International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 3, ISSN No. 2455-2143, Pages 109-113
Published Online July 2019 in IJEAST (http://www.ijeast.com)

AN IMAGE STEGANOGRAPHY METHOD USING CANNY EDGE DETECTION FOR HIDING DATA IN IMAGE USING HSI COLOR SPACE

Hiriyanna G S
Department of CSE
JNNCE, Shivamogga
Karnataka, India

Sayyed Johar
Department of CSE
JNNCE, Shivamogga
Karnataka, India

Kavya D N
Department of CSE
JNNCE, Shivamogga,
Karnataka, India

Abstract— **This paper presents a new technique for image steganography in a HSI color cover images, hiding the secret image/message in its edges of the carrier image using 2-bit LSB substitution for embedding the message/image. Edge mapping takes place by mapping image pixels with corresponding threshold selection. Higher the threshold then higher hiding capacity. In order to get the true edges, Canny Edge Detection technique has been used. Where, the amount of embedding is high compared to other techniques. Embedding plays an important role on selection of edges. The major benefit of using HSI color model is, it produces an image with significantly large file size, hence the user can hide large amount of secret image/message, Experimental results have shown that the proposed technique performs better and provides high embedding capacity than existing systems.**

Keywords— **HSI color model, true edge, 2bit-LSB, image steganography, edge detection, hiding text/ image.**

I. INTRODUCTION

Steganography, from the Greek, means covered or secret writing, and is a long-practiced form of hiding information. Although related to cryptography, they are not the same. Steganography is the art and science of hiding communication; a stenographic system thus embeds hidden content in unremarkable cover media so as not to eavesdropper's suspicion. In the past, people used hidden tattoos or invisible ink to convey stenographic content[11].

Steganography's intent is to hide the existence of the message, while cryptography scrambles a message so that it cannot be understood.

More precisely, "the goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present[1]. Steganography includes a vast array of techniques for hiding messages in a variety of media. Among these methods are invisible inks, microdots, digital signatures, covert channels and spread-spectrum communications. Today, thanks to modern technology, steganography is used on text, images, sound, signals, video, and more [12].

In present, steganography categorize in different medium such as audio, video, images, or text file to secretly hide any information in it therefore it does not illustrate any interest and hence looks like a safe medium. Digital image, video, audio and photo become the first choice as cover medium[13]. Stego is the media that helps to contain secret data whereas the cover media are the plain file.

These days, the images turn into a most mainstream decision as a methods for cover medium primarily due to its excess to speak to and the capacity to get over applications in our day by day life. In most recent couple of years, numerous calculations are as subject of research. In our work, we have built up another system for steganography in RGB images. Data are covered up into the images in vector space. The cover picture is chosen and the secret message is inserted in it.

The information to be hidden in the cover data is known as the ``embedded" data. The ``stego" data is the data containing both the cover signal and the ``embedded" information [16]. Logically, the processing of putting the hidden or embedded data, into the cover data, is sometimes known as embedding. Occasionally, especially when referring to image steganography, the cover image is known as the Steganography container.

The objective of the present project is to use a new technique for steganography in a HSI color cover images, which hides secret message in the edges of the carrier images using 2-bit LSB substitution for embedding [15]. To get true edges, canny edge detection technique has been used. Amount of data to be embedded plays an important role on the selection of edges. The figure 1 shows the basic steganography system.

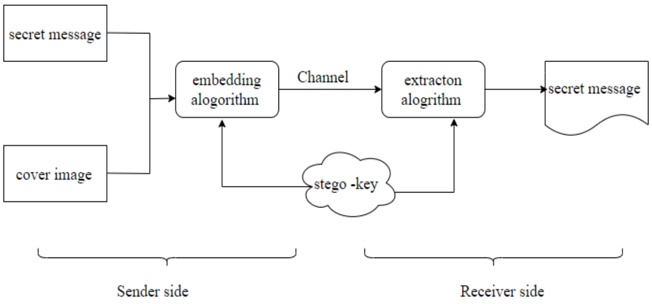

Fig. 1. An Example for Steganography

II. METHODOLOGY

In this paper a new technique for image steganography in a HSI color cover image is proposed, which hides secret message in the edges of the carrier images using 2-bit LSB substitution for embedding. To get true edges, canny edge detection technique has been used. Amount of data to be embedded plays an important role on the selection of edges. The main advantage of using HSI color model is that it produces image with a significantly large file size hence we hide large amount of secrete message. The figure 2 shows the block diagram of image steganography using HSI color model.

