IMAGE PROCESSING, PATTERN RECOGNITION

HYPERSPECTRAL REMOTE SENSING DATA COMPRESSION AND PROTECTION

M.V. Gashnikov [1], *N.I. Glumov* [1,2], *A.V. Kuznetsov* [1,2], *V.A. Mitekin* [1,2], *V.V. Myasnikov* [1,2], *V.V. Sergeev* [1,2]

[1] *Samara National Research University, Samara, Russia,*

[2] *Image Processing Systems Institute of RAS, – Branch of the FSRC "Crystallography and Photonics" RAS, Samara, Russia*

Abstract

In this paper, we consider methods for hyperspectral image processing, required in systems of image formation, storage, and transmission and aimed at solving problems of data compression and protection. A modification of the digital image compression method based on a hierarchical grid interpolation is proposed. Methods of active (on the basis of digital watermarking) and passive (on the basis of artificial image distortion detection) data protection against unauthorized dissemination are developed and investigated.

Keywords: Digital image processing, Image analysis, hyperspectral images, data compression, hierarchical grid interpolation method, digital watermarks

Citation: Gashnikov MV, Glumov NI, Kuznetsov AV, Mitekin VA, Myasnikov VV, Sergeev VV. Hyperspectral remote sensing data compression and protection. Computer Optics 2016; 41(5): 689-712. – DOI: 10.18287/2412-6179-2016-40-5-689-712.

Acknowledgements: The work was funded by the Russian Science Foundation, grant No. 14 31-00014.

Introduction

Hyperspectral Earth remote sensing (ERS) systems [1 – 3] are a natural evolution of means for obtaining digital images of the Earth surface. An increase in the number of spectral channels of optical sensors from several units to hundreds opens up prospects for a qualitatively new knowledge of the surface, and allows to solve a lot of new applied problems [4 – 11] of geology, wildlife management, ecology, etc. At the same time, hyperspectral remote sensing data hyperspectral images (HSI), are very specific "three-dimensional" informational objects, which require the development of special methods of their production, transmission, processing, and storage.

Firstly, the extremely large amount of hyperspectral data entails extremely high demands for the capacity of storage devices and communication channels. Under such conditions, it is necessary to use data compression [12 – 18].

Secondly, by analyzing remote sensing data, decisions at the municipal, regional and national levels are made. Due to the widespread use of such data (in particular, HSI), the task of potential falsification detection, and protection against unauthorized data dissemination, becomes more urgent (as a way to reduce the risks in making such decisions).

Obviously, the compression procedures, generating complex structured data sets, can be considered as a means to cryptographic protection of HSI. However, such protection stops after decompression, i.e. it is insufficient against many threats associated with unauthorized copying, distribution, or data modification. Consequently, we need special protection methods, which are separate from compression. Today, there are [19] two basic approaches to digital image (and, remote sensing data, in particular) protection and artificial distortion detection: active and passive. The basic element of active approach to the detection of artificial image distortion is the use of digital watermarks, which are embedded into image. Unlike active, the passive approach does not use watermarks, and is based on the assumption that modification traces can be detected by using image computer analysis. The simultaneous use of two approaches can provide reliable data protection against unauthorized copying, modification and dissemination, and ultimately enhance the information security of systems related to the production, storage, processing and analysis of visual information.

It should be noted that in real systems for production and processing of remote sensing data, the tasks of compression and image protection are usually solved simultaneously, and in many cases they are close in terms of information technology. That is why we consider them within one article.

1. Remote Sensing Data Compression

Studies in the area of hyperspectral remote sensing data compression are conducted by many researchers [20 – 22]. The Multispectral and Hyperspectral Data Compression (MHDC) working group [23] of the Consultative Committee CCSDS [24] should be noted separately. This working group has developed the CCSDS-123.0-B-1 standard [20], which is based on the fast lossless compression algorithm, and intended for onboard lossless coding of multi- and hyperspectral imagery.

However, these algorithms do not meet all the requirements [10 – 11] for hyperspectral remote sensing data compression method. In particular, these papers do not consider a comprehensive approach to compression, which takes into account the speed stabilization of compressed data formation, and protection against failures in the communication channel .

According to requirements [10 – 11], a method based on a hierarchical grid [25 – 28] interpolation (HGI) was chosen for hyperspectral image compression.

HGI method is based on non-redundant hierarchical representation of the original image $\mathbf{X}=\{x(m,n)\}$ as a union of a hierarchical (scale) levels $\mathbf{X}_l$:

$$\mathbf{X}=\bigcup_{l=0}^{L-1}\mathbf{X}_l\,,\quad \mathbf{X}_{L-1}=\left\{\,x_{L-1}(m,n)\right\}, \tag{1}$$

$$\mathbf{X}_l=\left\{\,x_l(m,n)\right\}\setminus\left\{\,x_{l+1}(m,n)\right\},\ l<L-1\,, \tag{2}$$

where L – is the number of hierarchical levels (HL), $\{x_l(m,n)\}$ – is a set of image counts, obtained by taking 2^l step for each coordinate.

During the compression levels are processed sequentially from the senior level $\mathbf{X}_{L\text{-}1}$, and besides less detailed level counts are used for more detailed level count interpolation. Interpolation error (post interpolation residual) sets are quantized for each level, compressed by entropy encoder and placed in an archive or a communication channel.

1.1. General Description of The On-Board Video Processing Method

Compression method based on HGI meets the [27–28] compression ratio, quality control and algorithm complexity requirements to the onboard video processing methods. However, the method needs the further improvement to provide not only data compression problem solution, but also constant speed of the compressed data output stream generation and high noise immunity of output data.

The general scheme of the proposed on-board video [29] processing method is given in Fig. 1. Three separate blocks of the scheme describe the solution for the problems of compression, output data speed stabilization and protection of encoded information against communication channel failures.

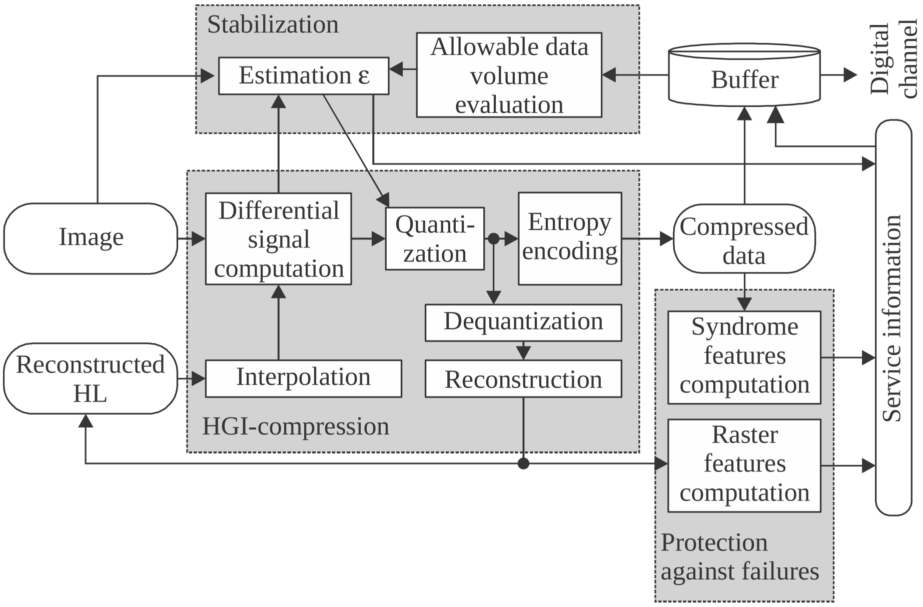

Fig. 1. A general scheme of onboard compression and hyperspectral remote sensing data protection

1.2. Speed Stabilization of Compressed Data Stream

A remote sensing system image is formed as a long fixed-width band. The image is divided into sub-bands (or blocks) which are compressed independently. After each compressed block is placed into the buffer memory, data can be transferred over a constant bandwidth communication channel. To obtain the output stream generation speed, which is close to constant, we need to determine the control parameter of compression for each block. For HGI-method we use maximum compression error as a control parameter

$$\varepsilon=\max_{m,n,s}\left|x^s(m,n)-\overline{x}^s(m,n)\right|\ . \tag{3}$$

where $x^s(m,n)$, $\overline{x}^s(m,n)$ are the s spectral component counts of original and decompressed images respectively.

Let the encoded image be an infinite vertical band of width equal to M counts. Such image format makes it expedient to implement a block-based compression. However encoding of different blocks of size $Z\times M$ counts (Z is the number of rows in a block) can be performed with various control parameters, which leads to output quality indicator (compression ratio and reconstruction error) deviation from the requirements. Thus, the encoded block buffering should simultaneously eliminate the error fluctuations and provide speed stabilization of transmitting the encoded data to the communication channel.

We built a mathematical model, describing a discrete time process of buffering the video, processed block by block. Let the buffer memory has a capacity of V_0 bit, and it is $V(K-1)$ bit full to the time the processing of the K^{th} block begins. Let b^0 is a count width of uncompressed image (bit/count). During a time slice an image block of size $\Delta V=ZMb^0$ bit is generated at a constant speed and encoded with a predefined maximum error $\varepsilon(K)$, which provides the practicable compression of block $B(K)$ (bit/count). At the same time $B_0\Delta V/b^0$ bits of information are transmitted to a communication channel, where B_0 is a compression ratio, corresponding to communication channel bandwidth. By the next time slice the relative buffer occupancy rate will be:

$$\begin{aligned}&\bar{V}(K)=V(K)/V_0=\\&=\frac{1}{V_0}\left(V(K-1)+\frac{B(K)\Delta V}{b^0}-\frac{B_0\Delta V}{b^0}\right)=\\&=\bar{V}(K-1)+(B(K)-B_0)/k_v b^0,\end{aligned}\tag{4}$$

where $k_v = V_0/\Delta V$ is a buffer parameter (the ratio of buffer capacity to the block capacity before compression).

The main problem is to determine the error ε(K) during the compression of each control parameter block. The predefined error should provide such a degree of compression which can prevent buffer overflow $\bar{V}(K)<1$. Next we consider two approaches to determining ε(K).

Algorithm I. At each step the error is adjusted depending on buffer occupancy rate by the time the K^{th} block processing begins, e.g.

$$\varepsilon(K)=\varepsilon(K-1)+\left[\left(\bar{V}(K-1)-\bar{V}_p\right)a\right],\tag{5}$$

where $0<\bar{V}_p<1$ is a threshold, exceeding of which increases the error and reduces the compression ratio; a is a parameter adjusted experimentally.

Algorithm II. A predefined maximum error of Kth block compression is determined by calculation of image block statistical characteristics (e.g. a dispersion D_x and a correlation coefficient ρ) and the required compression ratio $\hat{B}(K)$:

$$\varepsilon(K)=f\left(D_x,\rho,\hat{B}(K)\right),\tag{6}$$

where determining $\hat{B}(K)$ is based on the availability of free memory in the buffer:

$$\hat{B}(K)=B_0+\left(\bar{V}_p-\bar{V}(K-1)\right)b^0(k_v/k_s),\tag{7}$$

$\bar{V}_p\approx 1$ is a threshold, providing a "safety margin" for the buffer occupancy, $k_s \geq 1$ is a stabilization coefficient, providing a more smooth change of all parameters when encoding the image block by block.

Below we propose a specific method for calculating the maximum error ε_{max} via the required compression ratio $\hat{B}$, based on the assumption of an isotropic exponential model of autocorrelation function of an input signal.

Assuming that entropy encoding method is quite effective, we use value $\hat{H}_q$, which is an entropy of the quantized differential signal, predicted on the basis of image correlation coefficient ρ, image dispersion D_x and error ε_{max}, as a compression ratio estimation $\hat{B}$. Relying on the ratio of the number of counts (1 - 2) at separate hierarchical levels (HL), we can get the following expression:

$$\hat{H}_q=\left(\hat{H}^{(R)}+3\sum_{r=1}^{R}4^{r-1}\hat{H}^{(R-r)}\right)/4^R,$$

where $\hat{H}^{(r)}$ is a predicted value of the quantized differential signal entropy at the rth HL. According to the definition of entropy:

$$\hat{H}^{(r)}=-\sum_{i=0}^{N_q-1}p_q^{(r)}(i)\log\left(p_q^{(r)}(i)\right),$$

where $p_q^{(r)}(i)$ is a probability of the ith quantization level of differential signal, N_q is the number of quantization levels.

In accordance with the recommendations [29] we assume that the probability distribution of unquantized differential signal at each HL is exponential. Therefore, using the quantizer with a uniform scale we can determine the probability of the ith quantization level of differential signal as follows:

$$p_q^{(r)}(i)=\sum_{k=i\cdot(2\varepsilon_{max}+1)-\varepsilon_{max}}^{i\cdot(2\varepsilon_{max}+1)+\varepsilon_{max}}\left((1-p_r)/(1+p_r)\right)p_r^{|i|},$$

where p_r is a parameter of the above-mentioned exponential distribution, related to dispersion $D^{(r)}$ of differential signal at the rth HL as follows:

$$p_r=\left(2\cdot D^{(r)}+1-\sqrt{2\cdot D^{(r)}+1}\right)/D^{(r)}.$$

Besides, we can prove that:

$$D^{(r)}=a_1\cdot 2^r\ln(1/\rho)D_x+a_2\,\varepsilon_{max}^2,$$

where a_1, a_2 are coefficients determined via the prediction scheme.

Thus, we obtain the relation $\hat{H}_q=\hat{H}_q(\rho,D_x,\varepsilon_{max})$, which can be pre-tabulated. During encoding the minimum value ε_{max}, which satisfies the condition:

$$\hat{H}_q(\rho,D_x,\varepsilon_{max})<\hat{B},\tag{8}$$

is used as a required error value.

Thus, the proposed scheme of stabilization by defining the compression algorithm control parameter value (the maximum error) ε(K) during the Kth time slice requires:

- calculation of statistical characteristics $D_x(K)$, ρ(K) for the Kth image block;
- determination of a required compression ratio $\hat{B}(K)$ according to (7);
- selection (from the table) of ε(K) value, which satisfies the condition (8).

To investigate the considered algorithms we conducted a simulation of image compression process with speed stabilization of output encoded video stream using a large-format aerospace image of the Earth's surface. An example of such an image (of size 10000×512 counts), reflecting the dynamics of change in image local information content along a survey route is shown in Fig. 2. Measured by blocks of size 33×512, statistical characteristics ρ and $\sigma=\sqrt{D_x}$ are shown in Fig. 3.

In order to find the optimal speed stabilization algorithm for image compression we conducted an experimental research on the above-mentioned methods of generating a control parameter (maximum reconstruction error) for a compression algorithm. During the study we controlled the buffer memory occupancy according to a mathematical model (4). We used an average (over blocks) maximum reconstruction error E{ε(*K*)} as a quality criteria. All experiments were performed for the most typical required compression ratios: $B_0 = 1$ bit/count, 2 bit/count.

Algorithm I. Of particular interest is the control parameter generation depending on the relative buffer occupancy according to (5). The experimental research (by varying the coefficient within $2{,}5 \le a \le 20$, and buffer parameter $k_v = 2$, $\bar{V}_p = 0{,}5$) showed that, although the compression parameter is adaptively adjustable depending on the buffer state, in general, this method can be characterized as insufficiently stable one. The main reason, why the method cannot be recommended for actual use, is the pulsing character of dependences $\bar{V}(K)$, ε(*K*) (see Fig. 4), which can lead to unacceptable quality of output video, rapidly and repeatedly changing along one survey route.

