Parallel computations in problems of reconstruction of distorted images in spatial-spectral form

Boris V. Kostrov
Department of Electronic Computers
Ryazan State Radio Engineering University named after V.F. Utkin
Ryazan, Russia
kostrov.b.v@evm.rsreu.ru

Nataliya N. Grinchenko
Department of Electronic Computers
Ryazan State Radio Engineering University named after V.F. Utkin
Ryazan, Russia
grinchenko_nn@mail.ru

Angelina A. Vyugina
Department of Electronic Computers
Ryazan State Radio Engineering University named after V.F. Utkin
Ryazan, Russia
vyugina.an.a@gmail.com

Svetlana N. Baranova
Department of Electronic Computers
Ryazan State Radio Engineering University named after V.F. Utkin
Ryazan, Russia
baranova.sv.n@gmail.com

***Abstract*— The object of the study is the image processing process in preparation for data transmission, as well as subsequent recovery. The subject of the study is the application of parallel computing in image processing tasks. The purpose of the article is to study the method of image reconstruction with correction of distortion of the transmission vector based on the input of a value in the trend of its neighbors. The relevance of this topic is determined by the need to effectively perform the operations preceding and following the transmission over the communication channel. During the experimental part, the results were obtained in the form of execution time values with sequential execution and using parallel calculations, which gave the expected increase in speed-up. Since the image consists of unrelated parts, it can be successfully processed by applying data parallelization.**

Keywords*— *image processing, bit plane, parallel computing, complex error, recovery, data transfer, MATLAB.

I. INTRODUCTION

In the modern information world, all information is collected, stored, and transmitted via communication channels. Every year the amount of information generated by various media increases, that leads to the problem of improve the efficiency of data transmission and processing. Various kinds of interference may occur during data transmission [1]. Therefore, the issue related to the process of forming a data vector, its transmission and subsequent recovery is an integral part in solving the problem of data processing, including graphical information. After receiving the data by the receiving party, they must be restored to their original form, however, not the entire sequence can be accepted correctly, which leads to the need to calculate and correct distorted values. The calculations performed after receiving graphic data, especially if it is necessary to restore the parts damaged by interference, can take a large amount of time resources, which implies the task not only to correct the distorted pixels, but to perform these operations with sufficient speed. Parallelization can be carried out on different devices: multicore CPUs or video cards with a large number of stream processors.

When performing preprocessing of images for the formation of the transmission vector, as well as recovery after receiving data, methods based on the processing of independent parts of the vector are used. The distribution and independence of data makes it possible to effectively divide the execution into several nodes working in parallel.

The aim of the paper is to study the effectiveness of parallel computing in the tasks of preprocessing and postprocessing images for transmission over a noisy communication channel.

II. FORMATION OF THE TRANSMISSION VECTOR

The image can be represented in a spatial form, the elements of which are brightness values, for each of which a byte of memory is allocated (1).

$$\mathbf{G}_{N\times N} = \begin{pmatrix} g_{1,1} & \cdots & g_{1,N} \\ \vdots & \ddots & \vdots \\ g_{N,1} & \cdots & g_{N,N} \end{pmatrix}, \qquad (1)$$

where **G** is a matrix in spatial representation, $g_{i,j}$ is the brightness value of one pixel, N×N is the image size.

Data transmission via communication channels takes place at the physical level, while it is more convenient to represent discrete values of image brightness in a spatial-spectral form, which is formed in accordance with a certain function [2,3]. In the conducted study, Walsh functions [4] were used as non-trigonometric orthogonal basis functions, which were superimposed on separate blocks of values with a size of 32x32 (2) to decrease the value of the spectral component that reduces the amount of memory needed to store the image in a spatially spectral form.

$$\boldsymbol{G} = \begin{pmatrix} G_{1,1} & \cdots & G_{1,\frac{N}{n}} \\ \vdots & \ddots & \vdots \\ G_{\frac{N}{n},1} & \cdots & G_{\frac{N}{n},\frac{N}{n}} \end{pmatrix}, \qquad (2)$$

where **G** is a matrix in spatial representation, $G_{i,j}$ is a 32x32 submatrix, N is the number of pixels, n is the number of 32x32 blocks.

