2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)

MLSB optimal effective weighted container construction for WF5 embedding algorithm

Natalia Voloshina
Cyberphisical systems security Dept.
ITMO University
Saint Petersburg , Russia
nataliv@yandex.ru

Tamara Minaeva
Cyberphisical systems security Dept.
ITMO University
Saint Petersburg , Russia
minaeva-toma-ya@yandex.ru

Sergey Bezzateev
Information Security Technologies Dept.
University of Aerospace Instrumentation
Saint Petersburg , Russia
bsv@aanet.ru

***Abstract*—This paper proposes a multilevel digital watermarking method (WF5) research concentrated on weighted container structure. The primary goal of the research is to find the way how to construct an optimal, efficient weighted container to make an embedding errors invisible. Two types of effective weighted container structures were investigated for color digital images: One Pixel-One Codeword (OP-OC) and One Pixel-Two Codeword (OP-TC) constructions. For both approaches, efficient weighted container structures (masks) that correspond with the Human Vision System (HVS) model were developed, and efficiency of WF5 Multi-Level Significant Bit (MLSB) information embedding algorithms were investigated for different types of digital images.**

Keywords—Digital watermarking, WF5, MLSB, weighted container model, ECC codes, weighted Hamming metric

I. INTRODUCTION

Today there is a problem of copyright protection when distributing multimedia data in modern telecommunication systems. One of the most effective methods for protecting multimedia data is digital watermarking. The main task when embedding digital watermark for invisible watermarking methods is to add author information into the original data (container) in such a way that it completely fits in the container so that the embedding does not affect the quality of the protected object. Besides, the embedded watermark should be resistant to distortions if it is necessary. For example, some watermarks must be resistant to attacks such as filtering, geometric distortions, etc. The other type of watermarks should be destroyed at the slightest distortion of the original protected object. There are two types of embedding methods: embedding in the time domain and a frequency domain [1], [2]. These methods have limitations on the amount of embedded information. The effective volume of the container often determines the amount of embedding, the number of distortions introduced during the embedding process and the resistance to external distortions. Modern research is devoted to the search for methods to overcome these limitations.

One of the ways to solve the problem of efficiency is to apply multilevel embedding approach for multimedia data protection [3], [4], [5]. Multilevel embedding means a steganographic embedding in which the original object that should be protected (image, audio or video content) can be divided into several embedding zones. Each zone could be used for information embedding but with different characteristics. For example, depending on the level of the bit plane in the image it is possible to change the number of bits involved in the embedding process thereby reducing the probability of distorted bits in the higher bit planes [3] and increasing the efficiency of the level of distortion introduced as a result. The problem with this approach is that with a fixed embedding mask the local context is not taken into account. For example, when all bits are used in the first bit plane for embedding and in the second bit plane every second bit is used the peculiarities of distortions perception by the human visual system (HVS) are not taken into account.

Further improvement of multilevel embedding is the use of error-correcting codes [2], [6]. Error correcting codes are used both to protect the watermark from distorting effects and to improve the efficiency of embedding by optimizing the ratio of the amount of embedded information to the number of occurred distortions taking into account the perception of distortion by the human visual system [4],[5]. The implementation of multi-level embedding on the basis of the steganographic embedding method F5 is based on error-correcting codes constructed in the weighted metric and the weighted container model [7], [8]. A weighted container is a structure that should be constructed from an original protected object and should have several significance zones for which weights are assigned. The simplest example of a weighted container can be considered as a structure consisting of bit planes of any digital image. For each bit plan (zone) the weight could be assign. Taking into account the peculiarities of perception of distortions by the human visual system the higher the number of the bit plane, starting with the least significant one, the stronger the distortions introduced in it are and the higher the weight of the zone of significance.

At the moment the research is aimed at finding ways to construct weighted containers and estimate the effects of their structure on the performance characteristics of embedding methods. For example, for the LEBC method the disadvantage is the need for preliminary container analysis when embedding [4]. And for the WF5 [5] method the rules for building a container structure are not defined yet.

