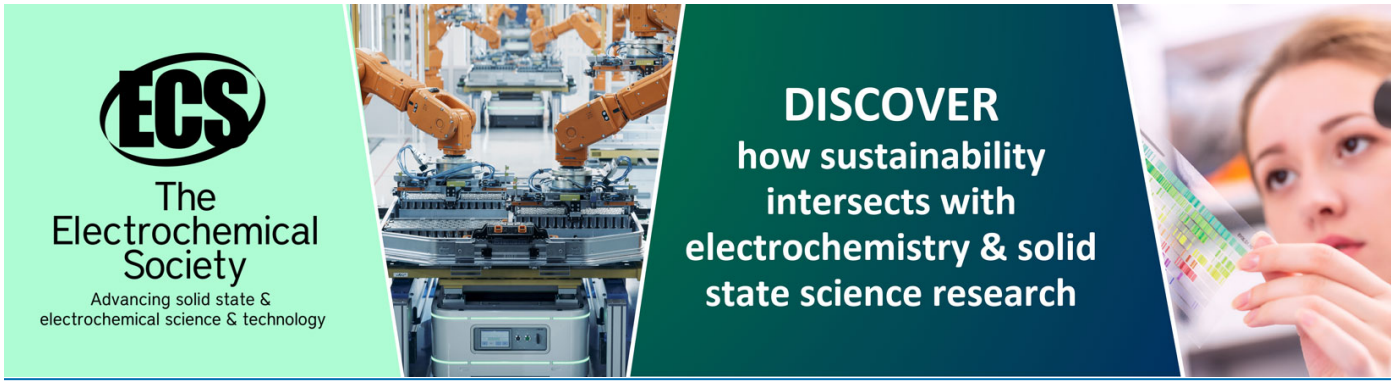
Algorithm for detection of steganographic inserts type LSB-substitution on the basis of an analysis of the zero layer



Algorithm for detection of steganographic inserts type LSB-substitution on the basis of an analysis of the zero layer

S V Belim, D E Vilkhovskiy

Dostoevsky Omsk State University, pr. Mira 55a, Omsk, 644077, Russia

e-mail: sbelim@mail.ru, vilkhovskiy@gmail.com

Abstract. All articles *must* contain an abstract. The abstract text should be formatted using 10 point Times or Times New Roman and indented 25 mm from the left margin. Leave 10 mm space after the abstract before you begin the main text of your article, starting on the same page as the abstract. The abstract should give readers concise information about the content of the article and indicate the main results obtained and conclusions drawn. The abstract is not part of the text and should be complete in itself; no table numbers, figure numbers, references or displayed mathematical expressions should be included. It should be suitable for direct inclusion in abstracting services and should not normally exceed 200 words in a single paragraph. Since contemporary information-retrieval systems rely heavily on the content of titles and abstracts to identify relevant articles in literature searches, great care should be taken in constructing both. Keywords – search for LSB-inserts, analysis of steganography container, revealing of steganography inserts.

1. Introduction

The most common method of embedding steganographic inserts to date is to replace the least significant bits (LSB- substitution) [1]. This method is based on the fact that the replacement of one to four least significant bits of the color representation of the image pixels remains almost invisible to the human eye. The greatest way for hiding information is the blue component, which is due to the structure of the retina. To date, many algorithms have been developed for embedding information in images, audio and video streams, but the method of LSB-replacement, historically the first one, is widely used. All methods of hiding information are focused on such an image transformation, which is not visible to the human eye. In this connection, the actual task is to develop image analysis algorithms for the presence detection of steganographic inserts.

To date, all existing algorithms are focused on the definition of the fact of the presence or absence of the steganographic insert in the image. In articles [2, 3] a method of statistical analysis Chi-square based on the assumption of random distribution of the least significant bits of color in image. This method gives good results when the container is evenly filled and is weakly applicable when randomly selecting pixels to replace the lower bits. In the article [4], steganoanalysis is performed on the basis of comparing the least significant bits in neighboring bytes using the Markov chain formalism. In [5], a method for detecting a steganographic insert based on the use of artificial neural networks. It is shown that for a sufficiently large volume of the training sample, the neural network is able to determine the presence of an insert with an error not exceeding 15%. All known to date methods of stegoanalysis of the LSB-substitution method are effective when filling the steganographic container by at least 50% [6]. In [7], a method for detecting embedded information based on information compression

algorithms. The main idea of the method is that the random data is compressed more weakly than the ordered ones. This approach makes it possible to determine with high accuracy the presence of a steganographic insert when filling a container from 40%. This method was further developed in [8] based on the usage of pre-processing for image, allowing to use it at a much lower container coverage.

