PIXEL INDICATOR HIGH CAPACITY TECHNIQUE FOR RGB IMAGE BASED STEGANOGRAPHY

Adnan Gutub, Mahmoud Ankeer, Muhammad Abu-Ghalioun, Abdulrahman Shaheen, Aleem Alvi

Computer Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

ABSTRACT

Image based steganography uses the images as the cover media. LSB is a commonly used technique in this filed. Several scenarios of utilizing least significant bits within images are available. We merge between the ideas from the random pixel manipulation methods and the stego-key ones to propose our work, which uses the least two significant bits of one of the channels to indicate existence of data in the other two channels. This work showed attractive results especially in the capacity of the data-bits to be hidden with relation to the RGB image pixels.

Keywords: Steganography, RGB Bitmaps, Pixel Indicator Algorithm.

1. INTRODUCTION

Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless messages [6,8]. Steganography, meaning *covered writing*, dates back to ancient Greece, where common practices consisted of etching messages in wooden tablets and covering them with wax, and tattooing a shaved messenger's head, letting his hair grow back, then shaving it again when he arrived at his contact point. Different types of Steganographic techniques employ invisible inks, microdots, character arrangement, digital signatures, convert channel, and spread spectrum communications.

Steganography techniques require two files: cover media, and the data to be hidden [7]. When combined the cover image and the embedded message make a stego-file, which is in our work for image steganography known as stego-image image [3]. One of the commonly used techniques is the *LSB* where the least significant bit of each pixel is replaced by bits of the secret till secret message finishes [2,4,5,6]. The risk of information being uncovered with this method as is, is susceptible to all 'sequential scanning' based techniques [1], which is threatening its security.

The random pixel manipulation technique attempts at overcoming this problem, where pixels, which will be used to hide data are chosen in a random fashion based on a stego-key. However, this key should be shared between the entities of communication as a secret key. Moreover, some synchronization between the entities is required when changing the key [1]. This will put key management overhead on the system. Another technique, is the Stego Color Cycle (SCC). This SCC technique uses the RGB images to hide the data in different channels. That is, it keeps cycling the hidden data between the Red, Green and Blue channels, utilizing one channel at a cycle time. The main problem of this technique is that, hiding the data in the channels is done in a systematic way. So, being able to discover the data in the first few pixels will make the discovery of the technique easy. StegoPRNG is also a different technique that uses the RGB images. However in this technique, a pseudo random number generator (PRNG) is used to select some pixels of the cover image. Then, the secret will be hided in the Blue channel of the selected pixels. Again this technique has the problem of managing the key, and problem of capacity since it uses only the Blue channel out of the three channels of their available channels [6]. Our suggested technique tries to solve the problem of the previous tow techniques by using one of the channels as an indicator for data existence in the other two channels and the indicator is set randomly by nature.

The flow of the paper is as follows. Section 2 lists the three main requirements for designing stego-algorithms. Section 3 discusses our new proposed pixel indicator stego-technique. Modeling the algorithm and testing its results are provided in Section 4. Section 5 presents comparisons and conclusion remarks of the work.

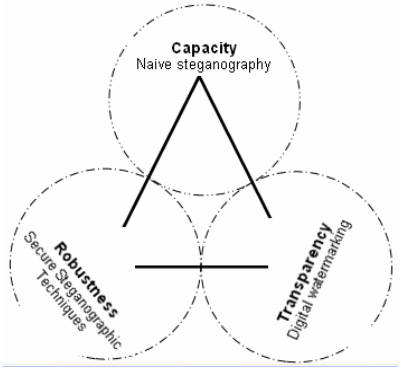

Figure 1. Steganography tradeoff parameters

2. REQUIREMENTS

Designing any stego algorithm should take into consideration the following three aspects [8] (Figure 1):

- Capacity: The amount of data that can be hidden without significantly changing the cover medium.
- Robustness: the resistance for possible modification or destruction in unseen data.
- Invisibility (Security or Perceptual Transparency): The hiding process should be performing in a way that it does not raise any suspicion of eavesdroppers.