In this paper the study is devoted to the analysis of the construction of the optimal container configuration when embedding by MLSB (Multi-Level Significant Bit) method so called WF5 method [7].

This work was partly financially supported by Russian Foundation for Basic Research in 2017 (grant 17-07-00849-A).

II. EMBEDDING METHOD

To realize the possibilities of effective use of the original container usage of the codes in the weighted Hamming metric for the F5 syndrome coding algorithm is proposed in [5]. The main feature of such codes [9] is the different error's significance (weight) determined by its position. This allows to distribute the information embedded in the container according to the significance of the container elements. That is, the higher the significance of the container element, the greater the weight of the error on the corresponding position of this component of the codeword for the error-correcting code used. Thus, the structure of the used container is initially determined in terms of the significance of its elements. For example, suppose the set of container elements selected for information embedding is M bits. Moreover, on these m bits, the structure of the position significance is determined in some way

$\{M_1, M_2, \ldots, M_l\}, \mid M_1+M_2+ \ldots +M_l = M$

with corresponding values of their significance $\{v_1, v_2, \ldots, v_l\}$.

For reasons of integrity, ease of use, etc., the block size n corresponding to the length of the codeword is selected.

$$n=n_1+n_2+\ldots+n_l,$$

where $\frac{M}{n} * \sum_{i=1}^{l}(M_i - n\_i) \to \; min, \; n_i \cdot M/n \leq M_i, \; i=1,\ldots,1.$

In this case, each fragment of the codeword of length n_i corresponds to its significance w_i. For simplicity, but without loss of generality, we assume that the significance of the positions in the original container and the corresponding significance of the positions of the codeword are defined as follows

$$M_1=w_1=1, \; M_2=w_2=2, \; \ldots, \; M_l=w_l=l.$$

To define the error-correcting code in the weighted Hamming metric, it is natural to define the distance $d_{wH}(a, b)$ between two vectors a and b as :

$$d_{wH}(a, b) = \sum_{i=1}^{l} \sum_{j=1}^{n\_i} \#(a_{ij} \neq b_{ij}),$$

where $\#(k \neq m)=1$ if $k \neq m$ and $\#(k \neq m)=0$ if $k=m$.

$$a = (a_{11}, a_{12}, \ldots, a_{1n_1}, a_{21}, \ldots, a_{l1}, \ldots a_{ln_l}),$$

$$b = (b_{11}, b_{12}, \ldots, b_{1n_1}, b_{21}, \ldots, b_{l1}, \ldots b_{ln_l}),$$

In the general case, to construct the error-correcting code with the selected parameters in the weighted Hamming metric, we can use a well-known in the coding theory approach used in the proof of the Varshamov-Gilbert bound. In some cases, with a certain ratio of weights w_i and lengths n_i in the codeword structure, it is possible to use a simpler procedure [10] to build error-correcting code in the weighted Hamming metric.

Here, as a simple example, we consider the use of the embedding method for a weighted container in the case where $n_1=4, n_2=3$ and $n=n_1+n_2=7$. The significance of the positions of the source container and the error-correcting code used is defined as $w_1=1, w_2=2$. In this case, we will use the error-correcting code in the weighted Hamming metric of length $n=7$ $(n_1=4, n_2=3)$ with the parity check matrix H.

$$H = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}$$

III. THE CONTAINER CONSTRUCTION IN CASE OF OF THE MULTI-LEVEL EMBEDDING.

Let's, for example, consider a digital color static image in BMP format as an object of protection. In steganographic embedding, this object is defined as a container. To implement multilevel embedding procedure, it is necessary to construct a container structure corresponding to its weighted model.

A weighted container model is a representation of the source object, in which it is allocated several significance zones. The significance is defined by the degree of sensitivity to distortions occurring in a given zone, i.e., for example, the degree of influence of changes in bits on the visual quality of the resulting image. Therefore, to each zone in the weighted model (significance zone) a certain weight can be assigned. An example of a weighted static image model is shown below on Fig. 1.

