K. Huang), rchuanx@mail.sysu.edu.cn (C.-X. Ren), yuyufeng220@163.com (Y.-F. Yu), laizhr@jnu.edu.cn (Z.-R. Lai).

http://dx.doi.org/10.1016/j.image.2017.03.011
Received 6 July 2016; Received in revised form 16 March 2017; Accepted 16 March 2017
Available online 18 March 2017
0923-5965/© 2017 Elsevier B.V. All rights reserved.

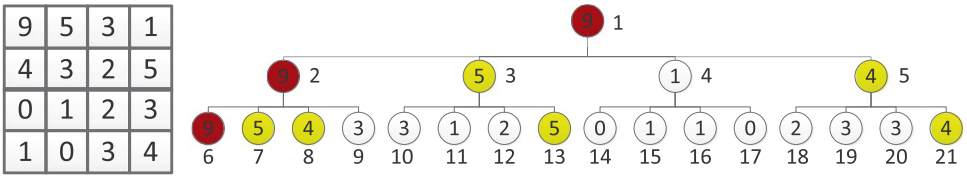


Fig. 1. A simple example to illustrate the proposed method. Left: a block to be encoded. Right: The corresponding quadtree. The number in the tree node indicates the value of corresponding coefficient, and the number on the right of the tree node denotes the index of the node.

have been presented for the CCSDS-IDC [8,9]. JPEG2000 [10] is a standard image compression algorithm based on wavelet transform, and it is also extended for space-borne equipments. In [11], the wavelet transform in JPEG2000 is replaced by a two-dimensional oriented wavelet transform and it outperforms JPEG2000 for remote sensing images. Kulkarni et al. [12] presented a scan-based method that can use JPEG2000 with incrementally acquired data. However, some components of JPEG2000 that help to provide high compression performance also have high implementation complexity, which limits JPEG2000 to be a standard for airborne mission.

Besides JPEG2000 and CCSDS-IDC, the other compression algorithms are also applied for remote sensing image. In [13], a modified listless strip based SPIHT is proposed to reduce system complexity and minimize processing time and memory usage. In [14], an improved SPIHT is proposed for multispectral image compression for various band images with high resolution. In [15,16], two improved compressions of multispectral images based on classified transform coding are proposed, which are superior to that of the original class-based multispectral image coder [17]. In [18], the lapped transform and Tucker decomposition are proposed for hyperspectral image compression. In [19], a compression algorithm of hyperspectral remote sensing images is proposed, which introduces a tensor decomposition technology to approximately decompose the original tensor data into a core tensor multiplied by a factor matrix. In [20], a new efficient region-based scheme for the compression of multispectral remote-sensing image is proposed, which outperforms some state-of-the-art methods. In [21], an overview of several standards for remote sensing data compression is provided, discussing both mono-band and multi-band compression, and lossless, lossy and near-lossless compression. In [22], a novel wavelet-based scheme to increase coefficient independence in hyperspectral images is introduced for lossless coding, which uses multivariate regression to exploit the relationships among wavelet transformed components, and outperforms some most recent coding methods.

The tree-based partitioning methods are also applied for space-borne equipments. In [23], a SAR complex image data compression algorithm based on QC in wavelet transform domain is proposed, showing that QC achieves the best performance for SAR complex image compression. In [24], the quadtree in QC is replaced by the binary tree, which achieves better performance. In [25], a Human vision-based Adaptive Scanning (HAS) for the compression of remote sensing image is proposed, which generates an importance weighting mask according to the human visual characteristics. In [26], a content-based adaptive scanning scheme is proposed for remote sensing image compression, which provides different scanning orders among and within subbands.

However, the performance of QC is not good enough. We find that the neighbors of the previous significant coefficients are more likely significant at current bit-plane, so we develop an adaptive scanning order for quadtree to achieve higher compression ratio. Moreover, QC needs a lot of memory to store the quadtree for large-scale images, and does not provide random access property, so we divide the entire wavelet image to several blocks and encode them individually, then apply rate-distortion optimization to achieve higher performance.

Our main contributions can be listed as follows:

- We propose an adaptive scanning order for quadtree, which traverses prior the neighbors of previous significant nodes from bottom to the top of quadtree, so that more significant coefficients are encoded at a specified bit rate.
- We divide the entire wavelet image to several blocks and encode them individually, then apply rate-distortion optimization to achieve higher performance.

The remainder of the paper is organized as follows: in Section 2, we describe the proposed method in detail. The experimental results are given in Section 3. Finally, the conclusion is provided in Section 4.

2. The proposed method

2.1. Quadtree coding

The Quadtree Coding (QC) takes the advantage of the clustering characteristic within each subband, and has been reported in some literatures [4–6]. The core of this approach is based on hierarchical set partitioning. The bottom level of the quadtree consists of all coefficients. Each node of the next higher level is then set to the maximum value of its four child nodes. In Fig. 1, there is a 4×4 wavelet image, and the bottom level of the quadtree consists of 16 nodes. The higher level consists of 4 nodes, where the first red node (9) is the maximum of the four red coefficients at the bottom level (9,5,4,3).

We provide the notation for the quadtree tree as follows. Suppose there is a wavelet image with size of $L_0 = 2^N \times 2^N$, which is transformed into one-dimensional vector using Morton scanning order, denoted by $I(t)$ for $t = 1, 2, \ldots, L_0$.

From bottom to the top, we construct the quadtree $\Psi(t)$ for $1 \le t \le L_0$. The number of nodes at the bottom level of the quadtree is equal to L_0, and the number of nodes at the next upper level is equal to $\frac{L_0}{4}$, and so on. The depth of the quadtree is $\frac{\log(L_0)}{\log(4)} = N$, and the total number of nodes of the quadtree is $L = \sum_{d=0}^{N} 4^d$.