Fig. 2. Test image "Survey Route" for compressed data speed stabilization algorithms

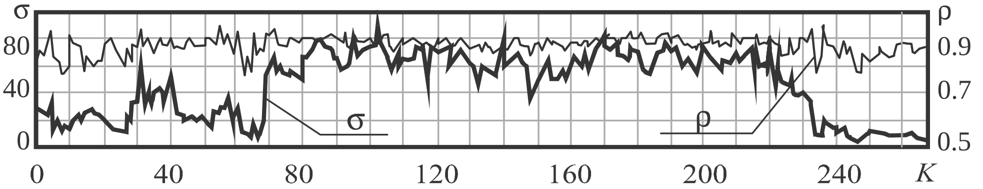

Fig. 3. The statistical characteristics of the test image "Survey Route"

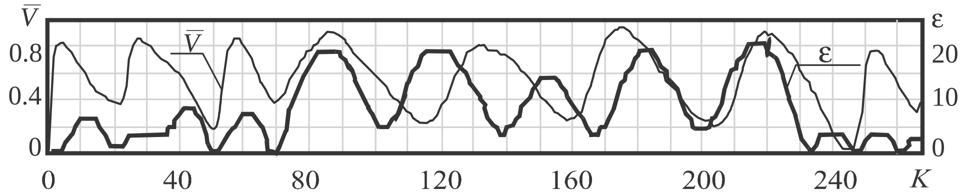

Fig 4. The results of simulation of the stabilization algorithm I (B_0 = 2, a = 10) for "Survey Route" image

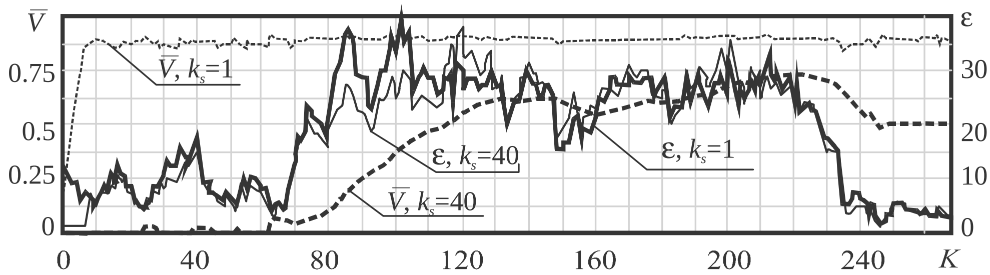

Fig. 5. The results of simulation of the stabilization algorithm II (B_0 = 1) for "Survey Route" image

Algorithm II. The most applicable method of predefined error generation appears to be based on the encoded block statistical characteristic analysis.

The dependence parameters (6) calculated for real hyperspectral images are shown in paper [30]:

$$\varepsilon_{\max}(t) = -96{,}4 + 7{,}71 D(t)^{-0{,}62} + \\ + 75{,}14 \rho(t)^{0{,}01} - 3{,}61 \hat{B}(t)^{6{,}5}.$$

Thus, the standard deviation of the calculated value $\varepsilon_{max}(t)$ from the predefined value $\hat{\varepsilon}_{\max}(t)$ was 0.66 for each block, which demonstrates the high approximation accuracy. A histogram of error deviation distribution $\Delta\varepsilon = \hat{\varepsilon}_{\max}(t) - \varepsilon_{\max}(t)$, demonstrating that in 98.0 % of cases the error deviation is $|\Delta\varepsilon| \le 1$, is shown in Fig. 6.

The deviation of the resulting compression ratio *B*(*t*) from the required one $\hat{B}(t)$ was also evaluated in [30]. The standard deviation was 0.28 bit/count. It has been also showed, that, in 95.1% of cases the compression ratio deviation was $|\Delta B| \le 0.2$ bit/count.

In general, the results of speed stabilization method simulation show sufficiently high effectiveness and sustainability. When simulating according to formulae (4), (7), attention should be paid to stabilization coefficient k_s, which allows to mitigate the effects of passing the highly informative image regions and get better quality indicators (see Fig. 5, 7*a*). The effect of buffer parameter k_v (for $Z = 33$ and optimal k_s) and the number of rows in a block *Z* (for $k_v = 1$) on the reconstruction quality is shown in Fig. 7*b*, 7*c*. An increase in the value k_v, corresponding to an increase in the memory size, results in better quality (see Fig. 7*b*). However, to increase the size of the encoded blocks, it is more appropriate to use the memory enhancement, if it is possible (see Fig. 7*c*).

The results of computational experiments do not make the final recommendations on the selection of specific stabilization method parameter values, because we conduct the experiments only under the following conditions:

- precisely specifying the limitations due to technical resources;

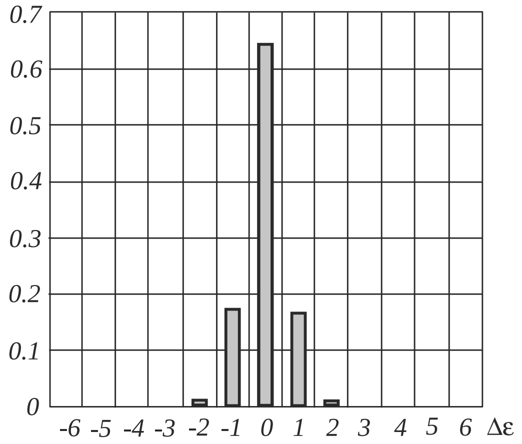

Fig. 6. A histogram of error deviation distribution

$$\Delta\varepsilon = \hat{\varepsilon}_{\max}(t) - \varepsilon_{\max}(t)$$

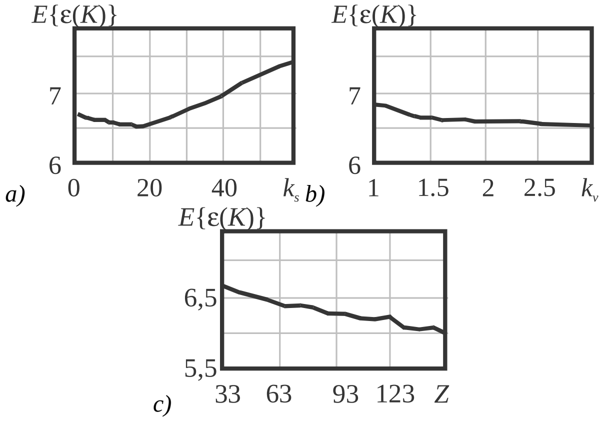

Fig. 7. Influence of stabilization parameters on error

- solving the time-consuming optimization problem in a multidimensional parameter space $\left(k_s, k_v, Z, \overline{V}_P\right)$ using a large set of real plots.

However, the experimental results, outlined above, allow us to compare different stabilization algorithms and outline the ways to achieve the best quality indicators.

Thus, the results of the conducted simulation show, that for a sufficiently precise prediction of error $\varepsilon(K)$, which provides a required compression ratio in the current block and a stable speed of compressed video formation, we cannot rely only on the current state of the buffer and the compression parameter values, used in the previous steps. So the statistical properties of an image block, encoded at this moment, should be considered. Only in this case the data compression algorithm will be easily adapt to even the most adverse (in terms of feasible compression ratio under acceptable quality) situation along a survey route without exceeding the limitations, imposed by the technical resources.

1.3. Compressed Data Noise Immunity Enhancement

In systems of digital image generation and transmission over communication channels the most important information transformation algorithms are as follows. First, to reduce the amount of information, transmitted over a communication channel with limited bandwidth, a compression algorithm is used. Secondly, in order to protect the transmitted data from communication channel noise the noiseless coding algorithm is used.

Since these algorithms are implemented in different subsystems of digital image generation and transmission systems, the algorithm development is usually carried independently, without consideration of mutual specifics. On the one hand, when choosing a compression algorithm we do not take into account the failure tolerance of the compressed data. On the other hand, all of the known noiseless coding algorithms (NCA) are versatile with respect to the original data, and such algorithms do not use the limitations on possible combination set of the compressed data array of known length. Besides, we do not consider the probabilistic nature of uncorrected failure effects, which can lead to both minor distortions of decompressed data (compared with distortions, introduced by compressing), and to significant distortions, up to a total loss of a picture.

This section examines a comprehensive approach to selection of compression and noiseless coding algorithms for implementation in systems of digital image generation and transmission. Here we assume the following model of formation and transformation of the information, transmitted to the communication channel. Suppose that an image is formed by the progressive scanning. Image rows are combined into frames of size $N \times M$ pixels. Frames are compressed independently with a predetermined compression ratio B_c. Next, the compressed information is divided into blocks of length k bit, each of which is transformed by NCA (the block size increases to size $n > k$) and transmitted to the communication channel. We assume that, when transferring data, failures (inversions) occur independently, and we know the probability p_0 of one bit failure.

Let S_i be an event, consisting in appearance of i failures in array of compressed data. By considering only $\left\{S_i\right\}_{i=1}^{I}$ events while simulation, it is possible to build a distribution histogram of the output quality indicator and obtain estimates of conditional probabilities $P(Q \le Q_{fr} / S_i)$ that the quality (error) Q of the decompressed image is limited by the value Q_{fr}.

Finally, the quality of output image is estimated using a cumulative histogram

$$F\left(Q_{fr}\right) = \sum_{i=0}^{I} P\left(Q \le Q_{fr} / S_i\right) P\left(S_i\right).$$

When comparing different algorithms it is more convenient to use integral estimates of probability. After specifying the threshold values Q_1, Q_2, we introduce the following classification of the image distortion: "undistorted" ($Q = 0$), "slightly distorted" ($0 < Q \le Q_1$), "considerably distorted" ($Q_1 < Q \le Q_2$), "unusable" ($Q > Q_2$). Let Ω_0, Ω_1, Ω_2, Ω_3 be the events, consisting in image assignment to one of the classes listed above, respectively.

In [31] a method of improving the noise immunity of compressed images is proposed. According to this method the information is divided into two parts: raster information and service information, including an image header, algorithm parameters, etc. The share of service information in a frame is less than 1 %, but it has a special value for the image recovery – its impairment almost always results in image loss. Therefore, to protect the service information we should use effective NCA (e.g., Bose–Chaudhuri–Hocquenghem (BCH) code, Reed–Solomon code) [32], which provides the required data transmission reliability.

When the service information recovery is guaranteed, the protection requirements of raster data can be reduced on the condition, that the distortions, occurred as a result of unfixed failures, are insignificant, compared with distortions, introduced by compression. Thus, the noiseless code redundancy will be reduced, the compression ratio B_c will be improved, and, finally, the quality of the output image will be increased.

A hierarchical structure of compressed data provides additional opportunities to enhance the noise immunity. Thus, according to the proposed HGI-method, compressed data are formed consecutively by image hierarchical levels, starting with the compression of 2^{L-1} thinned image (where L is a number of hierarchical levels). Since, when moving to the next level, the amount of data increases approximately 4 times, the probability of failures at the next level also increases. However, even when it is impossible to fix failures at a low level, the image is not lost completely, it can be obtained by interpolating the reconstructed senior levels.

In our research we examined three digital image compression methods – JPEG; compression, based on wavelet transform, and HGI-method, modified to increase the noise immunity of compressed data. The HGI-method modification consists in introducing control features in service information at each hierarchical level (raster features computation in Fig. 1), and in distinguishing and encoding the service information, using BCH-code (syndrome features computation in Fig. 1).

The control features calculation while image decompression and comparing them with true values allows to detect and fix S_1, S_2-type failures. Fig. 8 shows the dependence of the image hierarchical level loss probability on the one bit failure probability (for $B_cNM = 2^{15}$ bit). The probability of the errorless transmission of the entire image is $1 - P_{loss}(0)$. It is easy to see that the probability of senior level loss is negligible.

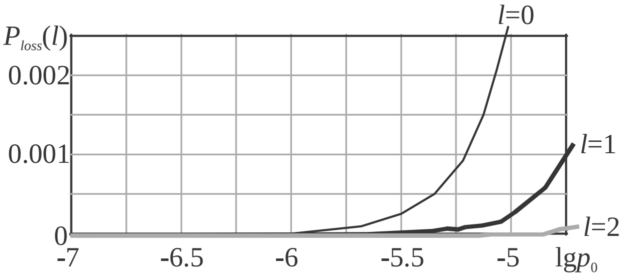

Fig. 8. Estimation of image hierarchical level loss probability for HGI-method

The results of the research are presented in Fig. 9, 10 and demonstrate the impact of failures on the image compression for different compression methods. It should be noted that we didn't use noiseless coding (except service information encoding in HGI-method). To compare the noise immunity of compressed data, the cumulative histograms of quality indicator (standard deviation ε_{mse}) and the dependence of probability, that decompressed image belongs to classes Ω_0 and Ω_3, on failure probability p_0, are given.

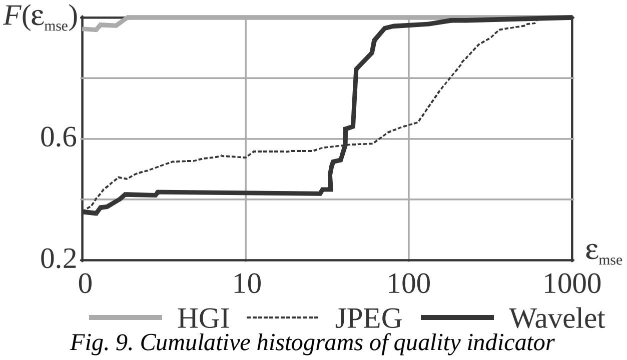

Fig. 9. Cumulative histograms of quality indicator (standard deviation)

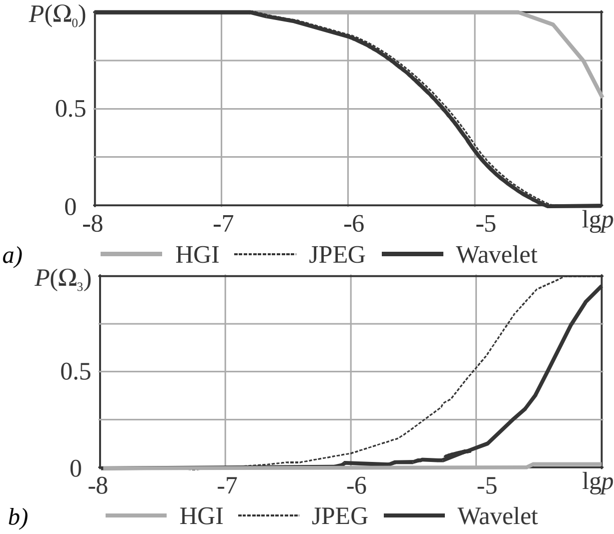

Fig. 10. Dependence of output data quality estimation on the probability of single bit failure: a) class Ω_0; b) class Ω_3;

The results demonstrate that HGI-method has a significantly superior noise immunity. Thus, with a probability of 0.986, the effects of HGI-method data compression failures will be completely eliminated (for $p_0 = 10^{-6}$), and, with a probability of 0.997, reconstruction error will not exceed 1, which corresponds to the class Ω_1. Studies have shown, that the failure effect on the reconstructed image quality can be neglected over the entire range $p_0 < 10^{-5}$.

For the known methods, like JPEG and method, based on wavelet transform, the problem of compressed data noise immunity is very urgent. With a probability of at least 0.5, the quality of reconstructed images will not be satisfactory, which corresponds to classes Ω_2 and Ω_3. To protect images, compressed by these methods, up to the level of HGI-method, it is necessary to introduce a redundancy of at least 10 % by using noiseless coding. However, if the communication channel bandwidth is fixed, the use of NCA requires the higher compression, which inevitably entails the deterioration of output data quality.

Thus, we propose a comprehensive approach to selecting algorithms of compression and noiseless coding for transmitting the compressed images over a communication channel. The effectiveness of such an approach is particularly significant when used for HGI-method compression, which allows to evaluate and considerably improve the performance of systems of digital image generation and transmission over communication channels during the design stage.

1.4. Compression During Storage of Hyperspectral Images

Analysis of hyperspectral image features [27 – 28] made the HGI-method a reasonable solution for hyperspectral image storage problem. A direction, connected with consideration of correlation between components, which is extremely high for hyperspectral images, was chosen as basic direction of method improvement (see Fig. 11).