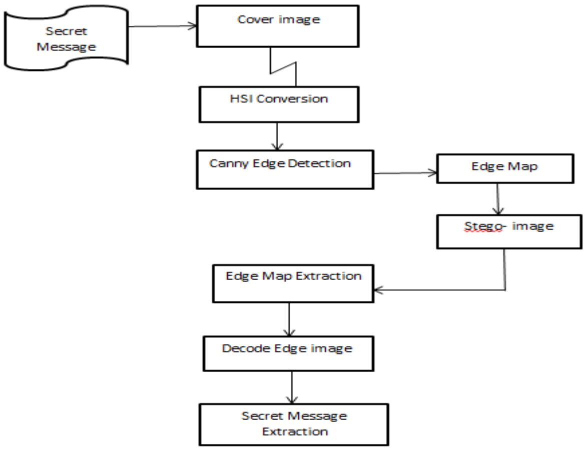

Fig. 2. Block Diagram of Image Steganography

Image processing, computer vision and machine vision generally need edge detection process as a very important tool, mainly in the area of feature detection and feature extraction as edges are the main features for analysis of the most necessary contained information in an image. The methodology of acquiring meaningful transitions in an image, is known as edge detection. The points where sharp changes in the brightness takes place usually from the boundaries between different separate objects. Many classical edge operators are now available in the literature of image processing. Such as, Sobel Edge Detector[4] ,Prewitt Edge Detector ,Robert Edge Detector , Laplacian of Gausian (Log) Edge Detector , Canny Edge Detector ,Fuzzy Edge Detector.

Among the above all edge detection methods , the most efficient, popular and widely used edge detection is the canny edge detection method. Good detection, good localization, and single response to an edge are the three most important attributes of canny edge operator, for which it selected as the best among the other available operator.

A. HSI (Hue, Saturation and Intensity)–

The HSI color space is very important and attractive color model for image processing applications because it represents color s similarly how the human eye senses colors. The HSI color model represents every color with three components: hue (H), saturation (S), intensity (I). The below figure 3 illustrates how the HIS color space represents colors. The Hue component describes the color itself in the form of an angle between [0,360] degrees. 0 degree mean red, 120 means green 240 means blue. 60 degrees is yellow, 300 degrees is magenta. The figure 4 shows the HSI color space with color intensities. The Saturation component signals how much the color is polluted with white color. The range of the S component is [0,1]. The Intensity range is between [0,1] and 0 means black, 1 means white.

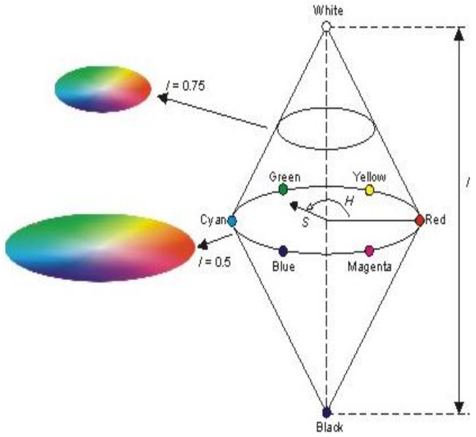

Fig. 3: HSI color model

B. Algorithm for Edge Detection

Step 1: Firstly, take a color image.
Step 2-Refining: Refining is used to remove the noise as possible without the damage of the true edges of it.
Step 3-Intensification: Apply differentiation to enhance the quality of edges.
Step 4 -Threshold: Edge magnitude threshold is used to reject the noisy edge pixels and other should be confined.
Step 5-Localization: Some applications to estimate the location of an edge and spacing between pixels, sub pixels resolution might be required.
Step 6: Get the image after edge exposures.

C. Canny Edge Detection

The Canny edge detection technique is one of the standard edge detection techniques. It was first created by John Canny for his Master's thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. To find edges by separating noise from the image before find edges of image the Canny is a very important method. Canny method is a better method without disturbing the features of the edges in the image afterwards it applying the tendency to find the edges and the serious value for threshold. Figure 4 shows the example of canny edge detection.

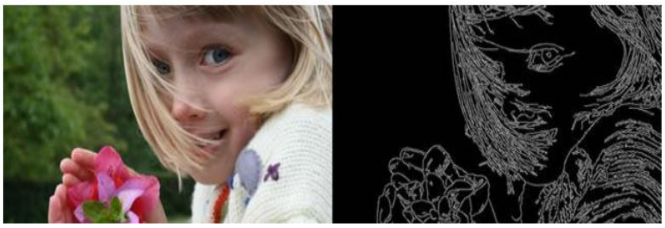

Fig. 4. Canny edge detector

The algorithm runs in 5 separate steps:

Step 1 -Smoothing: Blurring of the image to remove noise.

Step 2- Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.

Step 3- Non-maximum suppression: Only local maxima should be marked as edges.

Step 4- Double thresholding: Potential edges are determined by thresholding.

Step 5- Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

D. Edge map

Edge map is used to represent the vector fields of an image[14]. The vector fields can be represented in Edge Map using getEdgeMap method. It provides the continuous flow of vector mapping entry in sequence order. It provides entry as well as exit points which can be useful in determining the image contents. Edge Mapping is very robust and error free technique for storing points of image into it. Using edge map to store the edges of images to hide secret message into it. This will help to utilize the full capacity of image to store information which can be transferred to other person secretly.