The bottom level of the quadtree consists of all the coefficients:

$$\Psi(t) = I(t - (L - L_0)) \quad \text{for } L - L_0 + 1 \le t \le L. \tag{1}$$

The upper levels of the tree are defined iteratively:

$$\Psi(t) = \max\{|\Psi(4t-2)|, |\Psi(4t-1)|, |\Psi(4t)|, |\Psi(4t+1)|\} \quad \text{for } 1 \le t \le L - L_0. \tag{2}$$

After constructing the quadtree, we can traverse the tree by depth-first. For each bit plane, going from top to the bottom level of the quadtree, if a node is insignificant with respect to the current threshold, then it is coded by "0", otherwise it is coded by "1", and the process is recursively applied to the four children. If the process reaches the bottom level of the tree and the corresponding node is significant, then the sign of the coefficient is coded. We can neglect some codes as follows:

- If one node is a previous significant node, then this node must be significant with the current threshold.
- If one node is significant and its three children are insignificant, then the last child must be significant.

The process can be expressed as a function $code = QC(\Psi, t, T_k)$ as Algorithm 1, where t is a tree node index of the quadtree, and T_k is a threshold where $T_0 = 2^{\lfloor \log_2 \Psi(1) \rfloor}$ and $T_k = T_0/2^k$. The algorithm uses a stack

to avoid recursive function.

Algorithm 1. Function $code = QC(\Psi, j, T_k)$.

```
code = {};
Stack S; StackInit(S); Push(S, j);
while !Empty(S) do
  j = Pop(S);
  if |Ψ(j)| ≥ 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 jmod4 = 1 and |Ψ(j − 1)| < T_k and |Ψ(j − 2)| < T_k
  and |Ψ(j − 3)| < T_k then
  if 4j + 1 ≤ L then
  Push(S, 4j + 1);
  Push(S, 4j);
  Push(S, 4j − 1);
  Push(S, 4j − 2);
  else
  code = code ∪ sign(Ψ(j)).
  end if
  else if |Ψ(j)| ≥ T_k then
  code = code ∪ {1};
  if 4j + 1 ≤ L then
  Push(S, 4j + 1);
  Push(S, 4j);
  Push(S, 4j − 1);
  Push(S, 4j − 2);
  else
  code = code ∪ {sign(Ψ(j))}.
  end if
  else
  code = code ∪ {0};
  end if
end while
```

2.2. Adaptive scanning coding

In quadtree, if a node has a significant parent with larger threshold, in other words, some brothers of the node are significant previously, we find that this node is very likely to be significant with the current threshold. Moreover, the lower the level of the quadtree is, the higher the probability to be significant is. Fig. 2 gives an example. We construct a quadtree for a block of a wavelet image. We first find the nodes whose absolute values are greater than the threshold 64. With the threshold 32, for each node at the bottom level of the tree, if there are some previous significant brothers whose absolute values are greater than 64, then the node is collected to be traversed first. We find that 31% of these coefficients are significant with the threshold 32, i.e., the significance probability is 31%. At the higher levels of the tree, the significance probabilities are 27%, 20% and 14%, respectively. With the threshold 16, the significance probability decreases as the level of the tree increases too. Because at the higher levels of the tree, the leafs of the sub-trees rooted at the brothers of significant nodes are farther from the significant coefficients, so the significance probability decreases and we can traverse the quadtree from bottom to the top level.

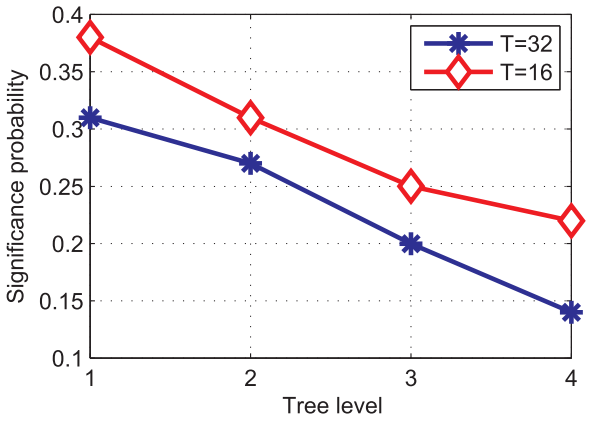


Fig. 2. The significance probability versus tree level. With each threshold, the significance probability decreases as the level of the tree increases.

In other words, we propose to scan the neighbors of the previous significant coefficients before other regions are scanned. The scanning order are adaptively determined by the previous significant nodes. The detail steps of the proposed method can be described as a function $code = QCA(\Psi, T_k)$ as Algorithm 2.

Algorithm 2. Function $code = QCA(\Psi, T_k)$.

```
d=N;
e=L;
b = e − 4^d + 1;
code = {};
while b > 1 do
  j=b;
  while j < e do
  if max|Ψ(j: j + 3)| ≥ 2T_k then
  if |Ψ(j)| < 2T_k then
  code = code ∪ QC(Ψ, j, T_k);
  end if
  if |Ψ(j + 1)| < 2T_k then
  code = code ∪ QC(Ψ, j + 1, T_k);
  end if
  if |Ψ(j + 2)| < 2T_k then
  code = code ∪ QC(Ψ, j + 2, T_k);
  end if
  if |Ψ(j + 3)| < 2T_k then
  code = code ∪ QC(Ψ, j + 3, T_k);
  end if
  end if
j = j + 4;
 end while
d = d − 1;
e = b − 1;
b = e − 4^d + 1;
end while
```

Fig. 1 gives the adaptive scanning order for the quadtree of a 4×4 block. Suppose the red nodes and the yellow ones are the previous significant nodes whose values are no less than 4. Then with the threshold $T=2$, the adaptive scanning order is defined as follows. At the bottom level, the 9th, 10th, 11th, 12th, 18th, 19th, 20th nodes are scanned first, because their brothers contain some previous significant nodes, while the other nodes at the bottom level should not be scanned at this time. At the next upper level, the subtree rooted with the 4th node is encoded.

2.3. Block coding and optimization

If we directly use QCA to encode the entire wavelet image, then it needs a lot of memory to store the quadtree and does not provide random access property. We can divide the wavelet image into several blocks and encode them by QCA independently. To control bit rate for each block, we can optimize the distortion-rate for each block. That is,

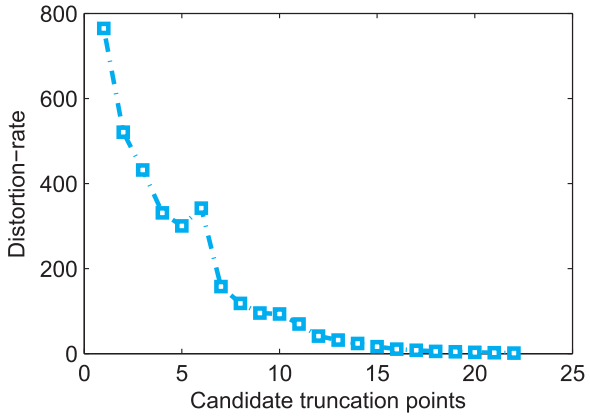


Fig. 3. Distortion-rates of the candidate truncation points of the proposed method.

we first select some coding stages at each block as the valid truncation points, such that the corresponding local distortion-rates are strictly decreasing for each block. Then we select a final truncation point for each block such that the rate reaches the specified value and the distortion-rate is maximized.

