ISSN: 0976-9102(ONLINE)
DOI: 10.21917/ijivp.2012.0055

EFFICIENT ADAPTIVE STEGANOGRAPHY FOR COLOR IMAGES BASED ON LSBMR ALGORITHM

B. Sharmila[1] and R. Shanthakumari[2]
[1]*Department of Computer Science and Engineering, Erode Builder Educational Trust Institutions, India*
E-mail: sharmilame@yahoo.com
[2]*Department of Information Technology, Kongu Engineering College,India*
E-mail: rsk_shan@yahoo.com

Abstract
Steganography is the art of hiding the fact that communication is taking place, by hiding information in other medium. Many different carrier file formats can be used, but digital images are the most popular because of their frequent use on the Internet. For hiding secret information in images, there exists a large variety of steganographic techniques. The Least Significant Bit (LSB) based approach is a simplest type of steganographic algorithm. In all the existing approaches, the decision of choosing the region within a cover image is performed without considering the relationship between image content and the size of secret message. Thus, the plain regions in the cover will be ruin after data hiding even at a low data rate. Hence choosing the edge region for data hiding will be a solution. Many algorithms are deal with edges in images for data hiding. The Paper 'Edge adaptive image steganography based on LSBMR algorithm' is a LSB steganography presented the results of algorithms on gray-scale images only. This paper presents the results of analyzing the performance of edge adaptive steganography for colored images (JPEG). The algorithms have been slightly modified for colored image implementation and are compared on the basis of evaluation parameters like peak signal noise ratio (PSNR) and mean square error (MSE). This method can select the edge region depending on the length of secret message and difference between two consecutive bits in the cover image. For length of message is short, only small edge regions are utilized while on leaving other region as such. When the data rate increases, more regions can be used adaptively for data hiding by adjusting the parameters. Besides this, the message is encrypted using efficient cryptographic algorithm which further increases the security.

Keywords:
Least Significant Bit(LSB) Based Steganography, Pixel Value Difference (PVD), Data Embedding, Data Extraction, Embedding Unit(EU), Vigenere Table

1. INTRODUCTION

Information hiding is one of the important areas of information security, which includes various methods like cryptography, steganography and watermarking. In Cryptography encryption is done results in a disordered and perplexing message. Though the message cannot read by the third party but attract eavesdroppers easily. Steganography overcome this problem by hiding the secret information behind a cover media (video, audio or image) because the presence of information cannot be noticed by any attacker. The goal of steganography is to embed secret data into a cover in such a way that no one apart from the sender and intended recipients even realizes there is a secret data. A few key properties must be taken into consideration when creating a digital data hiding system.

- Imperceptibility: Imperceptibility is the primary goal of steganography. When a person views a cover image, he or she should be unable to distinguish the image with embedded information from an image without embedded information. The goal is that the before and after images appear identical.
- Embedding Capacity: Capacity refers to the amount of information that can be embedded using a particular system. Capacity is often an issue in steganography; however, one may want to secretly transmit a long message, so the capacity of a steganographic algorithm may become a significant factor.
- Robustness: Robustness refers to the degree of difficulty required to destroy embedded information without destroying the cover work itself.
- Undetectability: Detectability refers to the ability to determine whether or not a cover medium contains embedded information using statistical or technological means. Undetectability is nearly as important a goal as imperceptibility in steganographic systems because steganography seeks to disguise the fact that a message is being transmitted.

This paper is arranged as follows Section 2 describes the limitations of relevant approaches and propose some strategies. In Section 3 data embedding and data extraction details are given. Section 4 discusses the experimental results.

2. ANALYSIS OF LIMITATIONS OF RELEVANT APPROACHES

There are lot of algorithms that deals with the image steganography till now. Mainly the LSB algorithm is the early one that was simplest and effective in the implantation of steganographic concepts. In this scheme, only the LSB plane of the cover image is overwritten with the secret bit stream. This was done by comparing each message bit with the LSB of each image pixels. This is shown in Fig.1.