It should be noted that to date, there are no algorithms that determine the bytes in which the least significant bit was done. This problem is close to the problem of detecting pixels damaged by impulse noise. For impulse noise, a color change of an arbitrarily selected byte is randomly selected. However, the task of searching for an embedded message is more complex, since the change amount is only one bit.

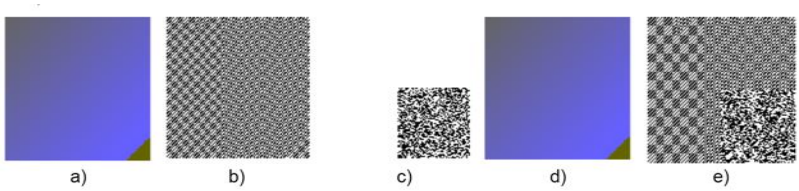


Figure 1. Comparison of the zero layers: a) the original image, b) the zero layer of the original image, c) the embedded pixel map, d) the image with the embedded message, e) the zero layer of the image with the embedded message.

To date, there are several methods for detecting corrupted pixels. First of all, it is necessary to single out the method SDROM [9], which, being historically the first, served as the basis for algorithms [10, 11]. There are also other approaches based on the search for associative rules [12, 13], the clustering method [14], and the hierarchy analysis method [15].
In this article an algorithm for detecting pixels in the image in which substitution is made at the least significant bit steganography embedding messages based on automatic analysis of the zero layer. Formatting the title, authors and affiliations
Please follow these instructions as carefully as possible so all articles within a conference have the same style to the title page. This paragraph follows a section title so it should not be indented.

2. Formulation of the problem

We will analyze images in which information can be embedded in the form of steganographic inserts in the lower bit of the blue component. The analysis of the blue component is due to the fact that embedding in it is the least noticeable visually, and therefore it is recommended to use it for making message embedding. Similarly, other components can be analyzed without loss of generality of the proposed method. We start with two assumptions. First, it is not known whether there is a steganographic insert or not. Secondly, we will assume that the steganographic insert fills a certain rectangular area whose dimensions and position are unknown. The task is not only to determine the presence of the steganographic insert, but also the area in which the embedding. The second assumption substantially complicates the problem, since a situation is possible in which all the lower pixels of the blue component are replaced. We solve the problem, based on the assumption of incomplete substitution of the zero layer.

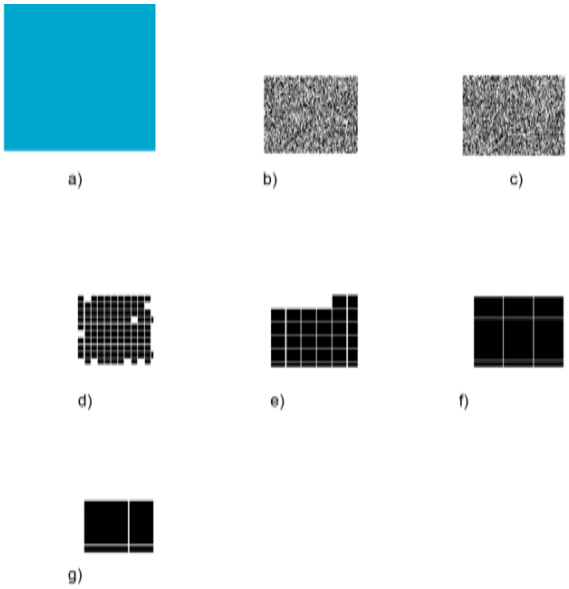


Figure 2. The results of the algorithm for automatic allocation of the embedding region for a uniformly filled image: a) Image with embedded insert, b) embedded pixels map, c) zero layer, d) Automatically selected area of embedding when R_0=15, e) Automatically selected area of embedding when R_0=30, f) Automatically selected area of embedding when R_0=50, g) Automatically selected area of embedding when R_0=100.