In the process of the distortion-rate optimization [10,27,28], how to select the truncation points is a key problem. We find that with the adaptive scanning order, the distortion-rate usually decreases as the level of the quadtree increases, so we can set the points exactly after coding each level as the candidate truncation points.

Let

$$S_j = \frac{D_{j-1} - D_j}{R_j - R_{j-1}}$$

denote the local distortion-rate for a block, where R_j is the bit rate and D_j is the corresponding distortion at jth candidate truncation point. Fig. 3 shows distortion-rates of the candidate truncation points when using the proposed method to encode a block with size of $2^6 \times 2^6$.

We select some of the candidate truncation points of the independent block such that the distortion is minimized at a specified bit rate. If $S_{j+1} > S_j$, jth candidate truncation point cannot be selected, and we need to update

$$S_{j+1} = \frac{D_{j-1} - D_{j+1}}{R_{j+1} - R_{j-1}} \tag{3}$$

and compare it with S_{j-1}. This process is performed until $S_{j+1} < S_{j_k}$. We can select the strictly decreasing S_{j_k} as a series of valid truncation points. Fig. 4 shows distortion-rates of the remaining valid truncation points. Because S_j is usually monotonically decreasing, it remains almost all candidate truncation points in Fig. 3. Then we find a final truncation point for each block, such that the rate reaches the specified value and the distortion-rate is maximized. The number of the final truncation points is equal to the number of blocks, so there are only a little extra storage to record the truncation points.

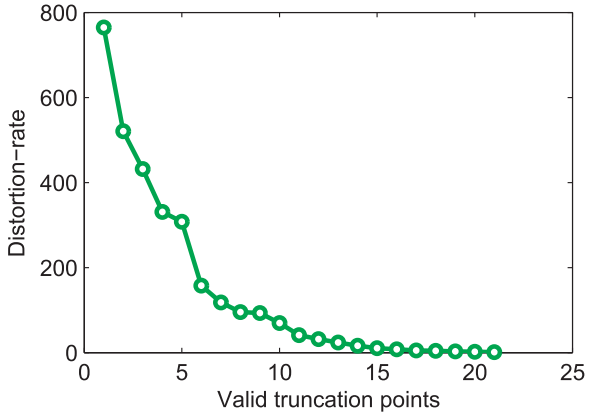


Fig. 4. Distortion-rates of the valid truncation points of the proposed method.

Table 1
The candidate truncation points of the proposed method for the example in Fig. 1.

No.	T	Level	D	R	S
0	–	–	210	0	Inf
1	8	–	138	10	7.2
2	4	2	102	15	7.2
3	4	1	66	26	3.3
4	4	–	58	27	8.0
5	2	2	16	40	3.2
6	2	1	16	41	0.0
7	2	–	7	46	1.8

Suppose the blocks are of size $2^n \times 2^n$, where $N = pn$. There are $2^p \times 2^p = 4^p$ blocks. For each block, there are $2^n \times 2^n = 2^{2n}$ coefficients consisting of the bottom level of the quadtree. There are $\sum_{d=0}^{n} 4^d$ elements in the quadtree for each block, while there are $\sum_{d=0}^{N} 4^d$ elements in the quadtree for QCA. The memory requirement of the quadtree for a wavelet image is about 4^p times than that for a block. So, applying the rate-distortion optimization, the proposed method requires much less memory. Because the proposed method exports the code length of each block, the decoder can decode any specified blocks so that random access property is attained. As long as we give multiple bit rates and record the code length of each block at each bit rate, the proposed method can provide the quality scalability.

The proposed method is called *Q*uadtree *C*oding with *A*daptive scanning order and *O*ptimized truncation, denoted by *QCAO*.

2.4. A simple calculation example

To better illustrate the proposed method, we give a simple example. Suppose that there is a 4×4 block whose coefficients are shown in the left side of Fig. 1. The block is transformed into one-dimensional vector with Morton scanning order, resulting $V = \{9, 5, 3, 1; 4, 3, 2, 5; 0, 1, 2, 3; 1, 0, 3, 4\}$. Then we can construct the quadtree according to Eqs. (1) and (2), as shown in the right side of Fig. 1. Table 1 shows the candidate truncation points for coding the block. We initialize the distortion by $D_0 = \sum_{i,j} x_{i,j}^2 = 210$, the rate $R_0 = 0$ and the distortion-rate $S_0 = \inf$. Then we first traverse the quadtree by depth-first with $T = 8$. The resulting code is "1110000000", and the reconstructed coefficients are 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, respectively, so $R_1 = 10$, $D_1 = 138$ and $S_1 = \frac{D_1 - D_0}{R_1 - R_0} = 7.2$.

Then we traverse the quadtree from bottom level to top level with $T = 4$. For each level, if the brother of a coefficient is significant with $T = 8$, then we prefer to traverse the coefficient and its descendants, i.e., with $T = 4$, we perform $QC(\Psi, 7, 4)$, $QC(\Psi, 8, 4)$, $QC(\Psi, 9, 4)$ for level = 2, $QC(\Psi, 3, 4)$, $QC(\Psi, 4, 4)$, $QC(\Psi, 5, 4)$ for level = 1 in turn. After traversing with $T = 4$, we perform magnitude refinement pass for all the previous significant coefficients, as No. 4 candidate truncation point in Table 1.

With $T = 2$, we traverse the coefficients of No. 9, 10, 11, 12, 18, 19,

Table 2
The valid truncation points of the proposed method for the example in Fig. 1.