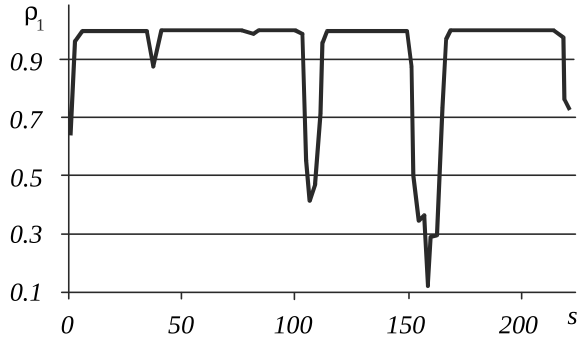

Fig. 11. Estimation of correlation coefficient ρ1 *between neighboring components (components number s and number s* + 1*) depending on the number of component s for "Low Altitude" image made by spectrometer AVIRIS*

"Sliding Component Approximation". The relationship between spectral components is used due to a component prediction [10 – 11]. Actually, the prediction is proposed to be implemented as component approximation, based on other components, which have already passed compression and recovery. A high correlation between components should provide a good prediction accuracy, and compression of the difference between the original and the predicted spectral component, instead of compressing the original spectral component, should significantly increase the compression ratio. The formal description of the algorithm is given below.

Let $\{X^s, 0 \le s < S\}$ be an S-component hyperspectral image, consisting of two-dimensional spectral components – one-component images X^s. These components are compressed successively, from smaller numbers to greater. During the compression of each component X^s we first compute its predicted (approximating) value:

$$\hat{X}^s = \sum_{i=0}^{N-1} k_i \overline{X}^{s-i-1}, \; 0 \le s < S \; ,$$

where $\overline{X}^i, i \ge 0$ are the previous components, which have already passed the compression and recovery, N is a number of previous reconstructed spectral components, used for the approximation (an algorithm parameter), $\{k_i, 0 \le i < N\}$ are the approximation coefficients, which are the solution of system of linear equations, according to Ordinary Least Squares:

$$\mathbf{Rk} = \mathbf{B} \; ,$$

where $\mathbf{k} = \{k_i, 0 \le i < N\}$ is a required vector of the approximation coefficients, $\mathbf{R} = \{R_{i,j}, 0 \le i, j < N\}$ is a the correlation coefficient matrix for the decompressed components $\overline{X}^{s-i-1}$ and $\overline{X}^{s-j-1}$, $\mathbf{B} = \{B_j, 0 \le i < N\}$ is a correlation coefficient vector for the currently predicted component X^s and the decompressed component $\overline{X}^{s-i-1}$.

Actually, using HGI-method, we compress (with a predefined maximum error) not the component X^s, but the difference between X^s and the predicted (approximating) component $\hat{X}^s$.

Thus, a set of support components for each currently predicted component is represented by a "sliding window", located in a spectral plane. That is why the compression algorithm described above is called "algorithm based on sliding component approximation".

"Independent Portions of Components". The compression algorithm based on "sliding component approximation" described in the previous section is not suitable for solving the problem of hyperspectral image storage. The reason is that for decompression of an arbitrary spectral component we have to decompress all previous components (there can be hundreds of them), which is a serious drawback for the organization of quick access to storable hyperspectral images.

Naturally, when storing hyperspectral images in database, we would like the decompression of an arbitrary spectral component to entail the decompression of the smallest possible number of components (probably unnecessary). To provide this opportunity an approach, based on component approximation within "independent portions of components" [10 – 11], is used.

During compression a set of spectral components is divided into independent "spectral portions" (N components each, see Fig. 12), and within each portion the above mentioned algorithm of "sliding" component approximation is used. Thus, to decompress an arbitrary component we do not need to decompress all previous components of the image, we only need to decompress previous components of the corresponding spectral portion.

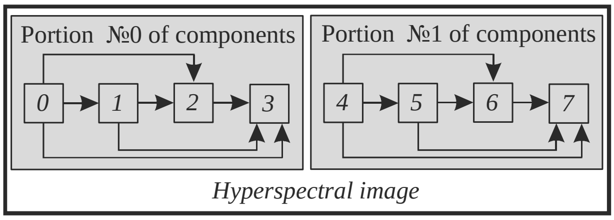

Fig. 12. Component approximation algorithm based on "independent portions of components" (N = 4*)*

"Shared Support Components". The compression ratio of the algorithm based on "independent portions of components" will inevitably be smaller, than the compression ratio of the algorithm based on "sliding component approximation". The reason is that, to approximate each spectral component, the "independent portions" algorithm uses (on average) much less basic components. The loss appear to be especially great during compression of the first component of each portion, since these components are not approximated. To reduce the effects described above, an approximation algorithm [10 – 11] based on "shared support components" is used (see Fig. 13).

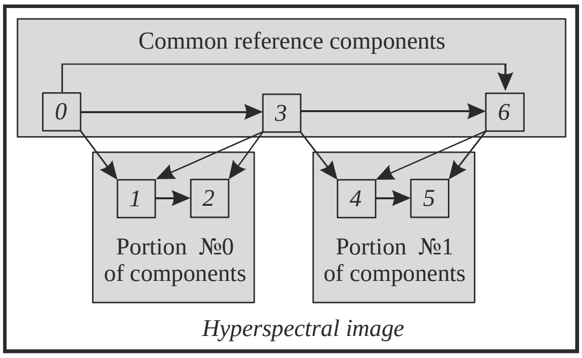

Fig. 13. Component approximation algorithm based on "shared support components" ($N = 3$, $C = 1$)

With a step N the "shared support" spectral components are selected from the image. An integer parameter C is also specified. The shared support components are compressed by the algorithm based on "sliding component approximation". To approximate each "shared support component", $(N + \mathrm{C} - 2)$ previous support components, which have already passed the compression, are used.

Next, after compressing "shared support components", the portions of components, located between "shared support" ones, are independently compressed. To compress these portions an approximation compression algorithm, similar to the one based on "sliding component approximation" is used. Components of each portion are compressed successively, with the use of approximation based on the components, which have already passed the compression. These approximating components are the previous components of the same portion, supplemented by the nearest "shared support components" in such a way that the total number of approximation components is equal to $(N + \mathrm{C} - 2)$. Thus, the parameter C is equal to the minimum number of "shared support components", used to approximate the "rest" of the components.

We can expect that the algorithm based on "shared support components" will have a better compression ratio, than the algorithm based on "independent portions of components". The price to pay is a speed reduction, since for the decompression of an arbitrary component, we have to decompress not only the previous components of the same portion, but also the required "shared support components".

To evaluate the compression effectiveness of proposed algorithms, we constructed the dependence of the compression ratio on the mean square / maximum error (see Fig. 14) for real 16-bit images made by hyperspectrometers SpecTIR [33] and AVIRIS [34].

The results obtained allow to draw the following conclusions:

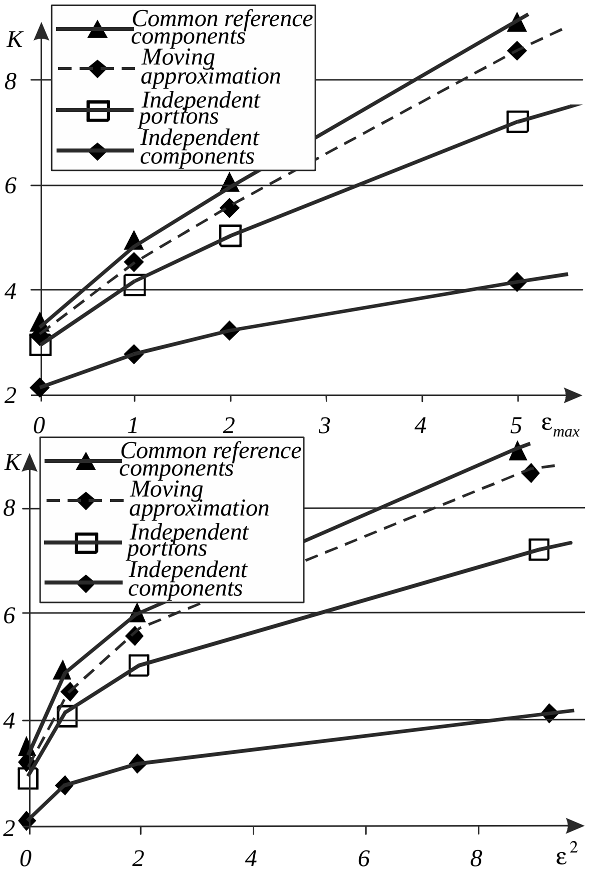

Fig. 14. Average for the five SpecTIR images compression ratio K depending on the maximum error $\varepsilon_{\max}$ *and on the mean square error* ε^2 *($L = 5$, $N = 7$, $C = 4$, $V_b \times H_b = 112 \times 614$)*

a) All algorithms demonstrate sufficiently high compression ratio, and can be recommended for use in hyperspectral image storage systems.

b) The use of any approximation of components allows to significantly improve the compression ratio. Approximation algorithms can be recommended for use.

c) Algorithms in a descending order of the compression ratio: "Shared Support Components", "Sliding Component Approximation", "Independent Portions of Components", "Independent Component Processing". Algorithm should be selected on the basis of allowable loss of decompression rate.

d) Benefit from the use of component approximation grows with the increase in compression ratio.

e) For zero and minor errors "Sliding Component Approximation" and "Shared Support Components" algorithms demonstrate the best results. Since these results are about the same, but "Shared Support Components" algorithm allows to reduce the access time, it is more preferable from these two algorithms (for minor errors).

2. Active HSI-Data Protection Methods based on Digital Watermarking

To solve problems in research and development of methods and algorithms for «active» HSI protection, new algorithms for digital watermark detection, extraction, and embedding into large-format digital multichannel and hyperspectral remote sensing images, were proposed, and also the robustness of these algorithms against the most common types of attacks and distortions was investigated.

2.1. HSI Protection against Unauthorized Copying and Dissemination using Robust High Capacity Digital Watermarks

To solve the problem of data protection against unauthorized copying, «robust» watermarks were developed. Such watermarks preserve embedded data while performing typical data manipulaions: compression, filtering, cropping, radiometric and geometric correction.

The analysis of existing robust watermarking approaches presented in [35 – 41] showed, that most of these approaches, and specific algorithms, require independent embedding of separate fragments of a watermark (an image or a bit vector) in each spectral channel of hyperspectral image; current approaches also do not consider specificity of hyperspectral data (in particular, a correlation of close spectral layers). An exception is a method described in a number of papers ([39, 40, 42, 43]) and based on the use of Karhunen-Loeve transform [43], or principal component analysis [39, 40], for hyperspectral data preprocessing before embedding or extracting a robust watermark. The advantage of this approach is the ability to «disperse» the watermark image across different "layers" of hyperspectral image (due to the use of Karhunen-Loeve transform), which can significantly reduce the amount of visible distortions introduced into individual HSI spectral channels by watermarking. At the same time, it is also one of the disadvantages of the proposed approach – since the original noise-like image used for watermark encoding is identical for all "layers" of hyperspectral image, it can be detected without knowing the embedding key, by comparing the number of "layers" available to attacker, and retrieving their common component. A detailed example of such an attack is considered in [40].

Unfortunately, the above-mentioned papers [39 – 42] lack any information on software implementation of these algorithms, and do not investigate the computational complexity of the proposed algorithms, which significantly complicates the evaluation of their practical effectiveness. This paper describes a comparison of computational complexity and robustness of methods proposed by the authors, and existing methods, using two most common algorithms: Digimarc algorithm and Cox algorithm, based on independent watermarking of individual HSI spectral channels.

One of the features of the proposed watermarking algorithm is an ability to embed a high-capacity watermark (from 100 to 4000 bits, depending on the number of spectral channels). The proposed watermarking algorithm is additive, i.e. it is based on addition of a two-dimensional noise-like signal encoding a watermark sequence (user identifier, right-holder code, serial number, or the date the image was obtained, etc.) to a carrier image. We can use an arbitrary sequence of bits $W(m)$, where $m \in [0, m-1]$, as the watermark sequence. During additive watermarking with the use of this bit sequence, a two-dimensional noise-like watermark image is generated. Such image does not depend on the carrier image, and it is robust against the most typical transformations of the carrier image (lossy compression, cropping, frequency domain filtering, nonlinear filtering, etc.).

Fourier synthesis of noise-like watermark signal $W_i(n_1, n_2)$ encoding a watermark sequence, is performed as follows. A $W(0)$ bit corresponds to a set of spectral points, equidistant from a zero component of a centered complex-valued spectrum at a distance $(R_{step})\cdot N_s/2$ (a set of such spectral points is hereinafter referred to as a "peak ring" of radius $(R_{step})\cdot N_s/2$). For all bits of sequence $W(m)$, where $m>0$, watermark image spectrum shaping is performed as follows. Using the bit sequence $W(m)$, we form an array of integers $\phi_{shift}(m')$, which determines a cyclic shift of the m' th spectrum peak ring relative to the vertical axis, and a bit sequence $\alpha(m')$, which determines the use of a pseudo-random bit sequence $S_1(n)$ (or $S_0(n)$) in order to form m' th spectrum peak ring:

$$\phi_{shift}(m') = \\ = \begin{cases} 0, & \text{if } m'=0, \\ \sum_{m=1}^{5} 2^{(m+1)} \cdot W\left(6\cdot(m'-1)+m+1\right), & \text{else}, \end{cases}$$

$$\alpha(m') = \begin{cases} W(0), & \text{if } m'=0, \\ W\left(6\cdot(m'-1)+m+1\right), & \text{else}. \end{cases}$$

Next, on the basis of the binary pseudo-random sequences $S_1(n)$ and $S_0(n)$, which are used as a steganographic key, and the watermark sequence $W(m)$, a complex-valued Fourier spectrum $C(u, v)$ of size $N_s \times N_s$ pixels (spectrum of the two-dimensional noise-like signal encoding the watermark) is generated:

$$real\left(C(u,v)\right) = \begin{cases} \cos\left(rand90(u,v)\right), \\ \quad \text{if } S_1\left(n_{shifted}\right)=1 \text{ и } \alpha\left(R_m\right)=1, \\ \cos\left(rand90(u,v)\right), \\ \quad \text{if } S_0\left(n_{shifted}\right)=1 \text{ и } \alpha\left(R_m\right)=0, \\ 0, \text{ else}, \end{cases}$$

$$imag\left(C(u,v)\right) = \begin{cases} \sin\left(rand90(u,v)\right), \\ \quad \text{if } S_1\left(n_{shifted}\right)=1 \text{ и } \alpha\left(R_m\right)=1, \\ \sin\left(rand90(u,v)\right), \\ \quad \text{if } S_0\left(n_{shifted}\right)=1 \text{ и } \alpha\left(R_m\right)=0, \\ 0, \text{ else}, \end{cases}$$

where $u \in [0, N_s/2-1]$, $v \in [0, N_s/2-1]$;

$rand90(u, v)$ is a random number in the range of [0, 90] (uniform distribution) generated for a point (u, v);

$$n_{shifted} = \left(\arctan\left(\frac{u}{v}\right) + \alpha\left(\frac{\sqrt{u^2+v^2} - \hat{R}\cdot {}^{N_s}\!/_2}{R_{step}\cdot {}^{N_s}\!/_2} \right) \right) \bmod N;$$

R_{step} is a distance between neighboring sequences of the spectrum, expressed in $N_s / 2$;

$\hat{R}$ is a distance from zero spectrum component to the first code sequence, expressed in $N_s / 2$;

$$R_m = (\sqrt{u^2+v^2} - \hat{R}\cdot N_s / 2) \div (R_{step}\cdot N_s / 2);$$

real($C(u, v)$) and *imag*($C(u, v)$) are real and imaginary parts of a complex spectrum respectively.

By computing the inverse DFT, the resulting spectrum is converted into image $W_i(n_1, n_2)$, where $n_1, n_2 \in [0, N_s - 1]$.

Next, the carrier image $I(m_1, m_2)$, where $m_1 \in [0, M_1 - 1]$, $m_2 \in [0, M_2 - 1]$, $M_1 > N_s$, $M_2 > N_s$, is watermarked as follows.

