

A new method for compression of remote sensing images based on an enhanced differential pulse code modulation transformation

Pedram Ghamisi^{a,*}, Farshid Sepehrband^b, Lalit Kumar^c, Micael S. Couceiro^d,
Fernando M.L. Martins^{e,f}

^a Faculty of Electrical and Computer Engineering, University of Iceland, Saemundargotu 2, 101 Reykjavik, Iceland

^b Centre for Advanced Imaging, University of Queensland, Brisbane, Australia

^c Ecosystem Management, School of Environmental and Rural Science, University of New England, Armidale NSW 2351, Australia

^d RoboCorp, Engineering Institute of Coimbra, Pedro Nunes, 3030-199 Coimbra, Portugal

^e Instituto de Telecomunicações, Universidade de Coimbra, Polo II, 3030-290 Coimbra, Portugal

^f RoboCorp, Coimbra College of Education of Coimbra, Dom João III Solum, 3030-329 Coimbra, Portugal

*Corresponding author, e-mail: p.ghamisi@gmail.com

Received 24 Dec 2012

Accepted 17 Oct 2013

ABSTRACT: Remote sensing sensors generate useful information about climate and the Earth's surface, and are widely used in resource management, agriculture, and environmental monitoring. Compression of the RS data helps in long-term storage and transmission systems. Lossless compression is preferred for high-detail data, such as from remote sensing. In this paper, a less complex and efficient lossless compression method for images is introduced. It is based on improving the energy compaction ability of prediction models. The proposed method is applied to image processing, RS grey scale images, LiDAR rasterized data, and hyperspectral images. All the results are evaluated and compared with different lossless JPEG and a lossless version of JPEG2000, thus confirming that the proposed lossless compression method leads to a high speed transmission system because of a good compression ratio and simplicity.

KEYWORDS: remote sensing, lossless compression, LiDAR technology, hyperspectral images, enhanced DPCM transform

INTRODUCTION

Remote sensing (RS) images have been widely used in numerous applications, such as resource management, agriculture, environmental monitoring, mineral exploration, and climate observation. Some types of sensors can generate more than 1 TB of data in one day. Hence the use of a robust data compression techniques has become very important for archiving and transferring purposes^{1,2}. Because of the importance of generating highly accurate information about the atmosphere, clouds, and surface parameters provided by the RS sensors, lossy compression techniques are not desirable³. The economics of transmission and mass-storage of the large volumes of data accumulated by these sensors demonstrate that efficient compression is very important in this technology⁴. In addition, in real time applications, a simple and efficient algorithm accelerates the execution time.

Lossless compression generally consists of transformation and entropy encoding^{4,5}. There are many

methods for lossless compression. Although most of these methods are efficient in terms of compression they have a high computational complexity, or vice-versa. For example, JPEG2000 has a powerful compression mechanism but at the cost of a high computational complexity. In this paper, a less complex method for lossless compression of RS data is proposed based on a new transformation method called enhanced differential pulse code modulation (DPCM) transformation (EDT) and optimized Huffman encoding. The EDT improves the energy compaction ability of the traditional DPCM. The proposed method is efficient in terms of compression and fast in terms of implementation. Finally, our proposed method is applied to different types of images and the results are evaluated.

BACKGROUND

We start by reviewing lossless JPEG and the lossless version of JPEG2000. JPEG is a very famous

A	B	C	1. $X=(B+D)/2$	4. $X=D$	7. $X=B+(D-A)/2$
D	X		2. $X=B+D-A$	5. $X=B$	8. $X=(A+B+C+D)/4$
			3. $X=D+(B-A)/2$	6. $X=A$	9. $X=(B+C)/4+D/2$

Fig. 1 Neighbouring pixels in DPCM and prediction equations⁶.

ISO/ITU-T standard that was created in the late 1980s⁷. Lossless JPEG is one of several JPEG standards. In its lossless mode, the image is transformed by DPCM, and then Huffman encoding is applied. The DPCM is based on predicting the image pixels from the neighbouring pixels with a certain error of prediction. The neighbouring position and predictor equations are shown in Fig. 1⁸.

JPEG2000 is based on the discrete wavelet transform (DWT), scalar quantization, context modelling, arithmetic coding, and post-compression rate allocation⁹. JPEG2000 works well and gives a good compression ratio, especially for high-detail images, because it analyses the details and the approximation in the transformation step and decorrelates them. However, this does not come without a high computational complexity.

PROPOSED METHOD

Fig. 2 illustrates the whole idea of the proposed compression method. The new transformation method, EDT, which is described in the next section, is first applied to the input image. Then, an optimized Huffman entropy encoder is applied to encode the transformed image to a compressed bit-stream. Huffman decoding and inverse transformations are then used to reconstruct the input image.

