Digital video watermarking method based on heteroassociative image compression and its implementation by artificial neural networks

M A Dryuchenko, A A Sirota
Computer Science Department, Voronezh State University, Voronezh, Russia
E-mail: m_dryuchenko@mail.ru

Abstract. In this paper, we consider video steganography method based on using heteroassociative compressive transformation to embed user information into fragments of video frames. The results of testing the method in relation to visual distortion evaluation and accuracy of hidden data extraction using a number of known video codecs for compressing watermarked videos are presented.

1. Introduction
The global spread and improvement of telecommunication systems necessitates the development of new techniques and means of confidentiality and data integrity provision, copyright protection, monitoring unauthorized use and distribution of data. The most promising technologies used to solve these issues are computer and digital steganography ones for protected hidden storage and transmission of information via open communication channels, as well as hidden robust or fragile watermarking of digital content objects with any user information. Digital multimedia objects are frequently used as a storage media of secret information (containers), having psycho-visual redundancy, i.e. images, audio and video streams. The latter objects are of particular interest when new methods and algorithms of steganographic data hiding are developed and studied, because of widespread use and popularity of video content and failure to find the best information hiding algorithms into video.

The known steganographic data hiding algorithms into video usually embed data into the spatial, temporal and frequency domain of the containers. The algorithms, dealing with frames spatial representation, often realize LSB or PVD methods [1] in regard to raster representation of separate frames. A major shortcoming of such algorithms is weak sustainability of hidden data to further container compression. Some other ways of steganographic data hiding into video, providing enhanced robustness of hidden data, but dependent on used coding algorithms (MPEG-4, H.264/AVC), are steganographic concealment directly into the compressed data flow at the level of spectral coefficients [2, 3], as well as embedding data using information of movement vectors [4, 5].

The aim of this paper is to justify steganographic data hiding method based on general compression transformations carried out at arbitrarily-shaped image fragments (video frames).
using feedforward neural networks [6, 7]. The specifics of the proposed method is that we suggest while steganographic data hiding to use contraction mutual mapping of two neighbouring image areas of arbitrary shape, i.e. heteroassociative compression. This approach helps to reveal significant connections for various image fragments. Autoassociative compression is a more common variant based on compressing single areas and is a sub-category of the latter one. This watermarking method of video frequencies frames does not depend on the final video format and certain codec types, gives minimum distortion and has adjusting persistence. The method includes the work with separate video frequencies frames as static images and the process of steganographic embedding and retrieval of information can be presented as

\[
\hat{Z} = F_1(Z, d, k), \|\hat{z}^{(i)} - \hat{z}^{(i)}\| \rightarrow \min, i = 1, n, \hat{Z}_c = C(\hat{Z}), d = F_2(\hat{Z}_c, k), \|d - \hat{d}\| \rightarrow \min,
\]

where \(Z = \{z^{(0)}, z^{(1)}, \ldots, z^{(n)}\}, \hat{Z} = \{\hat{z}^{(0)}, \hat{z}^{(1)}, \ldots, \hat{z}^{(n)}\}\) – are a variety of frames of empty and full video container; \(\hat{Z}_c\) – a full video container compressed via C codec; \(d, \hat{d}\) – an original and decoded message; \(k\) – steganographic key. Let us consider the process of \(F_1\) encoding and \(F_2\) decoding transformations more closely.

2. Theoretical justification of the heteroassociative compression method

Let the initial model of the image (video frame) be its random field realization defined on an octagonal discrete grid \(w(x, y), \Psi = \{x = \Gamma, n, y = \Gamma, m\}\), where \(w(x, y) \in \mathbb{R}^1\) for monochromatic images and \(w(x, y) \in \mathbb{R}^2\) for colour images. Let \(z \in \mathbb{R}^N, N = N_1 + N_2\) be a random vector of a certain area of a random field \(\Omega \subset \Psi\) determined by unwrapping \(w(x, y), (x, y) \in \Omega\) in random order. To make it clear, let the mathematical expectation and covariance matrix of vector \(z\): \(M[z] = 0, M[z z^T] = R_z\). When heteroassociative compression is applied, vector \(z\) can always be represented by a composite vector \(z = (z_1^T, z_2^T)^T\), where \(z \in \mathbb{R}^{N_1}\) maps a subarea of fragment \(\Omega_I\), called the input, and \(z_2 \in \mathbb{R}^{N_2}\) maps a subarea called the output \(\Omega_O\). In this case, the input data is transformed into the output data, with \(\Omega_I \cup \Omega_O = \Omega, \Omega_I \cap \Omega_O = \emptyset\). When autoassociative compression is applied, \(z_1 = z_2 = z\) coincides with \(\Omega_I = \Omega, \Omega_O = \Omega\).