Figure 1, in the previous page, shows the relation between these three main parameters. If we increase the capacity of any cover to store more data than a practical possible threshold, then its transparency or robustness will be affect and visa versa. Similarly, transparency and robustness are related; if any of these two parameters are influenced, it can affect the performance in the other one. The capacity, robustness, and security parameters relation issues can be driven by the application need and its priorities.

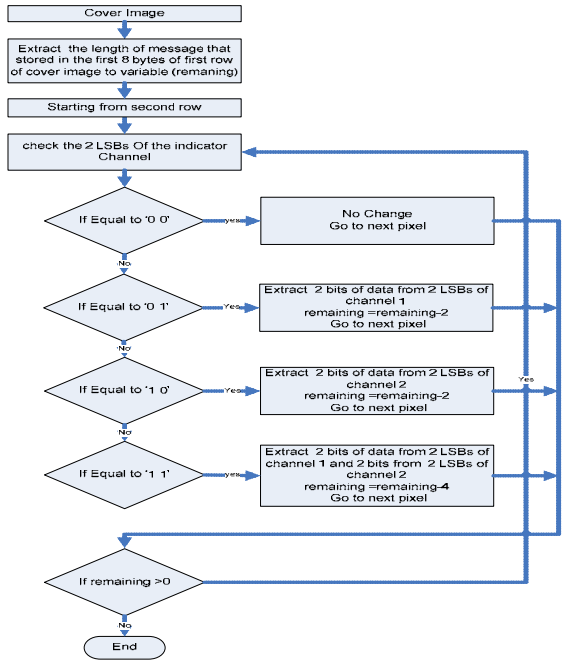

Figure 2. Hiding Process flowchart

3. PIXEL INDICATOR TECHNIQUE

We propose this pixel indicator technique for RGB images steganography. The technique uses least two significant bits of one of the channels Red, Green or Blue as an indicator of data existence in the other two channels. The indicator bits are set randomly (based on the image nature) in the channel. Table 1 shows the relation between the indicator and the hidden data inside the other channels.

To improve security, the indicator channel is not fixed. The indicators are chosen based on a sequence. In the first pixel Red is the indicator, while Green is channel 1 and Blue is the channel 2. In the second pixel, Green is the indicator, while Red is channel 1 and Blue is channel 2. In third pixel Blue is the indicator, while Red is channel 1 and Green is channel 2. The sequence of the algorithm is flowcharted in Figure 2. The recover algorithm (Figure 3) will stop based on the length of the secret message, which is stored in the first 8 bytes of the cover image.

Table 1. Meaning of indicator values.

Indicator	Ch 1	Ch 2
00	No hidden data	No hidden data
01	No hidden data	2bits of hidden data
10	2bits of hidden data	No hidden data
11	2bits of hidden data	2bits of hidden data

4. MODELING & SIMULATION

This section of the paper describes the method that was used to test the algorithm and the results. Figure 4, a BMP image of size 512 × 384 (196,608 pixels), have been used to hide text message of 11,733 characters length (that is 93,864 bits).