Let $Q \in [0, 1]$ be a user-defined parameter called "watermarking intensity". For the carrier image $I_{disp}(m_1, m_2)$ a local dispersion field is calculated:

$$I_{disp}(m_1, m_2) = \\ = \frac{1}{L^2}\sum_{i=-L/2}^{L/2}\sum_{j=-L/2}^{L/2} I(m_1+i, m_2+j)^2 - \\ - \frac{1}{L^2}\left(\sum_{i=-L/2}^{L/2}\sum_{j=-L/2}^{L/2} I(m_1+i, m_2+j)\right)^2,$$

where $L < N_s$ is a size of window used for local dispersion field $I_{disp}(m_1, m_2)$. calculation. Then, for a synthesized image $W_i(n_1, n_2)$ the average brightness and dispersion are calculated:

$$W_{disp} = \frac{1}{N_s^2}\sum_{n_1=0}^{N_s}\sum_{n_2=0}^{N_s} W_i\left(n_1, n_2\right)^2 - \frac{1}{N_s^2}\left(\sum_{n_1=0}^{N_s}\sum_{n_2=0}^{N_s} W_i\left(n_1, n_2\right)\right)^2,$$

$$W_{mean} = \frac{1}{{N_s}^2}\sum_{n_1=0}^{N_s}\sum_{n_2=0}^{N_s} W_i(n_1, n_2).$$

After that, the image $I(m_1, m_2)$ is watermarked according to the relation:

$$I'\left(m_1, m_2\right) = I\left(m_1, m_2\right) + Q\cdot\sqrt{(I_{disp}\left(m_1, m_2\right)/W_{disp})}\times \\ \times\left(W_i\left(m_1 \bmod N_s, m_2 \bmod N_s\right) - W_{mean}\right),$$

where $I'(m_1, m_2)$ is a watermarked image.

The major advantage of the proposed watermarking algorithm, compared with other robust additive watermarking algorithms, is the higher watermarking capacity, i.e. the allowable length of sequence $W(m)$, embedded into a separate color channel of the carrier image. For example, if the maximum allowable number of peak rings is 8, we can embed a 43-bit sequence $W(m)$ into a separate spectral channel of the image.

Algorithm for Pseudo-Holographic Redundant Coding of Digital Watermarks. During the development of presented algorithms for HSI protection, based on watermarking, we designed a new algorithm for pseudo-holographic redundant coding of the watermark sequence $W(m)$, which allows to increase the size of the embedded watermark in proportion to the number of spectral channels of the carrier image, while providing watermark robustness against the loss (intentional removal) of some spectral channels of the carrier image. The proposed algorithm provides a pseudo-random distribution of watermark bits between the spectral channels of the carrier image, and thus provides a higher watermark capacity (in comparison with existing algorithms).

Let us consider the proposed method for pseudo-random distribution of digital watermark bits between different spectral channels of HSI. Let us divide H – a bit sequence embedded into HSI as a robust watermark – into a set of L disjoint subsequences H_i ($i \in [0, L-1]$),each of K bit length. Before the watermarking, we form L independent fragments $S_0, S_1, S_2, S_3, \ldots, S_{L-1}$ from the sequence H. Each fragment in its turn consists of $M = ceil(\log_2(L) + K)$ bits (*ceil* is an operation of rounding down) and is formed according to the following rule:

$$S_i = n_0 n_1 n_2 n_3 \ldots n_{M-2} H_i,$$

where $n_0 n_1 n_2 n_3 \ldots n_{M-2}$ is a value of the index i in the binary number system. For instance, for $L = 8$, and $H_7 = 0$, the fragment S_7 is formed as 4-bit sequence "1110", where "111" is a binary representation of index 7. Hereinafter the first $M - 1$ fragment bits will be referred to as "index portion", and the last K bits will be referred to as "informational portion".

Next, in the process of watermarking the sequence of HSI spectral channels, for each channel we **randomly, or pseudo-randomly,** select one fragment from *a set* $S_0, S_1, S_2, S_3, \ldots, S_{L-1}$. After that, the selected fragment is embedded into the current HSI spectral channel, using the robust watermarking algorithm described above.

Thus, in addition to the embedding capacity increase, the proposed approach allows to avoid the static or repetitive structure of a payload, what makes the watermark more robust against the number of steganographic attacks [44].

Algorithm for Robust Watermark Extraction from Hyperspectral Satellite Images, Based on Blockwise Digital Image Processing. Watermark extraction from the watermarked image $I'(m_1, m_2)$ is performed as follows (in case of watermark extraction from HSI, we assume that $I'(m_1, m_2)$ is a separate HSI spectral channel).

In the first step, the watermarked image $I'(m_1, m_2)$ is divided into K disjoint fragments $I'_k\left(n_1, n_2\right)$ where $n_1, n_2 \in [0, N_s - 1]$. Next, for each fragment, a modulus of the centered spectrum, $C_k(n_1, n_2)$, and averaged (over all fragments) modulus of the spectrum,

$$C'\left(n_1, n_2\right) = \frac{1}{K}\sum_{k=0}^{K-1} C_k\left(n_1, n_2\right),$$

are calculated.

Then, a polar coordinate representation $C_{polar}(r, l)$ of the averaged spectrum modulus $C'(n_1, n_2)$ is calculated:

$$C_{polar}\left(r, l\right) = C'\left(N_s / 2 - r\sin l, N_s / 2 - r\cos l\right),$$

where $r \in [0, N_s / 2]$, $l \in [0, N - 1]$.

After that, we calculate values $R_0(r)$ and $R_1(r)$ of convolution of $C_{polar}(r, l)$ and each of the code sequences $S_1(n)$ and $S_0(n)$, which are used as a steganographic key:

$$\hat{C}_{polar}(r,l)=\frac{C_{polar}(r,l)-\frac{1}{N^2}\sum\limits_{l=0}^{N}C_{polar}(r,l)}{\frac{1}{N^2}\sum\limits_{l=0}^{N}C_{polar}(r,l)^2-\frac{1}{N^2}\left(\sum\limits_{l=0}^{N}C_{polar}(r,l)\right)^2},$$

$$R_0(r)=\max_{\Delta n}\sum_{n=0}^{N-1}S_0(n)\cdot\hat{C}_{polar}\left(r,\mathrm{mod}\left(n+\Delta n\right)/N\right),$$

$$R_1(r)=\max_{\Delta n}\sum_{n=0}^{N-1}S_1(n)\cdot\hat{C}_{polar}\left(r,\mathrm{mod}\left(n+\Delta n\right)/N\right).$$

Next, we search for the smallest $r_0\in[R_i, N_s/2]$ such that satisfies the condition $|R_1(r_0)-R_0(r_0)|>T$, where T is a detection threshold determined experimentally.

When extracting the bit $W(m)$, where $m>0$, we take into account not only the condition $R_1(r_0)<R_0(r_0)$, but also the calculated value Δn for the current row r_0. If for a certain r_n, where $n>0$, the condition $|R_1(r_n)-R_0(r_n)|>T$ is fulfilled, then next 6 bits of watermark sequence are extracted as follows:

$$W\left(6\left(n-1\right)+1\right)=\begin{cases}0, \text{ если } R_1\left(r_n\right)<R_0\left(r_n\right),\\ 1, \text{ если } R_1\left(r_n\right)>R_0\left(r_n\right).\end{cases}$$

$$W\left(6\left(n-1\right)+1+i\right)=\begin{cases}0, \text{ если } \Delta n\le 2^{i+1},\\ 1, \text{ если } \Delta n>2^{i+1},\end{cases}$$

where i is an integer, $i=[1.5]$.

The watermark extraction is considered complete, if at the next extraction step we cannot find r_n satisfying the condition $|R_1(r_n)-R_0(r_n)|>T$.

Then, after extracting the watermark from all spectral channels of the watermarked image, the original full watermark H is reconstructed.

We assume that, after extracting watermark sequence fragments, we obtain K fragments W_j, $j\in[1,K]$, which size matches the size of fragments when embedding (for the embedding algorithm discussed in this section, the fragment size is 43 bits). However, due to the possible loss (distortion) of the digital watermark in certain spectral channels, the number of extracted fragments K may be considerably smaller than the total number of spectral channels of hyperspectral image. Nevertheless, the loss of digital watermark fragments does not imply inability to extract the digital watermark. Actually, the fragment $W_j=n_0n_1n_2n_3\ldots n_{M-2}H_i$ extracted from an arbitrary frame can be unambiguously divided by digital watermark decoder into "index" and "informational" portions. After that the index i of the extracted "informational" portion of the fragment H_i of the original informational sequence H can also be calculated from the bit sequence $n_0n_1n_2n_3\ldots n_{M-2}$ without any additional data about the number of the current frame.

Thus, the proposed pseudo-holographic watermark coding algorithm allows to adapt the correlation-based watermarking approach to work with a wider class of multi- and hyperspectral images. First, binary watermark can be divided among any number of independent spectral channels, which allows to increase watermarking capacity proportionally to the number of channels available for embedding. At the same time, this "divided" watermark sequence can be restored even in case, when the majority of watermarked image spectral channels were damaged or removed, and the remaining spectral channels are reordered within the image. Next, if some spectral channels of the carrier image are highly correlated, then watermark fragments, prepared for embedding into these spectral channels, can be also formed as highly correlated or bitwise equal (this allows to decrease watermark visibility and prevent attacks of "watermark estimation" class).

2.2. HSI Protection Against Forgery Using Semi-Fragile Digital Watermarking and Adjustable Brightness Requantization

The proposed method for so-called "fragile" or "semi-fragile" digital watermarking allows to simultaneously perform the tasks of the image protection against modifications, and the task of hidden data transmission within the image. The developed algorithm provides robustness of the digital watermark against elementwise transformations of the watermarked image (contrasting), cropping and 90 degree multiple angle rotation.

The proposed watermarking algorithm is similar to a known QIM algorithm [45,46], and it can be briefly described as follows. Assume that an input image $I(n,m)$ and a binary watermark image $W(n,m)$, where $n\in[1,N]$, $m\in[1,M]$, are given.

For integers $I(n,m)$ (input image) and $I'(n,m)$ (output image), digital watermark embedding is performed according to the following rule:

$$I'(n,m)=\begin{cases}\mathrm{floor}\left(\frac{I(n,m)}{q}\right)+\mathrm{mod}\left(\frac{2\cdot I(n,m)}{q}\right),\\ \qquad\qquad\text{if } W\left(m,n\right)=0,\\ \mathrm{floor}\left(\frac{I(n,m)}{q}\right)+\frac{q}{2}+\mathrm{mod}\left(\frac{2\cdot I(n,m)}{q}\right),\\ \qquad\qquad\text{if } W\left(m,n\right)=1,\end{cases}$$

where floor is an operation of rounding down, mod is a division remainder, q is a quantization step, which determines a distortion ratio of the watermarked image.

Digital watermark extraction is performed as follows:

$$W'(n,m)=\begin{cases}0, \text{ if } \mathrm{mod}\left(\frac{I'(n,m)}{q}\right)<\frac{q}{2}\\ 1, \text{ if } \mathrm{mod}\left(\frac{I'(n,m)}{q}\right)\ge\frac{q}{2}\end{cases},$$

where $W'(n,m)$ is an extracted watermark.

Next, let us consider the algorithm for embedding one watermark bit into the pixel block of the carrier image. This algorithm simultaneously provides hidden data transmission and watermarked image protection against modification.

Let L be a linear size of the pixel block of the carrier image, into which the digital watermark pixel (bit) is embedded, $K(i,j)$, where $i\in[1,L]$; $j\in[1,L]$ is a pseudorandom binary image K, which is a secret "key", and the embedded binary image $W_b(u,v)$ $W_b(u,v)$ (verification information), where $u\in[1,\mathrm{floor}(N/L)]$, $v\in[1,\mathrm{floor}(M/L)]$ is a watermark image.

Thus,

$$I'(n,m)=\begin{cases}\mathrm{floor}\left(\dfrac{I(n,m)}{q}\right)+\mathrm{mod}\left(\dfrac{2\cdot I(n,m)}{q}\right), \\ \qquad\qquad \text{if } W_b^{'}(n,m)=0, \\ \mathrm{floor}\left(\dfrac{I(n,m)}{q}\right)+\dfrac{q}{2}+\mathrm{mod}\left(\dfrac{2\cdot I(n,m)}{q}\right), \\ \qquad\qquad \text{if } W_b^{'}(n,m)=1,\end{cases}$$

where

$$W_b^{'}(n,m)=W_b\left(\mathrm{floor}(n/L),\mathrm{floor}(m/L)\right)\oplus$$
$$\oplus K\left(\mathrm{mod}(n/L),\mathrm{mod}(m/L)\right),$$

$\oplus$ is an "exclusive OR" operation.

If the value q is known, and the watermarked image is not distorted, then extraction of watermark bit $W'(u,v)$ from the image block $I_b(n,m)$, where $n\in[uL,u(L+1)]$, $m\in[vL,v(L+1)]$, is performed as follows:

$$W'(u,v)=\begin{cases}0, \\ \text{if } \sum\limits_n\sum\limits_m\left(\mathrm{mod}\left(\frac{I'(n,m)}{q}\right)-K_{\mathrm{mod}}(n,m)\right)=0, \\ 1, \\ \text{if } \sum\limits_n\sum\limits_m\left(\mathrm{mod}\left(\frac{I'(n,m)}{q}\right)-\overline{K}_{\mathrm{mod}}(n,m)\right)=0, \\ \text{is not defined, else,}\end{cases}$$

where

$$K_{\mathrm{mod}}(n,m)=K\left(\mathrm{mod}(n/L),\mathrm{mod}(m/L)\right)\cdot(q/2)$$

$$\overline{K}_{\mathrm{mod}}(n,m)=1-K_{\mathrm{mod}}(n,m)$$

If after digital watermark extraction, a number of values $W'(u,v)$ are not determined, it is implied that the respective image block was modified, which led to the destruction of watermark bit. In this case the calculation of $W'(u,v)$ combines the extraction and the detection procedures (i.e. the detection of watermarked image modifications, if $W'(u,v)$ is not determined).

In case when the watermarked image is contrasted, the embedded watermark can be detected by performing the following procedure. For all pixels of the selected block $I_b'(n,m)$ such that $K(\mathrm{mod}(n/L), \mathrm{mod}(m/L))=0$, a brightness histogram $H_0(b)$ is constructed. Similarly, for all pixels of the selected block $I_b'(n,m)$ such that $K(\mathrm{mod}(n/L), \mathrm{mod}(m/L))=1$ a brightness histogram $H_1(b)$ is constructed. Next, for each block we calculate

$$D(u,v)=\begin{cases}0, \sum\limits_b\left(H_0(b)\cdot H_1(b)=0\right), \\ 1, \sum\limits_b\left(H_0(b)\cdot H_1(b)>0\right).\end{cases}$$

A binary image $D(u,v)$ represents a "map" of modified blocks (0 – the block is not distorted, 1 – the block is distorted).

The robustness of watermark detection procedure against 90 degree multiple angle rotation of the watermarked image is provided by introducing additional limitations on the form of the key $K(i,j)$ used for embedding:

$$\forall i,j\in[1,L/2]$$
$$K(i,j)=K(L-j,i)=K(L-i,L-j)=K(j,L-i)$$

If the watermarked image is cropped, and, consequently, the original block boundaries are moved, then for a given pixel block the watermark detection procedure takes the following form.

Step 1. For all possible offset values $\Delta n,\Delta m\in[0,L-1]$, a set of histograms $H_0(\Delta n,\Delta m,b)$, which include all pixels of the selected block $I_b'(n,m)$ such that

$$K\left(\mathrm{mod}((n+\Delta n)/L),\mathrm{mod}((m+\Delta m)/L)\right)=0,$$

is constructed. Similarly, a set $H_1(\Delta n,\Delta m,b)$ is constructed.

