Journal of Information and Operations Management
ISSN: 0976–7754 & E-ISSN: 0976–7762 , Volume 3, Issue 1, 2012, pp-265–269
Available online at http://www.bioinfo.in/contents.php?id=55

STEGANOGRAHY IN RGB IMAGES WITH IMPROVED INTENSITY EQUALIZATION OF PIXELS

MANISH MAHAJAN[1]* AND NAVDEEP KAUR[2]

Department of IT at CEC , Landran Mohali (Punjab), INDIA
*Corresponding Author: Email- manishmahajan4u@gmail.com

Received: December 12, 2011; Accepted: January 15, 2012

Abstract- Steganography is the process of hiding one message or file inside another message or file. For instance, steganographers can hide an image inside another image, an audio file, or a video file, or they can hide an audio or video file inside another media file or even inside a large graphic file. Steganography differs from cryptography in that while cryptography works to mask the content of a message, steganography works to mask the very existence of the message. With the war on terrorism and the hunt for those responsible for the September 11 attacks mounting, steganography is increasingly in the news. Some experts theorize the al Qaeda terrorists used the Internet to plan the attacks, possibly using steganography to keep their intentions secret [13]. The aim of this study was to investigate the various steganography methods & how they are implemented .LSB is a very well known method in this field. In binary images we are very much restricted in the scope as there are only 4 bits or 8 bits to represent a pixel so we are very much restricted to most popular LSB methods .But in colored images there are generally up to 24 bits images with three different RGB channels, if using RGB color space .So, we can explore a lot many new methods which can manipulate or use various channels of colored images in regular or arbitrary pattern to hide the information. Using this concept we have explored the various existing methods of data hiding in colored images & taken an intersection between the arbitrary pixel manipulation & LSB method to propose our work which uses arbitrary channel of a pixel to reflect the presence of data in one or two other channels. We are sure that this work will show an attractive result as compared to the other present algorithms on the various parameters like security, imperceptibility capacity & robustness.
Keywords- Steganography, stegoimage, information reflector

Citation: Manish Mahajan and Navdeep Kaur (2012) Steganograhy in RGB Images With Improved Intensity Equalization of Pixels. Journal of Information and Operations Management ISSN: 0976–7754 & E-ISSN: 0976–7762, Volume 3, Issue 1, pp-265-269.

Introduction

The word steganography means "covered or hidden writing" [9, 14]. The object of steganography is to send a message through some innocuous carrier (to a receiver while preventing anyone else from knowing that a message is being sent at all. Computer based stenography allows changes to be made to what are known as digital carriers such as images or sounds. The changes represent the hidden message, but result if successful in no discernible change to the carrier. The information may be nothing to do with the carrier sound or image or it might be information about the carrier such as the author or a digital watermark or fingerprint [9].

Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions [9]. When the message is hidden in the carrier a stego-carrier is formed for example a stego-image. Hopefully it will be perceived to be as close as possible to the original carrier or cover image by the human senses. Images are the most widespread carrier medium [10]. They are used for steganography in the following way. The message may firstly be encrypted. The sender (or embedder [12]) embeds the secret message to be sent into a graphic file [11] (the cover image [12] or the carrier). This results in the production of what is called a stego-image. Additional secret data may be needed in the hiding process e.g. a stegokey. The stego-image is then transmitted to the recipient [11]. The recipient (or extractor [12]) extracts the message from the carrier image. The message can only be extracted if there is a shared secret between the sender and the recipient. This could be the algorithm for extraction or a special parameter such as a key [11] (the stegokey). A stegoana-

lyst or attacker may try to intercept the stego-image. Figure 1 below shows the steganographic system. 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 [2,4,5,6,3]. The risk of information being uncovered with this method as is very much prone to 'sequential scanning' based techniques [1], which are 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. 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 [6,7,8]. Our suggested technique tries to solve the problem of the previous techniques by using an arbitrary channel of a pixel to act as an information reflector which reflects the presence of data in one or two other channels.

The flow of this paper is as follows: Section 2 represents the various parameters that should be taken into consideration while designing a technique for steganograhy. Section 3 gives as outline about the proposed method.

