

# Elimination of Information Redundancy of Hyperspectral Raster Images Technology Enhancement

Dmitriy Vasin <sup>1</sup>, Pavel Pahomov <sup>1</sup> and Sergey Rotkov <sup>2</sup>

<sup>1</sup> National Research Lobachevsky State University of Nizhny Novgorod, Prospekt Gagarina, 23, Nizhny Novgorod, 603950, Russia

<sup>2</sup> Nizhny Novgorod State University of Architecture and Civil Engineering, Ilyinskaya street, 65, Nizhny Novgorod, 603952, Russia

## Abstract

The work is a continuation of the authors' research on the problem of adaptive compression of raster hyperspectral images of Earth remote sensing. In the first part of the article, the authors give an overview of the current state of affairs in the processing of images of remote sensing of the Earth, the characteristic properties of raster hyperspectral images in the context of the prospects for lossy compression, the problems of the effectiveness of existing compression methods of this type of graphic documents are indicated. Further, the article highlights the issues of increasing the efficiency of methods for eliminating information redundancy of raster hyperspectral images of Earth remote sensing. The problems of designing and creating parallel methods and algorithms for the compression of raster hyperspectral ERS images are considered. A method for the development of a parallel algorithm for constructing a system of local homogeneous "well-adapted" basis functions for raster hyperspectral images, based on the Chebyshev approximation for systems using the CUDA graphics processor, is proposed.

## Keywords

Raster remote sensing hyperspectral data, adaptive compression of hyperspectral remote sensing data, design of parallel algorithms for raster data processing

## 1. Introduction

In recent years, there has been a significant increase in interest in the development of information technologies related to the receipt, processing, transmission and storage of Earth remote sensing data. Moreover, the work is carried out both in the hardware and software segments: on the one hand, the equipment of the spacecraft themselves is constantly being improved, the existing ground infrastructure is developing, the network of ground control stations and the range of their functional capabilities and tasks are being modernized and expanded, and on the other hand, they are being developed and highly efficient methods of computer processing of Earth remote sensing data are being improved, including high-resolution space images and the latest hyperspectral data, new software systems for their processing are being created, etc.

Earth remote sensing data are used in solving a wide range of problems of ensuring state security, environmental monitoring and environmental protection, prevention and analysis of natural disasters and various emergencies, urban infrastructure development and many other pressing problems.

Most of the modern methods for processing remote sensing data of the earth's surface are aimed at solving the problems of preprocessing and analyzing the received space images, identifying and recognizing thematic objects on them in order to create digital models of the Earth's surface, as well as solving the problems of monitoring the selected objects. Along with classical methods, hybrid methods of remote sensing data analysis are being developed using technologies for high-performance processing of large volumes of video data and artificial neural networks [1].

---

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia

EMAIL: dm04@list.ru (D. Vasin); reg\_edit@bk.ru (P. Pahomov); rotkovs@mail.ru (S. Rotkov)

ORCID: 0000-0002-4341-2457 (D. Vasin); 0000-0002-1562-7610 (P. Pahomov); 0000-0002-0662-7619 (S. Rotkov)



© 2021 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

The creation of the latest global Geoinformation Monitoring Systems of a new generation, along with the development of the mathematical apparatus, requires the accumulation of a data bank of space images of the Earth's surface, including dual-purpose space data obtained from various sources: spacecraft, ground and surface stations, flying laboratories, etc. also the creation of a network of ground-based information processing centers. The relevance of such monitoring systems is determined by the possibility of creating on their basis highly efficient IT technologies for performing various interdisciplinary research [1].

Currently, high-performance technologies are widely used to process Earth remote sensing data on computing clusters with GPUs [2, 3].

The study [4] presents the results of the practical use of hardware and software systems based on a distributed grid–infrastructure of supercomputer resources for solving resource-intensive applied problems of remote sensing of the Earth.

Note that the growth of complex problems, the solution of which is associated with the use of modern IT technologies, leads to the need to use parallel computing. Parallel computing is interdisciplinary in nature. They affect, in particular, such areas as numerical methods, structures and algorithms for data processing, hardware and software, systems analysis. This allows you to apply the knowledge gained in the study of parallel computing in various areas of scientific and practical activities.

## **2. Characteristic properties of space hyperspectral images in the context of compression perspectives**

At present, remote sensing devices, placed on airplanes and spacecraft, are quite effective in solving various problems of monitoring the earth's surface [5]. Most of these systems render a scene image using a multichannel shooting mode. A promising type of such a regime is hyperspectral imaging, which covers the optical and near infrared ranges of electromagnetic radiation, has high spectral and good spatial resolutions and simultaneously forms from several hundred to one and a half thousand raster hyperspectral images that are practically combined with each other. The peculiarities of hyperspectral imaging determine a high degree of redundancy of the resulting raster images, since the images in two adjacent spectral channels usually have a correlation coefficient close to one, because for neighboring spectral channels, the difference in wavelengths is minimal [5–11, 15, 16].

