Improved Hash Based Approach for Secure Color Image Steganography using Canny Edge Detection Method

Saurabh Singh, Ashutosh Datar
PG Student, Department of E & C Engineering, S.A.T.I. Vidisha, INDIA (M.P.)
Head, Department of Bio-Medical Engineering, S.A.T.I. Vidisha (M.P.)

ABSTRACT
In this paper, we proposed a hash based approach for color image steganography using canny edge detection method. One of the advantages of using edge detection technique is to secure our data. Proposed methodology follows encoding and decoding procedure. For encoding the data in an image we follow a steganography procedure. Steganography is the art of hiding information like-text, audio, video etc. in which anyone can't predict the difference between the original images and the encoded image. Firstly we take different type of color (RGB) image and text data. Edge detection is done by canny method and then hash function is used to embed text data in the image. We take text data for hiding. We have used Matlab2010a version for simulating the results. For edge detection in any image canny edge detection gives better results. We prefer larger edge detected image for secure steganograpgy. Now a day's security aspect is one of the essential issues for any type of work. So our proposed method provides a better security. This work is supports different types of file format like-jpg, jpeg, bmp, tiff etc. The hash is a fast and secure approach for image Steganography. Our proposed method is more robust than previous one [19].

Keywords
Edge detection, Steganography, Encoding & Decoding, Hash function, Mtalab2010a.

1. Introduction

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message [1-4]. Is is derived from the two words-

STEGANOS–"Covered"
GRAPHIE– "Writing"

Below shows the figure1of stegosaurus: a covered lizard-

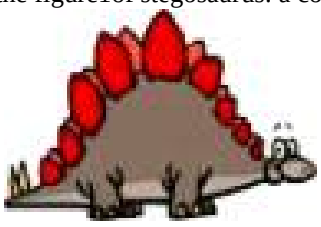


Fig.1. Stegosaurus: a covered lizard (But not a type of cryptography)

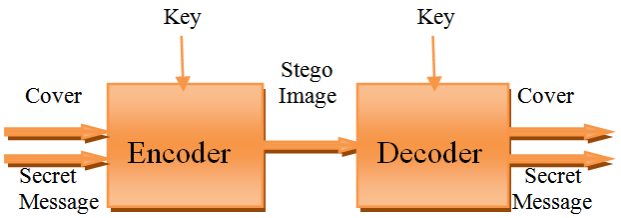


Fig.2. Block diagram of Steganography

The first step in embedding and hiding information is to pass both the secret message and the cover message into the encoder. Inside the encoder, one or several protocols will be implemented to embed the secret information into the cover message. The type of protocol will depend on what information you are trying to embed and what you are embedding it in.

For example, you will use an image protocol to embed information inside images. A key is often needed in the embedding process. This can be in the form of a public or private key so you can encode the secret message with your private key and the recipient can decode it using your public key. In embedding the information this way, you can reduce the chance of a third party attacker getting hold of the stego object and decoding it to find out the secret information.

In general the embedding process inserts a mark, M, in an object, I. A key, K, usually produced by a random number generator is used in the embedding process and the resulting marked object, I, is generated by the mapping-

$$I \times K \times M - I$$

Having passed through the encoder, a stego object will be produced. A stego object is the original cover object with the secret information embedded inside. This object should look almost identical to the cover object as otherwise a third party attacker can see embedded information. Having produced the stego object, it will then be sent off via some communications channel, such as email, to the intended recipient for decoding. The recipient must decode the stego object in order for them to view the secret information. The decoding process is simply the reverse of the encoding process. It is the extraction of secret data from a stego object. In the decoding process, the stego object is fed in to the system. The public or