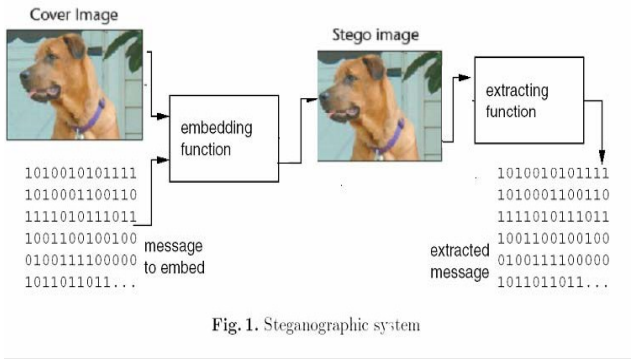


Fig. 1. Steganographic system

Fig. 1- Steganographic Systems

Parameters

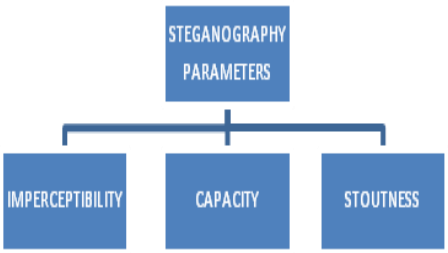


Fig. 2- Various Steganography parameters

The above outlined fig 2 shows the various parameters that are responsible for deciding the strength of some steganography technique. These 3 parameters are not independent rather they are very much dependent upon each other .For e.g. if we want to increase the imperceptibility of a stego image then we will choose some random pixels to store the secret message but due to this the capacity of stego image will definitely be decreased & the stoutness will also increase to some extent.

Problem Definition

In steganography the main objective of any algorithm is to minimize the perceptibility of secret information in image by an observer. Many authors have tried to improve the image imperceptibility factor after hiding the secret message. One such effort came into literature by Jae-Gil Yu et al. in their work titled "A New Image Steganography based on 2k Correction and Edge-Detection". They have used a method named 2k correction to improve the image after hiding the secret data. Firstly we will try to understand what 2k correction is with a detailed example given below:

A. Detailed example of 2^k correction

Actual pixel value (APV)175= 10101111
Stego pixel value (SPV) 169= 10101001
Error value |175-169| = 6
If Error value<=2^{k-1}
No need to change
Else //if error value is>2^{k-1} then
New stego pixel value = Either SPV - 2^k OR SPV + 2^k
Which ever is close to APV
In our case Error value =6 > (2^{3-1} = 4) So
New stego pixel value= Either SPV – 2^3 OR SPV + 2^3
Which ever is close to APV
= (169-8 =161) OR (169+8 =177)
Which ever is close to 175
= 177 (10110001)

In this way the 2^k correction makes the intensity of the channel nearer to the actual pixel value without affecting the secret data.

We have analyzed their algorithm in very detail & found certain problems with the algorithm in using it in steganography applications using gray scale or colored images.

B. Three main problems of 2k correction technique with edge detection are:

We cannot apply 2k correction if one of the MSBs is being changed due to theapplication of 2k correction for e.g.

27(00011011)-------32(00100000).....(24-39)
58(00111010)--------66(01000010).....(56-71)
92(01011100)--------100(01100100).....(88-103)
125(01111101)--------133(10000101).....(120-135

Because these will be used at receiving end to determine the edge regions with the help of edge detection algorithm.

In this way 7 cases each containing 15 values are there for which apply it is difficult to 2k correction So 15*7=105values.

We cannot apply 2k correction if corrected value is crossing the threshold value(i.e.90) 92------87 or (88-----91)

Because this threshold value will be used at receiving end to dete mine the no of bits replaced in edge region (whether 3 bits or not) so it cannot be changed. In this way there are 15 values from (83-----98) for which it is difficult to apply 2k correction. But most values area already considered in Ist case so only 5 va ues for this case.(83—87)

Other than these cases boundary values(16 values) are there for

which we need to take care that after 2k correction value should not go out of range 0-255.So at values (0-7) & (248-255) it is diff cult to apply 2k correction.
So in total for 3 cases 105+5+16 = 126 values