III. PROPOSED ALGORITHM

Algorithm for Threshold Selection, Embedding and Decoding Technique

a) Algorithm for getting threshold.

```
getThresholdValue(I, N, w)
Data: I: Image, N: Length of augmented message to be embedded, w: width of the Gaussian Kernel.
Result: threshold: threshold th for Canny to get N pixels
Step 1 : tl =0.4*th and w
Step 2: 0.01*N
Step 3: threshold_max ← 1 and threshold_min ← 0
Step 4: set ←false
Step 5: repeat step 1.
        th ←[(tmax+tmin)]/2;
       ne ←getEdgePixelCount(Canny(I , th, tl, w));
Step 6 : diff ←ne-N;
Step 7: if diff >limit the then
Step 8: then goto step 3 [threshold_min ←th]
Step 9:end
Step 10: else if diff<0 then
Step 11: goto step 3 [threshold_max←th]
        end
Step 12: else goto step 4
        end until set =true; return threshold (th)
```

b) Embedding of message

Notations used in embedding secret message algorithm, embedding (I, M , P , w). some data's that are incorporated are , Data : I = image, M= Augmented message in binary, P= stego key, w= width of the Gaussian kernel. The results that are obtained are, S = stego image. Figure 3.9 shows the flow chart of embedding the message.

Initially, S assigned to Image, image contains bitand (I, 252), L ←|M| , threshold← -get Threshold(I , L , w); e← -CannyEdgeDetection (I, th, tl, w); // shuffle e and s using stego key p. e← - randomPermutation (e,P); S← - randomPermutation (S,P); index ← 0.

Algorithm for embedding

```
Step 1: start
Step 2: for each edge pixel i in e, do
Step 3: S x,y =bitand (S x,y,252)
Step 4: S x,y = S x,y +2*Mindex +1+ Mindex
Step 5: index ← index+2
Step 6: End
```

After embedding the edge pixel the threshold and width of the image must be calculated and embedded properly in order to get the true edge at the decoding phase. The algorithm for embedding the pixel with appropriate threshold and width is given below.

Embedding the Threshold:

```
Step 1 : **Start**
Step 2: embed with appropriate threshold
Step 3: for i= 1:16 in e' do
Step 4: S( x,y) = bit AND operation (S x,y,254)
Step 5: S( x,y) = S x,y + th(i)
End
```

i. Embedding the Width :

```
Step 1: embed with the appropriate width
Step 2: for I = 17:32 in e' do
Step 3: S( x,y) =bitand (S x,y,254)
Step 4: S x,y = S x,y +w(1-16)
End
```

c) Decoding technique

After encoding of secret message into the image, the receiver side decoder requires. The decoder construction is same as the encoder side algorithm. If there is mismatch with any of the algorithm used on encoder side, than the decoding process cannot be successfully carried out.

Decoding of secret messages: decodes and retrieve the secret message or image. some inputs that are required are, I: stego imge, T: Threshold, P : stego key, w: kernel width. The result after decoding process is secret message or image.

Stego image contains image with S🡐 -bitand (S,252); with threshold 🡐 -T; tl 🡐 -0.4*th; e🡐 -Canny (S', th, tl, w); e 🡐 -randmPermutation (e,P); // shuffle S to get order of embedding. S 🡐 -randomPermuations (S,P); with index 🡐 0.

Algorithm for decoding

```
Step 1: Start
Step 2: for each edge pixel i in e , do
Step 3: val 🡐 bitand (S x,y,3)
Step 4: Mindex +1 🡐 val mod 2
Step 5: val 🡐 val/2
Step 6: Mindex = val
Step 7: index 🡐 index +2
Step 8: End
```

To extract first C bits to get message size. Msg_size 🡐 Message[1:C]
Message 🡐 Message [C+1 : msg_size]; return (Message); returns the secret message/ image.

IV. EXPERIMENT AND RESULT

The proposed methodology is tested and run successfully on MATLAB software. The proposed work gives the better results compared with other existing techniques as well [4]. It has high efficiency and better hiding capacity and yields much better results. Performed the experiment and analyzed using PSNR and MSE metrics and also time required for algorithm to run.

$$PSNR= \frac{-10\log10 \ e \ MSE}{S2}$$

(Eq 1)

$$MSE = \frac{1}{MN} \sum_{n=1}^{M} \sum_{m=1}^{N} [g\hat{}(n,m) - g(n,m)]^{2} \qquad \text{(Eq 2)}$$