Manuscript received July 5, 2015
Manuscript revised July 20, 2015

private key that can decode the original key that is used inside the encoding process is also needed so that the secret information can be decoded. Depending on the encoding technique, sometimes the original cover object is also needed in the decoding process. Otherwise, there may be no way of extracting the secret information from the stego object. After the decoding process is completed, the secret information embedded in the stego object can then be extracted and viewed. The generic decoding process again requires a key, K, this time along with a potentially marked object, I'. Also required is either the mark, M, which is being checked for or the original object, I, and the result will be either the retrieved mark from the object or indication of the likelihood of M being present in I'. Different types of robust marking systems use different inputs and outputs. A possible formula of the process:

Cover medium + Secret message + Stego key= Stego-medium

There are lots of algorithm are available for image steganography like LSB (Least significant bit method), masking and filtering etc. Least significant bit method is the simplest and most popular method for data hiding. Researchers are focus in a long time but the security aspect is not good at all.

A hash function is any algorithm that maps data of variable length to data of a fixed length [16]. The values returned by a hash function are called hash values, hash codes, hash sums, checksums or simply hashes. One practical use is a data structure called a hash table where the data is stored associatively. Hash function is used to generate a pattern, which is random selection of edge pixels, for more security and better stego-image quality. The benefit of using this approach is that each time data is coded; data is coded on a new pattern that makes the coding of data very efficient. In this paper our effort to produce a high capacity and higher quality stego images under human visual system (HVS).

2. EDGE DETECTION

Edges contain shape information [5]-[7]. Our goal is to extract a "line drawing" representation from an image, useful for recognition. Edge detection is the process of finding meaningful transitions in an image. It is a problem of fundamental importance in image analysis. The purpose of edge detection is to identify areas of an image where a large change in intensity occurs. These changes are often associated with some physical boundary in the scene from which the image is derived. Edge detector produce set of edge contains- correct edges corresponding to real - useful information e.g., object boundaries and false edges do not correspond to real edges e.g., noise. Goal of edge detection is to Produce a line drawing of a scene from an image of that scene, Important features can be extracted from the Edges of an image (e.g., corners, lines, curves).These features are used by higher-level computer vision algorithms (e.g., recognition). Edge detection is an active area of research as it facilitates higher level image analysis. There are three different types of discontinuities in the grey level like point, line and edges. Spatial masks can be used to detect all the three types of discontinuities in an image. Below figure shows the edge detection example-

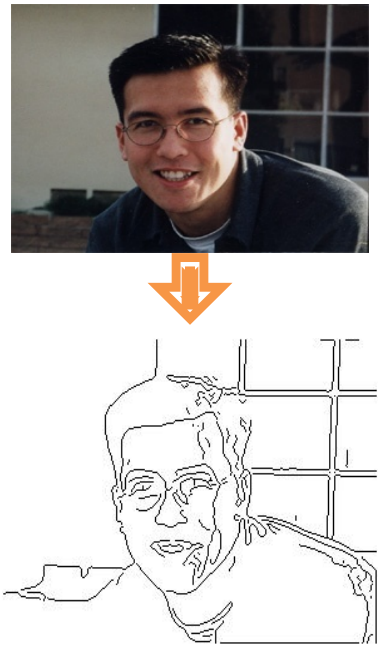


Fig.3. Edge detection example

Four steps of edge detection-

(1) Smoothing: suppress as much noise as possible, without destroying the true edges.

(2) Enhancement: apply a filter to enhance the quality of the edges in the image (sharpening).

(3) Detection: determine which edge pixels should be discarded as noise and which should be retained (usually thresholding provides the criterion used for detection).

(4) Localization: determine the exact location of an edge (sub-pixel resolution might be required for some applications, that is, estimate the location of an edge to better than the spacing between pixels). Edge thinning and linking are usually required in this step.

Different types of edge detection techniques [8]-[12] are available, described below-

2- A. Roberts Edge Detection

It was one of the first edge detectors and was initially proposed by Lawrence Roberts in 1963. It performs a simple, quick to compute, 2-D spatial gradient

measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. The input to the operator is a grayscale image the same as to the output is the most common usage for this technique. Pixel values in every point in the output represent the estimated complete magnitude of the spatial gradient of the input image at that point.

