

Hyperspectral regression lossless compression algorithm of aerospace images

Assiya Sarinova¹, Alexander Zamyatin²

¹Toraighyrov named after Pavlodar state university, Electrical Engineering and Automation Department, 140000 Lomov street 64, Kazakhstan

²National research Tomsk state university, Institute of Applied Mathematics and Computer Science, 634050 Lenina avenu 36, Russia

Abstract. In this work, we propose an algorithm for compressing lossless hyperspectral aerospace images, which is characterized by the use of a channel-difference linear regression transformation, which significantly reduces the range of data changes and increases the degree of compression. The main idea of the proposed conversion is to form a set of pairs of correlated channels with the subsequent creation of the transformed blocks without losses using regression analysis. This analysis allows you to reduce the size of the channels of the aerospace image and convert them before compression. The transformation of the regressed channel is performed on the values of the constructed regression equation model. An important step is coding with the adapted Huffman algorithm. The obtained comparison results of the converted hyperspectral AI suggest the effectiveness of the stages of regression conversion and multi-threaded processing, showing good results in the calculation of compression algorithms.

Introduction

Hyperspectral aerospace images (AI) are images obtained from Earth remote sensing spacecraft (ERS), designed to solve problems in the field of applied research. Studies to compress hyperspectral AI are the most interesting; this is subject to a large number of publications. In hyperspectral AI for each pixel, the hyperspectral camera accepts light intensities for a large number of adjacent spectral ranges reaching several hundred. Hyperspectral AI due to the rich content of information are effectively used in the tasks of automated processing of remote sensing images. The stages of hyperspectral (AI) processing can include: internal representation of images, image conversion, geometric correction of scenes, and image preprocessing. Compression efficiency can consist of the following steps - a change in the internal representation of images, converting it to a compact form, and using pre-processing. Compression is the preprocessing and removal of redundancy in images. There are two types of redundancy in hyperspectral AI, spatial and spectral. These redundancies will allow the development of efficient compression algorithms.

Based on researches of hyperspectral AI in the field of compression [1–20] presented in the works of scientists from Russia, China, the USA, India, etc., it can be assumed that the developed methods and lossless compression algorithms for hyperspectral AI can be improved by reducing their computational efficiency and increasing the compression ratio by modifying the preprocessing steps using mathematical methods. In addition, new compression preprocessing steps can be proposed that effectively increase the compression ratio and reduce the compression process time. The latest best results in solving the compression problem, as in many other remote sensing problems, were obtained using well-known algorithms in their various combinations and groups, while having low compression ratios, the average compression ratio is 3.85. Based on the analysis of lossless GI compression methods and algorithms, it should be concluded that the most effective ways to solve the compression problem are:

- taking into account spectral correlation, which gives certain advantages on the basis of the calculated correlation matrix;
- application of a new method of ordering GI channels;
- use of interpolation based on mathematical methods;
- arithmetic coding and the Huffman algorithm, which are the best among statistical methods;
- the use and organization of parallel compression processing to reduce the cost of computing resources.

In a previous study of the authors of a multi-stage compression algorithm [10], the main drawback remains - the computational efficiency of compression is low, large computational costs are required, which makes it difficult to use them to solve GI compression problems.

Therefore, in this work, a modification of the algorithm is proposed taking into account the correlation [10], in which an attempt is made to eliminate this drawback.

Modification of the algorithm with regression analysis. The essence of the modification is to calculate the linear regression coefficients between the values of the generating channel (master) and the regressed channels (slave, compressible) of the GI by forming arrays of differences between master and slave. When using a PC with different computing capabilities, the modification of the algorithm uses a different number of threads, corresponding to the number of available processor cores. The proposed modification of the lossless compression algorithm taking into account inter-band correlation and regression analysis will increase the compression ratio by more than two times in comparison with the use of universal archivers. The proposed algorithm for finding the best channel groups for a given correlation value will increase the efficiency of the channel subtraction (difference transformation) application.

1 Description of the approach to the problem of compression of hyperspectral AI

The sequence of processing steps and the compression algorithm:

- 1) calculation of the correlation between all pairs of AI channels and determining the sequence of channel coding and decoding;
- 2) regression conversion algorithm;
- 3) obtaining channel differences and their block conversion;
- 4) compression by a statistical algorithm.