Table I: Set of input images

Table II: Comparison of Results with Existing System and Proposed Method

Cover image	Secret image	Existing method		Proposed method	
		PSNR	MSE	PSNR	MSE
Lena	Lena	51.132	180.496	102.87	0.466
Peacock	Berry	12.212	153.345	102.445	0.490
Tulip	Apple	11.682	169.97	101.638	0.538
Zebra	Zebra	36.831	936.514	104.906	0.3696
Berry	Berry	48.554	242.850	121.373	0.0555
Bonsai	Apple	48.495	55.745	105.98	0.326
Bonsai	Apple	48.495	55.745	105.98	0.326
Pepper	Berry	13.395	139.855	106.227	0.3174
Sea	Peacock	17.564	860.739	102.938	0.4636
Nature	Apple	13.298	146.778	102.599	0.4820
Berry	Bonsai	56.367	57.985	102.466	0.489

The above table 2 shows the comparison between some previous existing methods and the proposed method[4][12][15]. Which yields the better and efficient results as shown above. Higher in the peak signal to noise ratio the quality , efficiency of the image is maintained same throughout the encoding and decoding process. Lower the MSE (mean square error) value lower distortion created during the encryption and decryption process. Hence lower MSE and higher PSNR metrices must be maintained to get best results.

V. CONCLUSION

In this work, developed a new technique for steganography in RGB images. Information are hidden into the images in vector space. The cover image is selected and the secret image is embedded in it. Data are secretly hidden into edges which are dynamically selected based on text size or image size. The proposed method is free from structural attack as it uses 2 bit LSB technique. The 2 bit LSB technique is free from structural attack. Canny Edge detection is used to detect the edges based on threshold. For example described in table 1 consider peacock as a cover image and berry as secret image. The existing method obtained 12.21 PSNR and 153.34 MSE value. While the proposed method provide better result comparatively. It provides 102.44 PSNR and 0.49 MSE value. The various threshold levels for an image. For example low and high threshold value. The higher will be the threshold value the higher is the space allocated by an image. Means with higher threshold value we can hide more number of textual information into image.

VI. REFERENCES

[1] Akash Modi, Manu Bansal, (2015),“International Journal of Advanced Research in Computer Science and Software Engineering,” vol. 5, pp. 2277–2128, 2015.

[2]. Abhilasha Ramdas Bhagat, Ashish B Dhembhare(2015), “An Efficient and Secure Data Hiding Technique – Steganography,” vol. 3, pp. 2320–9798, 2015

[3]. Reza Tivoli,Maryam bhakshi, Fatemeh salehian,(2015) “A New Method for Text Hiding in the Image by Using LSB,” vol. 7, pp. 126-132, 2016

[4]. Smitha GL1, Baburaj E,(2017) “Sobel edge detection technique implementation for image steganography analysis, 2017.

[5]. Sneha Arora., Sanyam Anand(2013) “A New Approach for Image Steganography using Edge Detection Method, 2013”

[6]. Sadaf Bukhari, Mohammad shoaib arif,(2016), M R Anujum., “Enhancing security of image by steganography and cryptography technique,” in Computing, Management and Telecommunications (ComManTel), 2016 International Conference on, pp. 309–314, IEEE, 2016.

[7]. Rashmi A. sonawane, Mrs. Ditpti sonawane,(2017) “Reversible texture synthesis using three level security in steganography” in Vol 3 National Conference on, pp. 52–55, ILSRST, 2017.

[8]. Sujarani Rajendran., Doraipandian(2017) ‘Choatic map based random image steganography using LSB technique,. International Journal of Network Security, Vol.19, No.4, PP.593-598, July 2017.

[10]. Jaspreet kaur.,Gurbinder singh brar.(2014), Dr. Rahul Malhotra “Steganography techniques on edges,” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181mVol. 3 Issue 11, November-2014

[11]. "Detecting LSB Steganography in Color and Gray-Scale Images" Jessica Fridrich, Miroslav Goljan, and Rui Du State

University of New York, Binghamton

[12]. P. Kruus, C. Scace,M. Heyman, and M. Mundy. (2003), "A survey of steganography techniques for image files." Advanced Security Research Journal.

[13]. M.M. Amin, M. Salleh, S. Ibrahim, et al., "Information Hiding Using Steganography", 4th National Conference on Telecommunication Technology Proceedings (NCTT2003), Shah Alam, Malaysia, 2003

[14]. Saiful Islam,Mangat R Modi and Phalguni Gupta,(2014) “Edge-based image steganography”, EURASIP Journal on Information Security a Springer open Journal,2014.

[15]. Mukesh Garg and A.P. Gurudev Jangra,(2014) “An Overview of Different Type of Data Hiding Scheme in Image using Steganographic Techniques” ,International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, January 2014.

[16]. Arun Navjot Kaur and Manpreet Singh,(2015) “Modified Approach Using Lsb In Image Steganography” International Journal Of Research–Granthaalayah, Vol.3(Iss.5):,May,2015.