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

2- B. Sobel Edge Detection

The Sobel operators are named after Erwin Sobel. The Sobel operator relies on central difference, but gives greater weight to the central pixels when averaging. The Sobel operator can be thought of as 3×3 approximations to first derivative of Gaussian kernels. Sobel operators which are shown in the masks below:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

2- C. Prewitt Edge Detection

The Prewitt operators are named after Judy Prewitt. Prewitt operator based on the idea of central difference. The Prewitt operator measures two components. The vertical edge component is calculated with kernel Gx and the horizontal edge component is calculated with kernel G_y. $[G_X + G_Y]$ Give an indication of the intensity of the gradient in the current pixel.

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

2- D. Kirsch Edge detection

The Kirsch operator or Kirsch compass kernel is a non-linear edge detector that finds the maximum edge strength in a few predetermined directions. It is named after the computer scientist single mask and rotating it to eight main compass directions: North, Northwest, West, Southwest, South, Southeast, East and Northeast. Kirsch operator represented by the mask:

$$E = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad NE = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

$$N = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad NW = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$W = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad SW = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$S = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad SE = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

2- E. Kiresh Edge detection

Kiresh operator represented by the mask:

$$W_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad W_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad W_4 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

2- F. Laplacian of Gaussian Edge detection (LOG)

This LOG operator smooths the image through convolution with Gaussian-shaped kernel followed by applying the laplacian operator. Laplacian of Gaussian edge detection mask is:

$$G_x = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

2- G. Laplacian Edge detection

The Laplacian of an image f(x, y) is a second order derivative defined as:

$$\Delta^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

3×3 Laplacian operator mask below-

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

2- H. Canny Edge detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny's aim was to discover the optimal edge detection algorithm. In this situation, an "optimal" edge detector means:

- Good detection – the algorithm should mark as many real edges in the image as possible.
- Good localization – edges marked should be as close as possible to the edge in the real image.
- Minimal response – a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

This method can be summarized below:

1. The image is smoothed using a Gaussian filter with a specified standard deviationσ, to reduce noise.
2. The local gradient and edge direction are computed at each point using different operator.
3. Apply non-maximal or critical suppression to the gradient magnitude.
4. Apply threshold to the non-maximal suppression image.

5. Edges tracking by hysteresis/edge linking-Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

Idea: Maintain two thresholds khigh and klow

5.1. Use k_{high} to find strong edges to start edge chain

5.2. Use k_{low} to find weak edges which continue edge chain. Typical ratio of thresholds is roughly

$$K_{high} / k_{low} = 2$$

5.3. Another method for edge linking is Hough transform-

In edge detection technique the resulting pixels seldom characterized an edge completely because of noise, breaks in the edge from non-uniform illumination and other effects that introduce spurious intensity discontinuities. Thus edge detection technique algorithms typically are followed by linking procedures to assemble edge pixels into meaningful edges. One method used for this type of problem that is Hough transforms (Hough [13]-[15]). Steps for this-

1. Peak detection-first step for line detection and linking is peak detection. For each peak, the first step is to find the location of all nonzero pixels in the image that contribute to that peak.

2. Line detection and linking-once a set of peaks has been identified in the Hough transform, it remains to be determined if there are line segment associated with those peaks, as well as they start and end.

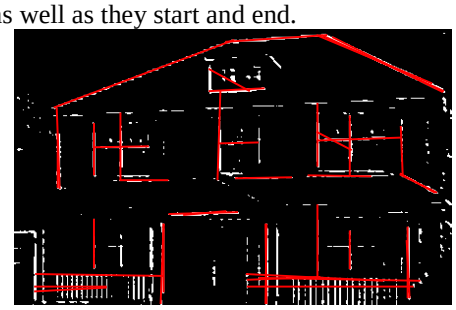


Fig.4. Hough transform based edge detection

3. Hash Function

To create a digest of the message, we use hash function [16]-[18]. The hash function creates a fixed digest from a variable –length message as shown in below figure-

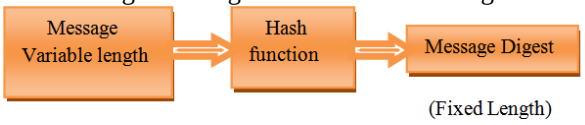


Fig.5. Signing the digest

The two most common hash functions are called MD5 (Message Digest 5) and SHA-1 (Secure Hash Algorithm 1). The first one produces a 120-bit digest. The second produces a 160-bit digest. Hash functions have must two properties to guarantee is success.

