Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol.6 No.2, 2017, pp.14-17

DOI: https://doi.org/10.51983/ajeat-2017.6.2.822

A Capable System for Hiding Image in an Image

R.Ganesh[1] and S.Thabasu Kannan[2]
[1]Research Scholar and Assistant Professor, Department of Computer Science,
NMSSVN College, Madurai, Tamil Nadu, India
[2]Principal, Pannai College of Engineering and Technology, Sivagangai, Tamil Nadu, India
E-Mail: yourmrganesh@gmail.com, thabasukannan@gmail.com
(Received 15 June 2017; Revised 29 June 2017; Accepted 15 July 2017; Available online 22 July 2017)

***Abstract* - In this paper, we have proposed a new method by for hiding an image using steganographic called GanKan. This method is based on Least Significant Bit (LSB) replacement and can be used for 24 bit color image capable of producing a secret-embedded image. This paper clearly showed that this GanKan new method is better than LSB technique for 8 bit color image and 24 bit color image. Firstly LSB method for both 8 bit and 24 bit color image are described and then the GanKan method for 24 bit color image, compare their result by calculating Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Root Mean Square Error (RMSE) and histogram analysis. LSB Algorithm embedded MSB of secret image into LSB of cover image. In the case of 24 bit color image two methods are described. In first method, last 2 LSB of each plane (RGB) of cover image, is replaced by 2 MSB of secret image. In the second method, last LSB of each red plane is replaced by first MSB of secret image, last 2 LSB of each green plane by next 2 MSB of secret image and then last 3 LSB of blue plane is replaced by next 3 MSB of secret image. This means that total 6 bits of secret image can be hided in 24 bit color image. Experimental results showed that the embedded-image is visually indistinguishable from the original cover-image in the case of 24 bit.**
Keywords: **LSB, MSB, RMSE,MSE, PSNR**

I. INTRODUCTION

In the modern world of digital communication, there are several techniques used for hiding information in any medium. One of such technique is steganography in which digital images are used as a medium for hiding information and the information in the form of text, digital image, video or audio file may be used as secret message. It is used to increase the level of communication security by inserting secret message into the digital image, modifying the redundancy or nonessential pixels of the image [3], and is recently become important in a number of application areas especially military and intelligence agencies which require modest communications. In case of 24 bit color image each pixel has three color components: Red, Green, and Blue (RGB). Each pixel is represented with three bytes to indicate the intensity of these three colors (RGB). Cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret [4]. Encryption encodes the data into an unreadable format called cipher so that an unintended recipient cannot determine its intended meaning. Steganography attempts to prevent an unintended recipient from suspecting that the data is there [5]. Nowadays, using a combination of steganography and the cryptography, information security has improved considerably. Steganography is used in many fields such as copyright, preventing e-document forging. A simple and well known method is directly hiding secret data into the least-significant bit (LSB) of each pixel in an image.

II. 8 BIT COLOR IMAGE

8 bit color system is very limited but true direct color system, there are three bits (2^3= 8 possible levels) for each of the R and G components, and the two remaining bits in one byte pixel to the B component (four levels), enabling 256 (8 × 8 × 4) different colors. Since normal human eye is less sensitive to the blue component than to the red or green, so it is assigned one bit less than the both. There are two forms of 8- bit color graphics. The most common graphics uses a separate palette of 256 colors, where each of the 256 entries in the palette maps to given RGB values. In most color maps, each color is usually chosen from a palette of 1,67,77,216 colors (24 bits: 8 R, 8 G, 8 B). The other form is one in which the 8 bits directly describe RGB values, typically with 3 bits for R, 3 bits for G and 2 bits for B. This second form of color graphics is often called 8-bit true color, as it does not use a palette at all.

III. 24 BIT COLOR IMAGE

24 bit color image is defined by RGB color model in which each color appears in its primary spectral component of RGB. This model is based on Cartesian coordinate system. In which RGB primary value are at three corner, the secondary color cyan , magenta and yellow are at three other corner, black is ate the origin and white is at the corner farthest from the origin. Line joining the two corners has equal values for RGB. This produces various shades of grey. The focus of all these points is called the grey line. In RGB model, each pixel is composed of RGB values and each of these colors requires 8-bit for its representation. Hence each pixel is represented by 24 bits. So total number of color possible with 24-bit RGB image is $(2^8)^3$ = 16,777,216

IV. LEAST SIGNIFICANT BIT

In LSB technique, the message bits are embed in LSB of cover image. In LSB steganography, the LSBs of the cover

media's digital data are used to conceal the secret message [6]. LSB Steganography can be classified by two methods LSB replacement and LSB matching.