Splitting the image into parts allows independent pixel processing. Each of the individual blocks is transformed to a spatially spectral form by one method, which means that the computational process can be parallelized for each submatrix of pixels.

As a result of the transformation, a matrix is obtained representing the image in the spatial-spectral form (3).

$$\mathbf{F}(G)_{N\times N} = \mathbf{G} \times \mathbf{H} = \begin{pmatrix} f_{1,1} & \cdots & f_{1,N} \\ \vdots & \ddots & \vdots \\ f_{N,1} & \cdots & f_{N,N} \end{pmatrix}, \quad (3)$$

where $\mathbf{F}$ is the matrix of values of the function of G, $\mathbf{G}$ is the matrix of image elements, $\mathbf{H}$ is the matrix of basic functions, $f_{i,j}$ is the value of the spectrum [5], N×N is the size of the spectrum matrix.

To form a data transmission vector, it is necessary to represent the spatial-spectral form of the matrix in binary representation: one value will be stored in an 8-bit representation (4) [6].

$$\mathbf{F}_{(N\times N)\ 10} = \mathbf{F}_{((N*8)\times N)\ 2}, \quad (4)$$

where $\mathbf{F}$ is the matrix of values of a function of G, N×N is the size of the spectrum matrix, N*8 is the number of values in the bit representation.

The number of bit planes (k) takes values from 1 to 8, since one byte of memory is allocated for the spectrum. The formation of bit plane matrices is performed in accordance with expression (5) [7].

$$\mathbf{B}_{N\times N}^{k} = \begin{pmatrix} f_{1,k} & f_{1,k+8} \cdots & f_{1,k+8*N} \\ \vdots & \ddots & \vdots \\ f_{N,k} & f_{N,k+8} \cdots & f_{N,k+8*N} \end{pmatrix}, \quad (5)$$

where $\mathbf{B}$ is the matrix of the bit plane, k is the bit number of the byte whose plane is formed, k=(1,8), $f_{i,j}$ is the value of the corresponding bit (0, 1).

When divided into bit planes, the original spatial-spectral matrix can be represented as sums of digits of the number (6).

$$\mathbf{F}_{N\times N} = 2^7 \begin{pmatrix} f_{1,1}^7 & \cdots & f_{1,8*N-7}^7 \\ \vdots & \ddots & \vdots \\ f_{N,1}^7 & \cdots & f_{N,8*N-7}^7 \end{pmatrix} + \\ +2^6 \begin{pmatrix} f_{1,2}^6 & \cdots & f_{1,8*N-6}^6 \\ \vdots & \ddots & \vdots \\ f_{N,2}^6 & \cdots & f_{N,8*N-6}^6 \end{pmatrix} + \cdots + \\ +2^0 \begin{pmatrix} f_{1,8}^0 & \cdots & f_{1,8*N}^0 \\ \vdots & \ddots & \vdots \\ f_{N,8}^0 & \cdots & f_{N,8*N}^0 \end{pmatrix}, \quad (6)$$

where $\mathbf{F}$ is a matrix in the space-spectral representation, $f_{i,j}^k$ is the k digit of the binary representation of the spectrum value.

The image transmission vector is formed by digits, that is, from the sign plane, and then from the highest bit to the lowest bit (7) [8].

$$V_{1\times N*8} = \left(f_{1,1}^1, \dots, f_{N,N}^1, f_{1,1}^2, \dots, f_{N,N}^2, \dots, f_{1,1}^8, \dots, f_{N,N}^8\right), \quad (7)$$

where V is the transmission vector, N is the number of elements in the bit plane, $f_{i,j}$ is the value of the corresponding bit (0, 1), the superscript from 1 to 8 is the number of the bit plane.