1. Hashing is one-way; the digest can only be created from the message, but not vice-versa.

2. Hashing is one-to-one function; there is little probability that two message will create the same digest.

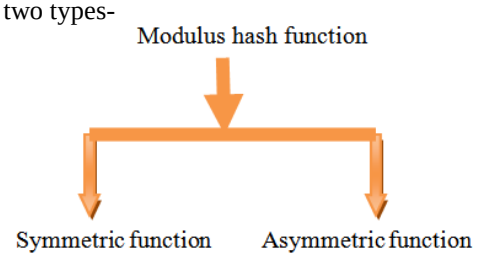
One practical use is a data structure called a hash table where the data is stored associatively. Searching for a person's name in a list is slow, but the hashed value can be used to store a reference to the original data and retrieve constant time (barring collisions). Another use is in cryptography, the science of encoding and safeguarding data. It is easy to generate hash values from input data and easy to verify that the data matches the hash, but hard to 'fake' a hash value to hide malicious data. This is the principle behind the Pretty Good Privacy algorithm for data validation. Hash functions are also used to accelerate table lookup or data comparison tasks such as finding items in a database, detecting duplicated or similar records in a large file, finding similar stretches in DNA sequences, and so on.

Different types of hash function are available but most types of hashing function the choice of the function depends strongly on the nature of the input data. Types of hash function-

1. Trivial hash function
2. Perfect hash function
3. Minimal perfect hash function
4. Cryptographic hash function
5. Hashing with checksum
6. Hashing variable length data
7. Modulus hash function etc.

In this paper I have used Modulus hash function that is the mod of the division hash function could be h = z mod n (the remainder of z divided by n). This is the combination of addition, subtraction, multiplication and division function.

It is of two types-



1. Symmetric hash function-This type of hash function insure that sender and receiver used the same key for data encoding and decoding.

2. Asymmetric hash function- This type of hash function insure that sender and receiver used the different-different key for data encoding and decoding.

In our methodology process used symmetric modulus hash function for text data hiding and retrieval purpose in the Red, Green and Blue pixels of the image.

4. Approach for Steganography

This approach allows the user to embed their secret textual information in images in a way that can be invisible and doesn't degrade or affect the quality of the original image [2], [4]. Users want to make their information secure or protect their work form other or illegal use. This approach is able to manipulate with different file formats e.g. bitmap, jpeg, jpg, gif and tiff etc.
Below I have describe the steganography steps in the image-

1. Input Image-An input interface (see fig.6) is provided so that a user can input a (bmp, jpg, gif or tiff etc.) image in which he/she wants to hide his/her personal data for privacy purposes.
2. Input Textual Data- Input the text file containing the textual data which the user wants to code in the image. The input text file is read by our system (see fig.6).
3. Coding Data in Image-For coding textual data in the image, a hash-based algorithm is used. Basic purpose of using the hash-based algorithm is to pick pixels randomly to store text data. The text data are stored in red, blue and green pixel of the image.
4. Decoding Data from Image-For decoding textual data in the image, the hash key is used that was generated during coding. The pixels (red, green, blue byte) values of each position are read one by one and generated characters concatenated to form a complete message string.

