Methodology for Developing Algorithms for Compressing Hyperspectral Aerospace Images used on Board Spacecraft

Assiya Sarinova[1], *Alexander* Zamyatin[2]

[1]Toraighyrov university, Electrical Engineering and Automation Department, 140000 Lomov street 64, Kazakhstan
[2]National research Tomsk state university, Institute of Applied Mathematics and Computer Science, 634050 Lenina avenu 36, Russia

Abstract. The paper describes a method for constructing and developing algorithms for compressing hyperspectral aerospace images (AI) of hardware implementation for subsequent use in remote sensing Systems (RSS). The developed compression methods based on differential and discrete transformations are proposed as compression algorithms necessary for reducing the amount of transmitted information. The paper considers a method for developing compression algorithms, which is used to develop an adaptive algorithm for compressing hyperspectral AI using programmable devices. Studies have shown that the proposed algorithms have sufficient efficiency for use and can be applied on Board spacecraft when transmitting hyperspectral remote sensing data in conditions of limited buffer memory capacity and communication channel bandwidth.

Introduction

Hyperspectral remote sensing AI is important for observing and studying changes in the Earth's surface, monitoring natural resources and the consequences of emergencies, etc. Currently, the development of software systems for transmitting such data is an urgent task. In solving this problem, there are two areas of research: the development of compression algorithms used in ground-based remote sensing data reception and processing centers; and those used on Board SPACECRAFT. Research is actively conducted in the development of compression algorithms of the first direction, in which there are many publications [1-10, 12]. In the second direction of research, there is a potential for developing algorithms, dictated by the necessary list of problems for solving the compression problem.

It is proposed to formulate some problems of developing algorithms for compressing hyperspectral AI that are applicable on Board.

1. The amount of incoming data. Hyperspectral AI has a range of several hundred and thousands of spectral channels. At the same time, the size of channels reaches hundreds of thousands of pixels.

2. Memory resources. The technical characteristics of the hardware that receives, stores, and transmits such data are limited in memory and processing power.
3. Data transmission. Huge volumes are formed on Board the SPACECRAFT at a certain speed exceeding the capabilities of the transmitted communication channel, which has a fixed bandwidth that is insufficient for transmitting uncompressed hyperspectral AI.
4. Data quality. Data quality requirements are very high, as the incoming information is unique.

Compression cannot be applied under these conditions. The problems listed above that arise when developing software systems that are applicable on Board the SPACECRAFT give rise to the following range of requirements for the compression algorithm:

- the speed of a stream of compressed data;
- low computational complexity;
- high compression ratio;
- error control for lossy compression.

Currently, well-known algorithms based on truncated block encoding [1], differential pulse modulation [1], discrete cosine transform [2], and discrete wavelet transform [2] exist and are used for compressing hyperspectral AI. These algorithms on the SC bot, which require large computing resources, do not always meet the above compression requirements. Therefore, it is necessary to develop new methods for compressing hyperspectral AI, which are not inferior in their efficiency to the known methods, but require less computing resources.

Thus, in this paper, we propose to consider a method for developing algorithms for compressing hyperspectral AI that meet these requirements.

1 Description of the methodology for developing compression algorithms

The methodology for constructing the development of an algorithm for the hardware implementation of compression should be divided into three stages.

The first stage is the selection and justification of hardware and software for the compression of hyperspectral AI.

The second stage is the development of lossless and lossless compression algorithms on programmable integrated circuits.

The third stage is testing and conducting experiments of the developed algorithms [11,13] in comparison with the known algorithms in terms of performance, speed and efficiency in compression ratio.

Stage 1. To test the developed algorithms at the first stage, a 32-bit STM32 microcircuit from Electronics and CocoxIDE software with embedded C ++ language was chosen as a tested programmable integrated circuit.

Some of the latest processors for embedded systems are those based on the ARM Cortex-M3 architecture. These processors are designed for use in Digital Signal Processing (DSP).

General description of ARM architecture and 32-bit STM microcontrollers. ARM processors are a key component in many successful 32-bit embedded systems. ARM processors are widely used in mobile phones, tablets and other portable devices. ARMs are based on RISC architecture, which reduces processor power consumption and thus makes them ideal for embedded systems.

Benefits of using:

1. Variable number of execution cycles for simple instructions. Simple ARM instructions require more than one cycle to execute. For example, the execution of the Load and Save instructions depends on the number of registers that are passed to them.

2. Ability to combine commands of shift and rotation with commands for information processing.

3. Conditional execution - the statement is executed only if a specific condition is met. This improves performance and eliminates branching statements.

4. Enhanced instructions - ARM processors support enhanced DSP instructions for digital signal operations.

There are two software products that provide functionality for executing firmware without using a development environment: STM32 ST-LINK Utility and ST Visual Programmer.

STM32 ST-LINK Utility is designed to work with 32-bit controllers via ST-LINK interface.

The general characteristics of the core of STM32 microcontrollers are presented in table 1.