No.	T	Level	D	R	S
0	–	–	210	0	Inf
1	8	–	138	10	7.2
2	4	2	102	15	7.2
4	4	–	58	27	3.7
5	2	2	16	40	3.2
7	2	–	7	46	1.5

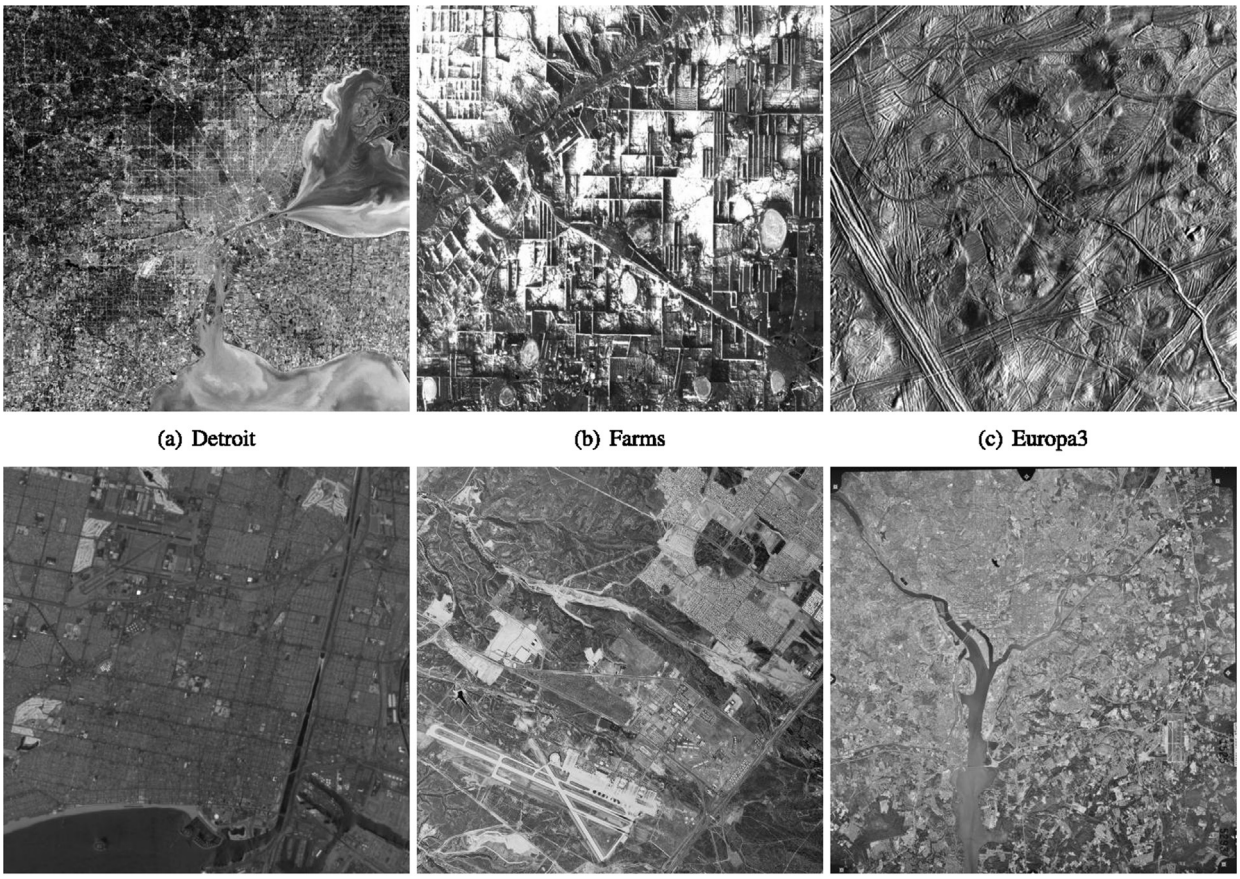


Fig. 5. Space-borne images for experiment.

20 and 4 by function *QC* in turn i.e., we traverse the neighbors of the yellow coefficients in Fig. 1 first. After traversing with $T=2$, we get 10 candidate truncation points, as shown in Table 1.

After the candidate truncation points are attained, we need to calculate the valid truncation points. Because $S_4 > S_3$, the No. 3 candidate truncation point cannot be selected as a valid one, and we need to update $S_4 = \frac{D_2 - D_4}{R_4 - R_2} = \frac{102-58}{27-15} = 3.7$. Because $S_7 > S_6$, we need to update $S_7 = \frac{D_5 - D_7}{R_7 - R_5} = \frac{16-7}{46-40} = 1.5$. This process can be repeated and we can get all the valid truncation points, as shown in Table 2.

3. Experimental results

In this section, to evaluate the performance of the proposed method, some experiments are conducted on three space-borne image sets: the Landsat images [29], the CCSDS image test corpus [30] and the USC-SIPI Image Database [31]. The Landsat images were produced by the Landsat-7 flight, including eight spectral bands and one panchromatic band. We select two images, i.e., Detroit and Farms, for experiment, as shown in Fig. 5(a), (b). The CCSDS reference test image set includes a variety of space imaging instrument data such as solar, stellar, planetary, earth observations, etc. We select Europa3 and SpotLAb3 for experiment, as shown in Fig. 5(c), (d). The USC-SIPI image database contains four volumes. The Aerials volume of the database consists of 38 images. We select Miramar NAS and Washington for experiment, as shown in Fig. 5(e), (f).

We convert these images into gray ones and size of $512 \times 512 \times 8$ before compression. In addition, the 9/7-tap biorthogonal wavelet filters [32] are used for wavelet transform. The size of each block of the proposed QCAO method is 64×64. The Peak-Signal-to-Noise-Ratios (PSNR) for different Compression Ratios (CR) of our method are compared to those of SPIHT [2], SPECK [3], CCSDS [7], JPEG2000 [12], BTCA [24] and BTCA-S [24]. The Matlab source code for BTCA and QCAO are available.[1]

The PSNR and CR are expressed by the following relations [33]:

$$PSNR(dB) = 10\log_{10}\frac{255^2}{MSE}, \tag{4}$$

$$MSE = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}(x_{ij} - y_{ij})^2}{W \times H}, \tag{5}$$

$$CR(bpp) = \frac{\text{number of coded bits}}{W \times H}, \tag{6}$$

where x and y denote the original image and reconstructed one, respectively, and the images are of size $W \times H$.