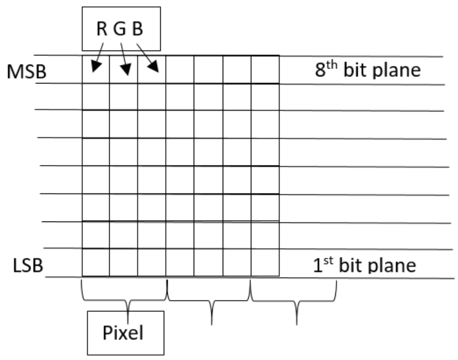

Fig.1 The weighted model of a static image

In the realization of embedding in a weighted structure, it is necessary to allocate the effective part of the container into which the embedding will be performed. The errors will occur in this area in the process of the information embedding.

The simplest example of such a structure is LSB embedding [1]. In this procedure, the lower bit plane of the image acts as an effective container. A weighted container model, in this case, contains only two zones: LSB (least significant bits) and MSB (most significant bits).

The size of an efficient container, i.e., its volume, is one of the most important limitations on the maximum amount of embedded information. That is, when the volume of an effective container increases, the achievable limit of the embedding volume increases too.

The effective container volume increase for the images means that we need to use the higher bits of the bit planes of the images than the LSB in embedding procedure. At the same time, the structure of the efficient container, i.e., the embedding mask, becomes more complex, since the distortions in bit planes higher than LSB, cause distortions in the image, which have different visibility. For example, for an RGB colorimetric representation system, each pixel in an image can be represented as a vector of three 8 – bit color components-red, green, and blue. The least significant bit, however, is the least significant bit of each component. It is also known that the G component is the most significant after it is more significant is the R component, and the least significant is the B component. Hence, the weighted model of the image can be represented as on Fig. 2.

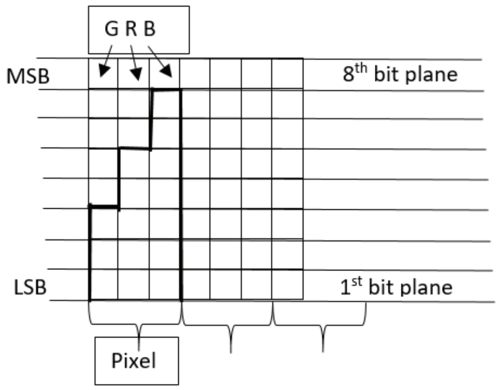

Fig. 2 The weighted model of a static RGB image

When implementing a multi-level embedding procedure it is necessary to consider this model in designing an efficient container. Research on how to build such containers is presented below.

IV. One Pixel-One Codeword (OP-OC) Construction

To analyze the results of the effective container structure on the embedding efficiency, the WF5 method was first considered. In WF5 approach one codeword of length 7 was embedded in each pixel of the image, i.e., 4 bits of the information sequence were embedded. The embedding process is shown on Fig.3.

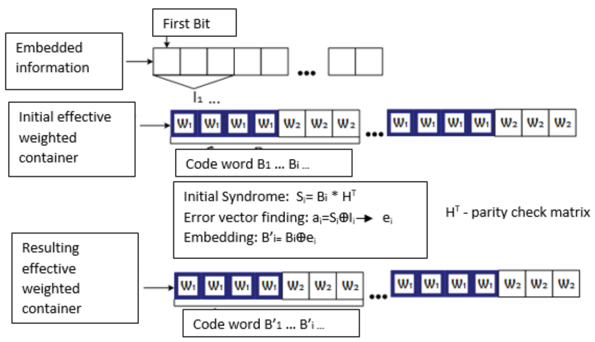

Fig. 3. The procedure of the data sequence embedding

By the using the weighted model of the container described above, with the length of the codeword equal to 7, various options for building an effective container are possible. Hence the influence of its configuration on the effectiveness of embedding process in evaluating the effectiveness using the PSNR metric was studied.

The figure below shows the variants of the effective container structure. The least significant bits of the vector with the minimum weight w_1 are located in the lower bit planes, and the bits with the higher weight, in the higher ones. In this paper, we consider the principle of embedding information dense placement with minimal usage of higher bit planes, which helps to minimize visual distortion. Taking into account the length of the codeword vector embedding procedure is carried out no more than 3 bit planes.