The presented sequence of transformation of the source image to the transmission vector contains certain steps: conversion to a spectrum, splitting into bit planes, which can be distributed into separate processing units of 32x32 pixels, since there is no dependence between its component parts in the graphic data. An identical situation occurs on the receiving side: correction of distorted bits, conversion to brightness values.

The scheme of image transmission over the communication channel [9], including the process of its restoration, is shown in figure 1.

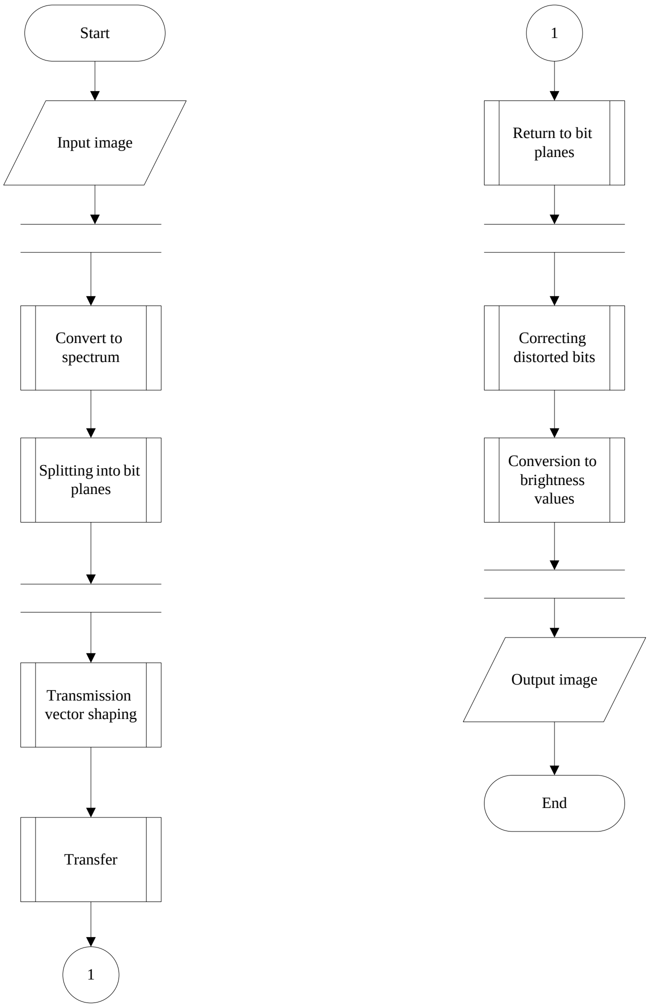


Fig. 1. Image transmission scheme

III. PARALLEL COMPUTING IN THE MATLAB ENVIRONMENT

Modeling of transformations of the source and received data, as well as the noisiest communication channel, was carried out in the Matlab environment. For this purpose, functions corresponding to the stages of the data transfer scheme have been developed (figure 1).

Matlab supports the use of parallel computing technologies on both CPU and GPU.

Implementation of parallel computing using a multicore processor uses the parallel loop operator parfor, which automates the creation of parallel pools and manages file dependencies [10].

Parallel computing on a GPU can be started using the "gpuArray" class, which supports automatic launch of standard functions using GPU [10].

The study was conducted on the principle of data separation, that is, each core processes a separate part of the image with the same function.

When running parallel computations, the main thread initializes workflows on nodes that correspond to the execution of a specific thread. When processing an image, parallelization is performed at the data level by dividing the original matrix of elements into separate blocks. When switching to bit planes, separate matrices appear containing bits of a certain digit (6), which are also independent of other matrices, which allows you to work on them in parallel. Figure 2 shows a scheme of parallel calculations when translating an image to bit planes.