Besides PSNR, we also use the Structural Similarity Index Measure (SSIM) to evaluate the performance. SSIM is correlated with the quality perception of the human visual system [34], which is defined as:

$$SSIM = l(x, y)c(x, y)s(x, y), \tag{7}$$

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2\mu_y^2 + C_1}, \tag{8}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2\sigma_y^2 + C_2}, \tag{9}$$

[1] **Matlab code:** http://www.mathworks.com/matlabcentral/profile/authors/5133554-ke-kun-huang/

Table 3
PSNRs of the proposed QCAO method and other algorithms at different bit rates for different images.

Image	Algorithm	PSNR					
		1 bpp	0.5 bpp	0.25 bpp	0.125 bpp	0.0625 bpp	0.0313 bpp
Detroit	SPIHT	27.27	24.01	21.78	20.31	19.07	17.93
	SPECK	27.40	24.30	22.24	20.72	19.68	18.98
	CCSDS	25.53	22.52	20.49	18.86	15.26	10.55
	JPEG2000	26.12	22.82	20.83	19.40	18.44	17.60
	BTCA	27.59	24.42	22.30	20.77	19.72	18.95
	BTCAS	27.50	24.33	22.33	20.75	19.73	18.94
	QCAO	**27.59**	**24.68**	**22.44**	**22.51**	**19.85**	**19.63**
Farms	SPIHT	26.92	22.80	20.13	17.99	16.38	14.71
	SPECK	26.82	23.01	20.46	18.53	17.08	15.98
	CCSDS	26.85	22.89	19.97	16.87	11.73	8.784
	JPEG2000	**27.48**	23.37	**20.73**	18.46	16.83	15.14
	BTCA	26.98	23.10	20.57	18.58	17.08	15.92
	BTCAS	26.96	23.11	20.56	18.58	17.05	15.90
	QCAO	27.00	**23.52**	20.65	**18.70**	**17.54**	**16.21**
Europa3	SPIHT	26.77	23.09	20.69	19.01	17.82	16.57
	SPECK	26.99	23.29	21.04	19.54	18.35	17.63
	CCSDS	26.89	23.06	20.50	18.30	12.47	10.30
	JPEG2000	**27.42**	**23.54**	21.18	19.43	18.06	17.15
	BTCA	27.17	23.47	21.17	19.58	18.48	17.61
	BTCAS	27.14	23.47	21.13	19.56	18.44	17.59
	QCAO	27.23	23.48	**21.39**	**20.09**	**18.59**	**17.75**
SpotLAb3	SPIHT	39.19	35.29	32.35	29.90	27.58	24.73
	SPECK	39.24	35.53	32.72	30.60	28.97	27.63
	CCSDS	39.34	35.49	32.55	29.88	25.54	13.24
	JPEG2000	**39.76**	35.67	32.87	30.51	28.65	26.68
	BTCA	39.69	35.84	33.00	30.81	29.08	27.66
	BTCAS	39.50	35.74	32.91	30.80	29.05	27.61
	QCAO	39.60	**35.84**	**33.21**	**30.86**	**30.08**	**27.96**
MiramarNAS	SPIHT	28.67	25.99	24.27	22.69	21.21	19.37
	SPECK	28.66	26.12	24.47	23.09	21.95	21.01
	CCSDS	28.78	26.20	24.38	22.38	14.47	10.75
	JPEG2000	**29.28**	**26.46**	24.57	22.99	21.70	20.32
	BTCA	29.08	26.32	24.60	23.22	22.01	21.05
	BTCAS	29.02	26.39	24.59	23.21	22.00	21.03
	QCAO	29.12	26.40	**24.61**	**23.30**	**22.34**	**21.14**
Washington	SPIHT	29.61	26.66	24.65	22.85	21.23	19.49
	SPECK	29.69	26.88	24.91	23.33	22.06	21.08
	CCSDS	29.75	26.94	24.73	22.44	16.65	10.39
	JPEG2000	**30.34**	27.19	24.94	23.23	21.76	20.45
	BTCA	30.07	27.21	25.06	23.45	22.14	21.11
	BTCAS	29.96	27.10	25.01	23.43	22.12	21.12
	QCAO	30.09	**27.29**	**25.50**	**23.52**	**22.63**	**21.24**

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}, \tag{10}$$

where the detail parameters can be found in [34].

Tables 3 and 4 list the PSNRs and SSIMs of the proposed method and other algorithms at six bit rates for different images, respectively. From the results, we can draw the following conclusions:

- The performance of CCSDS is the lowest, because it balances the complexity and PSNR.
- The performance of JPEG2000 is much better than SPIHT or SPECK. However, JPEG2000 is too complex to be a standard for airborne mission.
- BTCA outperforms SPIHT and SPECK, and is better than JPEG2000 at lower bit rates, which proves that BTCA is state-of-the-art methods.
- BTCA-S sacrifices the compression ratio in order to provide position scalability, so the performance is worse than BTCA.
- The proposed QCAO achieves the highest performance. For Detroit image, the PSNR of QCAO increases an average of 1.06, 0.56, 3.91, 1.92, 0.49 and 0.52 dB than SPIHT, SPECK, CCSDS, JPEG2000, BTCA and BTCA-S respectively. For all the six testing images, the PSNR of QCAO increases an average of 0.96, 0.38, 3.28, 0.62, 0.24, 0.28 dB than the above methods, respectively.

In order to strengthen the conclusions, we consider a larger set of test images to evaluate the performance. Here, all the 40 images from CCSDS reference testing image set are used [30]. These images include 14 images with 8 bit-depth such as “costal-b1”, “marstest” and “lunar”, and 18 images with 10 bit-depth such as “ice-2kb1”, “‘india-2kb1”, “india-2kb4”, “ocean-2kb1”, “landesV-G7-10b” and “marseille-G6-10b”, 6 images with 12 bit-depth and 2 images with 16 bit-depth. Table 5 lists average PSNR and average SSIM for all the testing images of different methods at six bit rates. We can find that the mean PSNR of CCSDS is always the lowest of all methods. The proposed method is the highest of all methods at 0.5 bpp, 0.25 bpp, 0.125 bpp, 0.0625 bpp and 0.0313 bpp. At 1 bpp, the mean PSNR of QCAO is only a little smaller than JPEG2000. Regarding the SSIM criterion, the similar results are attained. Specially, the mean SSIM of QCAO and that of JPEG2000 are equal at 1 bpp, which proves that QCAO can generate good visual effect even the PSNR is not the best.