Let us assume that the number of such non-overlapping areas covers \(\Psi; \bigcup_{p=1}^{P} \Omega^{(p)} = \Psi, \bigcap_{p=1}^{P} \Omega^{(p)} = \emptyset\). Hence, each area \(\Omega^{(p)}\) corresponds to the realization of vector \(z: z^{(p)}\). Then the initial image (video frame) can be presented as \(Z = \{z^{(p)}, p = 1, P\}\). We can thus determine a set of realizations \(\{z_1^{(p)}, z_2^{(p)}\}, p = 1, P\) of the input and the output of fragments \(\{\Omega^{(p)}, p = 1, P\}\), which will be applied as a training set for the heteroassociative artificial neural network (ANN) used for compression.

For universal heteroassociative compression, areas \(\Omega_I, \Omega_O\) may be of arbitrary configuration. Fragments of different configurations were considered by the authors in [8]. In this work the input areas \(\Omega_I\) were defined as octagonal areas placed inside the octagonal areas \(\Omega\). And \(\Omega_O\) were defined as a local set of bounding pixels of \(\Omega_I\).

Random vectors \(z_1, z_2\) are generally connected by the following relation

\[
z_2 = \hat{z}_2_{1/1} + V = H z_1 + V, \quad H = R_{z21} R_{z11}^{-1}, \quad M[V] = 0, \quad M[VV^T] = R_{z22} - R_{z21} R_{z11}^{-1} R_{z12}, \quad (1)
\]

where \(\hat{z}_2_{1/1}\) acts as an optimal linear estimator \(z_2\) in respect to the observed \(z_1\); \(V\) is the stochastic component uncorrelated with \(\hat{z}_2_{1/1}; R_{z11} = M[z_1 z_1^T], R_{z22} = M[z_2 z_2^T], R_{z21} = M[z_2 z_1^T]\).

Let us now consider the architecture of universal data converters that can be used for heteroassociative compression. As we know, for autoassociative compression, feedforward ANN

Let us now consider the architecture of universal data converters that can be used for heteroassociative compression. As we know, for autoassociative compression, feedforward ANN
are used as such converters. The weighting coefficients of such networks can be either calculated or set by backpropagation training [9, 10].

Figure 1 shows a standard architecture of an ANN that can be used for data compression. Here $z_1 \in \mathbb{R}^{N_1}$ is the input vector, $y = z_2 \in \mathbb{R}^{N_2}$ is the required output target vector. The specifics of the architecture is that it uses fewer neurons $M < N_1$, $M < N_2$ in the hidden layer with respect to the dimensions of the input and output vectors. The network’s matrix of weighting coefficients $W^{(1,2)}$ is $W^{(1,2)} = W^{(2)}W^{(1)}$, where $W^{(1)}$, $W^{(2)}$ are the weighting matrices of the 1st and 2nd layers respectively. When $z_1 \neq z_2$ the network is heteroassociative, when $z_1 = z_2$ the network is autoassociative. Heteroassociative compression of vector $z_1$ results in output vector

$$\tilde{z}_2 = W^{(2)}W^{(1)}z_1. \quad (2)$$

For further analysis, it is also necessary to consider the situation of a single-layer ANN converter (figure 2) with $N_1$ inputs and $N_2$ neurons in the output layer, and the weighting coefficients matrix $W$. After this ANN is trained by a training set of $P > N_1$, $P > N_2$ realizations of random vectors $\{z_1^{(p)}, z_2^{(p)}\}$, $p = 1, P$, a quasi-optimal estimate is formed at the output. The structure of the quasi-optimal estimate complies with the structure of optimal linear estimate $\tilde{z}_2$ with respect to $z_1$ (i.e. the structure of $\tilde{z}_{2/1}$)