Transformation (EDT)

EDT is based on predictive models. However, more energy compaction is obtained by improving the prediction ability. Energy compaction and low complexity are two important attributes of any good transformation⁸. From a hardware implementation point of view, the transformation method is obtained by adding only two shift registers to the previous DPCM. Hence the complexity of the EDT stays low. Our new transformation method engine is illustrated in the block diagram in Fig. 3. Division is first applied to the input image. Then, the quotients of the division are predicted by one of the prediction equations of Fig. 1. Equation (1) can be used as the prediction step. Furthermore, the predicted matrix is rescaled in the multiplication step. By adding the predicted matrix to the remainders of the division step, the predicted

image is produced. The subtraction of the predicted image from the original image gives the prediction error matrix which is the transformed image^{6,10}.

$$X = \begin{cases} \min(B, D), & \text{if } A \leq \max(B, D) \\ \max(B, D), & \text{if } A \geq \max(B, D) \\ D + B - A, & \text{otherwise.} \end{cases} \quad (1)$$

When intensities of an image are in the range of $[0, 2^k - 1]$, the image is of k -bit depth. Typical greyscale images are of 8–16 bit depth¹¹. RGB images consist of the R, G, and B matrices, each of which has the same depth as a greyscale image. As shown in Fig. 3, the input image is divided by n and the rounded quotient and the remainder of the division are kept in Quotient and Remainder, respectively. Divisor is chosen in a fashion that the sum of the quotient depth and the remainder depth is equal to the input image depth. Hence in a k -bit depth image, the divisor is a power of two and smaller than the largest possible intensity value. For a specific n of $n = 2^m$, $m = 1, 2, 3, \dots, k-1$, the quotient and remainder are

$$q(i, j) = I(i, j) \operatorname{div} n, \quad r(i, j) = I(i, j) \bmod n. \quad (2)$$

For the input image $I(i, j)$, intensities are in the interval $0 \leq I(i, j) \leq 2^k - 1$. Hence from (2)

$$\begin{aligned} 0 \leq I(i, j)/n &\leq \frac{2^k - 1}{n}, \\ 0 \leq q(i, j) &\leq \frac{2^k - 1}{2^m}, \\ 0 \leq r(i, j) &\leq 2^m, \end{aligned}$$

$$q(i, j) \in [0, 2^{k-m} - 1] \text{ and } r(i, j) \in [0, 2^m]. \quad (3)$$

It follows from (3) that $q(i, j)$ has $(k-m)$ -bit depth and $r(i, j)$ has m -bit depth. If the depths of $q(i, j)$ and $r(i, j)$ are added together, the total depth will be k which is equal to the depth of the input image. Another advantage of n being a power of two is from a hardware implementation point of view since it changes the division process to m -bit right shifting, which is easily implemented by using a shift register instead of a divisor. The question here is *why should the division of intensity values take place before prediction?* Quotients of the division step are the input of the prediction part. As shown below, if image intensities are divided by n , the probability of the correct prediction increases n times.

Assume that a linear prediction model such as Prediction Number Five ($X = B$) of Fig. 1 is used. Then the probability of the correct prediction is the probability that the intensity value shown in (4). R_r is

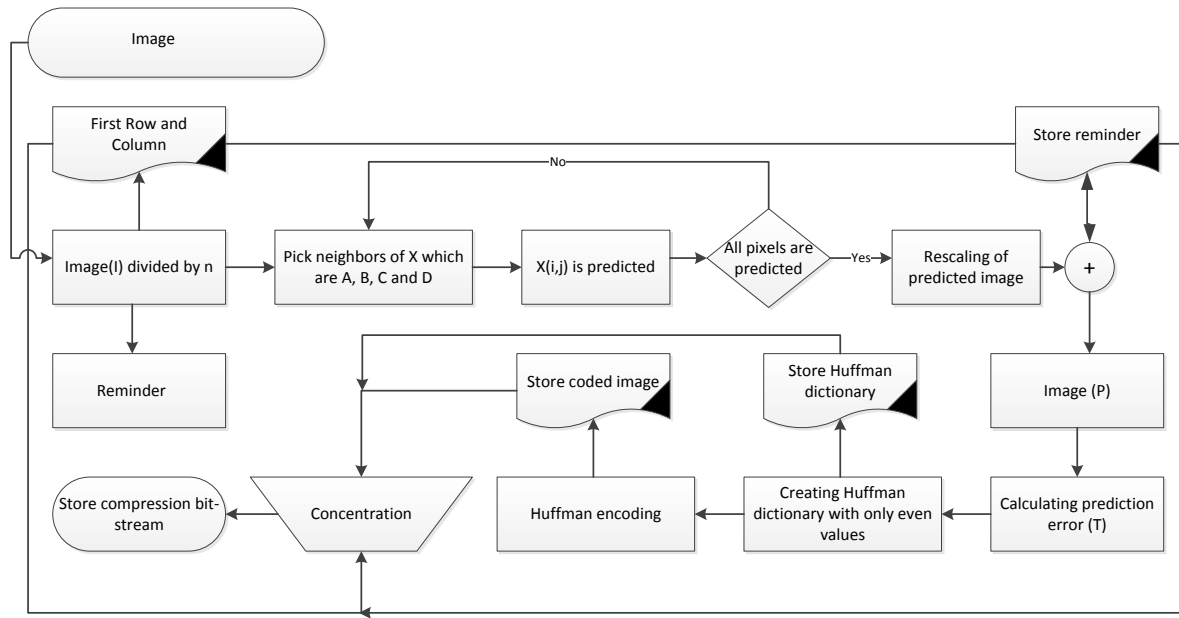


Fig. 2 Flowchart of the proposed compression method.