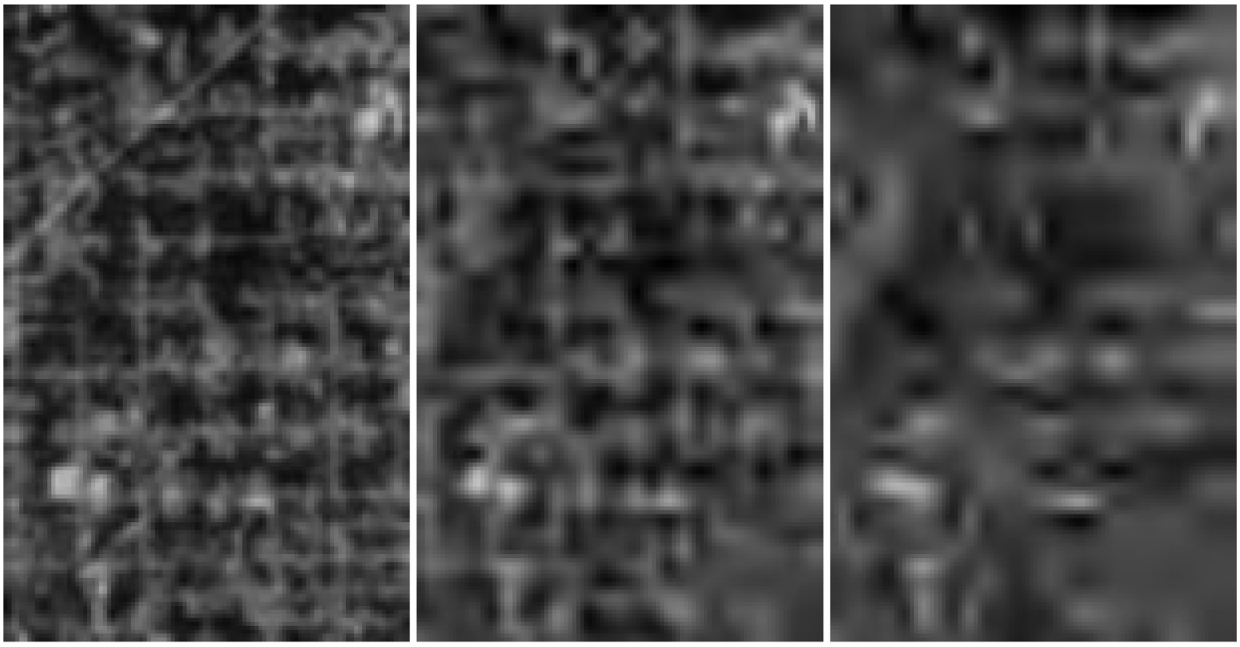
To give a visual effect, Fig. 6 shows the comparison of two compression algorithms with the image of Detroit at 0.5 bpp. We can find that the reconstructed image with QCAO is clearer than that with

Table 4
SSIMs of the proposed QCAO method and other algorithms at different bit rates for different images.

Image	Algorithm	SSIM					
		1 bpp	0.5 bpp	0.25 bpp	0.125 bpp	0.0625 bpp	0.0313 bpp
Detroit	SPIHT	0.9755	0.9471	0.9093	0.8669	0.8145	0.7449
	SPECK	0.9763	0.9507	0.9192	0.8805	0.8441	0.8097
	CCSDS	0.9661	0.9299	0.8835	0.8186	0.6153	0.3649
	JPEG2000	0.9704	0.9349	0.8925	0.8453	0.7972	0.7431
	BTCA	0.9773	0.9521	0.9206	0.8821	0.8459	0.8084
	BTCAS	0.9768	0.9508	0.9196	0.8811	0.8441	0.8062
	QCAO	**0.9773**	**0.9550**	**0.9219**	**0.9240**	**0.8494**	**0.8389**
Farms	SPIHT	0.9862	0.9639	0.9319	0.8848	0.8249	0.7149
	SPECK	0.9860	0.9657	0.9372	0.8994	0.8527	0.8034
	CCSDS	0.9860	0.9645	0.9283	0.8407	0.5589	0.2711
	JPEG2000	**0.9879**	0.9684	**0.9408**	0.8957	0.8418	0.7465
	BTCA	0.9865	0.9665	0.9389	0.9012	0.8536	0.8003
	BTCAS	0.9864	0.9665	0.9387	0.8990	0.8520	0.7959
	QCAO	0.9865	**0.9696**	0.9401	**0.9023**	**0.8694**	**0.8122**
Europa3	SPIHT	0.9666	0.9192	0.8536	0.7723	0.6695	0.5050
	SPECK	0.9683	0.9235	0.8669	0.7995	0.7223	0.6493
	CCSDS	0.9675	0.9184	0.8441	0.7068	0.3147	0.1817
	JPEG2000	**0.9713**	**0.9277**	0.8682	0.7917	0.6871	0.5832
	BTCA	0.9696	0.9266	0.8712	0.8012	0.7314	0.6459
	BTCAS	0.9694	0.9266	0.8662	0.8004	0.7211	0.6425
	QCAO	0.9700	0.9268	**0.8763**	**0.8268**	**0.7337**	**0.6614**
SpotLAb3	SPIHT	0.9913	0.9784	0.9569	0.9221	0.8595	0.7079
	SPECK	0.9917	0.9799	0.9608	0.9350	0.9016	0.8631
	CCSDS	0.9916	0.9794	0.9588	0.9214	0.7928	0.2089
	JPEG2000	**0.9924**	0.9803	0.9619	0.9327	0.8930	0.8217
	BTCA	0.9922	**0.9810**	0.9631	0.9380	0.9041	0.8623
	BTCAS	0.9919	0.9806	0.9623	0.9371	0.9033	0.8597
	QCAO	0.9921	0.9810	**0.9649**	**0.9381**	**0.9253**	**0.8740**
MiramarNAS	SPIHT	0.9741	0.9511	0.9252	0.8896	0.8387	0.7311
	SPECK	0.9741	0.9527	0.9288	0.9005	0.8658	0.8287
	CCSDS	0.9747	0.9535	0.9270	0.8787	0.4770	0.1473
	JPEG2000	**0.9774**	**0.9559**	0.9305	0.8964	0.8563	0.7926
	BTCA	0.9765	0.9550	0.9310	0.9037	0.8678	0.8315
	BTCAS	0.9761	0.9551	0.9309	0.9019	0.8676	0.8281
	QCAO	0.9767	0.9552	**0.9312**	**0.9041**	**0.8784**	**0.8321**
Washington	SPIHT	0.9707	0.9412	0.9023	0.8460	0.7623	0.5979
	SPECK	0.9714	0.9445	0.9088	0.8652	0.8092	0.7504
	CCSDS	0.9717	0.9447	0.9041	0.8253	0.4799	0.1247
	JPEG2000	**0.9752**	0.9475	0.9091	0.8585	0.7905	0.6966
	BTCA	0.9737	0.9478	0.9121	**0.8692**	0.8128	0.7533
	BTCAS	0.9728	0.9462	0.9105	0.8651	0.8118	0.7469
	QCAO	0.9738	**0.9487**	**0.9212**	0.8691	**0.8360**	**0.7579**