We describe each step of the algorithm.

1 step. We calculate the values of the correlation matrix between all pairs of AI channels, revealing the most correlated groups of channel pairs. Based on the matrix, we

will form and determine the sequence of transformation (coding) and inverse transformation by the method of constructing a strongly branching tree.

2 step. Regression analysis based on step 1. Let us calculate the linear regression coefficients between the values of the generating and regressing channels of the hyperspectral AI by creating optimal values for generating arrays of differences between master and slave.

3 step. Compression by the Huffman statistical algorithm [2].

2 Description of the compression algorithm

2.1 Compression Sequencing

At the first stage, we transform the data structures based on the initial hyperspectral AI, using mathematical models for finding the correlation between all pairs of channels, and we determine the coding sequence. Knowing that hyperspectral AI are obtained in the spectrum of a single wave, we assume that a certain degree of dependence can be determined between pairs of channels. To determine the magnitude of this dependence, we use the formula for calculating the Pearson correlation coefficient (aka linear correlation coefficient). The formula requires two sequences of data, so we first extract two sequences of samples from a pair of channels. This happens as follows: we represent the two-dimensional data matrix of one channel in the form of a linear array (passing the matrix row by row from left to right and rows from top to bottom), then select a certain number of samples in it (we denote the number by the letter m) dividing the array into approximately equal segments. From the second channel, we extract a sequence of samples that are located in the matrix at the same positions as the samples from the first channel. We denote the obtained sequences by the letters x and y , and the unit values are x_i and y_i (i from 1 to m inclusive). It is also necessary to calculate the arithmetic mean values of both sequences of samples.

Counting the coefficients for all possible pairs of channels of the same image, we build the coding sequence - naturally, starting from larger values to smaller ones.

1 The Construction of the correlation matrix of AI channels.

2 Build a highly branching tree to determine the sequence of channel conversion depending on the correlation value using the Pearson correlation calculation formula.

3 Formation of the hyperspectral AI processing sequence for step 1.2, necessary for AI recovery.

Example. There is a 12-channel GI, respectively, the index tree will contain 11 vertices, we will build them in the correlation table, table. 1. From table 1 it is seen that the interchannel correlation increases, starting from channel 5. This suggests that the channels whose indices lie in the range [4-11] have a high correlation dependence, the maximum correlation value between channel 10 and 11.

Table 1. Correlation matrix of selected channels for compression

	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0,35	0,29	0,3	0,23	0,23	0,22	0,23	0,22	0,23	0,22	0,23
1	0,35	1	0,34	0,34	0,3	0,27	0,29	0,29	0,28	0,29	0,29	0,3
2	0,29	0,34	1	0,6	0,56	0,51	0,53	0,52	0,5	0,5	0,49	0,49
3	0,3	0,34	0,6	1	0,89	0,85	0,85	0,85	0,82	0,81	0,8	0,79
4	0,23	0,3	0,56	0,89	1	0,95	0,95	0,95	0,95	0,92	0,92	0,9
5	0,23	0,27	0,51	0,81	0,95	1	0,97	0,97	0,97	0,96	0,95	0,95
6	0,22	0,29	0,53	0,85	0,95	0,97	1	0,985	0,983	0,978	0,972	0,968
7	0,234	0,29	0,52	0,845	0,945	0,975	0,985	1	0,990	0,981	0,984	0,982
8	0,23	0,29	0,50	0,82	0,935	0,970	0,983	0,990	1	0,992	0,991	0,988
9	0,23	0,29	0,50	0,81	0,924	0,963	0,978	0,989	0,992	1	0,995	0,994
10	0,224	0,295	0,491	0,803	0,912	0,955	0,972	0,984	0,991	0,995	1	0,996
11	0,23	0,30	0,499	0,798	0,907	0,950	0,968	0,982	0,988	0,994	0,996	1

3.1 Based on the constructed correlation matrix, we construct the L tree of indices - these are ordered pairs of channels (for example, channels numbered 10 and 11). Let there be 11 pairs and 11 vertices of the tree. Channel 12 under index [11] is converted through the first vertex of the tree [# 0] (the numbering of vertices and indices starts from zero), Channel 10 under index [9] is converted through the second vertex [# 1], Channel 9 under index [8] is converted through the third peak [# 2], etc., fig. 1.