Proposed Work

I. Analyzing root cause of 2k correction problem & conceiving a new idea for its solution

The main problem of 2k correction is because it is using the same channel for data & for the detection of edges which ultimately decides the no of bits in a channel
For e.g.
11000011 if 3 MSb(110) decides about the edge region & no of data bits in LSB(011) then after adding secret data while applying 2k correction we need to take care that MSB should not be defected because same LSB will be used while decoding. So as mentioned above in most of the cases we will not be able to apply 2k correction.
The main reason for this efficiency is there in the channel if we will see carefully i.e. it is using same channel for edge detection & for data hiding. So they are dependent upon each other. For e.g.11000011 if 3 MSB(110) bits decides about the edge region & no of data bits that can be embedded in LSB(011) then after adding secret data bits while applying 2k correction we need to take care that MSB should not be altered because same MSB will be used while decoding. So as mentioned above in most of the cases we will not be able to apply 2k correction. The main reason for this inefficiency is there in the channel itself if we will see carefully i.e. it is using same channel for edge detection & for data hiding. So they are dependent upon each other.
So the clear solution that came to anyone's mind is that we should use different channels for reflection of information & for hiding secret data. For this purpose there should be at least 2 channels in a pixel so we can implement a possible solution on 24 bit, 32 bit or 40 bit image. But in this research work we have used 24bit colored image. When we analyze the 24 bit image it always carries three components that is red, green & blue component in RGB color model as shown in fig. 3 below.

Fig. 3- Structure of a RGB pixel

The three layer security model, shown in fig 4 below, of our proposed technique adds towards the good imperceptibility factor of our technique.

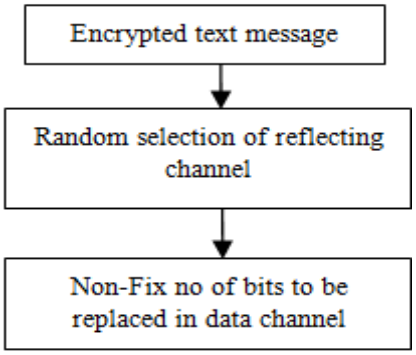


Fig. 4- Three layer security of Proposed Technique

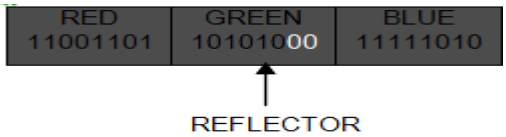
II. Proposed Random Channel Steganography Algorithm

Firstly message to be hidden will be taken by the user followed by the path of image which will be used as cover image & the path of the image that will act as the output image. Then the characters of message will be converted to its ascii value. After this we will encrypt these ascii values with the help of some encryption algorithm. Then these ascii values will be converted to 8 bit binary code & will be stored in a linear array or vector.
To implement the proposed method we use one of the RGB channels of a pixel of a colored image as a reflector to represent whether the hidden information bits are present in one or both of the other channels. Firstly we will read the input image in a three dimensional array & then each pixel will be taken one by one & information reflector will be chosen for that pixel. To choose the information reflector channel we will use the pseudo random number generator (PRNG) based on certain seed shared between transmitter & receiver end. Then the modulus of the number, generated by PRNG, with 3 will give information that whether the information reflector channel is red, green or blue. In this method firstly we check the 2 least significant bits of the reflector channel, if it is 11 it means the secret text is in both the other channels so we can insert or retrieve the least 2 or 3 significant bits of both the channels in the order first channel followed by 2nd channel. But if the least significant bit of reflector channel is 00, it means that the pixel does not contain any secret data. On the other hand if the value of 2 LSBs of reflector channel is 01 or 10 it means that the secret text is only in one of the other channels. If the value is 01 then the data is in first channel & if the value is 10 then the data is in 2nd channel in accordance to table1. But in both the cases data channel can carry 2 or 3 bits of secret data. To decide the no of bits in a data channel we will look upon the original decimal value of data channel if it is less than or equal to certain threshold value (i.e. Global Adaptive Thresholding Using Simple Image Statistics (SIS) [17,18] in our case) then the least 3 significant bits of data carrier channel will be replaced by secret data else if the value is greater than threshold value then the least two significant bits will be replaced by secret data.