LSB replacement steganography replace the last bits of cover image with each bits of the message that needs to be hidden and in LSB matching [8] each pixel of the cover image is taken mainly in a pseudo-random order which is generated by a secret key, if the LSB of the cover pixel matches the bit of secret data no changes are done otherwise, one is added or subtracted from the cover pixel value, at random. If the length of secret message contains fewer bits than the number of pixels in the cover image, changes are spread uniformly throughout the image by pseudo-random permutation. Since there is change of each bits by so the degradation of cover image caused by this embedding process would be perceptually transparent. In LSB of 24 bit color image, the LSB of each pixel of a specific color channel or all color channels are replaced with a bit from the secret data. For RGB we analysis this LSB replacement technique [6] that replace least two significant bits of each channels Red, Green or Blue with message bits. Altering the LSBs will only cause minor changes in color, and thus is usually not noticeable to the human eye.

A. A LSB-based Embedding Algorithm

Get cover C
for i = 1 to Length(c), do
Sj ← Cj
for i = 1 to Length(m), do
Compute index j_i where to store the i^{th} message bit of m
$S_{ji} \leftarrow LSB(C_{ji}) = m_i$
End for
Output -: Stego image S

A LSB-based Extracting Algorithm

Input -: Secret image s
for i = 1 to Length (m), do
Compute index j_i where to store the i^{th} message bit of m
$m_{ji} \leftarrow LSB(C_{ji})$
End for

In the extraction process, the embedded messages can be readily extracted without referring to the original cover-image from the given stego-image S. The set of pixels storing the secret message bits are selected from the stego-image, using the same sequence as in the embedding process. The n LSBs of the selected pixels are extracted and lined up to reconstruct the secret message bits

B. LSB Method for 8 Bit Color Image

Consider an 8-bit color image (cover image) where each pixel is stored as a byte representing a grayscale value. Suppose the first three pixels of the original image have the following values:

[11 00 1001 11011110 11101001]

And that of secret image :

[11100101 10110110 11110001]

Firstly removing the last 4 LSB of cover image by multiplying each pixel with [11110000]. In Matlab it is done by using "bitand" command, resulting original image pixels [11000000 11010000 11100000]. After shifting secret image pixels bits by 4 toward right and then adding with original cover image pixel by using "bitor" command, last 4 LSB of cover image get replaced by first 4 MSB of secret image and we get stego-image, whose first three pixels are :

[1100**1110** 1101**1011** 1110**1111**]

Before Histogram Equalization (Cover, Secret and Stego image)	After Histogram Equalization (Cover, Secret and Stego image))
	

Fig.1 Before and After Histogram

Histogram - Embedded image of 8 bit, Embedded image of 24 bit color image and Embedded image of 24 bit color image by using GANKAN method)

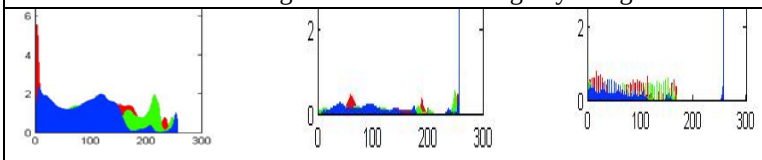

Fig.2 Histogram-Embedded image of 8 bit & 24 bit

C. LSB Method for 24 Bit Color Image

In the case of 24 bit color image each pixel is composed of RGB values and each of these colors requires 8-bit for its representation. [R (8 bits), G (8 bits), B (8 bits)]. If we read color image in Matlab and display its first pixel, we get-:

a = imread('coverimage.jpg')
a (1,1,1) = 220
a (1,1,2) = 198
a (1,1,3) = 135
Here last term (1, 2 and 3) represent RGB component of pixels (1,1) So RGB first pixel can be represent as

[11011100 11000110 10000111]

For embedding secret image whose first pixel is [11001001] firstly we have to replace last 2 LSB of each of RGB component and then embedding first 2 MSB of first pixel of secret image to R component, then next 2 MSB of first pixel of secret image to G component and lastly another next 2 MSB of first pixel of secret image to B component. In this way we get stego image whose first pixel is:

[11011111 11000100 10000010].

In this method 6 bits of secret image get hide by replacing only 2 bits of RGB component so stego-image is visually indistinguishable from the original cover-image in the case of 24 bit

D. GANKAN – A new method for hiding 24 bit color image

In this method cover image is 24 bit color image. This cover image is first split into its 3 plane (RGB). The main aim of this method is to hide most of the secret image bits in blue plane rather than red and green. One survey reveals that 65% of human eyes are sensitive to R, 33% to G and only 2 % to B, as a result visual perception of intensely B objects is less distinct that the perception of objects of R and G.

Steps in the new method and implement in MATLAB are-:

1) a) Select the cover image
   b) multiply the R plane by 254 // used to make last bit 0: by bitand(c(x,y,1), uint8(254))
   c) Obtain first MSB // 8th bit of secret image and then embedded it in last LSB of R plane

2) a) Take G plane of cover image and convert its last 2 LSBs to 0
   // by using bitand command,
   // multiplying each pixel with 252: by using bitand (c(x,y,2), uint8(252)).
   b) Obtain next to MSB // i.e 7th and 6th bit of secret image
   c) Embedded it into G plane.