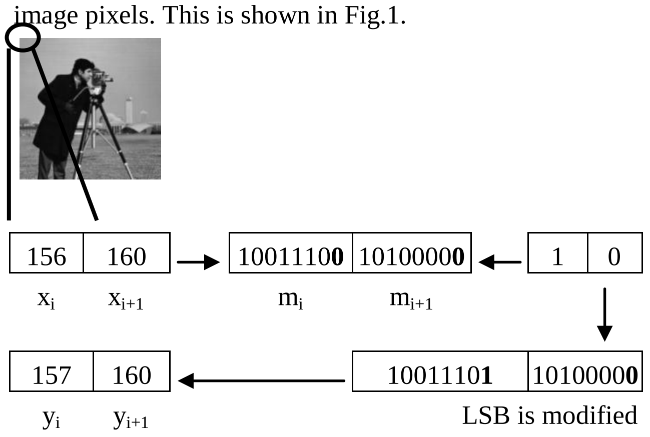


Fig.1. Example for LSB replacement

Letter A can be hidden as follows. Consider these 9 pixels,

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

The binary value for A is 01000001. Inserting the binary value for A in the three pixels would result in

(00100110 11101001 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

The underlined bits are replaced bits. Thus only three bits actually changed in the 8 bytes used. On average, LSB requires that only half the bits in an image be changed. It is very easy to detect the existence of hidden message even at a low embedding rate using steganalytic algorithms, such as the Chi-squared attack [5], and Regular/Singular groups (RS) analysis [6]. The LSB matching (LSBM) scheme employs a minor modification to LSB replacement. If the secret bit does not match the LSB of cover image, then +1 or -1 is randomly added to the corresponding pixel value. Several steganalytic algorithms have been proposed to analyze the LSBM scheme. Harmsen.J [7] uses the center of mass (COM) of the histogram characteristic function (HCF) for steganalysis. In LSB Matching Revisited (LSBMR) [2] uses a pair of pixels as an embedding unit, in which the LSB of the first pixel carries one bit of secret message, and the relationship (odd–even combination) of the two pixel values carries another bit of secret message.

The typical LSB-based approaches, including LSB replacement, LSBM, and LSBMR, deal with each given pixel/pixel pair without considering the difference between the pixel and its neighbors. Many Edge adaptive schemes have been came in practice which is based on the fact that Human eye can easily detect the changes in the flat regions, while it is very difficult to identify the changes in the edge regions. Hempstalk.K [8] proposed a hiding scheme by replacing the LSB of a cover according to the difference values between a pixel and its four touching neighbors. Although this method can embed most secret data along sharper edges and can achieve more visually imperceptible stego. But, the security performance is poor. Since the method just modifies the LSB of image pixels when hiding data, it can be easily detected by the existing steganalytic algorithms, such as the RS analysis. Another method given by singh et al., [4] first employs a Laplacian detector on every nonoverlapping block within the cover to detect edges, and then performs data hiding on center pixels whose blocks are located at the sharper edges according to a threshold. Here the maximum embedding capacity of such a method is relatively low. Furthermore, the threshold is predetermined and thus it cannot change adaptively according to the image contents and the message to be embedded.

The pixel-value differencing (PVD) based scheme [3] uses another kind of edge adaptive scheme, in which the number of embedded bits is determined by the difference between a pixel and its neighbor. The larger the difference, larger the number of secret bits that can be embedded. To a certain extent, existing PVD-based approaches are edge adaptive since more secret data is embedded in the busy regions. They are poor at resisting some statistical analyses. In most of the steganographic methods mentioned above the pixel/pixel-pair selection is mainly determined by a PRNG while neglecting the relationship between the image content and the size of the secret message. These methods can spread the secret data over the whole stego image randomly even at low embedding rate. Assuming that a cover image is made up of many nonoverlapping small subimages (regions) then different regions usually have different capacities for hiding the message. We should use those good hiding characteristics of edges in images while leaving the others unchanged. Therefore, deciding how to select the regions is the main key issue. The idea given by Weiqi Luo et al., [1] Edge adaptive Image steganography based on LSB Matching Revisited Algorithm find the edge region using difference between two consecutive pixels that satisfying the threshold. Though this scheme is good enough to embed message in edges it uses the threshold value up to 30 only and it cannot make full use of edge information for data hiding. Also this algorithm is analyzed for grey scale images only.