$$\tilde{z}_{2/1} = Wz_1 = \tilde{R}_{z21}\tilde{R}_{z11}^{-1}z_1, \quad \tilde{R}_{z21} = \frac{1}{P - 1} \sum_{p=1}^{P} z_2^{(p)}(p), \quad \tilde{R}_{z11} = \frac{1}{P - 1} \sum_{p=1}^{P} z_1^{(p)}(p), \quad (3)$$

where $\tilde{R}_{z21} = \|r_{jn}^{(2,1)}\|$, $\tilde{R}_{z11} = \|r_{in}^{(1,1)}\|$ are covariance matrices of random vectors that correspond to matrices $R_{z11}, R_{z21}$ [8].

Let us consider the ANN presented in figure 1 more closely. Let the objective function that should be minimised during the ANN training, with respect to the realization set $z_1$, $z_2$, be presented as

$$E = \frac{1}{2} \sum_{p=1}^{P} \left( z_2^{(p)} - \tilde{z}_2^{(p)} \right)^T \left( z_2^{(p)} - \tilde{z}_2^{(p)} \right) = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{N_2} \left( z_{2,j}^{(p)} - \sum_{k=1}^{M} w_{jk}^{(2)} \sum_{i=1}^{N_1} w_{ki}^{(1)} z_1^{(p)} \right)^2,$$

where $z_2^{(p)} = W^{(2)}W^{(1)}z_1^{(p)}$ is the reaction to the input signal $z_1^{(p)}$. Mean squared error (MSE) is given by

$$E = \frac{1}{2} \text{tr} \left( \frac{1}{2} \sum_{p=1}^{P} ( z_2^{(p)} - \tilde{z}_2^{(p)} ) ( z_2^{(p)} - \tilde{z}_2^{(p)} )^T \right) =$$
is equal to autoassociative compression of the linear estimate of the output, with respect to the set of vectors \( \tilde{z} \) covariance matrix of the stochastic component of the optimal linear estimate in (1). The second for the last two summands of the expression for \( \tilde{z} \) is presented as

\[
\sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p)) (\tilde{z}_2(p) - \tilde{z}_1(p))^T = (\tilde{R}_{21} \tilde{R}_{11})^{-1} \tilde{R}_{12} \sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p))^T,
\]

where \( \tilde{z}_2(p) = \tilde{R}_{21} \tilde{R}_{11}^{-1} \tilde{z}_1(p) \) is estimate \( \tilde{z}_2(p) \) with respect to the observed \( \tilde{z}_1(p) \), which is the output reaction of the converter presented in figure 2. As

\[
\sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p))^T = (\tilde{R}_{21} \tilde{R}_{11})^{-1} \tilde{R}_{12} \sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p))^T = (\tilde{R}_{21} \tilde{W}^{(1)} T \tilde{W}^{(2)} T,
\]

the last two summands of the expression for \( E \) are zero. This means that the error minimised for \( \tilde{W}^{(1)} \), \( \tilde{W}^{(2)} \) is presented as

\[
E = \frac{1}{2} \left( \sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p))^T (\tilde{z}_2(p) - \tilde{z}_1(p)) \right) + \frac{1}{2} \left( \sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p))^T (\tilde{z}_2(p) - \tilde{z}_1(p)) \right) =
\]

\[
= \frac{1}{2} \sum_{p=1}^{P} V^{(p)} T V^{(p)} + \frac{1}{2} \left( \sum_{p=1}^{P} (Wz_1(p) - Wz_1z_1(p))^T Wz_1(p) - Wz_1z_1(p) \right) = \tilde{R}_{VV} + E'.
\]

The first one is independent of the \( \tilde{W}^{(1)} \), \( \tilde{W}^{(2)} \) elements and corresponds with the sample covariance matrix of the stochastic component of the optimal linear estimate in (1). The second one determines the MSE of restoring the quasi-optimal linear estimate \( E' \) after compression of a set of vectors \( \tilde{z}_2(p) = \tilde{R}_{21} \tilde{R}_{11}^{-1} \tilde{z}_1(p) \), \( p = 1, P \), performed by the ANN with the architecture presented in figure 1. In this case, the ANN is trained as an autoassociative. Basing on this theory, we can make the following assumptions.