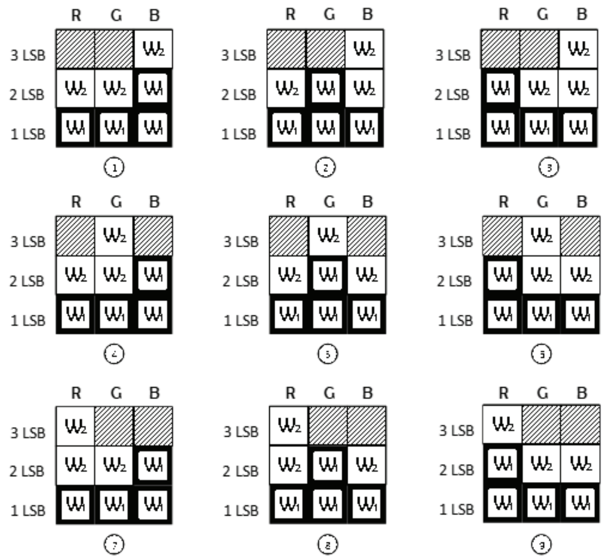

Fig. 4. The distribution of the codewords elements in the pixel.

The visual quality indicator PSNR for different components and brightness, depending on the type of test image for the first version of the effective container structure Fig. 4. Also on Fig. 6 and Fig. 7 a graph of PSNR brightness for all variants of embedding process and all variants of the effective container structure are presented.

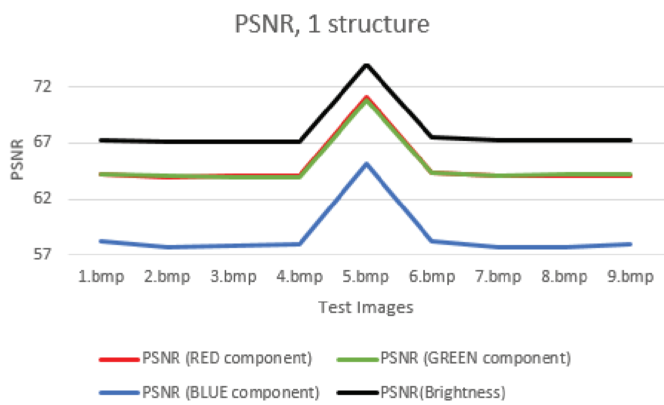

Fig. 5. PSNR graph of different components and brightness for the first variant of the effective container structure depending on the type of test image

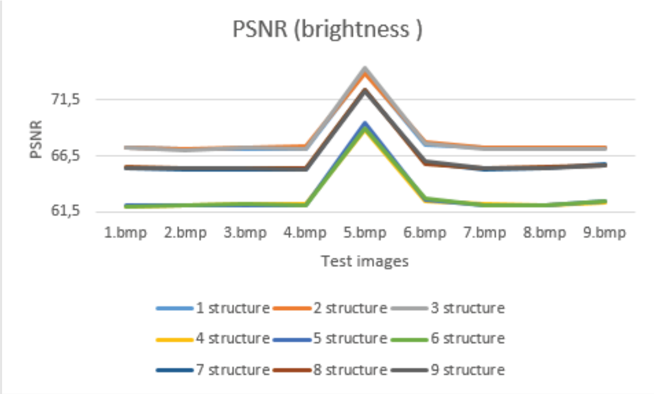

Fig. 6. PSNR brightness graph on the type of test image for all variants of effective container structure

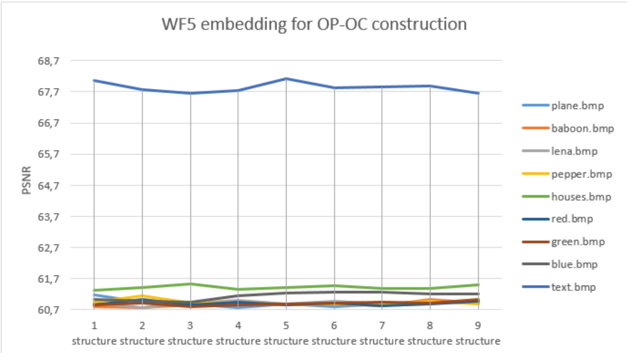

Fig. 7. PSNR brightness graph on the structure of an effective container for all variants of effective container structure.