Table 1. Main characteristics of the core of STM32 microcontrollers

Characteristic	*Value*
data word width, bit	32
architecture	harvard
conveyor	3- stepped
set of instructions	RISC
organization of program memory, category	32
prefetch buffer, bit	2x64
average instruction size, bytes	2
interrupt type	vectorized
interrupt response delay	12 cycles
power management modes	sleep, sleep on exit, deep sleep
debug interface	ST-LINK, JTAG

Microcontrollers of this type are built on the Harvard architecture and have a 3-stage pipeline that minimizes command execution time. They are designed to build systems with maximum energy efficiency and have multiple power management modes. They use internal memory interfaces that are wider than the average instruction length. This minimizes the number of accesses to the memory bus, and hence the power consumption associated with bus operations and non-volatile memory reads. Continuous interrupt processing technology with the exception of internal stack operations reduces the response time to interrupts and eliminates unnecessary stack operations.

The STM32F4 Discovery Board is equipped with:

- A stm32f407vgt6 microcontroller with a Cortex-M3 core clocked at 168 MHz, 1 MB of Flash memory, 192 KB of RAM;
- ST-Link debugger for debugging and programming;
- power the Board via USB or an external 5V power supply;
- ST MEMS LIS302DL motion sensor and digital accelerometer outputs;
- sound sensor MP45DT02, ST MEMS;
- CS43L22 audio DAC;
- eight LEDs;
- two buttons (for user programming and for restarting).

Thus, the debugging Board is equipped with a large number of peripherals, which allows you to implement algorithms of varying complexity on it.

2. Stage. An original and efficient algorithm for lossless compression of hyperspectral AI by regression transformation is developed. An algorithm for lossless compression of hyperspectral aerospace images, characterized by the use of channel-difference linear regression transformation, which significantly reduces the range of data changes and increases the compression rate due to this. The main idea of the transformation is an

algorithm that finds pairs of correlated channels and then creates lossless transformed blocks using regressin analysis, which reduces the size of the aerospace image channels and converts them before compressing the modified Huffman algorithm.

Several orthogonal transformations adapted for lossy compression of hyperpectral AI have been developed. An adaptive algorithm for discrete-cosine transformation with subsequent quantization with a loss level and compression by Huffman encoding is developed.

Adapted Huffman algorithm. The well-known standard Huffman code table is recommended for JPEG images, not for hyperspectral AI, so the generated one is suggested for encoding. Modification of the table is as follows:

1) each possible pair of RZ is assigned a natural number intMerge starting with two;

2) consider the binary notation of the number intMerge and pay attention to its length (lenMerge). For example, for (R;Z)=(4;7) intMerge = 117, binary entry -1110101, lenMerge = 7;

3) let's build the code as follows: take one (lenMerge - 1) times, assign 0 and the binary entry of the number intMerge without the leading unit. For clarity, after zero, we will add an apostrophe: 1111110’110101.

For an example from the table: (R;Z) = (14; 1) intMerge = 17, binary entry-10001, lenMerge = 5. Huffman Code: 11110 ' 0001.

The number Z does not exceed 14, since we replace the sequence of 15 zeros with the code 111111110 ' 00000000 (as if intMerge was 256). The code for the end of the block is 00. The pair (R;Z)=(15;0) is encoded as 01. If R exceeds 16, we assign intMerge, starting from 257, in the same way.

This modification made it possible to increase the compression rate of hyperspectral AI by analyzing the frequencies of the R and Z pairs, since codes of shorter length are allocated to more common values during encoding.

At the second stage, an algorithm for multithreaded compression processing was developed and organized to improve the computational efficiency of compression.

Algorithm for multiple streams of hyperspectral AI encoding.

1. determining the number of encoded pairs in the set of ordered pairs (CBM);
2. the first thread receives and encodes the main generating file;
3. the second and subsequent streams encode subsequent pairs in the CBM;
4. upon release, all subsequent streams receive the next pair of channels until the CBM is empty;
5. when encoding is complete, we release all threads.

The algorithm is a single thread during decoding.

1. consider the next pair in the CBM. Let it be k1 and k2, where k1 is the generating file and k2 is the regressed file;
2. if the file k1 is the main generating file, or it is already decoded, then we decode the pair (k1- k2);
3. if the file k1 has not been decoded yet (it may be in the process), then we consider the next pair after k1, k2 in the CBM (without discarding it);
4. go to step 2 – if such a pair is found, go to standby mode-if not.
5. at the end of decoding, we notify the rest of the threads in waiting, go to step 1 (if the CBM is not empty), or complete the execution of the thread). In this case, we discard the encoded pair (k1 - k2) from the CBM.

2 Experiments of the developed hardware implementation algorithms

To determine the effectiveness of the proposed adaptive algorithm in terms of compression ratio and computational efficiency, as well as the limits of its applicability, a number of experiments were performed on hyperspectral AI (Aviris remote sensing system), (Fig.1).