**Assumption 1.** Heteroassociative compression of the input vector of a random field fragment is equal to autoassociative compression of the linear estimate of the output, with respect to the input.

The required minimum \( E' \) is determined by the following system of equations:

\[
\frac{\partial E'}{\partial w_{mn}^{(1)}} = 0, n = 1, N_1, m = 1, M, \quad \frac{\partial E'}{\partial w_{mn}^{(2)}} = 0, n = 1, N_2, m = 1, M.
\]

After differentiating and reduction of similar terms, we receive two matrix equations

\[
\tilde{R}_z W^{(2)} T - \tilde{R}_z W^{(1)} T W^{(2)} T = 0, \quad \tilde{R}_z W^{(1)} T - W^{(2)} T W^{(1)} T = 0,
\]

(5)

\[
\tilde{R}_z = \frac{1}{P-1} \sum_{p=1}^{P} (\tilde{z}_2(p) - \tilde{z}_1(p))^T = \tilde{R}_{21} \tilde{R}_{11}^{-1} \tilde{R}_{12}.
\]

It is necessary to point out, that in (5) \( \tilde{R}_z \) acts as a sample covariance matrix of estimate \( \tilde{z}_2(p) = Wz_1 = \tilde{R}_{21} \tilde{R}_{11}^{-1} z_1 \).
We have already given a thorough mathematical analysis of matrix equation (5) in [8]. The analysis allowed us to prove the following assumption.

**Assumption 2.** Being trained by a set of realizations of random vectors \( \{z_1^{(p)}, z_2^{(p)} \}, P > N_1, P > N_2 \) heteroassociative two-layer ANN with the architecture presented in figure 1 and \( M < N_1, M < N_2 \) neurons in the hidden layer, forms an approximate expression \( \hat{z}_{2/1} = W_2 W_1 z_1 \) of estimate \( \hat{z}_{2/1} = \hat{R}_{z21} \hat{R}_{z11}^{-1} z_1 \) of vector \( z_2 \) with respect to \( z_1 \) presented as resolution of the first \( M \) eigenvectors \( \{\psi_i, i = 1, \cdots, M \} \) of the sample covariance matrix \( \hat{R} \).

\[
\hat{z}_{2/1} = \sum_{i=1}^{M} \alpha_i \psi_i.
\]

The ANN processing matrix implements transformation \( W^{(1,2)} = W^{(2)}W^{(1)} \), equivalent to direct and backward decompositions of the input vector into a limited set of eigenvectors of the matrix \( \hat{R} = \hat{R}_{z21} \hat{R}_{z11}^{-1} \hat{R}_{z12} \). Similarly, \( \hat{z}_{2/1} \) presented by \( z_{2/1} \) for a fixed \( M \) has the minimal dispersion of residual error.

**Deduction.** Autoassociative neural network of \( (z_1 = z_2 = z) \) type, with \( M < N, N = N_1 = N_2 \) neurons in the hidden layer being trained by a set of \( \{z_1^{(p)}, z_2^{(p)} \}, p = \{1, \cdots, P\}, P > N \), gives output reactions equivalent to the first \( M \) basis functions of Karhunen–Loève expansion of the covariance matrix of a random vector \( z \). These results have already been discussed in [9] and proved experimentally.

The theory above demonstrates that it is possible to perform data compression with minimal distortion. This, in turn, yields a practical result of great importance: heteroassociative compression can be performed in two ways. The first supposes training the ANN presented in figure 1. The second way is to calculate the weighting coefficients matrix of an autoassociative converter of linear estimate \( \hat{z}_{2/1} = \hat{R}_{z21} \hat{R}_{z11}^{-1} z_1 \) by solving the eigenvalue/eigenvector problem of matrix \( \hat{R} = \hat{R}_{z21} \hat{R}_{z11}^{-1} \hat{R}_{z12} \). As a rule, backpropagation training takes longer, than calculating the weighting matrices while configuring the converter. On the other hand, as follows from Assumption 1, by using the ANN from figure 1 trained by the sets \( \{z_1^{(p)}, z_2^{(p)} \}, p = \{1, \cdots, P\}, \) to perform heteroassociative compression, we can avoid the linear transformation stage for estimates \( \hat{z}_{2/1}^{(p)} \) that are than to be compressed by means of autoassociative ANN, if we choose the second way. Thus, the compression time can be reduced, if the converter has already been trained.