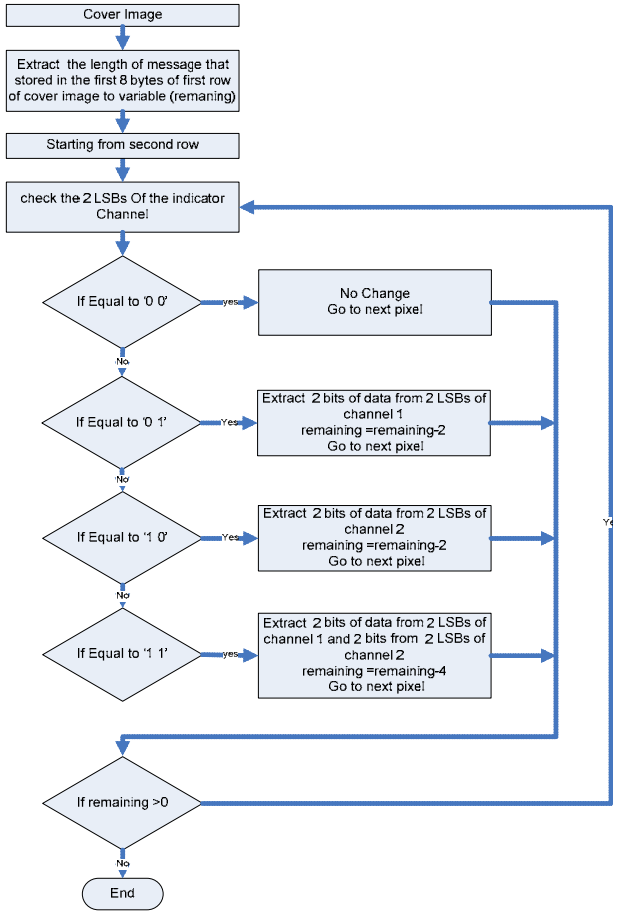

Figure 3. Recovery Process flowchart

The randomization is in the indicator channel content, which gave different results in every run based on the starting pixel to hide data. After many runs of the algorithm we came up with some results. Frequency analysis, histogram, was performed to check the change

in the cover image. Also, the number of pixels used in each run to hide data was recorded. Note that the visual change between the original image and stego-image cannot be predicted (Figure 4). However, the differences between the images before and after hiding the data can be sensed through histograms. Histograms of the RGB channels: Red, Green, and Blue, are tested separately to study the method security.

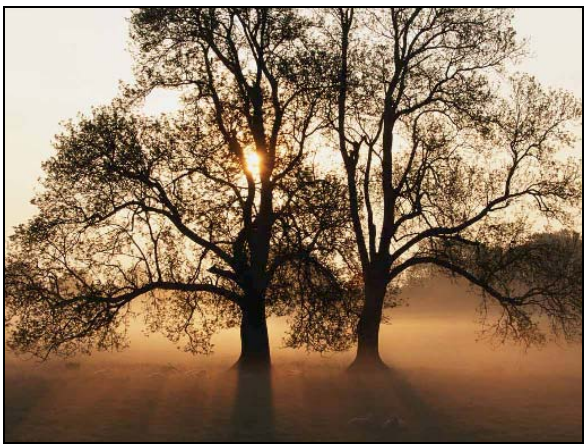

Figure 4. BMP cover image, no difference can be detected before & after imbedding secret data with our stego-technique

Figures 5 and 6 show the histogram of the Red channel pre and post the hiding process. Similarly, Figures 7 and 8 show the histogram of the Green channel pre and post the hiding process. Also, Figures 9 and 10 show the histogram of the Blue channel pre and post the hiding process.

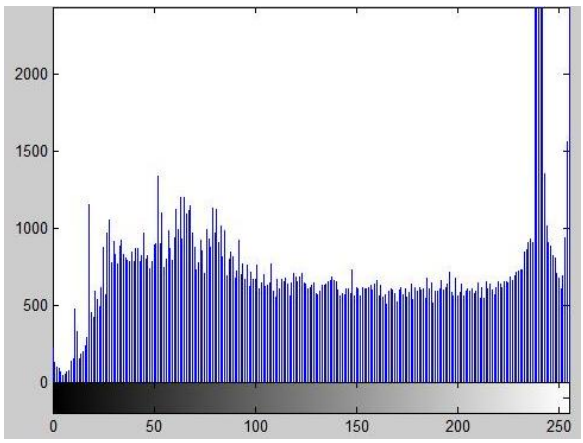

Figure 5. Original image histogram of the red channel

5. COMPARISON CONCLUSIONS

By comparing the two histograms of the three RGB channels before and after hiding data, some security feedback can be concluded. Observe the red and green channel histograms before and after the modification (Figures 5 & 6, and Figures 7 & 8), the change cannot be detected; both, Red and Green, channels can be used to give some security promise. However, changes can be detected clearly in the histogram of the blue channel, which puts some future work to investigate the reasons and implications of this issue.