3.2 Formation of a sequence of pairs:

Step 1. Find the maximum element among the available pairs in the correlation matrix, form the first pair of channels.

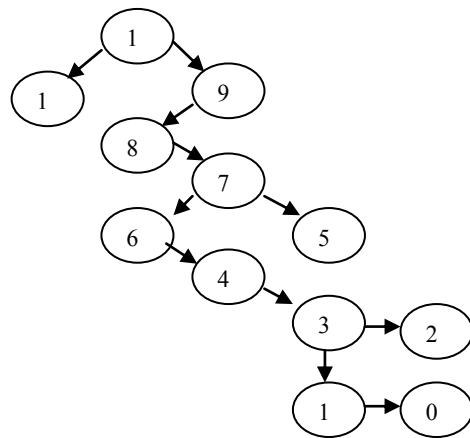
Step 2. We put in a set of ordered pairs (SOP) a new pair of channels.

Step 3. We remove from the table of the correlation matrix all pairs whose elements are already defined in the SOP.

Step 4. While all the elements are not in the SOP, we return to step 1.

#0	10 ->	11
#1	10 ->	9
#2	9 ->	8
#3	8 ->	7
#4	7 ->	6
#5	7 ->	5
#6	6 ->	4
#7	4 ->	3
#8	3 ->	2
#9	3 ->	1
#10	1 ->	0

a)



b)

Fig. 1. The tree of correlated channels:

a) the set of ordered pairs, b) the sequence of ordered pairs

2.2. Regression (regression) transformation

The essence of the transformation is to bring into conformity to the encoded pair of channels some structure that would:

1) allowed to unambiguously restore one of the source channels according to the data of another channel,

2) occupied as little disk space as possible.

The subtracted channel is converted using the linear regression equation (LRE) as an example. The generalized form of the transformation will be like this.

LRE - “ $Y = a \times x + b$ ”. Y are unknown values, x are the values of the main channel, y are the values of the second channel, we find a and b :

1) we find a - coefficient by intermediate calculations;

2) we find b - coefficient by subtracting the average of y and the available coefficient a ;

3) calculate $Y = a \times x + b$. where x - values of the main channel, found LRE.

For example, the value of the generating channel x is equal to 918, y - the value of the second channel is 1312. Between the channels there is a weak correlation of 0.3636 (the difference between the channels x and $y = 394$), the coefficients $a = 0.49$, $b = 784.01$ are

found. Then $Y = 1233$. The difference is $d = y - Y = 79$. The value is $79 < 394$ and requires fewer bits to store them.

We apply this type of generalization of LRE to hyperspectral AI. The coefficients of the constructed regression model $Y_i = a \cdot x + b$ for the generating channel were determined.

The idea of linear regression in our case is to find such real values a and b such that the matrix formed from the data of the encoded pair by the following formula: $d_{ij} = (x_{ij} \cdot a + b) - y_{ij}$, (hereinafter, x_{ij} and y_{ij} are the values in the matrices master and slave, respectively) would have in itself the smallest possible values. Below we will call the matrix d with the values d_{ij} the “difference matrix”. A sufficiently good linear correlation index (close to unity), calculated previously, leads to rather low values of d_{ij} . Knowing the values of a , b and the matrix d , we can restore the matrix y from the values of the matrix x :

$$y_{ij} = (x_{ij} \cdot a + b) - d_{ij}$$

The standard formulas used to calculate linear regression parameters are:

$$a = \frac{\overline{xy} - \bar{x} \cdot \bar{y}}{\overline{x^2} - (\bar{x})^2}, \quad b = \frac{\overline{x^2} \cdot \bar{y} - \bar{x} \cdot \overline{xy}}{\overline{x^2} - (\bar{x})^2},$$

where:

$$\bar{x} = \frac{1}{i \cdot j} \sum_{i=1}^m \sum_{j=1}^n x_{ij}; \quad \bar{y} = \frac{1}{i \cdot j} \sum_{i=1}^m \sum_{j=1}^n y_{ij}; \quad \overline{xy} = \frac{1}{i \cdot j} \sum_{i=1}^m \sum_{j=1}^n (x_{ij} \cdot y_{ij}); \quad \overline{x^2} = \frac{1}{i \cdot j} \sum_{i=1}^m \sum_{j=1}^n x_{ij}^2;$$

(m - image height, n - width).