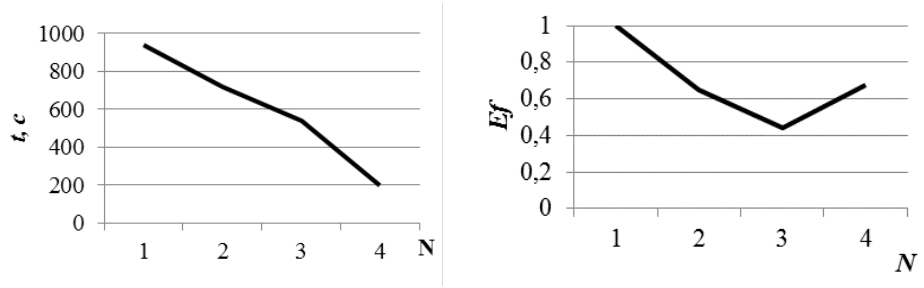

Fig. 1. Adapted compression algorithm for hyperspectral AI:
a) calculation time; b) execution efficiency

The experiments were performed on a PC with an IntelCore i7, 2.5 GHz processor and 4 GB of RAM running the Windows 10 operating system

JPEG Lossless/ Lossy compression algorithms, 7z archiver, and Winrar were used as analogs. The results of compression experiments using the proposed method are presented in tables 2-3. For comparison, the compression ratio and processing speed averaged over all experiments are taken

Table 2. Comparison of lossless compression algorithms

Compression algorithm	*Compression ratio*	*Calculation time, s*
JPEG Lossless	3,5	221452
7z	3,8	205146
Winrar	3,9	220738
Adaptive algorithm	5,9	74816

Table 3. Comparison of lossy compression algorithms

Compression algorithm	*Compression ratio*	*Calculation time, s*
JPEG Lossy	5,7	225672
Adaptive algorithm	9,2	78235

It should be noted that in these experiments, the adaptive algorithm shows results in a compression ratio higher than the known solutions by more than 55 %, and significantly exceeds the processing speed by 3 times.

Conclusion

Based on the results of the study, it is shown that the proposed method for developing lossless and lossy compression algorithms based on hardware implementation is highly effective in the compression ratio and processing speed. This is due to the fact that the algorithms were adapted for this implementation in order to be able to be used on Board the spacecraft. This method of constructing algorithms has a high speed due to the transition to hardware that operates with bits, providing an increase in speed at times.

Adaptive algorithms can be used both in remote sensing systems and in any other applications for processing and compressing hyperspectral AI.

References:

1. Gonsales R., Vuds R. Cifrovaya obrabotka izobrazhenij. - M.: Tekhnosfera, 2012. – pp. 55-67. (In Russian).
2. Kashkin V. B., Suhinin A. I. Cifrovaya obrabotka aerokosmicheskih izobrazhenij // Krasnoyarsk: SFU. 2008. 278 p. (In Russian).
3. Gashnikov, M.V. Bortovaya obrabotka giperspektral'nyh dannyh v sistemah distancionnogo zondirovaniya Zemli na osnove ierarhicheskoj kompressii / M.V. Gashnikov, N.I. Glumov // *Komp'yuternaya optika*. – 2016. – vol. 40, no 4. – pp. 543–551. (In Russian).
4. Petrov E.P., Harina N.L., Suhih P.N. Metod szhatiya mnogorazryadnyh sputnikovyh snimkov bez poter'. Sovremennye problemy distancionnogo zondirovaniya Zemli iz kosmosa. 2016. vol. 13. no 2. pp. 203–210. (In Russian).
5. Fernando García-Vílchez, Jordi Muñoz-Marí, Maciel Zortea, Ian Blanes, Vicente González-Ruiz, Gustavo Camps-Valls, Antonio Plaza. On the Impact of Lossy Compression on Hyperspectral Image Classification and Unmixing. *IEEE Geoscience and remote sensing letters*, vol. 8, no. 2, 2011.
6. Meena B. Vallakati and Dr. R. R. Sedamkar. Low Complexity DCT-based DSC approach for Hyperspectral Image Compression with Arithmetic Code. *IJCSI International Journal of Computer Science Issues,* vol. 9, Issue 5, no 1, September 2012.
7. Yongjian Nian, Mi He, and Jianwei Wan. Low-Complexity Compression Algorithm for Hyperspectral Images Based on Distributed Source Coding. Hindawi Publishing Corporation. *Mathematical Problems in Engineering*. Vol. 2013, Article ID 825673, 7 p.
8. Ganeshraj P. and Sivasankar A. Scalable Compression Method for Hyperspectral Images // *Research Journal of Engineering Science.* vol. 2(3), pp.1–5. 2013.
9. Diego Valsesia, Enrico Magli . A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images. 2014.
10. Dr. S.M.Ramesh, P.Bharat, J.Anand, J.Anbu Selvan. Analysis of Lossy Hyperspectral Image Compression Techniques // *International Journal of Computer Science and Mobile Computing*, Vol.3 Issue.2, February – 2014, pp. 302–307.
11. A. Sarinova, A. Zamyatin. E3S Web Conf **149**, 02003 (2020).
12. S. Kudubayeva, N. Amangeldy, A. Sundetbayeva and A. Sarinova. *ACM International Conference Proceeding Series* **8**, (2019)**.**
13. A. Sarinova, A. Zamyatin and P. Cabral. DYNA (Colombia) **82** (190), 166-172 (2015).