## **3. Brief characteristics and problems of efficiency of existing methods of compression of space raster hyperspectral images**

All existing methods of image compression in general are divided into two groups, providing lossless and lossy compression.

Lossless compression methods provide not very high compression ratio of raster hyperspectral images of Earth remote sensing ( $\sim 3 - 4$ ), but their use does not introduce any distortions into decoded images [8], which is their indisputable advantage. However, since the obtained values of the compression ratio do not always meet the requirements of practice, the problem of developing and applying methods of compression of raster hyperspectral images with losses [9 - 12], which form higher values of the compression ratio at the output compared to the methods of the first group, is urgent.

Lossy compression methods use an approach based on the expansion of the initial signals in terms of a particular system of basis functions with a given approximation accuracy  $\varepsilon$ . In this case, the optimal coding of raster hyperspectral images will be provided only by such a system of basis functions, which, for a given root-mean-square error  $\delta$ , provides the minimum, or close to it, the number of basis functions  $\varphi^1(t), \varphi^2(t), \dots, \varphi^m(t)$ . Then the process  $f(t)$  ( $t_1 \leq t \leq t_2$ ) can be approximately represented in the form

$\tilde{f}(t) = \sum_{k=1}^m C_k \varphi^k(t)$  of basis functions  $\varphi^1(t), \varphi^2(t), \dots, \varphi^m(t)$ . The coefficients  $C_1, C_2, \dots, C_m$  are

considered as a code of the curve  $f(t)$ . The approximation error in this case is:  $\varepsilon(t) = f(t) - \sum_{k=1}^m C_k \varphi^k(t)$ .

Obviously, different types of encoded hyperspectral data will require different optimal systems of basis functions.

To be able to apply the methods of this group, the original raster hyperspectral data must be converted into a set of vectors obtained as a result of sequential line-by-line scanning of the original rasters.

Certain difficulties in using these methods are associated with the fact that the choice of an informative system of basis functions is mainly determined either from the experience and intuition of the researcher, or from the external similarity of the initial signal with the form of basis functions. This can lead to the fact that the optimal, or close to it, the system of basis functions may not be found, especially in the case of a complex form of the initial signal, which is certainly characteristic of the studied raster images.

The solution to the problem can be found as a result of the use of objective methods for choosing a system of bases, in which they do not rely on human knowledge, but proceed only from the properties of a set of initial continuous data. One of such optimal basis systems is the system of eigenvectors of the covariance matrix calculated for a given set of initial data, and the corresponding eigenvalues characterize the accuracy of the approximation. However, this method is quite cumbersome in terms of computations, in addition, the system of basis functions obtained in this way allows, with a given accuracy, to encode not every function of the original set of signals, but only on average over the set. In this case, the main part of the data will be encoded with a given approximation accuracy, however, the encoding of extreme data can occur with a significant error. Obviously, when coding hyperspectral raster images, this is an undesirable property of the resulting basis system.

The problem of compression of raster hyperspectral images of the earth's surface remains a rather urgent problem of image processing and information transfer. Despite the abundance of the proposed compression methods and software compression systems created on their basis, the problem of increasing the resource efficiency of the proposed methods remains quite acute.

In [12], the authors proposed a sequential quasi-optimal algorithm for eliminating the information redundancy of raster hyperspectral images of remote sensing of the Earth with controlled losses based on the construction of a system of local homogeneous "well-adapted" basis functions [17, 18]. Practical use of the algorithm revealed its sufficient efficiency in terms of the obtained compression ratio of raster hyperspectral images, however, the temporal efficiency of its operation can hardly be called satisfactory.

#### **4. Statement of the problem**

Develop a parallel algorithm for the formation of a system of local homogeneous "well-adapted" basis functions for compression with controlled losses, and analyze its efficiency in comparison with the sequential version when encoding a sample of 360 grayscale 16-bit raster hyperspectral images of the Earth's surface (spectral channel data).

#### **5. Problems of design and creation of parallel methods and algorithms for compression of raster hyperspectral images**

The main task of increasing the efficiency of high-performance tools used for processing streams of hyperspectral rasters of the earth's surface is the parallelization of computations. To solve this problem, two approaches are distinguished [19], which consist in:

1. dividing each set of raster images of spectral channels into separate fragments - streams 1, 2, ..., N elements - pixels, each of which is processed separately in a parallel mode, while the division

goes along the width and / or height of the image, and also due to portioned loading of snapshot data into RAM;

2. formation of data streams 1, 2, ..., N, each of which is a combination of groups of raster images of different spectral channels. Each of these generated streams is processed separately in parallel. The total number of channels is divided evenly by the number of processing cores of the processor(s).