In this paper, an efficient adaptive scheme based on LSBMR algorithm for color images with some modifications to the parameters (threshold & block size) used in Edge adaptive scheme [1] is analyzed. Experimental results are also shown for different color images. The performance is compared on the basis of PSNR and MSE

3. PROPOSED SCHEME

Most steganographic approaches usually assume that the LSB of natural covers is insignificant and random enough, and thus those pixels/pixel pairs for data hiding can be selected freely using a PRNG. Such an assumption is not always true, especially for images with many smooth regions. Generally, the regions located at the sharper edges present more complicated statistical features and are highly dependent on the image contents. Moreover, it is more difficult to observe changes at the sharper edges than those in smooth regions. In this paper, an efficient edge adaptive scheme using LSBMR algorithm for color image is analyzed.

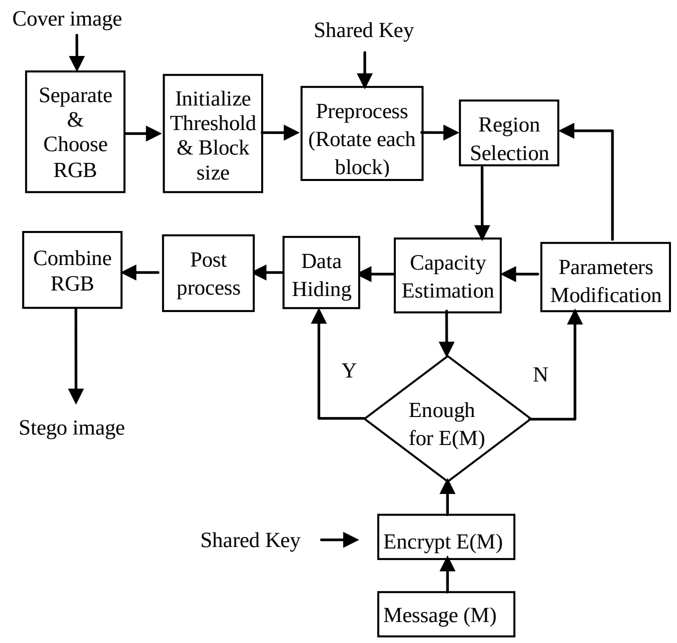


Fig.2. Data Embedding

The flow diagram of our proposed data embedding and extraction is illustrated in Fig.2 and Fig.3.

In the data embedding stage, first separate RGB component of chosen image, then decide which of the RGB channel to hide data. Now initializes some parameters, which are used for subsequent data pre-processing and region selection. The message is encrypted using any of cryptographic algorithms. Now, estimate the capacity of those selected regions. If the regions are large enough for hiding the given encrypted message, then data hiding is performed on the selected regions. Some post processing is done to obtain the stego image. Otherwise change the parameters, and then repeat the process of region selection and capacity estimation until encrypted message can be embedded completely. Here the secret message is hidden using parameters are different for different image content and encrypted message. We need them as side information to guarantee the validity of data extraction. Finally combine the entire RGB component, resulting in stego color image. In this paper, adaptive scheme to the spatial LSB domain is used. Absolute difference between two adjacent pixels is chosen as the criterion for region selection and use LSBMR as the data hiding algorithm. LSBMR applies a pixel pair (x_i, x_{i+1}) in the cover image as an embedding unit. After message embedding, the unit is modified as (x'_i, x'_{i+1}) in the stego image which satisfies[1],

$$LSB(x'_i) = m_i,\ LSB((x'_i/2) + x'_{i+1}) = m_{i+1}$$

Here the function LSB(x) denotes the LSB of the pixel value x. m_i and m_{i+1} are the two secret bits to be embedded.

In data extraction, the scheme first separate and choose the RGB component then extracts the parameters from the stego image. Based on the side information, it then does the pre-processing and identifies the regions that have been used for data hiding. It obtains the encrypted data according to the corresponding extraction algorithm. Finally, decryption is performed on the extracted data to obtain the secret message.

In data extraction, it first generates a travelling order by a PRNG with a shared key. For each embedding unit along the order, two bits can be extracted. The first secret bit is the LSB of the first pixel value, and the second bit can be obtained by calculating the relationship between the two pixels.