Step 2. Next, for each block, the "map" of modified blocks is calculated:

$$D(u,v)=\begin{cases}0, \text{ если} \\ \min\limits_{\Delta n,\Delta m}\sum\limits_b\left(H_0(\Delta n,\Delta m,b)\cdot H_1(\Delta n,\Delta m,b)\right)=0, \\ 1, \text{ если} \\ \min\limits_{\Delta n,\Delta m}\sum\limits_b\left(H_0(\Delta n,\Delta m,b)\cdot H_1(\Delta n,\Delta m,b)\right)>0,\end{cases}$$

and the digital watermark is extracted similarly to the previous case.

Thus, the proposed algorithm for blockwise watermarking has the following advantages over the existing methods:

– the algorithm allows to embed an arbitrary watermark sequence into the carrier image, and detect modifications, introduced into watermarked image, simultaneously;

– the algorithm provides watermark robustness against cropping, linear contrasting, and 90 degree multiple angle rotation.

2.3. Experimental Study of HSI Protection Effectiveness and Watermark Robustness Against Different Types of Distortion

During the experimental study of robust watermarking algorithms considered earlier in this section, we used a set of (single-channel) satellite images of size 4000×4000 pixels, represented in uncompressed TIFF format. These images, in turn, represented individual spectral channels of AVIRIS HDR format HSI comprising 450 spectral channels.

First of all, we experimentally evaluated robustness against unintentional watermark distortion, i.e. detector ability to properly extract a watermark fragment from a separate distorted spectral channel of the watermarked hyperspectral image. For different values of the parameter Q (the watermarking intensity), a probability of the correct detection and extraction of the digital watermark fragment from the distorted image, was evaluated.

To conduct this experiment the following scenarios for distortion of separate watermarked spectral channel have been developed (we assume that the scenario is applied independently to all spectral channels of the watermarked image).

Scenario 1 – *Cropping***.** We cut a 2048×2048 pixel fragment from the watermarked image; a fragment position on the original image was selected randomly. After that, the cropped fragment was used as the watermarked image.

Scenario 2 – *Additive noise adding.* We added additive white Gaussian noise with a standard deviation parameter $\sigma = 15$ to a watermarked image.

Scenario 3 – *Median filtering.* We subjected the watermarked image to median filtering, using 3×3 pixel window.

Scenario 4 – *Rotation and cropping.* We rotated the watermarked image by 15 and then cut a fragment of size 2048×2048 pixels.

Scenario 5 – *JPEG- compression.* We subjected the watermarked image to JPEG-compression; the value of a compression quality parameter was set to 80.

Scenario 6 – *Resampling*. We subjected the watermarked image to resampling with a scaling factor of 1.5, and using a linear interpolation algorithm as a resampling algorithm.

For the considered set of watermarked images, the effectiveness of independent watermark fragment extraction $P_{ext} = N_{ext} / 100$ was calculated for each distorted spectral channel after applying one of distortion scenarios. The value $N_{ИЗВ}$ represented the number of images in the set, for which the watermark was detected and properly (with a bit-precision) extracted. The results of experimental research on watermark extraction effectiveness are shown in Table 1.

Table 1. Effectiveness of robust watermark extraction using different image distortion scenarios

Watermarked image distortion	Watermark extraction effectiveness, P_{ext}			
	$Q = 0.07$	$Q = 0.15$	$Q = 0.25$	$Q = 0.3$
Scenario 1	0.99	1.0	1.0	1.0
Scenario 2	0.993	0.997	1.0	1.0
Scenario 3	0.97	0.981	1.0	1.0
Scenario 4	0.98	0.99	1.0	1.0
Scenario 5	0.96	0.96	0.98	1.0
Scenario 6	0.994	0.998	1.0	1.0

The obtained results show, that the developed algorithms for watermark extraction are robust enough (for application in practice) against common distortions of the watermarked image.

During the experiment, it was shown, that if the watermark fragment size is more than 1 bit (implementation of watermarking algorithm, proposed earlier, implies the watermark fragment size of 43 bit/spectral channel), the probability of false watermark fragment detection is less than 0.000001, which is acceptable for practical application of this algorithm.

Also, we conducted a research on the effectiveness of the applied scheme for pseudo-holographic watermark encoding, which allows you to restore the original watermark from fragments even in case of removal and \ or distortion of some watermarked spectral channels.

We considered such variants of original watermark partition into subsequences, that the number of different subsequences embedded into the individual spectral channels, was from 8 to 128 (in all cases the total number of embedded watermark fragments was equal to the number of spectral channels – 450). Using each of the considered partitions, we formed fragments of watermark sequence and embedded them into hyperspectral image of size 4000×4000 pixels (450 spectral channels). Next, N_{dist} spectral channels were randomly removed from watermarked hyperspectral image, and the watermark was extracted from the remaining channels. Extraction was considered successful, if after extracting fragments it was possible to restore the full original watermark. The results of experimental estimation of the probability of complete watermark extraction using only part of the spectral channels, are given in Table 2 (for each parameter combination N_{dist} / N_{seq}).

Due to the obtained results we can state that the proposed information technology provides a probability $P_{00} > 0.96$ of hyperspectral image forgery detection without using pseudo-holographic redundant coding, and a probability $P_{00} > 0.995$ when using pseudo-holographic coding.

To study the effectiveness of a fragile watermarking algorithm proposed by the authors, and its properties in the case of remote sensing data (satellite images) protection against falsification, we embedded a watermark of size 256×256 pixels into a set of 22 grayscale carrier images of size 4096×4096 pixels (an example of the image is shown in Fig. 15) using the pseudorandom embedding key of size 16×16 pixels.

Table 2. Experimental estimation of the probability of successful restoration of full watermark from fragments in case of removal or distortion of some spectral channels (for a set of 1000 realizations)

The number of watermark subsequences N_{seq} (**watermark length**)	The number of distorted spectral channels, N_{dist}					
	50	100	150	200	250	300
128(**4608**)	0.01	0	0	0	0	0
64 **(2368)**	0.88	0.77	0.55	0.26	0.04	0.001
32(**1216**)	0.99	0.99	0.99	0.98	0.94	0.75
16(**624**)	1	1	1	1	1	1

Next, we introduced the following modifications into each watermarked image (Fig. 15):

– white Gaussian noise was added to image block 1 (noise standard deviation was $\sigma = 15$);

– block 2 was subjected to a Gaussian blur (defocus parameter $\sigma_p = 5$);

– block 3b was replaced by block 3a.

The location and size of blocks differed for all watermarked images.

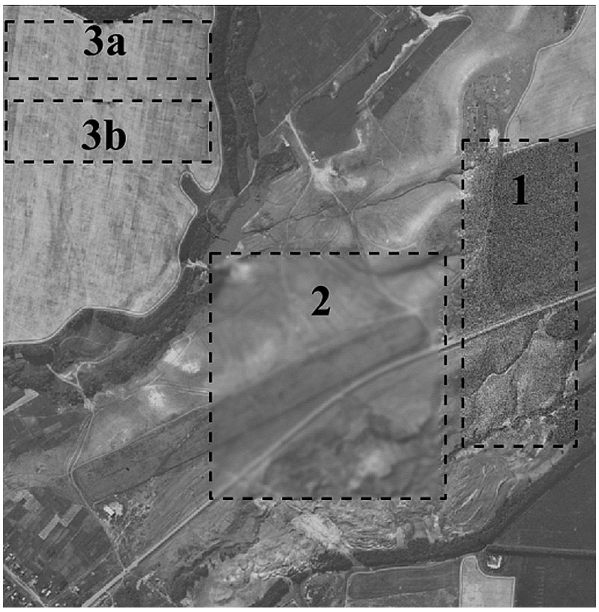

Fig. 15. Distorted watermarked image

For each modified image, procedures of watermark detection and extraction were performed. According to a study conducted for the whole set of watermarked images, modified parts of the image were detected and marked with the block size precision (16×16 pixels) for all images of the test set.

Besides, for the test set of 22 grayscale images of size 4096×4096 pixels we determined minimum parameters for distortion of types discussed above (Gaussian blur, noise adding, block replacement), for which the probability of modified area detection was $P_{00} > 0.9999$. For digital watermark embedded into each block pixel with parameters $L = 16$, $q = 2$, we obtained the following minimum distortion parameter values:

– minimum size of the replaced block -5×5 pixels (for the replacement of watermarked image block);

– minimum standard deviation of white Gaussian noise $\sigma = 0.65$ (for additive noise adding);

– minimum defocus parameter $\sigma_p = 0.45$ (for Gaussian blur).

According to obtained results, we state, that the watermark detection algorithm allows to detect visually significant image modifications of the most common types (noise adding, block replacement, blur) with a probability $P_{00} > 0.9999$.

The results show, that the watermark embedded by using the developed algorithm is robust against a previously listed set of image transformations, and also, even in case of such distortion, allows to detect the modified image regions.

2.4. Development of Algorithm for Generating a Robust and Semi-Fragile Large-Size Watermark with Improved Resistance Against Intentional Attacks

This section describes a method for improving robustness of digital watermarks, embedded into large-format multi-channel and hyperspectral images, based on the use of pre-generated multidimensional (two- or three-dimensional) noise-like digital watermarks. Moreover, we propose a new algorithm for generating such noise-like binary digital watermarks based on a secret key (user password). Such algorithms and methods allow to implement (without reducing the overall steganographic robustness of the embedded watermark) the most computationally complex embedding operations (e.g., generation of a noise-like additive watermark, robust to a wide range of distortions) in a preprocessing step, and, thereby, significantly (up to 5-10 times) reduce the computational complexity of the procedure for watermarking the large-format remote sensing image.

It is known that, in case of implementing an intentional attack in order to detect the embedded watermark, it can be extracted or removed with the use of a so-called "brute force" strategy. In this case, the attacker sequentially tries all possible embedding keys, and chooses the key, that allows to detect and extract the watermark from a given image, as the "true" one (it is assumed, that the attacker is certain about the presence of the watermark in this image). Later, knowing the "true" embedding key allows the attacker both extract and remove all watermarks embedded with the use of this key.

As an example, consider some robust watermarking algorithms based on spread spectrum coding [47 - 50].

These algorithms use the secret steganographic key for the selection (generation) of an m-sequence, or Kasami sequence, and then use the selected sequences for modulation and coding of the embedded binary watermark. In fact, in these algorithms the m-sequence used for modulation is the only secret information required for watermark extraction. When such approach is used, the length of m-sequence is approximately equal to the number of carrier image pixels: thus, for the image of size 640×480 pixels we need to use the m-sequence of length approximately 2^{18} bit. It is known that the number of different sequences of a given length is bounded above by the value of Euler function; for the m-sequence of length 2^{18} bit the number of different sequences can be evaluated (according to [51]) as $\approx 5 \cdot 10^4$. Therefore, if attackers know about the use of m-sequences for watermark modulation, they can try to extract the watermark by brute forcing all m-sequences of a given length. Consequently, using the found original m-sequence, attackers can implement a "watermark estimation attack", and remove the embedded watermark with minimal loss of carrier image quality.

Next, we propose new algorithms for generation of multidimensional noise-like digital watermarks, which allow both to improve watermark robustness against attacks, and to reduce computational complexity of watermarking procedures.

For the majority of watermarking algorithms it is required to use two-dimensional key bit sequences instead of one-dimensional (such as the user password). In this case, the secret "key" (user password) is first converted into two-dimensional noise-like "template" image, which is directly embedded into the carrier image.

On the one hand, the algorithm for key sequence generation proposed in this section provides the low (compared with the truistic case discussed above) probability of collisions due to the adjustability of minimum cyclic Hamming distance [50]. In this case, the collision means a situation, when the watermark embedding is performed using one secret embedding key, and the extraction and verification can be performed using a number of "false" keys, which are close to the original one, according to the Hamming distance criterion. On the other hand, the proposed algorithm significantly exceeds the existing analogues by the number of various key sequences of a given length (and, hence, by the robustness against brute-force attacks).

At the first stage of the developed algorithm, the secret user password is converted into the one-dimensional key sequence with fixed minimum cyclic Hamming distance. In its turn, the first stage of the algorithm, discussed in this section, can be divided into the following steps.

In the first step, the user password, represented as a bit sequence $P = p_1p_2p_3\ldots p_n$, is encoded using a redundant BCH code with a minimum code distance l. The result of the encoding is a bit sequence $T_{pass} = t_1t_2t_3\ldots t_{n+k}$ (where bits $t_{n+1}t_{n+2}t_{n+3}\ldots t_{n+k}$ are added as a result of redundant coding). Obviously, in this step, the number of possible user passwords of length n, and the number of the corresponding binary sequences T_{pass}, is 2^n. Due to the properties of BCH code, we can also assert, that the Hamming distance between two arbitrary sequences T'_{pass} and T''_{pass} of equal length is not less than l, where l is a minimum BCH code distance used in this step.

In the second step, for the further transformation of the user password into the key sequence C we use the comma-free binary code, which consists of four code words $\tilde{S} = \{S_{00}, S_{01}, S_{10}, S_{11}\}$ (the length of all code words is s bit), and the m-sequence $M = m_1m_2m_3\ldots m_{n+k}$, where m_i is the i^{th} bit of the sequence M, $1 \le i \le n+k$. The key sequence $C = c_1c_2c_3\ldots c_{(n+k)\cdot s}$ is formed on the basis of the sequence T_{pass} obtained previously, according to the following equation:

$$\{c_j c_{j+1} c_{j+2} \ldots c_{j+s-1}\} = \begin{cases} S_{00}, \text{ if } t_i = 0 \text{ и } m_i = 0, \\ S_{01}, \text{ if } t_i = 1 \text{ и } m_i = 0, \\ S_{10}, \text{ if } t_i = 0 \text{ и } m_i = 1, \\ S_{11}, \text{ if } t_i = 1 \text{ и } m_i = 1, \end{cases},$$

where $j = i\cdot s$, $1 \le i \le n+k$.

The cyclic Hamming distance between two key sequences C' and C'' of length L can be defined as

$$\lambda_{c'c''} = \min_{\Delta h} \sum_{h=1}^{h} c'_h \oplus c''_{h+\Delta h},$$

where $0 \le \Delta h \le L$ and $(h + \Delta h)$ is calculated modulo L.

For cyclic invariant code $\tilde{C}$, the minimum cyclic Hamming distance is defined as a minimum possible value of the cyclic Hamming distance among all possible pairs of code words in $\tilde{C}$. As shown in [52], the minimum cyclic Hamming distance of key sequences used for watermarking directly affects the robustness of the embedded watermark against brute-force attacks. In [52] it is shown, that for two arbitrary key sequences C' and C'', according to the rule (5), the cyclic Hamming distance does not exceed the value:

$$\lambda = \min(l \cdot h_s, h_s \cdot (n+k)/2, h_c \cdot (n+k)),$$

where h_c is a comma-free Hamming distance determined for comma-free code $\tilde{S}$; h_s is a minimum code distance of $\tilde{S}$ (without considering possible shifts). Some examples of generating the comma-free code $\tilde{S}$, which consists of four code words of equal length b, are given in Table 3.

Besides, in [47] it is shown, that by the number of possible key sequences of a given length, the code, considered in this section, is significantly superior to other known cyclic invariant codes, which allow to generate the code word on the basis of the password set by user.

Table 3. Examples of comma-free codes, comprising 4 code words

b	h_c	h_s	S^{00}	S^{01}	S^{10}	S^{11}
5	1	1	10110	01001	10111	01000
6	1	1	101100	010011	101000	010111
7	2	1	0001101	1110010	0011101	1100010

At the second stage of the algorithm for generating two-dimensional noise-like templates, the one-dimensional key sequences, generated at the first stage, are converted into the two-dimensional noise-like binary templates, while maintaining a fixed minimum cyclic Hamming distance.

Next, to convert the one-dimensional key sequence C into two two-dimensional noise-like templates M_0 and M_1 for further watermarking, we need (according to the requirements discussed earlier in this section) to perform the following steps.