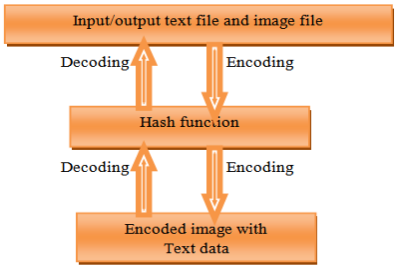


Fig.6. Basic block diagram of color image Steganography using hash algorithm

5. Proposed Methodology

In this method I have proposed two steps for Steganography [19]-[22]-
1. Encoding,
2. Decoding

5-1. Encoding Procedure-

Proposed Algorithm for Encoding Data in Image:-
Step 1: Read the RGB image I of any size.
Step 2: Detect the edges of the input image by canny method.
Step 3: Find all edge pixels and use these edge pixels as the hash Key (K).
Step 4: Read the text file (.txt), store the data in an array list.
Step 5: The hash function uses the hash key (K), input image and the text data to generate a pattern i.e. sequence of hash values those are the position of the pixels where data will be stored.
Step 6: Afterward replace text data, to the Red, Green and Blue pixels of the image.
Step 7: Finally output image containing coded data.

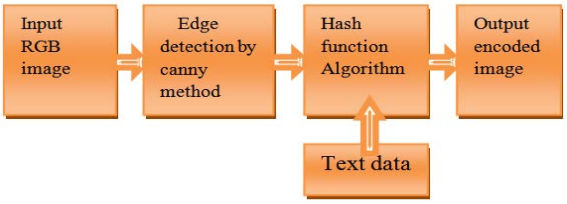


Fig.7. Block diagram of encoding model

5-2. Decoding Procedure-

Proposed Algorithm for Decoding Data in Image:-
Step 1: Read the RGB image that contains encoded information.
Step 2: Input hash key is used with hash function to generate the pattern (hash value) where data has been stored.
Step 3: Same as encoding method to encode text into the image, follow for decoding text from image. Values of red, green and blue-byte are read one by one, further store in the form of string.
Step 4: The output is the text file that contains the decoded data from the image.

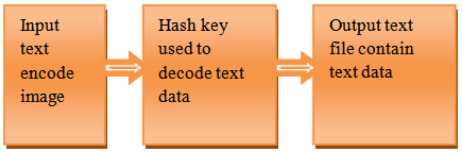


Fig.8. Block diagram of decoding model

6. Image Quality Parameter

Image quality is a characteristic of an image that measures the perceived image degradation (typically, compared to an ideal or perfect image) [5]. Two parameters are there:
1. MSE
It is defined as the squared difference between the original image and estimated image.

$$\text{MSE}=\frac{1}{N}\sum_{i=1}^{N}[X-\hat{X}\,]\text{^}2$$

2. PSNR (Peak signal to noise ratio):- Peak Signal-to-Noise Ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation[10]. Because many signals

have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most easily defined via the mean squared error (MSE):

$$PSNR=10.\log_{10}(L^2/MSE) = 20.\log_{10}(L^2/\sqrt{MSE})$$

Where: L = maximum value, MSE = Mean Square Error, X = original value, $\hat{X}$ = stego value and N = number of sample

7. Simulation Results

In our proposed methodology we have taken three different color image Lena, Peppers and Baboon of standard size. Simulation results are performed in Matlab2010a version. Below figure 9 shows the original image of size 512*512 pixels, figure 10 shows the edge detected of original image, figure 11 shows the encoded image with text data of 2547 bytes, figure 12 shows the histogram of original image, figure 13 shows the histogram of encode image with text data of 2547 bytes. With the help of figure 9 and 11 anyone can't predict the difference between the original and encoded image. Also according to the histogram of figure 12 and 13 we cannot see any difference between of them. Thus only with the help of PSNR (Peak Signal to Noise Ratio), It is clearly we say that this is the steganographed image or not. Below in Table 1 & 2, PSNR is calculated for all three types of standard size image. In Table 1 Shows the previous method results by [19] and in Table 2 proposed results.