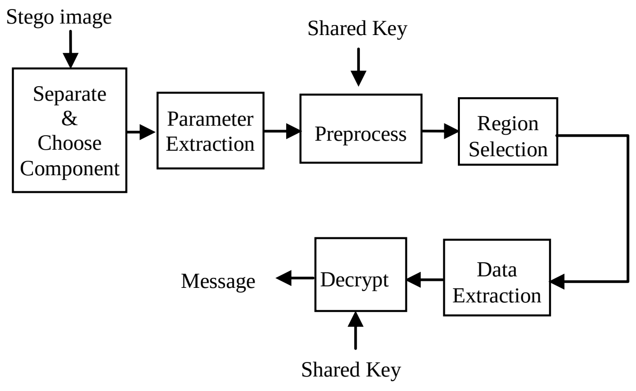


Fig.3. Data Extraction

The details of the data embedding and data extraction algorithms are as follows,

3.1 DATA EMBEDDING

STEP 1: Separate RGB component

Any image is made up of pixels each carrying different colors. For the color image, the pixel color is given by the combination of red, green and blue (RGB) component respectively. The first step in this proceeding is to separate this RGB component. For example it is illustrated in Fig.4 lenna image.

STEP 2: Choose the component

Decide which of the RGB component to hide the data; this is based on the shared key **key1**. Here, more than one component or all the three components may involve in hiding the data.

STEP 3: Parameter Initialization

The cover image of size of m x n is first divided into nonoverlapping blocks of Bz x Bz pixels. The Block size is randomly selected from the set of {2, 4, 8, 16, 32, 64, 128, 256}.

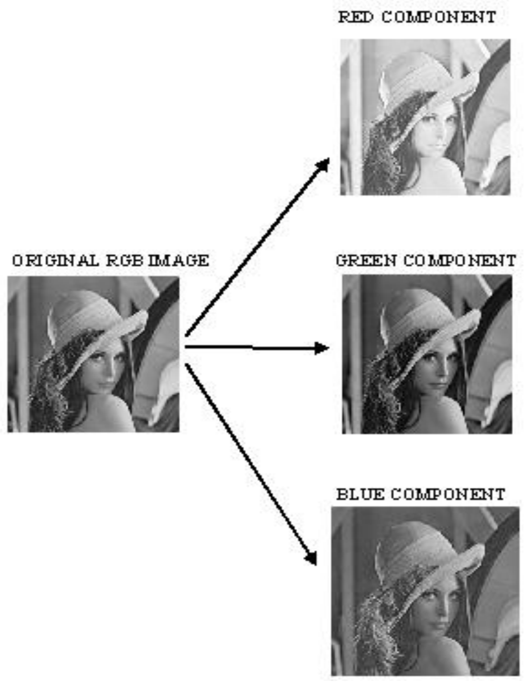


Fig.4. Separating Component

STEP 4: Preprocess

For each small block, rotate it by a random degree in the range of {0, 90,180,270} as determined by a secret key key_2.

STEP 5:

The resulting image is rearranged as a row vector V by raster scanning. These vector is divided into nonoverlapping embedding units with every two consecutive pixels, (x_i ,x_{i+1}) where i=1,3……….., m, n-1 assuming 'n' is an even number.

Two benefits can be obtained by the random rotation.

a) It can prevent the detector from getting the correct embedding units without the rotation key, and thus security is improved.

b) Both horizontal and vertical edges (pixel pairs) within the cover image can be used for data hiding.

STEP 6: Encryption

a) Get the input message and the symmetric key **key$_3$**.

b) Encrypt the given message using polyalphabetic cipher. It is performed as follows

i. Repeat the key if key length is less than plain text length.

ii. Construct the vigenere table.

iii. Use this table to find the cipher text for the given key and plain text.

c) Convert these cipher text into ascii value and then to binary bits.