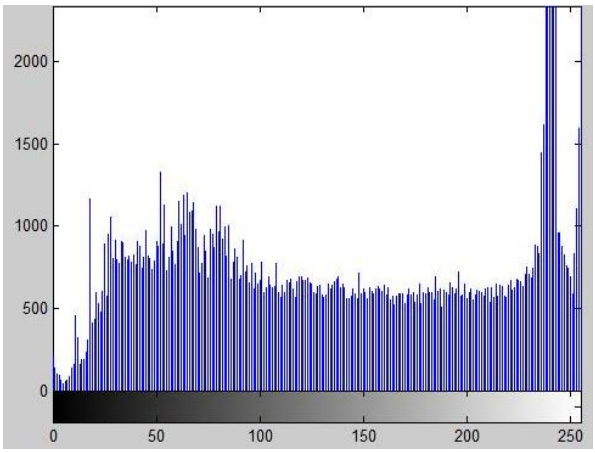

Figure 6. Modified image histogram of the red channel

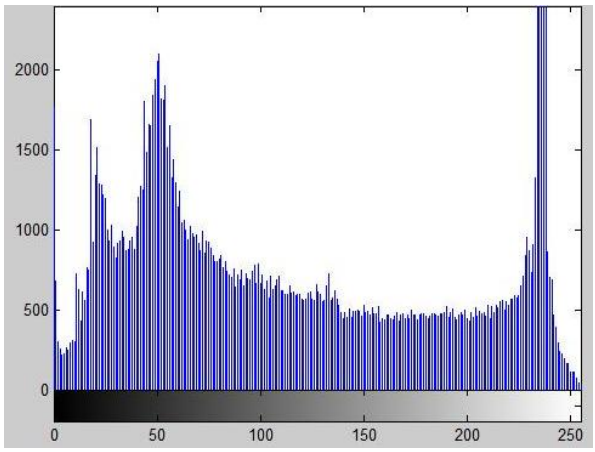

Figure 7. Original image histogram of the green channel

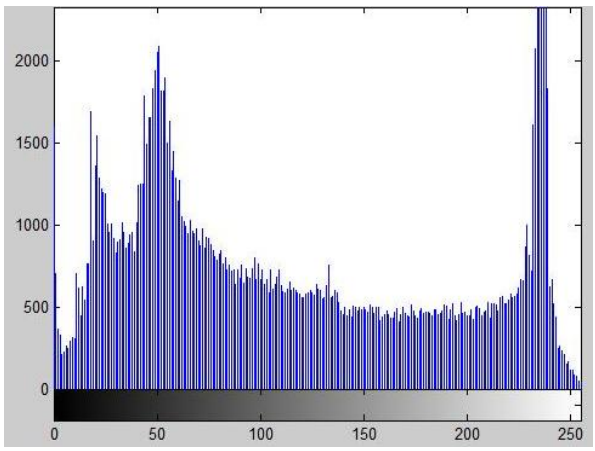

Figure 8. Modified image histogram of the green channel

Interestingly, from the different test runs, we got different distributions between the three channels, which continued varying between the channels with no

apparently detected pattern. This undetectable pattern changing within RGB channels gave us the promise that our proposed technique may be considered random or pseudorandom based on the randomness of the indicator channel.

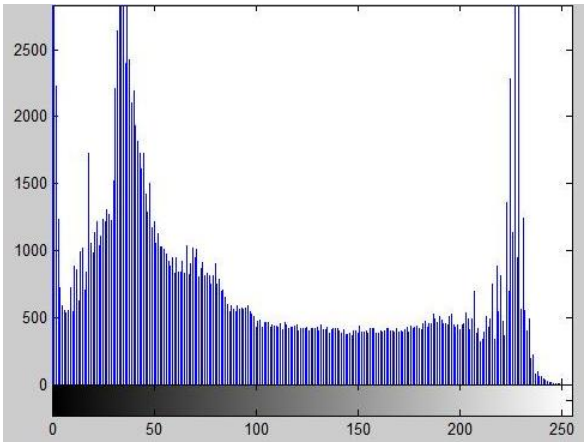

Figure 9. Original image histogram of blue channel

The different testing results are listed in Table 2, which shows the randomness in the results. This sample study gave us hint of the capacity of our proposed algorithm. Hiding the text message of 11,733 characters length (93,864 bits) within the image of 196,608 pixels utilized less than fourth of these pixels. It can be roughly concluded for the algorithm capacity that a pixel is needed to hide every two bits.