The zero layer is a matrix of zeros and ones. The patterns of the distribution of zero and unit values of the zero layer without embedding are due to the structure of the image. The embedding of a message makes a change to the zero layer, changing the density of the distribution of unit values. Figure 1 shows an image, its zero-layer without embedding, and a zero layer with embedded message.

As can be seen from Figure 1, the presence of an embedded message can be detected visually from the analysis of the zero layer. We set ourselves the goal of automatically determining the embedding area.

3. Algorithm for finding the embedding area

To select the embedding area, we use an algorithm based on the FOREL taxonomy algorithm [16]. In its classical form, FOREL combines points into taxa that lie inside the circle.

In our case, we will build rectangular taxa. We introduce the density of unit values$p$. If in some area of the image contains N pixels and N_1 of them has a value one, Unit density index p = N1/N. We will look for rectangular regions having a density of unit values of a given value p_0. Also, as an input parameter of the algorithm, we set the parameter R_0, which determines the initial size of the taxon. The algorithm consists of the following steps:

Step 1. Choose the initial value of the taxon size $R = R_0$.

Step 2. We randomly choose a point with coordinates(x_1, y_1), that will be the center of the taxon. We construct a square whose upper-left corner has coordinates $(x_1 - R,\ y_1 - R)$, and the lower-right corner of the coordinate $(x_1 + R,\ y_1 + R)$.

Step 3. We seek the center of mass coordinates of points lying inside a square-built(x_2, y_2).

Step 4. If the points $(x_1,\ y_1)$ and (x_2, y_2) is the same, then go to Step 5, otherwise $x_1 = x_2, y_1 = y_2$ and go to Step 2.

Step 5. Calculate the density of unit values p.

Step 6. If $p > p_0$, then $R := 1.1R$ and go to Step3.

Step 7. If $p < p_0$, then $R := 0.9R$ and go to Step 3.

Step 8. If $p = p_0$, then go to Step 2.

The algorithm is performed until all the points of the zero layer are combined into some taxa.

As areas in which a message can be embedded, we select taxa whose size is not less than 10% of the size of the original image. The results of this algorithm for a uniformly filled image are shown in Figure 2.

As can be seen from Figure 2, as a result of the operation of the algorithm on auto-generated image with a uniform fill, the embedding area is determined quite accurately. However, even for artifacts with gradient fills, complications arise, since a continuous change in the color of the image as a whole is manifested as bands of identical values on the zero layer (Figure 1, b). These bands can be eliminated by pre-processing the image.

4. Image preprocessing algorithm

As was shown above, the zero layer of a gradient-filled image represents bands of zeros and ones. We use the linear transformation:

$$d(x, y) = ax + by - e$$

Where $a = c(x + 1, y) - c(x, y)$, $b = c(x, y + 1) - c(x, y)$ – the color value of the pixel located at the point with coordinates (x, y). Let's define e, as minimal value of $c(x, y)$ on the set of all pixels of the image. In the case where the color of the image is a fill with a constant gradient, the function $d(x, y)$ will have a constant value ($d(x, y) = const$). An algorithm for making decisions about changing a pixel can be applied to a function $d(x, y)$, rather than to a function (x, y).

To apply this linear transformation to photographic images, it is necessary to define the gradient fill areas. We will calculate the second derivatives of the function (x, y) and identify the regions in which they have a zero value. Due to the fact that the areas of ideal gradient filling on photographic images are extremely rare we will require the fulfillment of three more "soft" conditions:

$$\left|\frac{\partial^2 c(x,y)}{\partial x^2}\right| \leq 2\ , \left|\frac{\partial^2 c(x,y)}{\partial y^2}\right| \leq 2, \left|\frac{\partial^2 c(x,y)}{\partial x \partial y}\right| \leq 2.$$

Not a strict inequality instead of zero is introduced in order to take into account small deviations from the gradient fill and not to lose the embedded bits.

After identifying connected regions which satisfy the conditions for the second derivatives necessary to determine the coefficients of the function $d(x, y)$. To find them, the least squares method was used. After that, the value of the function $d(x, y)$, to which the algorithm for searching the message embedding areas was applied was calculated.