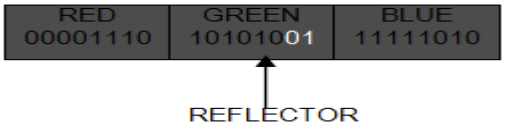
TABLE 1 Relation Between Reflector & Other Two Channels

REFLECTOR	FIRST CHANNEL	2ND CHANNEL
RED	GREEN	BLUE
GREEN	RED	BLUE
BLUE	RED	GREEN

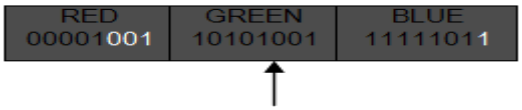
Application Example of the method
Let Secret message:-10110011100001101000101011010 0
First Pixel



2 LSBs = 00 So from Table 1, no secret data in the pixel
Remaining message: - 00111000011010001010110100
Next Pixel



2LSBs = 01 so from Table 1, red channel will carry data as decimal value of red channel is 14 that is less than threshold value (63) so red channel will carry 3 bits of data. Also modify the LSB of blue channel to reflect 3 bits of secret data in red channel.



As the difference in values after insertion is 5 which is greater than $4(2^{k-1})$ so we need to apply 2^k correction.00001001=9
9+8=17 = 00010001 which is nearer to original value that is 14 as compared to 9.

In this way message will be hidden in every Pixel of an image till the end of the message.

Results

A. Capacity & PSNR

We have analyzed the capacity of the method in terms of Bits per pixel (BPP).Additionally we have also tried to analyze the %ge pixels used & %ge capacity. We have constantly hidden the 3896 bits in various images of various resolutions & we found that bits per pixel, capacity & quality parameters are almost independent from the resolution of the image. For calculating bits per pixel & capacity we have used the following formulas:

BPP = (number of secret bits hidden / number of pixels used)

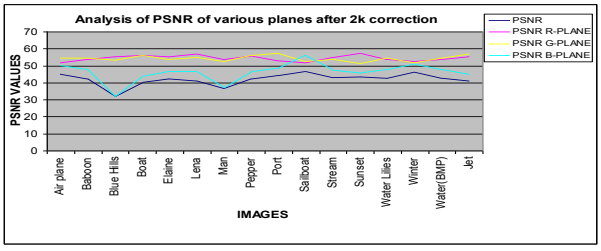
Capacity = (number of secret bits hidden / number of pixels used * 24) *100

Results are summarized in given table 2 below.

Table 2- Results of Various Parameters on Different Image Sets

Image	Resolution (m x n)	Bits per pixel (Bpp)	Percent capacity (%ge)	Percent pixels used (%ge)
Air plane	150 x 100	2.68504	11.1877	9.673300
Baboon	131 x 131	2.25463	9.39429	10.06930
Blue Hills	800 x 600	3.68531	15.3555	0.219792
Boat	127 x 88	2.03659	8.48580	17.11700
Elaine	105 x 135	2.27703	9.48763	12.07050

Also the PSNR values for planes of an image has been analyzed & it is found that B-plane is dominating in deciding the overall PSNR of the image. Results are summarized in given graph 1 below. Due to insufficient space we are presenting the PSNR values graphically instead of table.



Graph. 1- PSNR values of various planes of images with 2k correction

B. Visual Analysis of Cover image & Stego image

We have also compared the original cover image & stego image for the proposed steganography technique with & without 2k correction. The proposed technique is good enough that there is almost unnoticeable difference between original & stego image. This is not the case of any single image rather we have performed the analysis for 16 different image sets of different resolutions & different sizes. But the results of visual analysis are constantly unnoticeable by human naked eye for both with or without 2k correction. In fact stego image has improved a lot with 2k correction. Here we are showing the results of single image set of famous baboon image(from fig. 5 to fig. 12) of size 5.31KB & with resolution 131 X 131 for both 2k correction & without 2k correction.

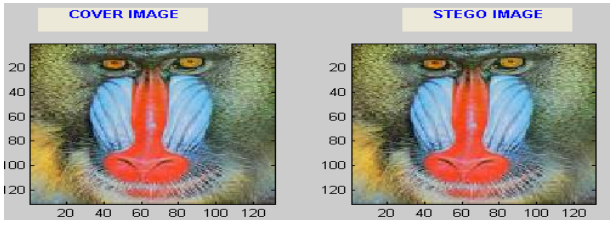


Fig. 5- Cover image & Stego image with 2k correction

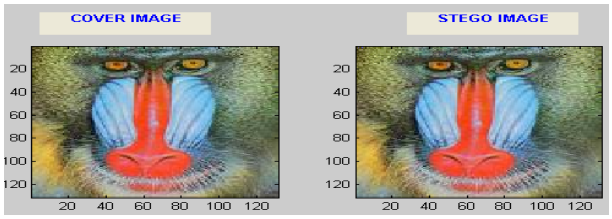


Fig. 6- Cover image & Stego image without 2k correction

We can see that there is not much difference in original & stego image for any image. But that is also very much improved after 2k correction.