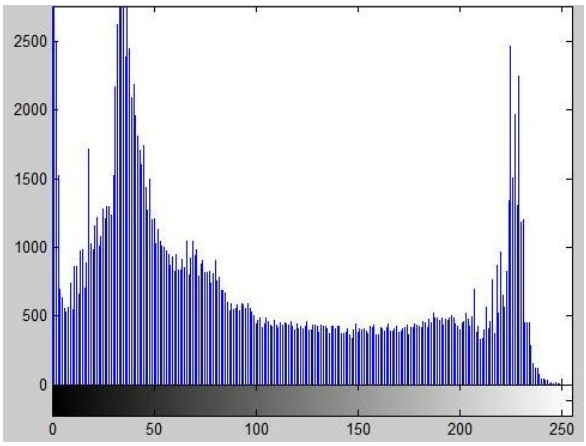

Figure 10. Modified image histogram of blue channel

The robustness of our proposed algorithm is not investigated thoroughly. The first impression about robustness is that it is achieved electronically, as long as the image is not modified or compressed. If this robustness issue is true, it can be considered as clear drawback of the proposed method needing more future study. In general, this proposed algorithm may open new directions in steganography research leading to interesting results.

Table 2. Capacity of cover image

Fixed Total number of pixels of cover image	Used pixels for hiding text message of 93,864 bits	Percentage of unused pixels
196,608	46,880	76.16 %
196,608	47,004	76.09 %
196,608	51,679	73.72 %
196,608	47,108	76.04 %
196,608	46,751	76.22 %
196,608	46,907	76.14 %
196,608	46,807	76.19 %
196,608	46,850	76.17 %
196,608	47,305	75.94 %
196,608	46,699	76.25 %

ACKNOWLEDGMENTS

The authors would like to thank the Computer Engineering Department within King Fahd University of Petroleum and Minerals (KFUPM), in Dhahran-Saudi Arabia, for supporting this work.

REFERENCES

[1] S. Venkatraman, A. Abraham, M. Paprzycki, "Significance of Steganography on Data Security", *International Conference on Information Technology: Coding and Computing (ITCC'04)*, Las Vegas, 5-7 April 2004.

[2] Kathryn Hempstalk, "Hiding Behind Corners: Using Edges in Images for Better Steganography", *Proceedings of the Computing Women's Congress*, Hamilton, New Zealand, 11-19 February 2006.

[3] N.F. Johnson, S. Jajodia, "Exploring Steganography: Seeing the Unseen", *IEEE computer*, Vol. 31, No. 2, pages 26-34, February 1998.

[4] G.C. Kessler, "An Overview of Steganography for the Computer Forensics Examiner", *Forensic Science Communications*, Vol. 6, No. 3, July 2004.

[5] D. Artz, "Digital Steganography: Hiding Data within Data", *IEEE Internet Computing: Spotlight*, pages 75-80, May-June 2001.

[6] K. Bailey, K. Curran, "An Evaluation of Image Based Steganography Methods", Multimedia Tools & Applications, Vol. 30, No. 1, pages 55-88, July 2006.

[7] A. Gutub, L. Ghouti , A. Amin, T. Alkharobi, M.K. Ibrahim, "Utilizing Extension Character 'Kashida' With Pointed Letters For Arabic Text Digital Watermarking", *Inter. Conf. on Security and Cryptography - SECRYPT*, Barcelona, Spain, July 28 - 31, 2007.

[8] A. Gutub, M. Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", *WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE)*, Vienna, Austria, May 25-27, 2007.