From the graphs, one can see that for any embedding option, the PSNR value for both components and brightness does not drop below 57 dB, which is well above the 40 dB distortion visual visibility threshold. The best result on the invisibility of distortion is achieved for the text image.

The obtained results suggest that for pixel-by-pixel embedding procedure by the WF5 method with the codeword length equal to 7, any variant of the effective container structure for different types of source images and their components can be used.

To further increase the embedding information size, let's consider the option of setting two codewords in one pixel by the WF5 method.

V. One Pixel-Two Codewords (OP-TC) Construction

Let's consider the impact of increased number of the codewords embedding in the pixel to increase the size of embedding information by method WF5. The figure shows the structure of an efficient container in embedded procedure in four lower bit planes. The figure shows the structure of an efficient container when embedding procedure use the lower four bit planes of an image. Using only four lower bit planes ensures high visual quality of the resulting images. Bit distribution in the structure (Fig. 8) corresponds to the weighted image model described above.

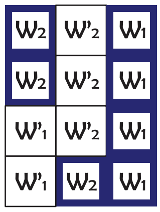

Fig. 8. Working area in case of double-embedding procedure

In the presented structure of an efficient container, bits w_1 and w'_1 are the least significant bits, and bits w_2 and w'_2 are the most significant bits. At first step of embedding procedure the bits marked in blue are filled. Then the second rest bits are filled. In this case, when each codeword constructed, the columns of the parity check matrix H are changed pseudorandom. For such embedding procedure we will obtain the following PSNR for 9 different original pictures (Fig. 9). For all pictures we use same information for embedding, the container is completely filled with embedding information.

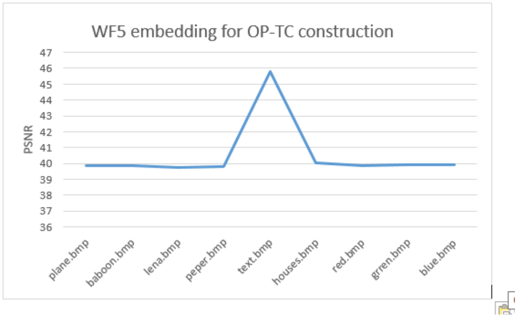

Fig. 9 PSNR with double-embedding procedure (number of parity check matrix is changed pseudorandom)

0	*0*	*0*	*0*	*1*
1	*0*	*0*	*1*	*0*
2	*0*	*1*	*0*	*0*
3	*1*	*0*	*0*	*0*
4	*1*	*1*	*1*	*0*
5	*1*	*1*	*0*	*1*
6	*1*	*0*	*1*	*1*

Fig. 10. Seven columns of the parity check matrix H_0

H_1	*0123456*	H_{11}	*4352061*
H_2	*4106532*	H_{12}	*1526430*
H_3	*0364215*	H_{13}	*2160345*
H_4	*4031526*	H_{14}	*0432615*
H_5	*5431620*	H_{15}	*5216430*
H_6	*5326041*	H_{16}	*3456012*
H_7	*4631520*	H_{17}	*0163245*
H_8	*6015342*	H_{18}	*5214603*
H_9	*5032461*	H_{19}	*4301526*
H_{10}	*5016342*	H_{20}	*0651234*

Fig. 11. Variants of different parity check matrix obtained from H_0 by permutation

On Fig. 13 we show PSNR indicator values for different parity check matrix in case when for embedding process the same information (the same content and size) and different types of images (Fig. 12) are used. An example is given for the color image plane.bmp containing the general plan's image without the lack of dominant color channel. The figure shows that PSNR is not reduced below 36 dB when embedded process using the proposed method of container formation with fixed parity check matrix. This fact is indicates the absence of distortions visualization. In case of embedding process with step-by-step change of the parity check matrix, the PSNR value is fixed at 40 dB.