Parallel computing systems are physical computer systems, as well as software systems that implement in one way or another parallel data processing on many computing nodes, which is most suitable for the problem being developed, taking into account the fact that most modern computer systems are multiprocessor. The idea of parallelizing computations is based on the fact that most problems can be divided into a set of smaller problems that can be solved simultaneously.

The main idea of the distribution of the computational process is the transition from parallelism at the instruction level to parallelism at the data and task level. The development of methods and algorithms that implement parallel computing mechanisms for solving complex scientific and technical problems is often a significant problem.

When developing software, we will rely on a cascade with feedback model, as one of the basic ones when creating parallel methods [21]. This model allows, based on the operational analysis of the effectiveness of the developed computational schemes, at any stage of development, to return to the previous steps. Efficiency is assessed by analyzing parameters such as speedup, efficiency, scalability generated by parallel computing.

When implementing a parallel version of the algorithm for forming a system of local homogeneous "well adapted" basis functions, we take into account that the computational scheme for solving the problem is known. This determines the composition and order of subsequent actions to determine effective ways of organizing parallel computing. Let us highlight the three most essential steps [19–21]:

- sequential analysis of the possibility of independent implementation of individual blocks and components of the code being developed with further segmentation into a set of subtasks. Successful code segmentation will inevitably speed up your program;
- determination of a set of information interactions between individual subtasks in the process of solving the original problem, ensuring the minimization of such interactions;
- selection of the necessary computing system and implementation of the distribution of the formed set of subtasks among the processors. At the same time, we take into account that the computational load on the processors must be balanced.

The analysis shows that most often a return is carried out in order to correct from the last step to the first step of the given scheme. This is due to the fact that after determining the available number of processors, it is the composition of the formed set of tasks that must be adjusted. At the same time, in the presence of a small number of processors, subtasks can be enlarged, or, conversely, detailed. Some developers distinguish these actions as an independent stage of parallel computing design and are defined as scaling of the developed algorithm [21].

As a result, we can distinguish 4 fundamental stages of the process of developing parallel algorithms:

1. decomposition of the task into subtasks that are implemented independently;
2. determination of information interactions for the formed set of subtasks;
3. scaling of subtasks, determining the number of processors;
4. definition of system architecture, assignment of subtasks to processors, scheduling.

Steps 1-4 can be repeated as needed, for example, to improve the efficiency of the algorithm. If the desired indicators are not achieved, then the mathematical model of the problem should be changed.

The above scheme is general, in this regard, in each specific case, the sequence of steps may vary.

*Decomposition stage.* The original task is analyzed and segmentation is carried out into a set of basic subtasks. At the same time, the minimum requirements are imposed to ensure the same amount of computational load in the allocated subtasks and to ensure the minimum information exchange between the processors.

*The stage of analyzing information dependencies between subtasks.* There are 4 types of dependencies between executed subtasks: local or global, structural or arbitrary, static or dynamic, synchronous or asynchronous. Local data transfer schemes are used for neighboring processors, in

contrast to global schemes, in which all processors take part. According to the methods of interaction, structural, corresponding to typical communication topologies, and arbitrary are distinguished. Static constraints are defined at design time, and dynamic constraints are defined during calculations). If the next operation is performed after the previous one has been completed by all processors, then a synchronous method of interaction is selected, and if the processes cannot wait for the complete completion of data transfer actions, then we are dealing with an asynchronous method of interaction. At the same time, subtasks have a high degree of informational interdependence.

*Scaling stage.* Executed if the number of subtasks differs from the number of processors. Then the transition to the stage of decomposition is carried out. At the same time, the number of subtasks is reduced due to the enlargement of the source data area. First of all, it is necessary to combine areas for which subtasks have a high degree of informational interdependence.

*The stage of assigning tasks to processors.* At this stage, we take into account the presence of information interactions between the data areas of these tasks and, if they exist, we place such tasks on processors connected by direct data transmission lines.

When developing a parallel algorithm, we take into account that it is characterized by data parallelism and parallelism of tasks, and its efficiency directly depends on the ratio of the time spent on performing computations on fragments of the initial data and the time for transferring data.

In the presence of data parallelism, the task is reduced to dividing the original data array into fragments with their subsequent independent processing on different processors, subject to the requirement for their relatively uniform loading, taking into account their possible different performance.