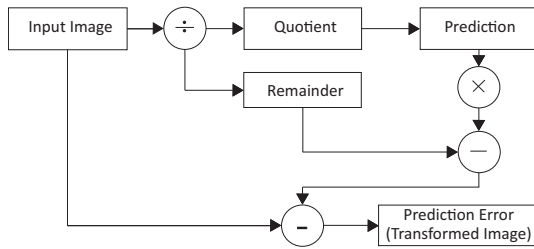


Fig. 3 Block diagram of the new transformation method⁶.

the number of repetitions of intensity r in an $M \times N$ image and P_r is the probability of r . Hence in this case, P_r will be the probability of the correct prediction.

$$P_r = \frac{R_r}{MN}. \quad (4)$$

We have $q(i, j)$ from (2), which is obtained by dividing each pixel of the input image by n . The probability of the correct prediction for $q(i, j)$ is shown in (5). R_q is the number of repetitions of the new intensity value q in the divided image. Hence

$$P_q = \frac{R_q}{MN}, \quad (5)$$

$q = 0, 1, \dots, (2^k - 1)/n$. When one number is considered instead of a series of values in an interval, the total number of repetitions increases. From (5), the new value of repetition R_q and then P_q are

$$R_q = dR_r, \quad P_q = dP_r, \quad (6)$$

$$1 \leq d \leq n.$$

Equation (6) allows one to easily deduce that the probability of the correct prediction increases. In other words, the probability of the correct prediction in a smaller interval is higher than a bigger interval which leads to better predictions. The objective of our method is to estimate the prediction error, so it would be necessary to have a predicted image. Next, the predicted matrix is re-scaled by multiplying the predicted matrix by n and then adding it to the remainder. Hence the predicted image is calculated from

$$p(i, j) = q_p(i, j)n + r(i, j), \quad (7)$$

where $q_p(i, j)$ is the output of the prediction step and can be called the predicted matrix. Then the prediction error, which is the output of the transformation $T(i, j)$, is obtained from

$$T(i, j) = I(i, j) - p(i, j). \quad (8)$$

The transformed image using this method has a lower range of intensities than the traditional DPCM. Hence more energy compaction will take place. This happens because the transformed image values have less variety than the output of the DPCM. Actually, as shown by the mathematical proof below, variations of the transformed image values by this method are n times smaller. As with (7),

$$I(i, j) = q(i, j)n + r(i, j). \quad (9)$$

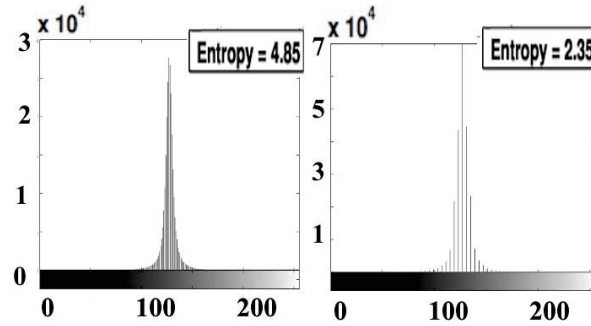


Fig. 4 Histogram of (a) lossless JPEG transformation, (b) transformed image by new method with $n = 32$ of Barbara image⁶.

r is equal in both (7) and (9). Hence

$$I(i, j) - p(i, j) = (q(i, j) - q_p(i, j))n. \quad (10)$$

From (8) and (10), we have the following equation for the transformed image:

$$T(i, j) = (q(i, j) - q_p(i, j))n.$$

As q_p and q are in the range of $[0, 2^{k-m} - 1]$,

$$0 \leq \|q(i, j) - q_p(i, j)\| \leq 2^{k-m} - 1.$$

So all possible prediction error values $T_v(i, j)$ for our transformation is calculated from

$$T_v(i, j) = nV, \quad V = 0, 1, \dots, 2^{k-m} - 1. \quad (11)$$

According to (11), the variety of outputs for the new method is n times smaller than the DPCM. So energy compaction is n times better, which is a good attribute. All outputs of the method are coefficients of n . The new method is applied to a 512×512 Barbara image by selecting $n = 32$. Further, the histogram of lossless JPEG transformation output for the Barbara image is compared with the histogram of the new method output in Fig. 4. From (11), the variety of prediction error values will be equal to $(T_v(i, j) = 32V)$, where V is in the range of $[0, 7]$. Hence there will be only 8 different possible values for the prediction error, which are $[0, 32, 64, \dots, 224]$. On the other hand, the DPCM outputs have a distribution in the range of $0-255$. As can be seen from Fig. 4, the new method causes less variety in the output and, as a result, more energy compaction and more redundancy reduction would be achieved. Hence the intensity distribution becomes n times lower. Thus, in general,

$$V = \frac{2^k}{n} = 2^{k-m} \quad (12)$$

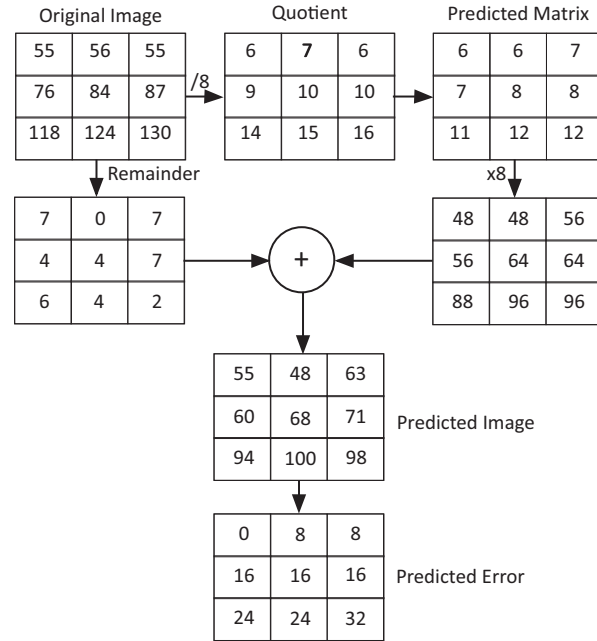


Fig. 5 Numerical example of the new transformation method.