[8] presents the results of experiments carried out with both random fields realizations, and real images. The aim was to study the performance of the compression algorithms applied to arbitrarily-shaped fragments, as well as to analyse the properties of the algorithms when applied to data that had not been used for training. As proved by the experiments, the results of heteroassociative compression performed by the ANN from figure 1, are very close to the results of a two-stage transformation based on eigenvalue/eigenvector problem. We also demonstrated, that when the spatial correlation of training and test realizations of a random field is high, the data representation error after compression is practically the same for the train and the test sets.

### 3. Video steganography based on heteroassociative compression

The theoretical assumptions above justify the usage of data compression for further steganographical data embedding into various container objects with minimal distortion. The secret information is represented by a bit sequence \( d^{(p)} \), \( p = \{1, \cdots, P\} \), where \( d^{(p)} \in \{-1, +1\} \). Before information embedding the original video or its fragment should be decoded into a sequence of
frames. Then each frame is processed by steganographic algorithm and finally modified frames are compressed using initial or new video coding parameters.

We apply a block-based embedding algorithm with each element of the sequence \(d^{(p)}\), \(p = \overline{1,P}\) being embedded into a separate fragment \(\Omega^{(p)}\) of arbitrary shape, represented by vector \(z^{(p)}\), selected on an uncompressed video frame. The total of \(\{\Omega^{(p)},\ p = \overline{1,P}\}\) covers all the container. When heteroassociative compression is applied, each fragment of the container \(\Omega^{(p)}\) is divided into the input and output parts. The data \(\{z_{1}^{(p)}, z_{2}^{(p)}, p = \overline{1,T}\}\), defining the input and output of all the fragments is used for training ANN with architecture presented in figure 1. Such network has reduced the number of neurons in the hidden layer \(M < N_{1}, M < N_{2}\), and implements data compression. \(N_{1}\) and \(N_{2}\) define the number of elements (pixels) in input \(\Omega_{I}\) and output \(\Omega_{O}\) fragments. The data from one part of the fragment \(z_{1}^{(p)}\) are given to the input of such a ANN, and at the output of ANN we estimate data from the other part of the fragment \(z_{2}^{(p)}\). The embedding procedure of the sequence \(d^{(p)}, p = \overline{1,P}\) is based on the following equation

\[
z_{2}^{(p)} = z_{2/1}^{(p)}(p) + A_{m}d^{(p)}\varphi_{N_{2}} + V^{(p)}, \tag{6}
\]

where \(A_{m}\) is the amplitude of the introduced sequence, \(z_{2/1}^{(p)}(p) = W_{2}W_{1}z_{1}^{(p)}\) – vector obtained at the output of a compressing ANN (figure 1), \(W_{1}, W_{2}\) – weight matrices of the first and second layers of the compressing ANN, \(V^{(p)} = z_{2}^{(p)} - z_{2/1}^{(p)}\) – stochastic components of the prediction \(\hat{z}_{2}^{(p)} = z_{2/1}^{(p)} + V^{(p)}, z_{2/1}^{(p)}\) – optimal in the class of linear estimation of \(z_{2}^{(p)}\) on observations \(z_{1}^{(p)}\), which is received on the output layer of ANN (figure 2). This single-layer network estimates the output of the container fragments \(z_{2/1}^{(p)}, p = \overline{1,P}\) using the input data \(\{z_{1}^{(p)}, z_{2}^{(p)}, p = \overline{1,P}\}\), and thus allows for fixing the stochastic components of the prediction (1). The obtained values \(z_{2/1}^{(p)}\) help to define the normalized vector \(\psi_{N_{2}} = r_{\min}/\sqrt{\sigma_{\min}^{2}}, r_{\min} = 1/P \sum_{p = 1}^{P} (z_{2/1}^{(p)} - \hat{z}_{2/1}^{(p)}), p = \overline{1,P}\).