In the first step, we need to generate two temporary binary arrays $R_0(v, h)$ and $R_1(v, h)$ of size $V \times H$ bit, where $H = (n+k)\cdot s$, $V\cdot H = M\cdot N$, by using the algorithm for producing optical orthogonal codes, represented in [50]. Thus, the first rows of arrays $R_0(v, h)$ and $R_1(v, h)$ are formed according to the relations $R_0(1, h) = C_h$, $R_1(1, h) = C_h$. Then, the g^{th} row of the array $R_0(v, h)$ is formed through cyclic shift of $(g-1)$ th row by $g\cdot q_0$ positions (where q_0 is prime, and the value $g\cdot q_0$ is calculated modulo integer H). Similarly, the array R_1 is formed: $(g-1)^{\text{th}}$ row of the array is formed through cyclic shift of the previous row by $g\cdot q_1$ positions ($q_1 \ne q_0$ is prime, the value $g\cdot q_0$ is calculated modulo H).

In papers [49] and [50] it was proved, that for two-dimensional binary arrays $R_0(v, h)$ and $R_1(v, h)$ the value of the cyclic Hamming distance is close to the maximum (i.e. to $V\cdot H$) in case of non-zero shift values (in case of cyclic shift of two-dimensional arrays, it is assumed that the array $R_0(v, h)$ remains unchanged, and the array $R_1(v, h)$ is cyclically shifted by Δh rows and Δv columns).

In the last step, arrays $R_0(v, h)$ and $R_1(v, h)$ are converted into templates M_1 and M_0 of size $N\cdot M$, according to the following relations:

$$M_0\left(u \bmod N, u \bmod M\right) = R_0\left(u \bmod V, u \bmod H\right),$$
$$M_1\left(u \bmod N, u \bmod M\right) = R_1\left(u \bmod V, u \bmod H\right),$$

where $u \in [1, N \cdot M]$.

According to [49] and [50], the proposed sequence of transformations of original key sequence C also allows to keep the minimum cyclic Hamming distance equal to λ for resulting two-dimensional templates, and such cyclic Hamming distance takes place only for zero shift values. The difference between obtained two-dimensional templates from original key sequences only lies in the fact, that in the case of two-dimensional templates, the template cyclic shift is also two-dimensional, i.e. templates are shifted by Δh rows and Δv columns. With this in mind, and taking into account the fact that obtained noise-like templates M_1 and M_0 consist of V repetitions of original key sequence C, the minimum cyclic Hamming distance for templates M_1 and M_0 can be calculated as:

$$\lambda_{2d} = V \cdot \min(l \cdot h_s, h_s \cdot (n+k)/2, h_c \cdot (n+k)).$$

Actually it means that there are no such two-dimensional key sequences M_0 and $M_0^{'}$, that cyclic Hamming distance between them is less than λ_{2d}.

To investigate the robustness of developed algorithms for generating key sequences, both against unintentional distortions and intentional attacks on the digital watermark, we conducted a number of computational experiments. For all experiments we used binary templates M_1 and M_0 of size 4095 × 4096 pixels, based on 47-bit password (secret key) and redundant BCH code (4095,47,955). To construct the key sequence C, the comma-free code, given in Table 3 (codeword length $s = 5$), was used. The minimum cyclic Hamming distance for this class of noise-like embedding templates is:

$$\lambda_{2d} = 819 \cdot \min(1910 \cdot 5, 2047 \cdot 1, 4095 \cdot 1) = 1676493.$$

Experiment on watermark embedding was conducted for 15 hyperspectral images of size 4095 × 4095 pixels. Each image comprised 450 spectral channels; the watermark was embedded into all channels.

Algorithms for robust watermark extraction, described earlier, are based on the use of so-called "informed" watermark detector; in fact, during the watermark extraction from the frame we calculate a cross-correlation function of arrays M_1 and M_0, which are already known for detector, and individual spectral channel of the analyzed image. The security of such scheme against intentional attacks depends primarily on the complexity of brute force attack, i.e. against watermark detection without knowing the proper embedding key. Actually, the watermark is vulnerable to such an attack, if the attacker can try all possible key values (and the corresponding values M_1 and M_0) in acceptable time, and select the one that leads to successful watermark detection.

Suppose that an attacker sequentially tries all possible secret embedding keys (for the proposed algorithm their number is 2^{47}), and, based thereon, generates pairs of arrays M_1 and M_0, and uses them to extract the watermark from the selected image. The computational complexity of such an attack is proportional to the value:

$$N_{br} = \left|\tilde{M}\right| / (2 \cdot (\bar{N}_{collis} + 1)),$$

where $\left|\tilde{M}\right|$ is a number of different array pairs (since the pair M_1 and M_0 is generated on the basis of one secret key), $\bar{N}_{collis}$ is the average number of collisions per array pair. In the present case, the value $\bar{N}_{collis}$ should be evaluated experimentally, and the value $\left|\tilde{M}\right|$ is determined by the number of possible secret embedding keys, and equal to 2^{47}.

For the experimental evaluation of the value $\bar{N}_{collis}$, the following experiment was conducted. At the stage of watermark embedding, 10 different "true" secret keys were used, for each key the pair of templates, M_1 and M_0, was generated. Each pair of templates was used for watermarking the separate hyperspectral image, comprising 450 spectral channels. After watermarking, for each image we ran the brute force attack, i.e. we sequentially tried to extract the watermark using all possible embedding keys. As noted earlier, the number of such "false" keys is $(2^{47} - 1)$, what makes it almost impossible to implement the brute force attack during the experiment. Due to this fact, the experiment was simplified as follows: to implement the attack, from all $(2^{47} - 1)$ "false" embedding keys we used only 5000 template pairs M_1 and M_0, which are the "closest" to the original pair by the minimum cyclic Hamming distance criterion. Next, we evaluated the number of collisions under the assumption, that the vast majority of collisions occurred during watermark extraction, is represented by these 5000 "closest" templates.

After that, for all 10 generated embedding keys, the average number of collisions was calculated:

$$\bar{N}_{collis} = \frac{1}{10} \sum_{j=1}^{10} N_j,$$

where N_j is the number of detected collisions for j^{th} "true" embedding key.

The results of the conducted experiment show, that, after 50000 attempts to extract the watermark using the "false" keys, no collision was found, i.e. the value $\bar{N}_{collis} = 0$ was obtained. Accordingly, the estimation of the computational complexity of brute force attack takes the form of:

$$N_{br} = \left|\tilde{M}\right| / 2 = 2^{47} / 2 \approx 10^{14}.$$

Such computational complexity of brute force attack unambiguously allows us to speak about the ineffectiveness of such attacks against the proposed algorithm. In comparison, the above mentioned watermarking algorithms based on m-sequences [51, 52] require only $5 \cdot 10^4$ attempts to extract the watermark for the implementation of such attack.

By increasing the capacity of the set of all possible steganographic keys, the developed algorithms and methods allow to make the watermark much more robust against its extraction without knowing the embedding key K. Thus, the average complexity of finding the correct steganographic key by the attacker was increased from 10^5 (100000) iterations for existing algorithms to 10^{14} iterations for developed modified algorithm with the same parameters of watermark embedding and generation.

3. Methods for Passive Protection of HSI Data

Unified digital formats for storing and processing visual information, including remote sensing data, became widespread in recent decades. Consequently, such information became much more vulnerable to intentional distortion (falsification) and unauthorized use (in particular copying and distribution).

Unlike active methods based on the use of digital watermarks, passive approach to detecting unauthorized modifications does not imply any pre-distortion of hyperspectral image (i.e. digital watermarks are not used). It is based on the assumption, that even if the altered (falsified) image does not contain visually detectable traces of modifications, these modifications can be detected by the image feature analysis.

Passive methods for digital hyperspectral image authentication can be conventionally divided into 5 following groups [19]:

1) methods based on detecting changes in the image pixel level (Pixel-based) [53, 54];

2) methods aimed at detecting changes of interpixel correlations, introduced, for instance, by using lossy compression algorithms (Format-based) [55, 56];

3) methods for determining various artifacts of a photographing apparatus, introduced by a camera lens, a sensor, or a built-in post-processing algorithm (Camera-based) [57];

4) methods for detecting inconsistencies in mutual arrangement of physical objects, light sources and camera (e.g. different illumination of objects, mismatch between objects and their shadows) (Physically-based);

5) methods based on detecting mismatches between geometrical characteristics of real objects and digital image objects (e.g. inconsistencies in geometric dimensions of objects with respect to each other or to the camera) (Geometric-based).

Next, we consider the basic ways of hyperspectral image falsification, and the corresponding passive security methods. For each of selected directions we will analyze the existing methods and algorithms for detection, and then we will distinguish the direction of our research.

<u>*3.1. HSI Protection Against Introducing Distortion of "Resampling" Type*</u>

Any geometric transformation (resampling) is implemented by means of interpolation algorithm (e.g. bicubic), for which an output pixel value is formed as a weighted sum of values of neighboring pixels on a fragment (before interpolation). In this case the correlation between neighboring pixels of the resampled fragment significantly increases [58], i.e. the value of each pixel depends on its environment. With this in mind, we developed an algorithm for resampled area detection on the basis of Expectation Maximization (EM) algorithm.

In order to solve the denoted problem, we developed an algorithm for distinguishing resampled areas, wherein the analyzed hyperspectral image is divided into blocks, and for each block the correlation between pixels is evaluated.

For the algorithm implementation, two hypotheses of referring a block hyperpixel to one of two classes are introduced: the first class M1 contains pixels which correlate with the neighboring, the second class M2 contains the rest pixels. According to Bayes' rule, for each pixel the probability of getting into M1 or M2 class is calculated. The implementation of this algorithm is an iterative process, wherein with each iteration the dependence coefficients are specified, until the estimated error becomes lower than the allowable (taking into account the partition into classes M1 and M2). From the obtained data, a probability matrix is formed, such that in the resampled areas a periodic structure is observed, and it can be easily detected by Fourier transform. The analysis of the Fourier spectrum of the probability matrix shows, that for the embedded block, bright peaks are well recognizable (Fig. 16), and, generally, they can be used to determine parameters of geometric transformation (scaling factor and rotation angle).

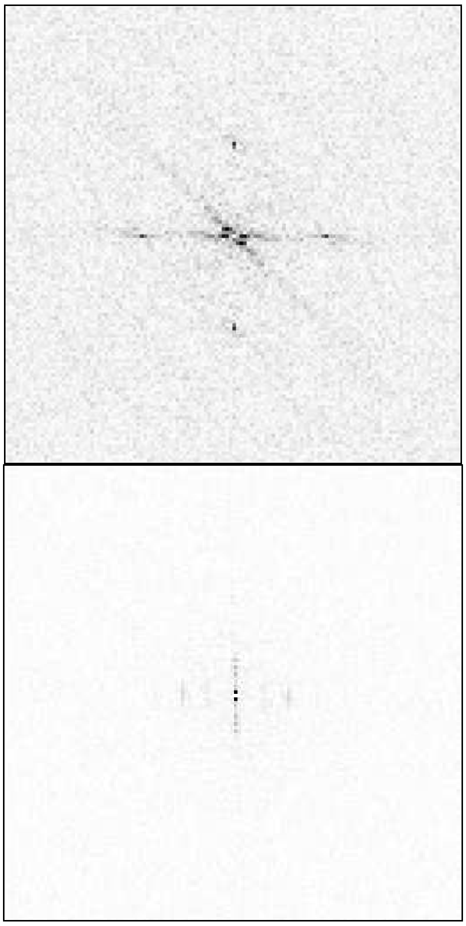

Fig. 16. Fourier spectrum (left - for embedded block, right – for unmodified image block)

In practice, when we use the algorithm for distinguishing image resampled areas, we should consider, that their location is unknown, or there is no such areas at all. Thus, an important step of the algorithm is the localization of resampled areas in the probability matrix. To solve this problem, we propose an algorithm of local spectrum analysis, calculated recursively (which significantly reduces the computational complexity of an algorithm) in a sliding window. Spectrum emissions are recognized via peak filter, and a decision, whether the resampled area exists, is made by comparison with a threshold (determined in the step of algorithm adjustment). Thus, spectrum analysis is conducted for all image blocks taking into account the possible shift, which provides a more precise localization of the resampled area.

The Study of Algorithm for Resampled Area Detection. To study the developed algorithm, we used a hyperspectral image obtained under AVIRIS program. An area containing water texture was embedded into each image channel. Embedded area was subjected to geometric distortions with rotation angle in the range of [0°, 40°], and scaling factor in the range of [0.65; 1.4].

To study the algorithm quality, we used a ratio of the relative area of the detected image fragment to the relative area of the embedded fragment as a criterion:

$$K_i = (S_{out} / S_{in}) \cdot 100\% ,$$

where $i \in \overline{0, C-1}$ is a number of spectral channel.

The value of the criterion for the whole hyperspectral image is calculated as follows:

$$K = (\sum_i K_i) / C .$$

During the quality parameter calculation, a parameter of false detection was also calculated, and the result in all cases was 2-10%. The parameter of false detection was calculated as a ratio of the area of incorrectly detected blocks S_{err} to the total area of the original image S_{img}:

$$K_{err} = (S_{err} / S_{img}) \cdot 100\% .$$

This parameter was also calculated for each channel. According to the dispersion of obtained values, it can be concluded that, in one area each channel is modified (dispersion is close to 0), or something else is embedded.

The following parameter values were used as initial conditions: the size of a processing window for large-format image $WS = 256$, the size of a sliding window $L = 64$ and $\mathrm{L} = 128$, the size of a peak filter mask $Size = 3$.

The experimental results show that the detection quality depends on the sliding window size. If its size increases, the quality of detection increases too (a small number of incorrectly detected blocks), but the quality of detection decreases, if the size of the embedded information is smaller than the size of the window. When the size of the window decreases, the quality of detecting the embedded information of small size increases, but also increases the number of incorrectly detected blocks. The optimal variant determination is a task for future research.

3.2. HSI Protection Against Introducing Distortion of "JPEG Compression" Type

When the hyperspectral image is modified using image processing software, and after resaving this image (or individual spectral channels), global features also change. In this case, methods for detection of (JPEG) compression application can be used to check, whether the image is modified. Well-known algorithms for JPEG tampering detection [56, 59] use only one channel for detection and cannot be applied for hyperspectral imagery.

Since the compression algorithm can be applied to a separate channel, and, thereby, modify its characteristics, the developed algorithm analyzes each channel to detect inconsistencies in characteristics of spectral channels. As a solution to the problem of detecting hyperspectral image fragments with different compression quality, or ratio, we developed DCT coefficient histogram spectrum analysis algorithm, which allows to establish some regularities numerically characterized by feature values calculated via spectra. Thus, the result of applying the JPEG-compression is well seen on histograms of DCT coefficients calculated by image blocks for fixed spectrum components.

By analyzing the histograms of DCT coefficients we can determine, whether JPEG compression was applied for the embedded part of hyperspectral image. In this case, the following situations are possible: JPEG compression was not performed; compression was performed once; compression was repeatedly performed with different quality parameters.

Along with the detection algorithm, we also developed a method of determining the shift of JPEG fragments relative to the embedding coordinates, which are multiples of 8. This method is used as a pre-processing operation for the analyzed image.

The proposed algorithms both perform the task within a unified information technology.

1. The basic shift of the block matrix is calculated. This operation is performed in order to provide the detection of embedded JPEG blocks shifted in relation to the original mesh.

1.1. An array S[8][8] is created, each array element is a so-called JPEG-feature, calculated as follows:

1) a random set of blocks, comprising not less than 1000 blocks, is created;

2) blocks of the resulting set are shifted by Δ_x, Δ_y, and cosine transform is calculated;

3) for all blocks the histogram of cosine transform coefficients is calculated with a fixed shift;

4) histogram spectra are calculated;

5) for obtained spectra, features are calculated;

6) calculated features are converted into one criterion (periodicity criterion), recorded in $S[\Delta_x][\ \Delta_y]$;

7) steps (2) – (7) are repeated.

1.2. The minimum of S array is determined, its indices are the shifts *Shift*[0], *Shift*[1]. Thus, an invariance to the shift of the embedded area is obtained.