3) Take B plane of cover image and convert its last 3 LSBs to 0
   // by using bitand command
   // multiply each pixel by 248: "bitand(c(x,y,3), 248)".
   b) obtain next 3 MSB i.e. 5th, 4th and 3rd bit of secret image and
   c) embedded it into blue plane

So if 24 bit color image first pixel is represent as [11011100 11000110 10000111]
Then for embedding secret image whose first pixel is [11001001], we follow above step and get stego image whose first pixel is: [11011101 11000110 10000010].

E. Histogram Analysis

Here we compare the histogram of all stego image with original one. Histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors. After comparing histogram of cover image with all stego image it is quite clear that histogram of stego image of 24 bit color image is almost similar to cover image. i.e there is almost no change or almost negligible change in color intensity. But histogram of stego image of 8 bit is different. This shows that LSB method is best for 24 bit color image rather than 8 bit color image.

V. SIMULATION RESULT

In this GANKAN method PSNR, MSE and RMSE are standard measurement used to test the quality of the embedded images. MSE measures the average of the squares of the errors. The error is the amount by which the pixels value difference between the embedded-image and the cover image. PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The signal here is the cover image, and the noise is the error introduced by bits of secret image. If the value of PSNR is higher, the quality of the stego image will be more. Let us consider, the cover image C of size $M \times M$ and the embedded image is S of size $N \times N$, then each cover image C and embedded image S will have pixel value (x, y) from 0 to M-1 and 0 to N-1 respectively.
PSNR = 10 log 10 Max^2 / MSE (db)
Where
MSE = 1/MN ($\alpha_{i,j}$ - $\beta_{i,j)}$
Here, $\alpha_{i,j}$ is the pixel of the cover image, and $\beta_{i,j}$ is the pixel of the embedded-image. M and N represent the size of the image. A larger PSNR value indicates that the difference between cover image and the embedded-image is more invisible to the human eye.

TABLE I PSNR VALUE INDICATES THAT THE DIFFERENCE BETWEEN COVER IMAGE AND THE EMBEDDED-IMAGE

Tools	8 bit color image	24 bit color image	24 bit color image by GANKAN
PSNR	30.49	40.62	41.73
MSE	46.98	4.58	3.52
RMSE	6.85	2.14	1.87

VI. CONCLUSION

In this paper we have described 2 way of applying LSB method to 24 bit color image and then compare its result with 8 bit color image. For comparison we have used PSNR, MSE, RMSE and histogram analysis which shows that PSNR for 24 bit color image is higher than that of 8 bit color image and MSE is quite low, that is embedded image quality becomes better. Also histogram comparison shows that in the case of 24 bit color image, embedded-image is visually indistinguishable from the original cover-image. In 8 bit color image 4 MSBs out of total 8 bits of secret image get embedded in cover image so while extracting secret image back we get an image that contain only 4 bits i.e it got slightly distorted. But in another case, first 6 MSB of secret image get embedded in RGB component, so while extracting secret image back we get an image containing 6 MSB, since LSB doesn't contain any information there is no loss of information and secret image recovering back become undistorted.

REFERENCES

[1] M. M Amin, M. Salleh, *S.* Ibrahim, M.R.K atmin, and M.Z.I. Shamsuddin, "Information Hiding using Steganography," *National Conference on Telecommunication Technology Proceedings*, Shah Alam, Malaysia. *IEEE*, 2003.

[2] T. Moerland, *Steganography and Steganalysis*, Leiden Institute of Advanced Computing Science, www.liacs.nl/home/ tmoerl/privtech.pdf

[3] J.B. Feng, I.C. Lin, C.S. Tsai, Y.P. Chu, . Reversible watermarking: current status and key issues. *International Journal of Network Security* 2 (May), pp. 161– 170, 2006.

[4] H. Wang, and S. Wang, "Cyber warfare: Steganography vs. Steganalysis", *Communications of the ACM*, 47:10, October 2004.

[5] A. Westfeld, and G. Wolf, Steganography in a Video conferencing system, *in proceedings of the second international workshop on information hiding*, Vol. 1525 of lecture notes in computer science,Springer, 1998. pp. 32-47.

[6] C.K. Chan and L.M. Cheng, "Hiding data in images by simple LSB substitution", *Pattern Recognition* 37 (March), pp. 469–474, 2004.

[7] T. Sharp, "An implementation of key-based digital signal steganography," in *Proc. Information Hiding Workshop*, Vol. 2137, Springer LNCS, pp. 13–26., 2001.

[8] R. Chandramouli and Nasir Memon, "Analysis of LSB Based Image Steganography Techniques", *IEEE* 2001.

[9] Neil F. Johnson and S.C. Katzenbeisser,"A survey of steganography technique" .

[10] Chung-Ming Wang a and Nan-I Wu a, "A high quality steganographic method with pixel-value differencing and modulus function", accepted 24 January 2007.

[11] Mohammad Tanvir Parvez and Adnan Gutub, "RGB Intensity Based Variable-Bits Image Steganography", *APSCC 2008 – Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference*, Yilan, Taiwan, 9- 12 December 2008.