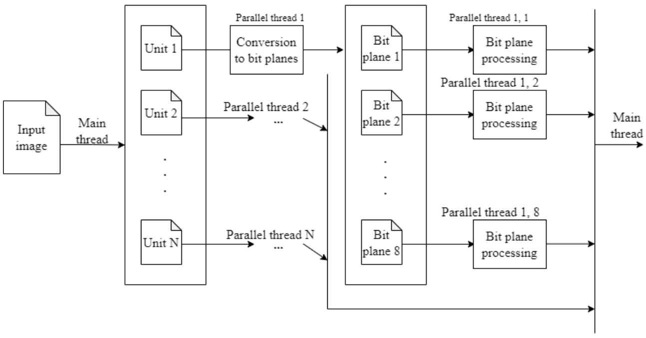


Fig. 2. Diagram of the processing process using parallel computing

The basic characteristic of determining the effectiveness of parallel computing is acceleration (8).

$$S_p = \frac{T_1}{T_p}, \tag{8}$$

where S_p is acceleration, T_1 is sequential execution time, T_p is parallel execution time.

IV. DISTORTION DURING DATA TRANSMISSION

When transmitting data through a communication channel, errors may occur due to various reasons: signal attenuation, noise, interference. Bit changes are considered distortions in the transmitted vector. There is a dependence of the effect of errors on the bit depth of the image: when an error hits the highest digit, the distortion introduced by the interference will affect the quality of the restored image more strongly than when the same error hits the lowest digit. This dependence is related to the weight of the digit in the binary representation of the number. All types of errors can be divided into 2 groups:

1) An error in one bit, which is a change to the opposite value of a random bit in the transmission vector.

2) A complex error, which is a change in a continuous sequence of bits. The extreme bits in the sequence with a packet error have a distorted as well as correct values may be contained inside such a sequence [11].

We study complex errors that differ in the nature of the distortion of the sequence of bits:

1) The sequence of bits takes the opposite value (error "NOT") (9).

$$V_{1\times N} = \{V_{(1,(1..i-1))}, \overline{V_{(1,(i..j))}}, V_{(1,(j+1..N))}\}, \tag{9}$$

where V is the transmission vector, N is the number of bits in the vector, i is the initial bit of the sequence with an error, j is the final bit of the sequence with an error.

2) The sequence of bits takes the value "0" (knocked out bits (KOB)) (10).

$$V_{1\times N} = \{V_{(1,(1..i-1))}, 0_{(1,(i..j))}, V_{(1,(j+1..N))}\}, \tag{10}$$

where V is the transmission vector, N is the number of bits in the vector, i is the initial bit of the sequence with an error, j is the final bit of the sequence with an error.

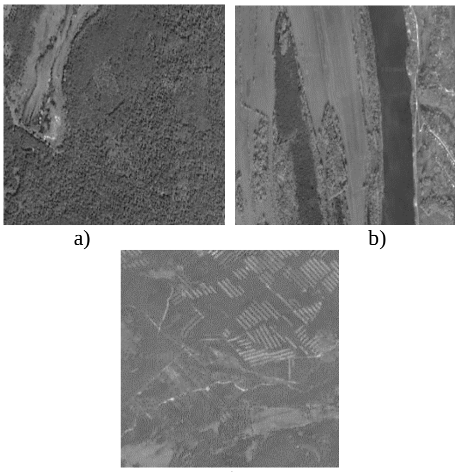
3) The sequence of bits takes values alternately "0" and "1" (batch error (BE)) (11).

$$V_{1\times N} = \{V_{(1,(1..i-1))}, E_{(1,(i..j))}, V_{(1,(j+1..N))}\}, \tag{11}$$

where V is the transmission vector, N is the number of bits in the vector, i is the initial bit of the sequence with an error, j is the final bit of the sequence with an error, $E_{(1\times M)}$={0..01..10..01..1}, where M is the sequence of the error vector.

V. THE IMPACT OF ERRORS ON IMAGE RECOVERY

During the experiment, various types of errors were superimposed on the original transmission vector: "NOT" error, knocked out bits and batch error. For the experiment, a set of images was collected, which can be conditionally divided into 3 categories: "forest", "land and water surface" and "urban area" (figure 3). The images have the same size 256×256, the brightness of the pixels varies from 10 to 175.