2. Blocks are pre-clustered on the basis of the basic shift. At this stage, background image blocks are separated from blocks, which demonstrate any JPEG properties. As a result of this procedure, two clusters are formed: first are the block coordinates without properties of periodicity and monotony, second are, respectively, all the rest, that are divided by quality and ratio of JPEG compression at a later clustering stage.

3. Image blocks are clustered on the basis of the found basic shift. An iterative procedure is repeated until blocks are divided into clusters, or the number of clusters reaches the maximum allowable number.

3.1. We assume, that all image blocks belong to the same cluster.

3.2. For all blocks of each cluster, histograms are constructed (see para. 2).

For each block we verify, whether it belongs to one of the current clusters. If the product of probabilities for given block frequencies is above the threshold, this block belongs to the cluster. If for all existing clusters the value of the product is smaller than the threshold, then a new cluster is created, wherein the block is placed.

The Study of Algorithm for Detection of Fragments Subjected to JPEG Compression.

Table 4 and Fig. 17 show the results of the study on the algorithm for detection of embedded information with JPEG properties. The dimensions of embedded areas with different JPEG compression properties were from 64 × 64 to 512 × 512 when the dimensions of the input hyperspectral images were from 2000 × 2000 to 6000 × 6000.

Table 4. Quality of detection of dual JPEG compression for different combinations of quality parameters

Q2 / Q1	50%	60%	70%	80%	90%
50%	–	100%	100%	100%	90%
60%	100%	–	100%	100%	95%
70%	100%	100%	–	100%	92%
80%	100%	100%	95%	–	80%
90%	95%	95%	95%	75%	–

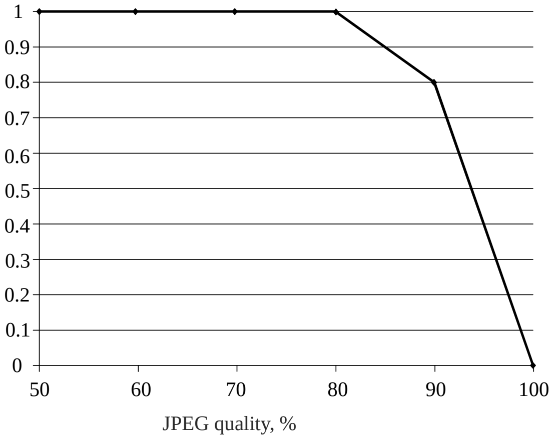

Fig. 17 - Quality of detection of single JPEG compression

The data, presented in the table and figures, are the average over the set of 5 hyperspectral images, obtained under the AVIRIS program. According to the data, the detection quality is very close to 1 in most cases, which indicates a high accuracy of the developed algorithm.

Also, during the study, we obtained the relationship between the quality of single JPEG compression and the values of peak periods of the DCT coefficient spectrum. The results are shown in Fig. 18. If we calculate the period via the DCT coefficient histogram spectrum by using this diagram, we can obtain the initial value of the JPEG compression quality.

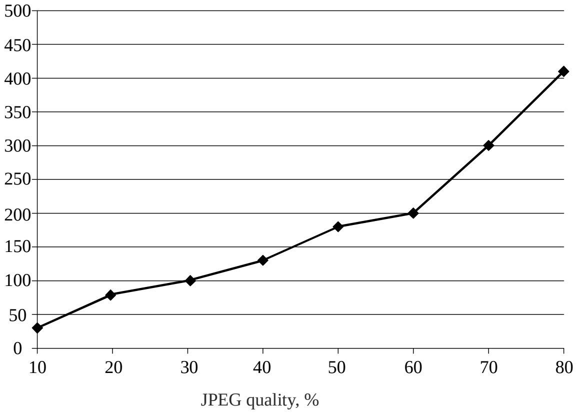

Fig. 18. The relationship between JPEG compression quality and values of DCT coefficient spectrum peak periods

3.3. HSI Protection Against Introducing Distortion of "Undistorted Duplicate" Type

The most common way to counterfeit a hyperspectral image is to copy some image areas to hide an object. In this case, an image part is copied, and pasted into another part of the image in the place of the object, which is expected to be hidden (Fig. 19).

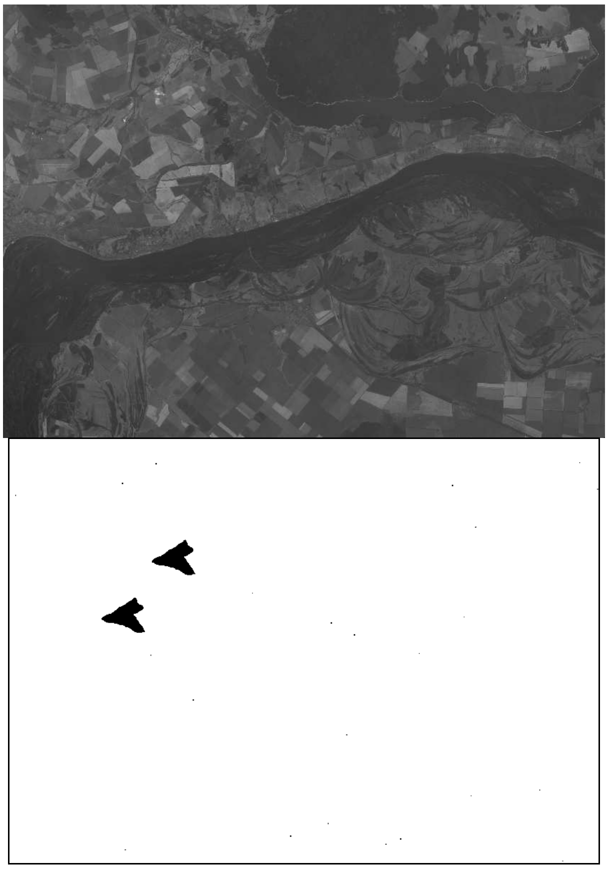

Fig. 19. Example of an image containing a duplicate, and the attack detection result

Such operation is performed for each spectral channel to hide the traces of modifications. If this is done carefully, even an expert will not be able to recognize a counterfeit and determine the modified area. To try all the possible combinations of dimensions of modifiable regions, and their location, is a time-consuming task, and for large-format images this problem cannot be solved at all.

The most well-known duplicate detection algorithms [53, 54] are based on block-based discrete cosine transform and principal component analysis. Duplicate areas are detected using lexicographical sorting of feature vectors, which consist of DCT coefficients, and the subsequent grouping of blocks with the same spatial shift.

In this paper, we propose to analyze all the image fragments of size $a \times b$ sequentially – in the so-called "sliding window" ("processing window") mode. For each position of a "sliding window" a value of hash-function (deterministic algorithm, which converts an input data array of an arbitrary length into an output bit string of a fixed length) is calculated using the corresponding image counts, and placed into a data structure, wherein each hash value corresponds to the number of occurrences of such values (and correspondingly, fragments) on the image, called the hash table. The values of the hash table, which exceed "1", correspond to the values of the hash function, which are derived from the potential duplicates (their position on the image is specified during the second run through the image). In this paper, the computational complexity of such a solution (analysis of all the image fragments) is mitigated by the method of construction and implementation of the hash function calculation (since the algorithm generalization to the case of using several hash functions is trivial, it is not considered).

Thus, instead of pairwise comparison of all possible fragments we propose to detect matching fragments using hash table.

We developed an algorithm for undistorted duplicate detection, which is based on the use of hash functions of analyzed areas. We used the following hash functions:

– bit projection (selection of fragment pixel bits in a way to comply with a condition of physical realizability of hash values);

– modular representation of the image fragment corresponding to a rectangular template.

Construction of algorithm for undistorted duplicate detection is based on the selection of the structural element, under which bits for hash generation are selected. First, an array for storing the resulting field $t(i,j) \in \mathbf{B}$ is formed. In the first step, for $r=0$ the hash table of hash values $H_0(m,n,f)$ is constructed: image bit positions, used to derive the hash value, are calculated under the selected structural element. Along with filling the hash table, the field $t(i,j)$ is filled with values. In the next step, for $r=1$ we analyze only those positions of the processing window, where duplicates may be located, i.e. in view of calculated values $t(i,j)$ for $r=0$. At each iteration of the algorithm, the number of false duplicate detections (collisions) decreases. The iterative process stops, when the number of collisions on r^{th} iteration coincides with the number of collisions on $(r-1)^{\text{th}}$ iteration, or when they are not found.

The Study of Algorithm for Undistorted Duplicate Detection. The key parameters of the proposed hash functions are:

– a maximum value (a hash table length $T=2^{k-1}$);

– processing window parameters a and b.

A quality indicator for hash functions and for the algorithm (for a specific set of images) is the number of wrongly detected duplicates – collisions (K). Diagrams show the relative number of collisions, i.e. $\chi = K / MN$.

For the experiments we used 3 hyperspectral images without duplicates, obtained under AVIRIS program. To perform calculations we used a standard PC (Core 2 Quad Q8300, 4 Gb RAM) with 64-bit Windows 8 OS.

For simplicity, we will denote the hash function based on the bit projection – 1 or ♦, and based on the modular representation – 2 or ■.

Figure 20 shows the dependence of χ on parameters $a \times b$ of sliding processing window. As can be seen from the diagram, hash function 2 demonstrates the best result.

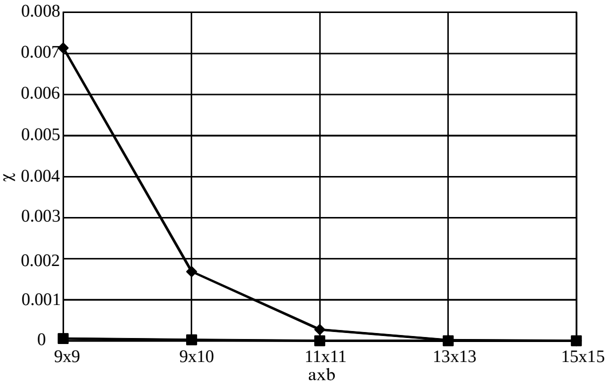

Fig. 20. The relationship between the relative number of collisions and parameters a × b

Figure 21 shows the dependence of χ on the number of bits k, used to represent hash values. As can be seen from the diagram, for all hash values χ has a tendency to increase with the decrease in k. It should be noted that for the hash function 2 the number of collisions increases slower than for the other two.

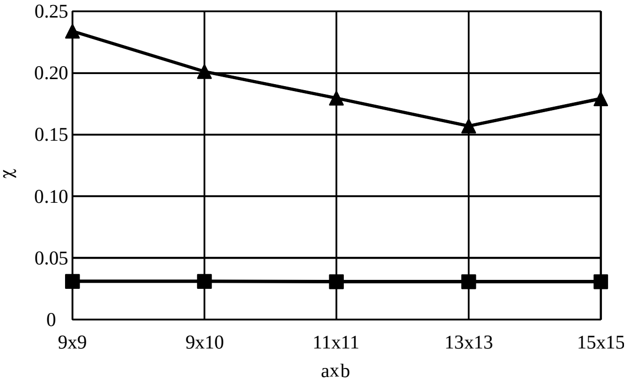

Fig. 21. The relationship between the relative number of collisions and the number of hash value bits

3.4 HSI Protection Against Introducing Distortion of "Geometrically Distorted Duplicate" Type

The existing algorithms for detecting geometrically transformed duplicates [55, 56, 60] are based on partitioning the analyzed image into blocks (usually non-overlapping), and their pair-wise comparison. Instead of block pixels, secondary features calculated by blocks that are invariant to transformations are commonly used when comparing the blocks. The proximity of two block feature vectors under a certain criteria indicates the similarity of two blocks, which may be a consequence of introduced duplicates.

A lot of problems occur when using this approach, the major ones are.

1. Block size selection. For a small block size the number of required block comparisons is unacceptably high for practical implementation of the algorithm. For a large block size, the block may appear to be greater than the duplicated area, which will lead to the impossibility of its detection.

2. Choosing the scheme for partitioning the image into blocks. When using a partition without overlapping, duplicates located on the block boundary can stay undetected. The use of overlapping leads to a sharp increase in the number of blocks.

3. Comparison algorithm selection. A direct comparison of each block pair in spatial or spectral domain is unacceptable from the computational standpoint. An alternative way is to use for comparison a small (compared to the number of pixels in the block), number of features.

4. Selection of features, invariant to possible distortions of the duplicated image fragment.

5. Selection of a decision rule for determining the identity of blocks, and determination of its parameters, i.e. parametric adjustment via existing samples of images with duplicates.

For reliable detection of duplicated areas, the size of the analyzed block should not exceed the size of the duplicate (otherwise, the block will "capture" unduplicated background, i.e. the original block and duplicates will differ). At the same time a significant decrease in the block size (to several dozen pixels) will lead to a huge number of false similar areas detected pairwise, and will, eventually, make the duplicate detection much more difficult. The block size should be smaller than the area we want to detect, to ensure that for a minimum shift (by 1 pixel) of the analyzed block, the block is in the duplicated image area.

Since the duplicated area form is a priori uncertain, it would be optimal to use the block in the form of a circle. However, the use of fast spectral or recursive algorithms for feature calculation makes it appropriate to use square blocks. Moreover, the use of such algorithms (e.g. fast Fourier transform) makes it appropriate to use block sizes, which are powers of two.

Speaking about the partitioning scheme, it should be noted that in case of simple image partition into non-overlapping blocks for subsequent comparison, a duplicate can appear in several blocks, covering only a part of each block, and, therefore, remain undetected.

The most reliable, in terms of detection quality, way is to compare each block with all possible (overlapping) blocks.

Taking into account the problems stated above, we developed an algorithm for detecting geometrically distorted duplicates on hyperspectral images with the use of features invariant to affine transformations. We used Fourier-Mellin transform coefficients as features.

The proposed information technology is based on the following principles, which provide the problem optimization in "solution reliability - computational complexity" coordinates:

– the use of Fourier-Mellin transform for calculating features used for the block comparison (these features are invariant to rotation and scaling of the image fragment; the invariance to linear brightness transformations is easily achieved by image block preprocessing);

– the use of scheme with block overlapping (which improves algorithm reliability), and the sliding window mode for recursive computation of the Fourier spectrum (which reduces the computational complexity), for the duplicate detection.

The Study of Algorithm for Geometrically Distorted Duplicate Detection. During the study on the sensitivity of the developed algorithm to changes in the angle of rotation of duplicate embedding area, the results shown in Fig. 22 were obtained.

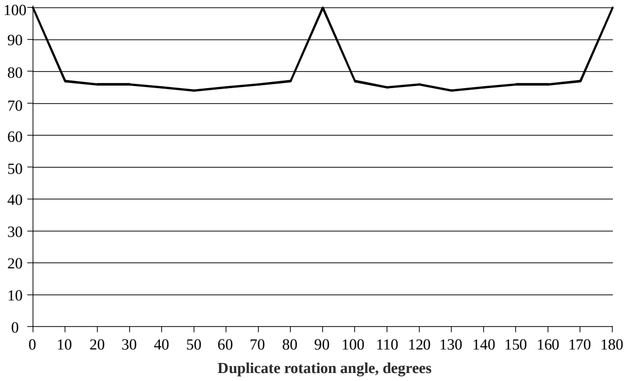

Fig. 22. The dependence of the detection quality on the rotation angle

As a quality criterion we selected the same criterion as for the study of resampling detection algorithm.

As can be seen from Fig. 22, the developed algorithm demonstrates a high precision invariance to the rotation angle, since during the feature formation in a sliding window, their values are calculated via the inscribed circle of the window to avoid redundant pixels, which contribute substantial error, when features are formed via the circumscribed circle or via the sliding window. Such an approach yields better results in comparison with the existing results, demonstrating the successful detection of geometrically transformed duplicates, subjected to rotation by an angle not exceeding 15°.