When there is task parallelism, there is no data parallelism. Then the original task is split into several independent subtasks, and each of them, ideally, is sent to a separate processor. The number of subtasks affects the number of processors. By ensuring that the processors are loaded evenly and the data exchange between them is minimized, significant acceleration can be expected.

Comparing the times spent by different parts of the program, we identify its most resource-intensive parts, which allows us to judge the efficiency of the generated code.

The authors have chosen the CUDA platform as a software and hardware solution that provides a sufficient level of development tools and high efficiency of calculations on the graphics processor. To get a significant increase in performance relative to the central processor, it is important to take into account that the structure of the graphics processor is significantly different from the central processor and the very approach to organizing computing programs is different. So, a graphics processor (in this case, we are talking about CUDA compatible processors) is capable of executing thousands of threads simultaneously. This is possible due to the organization of computational flows in blocks of 32. Each block is included in the block grid. The GPU itself consists of multiprocessors, which contain several grids with blocks of threads. At the same time, next to the GPU, there is a high-bandwidth graphics memory common to all threads. It is important that the exchange of data between RAM and graphics memory is relatively slow, and therefore graphics memory must be used as efficiently as possible for the entire parallel computing process, minimizing data loading and unloading. In addition, the computational cycles themselves must be assembled in such a way that they occupy as many threads as possible per unit of time. As a result, at stage 3 of the process of constructing parallel algorithms stated above, it is necessary to additionally distribute the data in the graphics memory in such a way that it is possible to organize the execution of simple computations many times, while occupying all blocks and all flow grids with computations.

It was shown in [12] that the process of constructing the unit vectors of a system of local homogeneous "well-adapted" basis functions consists of a set of stages described by formulas (1) and (2):

$$U^{L+1} = \left\{ x^{S_{L+1}} - \tilde{x} - \sum_{k=1}^L \left( x^{S_{k+1}} - \tilde{x}, U^k \right) U^k \right\} \times \\ \times \left\| x^{S_{L+1}} - \tilde{x} - \sum_{k=1}^L \left( x^{S_{k+1}} - \tilde{x}, U^k \right) U^k \right\|^{-1} \quad (1)$$

where  $\tilde{x} = \frac{1}{N} \sum_{j=1}^N x^j$ , and the point  $x^{S_{L+1}}$  is chosen so that

$$\begin{aligned} \lambda_{L+1} &= \left\| x^{S_{L+1}} - \tilde{x} - \sum_{k=1}^L (x^{S_{k+1}} - \tilde{x}, U^k) U^k \right\| = \\ &= \max_j \left\| x^j - \tilde{x} - \sum_{k=1}^L (x^j - \tilde{x}, U^k) U^k \right\|. \end{aligned} \quad (2)$$

To implement them, a sequential search is required until the  $\lambda_{L+1} < \varepsilon$  condition is satisfied, where  $\varepsilon$  is the specified approximation accuracy. This fact introduces significant complexity in the application of parallel computing to solve this problem.

It is seen that finding each next basic unit vector of the system of “well-adapted” basic functions is computationally laborious. It is also obvious that it is also impossible to perform all computational operations on the GPU on one set of data at a time by dividing the entire array of computations into many threads. As one of the technological options on the way of constructing a high-performance implementation, it is possible to single out the computations of scalar products in a separate computational procedure, since such computations can be efficiently performed in parallel. In addition, finding the maximum length of a vector can also be organized as a separate computational procedure in such a way that all lengths are stored in a buffer and then the maximum is selected. Thus, as a prototype, a combination of calculations on the central processor and the graphics processor was obtained, where the finding of each subsequent basis vector is organized by a cycle on the central processor, in the body of which the procedures for finding scalar products over the entire data set and finding the maximum length are performed.

Computational experiments show a reduction in computational time (depending on the amount of initial data) by at least an order of magnitude. Obviously, the result obtained is not optimal and intermediate. Research in this area will be continued by the authors in the future.