Table 5
Average PSNR and average SSIM for all the images from CCSDS reference set of different methods at six bit rates.

Criterion	Algorithm	1 bpp	0.5 bpp	0.25 bpp	0.125 bpp	0.0625 bpp	0.0313 bpp
PSNR	SPIHT	32.24	28.62	26.09	24.10	22.39	20.43
	SPECK	32.33	28.81	26.43	24.64	23.28	22.23
	CCSDS	32.14	28.56	25.97	23.49	18.05	12.59
	JPEG2000	**32.77**	28.97	26.44	24.45	22.89	21.44
	BTCA	32.69	29.13	26.62	24.78	23.37	22.26
	BTCAS	32.56	29.01	26.56	24.75	23.33	22.22
	QCAO	32.71	**29.18**	**26.79**	**25.11**	**23.86**	**22.59**
SSIM	SPIHT	0.9821	0.9590	0.9260	0.8805	0.8154	0.6860
	SPECK	0.9825	0.9611	0.9322	0.8955	0.8520	0.8050
	CCSDS	0.9814	0.9581	0.9226	0.8546	0.5894	0.2372
	JPEG2000	**0.9837**	0.9617	0.9311	0.8889	0.8340	0.7572
	BTCA	0.9836	0.9633	0.9349	0.8985	0.8554	0.8060
	BTCAS	0.9832	0.9624	0.9335	0.8965	0.8525	0.8018
	QCAO	**0.9837**	**0.9638**	**0.9373**	**0.9068**	**0.8675**	**0.8207**



(a) Part of original "Detroit" image (b) Reconstructed image with QCAO (c) Reconstructed image with SPIHT

Fig. 6. Comparison of two compression algorithms with the image of Detroit at 0.5 bpp. The reconstructed image with QCAO is clearer than that with SPIHT. There are more textures in the reconstructed image with QCAO than those with SPIHT.

Table 6
The encoding speed for different methods.

Methods	Coding time (s)		
	0.25	0.50	1.00
SPIHT	0.09	0.10	0.12
SPECK	0.10	0.12	0.14
CCSDS	0.28	0.75	1.56
JPEG2000	0.34	0.95	1.99
BCTA	0.16	0.18	0.21
BCTAS	0.17	0.19	0.23
QCAO	0.15	0.16	0.19

SPIHT. There are more textures in the reconstructed image with QCAO than those with SPIHT. Because the number of coded significant coefficients with QCAO is 12 584, while it is 11 627 with SPIHT. QCAO encodes more significant coefficients than SPIHT.

Table 6 shows the encoding speed of different methods. We can find that the encoding speeds of QCAO and SPIHT are similar, and QCAO is much faster than CCSDS. Specially, the encoding process of JPEG2000 is the lowest, which limits JPEG2000 to be a standard for space-borne equipments.

4. Conclusion

In this paper, an efficient compression method is proposed for space-borne image. We first propose an adaptive scanning order for quadtree coding, then we divide the entire wavelet image to several blocks and encode them individually. To control bit rate for each block, we set the points exactly after coding each level of the quadtree as the candidate truncation points. The proposed method gets higher compression ratio, less memory requirement and fast encoding speed.

References

[1] J.M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. Signal Process. 41 (12) (1993) 3445–3462.

[2] A. Said, W.A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Syst. Video Technol. 6 (3) (1996) 243–250.

[3] W.A. Pearlman, A. Islam, N. Nagaraj, A. Said, Efficient, low-complexity image coding with a set-partitioning embedded block coder, IEEE Trans. Circuits Syst. Video Technol. 14 (11) (2004) 1219–1235.

[4] P. Strobach, Quadtree-structured recursive plane decomposition coding of images, IEEE Trans. Signal Process. 39 (6) (1991) 1380–1397.

[5] R.L. Joshi, V.J. Crump, T.R. Fischer, Image subband coding using arithmetic coded trellis coded quantization, IEEE Trans. Circuits Syst. Video Technol. 5 (6) (1995) 515–523.

[6] A. Munteanu, J. Cornelis, G.V.D. Auwera, P. Cristea, Wavelet image compression – the quadtree coding approach, IEEE Trans. Inf. Technol. Biomed. 3 (3) (1999) 176–185.

[7] [Online], Image Data Compression, November 2005. Ser. Blue Book, CCSDS 122.0-b-1 ⟨http://public.ccsds.org/publications/BlueBooks.aspx⟩.

[8] F. Garcia-Vilchez, J. Serra-Sagrista, Extending the CCSDS recommendation for image data compression for remote sensing scenarios, IEEE Trans. Geosci. Remote Sens. 47 (10) (2009) 3431–3445.

[9] X.S. Hou, J. Yang, G.F. Jiang, X.M. Qian, Complex sar image compression based on directional lifting wavelet transform with high clustering capability, IEEE Trans. Geosci. Remote Sens. 51 (1) (2013) 527–538.

[10] D. Taubman, High performance scalable image compression with EBCOT, IEEE Trans. Image Process. 9 (7) (2000) 1158–1170.