5. Discussion of results and conclusions

A computer experiment was conducted for various color images, both auto-generated and photographic. The embedded message was a text string which is represented as a sequence of bits. Embedding was made in the blue component, as the least visible to the human eye. Embedded bits filled the rectangular area. The task of stegoanalysis was to determine the area in which the substituted bits.

Initially, the algorithm was tested on a rectangular auto-generated image with a gradient fill. The results are shown in Figure 3.

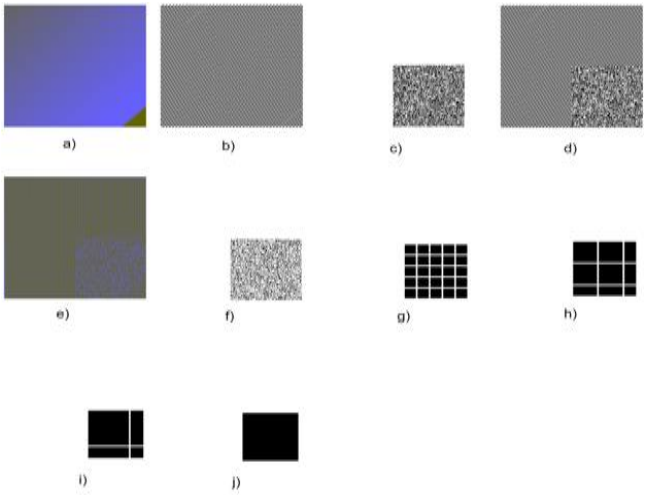


Figure 3. The results of the algorithm for automatic allocation of the embedding region for a gradient-filled image: a) Original image, b) Zero layer of the original image, c) Embedded pixel map, d) Zero layer with embedded message, e) Image after pre-processing, f) Zero layer of image after pre-processing step, g) Automatically selected area of embedding when R_0=15, h) Automatically selected area of embedding when R_0=30, i) Automatically selected area of embedding when R_0=50, j) Automatically selected area of embedding when R_0=100.

As can be seen from Figure 3, the image preprocessing algorithm allows processing gradient-filled images as efficiently as single-color images. Both the uniform and the gradient fill on the results of the algorithm work is affected by the choice of the initial size of the taxon R_0.

References

[1] Adelson E 1990 Digital Signal Encoding and Decoding Apparatus. U.S. Patent. No. 4,939515
[2] Provos N Honeyman P 2001 Detecting steganographic content on the internet *CITI Technical Report 01-11* (University of Michigan)
[3] Westfeld A Pfitzmann A 2000 Attacks on Steganographic Systems: Breaking the Steganographic Utilities EzStego, Jsteg, Steganos and STools-and Some Lessons Learned *International Workshop on Information Hiding* pp 61–76
[4] Golub V A Dryuchenko M A 2009 Steganographic information detection in JPEG files with the help of complex usage of several stego-attackes *Infocommunication Technologies* 7(1) pp 44–50
[5] Abdenov A Zh Leonov L S 2010 Ispolzovaniye neyronnykh setey v slepykh metodakh obnaruzheniya vstroyennoy steganograficheskoy informatsii v tsifrovykh izobrazheniyakh *Polzunovsky vestnik* 2 pp 221–225
[6] A. Westfeld, A. Pfitzmann. 2000 Attacks on Steganographic Systems. Breaking the Steganographic Utilities EzStego, Jsteg, Steganos and S-Tools - and Some Lessons Learned. Lecture Notes in Computer Science, 1768 pp 61–75
[7] M.Yu. Zhilkin 2008 Stegoanaliz graficheskikh dannykh v razlichnykh formatakh. Doklady TUSURa **2**(18) pp 63–64
[8] Monarev V A 2012 Sdvigovyy metod obnaruzheniya skrytoy informatsii. *Vestnik SibGUTI* **4** pp 62–68

[9] Abreu E, Lightstone M, Mitra S K, Arakawa S K 1996 A new efficient approach for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Image Processing, IEEE Transactions* on 5 pp 1012–1025.
[10] Garnett R, Huegerich T, Chui C, He W 2012 A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Trans Image Proccess* ***14***(11) pp 1747–1754