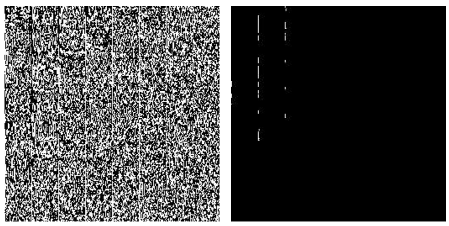
a) b)

c)

Fig. 3. Test Images: a) category "Forest", b) category "Land and water surface", c) category "Urban area"

In the Matlab environment, the bit representation of a number has an inverse numbering of digits: from the sign digit (7th bit) - the 1st plane to the lowest digit (0th bit) – the 8th plane [12,13.].

The formed initial bit planes for the "Forest" image are shown in figure 4.



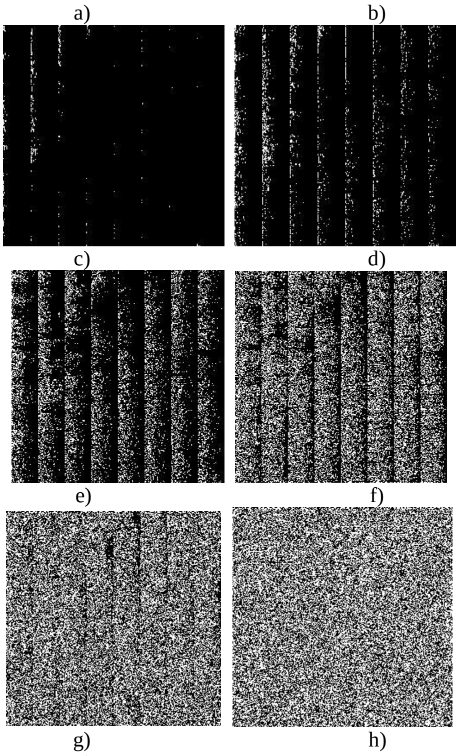


Fig. 4. Bit planes of the "Forest" image: a) 1st plane, b) 2nd plane, c) 3rd plane, d) 4th plane, e) 5th plane, e) 6th plane, g) 7th plane, h) 8tha th plane

To study the effect of errors, depending on the hit in different parts of the vector, 256-bit ranges corresponding to individual bit planes were distorted, that is, during the experiment, the error was superimposed separately on sections from the sign to the lowest digit [14]. Figure 5 shows 3 types of errors on the 2nd bit plane. The "knocked out bits" error has a minimal effect on the distortion of the plane, since the decimal value of the spatial-spectral representation for the image brightness matrix in most cases does not exceed 64, which in the bit representation gives the value "0" in the plane under consideration.

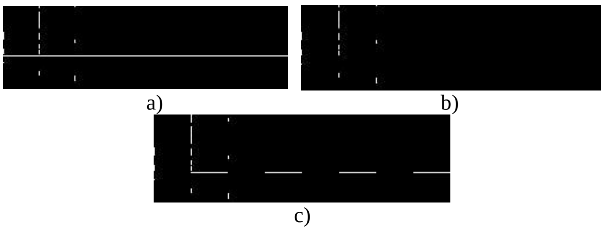


Fig. 5. Types of errors in image 1: a) error "NOT", b) knocked out bits, c) batch error

When errors appeared in one of the planes, their influence on the resulting image was analyzed, in which the existing distortions were not corrected: table 1 shows the values of the mean square deviation (MSD) for 3 categories of images and 3 types of error. The value of the MSD was calculated only for the part of the image in which the error was located [15].