STEP 7: Region Selection

According to the scheme of LSBMR, 2 secret bits can be embedded into each embedding unit. Therefore, for a given encrypted message E(M), the threshold T for region selection can be determined as follows. Let be EU (t) be the set of pixel pairs whose absolute differences are greater than or equal to a parameter t.

$$EU(t) = \{(x_i, x_{i+1}) \| x_i - x_{i+1} | \geq t, \text{ for all } (x_i, x_{i+1}) € V\} \quad (2)$$

STEP 8: Capacity Estimation

We calculate the threshold by

$$T = \max\{2 \times | EU(t)| \geq | E(M) |\} \quad (3)$$

where, T € {0,1,……200} ,|E(M)| is the size of the encrypted message E(M), EU(t) denotes the total number of elements in |EU(t)|.When, T=0, the proposed method becomes the conventional LSBMR scheme, which means that this method can achieve the same payload capacity as LSBMR. When we limits the T= 0 to 30, this method becomes Edge adaptive image steganography scheme [1].

STEP 9: Data Hiding

Perform data hiding on the set of,

$$EU(t) = \{(x_i, x_{i+1}) \| x_i - x_{i+1} | \geq t, \text{ for all } (x_i, x_{i+1}) € V\}$$

The above embedding units is processed in a pseudorandom order determined by a secret key k_3 for each unit, (x_i, x_{i+1}). These pixels pairs are converted to binary. The data hiding is performed according to the following four cases.

Case 1:

$LSB(x_i) = m_i$ & $LSB(f(x_i, x_{i+1}) = m_{i+1}$

then,

$(x_i', x_{i+1'}) = (xi, xi+1)$

Case 2:

$LSB(x_i) = m_i$ & $LSB(f(x_i, x_{i+1})) \neq m_{i+1}$ \

then

$(x_i', x_{i+1}') = (x_i, x_{i+1} + r)$. where r = ±1.

Case 3:

$LSB(x_i) \neq m_i$ & $LSB(f(x_i - 1, x_{i+1})) = m_{i+1}$

then

$(x_i', x_{i+1}') = (x_i - 1, x_{i+1})$

Case 4:

$LSB(x_i) \neq m_i$ & $LSB(f(x_i - 1, x_{i+1})) \neq m_{i+1}$

then

$(x_i', x_{i+1}') = (x_i + 1, x_{i+1})$

where m_i and m_{i+1} denote two secret bits to be embedded. The function 'f' is defined as f(a,b) = (a/2)+b. r is a random value in{-1,+1} and (x_i', x_{i+1}') denotes the pixel pair after data hiding. After the above modifications x_i' and x_{i+1}' may be out of [0,255], or the new difference $| x_i' - x_{i+1}'|$ may be less than the threshold T. In such cases, we need to readjust them as (x''_i, x''_{i+1}) by

$$x''_i = x_i' + 4k_1$$
$$x''_{i+1} = x'_{i+1} + 2k_2$$

k1 and k2 posses the value of either 0 or 1.

Finally, we have

$$LSB(x_i'') = m_i \quad \& \quad LSB(f(x''_i, x''_{i+1}) = m_{i+1}$$

STEP 10: Post process

After data hiding, the resulting image is divided into nonoverlapping Bz x Bz blocks. The blocks are then rotated by a random number of degrees based on key$_1$.The process is very similar to **Step 1** except that the random degrees are opposite. Then we embed the two parameters into a preset region which has not been used for data hiding.

STEP 11: Combine RGB.

This is the final step in forming color stego image. Now combine all the three red, green and blue pixel values of image resulting in color image with data hidden in it.

STEP 12: Preset Region

There are two parameters in this approach. The first one is the block size Bz for block, dividing the image in data preprocessing; another is the threshold T for embedding region selection. Bz, is randomly selected from the set of {1,2,4,8,32,64,128,256},and T belongs to { 0,1,…..200 } and can be determined by the image contents and the secret message.

In the preset region, 18 bits are used to indicate these parameters. First 8 bits (1 to 8) indicate the threshold value. Next 10 bits (9 to 18) is used to convey the block size information. Total of 18 bits are needed for side information for each image. The preset region chosen may be in the unused pixels or flat/smooth) region. Otherwise the region is decided between sender and receiver the suitable place to hide these parameters.

3.2 DATA EXTRACTION

STEP 1: Separate RGB component

The first step in data extraction is to separate RGB component from the stego image and use key$_1$ to decide which component is involved hidden data.