In each slave channel we put the values a and b in the Double format (8 bytes, 15 decimal places in decimal format). Encoding the only main generating channel (generating the channel of the first pair) for the image, independently of the others, we can subsequently restore it first, then gradually restore all other compressed channels (as mentioned earlier, through the coefficients a and b).

The absence of losses during the conversion is ensured as follows. After receiving the matrix d and before writing it to the file, we round off the values to the nearest integer (in the case of a fractional part equal to 0.5, to a smaller integer). This does not prevent lossless recovery. Let's explain why. Let's pay attention to the formula: $d_{ij} = (x_{ij} \cdot a + b) - y_{ij}$.

The matrices x and y are integer; therefore, the quantity has the same fractional part (we denote it by q) as the number d_{ij} before rounding. We will denote the integer part of numbers by square brackets, the fractional by curly brackets. So $q = \{d_{ij}\} = \{w_{ij}\}$. After rounding, rounded values are written to the difference file. We proceed to recovery:

$$\begin{aligned} y'_{ij} &= (x_{ij} \cdot a + b) - d'_{ij} = (d_{ij} + y_{ij}) - d'_{ij} \\ y'_{ij} &= (d_{ij} + y_{ij}) - d'_{ij} = q + y_{ij} - 1 = y_{ij} - (1 - q). \end{aligned}$$

The value of w_{ij} remains unchanged, since a and b were previously calculated and recorded in the Double format (that is, without loss), the values of x are integer and do not undergo losses.

$$\text{If } q \leq 0.5, \text{ to } d'_{ij} = [d_{ij}] \text{ and: } y'_{ij} = (d_{ij} + y_{ij}) - d'_{ij} = q + y_{ij}.$$

So, rounding, we get the initial value y_{ij} .

$$\text{If } q > 0.5, \text{ to } d'_{ij} = [d_{ij}] + 1, \text{ and: } y'_{ij} = (d_{ij} + y_{ij}) - d'_{ij} = q + y_{ij} - 1 = y_{ij} - (1 - q)$$

The value $(1 - q) \leq 0.5$, a y_{ij} is an integer. It also means: rounding, we get the initial value y_{ij} .

Thus, regression conversion does not incur losses. To reverse decode hyperspectral AI, we perform the following steps:

Step 1. Decoding arrays of differences by the Huffman algorithm.

Step 2. Formation of regression transformation arrays by finding the sums between the generating channel and its average value.

Step 3. Formation of the initial arrays based on the available coefficients of HRM and obtaining the initial data of hyperspectral AI.

3 Experimental researches

To determine the effectiveness of the proposed algorithm from the point of view of the degree of compression, as well as the limits of its applicability, a number of experiments were carried out on hyperspectral AI (*Aviris* remote sensing system), in table. 2.

Table 2. Hyperspectral AI Test Data

Count bands	Size	Size (byte)	Count bands	Size	Size (byte)
100	100*100	4080400	200	100*100	8160800
100	200*200	16160400	200	200*200	32320800
100	300*300	36240400	200	300*300	72480800
100	400*400	64320400	200	400*400	128640800
100	614*512	125747200	200	614*512	251494400
150	100*100	6120600	224	100*100	9140096
150	200*200	24240600	224	200*200	36199296
150	300*300	54360600	224	300*300	81178496
150	400*400	96480600	224	400*400	144077696
150	614*512	188620800	224	614*512	281673728

Conclusion

1. The lossless compression algorithm taking into account inter-band correlation and regression analysis allows to increase the compression ratio to ($D > 8$) than in the use of universal archivers.

2. The proposed approach for finding the best channel groups for a given correlation value increases the efficiency of the application of the difference transformation stage.