TABLE I. MSD FOR AN IMAGE WITH AN ERROR (WITHOUT CORRECTION)

Image	Type of error	1st bit.pl	2nd bit.pl	3rd bit.pl	4th bit.pl	5th bit.pl	6th bit.pl	7th bit.pl	8th bit.pl
Fig. 2 a)	NOT	4,08	10,6	5,92	3,52	1,99	1,13	0,73	0,59
	KOB	2,25	0,53	1,22	0,94	0,81	0,77	0,62	0,55
	BE	3,02	7,68	4,34	2,53	1,46	0,87	0,63	0,57
Fig. 2 б)	NOT	5,57	12	6,51	3,81	2,27	1,46	1,18	1,09
	KOB	4,2	1,44	1,66	1,52	1,25	1,17	1,1	1,07
	BE	3,73	8,19	4,52	2,68	1,8	1,29	1,12	1,08
Fig. 2 c)	NOT	6,87	11,8	7,16	3,98	2,05	1,12	0,72	0,59
	KOB	4,8	0,53	1,91	1,55	1,25	0,81	0,62	0,55
	BE	4,63	8,66	5,24	2,86	1,53	0,89	0,64	0,56

where "NOT" is the "NOT" error, KOB is the knocked–out bits, BE is a batch error.

Based on the MSD values presented in table 1, it can be concluded that the knocked-out bits do not greatly affect the restored image, since the original values of the spectrum in binary representation have the value "0" in the higher digits. The error, which is a reset of values to "0", gives a small decrease in the informativeness of the data when it falls into the described range of the vector.

Any type of error that falls on the lower bits (0-2 bit bits) leaves a small distortion that can be considered acceptable. However, at the same time, the result of an error hitting the higher digits has a fairly large impact on the information content of the output image. An example of a "Forest" image with a batch error is shown in figure 6.

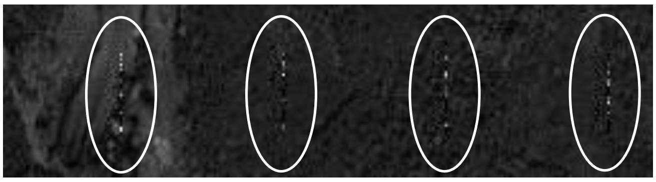


Fig. 6. The "Forest" image with a batch error in the 2nd plane

VI. DESCRIPTION OF THE ERROR CORRECTION PROCESS

After conducting experiments on the imposition of various types of errors, the task was set to correct the received vector in order to reduce the amount of distortion of the transmission vector to improve the quality of the information content of the restored image. To do this, an algorithm has been developed, which consists of the following steps:

1) Transformation of a vector to a matrix form in accordance with bit planes [16].

2) Calculation of the average value in decimal representation by adjacent rows of values (above and below) relative to the location of the error bit.

3) Calculation of the decimal value of the spectrum in which the error occurred.

4) Analysis of the obtained values for whether the value with an error falls into the trend of the values of its neighbors.

5) Correction of a bit in the plane to the opposite, if the value of the spectrum with an error does not fit into the trend set by its neighbors.

As a calculation of entering into the trend, the ratio was compiled (12). Figure 7 shows a graphical representation of the position of the pixels of the image from which the calculation is performed.

$$\frac{f_{(i,j)}}{\frac{(f_{(i-1,j)}+f_{(i+1,j)})}{2}}, \qquad (12)$$

where f is the value of the spectrum in decimal representation, (i,j) are the indices of finding an element in the spectrum matrix.

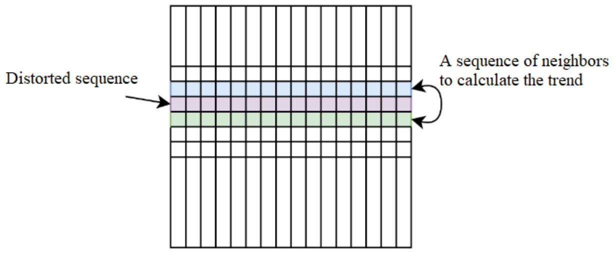


Fig. 7. Graphical representation of the expression (12)

As a result of the conducted research, a number of questions were raised for spatial-spectral matrices: which ranges of values for the ratio are acceptable in order to make a decision about the entry of the desired value into the trend, and what should this ratio strive for. The more accurate the hit in the trend, the closer the value of the spectrum is to the average value of its neighbors, that is, to the value of 1. However, during the experiment, the following problematic situations were identified:

1) Table 2 shows the corresponding initial values of the spectrum and their binary unsigned representation.