The entropy value decreases by enhancing the energy compaction and causes more compression. Assuming an image with equal probability distribution for all intensities, the entropy can be calculated¹²:

$$E = - \sum_{k=0}^{L-1} p_r(r) \log_2 p_r(r),$$

where r is the intensity value, L is the number of intensity values used to present the image, and $p(r)$ is the probability of intensity r . For example, for an image with a depth of 8 bits, L would be equal to 256 and r would be in the range of $0-255$. The histogram of the given image can be used to calculate the intensities' probabilities¹². An image with lower entropy indicates that more compression is possible. Transformation increases energy compaction and, as a result, the probability of each intensity or source symbols increase leads to less redundancy and less entropy. The compression ratio can be obtained by dividing the size of the original image by the size of the compressed bit-stream. Computational complexity is another factor that determines the efficiency of a method and can be obtained by counting the number of CPU cycles or hardware gates⁹. Then, for the entropy of such image, we will have

$$E = -MN \frac{\log_2 \left(\frac{1}{L} \right)}{MN} = \log_2 L, \quad (13)$$

where $L = 2^k$ is the number of intensities. For an 8-bit image, $L = 256$. Hence the entropy is equal to k or 8 for the described image. For a specific n , the intensity values decrease n times. Since $n = 2^m$,

$$L_n = 2^{k-m} = \frac{L}{n}, \quad (14)$$

where L_n is the number of intensities for the divided image. From (13) and (14), for an image transformed by the new method, we have

$$E = -\log_2 \left(\frac{n}{L} \right) = k - m.$$

It means that the required bits per pixel or entropy value of the transformed image decreases and, as a result, a higher compression would be achieved. As the lossless compression of the RS data is our main purpose, $n = 2$ must be used to prevent any loss of information. Hence the only possible values for the remainder would be 0 or 1 and there would be no loss of the critical information in the reconstruction phase. In the image reconstruction phase, one failure in intensity value estimation (e.g., reconstructing 201 instead of 200) causes only about 0.39% of intensity change. Another way to ensure that the method is completely lossless is to save or send remainders with the transformed image that is 1/8 of the main image. In this paper, for achieving the completely lossless compression, the remainders are saved and used in the reconstruction phase. Lossy compression is achieved by applying a larger n within the division step. If we choose a very large n , like 64, we may lose a lot of information and both the predicted and the reconstructed images will not have an acceptable quality.

As an example, the whole numerical process is shown in Fig. 5. A 3×3 portion of the Barbara image is selected and all the steps of the new method are shown. For this example $n = 8$ and prediction number one ($X = (B + D)/2$) is used. As can be seen from Fig. 5, all the values of prediction error are factors of 8. So all the energy will be compacted into 32 different values:

$$T_v(i, j) = 8V,$$

$$V = 0, 1, \dots, 31.$$

It is clear from the numerical example that keeping the remainder helps a lot with the energy compaction because it ensures that the answers are a factor of n . Also, it should be noted that a good predictor gives better results.

Entropy encoding

Huffman encoding is selected as the entropy encoder for the proposed lossless compression method. Huffman codes the image intensities based on their probability and forms them as a bit-stream. Hence the Huffman dictionary, which includes intensity codes, needs to be saved as an overhead too. After calculating the prediction error, all the values are in the range $[-2^k + 1, 2^k - 1]$. For example, for a greyscale image with an 8-bit depth, the interval is between -255 and 255 . Hence Huffman needs a dictionary which covers this interval. To have an efficient encoder, it was decided to change the negative values of the transformed image to positive values. This occurs by multiplying the input image and the predicted image by n , as stated in (11). Absolute values are saved in the subtraction process. As a result, the dictionary becomes smaller and needs to code only the values between 0 and 256. However, the signs need to be known for the reconstruction phase. To solve this problem, an extra overhead is added to the compressed image which includes the sign of each code in one bit. This overhead is 1/8 of the input image.