3. The obtained experimental results show the effectiveness of the application of the stages of regression conversion and parallel processing, which allow to obtain advantages compared to analogues

References:

1. Kashkin VB, Sukhinin AI (2008) Digital processing of aerospace images. Krasnoyarsk: SFU. 278 p. (In Russian)
2. Salomon D. (2007) Data compression. The complete reference. Springer-Verlag. 1118 p.
3. Sayood K. (2006) Introduction to Data Compression. USA. Universitet Nebraska. Morgan Kaufmann Publishers is an imprint of Elsevier. San Francisco, CA 94111. 703 p.
4. William K. Pratt. (2001) Digital Image Processing: PIKS Inside, Third Edition. John Wiley & Sons, Inc, 738 p.

5. Gashnikov MV, Glumov N.I. (2016) On-board processing of hyperspectral data in Earth remote sensing systems on the basis of hierarchical compression. *Computer Optics*. vol. 40. no 4. pp. 543–551. (In Russian).
6. Gashnikov M.V. (2017) Minimization of the entropy of post-interpolation residues during image compression based on hierarchical grid interpolation), *Computer Optics*. vol. 41. no. 2. pp. 266–275. (In Russian).
7. Petrov E.P., Harina N.L., Suhih P.N. (2015) Method of fast compression of images without losses. *Machine learning and data analysis*. vol. 1. no.12. pp.1762–1770. (In Russian).
8. Petrov E.P., Harina N.L., Suhih P.N. (2016) Method of compression of multi-bit satellite images without losses. *Modern problems of remote sensing of the earth*. vol. 13. no. 2. pp. 203–210. (In Russian).
9. Petrov E.P., Harina N.L., Suhih P.N. (2016) The method of image compression in ERS systems without losses. *Bulletin of the Samara State Aerospace University named after Academician S.P. Queen*. vol. 15. no.2. pp. 183–189. (In Russian).
10. Sarinova A., Zamjatin A. and Cabral P. Lossless compression of hyperspectral images with pre-byte processing and intra-bands. *Dyna (Colombia)*. – 2015.–Vol. 82 (190).– pp. 166-172.
11. Zamjatin A.V., Sarinova A.Zh. (2017) Algorithm for compressing hyperspectral aerospace images using mathematical processing and taking into account interband correlation. *Materials of the IV International Scientific Conference "Regional Problems of Earth Remote Sensing"*. pp.157–160. (In Russian).
12. Sergeev V.V. (2006) Analysis and Processing of Images Obtained from Earth Observations from Space. *Computer Optics*. no.29. pp.41–44. (In Russian).
13. Gashnikov M.V., Glumov N.I. (2014) Hierarchical GRID interpolation under hyperspectral images compression *Optical Memory and Neural Networks*. vol. 23(4). pp.246–253.
14. Gashnikov MV, Glumov NI, Kuznetsov AV, Mitekin VA, Myasnikov VV, Sergeev VV. (2016) Hyperspectral remote sensing data compression and protection. *Computer Optics*. vol. 41. no. 5. pp.689–712.
15. Gashnikov M.V., Glumov N.I. (2016) Hyperspectral image compression for transmission over communication channel. *Information Technology and Nanotechnology*. pp.334–339.
16. Hongda Shen, W. David Pan. (2017) Predictive Lossless Compression of Regions of Interest in Hyperspectral Images With No-Data Regions. *IEEE. Transactions on geoscience and remote sensing*, vol. 55. no. 1.pp.173–182.
17. Yuan L., Jianping L., Ke G. (2012) Lossless compression of hyperspectral images using hybrid context prediction. *Optics Express*. vol. 20. no.7. pp. 8199–8206
18. Changguo, Li., Ke G. (2014) Lossless Compression of Hyperspectral Images Using Three-Stage Prediction with Adaptive Search Threshold. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. vol.7. no.3. pp. 305–316.
19. Mahendran M., Jayavathi S.D. (2016) Compression of Hyperspectral Images Using PCA with Lifting Transform // *International Conference on Emerging Engineering Trends and Science*. P.68–73.
20. Sepehrband F., Ghamisi P., Mohammadzadeh A., Sahebi M.P., Choupan J. (2011) Efficient Adaptive Lossless Compression of Hyperspectral Data using Enhanced DPCM. *International Journal of Computer Applications*. vol. 35. no.4. pp. 6–11.