## 6. References

- [1] N. S. Abramov, D. A. Makarov, A. A. Talalaev, V. P. Fralenko, Modern Methods for Intelligent Processing of Earth Remote Sensing Data, Program Systems: Theory and Applications 9(4) (2018) 417-442.
- [2] E.V. Rusin, Technologies for the Processing of Earth Remote Sensing Data on NKS-30T+GPU Hybrid Cluster, Interekspo Geo-Sibir', in: Proceedings of the International Scientific Conference “Remote Sensing Methods and Photogrammetry, Environmental Monitoring, Geoecology”, Vol. 4(1), 2016, pp. 46 - 49.
- [3] E.V. Rusin, Technology of high performance image processing on multiprocessor computer, Pattern Recognition and Image Analysis 22(3) (2012) 470–472.
- [4] M.E. Kuleshova, N.I. Paramonova, N.N. Paramonov, O.P. Chizh, Development of technological solutions in creating and using specialized hardware-software complexes based on grid-infrastructure of supercomputer resources, in: Proceedings of the International Conference “Russian Supercomputing Days 2015”, CEUR Workshop Proceedings, vol. 1482, 2015, pp. 603-610.
- [5] Chang Chein-I., Hyperspectral Imaging: Techniques for Spectral Detection and Classification, Plenum Publishers, N.Y.: Kluwer Academic, 2003.
- [6] M.A. Popov, S.A. Stankevich, Methods of Spectral Channels Number Optimization in the Tasks of Processing and Analysis of Earth Remote Sensing Data, Modern problems of Earth remote sensing from space, Moscow: RAS Research Institute, 2006, Vol.3, T.1, pp. 106-112.
- [7] V. Lukin, Processing of Multichannel RS data for Environment Monitoring, in: Proceedings of NATO Advanced Research Workshop on Geographical Information, Processing and Visual Analytics for Environmental Security, Trento, Italy, Springer Netherlands, 2009, pp. 129-138.

- [8] A. Kaarna, Compression of Spectral Images, Vision Systems: Segmentation and Pattern Recognition, Ed. By G. Ohinata and A. Dutta, Vienna: I-Tech, 2007, pp. 269-298.
- [9] G. Yu, T. Vladimirova, M.N. Sweeting, Image compression systems on board satellites, *Acta Astronautica* 64 (2009) 988-1005.
- [10] N.N. Ponomarenko, V.V. Lukin, M.S. Zriakhov, A. Kaarna, J. Astola, Automatic Approaches to OnLand/OnBoard Filtering and Lossy Compression of AVIRIS Images, Proceedings of IGARSS, Boston, 2008, Vol. III, pp. 254-257.
- [11] G. Motta, F. Rizzo, J.A. Storer, Compression of hyperspectral imagery, Proceedings of Data Compression Conference, 2003, pp. 333-342.
- [12] D.Y. Vasin, V.P. Gromov, P.A. Pakhomov, Elimination of information redundancy of hyperspectral images using the "well- adapted" basic method, *Journal of Physics: Conference Series. V International Conference on Information Technology and Nanotechnology, ITNT 2019*, 2019, p. 032025. doi: 10.1088/1742-6596/1368/3/032025
- [13] E. Christophe, C. Mailhes, P. Duhamel, Hyperspectral Image Compression: Adapting SPIHT and EZW to Anisotropic 3- D Wavelet Coding, *IEEE Transactions on Image Processing*, 2008, Vol. 17, № 12, pp. 2334-2346.
- [14] M.J. Ryan, J.F. Arnold, A Suitable Distortion Measure for the Lossy Compression of Hyperspectral Data, Proceedings of IGARSS, 1998, pp. 2056-2058.
- [15] N. Ponomarenko, V. Lukin, M. Zriakhov, A. Kaarna, Preliminary automatic analysis of characteristics of hyperspectral AVIRIS images, Proceedings of MMET, Kharkov, Ukraine, 2006, pp. 158-160.
- [16] W.K. Pratt, *Digital Image Processing*. Fourth Edition, NY, USA, Wiley- Interscience, 2007.
- [17] Yu.I. Neymark, Z.S. Batalova, Yu.G. Vasin, M.D. Breido, *Raspoznavaniye obrazov i meditsinskaya diagnostika*, Moscow: Nauka, 1972. (in Russian)
- [18] Yu.I. Neymark, Yu.G. Vasin, Kodirovaniye bol'shikh massivov informatsii v svyazi s zadachami raspoznavaniya obrazov, *Izv. Radiophysics*. 1968, №7, pp. 1081–1086. (in Russian)
- [19] V.G. Bondur, Modern Approaches to Processing Large Hyperspectral and Multispectral Aerospace Data Flows, *Earth Exploration from Space*, 2014, № 1, pp. 4-16.
- [20] V.A. Shpenst, D.A. Egorov, Methods of paralleling the computational algorithm of image formation in radar stations with synthesized aperture, Proceedings of the Russian Academy of Rocket and Artillery Sciences (St. Petersburg) 1(67) (2011) 101-108.
- [21] V.P. Gergel, R.G. Strongin, *Osnovy parallel'nykh vychisleniy dlya mnogoprotsessornykh vychislitel'nykh system*, Nizhny Novgorod: Lobachevsky State University, 2003. (in Russian)