Reconstruction

The reconstruction phase consists of Huffman decoding and the inverse of our transformation method. The inverse of the transformation is obtained by calculating the main values from the inverse of the prediction equation. To achieve this goal and to have a complete reconstruction, the first row and the first column of the main image needs to be saved and then each pixel is reconstructed from its neighbours. The inverse transformation for pixel X of Fig. 1 is calculated as follows:

$$I(i, j) - I(i, j) \bmod n = T(i, j) + \lfloor (\text{prediction equation } \#k) / n \rfloor n,$$

where k is the equation number from Fig. 1, $I(i, j) \bmod n$ is the remainder which was saved as the overhead, and $T(i, j)$ is the predicted image. With the first row and column and using the same predictor which was used in the transformation step, the predicted image is reconstructed. Moreover, calculating the floor of division of the prediction equation by n and then multiplying it by n again, help us to obtain the predicted image. The main image is retrieved by adding the predicted image and the predicted error together. The only problem would be the remainders, which were passed over during the quantization step of transformation. $I(i, j)$ is used to solve this problem and to avoid the loss of information. $I(i, j) \bmod n$

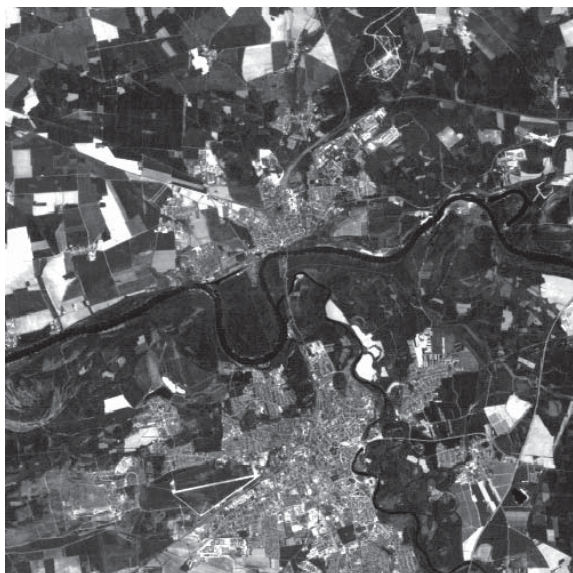


Fig. 6 Landsat TM image of the town of Dessau on the Elbe River in Saxony-Anhalt, Germany.

is the remainder which was saved as the overhead in the transformation step. It should also be noted that, as we mentioned before, for lossless compression, $n = 2$ and the remainders are saved. On the other hand, for visually lossless compression, the remainder overhead can be ignored and greater compression ratio would be achieved. The compression ratio (CR) can be calculated from

$$CR = \frac{\text{sum of all the bands sizes}}{\text{sum of compressed bit-streams sizes}}.$$

Image processing data set

First, we applied our method on fifty non-medical test cases. These test cases are standard test images for greyscale images and have different specifications. The Barbara, Lena, and baboon images are some of them. They are 512×512 greyscale images with 8-bit depth. Hence 262 kB of memory space is needed for the uncompressed version of the images (for example, TIFF version). All 50 test cases can be found in Ref. 3.

RS images

Ten RS greyscale images were selected and all 3 compression methods were applied. RS test images include information about the Earth in digital form, and were captured by satellites in different spatial resolutions. Our test cases range in size from 300×450 to 1893×1825 . Fig. 6 illustrates an example of one of our RS test images which was captured by a

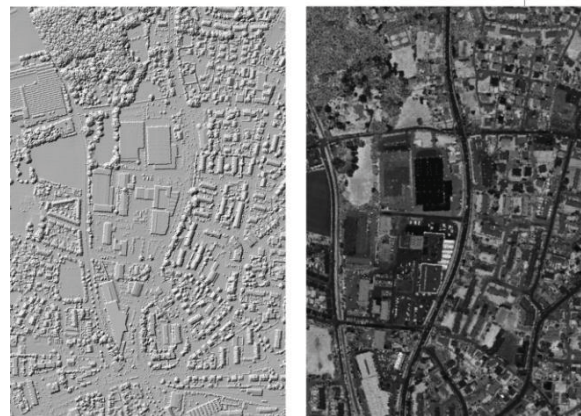


Fig. 7 Example of rasterized LiDAR data: (a) first return, (b) first intensity.

Landsat Thematic Mapper (TM) sensor with a 30 m spatial resolution.

LiDAR

LiDAR technology is used to obtain elevation and tree height data using eye-safe near-infrared laser light in the region of 1040–1060 nm. A laser pulse moves with the speed of light and so the travel time of the laser pulse from the transmitter to the target and back to the receiver can be measured accurately¹³. For converting these data to images, Inverse Distance Weighting or Triangular Irregular Networks was applied to return data and portray them in shaded-relief forms for data validation and looking for anomalies and errors in the data. Fig. 7a shows an example of a LiDAR image. Most LiDAR systems provide an intensity file in addition to the multiple return data¹³. Fig. 7b shows an example of an intensity image that was used in this study.

As can be seen from Fig. 7, LiDAR images have high neighbouring correlation. Hence we envisage that the prediction model will be efficient for LiDAR-rasterized data transformation since the possibility of correct prediction in the highly correlated images will be more than for other images and, as a result, the transformed image energy will be more compact. Hence the transformed image will have lower entropy.

Hyperspectral images

Two different hyperspectral test cases are used. The first test case was a hyperspectral image that consists of 242 bands. Each band consists of 256×3128 pixels and was captured by Hyperion. Hyperion is a hyperspectral sensor that is carried on the Earth Observer (EO-1) satellite in a 705-km sun-synchronous



Fig. 8 A band of the HYDICE image, our second test case.

Table 1 Compression ratio of different non-medical test images.

