TEXT STEGANOGRAPHY IN COLOR IMAGE USING ONE LSB METHOD ALONG WITH HISTOGRAM CLUSTERING

Seyed Enayatallah Alavi[1], Raana Mirzavand[2]

[1] Shahid Chamran University of Ahvaz, Ahvaz, Iran; se_alavi@yahoo.co.uk
[2] Shahid Chamran University of Ahvaz, Ahvaz, Iran; mrahnaz@yahoo.com

Abstract. There have been presented several methods of steganography in image. A lot of these methods of embedding have been used at Least Significant Bit (LSB) positions of each pixel. In the proposed method in this article, using the histogram packaging and specifying is the most proper package for hiding text. The most important specification in this method is the little discrepancies in the histogram.

1.Introduction. Steganography is basically a greck word which is a mix of stago and graphia. Stego means covering and graphia means writing[1]. In this paper, the 24-bit color image will be selected by a cover . These images are called cover images. The covering image will be hidden along with the secret message and it is called, stego image[2]. Steganography should be enough strong for hiding the secret data and it shouldn't cause fundamental changes in the carrier. Using LSB won't cause fundamental changes in the images[1] .yet it causes changes in the histogram which is one of the approaches in discovering the steganography in the image. We have proposed an approach to solve this problem. In this article, we have presented our proposed method in section 2, and the results, conclusion and suggestions are in sections 3 and 4.

2-Proposed Algorithm. Due to the fact that we use a color image as a cover image, each image pixel includes three bytes red, blue and green and if we use one LSB, we can embed three bits in each pixel. The procedure is as follow: 1-We replace the text characters with their equivalent Aski codes, i.e. each character becomes eight bits. 2-We find the grayscale image of the color image equivalent. 3-We divide the pixels of the grayscale image into 128 clusters. The first cluster includes the pixels with 0 or 1 amounts and the 128th cluster includes the pixels with 254 or 255. It means each cluster has two kinds of pixels, one has even amount and the other has odd amount. 4-We arrange the clusters according to their members in a rising manner. We call a cluster Max which contains the maximum member and also we review the clusters among the 64 clusters with the maximum members which include at least (8* message length/3), it means we can embed the message in it and we select the cluster in which there are the least members even and odd ones to be embedded and we call it Fit. 5-We alter the Fit cluster to an eight-bit number and also we alter the message length to a thirteen bit number and we save this (21 bits) in the first 7 pixels of the maximum cluster. 6-If the Max class and Fit class are the same, we will start embedding the equivalent text Aski codes from 8^{th} pixel belonging to Max class. Otherwise, we start embedding the information from the first pixel belonging to Fit class, i.e. the first pixel of the grayscale image and we check pixel by pixel and line by line and if there is a pixel belonging to the class Fit, we use the less-valued three bits equivalent in the color image to embed the information. For instance imagine that the pixel of the target class is in row 1 and column 2. We use the pixel which is located in the row 1 and column 2 of the color image to embed the information.

We send the stego image to the receiver after embedding the information. The receiver of the color image, finds its equivalent grayscale image, he calculates the length of the message and class which is used for embedding the information after finding the maximum class and exploiting the information of the first 7 pixels belonging to it and then he exploits the information by using the pixels of the image belonging to the target class. Exploiting the information is continued until it reaches eight times as long as the message length and we

consider the information eight bits by eight bits and we calculate their equivalent decimal and specify their character.

3- Test result. We have applied the test results for hiding this text "This is a secret message" in an image by MATLAB software. The stego and cover image, histogram of the three factors of the color image and its grayscale image before and after steganography of it is shown in figure 1.

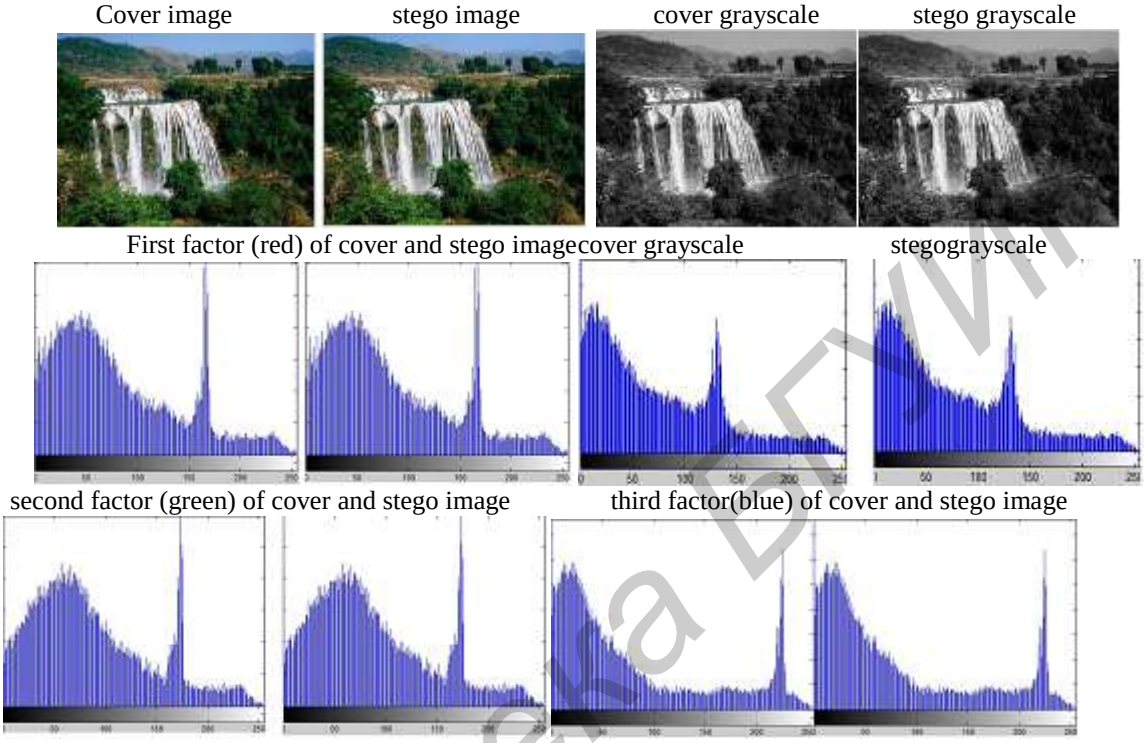

Figure 1 – Cover image, Stego image, cover grayscale, stego grayscale and their histograms

4- Conclusions. n this article, we used clustering the histogram to specify some of the color image pixels in order to embed the data and specify the proper class with respect to the message length. The results showed little changes in the histogram. The following researches can be followed: 1-Using two LSB, i.e. dividing the image pixels to 64 clusters. 2-Reviaing and comparing the histogram of three factors (red, blue and green) and using one of them for embedding.

Literature

1. R. S. Gutte, Y. D. Chincholkar," Comparison of Steganography at One LSB and Two LSB Positions", International Journal of Computer Applications (0975 – 8887) Volume 49– No.11, July 2012

2. Chi-Kwong Chan, L.M. Cheng,"Hiding data in images by simple LSB substitution", Pattern Recognition 37 (2004) 469 – 474