STEP 2: Parameter Extraction

To extract data, first extract the side information, i.e., the block size Bz, and the threshold from the chosen component.. Do exactly the same things as step 1 in data embedding.

STEP 3: Preprocess

The stego image is divided into Bz x Bz blocks and the blocks are then rotated by random degrees based on the secret key key$_2$. The resulting image is rearranged as a row vector V'. Finally, we get the embedding units by dividing V' into nonoverlapping blocks with two consecutive pixels.

STEP 4: Region Identification

Travel the embedding units whose absolute differences are greater than or equal to the threshold T according to a pseudorandom order based on the secret key key_3, until all the hidden bits are extracted completely.

STEP 5: Data Extraction

For each qualified embedding unit, say, (x'_I , x'_{i+1}) , where $| x'_I - x'_{i+1} | \geq T$, we extract the two secret bits m_i and m_{i+1} as follows:

$m_i = LSB\ (x''_i)$ & $m_{i+1} = LSB\ ((x_i/2)+x'_{i+1})$

STEP 6:

Convert these bits into ascii value followed by changing it into characters. Now we obtain the encrypted message. Then, apply poly alphabetic cipher algorithm using key_3 and get the secret message.

4. EXPERIMENTAL RESULTS AND ANALYSIS

This work is implemented by using MATLAB 7.0. The objective of the work is to extend the edge adaptive steganographic algorithm for grey scale images [1] to color images with some modification to parameters used and analyze the performance. The methods have been implemented on several color images. The performance of the methods have been evaluated and compared on the basis of two measures and the measures are: Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) and they are computed as follows

4.1 COMPUTING MEAN SQUARE ERROR

The mean square error (MSE) is a statistical measure of how far estimates or forecasts are from actual values. It is most often used in time series, but can be applied more widely, to any sort of statistical estimate. Here in this project it could be applied to pixel pairs, where one set is "Original" and the other is a “Stego image”.

Steps for Calculating Mean Square Error

(a) Set up the data by using two images. One is stego image (I’) and the other is original image (I).

(b) Subtract stego image values(pixel pair values) from Original image values

(c) Take the absolute value of each row. That is, if the difference is negative, remove the negative sign. If it is positive, leave it as is.

(d) Add up the absolute values

(e) Take the square of resultant

(f) Divide by total number of rows and columns.ie., m x n

$$MSE = \frac{1}{m\,X\,n}\sum_{i=1}^{m-1}\sum_{j-1}^{n-1}\left(I - I'\right)^2$$

In simple words MSE indicates average amount of modifications to the pixels.

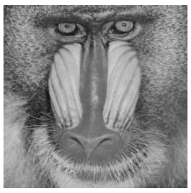
4.2 COMPUTING PSNR VALUE

The peak signal-to-noise ratio (PSNR) is the ratio between a signal's maximum power and the power of the signal's noise. Here the PSNR is used to measure the quality of reconstructed images (Stego image). Each picture element (pixel) has a color value that can change when an image is modified. Signals can have a wide dynamic range, so PSNR is usually expressed in decibels, which is a logarithmic scale. Peak Signal to Noise Ratio can be computed using the formula

PSNR = 10 * log 10 (256^2 / MSE)

In Image, PSNR value indicates the quality of stego image after modification.

These measures are calculated for several color images some of them are shown in Fig.5 and the results are shown in Table.1.


(a) Baboon

(b) Girl

(c) House

(d) Lady

(e) Lenna

(f) Couple

(g) Women

Fig.5. Sample images

Table.2 shows that effect of threshold on characters. From that we infer that if we want more data to embed threshold value chosen must be minimum, if the message size is small we can choose high values of threshold. Thus it uses the edge region of image effectively by giving first preference to more edge region followed by other regions are used adaptively. Though the edge adaptive method [1] uses the threshold range upto 30, the edge region is not efficiently used. Here in our method we extend the limits to 200, also the message are encrypted before performing data hiding. Hence this scheme is more secure. Also depends on message size the edge region is utilized.