TABLE II. EXAMPLE 1

Position	Spectrum values	Binary representation of the spectrum
(i-1, j)	7_{10}	0000111_2
(i, j)	21_{10}	0010101_2
(i+1, j)	19_{10}	0010011_2

The ratio value is $\frac{21}{13} \approx 1{,}6$, which means that the real values of the spectrum of the original image can give a sufficiently large deviation from the average value of its neighbors. Accordingly, it is impossible to compare unambiguously with the average value, so it is necessary to check whether the value with the changed bit would be better to enter the trend: $1010101_2=85_{10}$, the ratio $\frac{85}{13} \approx 6{,}5$, which is significantly higher than the original value.

Thus, falling into a trend cannot be judged by a constant value, but should be compared by 2 ratios: the initial value to the average value of its neighbors, the changed value to the average value of its neighbors.

2) Since the values of the spectrum can be both positive and negative, neighbors can have the same value of the opposite sign (for example, 2 and -2), then their average value will be 0.

After analyzing the matrices of the spatial-spectral representation, 2 conditions were revealed to determine the trend: a comparison of the relations for the original accepted sequence and the modified one; selecting the value that is closer to 1.

VII. RESULTS OF THE EXPERIMENTAL PART

In the experiment, the MSD ratio between the input and output images, that is, the brightness values, was calculated. However, during the conversion of the original brightness matrix into a spatial-spectral representation and the restoration from the spectral form to the brightness values, rounding operations are performed, which impose additional distortions. Table 3 shows the MSD values for the spectrum value matrices: initial and after error correction.

TABLE III. THE VALUE OF MSD FOR THE SPECTRUM

Image	Type of error	1st bit.pl	2nd bit.pl	3rd bit.pl	4th bit.pl	5th bit.pl	6th bit.pl	7th bit.pl	8th bit.pl
Fig. 2 a)	NOT	0,2709	0	0,6963	0,4606	0,3371	0,2787	0,1584	0,08
	KOB	0,2709	0	0,6963	0,4606	0,3371	0,2787	0,1584	0,08
	BE	0,2709	0	0,6963	0,4606	0,3371	0,2787	0,1584	0,08
Fig. 2 б)	NOT	0,7554	0,9545	1,074	0,8539	0,7398	0,699	0,6746	0,6578
	KOB	0,7554	1,3714	1,1815	0,8887	0,75	0,7017	0,6753	0,658
	BE	0,7554	0,9545	1,074	0,8539	0,7398	0,699	0,6746	0,6579
Fig. 2 c)	NOT	0,305	0,6963	0,9211	0,8349	0,5708	0,3168	0,1846	0,0877
	KOB	0,305	0	0,8528	0,8165	0,5641	0,3138	0,1846	0,0877
	BE	0,305	0,6963	0,9211	0,8349	0,5708	0,3168	0,1834	0,087

where "NOT" is the "NOT" error, KOB is the knocked–out bits, BE is a batch error.

The results obtained show that the analysis of the studied value for falling into the trend of the values of its neighbors gives a good result in correcting errors. When correcting errors on the lower digits of the numbers corresponding to the seventh and eighth bit planes, the MSD systems are almost identical with the results without restoration, which once again confirms the statement about the small effect of distortion on the original image.

To restore the distorted spectral components, only 3 values are required, which makes it possible to efficiently divide the data into nodes for parallel calculation of the algorithm described in the article. Table 4 contains the time values for post-processing images on a personal computer with an intel Core i5 CPU and an Nvidia GeForce RTX 3060 GPU.

TABLE IV. RUN-TIME VALUES

Size of the original data set	Sequential calculation time (sec)	Parallel computing time on CPU (sec)	Parallel computing time on GPU (sec)
220 KB	2.79	1.62	0.26
560 KB	5.23	2.72	0.49
1280 KB	14.52	7.68	1.36

CONCLUSION

Restoration by analyzing the input of the spectrum values into the trend of neighboring values in most cases leads the resulting vector to a form whose deviation from the original is quite small. Depending on the characteristics of the image: the range of brightness, sharp boundaries, values of the spatial-spectral representation may arise, complicating the decision to enter the trend, which was taken into account when implementing the approach described in the article. The use of parallel computing has made it possible to increase the efficiency of processing received data, which is especially important in real-time systems.