Image	JPEG	JPEG2000	EDT
Baboon	1.23	1.30	1.35
Barbara	1.29	1.66	1.65
Lena	1.54	1.84	1.97
Average of other 47 images	1.54	1.57	1.71
Overall Average	1.53	1.58	1.71

orbit at a 98.7-degree inclination. Hyperion is a push broom spectrometer and provides resolution of the surface properties in hundreds of spectral bands in the range of 0.4–2.35 μm at 30 m spatial resolution. The second test case was as image captured as part of the Hyperspectral Digital Imagery Collection Experiment (HYDICE) project. HYDICE is a push broom imaging spectrometer, which collected data in 210 bands in the range of 0.4–2.5 μm at an IFOV of 1–4 m, depending on the altitude of the aircraft and ground speed. In this case, the sensor collected data over the Washington DC Mall in a 191 channel set. This dataset contained 1280 scan lines with 307 pixels in each scan line; thus each band is a 1280×307 image. Fig. 8 illustrates one of the bands of this test case.

EXPERIMENTAL RESULTS

Image processing test cases

We compressed all images with lossless JPEG, JPEG2000 and the new EDT compression methods. Lossless JPEG compression was achieved using MATLAB and JPEG2000 compression using ENVI software version 4.4. Table 1 shows the compression ratios for the image processing test cases (50 images, including the Barbara, baboon, and Lena images). Rather than providing all 50 compression ratios, we have included the Barbara, baboon, and Lena image compression ratios separately and then have an average value for the other 47 images as well as an overall average for all images and all compression methods.

We can see from Table 1 that our new method has a better compression ratio than the previous methods. Lossless JPEG has the lowest complexity among

Table 2 Average compression ratio of the 10 different remote sensing greyscale images.

Image	JPEG	JPEG2000	EDT
1	1.31	1.41	1.46
2	1.38	1.38	1.73
3	1.31	1.25	1.76
4	1.24	1.29	1.46
5	1.19	1.20	1.34
6	1.48	1.47	1.90
7	1.34	1.42	1.57
8	1.49	1.38	1.95
9	1.23	1.23	1.57
10	1.32	1.40	1.60
Average	1.33	1.34	1.63

these three methods. However, it also leads to the lowest compression ratio. JPEG2000 has a higher compression ratio than Lossless JPEG but lower than the proposed EDT method for most of the test cases. The overall average compression for the EDT method is much higher than the lossless JPEG and JPEG2000 methods. It should be noted that, to have the same experimental situation for each image, the equation used in the predictor block of our transformation is the same.

Remote sensing greyscale images

Our main focus was on lossless compression of RS images and, to provide lossless compression, we picked $n = 2$ as the divisor and save the remainder as the overhead. Table 2 illustrates the compression ratios achieved for the 10 greyscale satellite test images. For the different test images, results indicate that the proposed compression method improved compression by about 23% over lossless JPEG compression method and by around 22% over the JPEG2000 method. According to a detailed analysis of JPEG2000's transformation scheme, JPEG2000 is efficient in high detail images. However, the proposed method generally gives better results than JPEG2000. T-Test results show that the compression ratio for EDT is significantly different from JPEG2000 and lossless JPEG.

LiDAR rasterized data

To see the efficiency of the proposed method, 4 LiDAR rasterized images, such as that shown in Fig. 7, were compressed by the three methods and the compression ratios were compared. To achieve the adaptive compression, adaptive EDT (AEDT) is used for the transformation step. AEDT is obtained by using the adaptive equation of JPEG-LS, which have been shown in (1), for the prediction step of

Table 3 Compression ratio of different standard test images of LiDAR rasterized data.

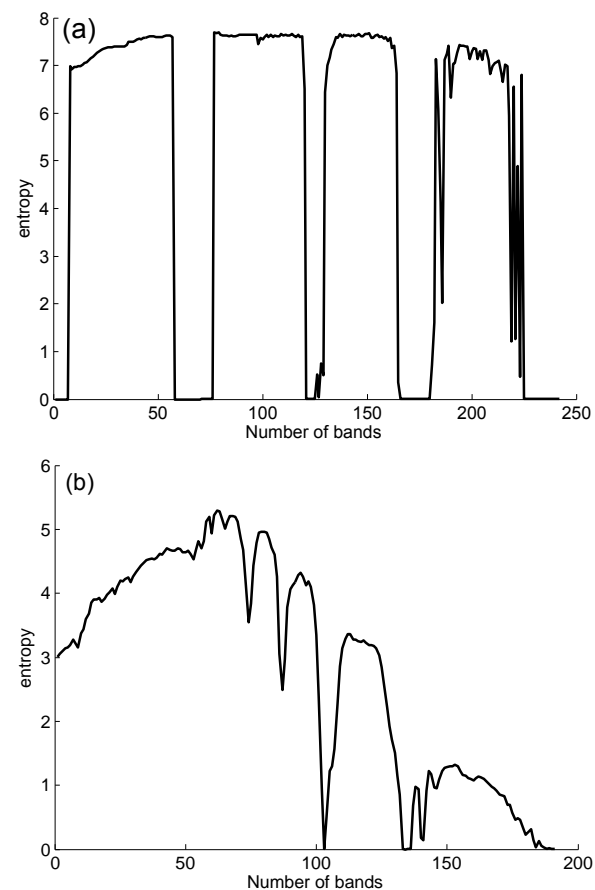
Image	JPEG	JPEG2000	EDT	AEDT
1	1.23	1.28	1.43	1.64
2	1.34	1.34	1.72	1.97
3	2.48	2.56	2.96	3.40
4	1.33	1.33	1.71	1.95