Table 1 shows the results of the embedding quality by PSNR for the nine test images. The table shows that the worst result is obtained for an image that contains only the blue component (blue.bmp) when the H_4 matrix is used. The best result is achieved for a text-type image (text.bmp), with the worst value also corresponds to the matrix H_4.

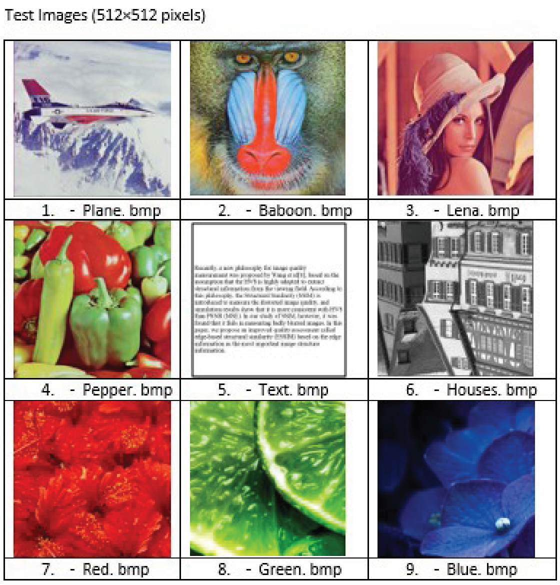

Fig. 12. Test RGB color images.

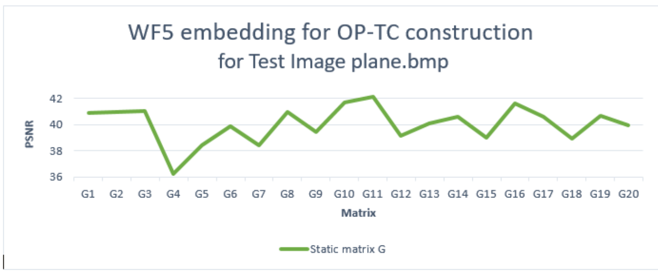

Fig. 13. PSNR Diagram in case of different matrices generation for image plane.bmp

Below is a graph of the experimental results on embedding information process when matrix H4 is used. In this case, the PSNR values on the test sample images are the worst. The figure shows that the PSNR value for any image does not reach 30 dB, which indicates the invisibility of the distortions.

VI. CONCLUSION

Two types of effective weighted container structures were investigated for color digital images: One Pixel-One Codeword (OP-OC) and One Pixel-Two Codewords (OP-TC) constructions. For both approaches, efficient weighted container structures (masks) that correspond with the Human Vision System (HVS) model were developed, and efficiency of WF5 MLSB information embedding algorithms were investigated for different types of digital images. It was shown that both constructions could be used for MLSB WF5 embedding with good resulting image quality. That means that it is enough to fix exact structure for embedding algorithm instead of creating it for each pixel. Even for test image with text the efficiency of proposed weighted container structures is high.

REFERENCES

[1] D. Neeta, K. Snehal, D. Jacobs,"Implementation of lsb steganography and its evaluation for various bits", Digital Information Management: 1st International Conference , 2006, pp. 173–178.
[2] A. Westfeld, "High Capacity Despite Better Steganalysis (F5-A Steganographic Algorithm)" Information Hiding. 4-th International Workshop. Lecture Notes Computer Science, Berlin; Heidelberg; New-York: Springer-Verlag, 2001. Vol. 2137, pp. 289–302.
[3] A. Pandey, J. Chopra, "Comparison of Various Steganography Techniques Using LSB and 2LSB: a Review", International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 6, Issue 5, May 2017, pp. 522-525
[4] R. Darit, E.M. Souidi, "An Application of Linear Error Block Codes in Steganography" , International Journal of Digital Information and Wireless Communications (IJDIWC), 1(2), 2011, pp. 426-433.
[5]S.Bezzateev, N.Voloshina, K.Zhidanov, "Steganographic method on weighted container", Problems of Redundancy in Information and Control Systems (RED) XIII, 2012, pp. 10-12.
[6] R. Zhang, V. Sachnev, H.J.Kim, "Fast BCH syndrome coding for steganography", In Lecture Notes in Computer Science , v. 5806 LNCS, 2009, pp. 48-58.
[7] S.Bezzateev ,N. Voloshina, K.Zhidanov, "Multi-level Significant Bit (MLSB) Embedding Based on Weighted Container Model and Weighted F5 Concept", Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, pp. 293-303
[8] N.Voloshina, K.Zhidanov, S.Bezzateev, "Optimal Weighted Watermarking for Still Images", 2014 XIV International symposium on problems of redundancy in information and control systems. pp. 98-102
[9] S.Bezzateev, N.Shekhunova,"Class of generalized Goppa codes perfect in weighted Hamming metric", Designs, Codes and Cryptography, 66(13), 2013, pp. 391-399.
[10] S.Bezzateev, N. Shekhunova,"Generalized Goppa codes for correcting localized errors", Proceedings of ISIT-98, Boston, USA, 1998, p.377.