As a result of (6) each of the completed ("filled") container fragments is then determined by vector \(\tilde{z}^{(p)} = (z_{1}^{(p)}, z_{2}^{(p)}, z_{2/1}^{(p)})\), which means that it consists of the unchanged input data and modified output data. The completed image is then presented as \(\tilde{Z} = \{\tilde{z}^{(p)}, p = \overline{1,P}\}\).

To recover the hidden data effectively, we need to classify the vector of the modified output of the container fragment \(z_{2}\) (or the whole fragment vector \(\tilde{z} = (z_{1}^{T}, z_{2}^{T})\)) as belonging to one of the classes \(H_{1}, H_{2}\). Each class is characterized by different mathematical expectation vectors \(H_{1}: m_{+} = A_{m}\varphi_{N_{2}}, H_{2}: m_{-} = -A_{m}\varphi_{N_{2}}\) and a common covariance matrix \(R_{\eta}\):

\[R_{\eta} = W_{12}R_{11}W_{12}^{T} + R_{V}, \quad R_{V} = M[VV^{T}] = R_{z22} + WR_{z11}W^{T} - WR_{z12} - R_{z12}^{T}W^{T}.\]

In this case, the decision rule is based on the application of a single-layer neural network (either linear or not) with the kind of architecture presented in figure 3. This ANN is trained to classify its input vector in order to define the value of the hidden component of the DWM sequence. The wavy line above the recovered sequence \(d^{(p)}, p = \overline{1,P}\) stands for the possible classification errors.

One of the essential requirements for the algorithms of steganographic data hiding into video is the sustainability of embedded information to video sequence lossy compression. To ensure the implementation of this requirement that is similar to the algorithm proposed by authors [11], steganographic data hiding in blocks has been performed, based on the transformation from modification of individual frame pixels values while embedding message bits to changing average values of neighbouring pixels groups forming blocks of \(k_{h} \times k_{h}\) \((k_{h} = 8\) by default). It should be noted that it is reasonable to use algorithm block modification provided that there are no strict requirements to the capacity of steganographic data hiding algorithm, as in this case the capacity decreases by \(k_{h} \times k_{h}\) times.
4. Experimental analysis of steganographic data hiding algorithm

The quality of cluster steganographic data hiding algorithm performance based on heteroassociative compression transformations was assessed by two indicators, i.e. the level of visual distortion of frames and the probability of error $P_{\text{err}}$ in retrieving earlier embedded data. To carry out experiments we used program modules, developed in MATLAB and concerned with data embedding and retrieval procedures, as well as a number of libraries for coding and encoding of ffmpeg-3.2 video.

The steganographic data hiding algorithm was used for all types of video frames (I,P,B) provided by the modern compression standards, in case we use earlier compressed video sequences as input data. However, in order to assess better the influence of hiding steganographic transformation on visual quality of an image, we conducted an experimental analysis using a sequence of uncompressed frames (“akiyo”, ”city”, ”flower_garden”, ”foreman”, ”tempepte”, ”waterfall”, ”big_buck_bunny”) of a standard Xiph.org Video Test Media set, which have been compressed with x264, x265, VP9 codecs afterwards. Frames resolution was from 352x28 to 1280x720 pixels.

In each test sequence we chose 15 frames to set up and train ANN for the given parameters of the algorithm: the configuration of the processed fragments $\Omega^{(p)}$ and the amplitude of the embedded sequence $A_m$. For each colour component of the frame we chose 12×12 square fragments with square output placed inside $8 \times 8$ and surrounded by the input ($N_1 = 240, N_2 = 192$). The block fragment size was $8 \times 8$ when we calculated median values of groups of pixels. Embedded data $d^{(p)}(p) = \overline{1,P}$, were a black and white binary vector icon.

We hid data into every frame of all test video sequences, and then compressed watermarked frames with each of three chosen codecs x264, x265, VP9 with a given value of constant rate factor crf, that is an analogue of a parameter of compression quality - jpeg. The increase of crf values is accompanied by the increase in compression rate and decrease in the quality of video. We created test videos in a single-pass mode encoding. It should be noted that for the evaluation of the degree of influence of the steganographic data hiding algorithm on the quality of the video, we further compressed the original (empty) representations of frames with the same codec settings. Visual quality of the test video was evaluated by comparing the original uncompressed frames and their corresponding frames decoded from ”filled” and ”empty” video file. To compare frames, we used the criteria of peak signal-to-noise ratio (PSNR) and the modified index of structural similarity (SSIM) Bovika [12], calculated as the geometric mean of the indices obtained for each of the three colour components r,g,b of the current frame.