Table.1. PSNR and MSE Values

Picture (size = 256x256)	**EU (in pixels) that satisfying Threshold (T=50)**	**No. of pixels used**	**MSE**	**PSNR**
Baboon	4207	3297	0.075027	59.12597
Girl	436	392	0.008789	68.72537

House	1166	1113	0.029205	63.51018
Lady	538	53	0.011749	67.46469
Lenna	630	630	0.013168	66.96949
Couple	308	294	0.007187	69.59934
women	800	686	0.010864	67.8048

Table.2. Effect of Threshold on Characters

Threshold value	EU(in Pixels)
170	32
150	44
130	144
100	436
80	756
60	1312
50	1846

From above table, a trend is observed that when threshold is higher we can embed only minimum set of characters. As the threshold value decreases embedding units that satisfying threshold is high hence more characters can be implanted, this is illustrated graphically as shown in Fig.6.

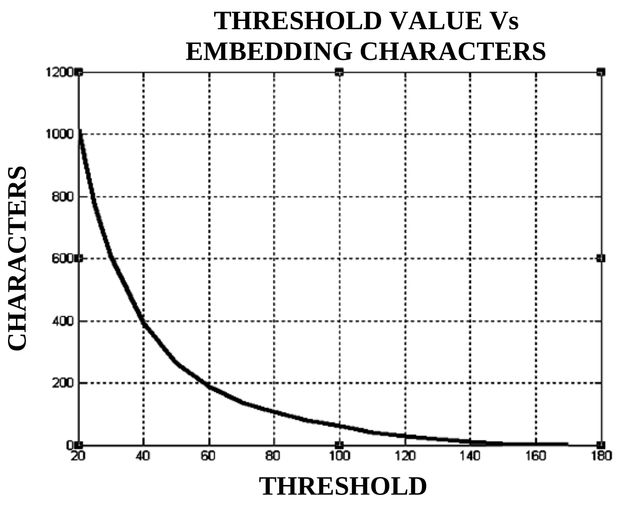


Fig.6. Graph showing the effect of threshold on characters

5. CONCLUSIONS AND FUTURE WORK

Performance of edge adaptive steganography for color image with increasing values of parameters is analyzed. In addition to that, color image is separated into RGB layers and then data hiding is performed. The image quality after data embedding is very important for better performance of steganography methods. The image quality is evaluated by Mean Square Error (MSE) and Peak Signal to Noise ratio (PSNR) for gray-scale. The following conclusions can be made from this analysis:

1) When threshold is high only minimum set of characters can be embedded. As the threshold value decreases more characters can be implanted.
2) Separation of image into RGB component results increases the embedding capacity.
3) This method encrypts the message, which improves the security
4) This scheme can be applied to other covers such as audio and video which is taken as the future work.

REFERENCES

[1] Weiqi Luo, fangjun Huang and Jiwu Huang, “Edge Adaptive Image Steganography Based on LSB Matching Revisited”, *IEEE transaction on Information forensics and security*, Vol.5, No. 2, pp. 201-214, 2010.

[2] Mielikainen J. “LSB matching revisited”, *IEEE signal Processing,* Vol.13, No. 5, pp.285-287, 2006.

[3] Yang C.H, Weng C.Y, Wang S.J, and Sun H.M , “Adaptive data hiding in edge areas of images with spatial LSB domain systems”, *Transaction on Information Forensics Security,* Vol. 3, No. 3, pp. 488–497,2008.

[4] Singh K.M, Singh L.S, Singh A.B, and Devi K.S,”Hiding secret Message in edges of the image”, *in Proceedings on International Conference on Information and Communication Technology*, pp. 238–241, 2007.

[5] Westfeld A and Pfitzmann A, “Attacks on steganographic systems”, *in Proceedings of the 3rd International Workshop on Information Hiding*, Vol. 1768, 1999.

[6] Fridrich J., Goljan J., and Du R., “Detecting LSB steganography in color, and gray-scale images”, *IEEE Multimedia*, Vol. 8, No. 4, pp. 22–28, 2001.

[7] Harmsen J. and Pearlman W., “Steganalysis of additive-noise modelable information hiding”, *Proceedings of SPIE Electronic Imaging*, Vol. 5020, pp. 131–142, 2003.

[8] Hempstalk K, ‘Hiding behind corners: Using edges in images for better steganography’, *in Proceedings of Computing Women’s Congress, Hamilton, New Zealand*, 2006.