Lena.jpeg Peppers.jpeg Baboon.jpeg

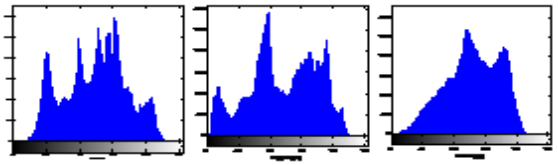
Fig.9. Original images of size 512x512

Lena.jpeg Peppers.jpeg Baboon.jpeg

Fig.10.After Applying Canny Edge Detection Algorithm

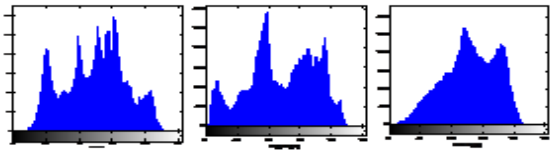
Lena.jpeg Peppers.jpeg Baboon.jpeg

Fig.11. Encoded Images with Text Data (2547 Bytes)



Lena.jpeg Peppers.jpeg Baboon.jpeg

Fig.12. Histogram of original image



Lena.jpeg Peppers.jpeg Baboon.jpeg

Fig.13. Histogram of encoded image with text data (2547 Bytes)

Table I. PSNR output for text encoding [19] (Previous Method Results)

S.N.	Text Data	Previous Method Results(1)		
		Lena	Peppers	Baboon
1.	849 Bytes	46.7704	42.4704	44.5141
2.	1698 Bytes	43.1161	39.8468	41.4249
3.	2547 Bytes	40.4854	37.9358	37.7590
4.	3396 Bytes	39.5810	36.7382	36.3541
5.	4287 Bytes	38.5342	35.7352	35.1693

Table II. PSNR output for text encoding (Proposed Method Results)

S.N.	Text Data	Proposed Method Results		
		Lena	Peppers	Baboon
1.	849 Bytes	47.5599	43.7486	46.5188
2.	1698 Bytes	43.7728	41.9565	44.6801
3.	2547 Bytes	41.5473	40.3967	43.2978
4.	3396 Bytes	40.0503	39.4238	42.3406
5.	4287 Bytes	39.6310	38.6527	41.5895

Graphical Results of Calculated PSNR

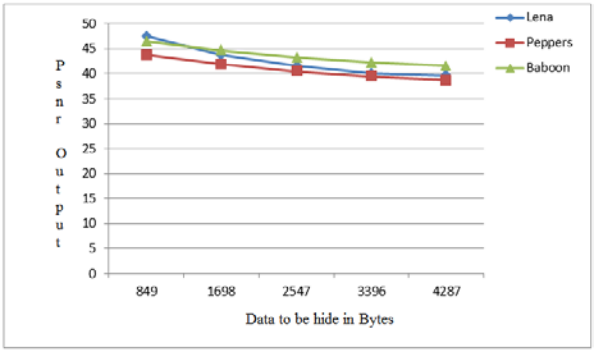


Fig.14.The graph shows that as the data to be hide increases PSNR decreases, baboon image gives better results than Lena and peppers image

8. CONCLUSION & FUTURE WORK

The proposed method produced better results than previous method [19]. In previous paper text data replaces only the blue pixels of the image but in our proposed work text data replaces red, green and blue pixels of the image. Proposed method produce high capacity and higher quality Stego-images under human visual system. Simulation results shows that the proposed scheme achieve Stego-image of good quality than the previous method. With the help of graph we show that the text data of 1250 bytes hidden in Lena image gives better results than Peppers and Baboon image. If size of text data greater than above value, Baboon image gives better results than Lena and Peppers image. Thus whenever we require to send a data, use a larger edge detected image for better security. The proposed method is good for security aspect because if any unauthorized users see the image and guess this is the Steganography image; try to extract the hidden information. If he knows what type of hash function is used for hiding the data, he can't extract the original message. Because in hash function we have used a constant, whenever any users don't know he can't do it. So the proposed method is best for security aspect as well as gives good PSNR values than previous method. This Color image (RGB) Steganography technique can be extending to further audio, video etc.