The reliability of information recovery was computed separately for each set of parameters of the steganographic data hiding and parameters of video compression, as an averaged error probability over many marked frames when retrieving the data. Table 1 and table 2 presents average over the realizations of the test data values of PSNR and SSIM for video sequences without stegohiding and video sequences containing hidden data. During the implementation of steganographic data hiding in each marked frame of test sequences we chose all potential areas for embedding the data. Thus, the coefficient of effective use for a block version of the steganographic data hiding algorithm based on the heteroassociative image compression has gone to 1.

As can be seen from table 1 and table 2, the proposed algorithm provides minimal visual distortion of the watermarked video frames, and hence, the visual obscurity of the procedure of steganographic data hiding.

For all types of codecs with a given set of parameters, the corresponding SSIM values for ”empty” and ”filled” video containers differ by not more than $3 \times 10^{-4}$. The maximum difference between the values of PSNR for compressed ”empty” and ”filled” video-container is about 2 dB for the codec VP9 ($\text{crf}=15$). With the increase in the degree of video compression PSNR values for ”empty” video containers decrease by 1-1.5 dB, and the PSNR values for ”filled” containers
Table 1. PSNR obtained by compressing the original (empty) and the steganographically marked (filled) test video sequences with different codecs.

| Type and parameters of codec | Without stego-hiding | $A_m = 4$ | $A_m = 8$ | $A_m = 12$ |
|------------------------------|----------------------|----------|----------|-----------|
| x264, $crf=15$              | 42.303               | 41.955   | 41.484   | 40.794    |
| x264, $crf=25$              | 40.718               | 40.653   | 39.888   | 39.673    |
| x265, $crf=15$              | 41.311               | 41.154   | 41.083   | 40.619    |
| x265, $crf=25$              | 40.486               | 40.211   | 39.971   | 39.425    |
| vp9, $crf=15$               | 42.166               | 41.784   | 41.306   | 40.573    |
| vp9, $crf=25$               | 40.941               | 40.478   | 39.947   | 39.535    |

Table 2. SSIM obtained by compressing the original (empty) and the steganographically marked (filled) test video sequences with different codecs.

| Type and parameters of codec | Without stego-hiding | $A_m = 4$ | $A_m = 8$ | $A_m = 12$ |
|------------------------------|----------------------|----------|----------|-----------|
| x264, $crf=15$              | 0.9996               | 0.9996   | 0.9995   | 0.9994    |
| x264, $crf=25$              | 0.9995               | 0.9995   | 0.9994   | 0.9992    |
| x265, $crf=15$              | 0.9995               | 0.9995   | 0.9995   | 0.9994    |
| x265, $crf=25$              | 0.9995               | 0.9994   | 0.9994   | 0.9992    |
| vp9, $crf=15$               | 0.9996               | 0.9996   | 0.9995   | 0.9994    |
| vp9, $crf=25$               | 0.9995               | 0.9995   | 0.9994   | 0.9993    |

Table 3. Probability of errors of hidden information recovery, obtained by compression of the steganographically marked test sequences of frames with different codecs for different $crf$.