EDT. Hence the transformation uses efficient prediction equations for each specific pixel. The predictor detects if the pixel lies on vertical or horizontal edge. These statistics are kept to improve the value of the entropy encoder parameter. Table 3 illustrates the results. The images number one and two are the two related images of Fig. 7a and b. As can be seen from Table 3, the proposed method is efficient in the compression ratio and improves on the previous methods quite considerably. The average percentage increase in the compression ratio of EDT over JPEG was 22%, while that of EDT over JPEG2000 was 21%. With AEDT, the improvement over JPEG was 41%, over JPEG2000 39% and over EDT 15%. The ability of EDT in energy compaction, suitability of prediction schemes for high correlated images and using the optimized Huffman encoder were the main reasons in achieving such results. In addition, computational complexity of the new lossless compression method is low and causes high speed hardware implementation. Complexity is less than JPEG2000 and is approximately similar to lossless JPEG. Hence the proposed method is efficient for lossless compression of RS images and LiDAR rasterized data and especially for real time applications due to its compression ratio and low computational complexity.

Hyperspectral images

We applied the compression techniques on two different Hyperspectral datasets. Fig. 9 illustrates the entropy value of all the bands of these two Hyperspectral data test cases. In our test cases, it can be seen that there are just a few bands with an entropy equal to zero, as well as some bands with low entropy values. These are highly correlated in the neighbouring pixels and include less information than the other bands. The proposed methods include the predictive scheme, and due to the neighbouring correlation of the large number of bands, it can be efficient for this test case as well.

Table 4 gives the compression ratios of the 3 compression techniques on the two hyperspectral test images. In the first test case, the proposed method with

**Fig. 9** The entropy values of all the bands of (a) first test case (b) second test case.**Table 4** Lossless compression ratio of both test cases obtained by different compression algorithms.

Test Case	JPEG	JPEG2000	EDT
First	2.38	2.78	2.58
Second	2.38	2.04	3.76

EDT works better than the lossless JPEG. However, JPEG2000 has a better compression ratio than the EDT based method. The only advantage of EDT is its simplicity, which is suitable for real-time systems that need simple and fast algorithms. As shown in Fig. 10, in the corrupted bands (bands with lower information or lower entropy), JPEG2000 works better than the EDT. On the other hand, the proposed method with EDT acts more efficiently than JPEG2000 for compression of bands with more information.

EDT demonstrates the best result in the second test case. One of the differences here is that JPEG2000 shows the worst result among the three methods. It

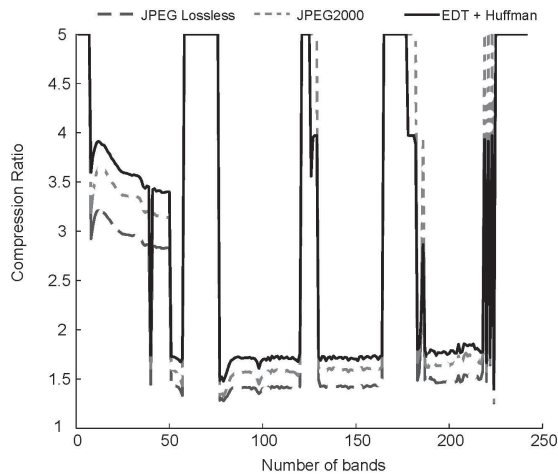


Fig. 10 The compression ratio of different methods for all the bands of the first test case.

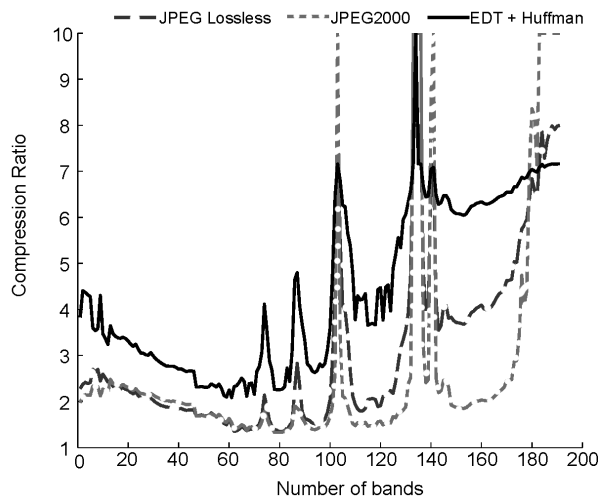


Fig. 11 The compression ratio of different methods for all the bands of the second test case.

shows that a predictive scheme is better for this test case. Also, due to the improvement of the new method from lossless JPEG, the improvement in the compression ratio is predictable. The compression ratio plot is shown in Fig. 11. As expected, the compression ratios are high in the low-detail bands that had lower entropy values. Furthermore, in the second test case, about 50% improvement in the compression ratio could be observed.