As a result of the experiment, an image transmission scheme with a noisy communication channel was constructed, where the processes of preprocessing and restoring the original image were performed on the basis of parallel calculations. The main task was to find a solution to improve the performance of program components by distributing data across individual cores. Testing applications of parallel computing on the CPU and GPU on different sets of images shows a sufficiently large acceleration for systems working with landscape images.

[1] Zolotarev V. V., Ovechkin G. Z-80 Noiseproof Coding. Methods and Algorithms: Handbook / Ed. by Yu. B. Zubarev. - Moscow: Hot Line-Telecom, 2004. - 126 pp.

[2] Svetlov G.V., Sumenkov N.A., Kostrov B.V., Fokina N.S. Application of the theory of discrete signals defined on finite intervals for aerospace image processing. Vestnik VKO Almaz-Antey Concern. 2017;(3):94-102. https://doi.org/10.38013/2542-0542-2017-3-94-102.

[3] Bastrychkin A.S., Kostrov B.V., Grinchenko N.N., Svetlov G.V. Calculation of image transmission partial spectrum // 9TH MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING, MECO 2020 Budva, 08–11 june 2020

[4] Kostrov B.V., Svirina A.G., Zlobin V.K. Spectral analysis of images in finite bases [Monograph] - KURS Publishing House (Moscow), - 2016, 172 p.

[5] Ahmed N., Rao K. R. Orthogonal transforms in digital signal processing / edited by I. B. Fomenko. M.: Svyaz, 1980. 248 p.

[6] Kostrov, B. V., Bastrychkin, A. S., Kostrov B. A., Stepanov D. S. Protocol of image transmission over communication channel // Intelligent and Information Systems: Materials of All-Russian Scientific and Technical Conference. Tula State University. Tula, 2016. pp. 220-225.

[7] Grinchenko N. N., Baranova S. N., Lobachev M. A., Vyugina A. A. Mathematical model of the image transmission process based on bit planes // Vestnik Koncerna VKO “Almaz – Antey”. 2023. No. 1. P. 82–89. https://doi.org/10.38013/2542-0542-2023-1-82-89

[8] Bastrychkin, A. S., Kostrov, B. V., Grinchenko N.N., Kostrov, B. A. Traffic shaping in data transmission systems // Modern technology in science and education - STNO-2022. Proceedings of the V International Scientific and Technical Forum. Ryazan, 2022, p. 115-120.

[9] Kostrov B. V., Solomentseva N. I. Modeling the communication channel // Proceedings of TulSU. Technical Sciences 2017. Issue. 2. pp. 95-100.

[10] Olenev N.N., Pechenkin R.V., Chernetsov A.M. PARALLEL PROGRAMMING IN MATLAB AND ITS APPLICATIONS Moscow: VC RAS, 2007. 120 p.

[11] Information Theory: Textbook / B.V. Kostrov; Ryazan State Radio Engineering University. Ryazan, 2006. 124 p.

[12] Amos Gilat MATLAB: Theory and Practice. 5-th edition / Translated from English Smolentsev N.K. - M.: DMK Press, 2016. – 416p.

[13] Potemkin V. G. Calculations in the MATLAB environment. Moscow: Dialog-MIFI, 2004. 720 p.

[14] Vyugina A.A., Grinchenko N.N., Baranova S.N. The study of bit traffic in the transfer of landscape images // Modern technology in science and education - STNO-2022. Proceedings of the V International Scientific and Technical Forum. Ryazan, 2022, p. 111-115.

[15] Shvedov A. S. Probability Theory and Mathematical Statistics. M: GU-VSHE, 2005. 254 p

[16] Gonzalez R., Woods R. Digital Image Processing. Edition 4th. Moscow: Technosphere, 2018. - 1184 p.