C. Comparisons with Other Techniques:

We have compared our proposed technique with famous LSB substitution technique, Tseng Cheng technique & NPI technique [15] on the very important parameter of PSNR for four famous image sets of lena, pepper, baboon & jet of different sizes as shown in table 3 below.

Table 3- Comparison of proposed technique with previous techniques for PSNR

Image	Resolution (m x n)	NPI Technique (PSNR)	Tseng Cheng Technique (PSNR)	LSB Technique (PSNR)	Proposed Technique (PSNR)
Lena	512 x 512	42.590	40.055	41.005	45.829
Pepper	125 x 129	43.924	39.897	42.374	45.760
Baboon	131 x 131	38.364	31.384	33.988	46.111
Jet	300 x 388	42.505	38.287	39.383	44.579

As shown in table 3 our technique is better than previous techniques for PSNR values.

We have also compared our proposed technique with Chang-Chin technique [16] for the parameter of bits per pixel as shown in table 4 below.

Table 4- Comparison of proposed technique with Chang-Chin technique for BPP

Image	Max DHC using Chang-Chin Technique (Bits)	Bits Per Pixel using Chang-Chin Technique	Bits Per Pixel using Proposed Technique
Lena	64876	0.24748	2.12432
Pepper	102023	0.38919	1.86144
Baboon	63205	0.24111	2.25463

From the above table it is clear that our proposed technique is better than Chang-Chin technique in hiding number of bits per pixel.

Conclusion

We have tried to check this method on basic goodness criteria of any steganography algorithm i.e. capacity, imperceptibility & stoutness. Basically the technique is proposed by keeping in mind about the safe capacity rather than about capacity which is justified with the improved values of MSE & PSNR.

References

[1] Fridrich J., Goljan M. Binghamton (2002) *Conferenc, San Jose CA, ETATS-UNIS*.

[2] Kevin Curran (2003) *International Journal of Digital Evidence* Fall, Vol 2, Issue 2

[3] Thampi S.M. (2004) *ISTE-STTP on Network Security & Cryptography,* LBSCE.

[4] Kefa Rabah (2004) *Information Technology Journal* 3 (3): 245 -269, ISSN 1682-6027.

[5] Hsien-Wen Tseng and Chin-Chen Chang (2004) *Fourth International Conference on Computer and Information Technology*.

[6] Ching Yu Yang (2007) *Third International Conference on International Information Hiding and Multimedia Signal* Processing.

[7] Moon S.K., Kawitkar R.S. (2007) *International Conference on Computational Intelligence and Multimedia Applications*.

[8] Nameer N. EL-Emam (2007) *Journal of Computer Science* 3 (4): 223-232, ISSN 1549-3636.

[9] Sahoo G. and Tiwari R.K. (288) *International Journal of Computer Science and Network Security*, Vol.8 No.1.

[10] Nameer N. EL-Emam (2008) International *Journal of Signal Processing* 4; 2.

[11] Jae-Gil Yu, Eun-Joon Yoon, Sang-Ho Shin andKee-Young Yoo (2008) *Fifth International Conference on Information Technology*.

[12] Junhui He, Shaohua Tang and Tingting Wu (2008) *Congress on Image and Signal Processing, Steganographic technique is a means of covert communication*. Vol 5, Issue, 27-30

[13] Rafel C. Gonzalez and Richard E. (2002) *Digital Image processing second edition Woods.*

[14] Ali Shariq Imran, M. Younus Javed and Naveed Sarfraz Khattak (2007) *International Journal of Computer Science and Engineering* 1;3.

[15] Tsai C.L., Kuo-Chin Fan, Thomas Chiang Chuang And Char-Dir Chung (2007) *Journal Of Information Science And Engineering* 23, 1481-1498.

[16] Kittle J.J. Illingworth and Foglein J. (1985) *Computer Vision, Graphics & Image Processing*, Vol. 30, 125-147.

[17] Yahia S. Halabi, Zaid SASA, Faris Hamdan, Khaled Haj Yousef (2009) *European Journal of Scientific Research*, ISSN 1450-216X Vol.28 No.1 (2009), pp.14-32.