It should be noted that, to have a better view of the values in Figs. 10 and 11, the compression ratio with a value greater than 5 is not included in the plot since the values of the compression ratio of the corrupted bands are high. For instance, band number 75, which is a

Table 5 Computation complexity of different transformations.

transformation	order	compression type
DPCM	n	lossless
EDT	n	lossy/lossless
DWT	$n \log_2 n$	lossy/lossless

corrupted band, has a compression ratio of 8.25 and 1190 for lossless JPEG and JPEG2000, respectively.

COMPUTATIONAL COMPLEXITY

Table 5 illustrates the computational complexity of the different transformation methods and the compression type which is used for comparison. The order of algorithm is $\mathcal{O}(n)$ in which $n = MN$ is the number of pixels. The comparison is based on the order of each transformation that may increase if a time-consuming loop occurs. Due to the fact that EDT and DPCM do not have any filter banking procedure or convolution, they are less computationally complex than the transformation of JPEG2000 (DWT). EDT has the same order as DPCM since there is no hierarchical loop added to the old DPCM algorithm. According to the additional division and multiplication process in EDT, it has a slightly higher computational complexity than lossless JPEG. To avoid this problem and to have a less complex lossless scheme, shift with carry is used. So the remainder of the dividing process is saved in carry. Then, in the rescaling process, one left shift with carry would be used to add the remainder to the predicted matrix. Hence, the process of reconsidering the remainder would be omitted and the total computational complexity will decrease. Hence it can be concluded that our proposed lossless transformation is achieved by adding only two shift registers to transformation of lossless JPEG and as a result, complexity stays low. It should be noted that selecting a less complex predictor such as $X = B$, can decrease the total computational complexity and accelerate the transformation process. For example, predictor $X = (B + D)/2$ needs a one bit shift register and an adder block, but $X = B$ can be implemented by only one selector. Because of the high neighbouring correlation of RS images, predictor $X = B$ is an efficient choice for the purpose of redundancy reduction too.

CONCLUSIONS

In this paper, an efficient method for lossless compression of a wide variety of RS images has been introduced. Our proposed EDT-based compression

method is more efficient for all types of images tested. Large improvements in the compression ratio over the Lossless JPEG and JPEG2000 methods were achieved. In other words, the new method has a great potential to compress different types of images in less CPU computational time. The proposed method is based on an enhanced DPCM transformation and an optimized Huffman entropy encoder. Future studies would look at hardware implementation, testing and evaluation.

Acknowledgements: This work was supported by a PhD scholarship (SFRH/BD/73382/2010) by the Portuguese Foundation for Science and Technology, the Institute of Systems and Robotics and the Instituto de Telecomunicações.

REFERENCES

- Pickering M, Ryan M (2006) An Architecture for the Compression of hyperspectral imagery. In: Motta G, Rizzo F, Storer JA (eds) *Hyperspectral Data Compression*, Springer pp 1–34.
- Wang H, Sayood K (2006) Lossless predictive compression of hyperspectral images. In: Motta G, Rizzo F, Storer JA (eds) *Hyperspectral Data Compression*, Springer pp 35–55.
- Mielikainen J, Toivanen P (2006) Lossless hyperspectral image compression via linear prediction. In: Motta G, Rizzo F, Storer JA (eds) *Hyperspectral Data Compression*, Springer pp 57–74.
- Calderbank R, Daubechies I, Sweldens W, Yeo BL (1997) Lossless image compression using integer to integer wavelet transforms. In: *Proceedings of ICIP*, Santa Barbara, pp 596–9.
- Foos DH, Muka E, Slone RM, Erickson BJ, Flynn MJ, Clunie DA, Hildebrand L, Kohm KS, Young SS (2000) JPEG 2000 compression of medical imagery. In: *Proceedings of SPIE, 3980, Medical Imaging 2000: PACS Design and Evaluation: Engineering and Clinical Issues*, p 85.
- Ghamisi P, Sepehrband F, Choupan J, Mortazavi M (2011) Binary Hybrid GA-PSO Based Algorithm for compression of Hyperspectral Data. In: *ICSPCS' 11*, Honolulu, Hawaii, pp 12–4.
- Skodras A, Christopoulos C, Ebrahimi T (2001) The JPEG2000 still image compression standard. *IEEE Signal Processing Magazine*, 36–58.
- Sepehrband F, Mortazavi M, Ghorshi S, Choupan J (2010) Efficient medical image transformation method for lossless compression by considering real time applications. In: *International Conference on Signal Processing and Communication Systems*, pp 1–8.
- Santa-Cruz D, Ebrahimi T, Askelof J, Larsson M, Christopoulos CA (2000) JPEG 2000 still image coding versus other standards. In: *Proceedings of SPIE 4115, Applications of Digital Image Processing XXIII*, pp 1–10.
- Sepehrband F, Ghamisi P, Mortazavi M, Choupan J (2010) Simple and efficient remote sensing image transformation for lossless compression. In: *ICSIP Proceedings*, Changsha, China, pp 66–70.
- Starosolski R (2007) Simple fast and adaptive lossless image compression algorithm. *Software-Practice and Experience*, Wiley, pp 37–56.
- Gonzales R, Woods R (2008) *Digital Image Processing*, Pearson Prentice Hall, Upper Saddle River, New Jersey.
- Jensen JR (2009) *Remote Sensing of the Environment: An Earth Resource Perspective*, 2nd edn, Pearson Education India.