TABLE I. EVALUATION OF THE QUALITY (PSNR) OF EMBEDDING FOR NINE TEST IMAGES

	PSNR, dB plane.bmp	**PSNR, dB baboon.bmp**	**PSNR, dB lena.bmp**	**PSNR, dB peper.bmp**	**PSNR, dB text.bmp**	**PSNR, dB houses.bmp**	**PSNR, dB red.bmp**	**PSNR, dB green.bmp**	**PSNR, dB blue.bmp**
H_1	40.84918737	40.83966112	41.0704	40.84150491	48.57136	40.87124	40.86351	40.88487	40.99109
H_2	40.9536321	40.93310425	41.15747	40.93331482	48.37431	40.98399	41.0176	41.01982	41.32561
H3	40.9927765	40.99486266	41.04359	40.99985581	46.26458	41.0372	41.07928	41.08644	41.28039
H_4	36.23688798	36.22085171	36.31425	36.06799796	36.34329	36.2588	35.72181	36.19058	35.55687
H_5	38.40885539	38.37776105	38.39453	38.36887919	43.63974	38.43971	38.39337	38.40661	38.56681
H_6	39.86040255	39.8773152	39.96059	39.87842248	44.04424	39.88546	40.0294	39.92087	40.17321
H_7	38.43042283	38.38093513	38.55527	38.35976921	45.25108	38.74298	38.43401	38.45547	38.5038
H_8	40.94756137	40.92664791	41.26268	40.94655269	48.5904	40.98003	41.06409	41.04845	41.29133
H_9	39.40139839	39.35262571	39.49908	39.37507883	46.07633	38.82851	39.44183	39.49236	39.43586
H_{10}	41.650048	41.56503936	41.79805	41.57921235	48.83049	41.59972	41.72562	41.7488	42.0217
H_{11}	42.131972	41.72411917	42.04682	41.72850957	49.39342	46.20722	41.8656	42.01377	42.30208
H_{12}	39.15780735	39.14844478	39.01545	39.15344081	46.30687	39.03212	39.21059	39.20803	39.24851
H_{13}	40.1118006	40.07848744	40.15502	40.08597053	45.79346	40.42388	40.10583	40.15235	40.35505
H_{14}	40.54864868	40.55771621	40.64819	40.57125429	47.51432	40.48023	40.5854	40.55734	40.8101
H_{15}	39.00224235	38.98560654	38.72704	38.97554017	46.00926	39.13294	39.0157	39.02208	39.19881
H_{16}	41.58860025	41.19299095	41.09947	41.21799516	48.94413	44.99025	41.39733	41.52331	41.80115
H_{17}	40.55554793	40.10015849	40.29036	40.10454607	47.3579	45.6365	40.22872	40.42371	40.55556
H_{18}	38.88388923	38.85767753	38.9179	38.84067094	46.03794	41.21628	38.89574	38.94208	39.02625
H_{19}	40.67450504	40.65210399	40.84229	40.66166133	48.28234	40.69596	40.69124	40.69888	40.8161
H_{20}	39.93723321	39.90324557	39.96233	39.96193351	47.98368	39.74947	39.99849	39.97143	39.99618