| Type and parameters of codec | $A_m = 4$ | $A_m = 8$ | $A_m = 12$ | $A_m = 4$ | $A_m = 8$ | $A_m = 12$ |
|------------------------------|----------|----------|----------|----------|----------|----------|
| x264, $crf=15$              | 0.0813   | 0.0454   | 0.0172   | 0.0079   | $<10^{-4}$ | $<10^{-4}$ |
| x265, $crf=15$              | 0.1766   | 0.0825   | 0.0705   | 0.1413   | 0.0029   | $<10^{-4}$ |
| vp9, $crf=15$               | 0.1351   | 0.0574   | 0.0338   | 0.1075   | $<10^{-4}$ | $<10^{-4}$ |
| x264, $crf=20$              | 0.1488   | 0.0671   | 0.0542   | 0.0653   | 0.0007   | $<10^{-4}$ |
| x265, $crf=20$              | 0.3642   | 0.1949   | 0.0921   | 0.3198   | 0.0244   | 0.0113   |
| vp9, $crf=20$               | 0.2975   | 0.1572   | 0.0949   | 0.2707   | 0.0053   | 0.0008   |
| x264, $crf=25$              | 0.2817   | 0.0746   | 0.0611   | 0.1697   | 0.0086   | 0.0006   |
| x265, $crf=25$              | 0.5136   | 0.2678   | 0.1814   | 0.3790   | 0.1174   | 0.1077   |
| vp9, $crf=25$               | 0.4697   | 0.2215   | 0.1172   | 0.4418   | 0.1423   | 0.0335   |

on average, by 1 dB, that characterizes a low level of distortion of containers, done by the
algorithm of steganographic data hiding.

The average probability of errors in recovering information for the data involved (bold text) and uninvolved in the training of ANN are presented in table 3.

As can be seen from table 3, the values $P_{err}$ for the frames involved in the training of ANN were better than expected (for small degrees of video compression almost everywhere $<10^{-4}$) than the corresponding results obtained for the frames that were not in train set.

The greatest resistance of hidden data to the distortion occurring during the compression of "filled" sequences of frames is achieved for large values of the amplitude of an embedded sequence ($A_m \geq 8$) and the given values of the parameters of the quality encoding, providing the final video files are of high quality ($crf \leq 20$). So, for the x264 codec with a constant flow coefficient $crf \leq 20$ $P_{err} \leq 0.06$ when data are embedded with the amplitude $A_m = 12$.

5. Conclusion

The results showed that the proposed block algorithm of steganographic data hiding based on heteroassociative image compression can effectively be used to implement steganographic information hiding (allowing a slight distortion when being recovered) into a sequence of frames and then further compressed with a selected codec in the resulting "filled" video file. The advantages of the algorithm include its versatility (the embedding is implemented not into the compressed sequence of frames and does not depend on the used method of compression) and the minimum insertion level of visual distortion of the container. The algorithm is probabilistic in recovery of hidden data, and the accuracy of the retrieval depends on the degree of compression of the "filled" container. In the absence of strict requirements to the capacity of the steganographic data hiding algorithm, the accuracy of the recovery can be significantly increased by repeated duplication of an embedded message, and through the use of codes of detection and correction of errors. Among the most promising practical uses of the proposed method of steganographic data hiding and the implemented on its basis algorithm is secretive watermarking of video sequences with the repetitive data lines of small size, for example, the lines that identify the copyright holder of the content.

References

[1] Hwang H C W, Wu N I and Tsai C S 2005 IEEE Proceedings-Vision, Image and Signal Processing 152 611–615
[2] Alattar A, Lin E and Celik M 2003 IEEE Transactions on Circuits and Systems for Video Technology 13 787–800
[3] Tew Y and Wong K 2014 Circuits and Systems for Video Technology IEEE Transactions 24 305–319
[4] Nguyen C V, Tay D and Deng G 2006 Circuits and Systems 2006. IEEE Asia Pacific Conference 81–84
[5] Qiu G and et al 2004 ICPR 2004. Proceedings of the 17th International Conference 4 865–868
[6] Sirota A, Dryuchenko M and Mitrofanova E 2015 Radioelectron. Commun. Syst. (in russ.) 58 3–16
[7] Sirota A, Dryuchenko M and Mitrofanova E 2014 C&T 2014, Voronezh (in russ.) 58 68–78
[8] Sirota A and Dryuchenko M 2015 Computer Optics (in russ.) 39 751–761
[9] Haykin S 2005 Neural Networks: A Comprehensive Foundation (in russ.) (Moscow: Izdatelstvo Viliams)
[10] Osovsky S 2002 Neural networks for information processing (in russ.) (PublisherHouse Finance and Statistics)
[11] Sirota A, Mitrofanova E and Dryuchenko M 2015 Scientific journal Proceedings of Voronezh State University, Systems analysis and information technologies (in russ.) 111–121
[12] Wang Z and Bovik A 2002 IEEE Signal processing letters 9 81–84