[11] B. Li, R. Yang, H.X. Jiang, Remote-sensing image compression using two-dimensional oriented wavelet transform, IEEE Trans. Geosci. Remote Sens. 49 (1) (2011) 236–250.

[12] P. Kulkarni, A. Bilgin, M.W. Marcellin, J.C. Dagher, J.H. Kasner, T.J. Flohr, J.C. Rountree, Compression of Earth Science Data with JPEG2000, in: Hyperspectral Data Compression, Springer US, Boston, MA, 2006, pp. 347–378.

[13] H. ZainEldin, M.A. Elhosseini, H.A. Ali, A modified listless strip based SPIHT for wireless multimedia sensor networks, Comput. Electr. Eng. 56 (11) (2016) 519–532, http://dx.doi.org/10.1016/j.compeleceng.2015.10.001.

[14] V.B. Raju, K.J. Sankar, C.D. Naidu, S. Bachu, Multispectral image compression for various band images with high resolution improved DWT SPIHT, Int. J. Signal Process. Image Process. Pattern Recognit. 9 (2) (2016) 271–286.

[15] M. Cagnazzo, L. Cicala, G. Poggi, L. Verdoliva, Low-complexity compression of multispectral images based on classified transform coding, Signal Process.: Image Commun. 21 (10) (2006) 850–861.

[16] M. Cagnazzo, S. Parrilli, G. Poggi, L. Verdoliva, Improved class-based coding of multispectral images with shape-adaptive wavelet transform, IEEE Geosci. Remote Sens. Lett. 4 (4) (2007) 566–570.

[17] G. Gelli, G. Poggi, Compression of multispectral images by spectral classification and transform coding, IEEE Trans. Image Process. 8 (4) (1999) 476–489.

[18] L. Wang, J. Bai, J.J. Wu, G. Jeon, Hyperspectral image compression based on lapped transform and tucker decomposition, Signal Process.: Image Commun. 36 (8) (2015) 63–69.

[19] L. Zhang, L. Zhang, D. Tao, X. Huang, B. Du, Compression of hyperspectral remote sensing images by tensor approach, Neurocomputing 147 (2015) 358–363.

[20] M. Cagnazzo, G. Poggi, L. Verdoliva, Region-based transform coding of multispectral images, IEEE Trans. Image Process. 16 (12) (2008) 2916–2926.

[21] I. Blanes, E. Magli, J. Serra-Sagrista, A tutorial on image compression for optical space imaging systems, IEEE Geosci. Remote Sens. Mag. 2 (3) (2014) 8–26.
[22] N. Amrani, J. Serra-Sagristà, V. Laparra, M.W. Marcellin, Regression wavelet analysis for lossless coding of remote-sensing data, IEEE Trans. Geosci. Remote Sens. 54 (9) (2016) 1–12.
[23] X.S. Hou, M. Han, C. Gong, X.M. Qian, SAR complex image data compression based on quadtree and zerotree coding in discrete wavelet transform domain: a comparative study, Neurocomputing 148 (2015) 561–568.
[24] K.K. Huang, D.Q. Dai, A new on-board image codec based on binary tree with adaptive scanning order in scan-based mode, IEEE Trans. Geosci. Remote Sens. 50 (10) (2012) 3737–3750.
[25] C.P. Shi, J.P. Zhang, Y. Zhang, A novel vision-based adaptive scanning for the compression of remote sensing images, IEEE Trans. Geosci. Remote Sens. 54 (3) (2016) 1336–1348.
[26] C.P. Shi, J.P. Zhang, Y. Zhang, Content-based onboard compression for remote sensing images, Neurocomputing 191 (2016) 330–340.
[27] A. Fiengo, G. Chierchia, M. Cagnazzo, B. Pesquet-Popescu, Rate allocation in predictive video coding using a convex optimization framework, IEEE Trans. Image Process. 26 (1) (2017) 479–489.
[28] C. Greco, I.D. Nemoianu, M. Cagnazzo, B. Pesquet-Popescu, Rate-distortion-optimized multi-view streaming in wireless environment using network coding, EURASIP J. Adv. Signal Process. 2016 (1) (2016) 17.
[29] [Online], United States Geological Survey, landsat project website ⟨http://landsat.usgs.gov/⟩.
[30] [Online], Consultative committee for space data systems, CCSDS image test ⟨http://cwe.ccsds.org/sls/docs/sls-dc/⟩.
[31] [Online], The USC-SIPI image database ⟨http://sipi.usc.edu/database/⟩.
[32] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Trans. Image Process. 1 (2) (1992) 205–220.
[33] D.J. Granrath, The role of human visual models in image processing, Proc. IEEE 69 (5) (1981) 552–561.
[34] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 8 (03) (2004) 600–612.

Hui Liu received the B.Sc. degree and the M.Sc. degree in mathematics from South China Normal University, Guangzhou, China, in 2000 and 2006, respectively. She is currently an Associate Professor with the Department of Mathematics, Jiaying University, Meizhou, China. Her current research interests include fractal and image processing.

Ke-Kun Huang received the B.Sc., M.Sc and Ph.D. degrees in applied mathematics from Sun Yat-sen University, Guangzhou, China, in 2002, 2005 and 2016, respectively. He is currently an Associate Professor with the Department of Mathematics, Jiaying University, Meizhou, China. His research interests include image processing and face recognition.

Chuan-Xian Ren received the Ph.D. degree from the Faculty of Mathematics and Computing, Sun Yat-Sen University, Guangzhou, China, in 2010. He is currently an Associate Professor of the Faculty of Mathematics and Computing, Sun Yat-sen University. His current research interests include image processing and face recognition.

Yu-Feng Yu received the B.Sc. degree in mathematics from Shangrao Normal University, Shangrao, China, in 2011, and the M.Sc. degree in mathematics from Shenzhen University, Shenzhen, China, in 2014. He is currently pursuing the Ph.D. degree in statistics in Sun Yat-Sen University. His research interests include statistical optimization and pattern recognition.

Zhao-Rong Lai received the B.Sc. degree in mathematics, the M.Sc. degree in computational science, and the Ph.D. degree in statistics from Sun Yat-sen University, Guangzhou, China, in 2010, 2012, and 2015, respectively. He is currently with the Department of Mathematics, JiNan University. His research interests include face recognition and statistical optimization.