Acknowledgment

My special thanks to Dr. S.N. SHARMA (Head, Department of E & C Engineering) & Dr. ASHUTOSH DATAR (Head, Department of Bio-Medical Engineering).

REFERENCES

[1] J. Krinn, "Introduction to Steganography," 2000, URL:http://rr.sans.org/covertchannels/steganography.php

[2] W. Bender, "Techniques for Data Hiding," IBM Systems Journal, Vol. 35, no. 7, Pgs 313-336, 1996

[3] Stego Archive, "Steganography Information, Software and News to enhance your Privacy", 2001, URL: www.StegoArchive.com

[4] Wang Qian, et .al," Steganography and Steganalysis based on digital image," IEEE 4th International Conference on Image and Signal Processing, Shanghai, 15-17 Oct. 2011, pp.252-255.

[5] Sarbjeet Singh, et. al,"A new image Steganograpgy based 2k correction method and canny edge detection," IEEE Fifth international Conference on Information Technology: New Generation, Las Vegas, NV, 7-9 April 2008, pp.563-568.

[6] D. Marr, E. Hildreth, "Theory of edge detection," Proc. Royal Society of London, vol. 207, no. 1167, pp.187–.217, Feb.1980.

[7] J. Canny, "A computational approach to edge detection," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol 8, no 6, pp. 679-698, June 1986.

[8] D. Ziou, S. Tabbone, "Edge detection techniques: An overview," International Journal of Pattern Recognition and Image Analysis, vol.8, no.4, pp.537–559, 1998.

[9] D. Marr, E. Hildreth, "Theory of edge detection," Proc. Royal Society of London, vol. 207, no. 1167, pp.187–.217, Feb.1980.

[10] J. Canny, "A computational approach to edge detection," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.8, no.6, pp. 679-714, June 1986.

[11] D. Ziou, S. Tabbone, "Edge detection techniques: An overview," International Journal of Pattern Recognition and Image Analysis, vol.8, no.4 pp.537–559, 1998.

[12] S Jayaraman, S Esakkirajan and T Veerakumar, "Digital Image Processing," Tata McGraw Hill Education ptd. Ltd, New Delhi, 7th ed., 2012, pp.368-393.

[13] R. S. Wallace, A modified Hough transform for Line, in IEEE CVPR Conf., San Francisco, 19-23, June, 1986, pp.665-667.

[14] R. D. Duda, P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures," association of computing machinery (ACM), pp.11-15, Jan 1972.

[15] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins,"Digital Image Processing Using MATLAB,"Pearson Education ptd. Ltd, Singapore, 3rd ed., 2005, pp.392-425.

[16] From Wikipedia, "Hash function," July 2010,URL: http://en.wikipedia.org/wiki/Hash_function

[17] Behrouz A. Forouzan,"Data communication and networking,"Tata McGraw-Hill Publication, 2nd ed., 2003, pp.799-800.

[18] William Stallings," Cryptographic and network security," Pearson Prentice Hall Publication, 4th ed., 2006, pp.320-375.

[19] Kritika Singla, Sumeet Kaur, "A Hash Based Approach for secure image stegnograpgy using canny edge detection method," International journal of computer science and communication , vol.3, no.1,pp.156-157,June 2012.

[20] Rubata Riasat, et. al, "A Hash Based Approach for Color Image Steganography," IEEE Sixth International Conference on Digital Information Management, Melbourne, Australia, 26-28 Sept. 2011, pp. 102-107.

[21] Wen-Jan Chen,et al, "High Payload Steganography Mechanism Using Hybrid Edge Detector," Expert Systems with Applications , ELSEVIER,vol.37, pp. 3292-3301, July 2010.

[22] Seyed Mohammad Seyedzade, et. al, "A Novel Image Encryption Algorithm Based on Hash Function," IEEE 6th Iranian Conference on Machine Vision, Isfahan,27-28 Oct. 2010, pp.1-6.