Conclusion

In this paper, methods of compression and protection of hyperspectral remote sensing images, are proposed. A general structure of the method for compression and protection of hyperspectral data, algorithms for speed stabilization of the compressed data stream generation, algorithms for noise immunity enhancement, and spectral component approximation algorithms, that are well adapted for the use of HGI-compression when solving hyperspectral image storage problem, are developed. We evaluated the algorithm effectiveness, and also compared these HGI-compression algorithms with previously developed, using the real 16-bit images made by hyperspectrometers. The prospects of using the HGI-compression for hyperspectral image storage problem solution are shown.

For data protection against unauthorized copying and modification, algorithms for digital watermark embedding and extraction, based on spread spectrum modulation, are proposed. The proposed algorithms for robust digital watermarking preserve embedded information in case of typical data conversion operations: compression, filtering, geometric correction, allocation of fragments and two-dimensional sections of hyperspectral images.

In the paper we have shown that, in comparison with existing analogues, the developed methods are much more robust against watermark extraction by means of brute force attack on the steganographic key. Thus, the average complexity of finding the correct steganographic key by the attacker increased from 10^5 iterations for existing protection algorithms to 10^{14} iterations for developed algorithms.

For passive HSI protection we developed algorithms for detection of four basic attack types: resampling, attack based on JPEG compression, undistorted duplicate embedding, and geometrically distorted duplicate embedding. The proposed methods demonstrate high quality of data distortion detection and low computational complexity. Experimental studies have shown the advantage of the developed algorithms over existing solutions.

Implementation of the proposed methods and algorithms in systems of remote sensing data formation, storage, and transmission over communication channels, is an important direction in security of information systems related to processing and analysis of visual information.

References

[1] Chang C. Hyperspectral Data Processing: Algorithm Design and Analysis. Wiley Press; 2013.

[2] Schowengerdt RA. Remote Sensing – Models and Methods for Image Processing. New York: Academic Press; 1997.

[3] Chang C. Hyperspectral imaging: techniques for spectral detection and classification. Springer; 2003.

[4] Borengasser M., Hungate W, Watkins R. Hyperspectral Remote Sensing – Principles and Applications. CRC Press; 2004.

[5] Chang C. Hyperspectral data exploitation: theory and applications. Wiley-Interscience; 2007.

[6] Gashnikov MV, Glumov NI, Myasnikov VV, Chernov AV, Ivanova EV. Regional Geographic Information Systems for Gas Network Monitoring. Pattern Recognition and Image Analysis 2015; 25(3): 418–422. DOI: 10.1134/S1054661815030062

[7] Chanussot J, Crawford M., Kuo B. Foreword to the Special Issue on Hyperspectral Image and Signal Processing. IEEE Transactions on Geoscience and Remote Sensing 2010; 48(11): 3871−387.

[8] Chang C, Chiang S. Anomaly detection and classification for hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing 2002; 40(6): 1314-1325.

[9] Benz U, Hofmann P, Willhauck G, Lingenfelder I, Heynen M. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. ISPRS Journal of Photogrammetry and Remote Sensing 2004; 58(3): 239-258.

[10] Gashnikov MV, Glumov NI. Hierarchical compression for hyperspectral image storage [In Russian]. Computer Optics 2014; 38(3): 482-488.

[11] Gashnikov MV, Glumov NI. Hyperspectral images repository using a hierarchical compression. 23-rd International Conference on Computer Graphics, Visualization and Computer Vision proceeding 2015; 1-4. ISBN 978-80-86943-67-1. ISSN 2464–4617.

[12] Salomon D. Data Compression. The Complete Reference. Springer-Verlag, 4ed; 2007.

[13] Vatolin D, Ratushnyak A, Smirnov M, Yukin V. Data compression methods. Archive program architecture, image and video compression [In Russian]. Moscow: DIALOG-MIFI; 2002.

[14] Pratt W. Digital image processing. Wiley, 4ed; 2007.

[15] Soifer VA, Chernov AV, Chernov VM, Chicheva MA, Fursov VA, Gashnikov MV, Glumov NI, Ilyasova NY, Khramov AG, Korepanov AO, Kupriyanov AV, Myasnikov EV, Myasnikov VV, Popov SB, Sergeyev VV. Computer Image Processing, Part II: Methods and algorithms. Ed by Soifer VA. VDM Verlag; 2010.

[16] Gashnikov MV. Parameterization of nonlinear Greham predictor for digital image compression [In Russian]. Computer Optics 2016; 40(2): 225-231. DOI: 10.18287/2412 -6179-2016-40-2-225-231.

[17] Woods E, Gonzalez R. Digital Image Processing. Prentice Hall, 3ed; 2007.

[18] Wallace G. The JPEG Still Picture Compression Standard. Communications of the ACM 1991; 34(4): 30-44.

[19] Sridevi M, Mala C., Sanyam S. Comparative study of image forgery and copy-move techniques. Second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012). New Delhi, India. – 2012. P. 715-723.

[20] Lossless Multispectral & Hyperspectral Image Compression. Recommendation for Space Data System Standards, CCSDS 123.0-B-1. Blue Book; 1. Washington, D.C.: CCSDS, 2012.

[21] Nian Y, He M, Wan J. Lossless and near-lossless compression of hyperspectral images based on distributed source coding. Journal of Visual Communication and Image Representation 2015; 28: 113–119.

[22] Valsesia D, Magli E. A novel rate control algorithm for onboard predictive coding of multispectral and hyperspectral images. IEEE Trans. Geosci. Remote Sens. 2014; 52(10): 6341–6355.

[23] Multispectral Hyperspectral Data Compression Working Group. Source: 〈http://cwe.ccsds.org/sls/default.aspx〉.

[24] Consultative Committee for Space Data Systems (CCSDS). Source: 〈http://www.ccsds.org〉.

[25] Gashnikov MV, Glumov NI, Sergeyev VV. The image compression method in real-time remote sensing [In Russian]. 9th All-Russian conference "Mathematical methods of pattern recognition" 1999 (Moscow); 160-163.

[26] Gashnikov MV, Glumov NI, Sergeyev VV. Compression Method for Real-Time Systems of Remote Sensing. 15th International Conference on Pattern Recognition 2000 (Barselona); 3: 232-235.

[27] Gashnikov MV, Glumov NI. Hierarchical grid interpolation for hyperspectral image compression [In Russian]. Computer Optics 2014; 38(1): 87-93. ISSN 0134-2452.

[28] Gashnikov MV, Glumov NI. Hierarchical GRID Interpolation under Hyperspectral Images Compression. Optical

Memory and Neural Networks (Information Optics) 2014; 23(4): 246-253. ISSN 1060-992X.

[29] Gashnikov MV, Glumov NI, Sergeev VV. A hierarchical compression method for space images. Automation and Remote Control 2010; 71(3): 501-513. ISSN: 0005-1179.

[30] Gashnikov MV, Glumov NI. Onboard processing of hyperspectral data in the remote sensing systems based on hierarchical compression. Computer Optics 2016; 40(4): 543-551. DOI: 10.18287/2412-6179-2016-40-4-543-551.

[31] Glumov NI. Improving Noise Immunity of Transmission of Compressed Digital Images. Pattern Recognition and Image Analysis, 2003; 13(2): 273-276.

[32] Lin S, Costello D. Error Control Coding: Fundamentals and Applications, second edition. New Jersey: Prentice-Hall, inc. Englewood Cliffs; 2004.

[33] SpecTIR Data – Advanced Hyperspectral and Geospatial Solutions. Corporate Headquarters SpecTIR Remote Sensing. Source: 〈Division // http://www.spectir.com/free-data-samples〉.

[34] AVIRIS Data – Ordering Free AVIRIS Standard Data Products. Jet Propulsion Laboratory. Source: 〈http://aviris.jpl.nasa.gov/data/free_data.html〉.

[35] Kbaier I, Belhadj Z. A novel content preserving watermarking scheme for multipectral images. Information and Communication Technologies (ICTTA'06, 2nd) 2006; 1: 322-327.

[36] Minguillón J. Evaluation of copyright protection schemes for hyperspectral imaging. Remote Sensing. International Society for Optics and Photonics 2004: 512-523

[37] Jing L, Zhang Y, Chen G. Zero-watermarking for copyright protection of remote sensing image. Signal Processing (ICSP 2008. 9th International Conference on) 2008: 1083-1086.

[38] Wang X, Guan Z, Wu C. A novel information hiding technique for remote sensing image. Advanced Data Mining and Applications 2005: 423-430.

[39] Kaarna A, Toivanen P. Digital watermarking of spectral images in PCA/Wavelet-transform domain. Geoscience and Remote Sensing Symposium (IGARSS'03) Proceedings 2003; 6: 3564-3567.

[40] Kaarna A, Parkkinen J. Digital watermarking of spectral images with three-dimensional wavelet transform. Image Analysis 2003: 320-327

[41] Melgani F, Benzid R, De Natale F. Near-lossless spread spectrum watermarking for multispectral remote sensing images. Journal of applied remote sensing 2007; 1(1): 3501-3517.

[42] Panyavaraporn J, Rangsanseri Y. Digital Image-in-Image Watermarking of Remote Sensing Images. 26th Asian Conference on Remote Sensing and 2nd Asian Space Conference ACRS2005 2005; 1: 1145-1150.

[43] Barni MF, Bartolinib F, Cappellinib V, Maglic E, Olmoc G. Near-lossless digital watermarking for copyright protection of remote sensing images. Geoscience and Remote Sensing Symposium 2002; 3: 1447-1449.

[44] Doërr G, Dugelay JL. Security pitfalls of frame-by-frame approaches to video watermarking. IEEE Trans-actions on Signal Processing 2004; 52(10): 2955-2964.

[45] Delp EJ, Lin ET. A review of fragile image watermarks. Proc. ACM Multimedia and Security Workshop 1999; 1: 25–29.

[46] Tirkel AZ, Hall TE. A unique watermark for every image. IEEE Multimedia 2001; 8(4): 30-37.

[47] Van Schyndel RG, Tirkel AZ, Svalbe ID. Key independent watermark detection. IEEE International Conference on Multimedia Computing and Systems 1999; 1: 580-585.

[48] Van Schyndel RG, Tirkel AZ, Svalbe ID, Hall TE, Osborne CF. Spread-Spectrum Digital Watermarking Concepts and Higher Dimensional Array Constructions. First International Online Symposium on Electronics Engineering 2000: 1-13.

[49] Van Schyndel RG, Tirkel AZ, Svalbe ID, Hall TE, Osborne CF. Algebraic construction of a new class of quasi-orthogonal arrays for steganography. Proc. SPIE 3657 1999: 354-364.

[50] Chen L, Gong G. Communication system security. CRC press; 2012.

[51] Mitekin VA, Timbay EI. A new watermarking sequence generation algorithm for collision-free digital watermarking. Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP) Eighth International Conference on 2012: 256-260.

[52] Mitekin VA, Fedoseev VA. A new method for high-capacity information hiding in video robust against temporal desynchronization. Seventh International Conference on Machine Vision (ICMV 2014), International Society for Optics and Photonics 2015: 94451A-94451A.

[53] Mahdian B, Saic S. A bibliography on blind methods for identifying image forgery. Signal Processing: Image Communication 2010: 389-399.

[54] Fridrich J, Soukal D, Lukas J. Detection of copy–move forgery in digital images. Proceedings of Digital Forensic Research Workshop, Cleveland 2003: 55-61.

[55] Mahdian B, Saic S. A cyclostationarity analysis applied to image forensics. IEEE Workshop on Applications of Computer Vision (IEEE WACV), Snowbird 2009: 1-6.

[56] Farid H. Exposing digital forgeries from JPEG ghosts. IEEE Transactions on Information Forensics and Security 2009; 1(4): 154-160.

[57] Ng TT. Camera response function signature for digital forensics - part II: signature extraction. IEEE Workshop on Information Forensics and Security 2009: 161-165.

[58] Popescu AC, Farid H. Exposing digital forgeries by detecting traces of re-sampling. IEEE Transactions on Signal Processing 2005; 53(2): 758-767.

[59] Poilpre MC, Perrot P, Talbot H. Image tampering detection using Bayer interpolation and JPEG compression. Proceedings of the 1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia Workshop 2008: 1-5.

[60] Bayram S, Sencar HT, Memon N. A survey of copy-move forgery detection techniques. Proceedings of the IEEE Western New York Image Processing Workshop 2009: 538-542.

Authors' information

Mikhael Valeryevich Gashnikov (b. 1975) graduated from S.P. Korolyov Samara State Aerospace University (SSAU), holds a candidate's degree in Engineering. Currently he is an associate professor at the Geoinformatics and Information Security sub-department at Samara National Research University. He has 80 scientific publications, including 30 scientific papers and 2 monographs (with coauthors). His research interests currently focus on image compression, space image processing, and geoinformation technologies. E-mail: *mgash@geosamara.ru* .

Nikolay Ivanovich Glumov (b. 1962) graduated with honours (1985) from S.P. Korolyov Kuibyshev Aviation Institute. He received his Candidate in Technics (1994) degree from Samara State Aerospace University (SSAU). He is the head of research laboratory of advanced technologies of remote sensing at Samara National Research University and the senior researcher at the Image Processing Systems Institute of RAS – Branch of the FSRC "Crystallography and Photonics" RAS. His current research interests include image processing and pattern recognition, images compression, digital images forming systems modelling. He has more than 100 publications, including more than 40 scientific papers, 2 monographs (in co-authorship). E-mail: *nglu@geosamara.ru* .

Andrey Vladimirovich Kuznetsov (b. 1987) graduated with honours (2010) from Samara State Aerospace University (SSAU), majoring in Applied Mathematics and Informatics. He studied as a post-graduate student at SSAU from 2010 and received his PhD in technical sciences in 2013. Nowadays he is a senior researcher at laboratory of advanced technologies of remote sensing at Samara National Research University and a researcher at Image Processing Systems Institute of the RAS– Branch of the FSRC "Crystallography and Photonics". His research interests are currently focused on image processing and analysis, pattern recognition, digital image forgery detection, geoinformatics. He has 37 publications, including 18 scientific papers and 1 monograph (with coauthors). E-mail: *kuznetsoff.andrey@gmail.com*. Web-page: *http://nil97.ssau.ru/employee/detail.php?ID=35*

Vitaly Anatolyevich Mitekin (b. 1983) graduated from S.P. Korolyov Samara State Aerospace University (SSAU), majoring in Applied Mathematics and Informatics in 2006. He received his Candidate in Technical Sciences degree from Samara State Aerospace University in 2009. Currently he is a senior researcher at the laboratory of advanced technologies of remote sensing at Samara National Research University and a researcher at Image Processing Systems Institute of the RAS– Branch of the FSRC "Crystallography and Photonics". His scientific interests include image processing and recognition, steganography and steganalysis, cryptography. Email: *vmitekin@gmail.com* .

Vladislav Valerievich Myasnikov (b.1971), graduated (1994) from the S.P. Korolyov Samara State Aerospace University (SSAU). He received his PhD in Technical sciences (2002) and DrSc degree in Physics & Maths (2008). He worked in Image Processing Systems Institute of RAS and SSAU. At present, he is a leading researcher at the Samara University. The area of interests includes digital signals and image processing, geoinformatics, neural networks, computer vision, pattern recognition and artificial intelligence. He's list of publications contains about 200 scientific papers, including 100 articles and 2 monographs. He is a member of Russian Association of Pattern Recognition and Image Analysis. Email: *vmyas@geosamara.ru* . Website: *http://www.ipsi.smr.ru/staff/MyasVV.htm*

Vladislav Victorovich Sergeyev (1951 b.), graduated (1974) from S.P. Korolyov Kuibyshev Aviation Institute (presently, Samara National Research University (or shortly, Samara University)). He received his Candidate's degree in Technical Sciences in 1978 and DrSc degree in Mathematics and Physics in 1993. At present, he is the head of Geoinformation Science and Information Security sub-department at Samara University, also holding a part-time position as the head of a laboratory at the Image Processing Systems Institute of the RAS – Branch of the FSRC "Crystallography and Photonics" RAS. The areas of research interests include digital signal and image processing, geoinformatics and pattern recognition. Email: *vserg@geosamara.ru* .

Received October 6, 2016. The final version – October 25, 2016.